Written by Larry Ashmun

For the Radio Shack
TRS-80* Model III

ALIEN DEFENSE COMMENTED

SSM SOFT SECTOR MARKETING, INCORPORATED

The Book Contains:

1. COMPLETE SOURCE CODE LISTING of Alien Defense (Mod III tape version.)
2. Details of one approach to game writing.
3. Explains some of the more complicated routines.
4. You are informed of the concept behind the structure of the routines.

Never Before Has Anyone Sold a Book Like This!

If you write in Machine language; if you have thought of writing a Machine language game; if you ever just wondered what a Machine language program looked like in its uncompiled form – This Is Your Book!

ALIEN DEFENSE COMMENTED

written by Larry Ashmun

IMPORTANT NOTICE

The Author, SOFT SECTOR MARKETING, INC., or any of it's distributors assume no liability with respect to the use of this publication nor any damages resulting from the use of any information contained herein. ALIEN DEFENSE COMMENTED is sold on an "as is" basis and though every effort has been made to insure the accuracy of the information it contains, it is not represented as being error free.

Introduction to ALIEN DEFENSE COMMENTED

This book is intended for those of you that have some experience in assembler language programming and need a little help getting started writing real-time (arcade style) games. It is not intended to be a detailed tutorial on assembler language programming but just a guide through the procedure of writing a game.

The source code listing in this book is for the ALIEN DEFENSE program, MODEL III TAPE version. It is the actual code that is used to create the program sold by SOFT SECTOR MARKETING, INC. at the time that the book was written.

The Editor/Assembler used to write this program was EDAS 3.5, written by ROY SOLTOFF and sold by:

MISOSYS
5904 Edgehill Drive
Alexandria, VA 22303

This particular Editor/Assembler has some special features that you will need if you intend to type in the source code. Specificaly, it allows:

Assembly from files (*GET statement).
Conditional assembly (IF statement).
Multiple data bytes on a DEFB line.
Abbreviated DEFB, DEFW, DEFM statements.
Using DB in place of DEFM.

EDAS 3.5 has many more features but you will have to have the above features in the assembler you use if you want to be able to type in the source code without making any changes to it.

Another assembler that supports some of these features is M-ZAL, written by Jeff Krantz & David Willen, sold by:

Computer Applications Unlimited
P.O Box 214, Dept. ABM
Rye, NY 10580

This assembler is also very good but the source code will have to be changed a little to use it.

The source code listing is in the standard Z80 Assembler combined OBJECT & SOURCE format explained as follows:

The first 4 columns make up the address (HEX) that the OBJECT CODE (assembled program) instruction resides at. The address is followed by a space. The address is not typed in, it is supplied by the assembler and is determined by the assembler "ORG" statement.

The next 8 columns are the OBJECT CODE instruction. All Z80 instructions do not need all 8 columns. The columns not used by the instructions are filled with blanks. The OBJECT CODE instruction is followed by a space. The OBJECT CODE is not typed in but is created by the assembler during program assembly.

The next 5 columns make up the LINE NUMBER of the SOURCE CODE. The numbers are created by the assembler when the SOURCE CODE is typed in. You will notice that the line numbers start over from time to time in the listing, this is because each MODULE of the program has its own line numbers. The SOURCE CODE for this program is too large to fit into memory all at one time so it was broken up into MODULES with only one module being in memory at a time. The assembler uses the "*GET" statement to link the modules together during assembly. The line number is followed by a space.

The next 8 columns are reserved for LABELS. Labels are assigned ADDRESS VALUES by the assembler during assembly and are used for JUMPS, CALLS, and any other references to ADDRESSES in the program. The assembler assigns the current address value to the label during assembly. The CURRENT ADDRESS would be defined as the ADDRESS that the LABELED INSTRUCTION would assemble to.

The next 8 columns are reserved for the Z80 MNEMONIC for the machine code INSTRUCTION (also called an OPCODE). A MNEMONIC is a string of characters that represent a machine code instruction and is designed to be a memory jogger for humans. The mnemonic JP is easier to remember as a JUMP instruction than the number 0C3H, which is the Z80 machine code for JUMP.

The next 8 columns are reserved for the OPERAND. Some Z80 instructions need more information than just the OPCODE itself to know what to do. This is where the additional information is placed. An example would be the JP (JUMP) instruction. The computer needs to know WHERE to jump to.

Any text following a semi-colon ";" is considered a COMMENT and is ignored by the assembler. A comment can start on any column following the line number and can be part of an instruction line or be on a line by itself.

For any additional information on EDITOR/ASSEMBLERS, you should read the manual that comes with your assembler. For additional information on the Z80 instruction set you should read the ZILOG Z80 Technical Manual.

In the four years that I've been programming I'd have to say that I've found real-time game programming to be the most enjoyable. I've found that the problems encounterd in writing game programs can be some of the most challenging and frustrating to solve.

What I hope to do in this book is show you one approach to solving those problems, using the ALIEN DEFENSE program to provide examples. I don't mean to say that my approach is the best or only approach to problem solving, it's just one way that works well for me. You may find that my way of doing things doesn't fit with the way that you like to do things and that's OK. If you can't use even one of the ideas presented here at least you will have eliminated some of the things that you can't use which brings you a little closer to finding the things that you can use.

The reason for using the combined OBJECT & SOURCE CODE listing of ALIEN DEFENSE is to make finding specific source code references a little easier. Whenever I refer to a specific section of code, I will use the MEMORY ADDRESS that it resides at. Hopefully this will be less confusing than referring to MODULES and LINE NUMBERS as the line numbers repeat from module to module and it may not be clear which module I'm talking about.

The hardest part (for me at least) of writing a game program is getting started. When I first decide what the game is going to be my head starts swimming with all of the details and problems that I will have to account for when I actually start coding the program. To avoid being overwhelmed by this mountain of details I start writing things down on paper as I think of them so that I won't forget them when it comes time to use them. When I first started programming, I read somewhere that a programmer should "THINK FIRST - CODE LATER". I've found that to be pretty good advice. I don't always follow it, though, and when I don't I always end up regretting it. It's no fun re-writing a section of code three or four times because I didn't think the problem all the way through.

I usualy start writing a game by trying to define the general rules of the game. That would include things like how many ships a player starts out with, what a player has to do to get bonus ships, are bonus points awarded for any special conditions, are there any penalties for any actions on the players part, any 'secrets' to the game that a player would have to learn to get high scores, a list of all of the characters in the game ...etc. I write all of this down on paper without doing any coding (most of the time). The following is an example of the list I used for ALIEN DEFENSE:

ALIEN DEFENSE - rules

1. Two players allowed.
2. Characters = players ship, man, lander, mutant, cruiser, pod, swarmer, bomber, bomb, shot character (alien), shot character (ship).
3. Each player starts with 3 ships & 3 smart bombs.
4. Smart bomb explodes all aliens on screen.
5. Extra ship & smart bomb for every 10,000 points.
6. Bonus for every 25 kills.
7. Players ship moves up/down/left/right.
8. Players ship has warp drive - unlimited use.
9. Ship destroyed if hit by alien or shot.
10. ...
11. ...
12. ...

After the general rules of the game have been defined I make a separate sheet for each character. I use the separate character sheets to define the general characteristics of each character. It would include how the character moves, what causes the character to appear, whether or not the character fires at the player, ...etc. Every detail of each character would be defined as well as possible. The follwing is how a character sheet might look for the LANDER character:

LANDER

1. Display character = 224.
2. Start with ten on playfield.
3. More Landers start appearing when a low limit is reached.
4. Moves left/right/up/down.
5. Seeks man character. When no man available seeks players ship.
6. Drops down and picks man up.
7. Moves to top of screen with man.
8. Changes to mutant character when it reaches the top with man.
9. Fires at players ship when close enough.
10. ...
11. ...
12. ...

The above lists are not complete but should give you an idea of what I mean. This process of writing lists helps me to organize my thoughts into an overall goal for the game. It also helps me to anticipate problems that may occur during coding.

If I started typing in code before I knew everything that the game was supposed to do and how the characters were going to interact with each other, I would end up making program design decisions without all of the information that I would need to make them. It is very easy to program yourself into a corner that way. The times that I have boxed myself in I've ended up throwing away large sections of code and starting over. That is not a very efficient way of doing things.

The next thing I usually do is decide what the focal point of the program is going to be. In SUPER VADERS it was the relentless advance of the invaders. All of the routines in that program revolve around that point. In ALIEN DEFENSE it wound up being the PLAYFIELD. The PLAYFIELD is a grid, 16 rows by 256 columns. Along the bottom four rows is a mountainous terrain that slides by as the players ship travels along the grid. Each and every character of the game is kept track of by it's position on the grid. Even the movements of the characters are made in relation to their position on the grid and not to their position on the screen.

One of the reasons for deciding on a focal point is to have continuity to the game. If compromises have to be made when the code is being written it, is better if the final decision on the compromises leans toward the overall goal of the game rather than have all of the different routines in the program fighting one another.

Until the focal point is decided upon it is hard to start writing code but sometimes I have to experiment with code to determine what the focal point is going to be. Instant BRAIN LOCK, also known as an endless loop. What I do is take the information that I have written down and try to determine all of the ways that the information could be implemented in a program. Then I throw away all of the ones that I believe are not practical (too slow, use enormous amounts of memory, have to use tricky coding, etc.). That usually leaves me with two or three practical approaches to the game.

Out of the choices left I try to pick out the one that can best account for problems that I can anticipate. This is where the experimenting comes in. I sometimes write short programs that implement the parts of the game that I belive will give me the most trouble and see how they look, work, and feel when they are running. With that I can usually make a final decision on the focal point.

After the focal point is decided on I generally make a list of the support routines that I will need. At this point it would be hard to anticipate every sub-routine that I would need but there are a few that are common to most games. These would include routines to initialize the computer housekeeping (interrupt vectors etc.), display messages, keep and display the scores, simple math, number conversions, ...etc.

ALIEN DEFENSE - rules

1. Two players allowed.
2. Characters = players ship, man, lander, mutant, cruiser, pod, swarmer, bomber, bomb, shot character (alien), shot character (ship).
3. Each player starts with 3 ships & 3 smart bombs.
4. Smart bomb explodes all aliens on screen.
5. Extra ship & smart bomb for every 10,000 points.
6. Bonus for every 25 kills.
7. Players ship moves up/down/left/right.
8. Players ship has warp drive - unlimited use.
9. Ship destroyed if hit by alien or shot.
10. ...
11. ...
12. ...

After the general rules of the game have been defined I make a separate sheet for each character. I use the separate character sheets to define the general characteristics of each character. It would include how the character moves, what causes the character to appear, whether or not the character fires at the player, ...etc. Every detail of each character would be defined as well as possible. The follwing is how a character sheet might look for the LANDER character:

LANDER

1. Display character = 224.
2. Start with ten on playfield.
3. More Landers start appearing when a low limit is reached.
4. Moves left/right/up/down.
5. Seeks man character. When no man available seeks players ship.
6. Drops down and picks man up.
7. Moves to top of screen with man.
8. Changes to mutant character when it reaches the top with man.
9. Fires at players ship when close enough.
10. ...
11. ...
12. ...

The above lists are not complete but should give you an idea of what I mean. This process of writing lists helps me to organize my thoughts into an overall goal for the game. It also helps me to anticipate problems that may occur during coding.

If I started typing in code before I knew everything that the game was supposed to do and how the characters were going to interact with each other, I would end up making program design decisions without all of the information that I would need to make them. It is very easy to program yourself into a corner that way. The times that I have boxed myself in I've ended up throwing away large sections of code and starting over. That is not a very efficient way of doing things.

The next thing I usually do is decide what the focal point of the program is going to be. In SUPER VADERS it was the relentless advance of the invaders. All of the routines in that program revolve around that point. In ALIEN DEFENSE it wound up being the PLAYFIELD. The PLAYFIELD is a grid, 16 rows by 256 columns. Along the bottom four rows is a mountainous terrain that slides by as the players ship travels along the grid. Each and every character of the game is kept track of by it's position on the grid. Even the movements of the characters are made in relation to their position on the grid and not to their position on the screen.

One of the reasons for deciding on a focal point is to have continuity to the game. If compromises have to be made when the code is being written it, is better if the final decision on the compromises leans toward the overall goal of the game rather than have all of the different routines in the program fighting one another.

Until the focal point is decided upon it is hard to start writing code but sometimes I have to experiment with code to determine what the focal point is going to be. Instant BRAIN LOCK, also known as an endless loop. What I do is take the information that I have written down and try to determine all of the ways that the information could be implemented in a program. Then I throw away all of the ones that I believe are not practical (too slow, use enormous amounts of memory, have to use tricky coding, etc.). That usually leaves me with two or three practical approaches to the game.

Out of the choices left I try to pick out the one that can best account for problems that I can anticipate. This is where the experimenting comes in. I sometimes write short programs that implement the parts of the game that I belive will give me the most trouble and see how they look, work, and feel when they are running. With that I can usually make a final decision on the focal point.

After the focal point is decided on I generally make a list of the support routines that I will need. At this point it would be hard to anticipate every sub-routine that I would need but there are a few that are common to most games. These would include routines to initialize the computer housekeeping (interrupt vectors etc.), display messages, keep and display the scores, simple math, number conversions, ...etc.

ALIEN DEFENSE - rules

1. Two players allowed.
2. Characters = players ship, man, lander, mutant, cruiser, pod, swarmer, bomber, bomb, shot character (alien), shot character (ship).
3. Each player starts with 3 ships & 3 smart bombs.
4. Smart bomb explodes all aliens on screen.
5. Extra ship & smart bomb for every 10,000 points.
6. Bonus for every 25 kills.
7. Players ship moves up/down/left/right.
8. Players ship has warp drive - unlimited use.
9. Ship destroyed if hit by alien or shot.
10. ...
11. ...
12. ...

After the general rules of the game have been defined I make a separate sheet for each character. I use the separate character sheets to define the general characteristics of each character. It would include how the character moves, what causes the character to appear, whether or not the character fires at the player, ...etc. Every detail of each character would be defined as well as possible. The follwing is how a character sheet might look for the LANDER character:

LANDER

1. Display character = 224.
2. Start with ten on playfield.
3. More Landers start appearing when a low limit is reached.
4. Moves left/right/up/down.
5. Seeks man character. When no man available seeks players ship.
6. Drops down and picks man up.
7. Moves to top of screen with man.
8. Changes to mutant character when it reaches the top with man.
9. Fires at players ship when close enough.
10. ...
11. ...
12. ...

The above lists are not complete but should give you an idea of what I mean. This process of writing lists helps me to organize my thoughts into an overall goal for the game. It also helps me to anticipate problems that may occur during coding.

If I started typing in code before I knew everything that the game was supposed to do and how the characters were going to interact with each other, I would end up making program design decisions without all of the information that I would need to make them. It is very easy to program yourself into a corner that way. The times that I have boxed myself in I've ended up throwing away large sections of code and starting over. That is not a very efficient way of doing things.

The next thing I usually do is decide what the focal point of the program is going to be. In SUPER VADERS it was the relentless advance of the invaders. All of the routines in that program revolve around that point. In ALIEN DEFENSE it wound up being the PLAYFIELD. The PLAYFIELD is a grid, 16 rows by 256 columns. Along the bottom four rows is a mountainous terrain that slides by as the players ship travels along the grid. Each and every character of the game is kept track of by it's position on the grid. Even the movements of the characters are made in relation to their position on the grid and not to their position on the screen.

One of the reasons for deciding on a focal point is to have continuity to the game. If compromises have to be made when the code is being written it, is better if the final decision on the compromises leans toward the overall goal of the game rather than have all of the different routines in the program fighting one another.

Until the focal point is decided upon it is hard to start writing code but sometimes I have to experiment with code to determine what the focal point is going to be. Instant BRAIN LOCK, also known as an endless loop. What I do is take the information that I have written down and try to determine all of the ways that the information could be implemented in a program. Then I throw away all of the ones that I believe are not practical (too slow, use enormous amounts of memory, have to use tricky coding, etc.). That usually leaves me with two or three practical approaches to the game.

Out of the choices left I try to pick out the one that can best account for problems that I can anticipate. This is where the experimenting comes in. I sometimes write short programs that implement the parts of the game that I belive will give me the most trouble and see how they look, work, and feel when they are running. With that I can usually make a final decision on the focal point.

After the focal point is decided on I generally make a list of the support routines that I will need. At this point it would be hard to anticipate every sub-routine that I would need but there are a few that are common to most games. These would include routines to initialize the computer housekeeping (interrupt vectors etc.), display messages, keep and display the scores, simple math, number conversions, ...etc.

After making all of my lists, I sit down and start coding. I start with the support routines and then go to the routines that I feel will give me the most trouble - generally the routines that deal with the focal point.

The biggest problem that I anticipated in ALIEN DEFENSE was keeping track of all of the characters. I wanted them to move and remain "true" to their character whether they were on the screen or not. I also had to come up with a way to have the terrain on the bottom four rows of the screen move in a way that gives the impression of traveling around a planet. That implies that the terrain wraps around on itself.

The best solution that I could come up with was a GRID. The grid doesn't actually exist in memory but is a system of coordinates used to keep track of the character's positions in relation with each other and in relation with the terrain. The grid is 16 rows (numbered 0 to 15) by 256 columns (numbered 0 to 255). The number of rows corresponds with the number of rows (lines) on the screen and the number of columns is four times the width of the screen. The top two rows of the screen are reserved for the score display so the top two rows of grid coordinates are not used for character positions.

The problem of the moving terrain was solved with the GRID. The terrain is on the bottom four rows (rows 12, 13, 14, and 15) and spans all 256 columns. At ADDRESS 4500H you will find two data tables. One is the GRAPHICS data for the mountain and the other is the ROW OFFSET for the GRAPHICS data. The routine that draws the mountains at ADDRESS 58F9H assumes that the data for each of these tables starts at a PAGE BOUNDRY (a 4 byte HEX ADDRESS that ends with two zeros) and that the GRAPHICS data is immediately followed by the ROW OFFSET data.

The ROW OFFSET determines which ROW on the SCREEN that the GRAPHICS data will be displayed on and assumes that the starting row is row 12 so that an offset of 0 would place it on row 12, an offset of 64 would place it on row 13, an offset of 128 would place it on row 14, and an offset of 192 would place it on row 15.

There are two variables (storage areas) that are dedicated to keeping track of where the VIDEO SCREEN is located on the grid. These are labled DPOINT and WINDOW. DPOINT (address 73BAH) stores a two byte address that POINTS to a position on the ROW OFFSET data table that is used as a starting position in the data when drawing the mountains. WINDOW (address 73BCH) stores a one byte value (0 to 255) that corresponds with where the left column of the SCREEN is positioned on the GRID.

What the DRAW routine does is draw the mountains on the bottom of the screen using a DATA POINTER that uses the DPOINT variable to determine where to start getting data from the tables and a SCREEN POINTER to place the data on the screen. The sequence is:

1. - Initialize the SCREEN POINTER to row 12 of column 0 on the screen (screen address 3F00H).
2. - Initialize the DATA POINTER to the starting position in the MOUNTAIN data using the DPOINT variable.
3. - Initialize the B register as a column counter (64 columns on the screen). This is the LOOP COUNTER.

4. - Save current values for screen pointer & counter.
5. - Erase all four rows in the current screen column.
6. - Get the ROW OFFSET data and calculate the screen position.
7. - Get the GRAPHICS data and display it at the current screen position.
8. - Retrieve current values for loop count and screen pointer.
9. - Increment screen & data pointers.
10. - Decrement loop counter and if not zero loop back to # 4.
11. - Return to caller.

If you examine the mountain DRAW routine closely, you'll see why I chose a grid width of 256 columns and why the mountain & row offset data has to start on a PAGE BOUNDARY. As the DATA POINTER (DE) is updated to the next graphics byte, only the E register is incremented (address 58A8H). When the E register is incremented past 255, it automaticaly starts back at zero, discarding the overflow. If the DE pair were incremented (INC DE) then it would be possible for the pointer to point PAST the mountain data because the overflow from the E register would be passed to the D register which would cause DE to point to the next page of memory. This could cause garbage to be put on the screen or other unpleasant things to happen.

This could be avoided by checking for the end of the table and then having the data pointer start back at the beginning of the table but this would require additional code. It could also cause problems in keeping track of the screen and character positions on the grid. With a limit of 256 columns a column number can be stored in one byte which would simplify any associated routines.

With the GRAPHICS & ROW OFFSET data starting on a PAGE BOUNDARY and each data table exactly one page (256 bytes) long all I need to do to switch back and forth between the graphics data page and row offset data page is to increment or decrement the D register (addresses 5910H and 5912H). This keeps the E register pointing to the same relative position in the page but changes the page number that the DE pair is pointing to.

To move the mountains along the bottom of the screen, all I have to do is increment or decrement the value (least significant byte only) stored in DPOINT and then call the DRAW routine. That would, in effect, move the mountains.

The only other VARIABLE associated with the DRAW routine is one labled DRFLG. The value stored there determines if the mountains are active or not. If all of a players MAN characters are destroyed, the variable is loaded with a non-zero value and the mountains are not drawn. A short delay routine is called in place of the draw routine to compensate for the time no longer used to draw the mountains. If this wasn't done, the game would speed up considerably when the mountains disappear.

With the grid, keeping track of the characters is very simple. Every CHARACTER TYPE (Lander, Cruiser, etc.) has a TABLE with an ENTRY on the table for every active character. If you go to address 5E6FH on the listing you will see how a table entry is defined for a LANDER. The entry definition is enclosed in asterisks. Each entry is made up of six bytes that define the STATUS of one character. What I want to point out at this time are the first two bytes of the entry. The first two bytes for every ENTRY on each CHARACTER TYPE TABLE are used the same way. They make up the character's POSITION (row and column number) on the GRID and any routines that affect the position of the character will have to update these two bytes. With these two bytes and the WINDOW variable, it can be determined if the character is currently on the screen, and if it is, exactly where on the screen it is located.

As is usually true with computers, if a particular approach solves some problems it will create other problems. In this case the grid makes it easy to keep track of all of the characters but complicates the screen handling.

When a character is moved on the screen it is generally erased at it's old position and then re-drawn at it's new position. That works fine when the character is kept track of by it's screen position. It doesn't work so well when the character is kept track of by a grid position and the screen can move along the grid in a different direction than, and at the same time as, the character.

For example, if the character moves one column to the left on the grid while, at the same time, the screen moves one column to the right on the grid, then the character, in effect, has moved two columns to the left on the screen. That can make it difficult to figure out where to erase the character on the screen. To complicate things further, some of the characters can move up or down at the same time they are moving left or right making the end result a diagonal move.

To solve this problem, if a character is found to be on the screen, I calculate the character's current position (before a move) on the screen and then erase three positions starting with the position to the left of the calculated position. Erasing three positions compensates for the possibliity of screen movement in either direction. Then, if the character is moving, I calculate the new grid & screen positions. The new grid co-ordinates are then saved on the characters table entry and the character is drawn on the screen. If the character is not moving, the old grid and screen positions are used.

Each CHARACTER TYPE has it's own update routine that follows this general pattern. Each character type has to have it's own update routines because all of the types move in different ways. The only routines associated with updating the characters that are shared by most of them are the routines that check if the character is on the screen, calculate it's position on the screen, and erase the character.

The routine at address 5BD7H called ONCHK takes care of finding out if the character is on the screen. If it is it calculates the screen address using the ROW and COLUMN information that is part of the characters table entry and the WINDOW variable.

The routine at address 5CD7H called MUCLR takes care of erasing all of the characters except the LANDER. The LANDER needs its own erase routine because it has the capability of carrying the MAN character which means that the MAN character would have to be erased and moved with the LANDER. The LANDER erase is at address 5EFFH and is called LNCLR.

When I'm writing a program, I usually like to write it in small sections, and test each section before going on to the next section. In ALIEN DEFENSE I wrote the mountain draw routine first with a temporary mountain move routine that checked for the left and right arrow keys and moved the mountains in the appropriate direction. That way I could see how the mountains looked and moved without having any other characters on the screen.

After the mountains looked OK I added the players ship. That turned out to be a bigger problem than I had anticipated. The ship not only moves in relation to its grid position but also moves in relation to its screen position. It has to be able to accelerate, decelerate, change direction when moving across the grid, change direction when not moving across the grid, move up and down in a straight line, move up and down diagonaly, move to either edge of the screen, catch a falling man, and a partridge in a pear tree.

To get all of these actions I needed to use more keys than just the left and right arrow keys so I wrote the keyboard scan at address 55F4H. The keyboard scan didn't start out the way it is now but slowly evolved as the ship's motion routines were developed.

One addition to the latest version of ALIEN DEFENSE is the ability to use a JOYSTICK to control the motion of the ship. That means that while checking the keyboard to see what the player wants to do I also have to check the JOYSTICK port which is incorporated into the keyboard scan.

The routines that process the ship's movement, including the draw and erase routines, start at address 5677H and end at 58F6H. The ship's movement is actually the movement of the screen across the grid. The ship itself has some limited movement on the screen, however the primary function of the motion routines is to move the screen across the grid.

The motion routines use something called SELF MODIFYING CODE, which is not considered a good programming practice. It not only can be very confusing for the programmer to write, but it can also be very confusing for the computer to execute. The reason for using it is that it can save a lot of space and speed up execution times. However it can be a nightmare to DEBUG.

Here is a general outline of what the routines do:

When the ship is at a dead stop and the thrust key (8) is pressed, the ACCELERATION routine at address 569DH is used to initialize the routines that bring the ship up to full speed.

After the ship has attained full speed the routine changes the CALL vector at address 5669H to the SLIDE BACK routine at address 56DFH. The slide back routine moves the ship backwards on the screen until it reaches a predetermined screen column.

After the ship has slid back to the limit, the CALL vector at address 5669H is changed to the FULL SPEED routine at 5677H.

When the THRUST key is released, the CALL vector at 5669H is changed back to the ACCELERATION routine and the CALL vector at 5673H is changed to the DECELERATION routine at 572DH.

After the ship has stopped moving, the vector at 5673H is changed back to the STOPPED routine at 5721H.

If the ship is moving, the THRUST key is being held down, and the CHANGE DIRECTION key (1) is pressed, the ship's direction is reversed and the CALL vector at address 5669H is changed to the routine at 5778H. This routine CALLS the DECELERATION routine until the ship has stopped and then changes the CALL vector at address 5669H back to the ACCELERATION routine, which executes its normal cycle.

While all of this is going on, the ship also has the ability to move up and down on the screen. Of course, I couldn't make it a simple up/down movement.

If the ship is stopped, the up/down movement is in a straight line.

If the ship is moving forward on the grid and the DOWN key (3) is pressed, the ship will move down and forward on the screen.

If the ship is moving forward on the grid and the UP key (2) is pressed the ship will move up and backward on the screen.

The reason for this is that when an aircraft goes into a dive, its forward speed increases and when it climbs, its forward speed decreases. Just trying to add a little realism to the game.

While the ship is doing all of the above on the screen it is also causing the screen to move across the grid. That is made evident by the mountains moving on the bottom of the screen.

The ship's movement routines have to take into account all the possible combinations of the above events. An example would be pressing the THRUST key from a dead stop and then releasing it before the ship has accelerated to full speed, hitting the CHANGE DIRECTION key before the ship has stopped, and then maybe pressing the THRUST key again while pressing the UP key. Try and figure out what the execution sequence of the code would be under those circumstances!

To top it all off, the ship can catch and carry a MAN character through all of these gyrations.

I did my best to see that all of the bases are covered but I won't guarantee that there aren't a set of conditions that will CRASH this section of code. That's one of the hazards of using self modifying code.

The primary function of the ship is to give the player a means to destroy the aliens. To do this it must be able to fire some kind of missle, in this case a LASER beam. The routine that processes this is called FIRE and is located at address 5938H.

This routine finds out which direction the ship is pointing and proceeds to draw the shot character (the underline character) starting at the nose of the ship to either the edge of the screen or until an alien character is encountered, which ever comes first. While the shot is being drawn, the number of total characters that make up the shot are counted and stored in a variable called LENCNT. The starting address on the screen and the direction of the shot are also stored in variables called SHTBL and SHTDIR respectively. These values are needed by the routine that erases the shot from the screen later in the GAME LOOP.

The reason the shot is not erased immediately after drawing it, is that it would not be on the screen long enough to be seen without having some kind of a DELAY LOOP between the draw and erase. By drawing the shot and then doing other processing that needs to be done before erasing the shot, it stays on the screen long enough to be seen without introducing any DELAY LOOPS that would slow down the action of the game. The OVERHEAD for doing it this way is only four bytes of memory storage which is a small price to pay to keep the game fast.

The FIRE routine as mentioned earlier, checks to see if a character is hit and stops drawing the shot at the point that the character is encountered and CALLS a routine called EXPL2 at address 599FH. This routine passes information to an explosion routine (explained later) and then checks to see what was hit.

At this point in writing the game, the EXPL2 routine was STUBBED OFF (the first instruction was a RET) because the routines for the different characters and explosions were not written. I had no idea what would have to be done when a character was destroyed or how I wanted to handle the explosions. Stubbing off a program section while developing other sections is a pretty common practice. It allows the code to be assembled and tested before all of the features and/or sections are completed.

After the ship motion and fire routines were as complete as possible, I started adding the other characters.

I figured that most of the characters could be handled with the same general procedure, as far as moving on the grid goes, but that the Lander/Man/Mutant/Player's Ship characters would interact with one another in ways that would require some special consideration.

1. - A Lander character always seeks a Man character whenever possible.
2. - A Man character cannot be sought by more than one Lander character at a time.
3. - When a Man character is not available, the Lander will seek the player's ship.
4. - When a Lander is directly over the Man it is seeking, it will move down and pick the Man up.
5. - When the Lander is carrying a Man, the Lander routines are responsible for erasing, drawing, and moving the Man character.
6. - If a Lander is destroyed while carrying a Man, the Man routines must take over moving and drawing the Man.
7. - If the Lander reaches the top of the screen while carrying a Man a new Mutant character is created and the Lander/Man are deleted from their tables.
8. - It is possible for a player's ship to carry a Man character.
9. - When a player's ship is carrying a Man, the ship routines are responsible for erasing, drawing, and moving the Man character.

When a Lander first becomes an active character its primary purpose is to seek out a Man character, travel towards it, drop down and pick it up, and then carry it to the top of the screen at which time the Lander/Man combination turns into a Mutant character.

To avoid the problem of two Landers seeking out the same Man character I had to develop a way to determine if a Man was already being sought by a Lander. To solve this problem I came up with a byte on the entry of the character called a FLAG BYTE.

If you will look at the definition of a Man character table entry at address 6033H, you will notice that the fourth byte (IX+3) of the entry is defined as a FLAG BYTE. For the record, a BYTE is 8 BITS (binary digits) with each BIT being able to have one of two values - 0 or 1. The BITS are numberd 0 to 7.

Each BIT represents a part of the status of the Man character. Bit number 1 of the flag byte for the Man is dedicated to FLAGGING that the Man is currently being sought by a Lander. If it is being sought, this bit will equal 1, if it is not being sought, this bit will equal 0.

Now if you will look at the Lander entry definition at address 5E03H you will see that the Lander also has a FLAG BYTE, the fourth byte, and that BIT number 5 is used as a "search for Man" flag. What that means is, if the BIT is RESET (equals 0), the Lander is not currently seeking a Man. If the bit is SET (equals 1), the Lander is currently seeking a Man. The STATE (set/reset) of the flag byte bits determines which routines will be used to move the Lander across the grid.

At address 5EA3H bit 5 of the flag byte is tested and if it is zero (not currently seeking Man) then a sub-routine at address 5FF7H is CALLED. This sub-routine checks each entry on the Man table to see if there is an ACTIVE Man character that is not already being sought by another Lander.

If a Man is found that is not already being sought then its TAKEN flag (BIT 1 of the FLAG byte) is SET to tell other Landers that this Man is taken, and the grid column # that the Man is on is placed in byte 5 of the Lander entry. This is done so the Lander will know where to move to on the grid. If a free Man is not found then the column # of the player's ship is placed in byte 5 of the Lander entry.

One other thing that has to be done at this time is to put the ENTRY NUMBER of the LANDER into byte number 5 of the MAN CHARACTER ENTRY. The entry number of the Lander is its position on the Lander Table. The reason for doing this is in case the Man character ends up being destroyed the Lander can be told to seek another Man. The only way to do this is to have some kind of POINTER on the Man entry that tells which Lander is seeking the Man.

As I said before, each BIT of the FLAG BYTE determines which of the Lander routines will be used to process the Lander. At address 5EC1H bit 5 of the flag byte is tested to see if a Man is being sought. If it is, bit 3 of the flag byte is tested to see if the Lander is directly over (same column) the Man character. If the Lander is directly over the Man, the LEFT/RIGHT move routine is skipped. If a Man is not being sought or if the Lander is not directly over the Man, the LEFT/RIGHT move routine is CALLED at address 5F20H.

The LEFT/RIGHT move routine for the Lander uses the value stored in the current entry's fifth byte (IX+4) as the column number to move towards. At this point, the number could be either the Man's column number or the player's ship column number. The Lander will move either left or right on the grid depending on which way is the shortest distance to the target column. If the Lander is seeking a Man and the move brings the Lander directly over the Man, bit 3 of the flag byte is SET.

After the left/right move is made then the UP/DOWN move is made. The routine to do this is at address 5F4FH.

This routine first checks bit 4 of the flag byte to see if the Lander is carrying a Man. If it is, the Lander is forced to move up by JUMPING to the routine at address 5F9DH.

If the Lander is not carrying a Man, bit 3 of the flag byte is tested to see if it is directly over the Man it is seeking. If it is, the Lander is forced to move down by JUMPING to the routine at address 5F63H.

If the Lander is not directly over the Man then it moves RANDOMLY up or down.

When a Lander is moving down, a check is made to see if it is moving past row 15 (address 5F67H). If it is, bit 3 of the flag byte is checked to see if it is directly over the Man it is seeking.

If it is over the Man, bit 4 of the Lander flag byte is set (carrying Man flag). This tells the Lander routines that the Lander must move up and that they also have the responsibility of drawing the Man.

Bit 5 of the Man's flag byte is also set (carried by Lander flag). This tells the routines that process the Man characters that this Man is now being carried by a Lander. This has to be done so that the routines won't draw the Man on the terrain while the Lander is carrying it. It now becomes the Lander routine's responsibility to draw the Man.

The entry number of the Man is then put on the Lander's entry in byte 5 in place of the Man's column number that was there (The column # is no longer needed). This has to be done in case the Lander is destroyed while carrying the Man. If the Lander is destroyed while carrying a Man, the responsibility for drawing the Man is returned to the Man routines.

When a Lander is moving up, a check is made to see if it is moving past the top (address 5FA1H). If it is, then bit 4 of the flag byte (carrying Man bit) is tested to see if a Man is being carried. If a Man is being carried, the Lander/Man combination will turn into a Mutant character. The routine that does that (address 5FB0H) deletes the Lander and Man from their tables and creates a new entry on the Mutant table.

As I stated earlier, all of the characters are processed in pretty much the same way. Each character type has its own table with entries on its table for each character. Bytes 0, 1, and 3 are used for the same purpose on every entry for all of the tables. In some of the character ENTRY DEFINITIONS, you will notice that some of the bytes are defined more than once. As the character is processed, the STATE of the bits in the flag byte determines which definition is valid.

The general sequence of events when updating each of the characters is as follows:

1. - Check to see if the character is active. If it is not active do KILLED processing and then go to # 8.
2. - Check to see if the character is WARPING IN. If it is do WARP-IN procedure and then go to # 8.
3. - Check to see if the character is on the screen. If it is not on the screen, process the character movement without any erase or draw and then go to # 8.
4. - Erase the character.
5. - Move the character position on the grid and calculate the new screen position.
6. - Draw the character on the screen at the new position.
7. - Shoot at player's ship.
8. - Get the next entry and loop back to # 1 until the end of the table is reached.

Not all of the characters have a special "KILLED" routine, not all of the characters use the "WARP IN" procedure, and not all of the characters shoot at the player's ship; however, the above is the general structure of the character processing.

I didn't sit down and write all of the character routines at one time, nor did I write all of one character's routines before going on to another character.

In the case of the Lander I wrote the general move routines without it seeking anything and then wrote some of the code for the Man character routines to find out the best way to handle the interaction between the two characters.

After the Lander and Man routines were almost finished the routines for the Mutant were started. This had to be done so that I would know how the Lander routine could create a new Mutant.

In the beginning, not all of the features of the characters were implemented, like shooting at the player's ship, warping in,...etc., because I wasn't sure exactly how to handle them.

Some of the support routines had to be written at the same time that the character routines were written so that the character routines would work properly. Some of these were the TABLE SEARCH routines, character erase routines, ONCHK,...etc. These were written using a general format so that all of the characters could use the routines.

I probably should say something about the two classes of CHARACTER TYPE TABLES at this time. The two classes are POSITION DEPENDENT and POSITION INDEPENDENT.

On the POSITION DEPENDENT tables, (Lander table and Man table) the ENTRY for any given character cannot change its position on the table. The reason the ENTRY cannot be moved on the table is because another character is expecting the entry to be at the same location at all times.

For example, when a Man character is being sought by a Lander, the TAKEN bit in the Man's flag byte is set. If the Lander is shot by the player's ship, the TAKEN flag for the Man has to be reset. This is accommodated by storing the Man's TABLE ENTRY NUMBER in the fifth byte of the Lander's entry and when the Lander is destroyed the value stored there is used to find the Man character's entry on the Man table. If the Man's entry moves on its table between the time that its entry number is placed on the Lander's entry and the time that the Lander is destroyed, things would definitely get confused.

On the POSITION INDEPENDENT tables (all of the rest) the entries can be and are moved on their tables.

The routine that moves the entries is called CRUNCH and is located at address 5B7FH. What it does is move all of the entries on the table back one entry position starting with the entry following the entry that the IX register is pointing to. This, in effect, overwrites the newly destroyed character's entry on the table. This routine is CALLED when an INACTIVE entry (bit 7 of the flag byte is set) is found on a POSITION INDEPENDENT table.

The reason for moving the entries back is when some of the characters are first destroyed some special processing has to be performed. In the case of the POD being destroyed, swarmers have to be created. Having the POD entry overwritten when it is first made inactive avoids the problem of duplicating the swarmers every time the same inactive POD entry is encountered. This could have been accomplished by using FLAGS but, as you probably noticed in the Lander/Man/Mutant explanation, that can get very confusing - to code, follow, and debug.

If you look at the source listing starting at address 73BDH you will see the start of all of the character tables. The first two bytes of each table tells how many ACTIVE ENTRIES are on the table and the MAXIMUM number of entries allowed on the table. On the POSITION DEPENDENT tables the ACTIVE count is first. On the POSITION INDEPENDENT tables the MAXIMUM count is first. This is so that both classes of tables can share the same TABLE SEARCH routines.

There are several SEARCH routines, starting at address 5AC3H and ending at address 5B7EH. They are used to either find a particular entry on a table using the characters screen address, grid column or entry number, or to find an empty spot on a table. The routine at address 5AFCH called COLCVT converts the screen address stored in the HL register pair to a grid column number that is put in the H register.

The search routines are used by the character routines to pass information back and forth between characters or to destroy the character and the empty search routines are used to create new characters.

After the character routines were more or less finished, I completed the EXPL2 routine. That entailed coming up with a way to process the explosions. I wanted the action of the game to be fast so I couldn't see stopping and processing an explosion all the way through every time an alien was hit. I finally elected to process the explosions in PHASES. I also decided to use the Z80 INTERRUPT to process each phase of the explosion rather than CALL the routine within the GAME LOOP.

Using a PHASED explosion and processing it with the INTERRUPTS means that it would be possible to hit a second alien before the explosion from the first was processed completely. There would also be the possibility of a large number of explosions when a SMART BOMB is activated.

To account for this I had to create another TABLE. This table is used more like a FIRST IN - FIRST OUT (FIFO) STACK than a table but I'm going to call it a table anyway. The table itself is located at address 5436H. The first two bytes tell what the MAXIMUM number of entries it will hold and how many explosions are currently waiting to be processed. Each entry on the table consists of three bytes. The first two make up the address on the screen that the explosion is to occur and the third byte tells which direction the explosion should go in.

The routine that creates new entries for the explosion table is called EXPSH and is located at address 53FCH. This routine is CALLED from EXPL2 which is part of the FIRE routine. What it does is find the last active entry on the table and put the new information on the entry following it. It also checks to make sure that the new entry will not OVERFLOW the table.

The routine that starts processing an explosion is called EXPLOD and is located at address 52F5H. It is CALLED from the INTERRUPT HANDLER starting at address 4FB4H. The INTERRUPT HANDLER will be explained later because it processes more things than the explosions.

The explosion routines use self modifying code for much the same reasons that the ship movement routines use it - speed of execution and storage space.

The routine at address 52F5H changes the CALL vector at 4FDFH to call EXPLD at address 532DH and then modifies the explosion routine to reflect the direction that the explosion will go in. The explosion will go UP - RIGHT, UP - LEFT, DOWN - RIGHT, or DOWN LEFT depending on the direction that the shot was fired and which row on the screen that it starts on.

The explosion goes through four phases. First the explosion data is drawn on the screen, then the graphics characters are reversed, then they are reversed once again, and finally, the explosion is erased from the screen.

Right after the erase phase, the CALL vector at address 4FDFH is changed back to the routine at address 52F5H and the entries on the explosion table are moved back one entry position, overwriting the entry just processed.

The EXPL2 routine also calls a routine that updates the players score with the value for the alien hit. This routine is called HIT and is located at address 4FEDH.

This routine takes the value in the BC register pair and adds it to the score of the current player. It also adds it to a counter that keeps track of every time the player's score cycles through 10,000 points. When the counter reaches or exceeds 10,000 points, the player is given an additional ship and smart bomb. Any remainder over 10,000 points is saved as the new starting value for the counter.

One more thing that this routine does is keep track of the individual KILLS that a player makes. For every 25 kills the player is awarded 100 points for every remaining Man character that the player has.

The bonus routine that starts at 5059H does not return to the normal flow of the game. When it is finished displaying the bonus message, it re-initializes the characters and then jumps to a WARM START of the game at address 4AB8H.

One other routine that uses PHASED processing is the WARP-IN routine at address 5CE7H. It is called by the character routines when a new character is brought on the grid and its position coincides with the screen position on the grid. Instead of just having it appear on the screen, I have it materialize in six phases. The fifth byte (IX+4) of a character's entry is used as the phase counter. Not all of the characters are "warped-in", some of them are created by other characters so they don't need to call this routine.

After the explosion and score routines were completed I could start work on the SMART BOMB routine at address 4C37H. What this routine does is scan the screen checking for characters. When it encounters a character it calls the routine at address 5978H called EXPL1, which is part of the hit detect of the FIRE routine for the ship.

Using the grid coordinate system to keep track of characters allows more than one character to occupy the same set of coordinates so the scan is done twice with an update of the characters after the first scan. This only accounts for having the characters two deep on a set of coordinates so it would be possible for some characters to still be on the screen after a smart bomb is invoked but that shouldn't happen too often. More scans could be done but I didn't want to take up too much time processing the smart bomb.

Another routine that I should probably point out is the warp drive routine at address 4D74H. This is also a PHASED routine but it is all done at one time with delay loops between phases. In other words, when the warp drive is activated all other action on the screen freezes. There are 6 sets of graphics data, 1 for each phase, with each set being used twice, once for warping out from the current position and once for warping back in at the new position. The new position is calculated using a random number taken from the REFRESH register.

Up to this point I haven't said anything about the SOUND routines. Some of the sounds are done with DEDICATED routines, routines that are used to produce sound only. The other sounds are produced within loops that are doing other things with the sound portion being incidental to the loop.

An example of a dedicated sound routine would be the one at address 4F2DH. This is the routine that makes the shot sound when the ship fires. It is a sliding pitch sound with the pitch starting high and sliding down the scale. It is relatively short in duration so it doesn't affect the action of the game too much. Sounds of longer duration or of complicated pitch patterns tend to disrupt the action of the game so a trade off usually has to be made between fast action and spectacular sound. I personaly like high speed reflex games so my choice on any trade offs leans towards fast action.

An example of the other type of sound would be the alien explosion sound at address 5350H. All it does is toggle the sound bits in port FFH while the graphics data is being drawn on the screen. This method doesn't disrupt the action of the game but it doesn't give much control over how it's going to sound either.

While I developed all of the characters and support routines, all of the tables were filled with pre-determined values. Doing it this way made it easier to debug each section as I went along. After the character routines were finished, I needed a way to initialize the tables for the game.

The COLD START initialization routines start at address 478AH. They fill the Lander and Man tables with random values for the row and column numbers and clear the rest of the characters tables.

This is done once for each player with the status saved to the player's STATUS tables starting at address 737AH. Each player's status table holds values that tell how many of each type of character are active, the 10,000 point counter, ship count, smart bomb count, kill count,...etc. These values are used to initialize the GAME STATUS table (address 7398H) and the character tables when the game switches back and forth between players.

The warm start initialization at address 4943H uses the values on a player's STATUS table to re-initialize the game variables and tables.

As you can tell from the cold start initialization routines not all of the CHARACTER tables start out with active entries. Some of the characters do not appear until the game has been underway for awhile. To accomplish this I decided to use the Z80 interrupts to process a delay counter for each character.

The INTERRUPT HANDLER starts at address 4FB4H and ends at address 4FECH. It uses a variable called MASK (address 44FFH) to determine what the current interrupt is going process.

If you will notice, bit 7 of the MASK byte is used to skip the interrupt routine. You are probably wondering why I didn't just use the DI (disable interrupts) instruction to stop the interrupts instead of using a FLAG to skip them. That is a hard one to explain.

When I first wrote the interrupt handler, I used the DI instruction and found that on some (not all) MODEL III's, when the interrupts were left off for too long a time the computer would go to sleep (stop working). I'm still not sure if it was a hardware problem or if I was doing something wrong with the software. After spending more hours than I care to think about trying to solve the problem, I decided to leave the interrupts on all of the time and just use a flag to skip them when I needed to disable them.

Problems with the interrupt is also why I read both port E0H and port ECH when clearing the interrupts. The tech. manual says that you only have to read port ECH to clear them but when I did that, strange things would happen. Reading both ports seems to have solved the problem but I don't know the reason why.

If the game is in the ATTRACT mode, bit 6 of the mask byte is set so that the interrupt will jump to a routine that strobes the keyboard to see if someone wants to play the game. When this happens, all of the rest of the interrupt routines are skipped.

If bit 6 is reset, a timer routine at address 5202H is called to bring on a BOMBER, POD, or CRUISER. The timer just increments a variable (TIME) and brings on each character as certain conditions are met.

After processing the timer a second timer routine at address 5282H is called to bring on additional LANDERS. New LANDERS will be created when there are less than 7 left.

If bit 1 of the mask byte is set, the SCORE DISPLAY routine at address 50CBH will be jumped to. The routine that adds a value to a player's score does not display the new score, it sets this bit in the mask byte and lets the interrupt process the display.

If bit 1 of the mask byte is 0 and bit 0 is 1, the alien explosion routine, as explained earlier, will be jumped to.

Using the interrupts to process housekeeping chores can help keep the action of the main portions of the game fast and eliminate the need for complicated delay or timing routines to process things that don't need to be processed each cycle through the game loop.

A disadvantage to using them is that the interrupts are processed a little differently on the MODEL I and aren't available at all on systems that don't have an expansion interface. To get around that I have to have three different versions of a game - MODEL III disk/tape, MODEL I disk, MODEL I tape. The only difference between the MODEL III disk and tape versions is that the disk version has routines to read and write scores on the diskette.

I've mentioned the game loop several times without really defining what it is. It is the loop that calls all of the routines that make up the game. The computer keeps cycling through the loop until certain sets of circumstances are detected.

The loop starts at address 4AD2H and loops back at address 4B01H. The normal order of events is as follows:

1. - Draw the mountains on the screen.
2. - Process Mutant table.
3. - Process Shot table (alien).
4. - Process Lander table.
5. - Process Man table
6. - Process Cruiser table.
7. - Process Bomb table.
8. - Process Bomber table.
9. - Process Pod table.
10. - Process Swarmer table.
11. - Draw player's ship.
12. - Check if warp drive activated.
13. - Check if smart bomb activated.
14. - Scan keyboard for motion routines.
15. - Loop back to # 1.

If the player's ship is destroyed an exit from the loop will be taken to a routine that processes the ship explosion at address 5477H. This routine draws the graphics data for the ship explosion and then jumps to a routine at address 4BB1H that saves the current players status, checks for end of game for current player, checks for any players left, ...etc.

If the game is not over for all of the players, the game is restarted at address 4AA9H with the new players status in the game variables/tables.

If the game is over for all players, control is transferred to the ATTRACT MODE routines starting at address 4732H. The attract mode just gives the computer something to do while nobody is playing the game. It can also ATTRACT interest in the game if it is on display at a store.

The attract mode is usually the last part of the game that I write because I like to use some of the routines of the game itself rather than write everything from sctatch for it. It also lets me know how much memory space I have left for the routines that I have to add.

After initializing, the attract mode is an endless loop with the only possible exit happening during the interrupts. The interrupt handler checks if a key is pressed and jumps to START2 (address 4769H) if one is.

All during the writing of the game, as each section was completed I assembled and tested the program. When things were running smoothly (rarely) and everything worked, I could go on to the next section of the program. When things didn't go so smoothly, I had to try to figure out what went wrong. As anyone who has written a program over one line long knows, this procedure is called DEBUGGING.

Debugging under any circumstances is not easy and real-time game debugging introduces some special problems that can be very difficult to overcome. Even though the computer cycles through the "game loop" over and over the actual path that it will take each time through the loop is highly dependent on what is going on with the keyboard or what path it took the previous time through the loop. When "self modifying code" is used it complicates things even more because it's hard to tell which section was being used when the program "crashed".

One way to debug a program is to use a MONITOR program that lets you SINGLE STEP through the game program. Single Stepping executes the game program one instruction at a time, displaying the values in the registers after each instruction. This would be OK except that the error condition might not show up until the computer has gone through the game loop thousands of times, executing hundreds of instructions each time through the loop. I have heard of some monitors that allow you to set BREAK POINTS that wont go into effect until certain conditions have been met, but I haven't seen one so I usually have to devise a way to find the bugs myself.

When developing a game I generally have a STOP ACTION key. A stop action key is a keyboard key that when pressed stops the game, freezing all of the action on the screen. This way I can see exactly what is going on when things start to go wrong.

Once I have determined which section of the code that the bug is probably in, I add some routines to that section of code that will display the values in the registers and/or the variables associated with the section somewhere on the screen. If the values are not what they should be I know I am getting close. Through trial and error I narrow it down to as small a section of code as possible and then do a printout of the suspect section. Then it's a matter of trying to figure out what's wrong with the logic of that section. More often than not it turns out to be a "typo", or loading the wrong register with a value, or more PUSHes than POPs when managing the registers. Usually it's something simple to fix but hard to find.

No matter how you do it, debugging is a long involved process that can try the patience of anyone. The best thing to do is write the program in little sections, working on and testing only one section at a time.

I hope that you have found this book useful and not too discouraging. Even though game programming can be complicated and time consuming, it's really not as hard to do as you might think.

```
               00010 ;********************************************
               00020 ;**            ALIEN DEFENSE               **
               00030 ;**                                        **
               00040 ;**          COPYRIGHT (C) 1981            **
               00050 ;**     by SOFT SECTOR MARKETING Inc.      **
               00060 ;**                                        **
               00070 ;**        written by Larry Ashmun         **
               00080 ;********************************************
               00090 ;
0001           00100 TAPE    EQU     1               ;1 FOR TAPE (ALIEN)
               00110                                 ;0 FOR DISK (ALIEN3)
               00120 ;
0000           00130 TLK     EQU     0               ;0 FOR NO TALKING
               00140                                 ;1 FOR TALKING
               00150         IFEQ    TAPE,0
               00160         ORG     3D80H
               00170         DM      '          >>>>>>>>> A L I E N   D E F E N S E <<<<<<<<<
      '
               00180         ORG     4FFFH
               00190         ENDIF
               00200 ;
               00210         IFEQ    TAPE,1
44FF           00220         ORG     44FFH
               00230         ENDIF
               00240 ;
44FF 00        00250 MASK    DB      0                       ;INTERRUPT FLAGS
               00260 ;
               00270 ;**************************
               00280 ;**     MOUNTAIN DATA    **
               00290 ;**************************
               00300 ;
               00310 ;        *****   THE FIRST BYTE OF THE MOUNTAIN
               00320 ;        *****   DATA MUST START ON A HEXADECIMAL
               00330 ;        *****   ADDRESS THAT ENDS IN TWO ZEROS
               00340 ;        *****
               00350 ;        *****   4F00H  IS VALID
               00360 ;        *****   4F50H  IS INVALID
               00370 ;
               00380 ;        *****   GRAPHICS DATA   *****
               00390 ;
4500 8C        00400         DB      140,142,156,134,131,131,131,131,131,173
450A 8D        00410         DB      141,172,176,141,172,176,131,140,176,176
4514 B0        00420         DB      176,176,176,176,152,154,131,152,134,165
451E B0        00430         DB      176,176,176,176,176,176,131,131,131,131
4528 83        00440         DB      131,131,131,140,140,140,140,140,176,176
4532 B0        00450         DB      176,176,140,140,140,140,131,131,131,131
453C 83        00460         DB      131,131,131,150,131,131,131,131,131,131
4546 89        00470         DB      137,134,152,140,142,131,152,140,140,140
4550 8C        00480         DB      140,140,140,140,140,140,140,140,172,137
455A A4        00490         DB      164,176,137,142,152,140,140,140,140,140
4564 8C        00500         DB      140,131,131,131,131,131,140,140,140,176
456E B0        00510         DB      176,176,176,176,176,176,140,140,140,140
4578 8C        00520         DB      140,140,142,176,176,176,173,173,176,176
4582 B0        00530         DB      176,176,176,176,176,176,176,176,176,131
458C 83        00540         DB      131,131,131,131,131,131,131,131,152,140
4596 8C        00550         DB      140,140,140,131,131,131,131,131,140,140
45A0 8C        00560         DB      140,176,176,176,176,176,176,176,140,140
45AA 8C        00570         DB      140,140,131,131,152,140,140,140,140,140
```

```
45B4 A4         00580           DB      164,152,164,152,164,176,139,188,131,131
45BE 83         00590           DB      131,131,131,131,131,176,131,131,131,152
45C8 86         00600           DB      134,152,140,140,164,176,176,176,176,176
45D2 89         00610           DB      137,140,140,164,176,154,152,140,134,131
45DC A9         00620           DB      169,165,181,176,186,131,131,131,131,131
45E6 83         00630           DB      131,140,140,140,140,140,140,176,176,176
45F0 B0         00640           DB      176,176,140,140,140,140,140,131,131,131
45FA 9A         00650           DB      154,176,152,140,140,140
                00660                                                ;256 BYTES
                00670 ;
                00680 ;         *****   ROW OFFSET DATA    *****
                00690 ;
4600 40         00700 MOUNT     DB      064,064,000,000,000,000,000,000,000,000
460A 40         00710           DB      064,064,064,128,128,128,192,192,192,192
4614 C0         00720           DB      192,192,192,192,192,192,192,128,128,128
461E 80         00730           DB      128,128,128,128,128,128,192,192,192,192
4628 C0         00740           DB      192,192,192,192,192,192,192,192,192,192
4632 C0         00750           DB      192,192,192,192,192,192,192,192,192,192
463C C0         00760           DB      192,192,192,128,128,128,128,128,128,128
4646 80         00770           DB      128,128,064,064,064,064,000,000,000,000
4650 00         00780           DB      000,000,000,000,000,000,000,000,000,064
465A 40         00790           DB      064,064,128,128,064,064,064,064,064,064
4664 40         00800           DB      064,064,064,064,064,064,064,064,064,064
466E 40         00810           DB      064,064,064,064,064,064,064,064,064,064
4678 40         00820           DB      064,064,064,000,000,000,064,128,128,128
4682 80         00830           DB      128,128,128,128,128,128,128,128,128,192
468C C0         00840           DB      192,192,192,192,192,192,192,192,128,128
4696 80         00850           DB      128,128,128,128,128,128,128,128,128,128
46A0 80         00860           DB      128,128,128,128,128,128,128,128,128,128
46AA 80         00870           DB      128,128,128,128,064,064,064,064,064,064
46B4 40         00880           DB      064,064,064,064,064,064,128,128,192,192
46BE C0         00890           DB      192,192,192,192,192,128,192,192,192,128
46C8 80         00900           DB      128,064,064,064,064,064,064,064,064,064
46D2 80         00910           DB      128,128,128,128,128,128,064,064,064,064
46DC 40         00920           DB      064,128,192,192,192,192,192,192,192,192
46E6 C0         00930           DB      192,192,192,192,192,192,192,192,192,192
46F0 C0         00940           DB      192,192,192,192,192,192,192,192,192,192
46FA 80         00950           DB      128,064,064,064,064,064
                00960                                                ;256 BYTES
                00970 ;
4700 00         00980 DSKFLG    DB      0                   ;DISK WRITE FLAG
4701 00         00990 TLKFLG    DB      0                   ;TALKING FLAG
4702 00         01000 TSPFLG    DB      0
                01010 ;
4703 F3         01020 START     DI
4704 ED56       01030           IM      1
4706 AF         01040           XOR     A
4707 32FF44     01050           LD      (MASK),A
470A 328971     01060           LD      (TENFLG),A          ;STOP HIGH SCORE DISPLAY
470D 320047     01070           LD      (DSKFLG),A          ;NO DISK WRITE
4710 320147     01080           LD      (TLKFLG),A          ;NO TALKING
4713 320247     01090           LD      (TSPFLG),A
4716 D3E0       01100           OUT     (0E0H),A            ;DISABLE INTERRUPTS
4718 3E38       01110           LD      A,38H
471A D3EC       01120           OUT     (0ECH),A            ;SET SCREEN MODE
471C 21B44F     01130           LD      HL,INTERR
```

```
471F 221340    01140          LD      (4013H),HL        ;CHANGE INTERRUPT VECTOR
4722 3EC3      01150          LD      A,0C3H
4724 321240    01160          LD      (4012H),A
4727 3E04      01170          LD      A,4
4729 D3E0      01180          OUT     (0E0H),A          ;SET INTERRUPT TYPE
               01190 ;
               01200          IFEQ    TAPE,0            ;ASSEMBLE IF DISK VERSION
               01210          LD      A,(4023H)
               01220          OR      A                 ;80 TRACK DRIVE ?
               01230          JR      Z,ZAPSKP          ;JUMP IF NO
               01240          XOR     A
               01250          LD      (ZAP),A
               01260 ZAPSKP   CALL    GETSCR            ;READ IN HIGH SCORES
               01270          ENDIF
               01280 ;
               01290          IFEQ    TLK,1             ;ASSEMBLE IF TALKING VER.
               01300          CALL    TLKCHK            ;GET TALK DATA
               01310          ENDIF
               01320 ;
472B 3AFF38    01330 RELEASE  LD      A,(38FFH)         ;WAIT HERE FOR ANY KEY
472E B7        01340          OR      A                 ;PRESSED TO BE RELEASED
472F 20FA      01350          JR      NZ,RELEASE
               01360 ;
               01370 ;        *****   START OF ATTRACT MODE SECTION    *****
               01380 ;
4731 FB        01390          EI
4732 31FA44    01400 CLDSRT   LD      SP,MASK-5
4735 210000    01410          LD      HL,0              ;SETUP STACK
4738 E5        01420          PUSH    HL
4739 E5        01430          PUSH    HL
473A 22FB51    01440          LD      (PL1SCL),HL       ;ZERO SCORES (BINARY)
473D 22FE51    01450          LD      (PL2SCL),HL
4740 228673    01460          LD      (S1PCNT),HL
4743 229573    01470          LD      (S2PCNT),HL
4746 AF        01480          XOR     A
4747 32FD51    01490          LD      (PL1SCH),A
474A 320052    01500          LD      (PL2SCH),A
474D 32A073    01510          LD      (DRWFLG),A
4750 3EC0      01520          LD      A,0C0H            ;SET MASK TO STROBE KEYS
4752 32FF44    01530          LD      (MASK),A          ;FOR START OF GAME.
4755 CD455C    01540          CALL    CLS               ;CLEAR THE SCREEN
               01550 ;
               01560 ;        *****   ATTRACT MODE LOOP
               01570 ;
4758 CD8E71    01580 ILOOP    CALL    TENPNT            ;DISPLAY TOP TEN SCORES
475B CD5E66    01590          CALL    WHO               ;DISPLAY SSM
475E CD5B6B    01600          CALL    WHAT              ;DISPLAY ALIEN DEFENSE
4761 CDC26C    01610          CALL    POINTS            ;HOW TO SCORE
4764 CD6273    01620          CALL    KEYS              ;CONTROL KEYS
4767 18EF      01630          JR      ILOOP             ;LOOP TIL KEY PRESSED
               01640 ;
               01650 ;        *****   START OF GAME     *****
               01660 ;
4769 FB        01670 START2   EI
476A CD455C    01680          CALL    CLS
476D 210000    01690          LD      HL,0              ;CLEAR SCORES
```

```
4770 22FB51   01700         LD      (PL1SCL),HL
4773 AF       01710         XOR     A
4774 32FD51   01720         LD      (PL1SCH),A
4777 320047   01730         LD      (DSKFLG),A      ;ENABLE DISK WRITE
477A 3C       01740         INC     A
477B 328971   01750         LD      (TENFLG),A      ;ENABLE HIGH SCORE DISPLAY
477E 211F4B   01760         LD      HL,STRTMS       ;SET UP TOP 2 ROWS OF SCREEN
4781 CD525D   01770         CALL    PRINT
4784 CDCB50   01780         CALL    SCORE
4787 CD8566   01790         CALL    ASK             ;ASK FOR HOW MANY PLAYERS.
              01800 ;
478A          01810 *GET    INIT/ASM                ;INITIALIZATION ROUTINES
              00010 ;***************************
              00020 ;**     INIT ROUTINE      **
              00030 ;***************************
              00040 ;
478A AF       00050         XOR     A
478B 320152   00060         LD      (PLFLG),A       ;CLEAR WHO'S UP FLAG
478E 329873   00070         LD      (FIRCNT),A      ;CLEAR FIRE COUNTER
4791 32A073   00080         LD      (DRWFLG),A      ;CLEAR MOUNTAIN DRAW FLAG
4794 3E03     00090         LD      A,3
4796 329E73   00100         LD      (SHPCNT),A      ;SHIP COUNTER
4799 329F73   00110         LD      (SMTCNT),A      ;SMART BOMB COUNTER
479C AF       00120         XOR     A
479D 329D73   00130         LD      (WAVCNT),A      ;25 KILL COUNTER FOR BONUS
47A0 210000   00140         LD      HL,0
47A3 229B73   00150         LD      (CMEAGN),HL     ;10,000 COUNTER
47A6 210020   00160         LD      HL,2000H
47A9 229973   00170         LD      (DLYCNT),HL     ;DELAY COUNTER FOR ALIENS
              00180 ;
47AC CDDE47   00190         CALL    MUTIT           ;INIT. MUTANT/CRUISER/BOMBER TABLES
47AF 060A     00200         LD      B,10            ;START WITH 10 LANDERS
47B1 CDBC49   00210         CALL    LNDIT           ;FILL LANDER TABLE
47B4 060A     00220         LD      B,10
47B6 CDEE49   00230         CALL    MANIT           ;FILL MAN TABLE
              00240 ;
47B9 CD0349   00250         CALL    TABINT          ;TRANSFER TO PLAYER 1 VARIABLES
47BC 3AC14C   00260         LD      A,(PLYCNT)
47BF FE02     00270         CP      2               ;TWO PLAYERS ?
47C1 3009     00280         JR      NC,PLYTWO       ;JUMP IF YES
47C3 210000   00290         LD      HL,0
47C6 229573   00300         LD      (S2PCNT),HL     ;CANCEL SECOND PLAYER
47C9 C3B14B   00310         JP      WRMSRT
              00320 ;
47CC 060A     00330 PLYTWO  LD      B,10
47CE CDBC49   00340         CALL    LNDIT           ;FILL LANDER TABLE
47D1 060A     00350         LD      B,10
47D3 CDEE49   00360         CALL    MANIT           ;FILL MAN TABLE
47D6 3E01     00370         LD      A,1             ;FLAG PLAYER 2
47D8 320152   00380         LD      (PLFLG),A
47DB C3B14B   00390         JP      WRMSRT
              00400 ;
              00410 ;       *** CLEAR MUTANT TABLE ***
              00420 ;
47DE DD216574 00430 MUTIT   LD      IX,MUTBL+1      ;MUTANT TABLE
47E2 3E80     00440         LD      A,80H           ;VALUE FOR FLAG BYTE
```

```
47E4 110600   00450        LD     DE,6          ;ENTRY OFFSET
47E7 060F     00460        LD     B,15          ;# OF ENTRIES
              00470 ;
47E9 DD36FF00 00480        LD     (IX-1),0      ;ACTIVE ENTRIE COUNT
47ED DD70FE   00490        LD     (IX-2),B      ;MAX ENTRIES ALLOWED
47F0 DD7703   00500 MUILP  LD     (IX+3),A      ;SET INACTIVE FLAG
47F3 DD360203 00510        LD     (IX+2),3      ;DELAY
47F7 DD19     00520        ADD    IX,DE         ;NEXT ENTRY
47F9 10F5     00530        DJNZ   MUILP
              00540 ;
              00550 ;      *** CLEAR CRUISER TABLE ***
              00560 ;
47FB DD21C774 00570        LD     IX,CRUTBL+1   ;CRUISER TABLE
47FF 0604     00580        LD     B,4           ;# OF ENTRIES
4801 DD36FF00 00590        LD     (IX-1),0      ;ACTIVE ENTRY COUNT
4805 DD70FE   00600        LD     (IX-2),B      ;MAX ENTRIES ALLOWED
4808 DD7703   00610 CRUILP LD     (IX+3),A      ;FLAG INACTIVE
480B DD360214 00620        LD     (IX+2),20     ;MOVE DELAY
480F DD36051E 00630        LD     (IX+5),30     ;SHOT DELAY
4813 DD19     00640        ADD    IX,DE         ;NEXT ENTRY
4815 10F1     00650        DJNZ   CRUILP
              00660 ;
              00670 ;      *** CLEAR BOMBER TABLE ***
              00680 ;
4817 DD21FF74 00690        LD     IX,BMRTBL+1   ;BOMBER TABLE
481B 0604     00700        LD     B,4           ;# OF ENTRIES
481D DD36FF00 00710        LD     (IX-1),0      ;ACTIVE ENTRY COUNT
4821 DD70FE   00720        LD     (IX-2),B      ;MAX ENTRIES ALLOWED
4824 DD7703   00730 BMRLP  LD     (IX+3),A      ;FLAG INACTIVE
4827 DD36020F 00740        LD     (IX+2),15     ;MOVE DELAY
482B DD19     00750        ADD    IX,DE         ;NEXT ENTRY
482D 10F5     00760        DJNZ   BMRLP
482F C9       00770        RET
              00780 ;
              00790 ;      *** CLEAR SHOT TABLE ***
              00800 ;
4830 CD535C   00810 INIT2  CALL   GMECLR        ;PARTIAL SCREEN CLEAR
4833 DD219E75 00820        LD     IX,SHOTBL+1   ;ALIEN SHOT TABLE
4837 110600   00830        LD     DE,6          ;ENTRY OFFSET
483A 3E80     00840        LD     A,80H         ;VALUE FOR FLAG BYTE
483C 0606     00850        LD     B,6           ;# OF ENTRIES
483E DD36FF00 00860        LD     (IX-1),0      ;ACTIVE ENTRY COUNT
4842 DD70FE   00870        LD     (IX-2),B      ;MAX ENTRIES ALLOWED
4845 DD7703   00880 SHOTLP LD     (IX+3),A      ;FLAG INACTIVE
4848 DD360208 00890        LD     (IX+2),8      ;DELAY
484C DD19     00900        ADD    IX,DE         ;NEXT ENTRY
484E 10F5     00910        DJNZ   SHOTLP
              00920 ;
              00930 ;      *** CLEAR BOMB TABLE ***
              00940 ;
4850 DD211F75 00950        LD     IX,BMBTBL+1   ;BOMB TABLE
4854 0614     00960        LD     B,20          ;ENTRY COUNT
4856 DD36FF00 00970        LD     (IX-1),0      ;ACTIVE ENTRY COUNT
485A DD70FE   00980        LD     (IX-2),B      ;MAX ENTRIES ALLOWED
485D DD7703   00990 BMBLP  LD     (IX+3),A      ;FLAG INACTIVE
4860 DD360214 01000        LD     (IX+2),20     ;DELAY
```

```
4864 DD19     01010          ADD     IX,DE            ;NEXT ENTRY
4866 10F5     01020          DJNZ    BMBLP
              01030 ;
              01040 ;        *** CLEAR POD TABLE ***
              01050 ;
4868 DD21CA75 01060          LD      IX,PODTBL+1      ;POD TABLE
486C 0602     01070          LD      B,2              ;ENTRY COUNT
486E DD36FF00 01080          LD      (IX-1),0         ;ACTIVE ENTRY COUNT
4872 DD70FE   01090          LD      (IX-2),B         ;MAX ENTRIES ALLOWED
4875 DD7703   01100 PODLP    LD      (IX+3),A         ;FLAG INACTIVE
4878 DD360203 01110          LD      (IX+2),3         ;DELAY
487C DD19     01120          ADD     IX,DE            ;NEXT ENTRY
487E 10F5     01130          DJNZ    PODLP
              01140 ;
              01150 ;        *** CLEAR SWARMER TABLE ***
              01160 ;
4880 DD21DE75 01170          LD      IX,SRMTBL+1      ;SWARMER TABLE
4884 060C     01180          LD      B,12             ;# OF ENTRIES
4886 DD36FF00 01190          LD      (IX-1),0         ;ACTIVE ENTRY COUNT
488A DD70FE   01200          LD      (IX-2),B         ;MAX ENTRIES ALLOWED
488D DD7703   01210 SRMLP    LD      (IX+3),A         ;FLAG INACTIVE
4890 DD360203 01220          LD      (IX+2),3         ;DELAY
4894 DD19     01230          ADD     IX,DE            ;NEXT ENTRY
4896 10F5     01240          DJNZ    SRMLP
              01250 ;
              01260 ;        *** INITIALIZE SHIP ***
              01270 ;
4898 215F3D   01280          LD      HL,3D5FH         ;SHIP STARTING SCREEN POS.
489B 22A773   01290          LD      (SHPOS),HL
489E 3E05     01300          LD      A,5
48A0 32A173   01310          LD      (SHROW),A        ;SHIP ROW ON PLAY GRID
48A3 3E1F     01320          LD      A,31
48A5 32A273   01330          LD      (SHCOL),A        ;SHIP COLUMN ON PLAY GRID
              01340 ;
              01350 ;        *** INITIALIZE MOUNTAINS ***
              01360 ;
48A8 AF       01370          XOR     A
48A9 32BC73   01380          LD      (WINDOW),A       ;MOUNTAIN COLUMN ON SCREEN
48AC 210046   01390          LD      HL,MOUNT         ;START OF MOUNTAIN DATA
48AF 22BA73   01400          LD      (DPOINT),HL
              01410 ;
              01420 ;        *** INITIALIZE VECTORS ***
              01430 ;
48B2 218757   01440 VECTOR   LD      HL,ERASE         ;SHIP ERASE ROUTINE ADDRESS
48B5 226B58   01450          LD      (UPOT1+1),HL
48B8 210358   01460          LD      HL,UPDT1         ;SHIP UPDATE ROUTINE ADDRESS
48BB 22A257   01470          LD      (UPDATE+1),HL
48BE 219D56   01480          LD      HL,SPEEDU        ;SHIP ACCELERATION ROUTINE ADDRESS
48C1 226A56   01490          LD      (MVCNG+1),HL
48C4 212157   01500          LD      HL,STOP          ;SHIP DECELERATION ROUTINE ADDRESS
48C7 227456   01510          LD      (MVCNG2+1),HL
48CA 217258   01520          LD      HL,DOWN          ;SHIP MOVE DOWN ROUTINE ADDRESS
48CD 223E56   01530          LD      (DWNCNG+1),HL
48D0 21F552   01540          LD      HL,EXPLOD        ;ALIEN EXPLODE ROUTINE ADDRESS
48D3 22E04F   01550          LD      (EXCNG0+1),HL
              01560 ;
```

```
48D6 AF       01570        XOR   A
48D7 32FF44   01580        LD    (MASK),A         ;CLEAR INTERRUPT MASK
48DA 323754   01590        LD    (EXTBL),A        ;EXPLOSION COUNTER
48DD 32B673   01600        LD    (MVFLG3),A       ;SHIP MOTION FLAG
48E0 32A973   01610        LD    (DIRFLG),A       ; ''    ''   DIRECTION
48E3 32AA73   01620        LD    (DIRF2),A        ;SHIP DIRECTION FLAG
48E6 3E0A     01630        LD    A,10
48E8 32B873   01640        LD    (SPDCNT),A       ;SPEED COUNTERS
48EB 32B973   01650        LD    (SPDCNT+1),A
48EE 3E04     01660        LD    A,4
48F0 32F355   01670        LD    (EXCNT),A        ;ALIEN EXPLODE PHASE COUNT
48F3 D3E0     01680        OUT   (0E0H),A         ;SET INTERRUPT TYPE
48F5 CDF958   01690        CALL  DRAW             ;DRAW MOUNTAINS
48F8 CDCB50   01700        CALL  SCORE            ;DISPLAY SCORE
48FB ED5BA773 01710        LD    DE,(SHPOS)       ;GET SHIP POSITION
48FF CDA157   01720        CALL  UPDATE           ;DRAW SHIP
4902 C9       01730        RET
              01740 ;
4903 3A0152   01750 TABINT LD    A,(PLFLG)        ;MOVE DATA FROM MAIN
4906 B7       01760        OR    A                ;TABLES TO PLAYERS TABLES
4907 2009     01770        JR    NZ,TWO           ;JUMP IF PLAYER 2
              01780 ;
4909 FD217A73 01790        LD    IY,TABLE1        ;PLAYER 1 VARIABLES
490D 118073   01800        LD    DE,FI1CNT
4910 1807     01810        JR    MOGO
              01820 ;
4912 FD218973 01830 TWO    LD    IY,TABLE2        ;PLAYER 2 VARIABLES
4916 118F73   01840        LD    DE,FI2CNT
              01850 ;
4919 010900   01860 MOGO   LD    BC,9             ;VARIABLE COUNT
491C 219873   01870        LD    HL,FIRCNT        ;GAME VARIABLES
491F EDB0     01880        LDIR
              01890 ;
4921 3ABD73   01900        LD    A,(LNDTBL-1)     ;ACTIVE LANDERS
4924 FD7700   01910        LD    (IY),A
4927 3A0174   01920        LD    A,(MANTBL-1)     ;ACTIVE MEN
492A FD7701   01930        LD    (IY+1),A
492D 3A6474   01940        LD    A,(MUTBL)        ;ACTIVE MUTANTS
4930 FD7702   01950        LD    (IY+2),A
4933 2AC574   01960        LD    HL,(CRUTBL-1)    ;L=MAX : H=ACTIVE
4936 FD7503   01970        LD    (IY+3),L
4939 FD7404   01980        LD    (IY+4),H
493C 3AFE74   01990        LD    A,(BMRTBL)       ;ACTIVE BOMBERS
493F FD7705   02000        LD    (IY+5),A
4942 C9       02010        RET
              02020 ;
4943 3A0152   02030 INIT3  LD    A,(PLFLG)        ;MOVE DATA FROM PLAYER
4946 B7       02040        OR    A                ;TABLES TO GAME TABLES
4947 2009     02050        JR    NZ,TWO2          ;JUMP IF PLAYER 2
              02060 ;
4949 FD217A73 02070        LD    IY,TABLE1        ;PLAYER 1 VARIABLES
494D 218073   02080        LD    HL,FI1CNT
4950 1807     02090        JR    MOGO2
              02100 ;
4952 FD218973 02110 TWO2   LD    IY,TABLE2        ;PLAYER 2 VARIABLES
4956 218F73   02120        LD    HL,FI2CNT
```

```
               02130 ;
4959 010900    02140 MOGO2    LD      BC,9              ;VARIABLE COUNT
495C 119873    02150          LD      DE,FIRCNT
495F EDB0      02160          LDIR
               02170 ;
4961 DD21BF73  02180          LD      IX,LNDTBL+1       ;CLEAR LANDER TABLE
4965 110600    02190          LD      DE,6              ;ENTRY OFFSET
4968 060A      02200          LD      B,10              ;ENTRY COUNT
496A 3E80      02210          LD      A,80H             ;FLAG INACTIVE
496C DD7703    02220 LXLP     LD      (IX+3),A          ;FLAG INACTIVE
496F DD19      02230          ADD     IX,DE             ;NEXT ENTRY
4971 10F9      02240          DJNZ    LXLP
               02250 ;
4973 DD210374  02260          LD      IX,MANTBL+1       ;CLEAR MAN TABLE
4977 060F      02270          LD      B,15              ;ENTRY COUNT
4979 DD7703    02280 MXLP     LD      (IX+3),A          ;FLAG INACTIVE
497C DD19      02290          ADD     IX,DE             ;NEXT ENTRY
497E 10F9      02300          DJNZ    MXLP
               02310 ;
4980 FD4600    02320          LD      B,(IY)            ;ACTIVE LANDERS FOR CURRENT PLAYER
4983 3AA073    02330          LD      A,(DRWFLG)        ;MOUNTAINS ACTIVE ?
4986 B7        02340          OR      A
4987 F5        02350          PUSH    AF                ;SAVE STATUS FLAGS
4988 2802      02360          JR      Z,LNIT9           ;JUMP IF YES
498A 060A      02370          LD      B,10              ;MAXIMUM ENTRY COUNT
               02380 ;
498C CDBC49    02390 LNIT9    CALL    LNDIT             ;FILL LANDER TABLE
498F FD4601    02400          LD      B,(IY+1)          ;ACTIVE MEN
4992 CDEE49    02410          CALL    MANIT             ;FILL MAN TABLE
               02420 ;
4995 FD4602    02430          LD      B,(IY+2)          ;ACTIVE MUTANTS
4998 F1        02440          POP     AF                ;GET STATUS FLAGS
4999 F5        02450          PUSH    AF                ;SAVE  ''    ''
499A 2802      02460          JR      Z,MUTIT9          ;JUMP IF MOUNTAINS ACTIVE
               02470 ;
499C 060A      02480          LD      B,10              ;MAXIMUM MUTANTS
499E CD204A    02490 MUTIT9   CALL    MUTIT2            ;FILL MUTANT TABLE
49A1 FD4E03    02500          LD      C,(IY+3)          ;MAX CRUISERS
49A4 FD4604    02510          LD      B,(IY+4)          ;ACTIVE CRUISERS
49A7 F1        02520          POP     AF                ;GET STATUS FLAGS
49A8 F5        02530          PUSH    AF                ;SAVE  ''    ''
49A9 2802      02540          JR      Z,CRUIT9          ;JUMP IF MOUNTAINS ACTIVE
               02550 ;
49AB 0604      02560          LD      B,4               ;MAXIMUM CRUISER COUNT
49AD CD4A4A    02570 CRUIT9   CALL    CRUIT2            ;FILL CRUISER TABLE
49B0 FD4605    02580          LD      B,(IY+5)          ;ACTIVE BOMBERS
49B3 F1        02590          POP     AF                ;GET STATUS FLAGS
49B4 2802      02600          JR      Z,BMRIT9          ;JUMP IF MOUNTAINS ACTIVE
49B6 0604      02610          LD      B,4               ;MAXIMUM BOMBER COUNT
49B8 CD7B4A    02620 BMRIT9   CALL    BMRIT2            ;FILL BOMBER TABLE
49BB C9        02630          RET
               02640 ;
               02650 ;        *** FILL LANDER TABLE ***
               02660 ;
49BC DD21BF73  02670 LNDIT    LD      IX,LNDTBL+1       ;INIT LANDER TABLE
49C0 DD36FF0A  02680          LD      (IX-1),10         ;MAX ENTRIES
```

```
49C4 DD70FE   02690           LD      (IX-2),B        ;ACTIVE ENTRIES
49C7 78       02700           LD      A,B
49C8 B7       02710           OR      A
49C9 C8       02720           RET     Z               ;RETURN IF NONE ACTIVE
49CA CD705C   02730           CALL    RNDCOL          ;GET RANDOM SELECTION
49CD 110600   02740           LD      DE,6            ;ENTRY OFFSET
49D0 7E       02750 LLP3      LD      A,(HL)          ;GET RANDOM COLUMN #
49D1 DD7701   02760           LD      (IX+1),A        ;PUT ON TABLE
49D4 DD360002 02770           LD      (IX),2          ;ROW # 2
49D8 DD36020F 02780           LD      (IX+2),15       ;DELAY
49DC DD360340 02790           LD      (IX+3),40H      ;SET WARP-IN
49E0 DD360407 02800           LD      (IX+4),7        ;PHASE COUNT FOR WARP-IN
49E4 DD360564 02810           LD      (IX+5),100      ;SHOT DELAY
49E8 DD19     02820           ADD     IX,DE           ;NEXT ENTRY
49EA 23       02830           INC     HL              ;NEXT RANDOM #
49EB 10E3     02840           DJNZ    LLP3
49ED C9       02850           RET
              02860 ;
              02870 ;         *** FILL MAN TABLE ***
              02880 ;
49EE DD210374 02890 MANIT     LD      IX,MANTBL+1
49F2 DD36FF0F 02900           LD      (IX-1),15       ;MAX ENTRIES
49F6 DD70FE   02910           LD      (IX-2),B        ;ACTIVE ENTRIES
49F9 78       02920           LD      A,B
49FA B7       02930           OR      A
49FB C8       02940           RET     Z               ;RETURN IF NONE ACTIVE
49FC CD705C   02950           CALL    RNDCOL
49FF 110600   02960           LD      DE,6            ;ENTRY OFFSET
4A02 7E       02970 MALP1     LD      A,(HL)          ;GET RANDOM COLUMN #
4A03 DD7701   02980           LD      (IX+1),A        ;PUT ON TABLE
4A06 DD36000F 02990           LD      (IX),15         ;ROW # 15
4A0A DD36020F 03000           LD      (IX+2),15       ;DELAY
4A0E DD360300 03010           LD      (IX+3),0        ;FLAGS
4A12 DD360400 03020           LD      (IX+4),0        ;LINK TO LANDER/SHIP
4A16 DD360500 03030           LD      (IX+5),0        ;EXTRA
4A1A DD19     03040           ADD     IX,DE           ;NEXT ENTRY
4A1C 23       03050           INC     HL              ;NEXT RANDOM #
4A1D 10E3     03060           DJNZ    MALP1
4A1F C9       03070           RET
              03080 ;
              03090 ;         *** FILL MUTANT TABLE ***
              03100 ;
4A20 DD216574 03110 MUTIT2    LD      IX,MUTBL+1
4A24 DD70FF   03120           LD      (IX-1),B        ;ACTIVE MUTANT COUNT
4A27 DD36FE0F 03130           LD      (IX-2),15       ;MAX ENTRIES ALLOWED
4A2B 78       03140           LD      A,B
4A2C B7       03150           OR      A
4A2D C8       03160           RET     Z               ;RETURN IF NONE ACTIVE
4A2E CD705C   03170           CALL    RNDCOL
4A31 110600   03180           LD      DE,6            ;ENTRY OFFSET
4A34 7E       03190 MUILP2    LD      A,(HL)          ;GET RANDOM COLUMN #
4A35 DD7701   03200           LD      (IX+1),A        ;PUT ON TABLE
4A38 DD360002 03210           LD      (IX),2          ;ROW # 2
4A3C DD360203 03220           LD      (IX+2),3        ;DELAY
4A40 DD360300 03230           LD      (IX+3),0        ;FLAGS
4A44 DD19     03240           ADD     IX,DE           ;NEXT ENTRY
```

```
4A46 23       03250          INC     HL               ;NEXT RANDOM #
4A47 10EB     03260          DJNZ    MUILP2
4A49 C9       03270          RET
              03280 ;
              03290 ;        *** FILL CRUISER TABLE ***
              03300 ;
4A4A DD21C774 03310 CRUIT2   LD      IX,CRUTBL+1
4A4E DD70FF   03320          LD      (IX-1),B         ;ACTIVE CRUISER COUNT
4A51 DD71FE   03330          LD      (IX-2),C         ;MAX CRUISERS ALLOWED
4A54 78       03340          LD      A,B
4A55 B7       03350          OR      A
4A56 C8       03360          RET     Z                ;RETURN IF NONE ACTIVE
4A57 CD705C   03370          CALL    RNDCOL
4A5A 110600   03380          LD      DE,6             ;ENTRY OFFSET
4A5D 7E       03390 CRULP2   LD      A,(HL)           ;GET RANDOM COLUMN #
4A5E DD7701   03400          LD      (IX+1),A         ;PUT ON TABLE
4A61 DD360002 03410          LD      (IX),2           ;ROW # 2
4A65 DD360214 03420          LD      (IX+2),20        ;TURN DELAY
4A69 DD360340 03430          LD      (IX+3),40H       ;WARP-IN
4A6D DD360407 03440          LD      (IX+4),7         ;WARP-IN PHASE COUNT
4A71 DD36051E 03450          LD      (IX+5),30        ;SHOT DELAY
4A75 DD19     03460          ADD     IX,DE            ;NEXT ENTRY
4A77 23       03470          INC     HL               ;NEXT RANDOM #
4A78 10E3     03480          DJNZ    CRULP2
4A7A C9       03490          RET
              03500 ;
              03510 ;        *** FILL BOMBER TABLE ***
              03520 ;
4A7B DD21FF74 03530 BMRIT2   LD      IX,BMRTBL+1
4A7F DD70FF   03540          LD      (IX-1),B         ;ACTIVE COUNT
4A82 DD36FE04 03550          LD      (IX-2),4         ;MAX ALLOWED COUNT
4A86 78       03560          LD      A,B
4A87 B7       03570          OR      A
4A88 C8       03580          RET     Z                ;RETURN IF NONE ACTIVE
4A89 CD705C   03590          CALL    RNDCOL
4A8C 110600   03600          LD      DE,6             ;ENTRY OFFSET
4A8F 7E       03610 BMRLP2   LD      A,(HL)           ;GET RANDOM COLUMN #
4A90 DD7701   03620          LD      (IX+1),A         ;PUT ON TABLE
4A93 DD360002 03630          LD      (IX),2           ;ROW # 2
4A97 DD36020F 03640          LD      (IX+2),15        ;DELAY
4A9B DD360350 03650          LD      (IX+3),50H       ;WARP-INIT
4A9F DD360407 03660          LD      (IX+4),7         ;WARP-IN PHASE COUNT
4AA3 DD19     03670          ADD     IX,DE            ;NEXT ENTRY
4AA5 23       03680          INC     HL               ;NEXT RANDOM #
4AA6 10E7     03690          DJNZ    BMRLP2
4AA8 C9       03700          RET
              03710 ;
4AA9          01820 *GET     GAME/ASM                 ;GAME CONTROL ROUTINES
              00010 ;*******************
              00020 ;**      GAME      **
              00030 ;*******************
              00040 ;
4AA9 3AFF38   00050 GAMES    LD      A,(38FFH)        ;WAIT FOR A KEY TO BE
4AAC 47       00060          LD      B,A              ;PRESSED
4AAD DB00     00070          IN      A,(0)            ;OR JOYSTICK TO BE MOVED
4AAF 2F       00080          CPL
```

```
4AB0 B0       00090        OR     B
4AB1 28F6     00100        JR     Z,GAMES
              00110 ;
4AB3 CD414F   00120        CALL   MOOSE           ;SPECIAL SOUND ROUTINE
4AB6 180A     00130        JR     GAME2
              00140 ;
4AB8 3AFF38   00150 GAME   LD     A,(38FFH)       ;START AFTER BONUS
4ABB 47       00160        LD     B,A             ;WAIT FOR KEY OR JOYSTICK
4ABC DB00     00170        IN     A,(0)
4ABE 2F       00180        CPL
4ABF B0       00190        OR     B
4AC0 28F6     00200        JR     Z,GAME
              00210 ;
4AC2 31FA44   00220 GAME2  LD     SP,MASK-5       ;RESET STACK FOR SAFETY
4AC5 210000   00230        LD     HL,0
4AC8 E5       00240        PUSH   HL
4AC9 E5       00250        PUSH   HL
4ACA 3AFF44   00260        LD     A,(MASK)
4ACD CBFF     00270        SET    7,A             ;SET INTERRUPT ENABLE
4ACF 32FF44   00280        LD     (MASK),A
              00290 ;
4AD2 CDF958   00300 BIGLP  CALL   DRAW            ;DRAW MOUNTAINS
4AD5 CD034B   00310        CALL   CHARAC          ;PROCESS ALIENS
4AD8 ED5BA773 00320        LD     DE,(SHPOS)      ;GET SHIP POSITION
4ADC CDA157   00330        CALL   UPDATE          ;DRAW SHIP
              00340 ;
4ADF 3A4038   00350        LD     A,(3840H)
4AE2 CB47     00360        BIT    0,A             ;WARP OUT ?
4AE4 F5       00370        PUSH   AF
4AE5 C4744D   00380        CALL   NZ,HYPER        ;DO HYPER SPACE IF YES
4AE8 F1       00390        POP    AF
4AE9 2006     00400        JR     NZ,HPSKP        ;JUMP IF NO
4AEB F5       00410        PUSH   AF
4AEC AF       00420        XOR    A
4AED 32C04C   00430        LD     (HYPFLG),A      ;CLEAR HYPER SPACE FLAG
              00440 ;
4AF0 F1       00450        POP    AF
4AF1 CB7F     00460 HPSKP  BIT    7,A             ;SMART BOMB ?
4AF3 F5       00470        PUSH   AF
4AF4 C4374C   00480        CALL   NZ,SMART        ;CALL IF YES
4AF7 F1       00490        POP    AF
4AF8 2004     00500        JR     NZ,POLSKP       ;JUMP IF NO
4AFA AF       00510        XOR    A
4AFB 32BF4C   00520        LD     (SMTFLG),A      ;CLEAR SMART BOMB FLAG
              00530 ;
4AFE CDF455   00540 POLSKP CALL   POLL            ;STROBE KEYBOARD
4B01 18CF     00550        JR     BIGLP           ;KEEP ON TRUCKIN'
              00560 ;
4B03 CDC960   00570 CHARAC CALL   MUTANT          ;PROCESS MUTANTS
4B06 CD5D5D   00580        CALL   SHOT            ;PROCESS SHOTS
4B09 CD6F5E   00590        CALL   LANDER          ;PROCESS LANDERS
4B0C CD3360   00600        CALL   MAN             ;PROCESS MEN
4B0F CD8361   00610        CALL   CRUISE          ;PROCESS CRUISERS
4B12 CD8B63   00620        CALL   BOMB            ;PROCESS BOMBS
4B15 CD8A62   00630        CALL   BOMBER          ;PROCESS BOMBERS
4B18 CDCD63   00640        CALL   POD             ;PROCESS PODS
```

```
4B1B CD4965    00650          CALL     SWARM           ;PROCESS SWARMERS
4B1E C9        00660          RET
               00670 ;
4B1F 20        00680 STRTMS   DM       '                    A L I E N   D E F E N S E
   '
4B5F 20        00690 SCMSG    DM       ' PLAYER 1 '
4B69 30        00700 SCORE1   DM       '00000000'
4B71 20        00710          DM       '                           '
4B8D 30        00720 SCORE2   DM       '00000000 PLAYER 2 '
4B9F 00        00730          DB       0
4BA0 20        00740 FLCLR    DM       '                   '
               00750 ;
4BB1 CD0349    00760 WRMSRT   CALL     TABINT          ;SAVE PLAYER STATUS
4BB4 3A9E73    00770          LD       A,(SHPCNT)      ;ANY SHIPS LEFT ?
4BB7 B7        00780          OR       A
4BB8 2015      00790          JR       NZ,CANSKP       ;JUMP IF YES
               00800 ;
4BBA CDC24C    00810          CALL     OVER            ;TELL 'EM IT'S OVER
4BBD 3AC14C    00820          LD       A,(PLYCNT)      ;CANCEL PLAYER
4BC0 3D        00830          DEC      A
4BC1 32C14C    00840          LD       (PLYCNT),A
4BC4 CA3247    00850          JP       Z,CLDSRT        ;JUMP IF NO PLAYERS LEFT
               00860 ;
4BC7 3A0152    00870          LD       A,(PLFLG)       ;CHANGE PLAYERS
4BCA EE01      00880          XOR      1
4BCC 320152    00890          LD       (PLFLG),A
               00900 ;
4BCF 3AC14C    00910 CANSKP   LD       A,(PLYCNT)      ;GET PLAYER COUNT
4BD2 3D        00920          DEC      A
4BD3 2808      00930          JR       Z,SWTSKP        ;JUMP IF ONLY ONE
               00940 ;
4BD5 3A0152    00950          LD       A,(PLFLG)       ;SWAP PLAYERS
4BD8 EE01      00960          XOR      1
4BDA 320152    00970          LD       (PLFLG),A
               00980 ;
4BDD CD4349    00990 SWTSKP   CALL     INIT3           ;PUT PLAYER VARIABLES INTO
               01000                                   ;GAME VARIABLES.
4BE0 3A0152    01010          LD       A,(PLFLG)
4BE3 B7        01020          OR       A               ;WHO'S UP ?
4BE4 2009      01030          JR       NZ,TWO3         ;JUMP IF PLAYER 2
               01040 ;
4BE6 3A8673    01050          LD       A,(S1PCNT)      ;PLAYER 1 SHIP COUNT
4BE9 3D        01060          DEC      A
4BEA 328673    01070          LD       (S1PCNT),A
4BED 1807      01080          JR       RESKP2
               01090 ;
4BEF 3A9573    01100 TWO3     LD       A,(S2PCNT)      ;PLAYER 2 SHIP COUNT
4BF2 3D        01110          DEC      A
4BF3 329573    01120          LD       (S2PCNT),A
               01130 ;
4BF6 329E73    01140 RESKP2   LD       (SHPCNT),A      ;GAME VARIABLE SHIP COUNT
4BF9 CD3048    01150          CALL     INIT2           ;INIT COMMON VARIABLES
4BFC CD054C    01160          CALL     FLASH           ;FLASH WHO'S UP
4BFF CDCB50    01170          CALL     SCORE           ;DISPLAY SCORE
4C02 C3A94A    01180          JP       GAMES           ;START THE GAME
               01190 ;
4C05 3A0152    01200 FLASH    LD       A,(PLFLG)       ;GET WHO'S UP
```

```
                01210 ;
                01220         IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
                01230         PUSH    AF
                01240         LD      HL,YUP1D        ;TALK DATA PLAYER 1 UP
                01250         OR      A
                01260         JR      Z,YUP2X
                01270 ;
                01280         LD      HL,YUP2D        ;TALK DATA PLAYER 2 UP
                01290 YUP2X   CALL    TALK            ;MAKE NOISE
                01300         POP     AF
                01310         ENDIF
                01320 ;
4C08 21604B     01330         LD      HL,SCMSG+1      ;PLAYER 1 SCORE
4C0B 11413C     01340         LD      DE,3C41H        ;SCREEN POS.
4C0E B7         01350         OR      A               ;WHO'S UP ?
4C0F 2806       01360         JR      Z,FLASH2        ;JUMP IF PLAYER 1
                01370 ;
4C11 218D4B     01380         LD      HL,SCORE2       ;PLAYER 2 SCORE
4C14 116E3C     01390         LD      DE,3C6EH        ;SCREEN POS.
4C17 060A       01400 FLASH2  LD      B,10            ;FLASH COUNT
4C19 C5         01410 FLSHLP  PUSH    BC              ;SAVE IT
4C1A E5         01420         PUSH    HL              ;SAVE SCORE POINTER
4C1B D5         01430         PUSH    DE              ;SAVE SCREEN ADDRESS
4C1C 011100     01440         LD      BC,17           ;CHARACTER COUNT
4C1F EDB0       01450         LDIR
4C21 CD146C     01460         CALL    DELAY6          ;PAUSE
                01470 ;
4C24 D1         01480         POP     DE              ;GET SCREEN POS.
4C25 D5         01490         PUSH    DE              ;SAVE IT
4C26 21A04B     01500         LD      HL,FLCLR        ;17 SPACES
4C29 011100     01510         LD      BC,17           ;CHARACTER COUNT
4C2C EDB0       01520         LDIR
4C2E CD146C     01530         CALL    DELAY6          ;PAUSE
                01540 ;
4C31 D1         01550         POP     DE              ;SCREEN POSITION
4C32 E1         01560         POP     HL              ;SCORE POINTER
4C33 C1         01570         POP     BC              ;FLASH COUNT
4C34 10E3       01580         DJNZ    FLSHLP
4C36 C9         01590         RET
                01600 ;
4C37 3ABF4C     01610 SMART   LD      A,(SMTFLG)      ;CAN WE DO A SMART BOMB ?
4C3A B7         01620         OR      A
4C3B C0         01630         RET     NZ              ;RETURN IF NO
                01640 ;
4C3C 3A9F73     01650         LD      A,(SMTCNT)      ;ANY SMART BOMBS LEFT ?
4C3F B7         01660         OR      A
4C40 C8         01670         RET     Z               ;RETURN IF NO
                01680 ;
4C41 3D         01690         DEC     A               ;UPDATE COUNT
4C42 329F73     01700         LD      (SMTCNT),A
                01710 ;
4C45 3E01       01720         LD      A,1             ;FLAG SMART BOMB
4C47 32A673     01730         LD      (SPFLG),A       ;IN PROGRESS.
                01740 ;
4C4A 3AFF44     01750         LD      A,(MASK)
4C4D CBBF       01760         RES     7,A             ;DISABLE INTERRUPTS
```

```
4C4F 32FF44   01770          LD      (MASK),A
4C52 3AA173   01780          LD      A,(SHROW)        ;SAVE SHIP ROW
4C55 32A573   01790          LD      (ROWSAV),A
              01800 ;
4C58 21803C   01810 SMTLP    LD      HL,3C80H         ;START POS. FOR SCAN
4C5B 3E02     01820          LD      A,2              ;ROW # TO START
4C5D 32A173   01830          LD      (SHROW),A
4C60 01400E   01840          LD      BC,0E40H         ;ROW/COL COUNT
              01850 ;
4C63 C5       01860 ROWLOP   PUSH    BC               ;SAVE COUNT
4C64 41       01870          LD      B,C              ;GET COLUMN COUNT
              01880 ;
4C65 7E       01890 SMTLOP   LD      A,(HL)           ;GET CHARACTER
4C66 FEC0     01900          CP      192              ;ALIEN ?
4C68 D4B44C   01910          CALL    NC,SMTHIT        ;CALL IF YES
4C6B 23       01920          INC     HL               ;NEXT SCREEN POSITION
4C6C 10F7     01930          DJNZ    SMTLOP
              01940 ;
4C6E 3AA173   01950          LD      A,(SHROW)        ;NEXT ROW
4C71 3C       01960          INC     A
4C72 32A173   01970          LD      (SHROW),A
4C75 C1       01980          POP     BC               ;GET ROW COUNT
4C76 10EB     01990          DJNZ    ROWLOP
              02000 ;
4C78 3ABF4C   02010          LD      A,(SMTFLG)
4C7B B7       02020          OR      A                ;DONE YET ?
4C7C 200C     02030          JR      NZ,SMTEXT        ;JUMP IF YES
              02040 ;
4C7E 32C975   02050          LD      (PODTBL),A       ;CLEAR PODS
4C81 3C       02060          INC     A
4C82 32BF4C   02070          LD      (SMTFLG),A       ;FLAG SECOND TIME
4C85 CD034B   02080          CALL    CHARAC           ;REDRAW REMAINING ALIENS
4C88 18CE     02090          JR      SMTLP
              02100 ;
4C8A CD535C   02110 SMTEXT   CALL    GMECLR           ;PARTIAL SCREEN CLEAR
4C8D 3AA573   02120          LD      A,(ROWSAV)       ;RESTORE SHIP ROW
4C90 32A173   02130          LD      (SHROW),A
4C93 AF       02140          XOR     A
4C94 329D75   02150          LD      (SHOTBL),A       ;CLEAR SHOT TABLE
4C97 32A673   02160          LD      (SPFLG),A        ;CLEAR SMART FLAG
4C9A 3A0152   02170          LD      A,(PLFLG)
4C9D B7       02180          OR      A                ;WHO'S UP ?
4C9E 3A9F73   02190          LD      A,(SMTCNT)
4CA1 2005     02200          JR      NZ,TWO4          ;JUMP IF PLAYER 2
              02210 ;
4CA3 328773   02220          LD      (S1TCNT),A       ;UPDATE SMART BOMB COUNT
4CA6 1803     02230          JR      SMDON
              02240 ;
4CA8 329673   02250 TWO4     LD      (S2TCNT),A       ;UPDATE SMART BOMB COUNT
4CAB 3AFF44   02260 SMDON    LD      A,(MASK)         ;ENABLE INTERRUPTS
4CAE CBFF     02270          SET     7,A
4CB0 32FF44   02280          LD      (MASK),A
4CB3 C9       02290          RET
              02300 ;
4CB4 FEFD     02310 SMTHIT   CP      253              ;MAN ?
4CB6 C8       02320          RET     Z                ;RETURN IF YES
```

```
4CB7 E5         02330           PUSH    HL                  ;SAVE SCREEN POINTER
4CB8 C5         02340           PUSH    BC                  ;SAVE COUNTER
4CB9 CD7859     02350           CALL    EXPL1               ;FIND OUT WHAT WAS HIT
4CBC C1         02360           POP     BC
4CBD E1         02370           POP     HL
4CBE C9         02380           RET
                02390 ;
4CBF 00         02400 SMTFLG    DB      0
4CC0 00         02410 HYPFLG    DB      0
4CC1 02         02420 PLYCNT    DB      2
                02430 ;
4CC2 CD535C     02440 OVER      CALL    GMECLR              ;PARTIAL SCREEN CLEAR
4CC5 CDCB50     02450           CALL    SCORE               ;DISPLAY SCORE
4CC8 3A0152     02460           LD      A,(PLFLG)           ;GET PLAYER UP
4CCB C631       02470           ADD     A,31H               ;CONVERT TO ASCII
4CCD 320C4D     02480           LD      (PLYEND),A          ;PUT IN DISPLAY MESSAGE
4CD0 11963D     02490           LD      DE,3D96H            ;SCREEN POSITION
4CD3 21F94C     02500           LD      HL,MSGEND           ;MESSAGE POINTER
4CD6 0606       02510           LD      B,6                 ;LOOP COUNT
4CD8 C5         02520 OVLP1     PUSH    BC
4CD9 E5         02530           PUSH    HL
4CDA D5         02540           PUSH    DE
4CDB 011400     02550           LD      BC,20               ;CHARACTER COUNT
4CDE EDB0       02560           LDIR
4CE0 CD146C     02570           CALL    DELAY6              ;PAUSE
4CE3 D1         02580           POP     DE                  ;GET SCREEN POINTER
4CE4 D5         02590           PUSH    DE
4CE5 210D4D     02600           LD      HL,OVCLR            ;20 SPACES
4CE8 011400     02610           LD      BC,20               ;CHARACTER COUNT
4CEB EDB0       02620           LDIR
4CED CD146C     02630           CALL    DELAY6              ;PAUSE
4CF0 D1         02640           POP     DE                  ;SCREEN POINTER
4CF1 E1         02650           POP     HL                  ;MESSAGE POINTER
4CF2 C1         02660           POP     BC                  ;LOOP COUNT
4CF3 10E3       02670           DJNZ    OVLP1
4CF5 CDEC6F     02680           CALL    TOPCHK              ;CHECK FOR TOP TEN
4CF8 C9         02690           RET
                02700 ;
4CF9 47         02710 MSGEND    DM      'GAME OVER - PLAYER '
4D0C 00         02720 PLYEND    DB      0
4D0D 20         02730 OVCLR     DM      '                    '
                02740 ;
4D21 7E         02750 COMP      LD      A,(HL)              ;GET SCREEN CHARACTER
4D22 FEC0       02760           CP      192                 ;GRAPHICS ?
4D24 3003       02770           JR      NC,CMPSKP           ;JUMP IF NO
4D26 EE3F       02780           XOR     3FH                 ;REVERSE THE CHARACTER
4D28 77         02790           LD      (HL),A              ;PUT BACK ON SCREEN
4D29 23         02800 CMPSKP    INC     HL                  ;NEXT POSITION
4D2A 0B         02810           DEC     BC                  ;UPDATE COUNT
4D2B 78         02820           LD      A,B
4D2C B1         02830           OR      C                   ;DONE ?
4D2D C8         02840           RET     Z                   ;RETURN IF YES
4D2E 18F1       02850           JR      COMP                ;KEEP ON TRUCKIN'
                02860 ;
4D30 3AA673     02870 SPACE     LD      A,(SPFLG)           ;SMART BOMB IN
4D33 B7         02880           OR      A                   ;PROGRESS ?
```

```
4D34 C0         02890        RET     NZ             ;RETURN IF YES
4D35 3C         02900        INC     A              ;CANCEL MOUNTAINS
4D36 32A073     02910        LD      (DRWFLG),A
                02920 ;
4D39 3AFF44     02930        LD      A,(MASK)
4D3C CBBF       02940        RES     7,A            ;DISABLE INTERRUPTS
4D3E 32FF44     02950        LD      (MASK),A
                02960 ;
                02970        IFEQ    TLK,1          ;ASSEMBLE IF TALK VERSION
                02980        LD      A,(TSPFLG)
                02990        XOR     1
                03000        LD      (TSPFLG),A
                03010        LD      HL,BOYD        ;TALK DATA
                03020        JR      Z,BOTD2X
                03030        LD      HL,YINTD       ;TALK DATA
                03040 BOTD2X CALL    TALK           ;MAKE NOISE
                03050        ENDIF
                03060 ;
4D41 21803C     03070        LD      HL,3C80H       ;SCREEN POSITION
4D44 017F03     03080        LD      BC,37FH        ;CHARACTER COUNT
4D47 3E08       03090        LD      A,8            ;LOOP COUNT
                03100 ;
4D49 F5         03110 CMPLP1 PUSH    AF             ;SAVE COUNT
4D4A E5         03120        PUSH    HL             ;SAVE SCREEN POSITION
4D4B C5         03130        PUSH    BC             ;SAVE CHARACTER COUNT
4D4C CD214D     03140        CALL    COMP           ;COMPLEMENT SCREEN
4D4F C1         03150        POP     BC
4D50 E1         03160        POP     HL
4D51 F1         03170        POP     AF
4D52 3D         03180        DEC     A
4D53 20F4       03190        JR      NZ,CMPLP1      ;DO IT 8 TIMES
                03200 ;
4D55 3E08       03210        LD      A,8
4D57 32C574     03220        LD      (CRUTBL-1),A   ;DOUBLE MAX CRUISER COUNT
4D5A 0608       03230        LD      B,8            ;8 MUTANTS
4D5C CD204A     03240        CALL    MUTIT2
4D5F 060A       03250        LD      B,10           ;10 LANDERS
4D61 CDBC49     03260        CALL    LNDIT
4D64 CD535C     03270        CALL    GMECLR         ;PARTIAL SCREEN CLEAR
4D67 CDCB50     03280        CALL    SCORE          ;DISPLAY SCORE
4D6A ED5BA773   03290        LD      DE,(SHPOS)     ;GET SHIP POSITION
4D6E CDA157     03300        CALL    UPDATE         ;DRAW SHIP
4D71 C3C24A     03310        JP      GAME2
                03320 ;
4D74            01830 *GET   HYPER/ASM              ;WARP DRIVE ROUTINES
                00010 ;**************************
                00020 ;**    HYPER SPACE      **
                00030 ;**************************
                00040 ;
4D74 3AC04C     00050 HYPER  LD      A,(HYPFLG)
4D77 B7         00060        OR      A              ;CAN WE DO IT ?
4D78 C0         00070        RET     NZ             ;RETURN IF NO
4D79 3C         00080        INC     A
4D7A 32C04C     00090        LD      (HYPFLG),A     ;SET FLAG
4D7D AF         00100        XOR     A
4D7E 32FF44     00110        LD      (MASK),A       ;DISABLE INTERRUPTS
```

```
            00120 ;
            00130          IFEQ    TLK,1          ;ASSEMBLE IF TALK VER.
            00140          LD      HL,COWARD      ;TALK DATA
            00150          CALL    TALK           ;MAKE NOISE
            00160          ENDIF
            00170 ;
4D81 3EC8   00180          LD      A,200          ;STARTING TONE
4D83 322C4F 00190          LD      (NUM),A
            00200 ;
4D86 3EC6   00210          LD      A,0C6H         ;ADD INSTRUCTION
4D88 32154F 00220          LD      (TNECNG),A     ;MODIFY CODE
            00230 ;
4D8B 2AA773 00240          LD      HL,(SHPOS)     ;GET SHIP POSITION
4D8E 01F3FF 00250          LD      BC,-13
4D91 09     00260          ADD     HL,BC          ;BACK IT UP 13 SPACES
4D92 EB     00270          EX      DE,HL          ;PUT IN DE
            00280 ;
4D93 215B4E 00290          LD      HL,HYPDT1      ;FIRST WARP PHASE
4D96 CD3D4E 00300          CALL    HYPDRW         ;DRAW IT
            00310 ;
4D99 21794E 00320          LD      HL,HYPDT2      ;SECOND PHASE
4D9C CD3D4E 00330          CALL    HYPDRW         ;DRAW IT
            00340 ;
4D9F 21974E 00350          LD      HL,HYPDT3      ;THIRD PHASE
4DA2 CD3D4E 00360          CALL    HYPDRW         ;DRAW IT
            00370 ;
4DA5 21B54E 00380          LD      HL,HYPDT4      ;FOURTH PHASE
4DA8 CD3D4E 00390          CALL    HYPDRW         ;DRAW IT
            00400 ;
4DAB 21D34E 00410          LD      HL,HYPDT5      ;FIFTH PHASE
4DAE CD3D4E 00420          CALL    HYPDRW         ;DRAW IT
            00430 ;
4DB1 21F14E 00440          LD      HL,HYPDT6      ;SIXTH PHASE
4DB4 E5     00450          PUSH    HL             ;SAVE IT
4DB5 CD3D4E 00460          CALL    HYPDRW         ;DRAW IT
            00470 ;
4DB8 011E00 00480          LD      BC,30          ;CHARACTER COUNT
4DBB EB     00490          EX      DE,HL
4DBC CD494E 00500          CALL    HPCP2          ;COMPLEMENT HYPER DATA
            00510 ;
4DBF 21803C 00520          LD      HL,3C80H       ;SCREEN POS. FOR COMPLEMENT
4DC2 017F03 00530          LD      BC,37FH        ;CHARACTER COUNT
4DC5 3E03   00540          LD      A,3            ;LOOP COUNT
            00550 ;
4DC7 F5     00560 HPLP1    PUSH    AF             ;SAVE COUNT
4DC8 E5     00570          PUSH    HL             ;SAVE SCREEN POSITION
4DC9 C5     00580          PUSH    BC             ;SAVE CHARACTER COUNT
4DCA CD214D 00590          CALL    COMP           ;COMPLEMENT SCREEN
4DCD C1     00600          POP     BC             ;CHARACTER COUNT
4DCE E1     00610          POP     HL             ;SCREEN POSITION
4DCF F1     00620          POP     AF             ;LOOP COUNT
4DD0 3D     00630          DEC     A
4DD1 20F4   00640          JR      NZ,HPLP1       ;LOOP TIL DONE
            00650 ;
4DD3 ED5F   00660          LD      A,R            ;BUILD A RANDOM #
4DD5 67     00670          LD      H,A
```

```
4DD6 ED5F      00680          LD      A,R
4DD8 84        00690          ADD     A,H
4DD9 67        00700          LD      H,A
4DDA 3ABC73    00710          LD      A,(WINDOW)       ;GET CURRENT WINDOW POSITION
4DDD 84        00720          ADD     A,H              ;MOVE IT
4DDE 32BC73    00730          LD      (WINDOW),A       ;SAVE NEW POSITION
4DE1 F5        00740          PUSH    AF
4DE2 210046    00750          LD      HL,MOUNT         ;CALCULATE NEW MOUNTAIN
4DE5 85        00760          ADD     A,L
4DE6 6F        00770          LD      L,A
4DE7 22BA73    00780          LD      (DPOINT),HL      ;SAVE NEW POINTER
4DEA F1        00790          POP     AF
4DEB C61F      00800          ADD     A,31             ;OFFSET FOR SHIP POS.
4DED 32A273    00810          LD      (SHCOL),A        ;NEW SHIP COLUMN
4DF0 3E05      00820          LD      A,5
4DF2 32A173    00830          LD      (SHROW),A        ;NEW SHIP ROW
4DF5 CD535C    00840          CALL    GMECLR           ;PARTIAL SCREEN CLEAR
4DF8 CDCB50    00850          CALL    SCORE            ;DISPLAY SCORE
4DFB CDF958    00860          CALL    DRAW             ;DRAW MOUNTAINS
4DFE 215F3D    00870          LD      HL,3D5FH         ;SHIP SCREEN POS.
4E01 22A773    00880          LD      (SHPOS),HL
               00890 ;
4E04 11523D    00900          LD      DE,3D52H         ;SCREEN POSITION
4E07 3ED6      00910          LD      A,0D6H           ;SUB INSTRUCTION
4E09 32154F    00920          LD      (TNECNG),A       ;MODIFY CODE
               00930 ;
4E0C E1        00940          POP     HL               ;SIXTH WARP PHASE
4E0D CD3D4E    00950          CALL    HYPDRW           ;DRAW IT
               00960 ;
4E10 21D34E    00970          LD      HL,HYPDT5        ;FIFTH PHASE
4E13 CD3D4E    00980          CALL    HYPDRW           ;DRAW IT
               00990 ;
4E16 21B54E    01000          LD      HL,HYPDT4        ;FOURTH PHASE
4E19 CD3D4E    01010          CALL    HYPDRW           ;DRAW IT
               01020 ;
4E1C 21974E    01030          LD      HL,HYPDT3        ;THIRD PHASE
4E1F CD3D4E    01040          CALL    HYPDRW           ;DRAW IT
               01050 ;
4E22 21794E    01060          LD      HL,HYPDT2        ;SECOND PHASE
4E25 CD3D4E    01070          CALL    HYPDRW           ;DRAW IT
               01080 ;
4E28 215B4E    01090          LD      HL,HYPDT1        ;FIRST PHASE
4E2B CD3D4E    01100          CALL    HYPDRW           ;DRAW IT
               01110 ;
4E2E CD034B    01120          CALL    CHARAC           ;UPDATE ALL ALIENS
4E31 2AA773    01130          LD      HL,(SHPOS)       ;GET SHIP POSITION
4E34 2B        01140          DEC     HL
4E35 3680      01150          LD      (HL),128
4E37 CDB248    01160          CALL    VECTOR           ;UPDATE VECTORS
4E3A C3C24A    01170          JP      GAME2            ;CONTINUE GAME
               01180 ;
4E3D 011E00    01190 HYPDRW   LD      BC,30            ;CHARACTER COUNT
4E40 D5        01200          PUSH    DE               ;SCREEN POS.
4E41 C5        01210          PUSH    BC               ;COUNT
4E42 EDB0      01220          LDIR
4E44 CD0F4F    01230          CALL    HYPTNE           ;MAKE SOUND
```

```
4E47 C1         01240        POP    BC                  ;COUNT
4E48 E1         01250        POP    HL                  ;SCREEN POSITION
4E49 E5         01260 HPCP2  PUSH   HL
4E4A 41         01270        LD     B,C                 ;COUNT TO B
4E4B 7E         01280 HPLPCP LD     A,(HL)              ;GET CHARACTER
4E4C FE81       01290        CP     129                 ;GRAPHIC ?
4E4E 3803       01300        JR     C,HPCP1             ;JUMP IF NO
4E50 EE3F       01310        XOR    3FH                 ;COMPLEMENT
4E52 77         01320        LD     (HL),A              ;PUT BACK ON SCREEN
4E53 23         01330 HPCP1  INC    HL
4E54 10F5       01340        DJNZ   HPLPCP
4E56 CD1E55     01350        CALL   DELAY               ;PAUSE
4E59 D1         01360        POP    DE                  ;SCREEN POS.
4E5A C9         01370        RET
                01380 ;
                01390 ;      *****  WARP DRIVE DATA   *****
                01400 ;
4E5B 80         01410 HYPDT1 DB     128,128,128,128,128,128,128,128,128,128
4E65 80         01420        DB     128,128,146,150,134,146,164,128,128,128
4E6F 80         01430        DB     128,128,128,128,128,128,128,128,128,128
                01440 ;
4E79 80         01450 HYPDT2 DB     128,128,128,128,128,128,128,128,128,128
4E83 A1         01460        DB     161,153,134,152,153,136,146,161,132,166
4E8D 80         01470        DB     128,128,128,128,128,128,128,128,128,128
                01480 ;
4E97 80         01490 HYPDT3 DB     128,128,128,128,128,128,128,128,161,130
4EA1 92         01500        DB     146,160,129,132,152,129,132,129,132,152
4EAB 82         01510        DB     130,136,162,128,128,128,128,128,128,128
                01520 ;
4EB5 80         01530 HYPDT4 DB     128,128,128,128,128,132,161,136,136,130
4EBF A0         01540        DB     160,136,130,144,132,136,136,144,132,160
4EC9 82         01550        DB     130,160,136,160,136,128,128,128,128,128
                01560 ;
4ED3 80         01570 HYPDT5 DB     128,128,132,144,129,132,146,136,136,161
4EDD A0         01580        DB     160,129,160,130,136,162,130,136,160,136
4EE7 88         01590        DB     136,160,136,144,144,132,164,128,128,128
                01600 ;
4EF1 A1         01610 HYPDT6 DB     161,136,136,137,160,136,160,129,144,132
4EFB 84         01620        DB     132,129,136,160,132,132,129,144,132,132
4F05 82         01630        DB     130,144,130,164,136,160,129,144,153,136
                01640 ;
4F0F C5         01650 HYPTNE PUSH   BC                  ;WARP DRIVE SOUND
4F10 F5         01660        PUSH   AF
4F11 3A2C4F     01670        LD     A,(NUM)             ;STARTING TONE
4F14 47         01680        LD     B,A
4F15 D614       01690 TNECNG SUB    20                  ;CHANGES TO SUB/ADD
4F17 322C4F     01700        LD     (NUM),A
4F1A 3E01       01710        LD     A,1
4F1C C5         01720 HPLLP  PUSH   BC
4F1D EE03       01730        XOR    3                   ;TOGGLE TONE BIT
4F1F D3FF       01740        OUT    (255),A
4F21 10FE       01750        DJNZ   $                   ;DELAY FOR PITCH
4F23 C1         01760        POP    BC
4F24 05         01770        DEC    B                   ;CHANGE PITCH DELAY
4F25 05         01780        DEC    B
4F26 05         01790        DEC    B
```

```
4F27 10F3      01800          DJNZ     HPLLP
4F29 F1        01810          POP      AF
4F2A C1        01820          POP      BC
4F2B C9        01830          RET
               01840 ;
4F2C 00        01850 NUM      DB       0
               01860 ;
               01870 ;************************************
               01880 ;**       SOUND ROUTINES            **
               01890 ;************************************
               01900 ;
4F2D C5        01910 DOWN4    PUSH     BC                 ;SHOT SOUND
4F2E 060A      01920          LD       B,10               ;INITIAL PITCH DELAY
4F30 3E02      01930          LD       A,2
4F32 EE03      01940 DSWNP    XOR      3                  ;TOGGLE TONE BIT
4F34 D3FF      01950          OUT      (255),A
4F36 C5        01960          PUSH     BC
4F37 10FE      01970          DJNZ     $                  ;PITCH DELAY
4F39 C1        01980          POP      BC
4F3A 04        01990          INC      B                  ;CHANGE PITCH DELAY
4F3B 04        02000          INC      B
4F3C F2324F    02010          JP       P,DSWNP
4F3F C1        02020          POP      BC
4F40 C9        02030          RET
               02040 ;
               02050 ;**      SPECIAL SOUND     **
               02060 ;
4F41 1614      02070 MOOSE    LD       D,20               ;THIS ROUTINE GENERATES
4F43 1EFF      02080          LD       E,255              ;A DISTORTED SOUND BY
4F45 DD21914F  02090          LD       IX,MTONE1          ;TOGGLING THE TONE BIT
               02100                                      ;AT TWO DIFFERENT RATES
               02110                                      ;SIMULTANEOUSLY WITH BOTH
               02120                                      ;OF THE RATES CONSTANTLY
               02130                                      ;CHANGING.
4F49 DD7200    02140 MOOSELP  LD       (IX+0),D           ;TONE 1
4F4C DD7301    02150          LD       (IX+1),E           ;TONE 2
4F4F 218002    02160          LD       HL,0280H           ;DURATION COUNT
4F52 01FFFF    02170          LD       BC,-1              ;SUBTRACT AMOUNT
4F55 3E02      02180          LD       A,2
4F57 CD8C4F    02190          CALL     MTONE              ;TOGGLE TONE BIT
4F5A 1810      02200          JR       MDOWN2
               02210 ;
4F5C DD5600    02220 MDO1     LD       D,(IX+0)           ;GET TONE VALUE
4F5F CD8C4F    02230          CALL     MTONE              ;TOGGLE TONE BIT
4F62 1808      02240          JR       MDOWN2
               02250 ;
4F64 DD5E01    02260 MDO2     LD       E,(IX+1)           ;GET TONE VALUE
4F67 CD8C4F    02270          CALL     MTONE              ;TOGGLE TONE BIT
4F6A 1808      02280          JR       MDOWN1
               02290 ;
4F6C 1D        02300 MDOWN2   DEC      E                  ;TOGGLE TONE BIT ?
4F6D 28F5      02310          JR       Z,MDO2             ;JUMP IF YES
               02320 ;
4F6F B7        02330          OR       A                  ;CLEAR CARRY FLAG
4F70 ED4A      02340          ADC      HL,BC              ;UPDATE DURATION
4F72 280A      02350          JR       Z,MCNGSND          ;JUMP IF DONE
```

```
                02360 ;
4F74 15         02370 MDOWN1  DEC     D               ;TOGGLE TONE BIT ?
4F75 28E5       02380         JR      Z,MD01          ;JUMP IF YES
                02390 ;
4F77 B7         02400         OR      A               ;CLEAR CARRY FLAG
4F78 ED4A       02410         ADC     HL,BC           ;UPDATE DURATION
4F7A 2802       02420         JR      Z,MCNGSND       ;JUMP IF DONE
4F7C 18EE       02430         JR      MDOWN2
                02440 ;
4F7E DD5600     02450 MCNGSND LD      D,(IX+0)        ;GET TONE VALUE
4F81 14         02460         INC     D               ;UPDATE
4F82 7A         02470         LD      A,D
4F83 FEFA       02480         CP      250             ;DONE ?
4F85 C8         02490         RET     Z               ;RETURN IF YES
4F86 DD5E01     02500         LD      E,(IX+1)
4F89 C3494F     02510         JP      MOOSELP
                02520 ;
4F8C EE03       02530 MTONE   XOR     3
4F8E D3FF       02540         OUT     (255),A
4F90 C9         02550         RET
                02560 ;
4F91 0000       02570 MTONE1  DW      00
                02580 ;
4F93 F5         02590 BINK    PUSH    AF              ;SOUND ROUTINE THAT IS USED
4F94 C5         02600         PUSH    BC              ;WHEN A MAN IS CAUGHT IN
4F95 D5         02610         PUSH    DE              ;MID AIR OR IS SET DOWN
4F96 110003     02620         LD      DE,0300H        ;BY THE SHIP
4F99 060C       02630         LD      B,12
4F9B 3E02       02640         LD      A,2
4F9D C5         02650 BNKLP1  PUSH    BC
4F9E EE03       02660         XOR     3               ;TOGGLE TONE BIT
4FA0 F5         02670 BNKLP2  PUSH    AF
4FA1 D3FF       02680         OUT     (255),A
4FA3 1B         02690         DEC     DE              ;UPDATE DURATION COUNTER
4FA4 7A         02700         LD      A,D
4FA5 B3         02710         OR      E
4FA6 2806       02720         JR      Z,BNKEXT        ;JUMP IF DONE
4FA8 F1         02730         POP     AF
4FA9 10F5       02740         DJNZ    BNKLP2
4FAB C1         02750         POP     BC
4FAC 18EF       02760         JR      BNKLP1
                02770 ;
4FAE F1         02780 BNKEXT  POP     AF
4FAF C1         02790         POP     BC
4FB0 D1         02800         POP     DE
4FB1 C1         02810         POP     BC
4FB2 F1         02820         POP     AF
4FB3 C9         02830         RET
                02840 ;
4FB4            01840 *GET    INTERR/ASM              ;INTERRUPT HANDLER/ SCORE ROUTINES
                00010 ;******************************************
                00020 ;**          INTERRUPT HANDLER           **
                00030 ;**                                      **
                00040 ;**      MASK = INTERRUPT FLAGS          **
                00050 ;**                                      **
                00060 ;**      BIT     0 = ALIEN EXPLODE FLAG  **
```

```
               00070 ;**            1 = SCORE UPDATE        **
               00080 ;**            6 = STROBE KEYBOARD     **
               00090 ;**            7 = CANCEL INTERRUPT    **
               00100 ;******************************************
               00110 ;
4FB4 F5        00120 INTERR  PUSH   AF
4FB5 DBE0      00130         IN     A,(0E0H)         ;CLEAR INTERRUPT
4FB7 DBEC      00140         IN     A,(0ECH)         ;  ''     ''
4FB9 3AFF44    00150         LD     A,(MASK)         ;GET MASK VALUE
4FBC CB7F      00160         BIT    7,A              ;IGNORE INTERRUPT ?
4FBE 282A      00170         JR     Z,INTEX2         ;JUMP IF YES
4FC0 E5        00180         PUSH   HL
4FC1 D5        00190         PUSH   DE
4FC2 C5        00200         PUSH   BC
4FC3 DDE5      00210         PUSH   IX
4FC5 FDE5      00220         PUSH   IY
4FC7 21E34F    00230         LD     HL,INTEXT
4FCA E5        00240         PUSH   HL               ;RETURN ADDRESS
4FCB CB77      00250         BIT    6,A              ;STROBE KEYBOARD ?
4FCD C2DD52    00260         JP     NZ,STROBE        ;JUMP IF YES
               00270 ;
4FD0 F5        00280         PUSH   AF               ;SAVE MASK VALUE
4FD1 CD0252    00290         CALL   TIMER            ;TIMER FOR CRUISER/BOMBER
4FD4 CD8252    00300         CALL   LNDON            ;TIMER FOR LANDER
               00310 ;
4FD7 F1        00320         POP    AF               ;GET MASK VALUE
4FD8 CB4F      00330         BIT    1,A              ;UPDATE SCORE ?
4FDA C2CB50    00340         JP     NZ,SCORE         ;JUMP IF YES
               00350 ;
4FDD CB47      00360         BIT    0,A              ;EXPLODE ALIEN ?
4FDF C2F552    00370 EXCNGO  JP     NZ,EXPLOD        ;JUMP IF YES
               00380 ;
4FE2 E1        00390         POP    HL               ;RETURN ADDRESS
4FE3 FDE1      00400 INTEXT  POP    IY
4FE5 DDE1      00410         POP    IX
4FE7 C1        00420         POP    BC
4FE8 D1        00430         POP    DE
4FE9 E1        00440         POP    HL
4FEA F1        00450 INTEX2  POP    AF
4FEB FB        00460         EI
4FEC C9        00470         RET
               00480 ;
               00490 ;       ** REGISTER HIT & ADD SCORE **
               00500 ;
4FED E5        00510 HIT     PUSH   HL
4FEE 3A0152    00520         LD     A,(PLFLG)        ;WHICH PLAYER ?
4FF1 B7        00530         OR     A
4FF2 2011      00540         JR     NZ,PLAY2         ;JUMP IF PLAYER 2
               00550 ;
4FF4 2AFB51    00560         LD     HL,(PL1SCL)      ;3 BYTE SCORE
4FF7 09        00570         ADD    HL,BC            ;ADD IN LOW ORDER
4FF8 22FB51    00580         LD     (PL1SCL),HL
4FFB 3AFD51    00590         LD     A,(PL1SCH)
4FFE CE00      00600         ADC    A,0              ;ADD IN CARRY
5000 32FD51    00610         LD     (PL1SCH),A
5003 180F      00620         JR     SCRSK1
```

```
               00630 ;
5005 2AFE51    00640 PLAY2   LD      HL,(PL2SCL)      ;3 BYTE SCORE
5008 09        00650         ADD     HL,BC            ;ADD IN LOW ORDER
5009 22FE51    00660         LD      (PL2SCL),HL
500C 3A0052    00670         LD      A,(PL2SCH)
500F CE00      00680         ADC     A,0              ;ADD IN CARRY
5011 320052    00690         LD      (PL2SCH),A
               00700 ;
5014 2A9B73    00710 SCRSK1  LD      HL,(CMEAGN)
5017 09        00720         ADD     HL,BC            ;ADD TO 10,000 COUNTER
5018 229B73    00730         LD      (CMEAGN),HL
501B 011027    00740         LD      BC,10000
501E B7        00750         OR      A
501F ED42      00760         SBC     HL,BC            ;10,000 ROLLOVER ?
5021 3822      00770         JR      C,SCRSKP         ;JUMP IF NO
               00780 ;
5023 229B73    00790         LD      (CMEAGN),HL      ;SAVE OVERFLOW
5026 3A9E73    00800         LD      A,(SHPCNT)       ;GET SHIP COUNT
5029 3C        00810         INC     A                ;GIVE 'EM ANOTHER ONE
502A 329E73    00820         LD      (SHPCNT),A
502D 3A9F73    00830         LD      A,(SMTCNT)       ;GET SMART BOMB COUNT
5030 3C        00840         INC     A                ;ADD 1
5031 329F73    00850         LD      (SMTCNT),A
5034 2A9E73    00860         LD      HL,(SHPCNT)      ;GET SMART/SHIP COUNT
5037 3A0152    00870         LD      A,(PLFLG)
503A B7        00880         OR      A                ;WHO'S UP ?
503B 2005      00890         JR      NZ,XX            ;JUMP IF PLAYER 2
               00900 ;
503D 228673    00910         LD      (S1PCNT),HL      ;SAVE NEW COUNTS
5040 1803      00920         JR      SCRSKP
               00930 ;
5042 229573    00940 XX      LD      (S2PCNT),HL
5045 3AFF44    00950 SCRSKP  LD      A,(MASK)         ;SET SCORE FLAG
5048 CBCF      00960         SET     1,A
504A 32FF44    00970         LD      (MASK),A
504D E1        00980         POP     HL
504E 3AA073    00990         LD      A,(DRWFLG)
5051 B7        01000         OR      A                ;MOUNTAINS ACTIVE ?
5052 C0        01010         RET     NZ               ;RETURN IF NO
               01020 ;
5053 3A9D73    01030         LD      A,(WAVCNT)
5056 FE19      01040         CP      25               ;HIT 25 ALIENS ?
5058 D8        01050         RET     C                ;RETURN IF NO
               01060 ;
5059 AF        01070         XOR     A                ;DISABLE INTERRUPTS
505A 32FF44    01080         LD      (MASK),A
505D 329D73    01090         LD      (WAVCNT),A       ;CLEAR WAVE COUNT
5060 3A0174    01100         LD      A,(MANTBL-1)     ;GET ACTIVE MAN COUNT
5063 B7        01110         OR      A
5064 C8        01120         RET     Z                ;RETURN IF NONE ACTIVE
5065 F8        01130         RET     M                ;RETURN IF ERROR
               01140 ;
5066 F5        01150         PUSH    AF               ;SAVE COUNT
5067 CD535C    01160         CALL    GMECLR           ;PARTIAL SCREEN CLEAR
506A F1        01170         POP     AF               ;GET COUNT
506B 47        01180         LD      B,A
```

```
506C C5         01190          PUSH    BC
506D 216400     01200          LD      HL,100          ;100 POINTS/MAN REMAINING
5070 E5         01210          PUSH    HL
5071 D1         01220          POP     DE
5072 05         01230          DEC     B
5073 2803       01240          JR      Z,BONEXT        ;JUMP IF ONLY 1 MAN
                01250 ;
5075 19         01260 ADDLP    ADD     HL,DE           ;CALCULATE BONUS
5076 10FD       01270          DJNZ    ADDLP
                01280 ;
5078 E5         01290 BONEXT   PUSH    HL              ;SAVE BONUS
5079 E5         01300          PUSH    HL
507A C1         01310          POP     BC              ;BONUS TO BC
507B CDED4F     01320          CALL    HIT             ;ADD TO PLAYERS SCORE
507E E1         01330          POP     HL              ;GET BONUS
507F DD21BF52   01340          LD      IX,BONSCR
5083 CDB151     01350          CALL    BIN1            ;CONVERT TO ASCII
5086 21C552     01360          LD      HL,BONMSG       ;"BONUS"
5089 11803D     01370          LD      DE,3D80H        ;SCREEN POSITION
508C CD555D     01380          CALL    PRNT            ;DISPLAY MESSAGE
508F C1         01390          POP     BC
5090 3EFD       01400          LD      A,253           ;MAN CHARACTER
5092 12         01410 BONLP1   LD      (DE),A          ;DRAW MEN
5093 13         01420          INC     DE
5094 13         01430          INC     DE
5095 10FB       01440          DJNZ    BONLP1
5097 21D152     01450          LD      HL,BNMSG2       ;"X 100"
509A CD555D     01460          CALL    PRNT
509D 21BF52     01470          LD      HL,BONSCR       ;PRINT BONUS
50A0 CD555D     01480          CALL    PRNT
                01490 ;
                01500          IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
                01510          LD      HL,BONUSD       ;TALK DATA
                01520          CALL    TALK            ;MAKE NOISE
                01530          ENDIF
                01540 ;
50A3 0604       01550          LD      B,4
50A5 C5         01560 BONLP4   PUSH    BC              ;DELAY
50A6 010000     01570          LD      BC,0
50A9 0B         01580 BONLP3   DEC     BC
50AA 78         01590          LD      A,B
50AB B1         01600          OR      C
50AC 20FB       01610          JR      NZ,BONLP3
50AE C1         01620          POP     BC
50AF 10F4       01630          DJNZ    BONLP4
                01640 ;
50B1 3E0A       01650          LD      A,10
50B3 32BD73     01660          LD      (LNDTBL-1),A    ;NEW ACTIVE COUNT
50B6 CD0349     01670          CALL    TABINT          ;MOVE VARIABLES
50B9 CD4349     01680          CALL    INIT3           ;INIT CHARACTERS
50BC CDCB50     01690          CALL    SCORE           ;DISPLAY SCORE
50BF CD3048     01700          CALL    INIT2
50C2 CD054C     01710          CALL    FLASH           ;FLASH WHO'S UP
50C5 CDCB50     01720          CALL    SCORE           ;RE-DISPLAY SCORE
50C8 C3B84A     01730          JP      GAME            ;CONTINUE GAME
                01740 ;
```

```
              01750 ;        ** DISPLAY SCORE **
              01760 ;
50CB DD21694B 01770 SCORE    LD      IX,SCORE1       ;PLAYER 1 SCORE (ASCII)
50CF 2AFB51   01780          LD      HL,(PL1SCL)     ;GET LOW ORDER
50D2 3AFD51   01790          LD      A,(PL1SCH)      ;GET HI BYTE
50D5 CD3951   01800          CALL    BXDEC2          ;CONVERT TO 8 DIGIT DEC.
              01810 ;
50D8 DD218D4B 01820          LD      IX,SCORE2       ;PLAYER 2 SCORE (ASCII)
50DC 2AFE51   01830          LD      HL,(PL2SCL)     ;GET LOW ORDER
50DF 3A0052   01840          LD      A,(PL2SCH)      ;GET HI BYTE
50E2 CD3951   01850          CALL    BXDEC2          ;CONVERT
              01860 ;
50E5 215F4B   01870          LD      HL,SCMSG        ;PRINT SCORES
50E8 11403C   01880          LD      DE,3C40H
50EB 013F00   01890          LD      BC,63
50EE EDB0     01900          LDIR
              01910 ;
50F0 ED4B8673 01920          LD      BC,(S1PCNT)     ;C=SHIPS, B=BOMBS
50F4 21003C   01930          LD      HL,3C00H        ;START SCREEN POS.
50F7 CD0D51   01940          CALL    SPDRW
              01950 ;
50FA ED4B9573 01960          LD      BC,(S2PCNT)     ;C=SHIPS, B=BOMBS
50FE 212E3C   01970          LD      HL,3C2EH
5101 CD0D51   01980          CALL    SPDRW
              01990 ;
5104 3AFF44   02000          LD      A,(MASK)        ;CLEAR SCORE FLAG
5107 CB8F     02010          RES     1,A
5109 32FF44   02020          LD      (MASK),A
510C C9       02030          RET
              02040 ;
510D 78       02050 SPDRW    LD      A,B
510E 81       02060          ADD     A,C
510F ED44     02070          NEG
5111 C611     02080          ADD     A,17
5113 57       02090          LD      D,A             ;SPACES BETWEEN SHIPS & BOMBS
5114 79       02100          LD      A,C             ;SHIPS
5115 B7       02110          OR      A
5116 2809     02120          JR      Z,SPNEX1
5118 FA2151   02130          JP      M,SPNEX1
511B 36FF     02140 SPLP1    LD      (HL),255
511D 23       02150          INC     HL
511E 3D       02160          DEC     A
511F 20FA     02170          JR      NZ,SPLP1
5121 7A       02180 SPNEX1   LD      A,D             ;SPACES
5122 B7       02190          OR      A
5123 2809     02200          JR      Z,SPNEX2
5125 FA2E51   02210          JP      M,SPNEX2
5128 3680     02220 SPLP2    LD      (HL),128
512A 23       02230          INC     HL
512B 3D       02240          DEC     A
512C 20FA     02250          JR      NZ,SPLP2
512E 78       02260 SPNEX2   LD      A,B             ;BOMBS
512F B7       02270          OR      A
5130 C8       02280          RET     Z
5131 F8       02290          RET     M
5132 36EE     02300 SPLP3    LD      (HL),238
```

```
5134 23       02310        INC    HL
5135 3D       02320        DEC    A
5136 20FA     02330        JR     NZ,SPLP3
5138 C9       02340        RET
              02350 ;
5139 FD219151 02360 BXDEC2 LD     IY,PTAB2         ;POWER OF TEN TABLE
513D DDE5     02370        PUSH   IX
513F DD21A951 02380        LD     IX,TEMP          ;TEMPORARY STORAGE
5143 DDE5     02390        PUSH   IX
5145 0600     02400 LOOP2  LD     B,0              ;SUBTRACT COUNTER
5147 FD5E00   02410        LD     E,(IY)           ;GET TABLE VALUE
514A FD5601   02420        LD     D,(IY+1)
514D FD4E02   02430        LD     C,(IY+2)
5150 B7       02440 LOOP3  OR     A                ;CLEAR CARRY
5151 ED52     02450        SBC    HL,DE            ;SUBTRACT POWER OF TEN
5153 99       02460        SBC    A,C
5154 3803     02470        JR     C,JUMP2          ;JUMP IF BORROW
5156 04       02480        INC    B                ;UPDATE COUNTER
5157 18F7     02490        JR     LOOP3            ;LOOP TILL BORROW
              02500 ;
5159 19       02510 JUMP2  ADD    HL,DE            ;BRING BACK TO POSITIVE
515A 89       02520        ADC    A,C
515B F5       02530        PUSH   AF
              02540 ;
515C 78       02550        LD     A,B
515D C630     02560        ADD    A,30H            ;CONVERT TO ASCII
515F DD7700   02570        LD     (IX),A           ;SAVE IN SCORE STORAGE
5162 DD23     02580        INC    IX               ;BUMP POINTER
5164 FD23     02590        INC    IY               ;NEXT TABLE ENTRY
5166 FD23     02600        INC    IY
5168 FD23     02610        INC    IY
516A 7B       02620        LD     A,E
516B FE01     02630        CP     1                ;DONE ?
516D 2803     02640        JR     Z,BXOUT          ;JUMP IF YES
516F F1       02650        POP    AF
5170 18D3     02660        JR     LOOP2
              02670 ;
5172 F1       02680 BXOUT  POP    AF
5173 E1       02690        POP    HL               ;TEMP STORAGE
5174 DDE1     02700        POP    IX               ;SCREEN POS.
5176 0608     02710        LD     B,8
5178 7E       02720 BXLP1  LD     A,(HL)
5179 FE30     02730        CP     '0'
517B 200A     02740        JR     NZ,BXOUT2
517D DD360080 02750        LD     (IX),128
5181 23       02760        INC    HL
5182 DD23     02770        INC    IX
5184 10F2     02780        DJNZ   BXLP1
5186 C9       02790        RET
5187 DD7700   02800 BXOUT2 LD     (IX),A
518A DD23     02810        INC    IX
518C 23       02820        INC    HL
518D 7E       02830        LD     A,(HL)
518E 10F7     02840        DJNZ   BXOUT2
5190 C9       02850        RET
              02860 ;
```

```
5191 80        02870 PTAB2   DB      80H,96H,98H      ;10,000,000
5194 40        02880         DB      40H,42H,0FH      ;1,000,000
5197 A0        02890         DB      0A0H,86H,01H     ;100,000
519A 10        02900         DB      10H,27H,00H      ;10,000
519D E8        02910         DB      0E8H,03H,00H     ;1,000
51A0 64        02920         DB      64H,00H,00H      ;100
51A3 0A        02930         DB      0AH,00H,00H      ;10
51A6 01        02940         DB      01H,00H,00H      ;1
               02950 ;
51A9 00        02960 TEMP    DB      0,0,0,0,0,0,0,0
               02970 ;
51B1 FD21F151  02980 BIN1    LD      IY,PTAB1
51B5 DDE5      02990         PUSH    IX
51B7 AF        03000 BNLP0   XOR     A
51B8 FD5E00    03010         LD      E,(IY)
51BB FD5601    03020         LD      D,(IY+1)
51BE B7        03030 BNLP1   OR      A
51BF ED52      03040         SBC     HL,DE
51C1 3803      03050         JR      C,BNSKP1
51C3 3C        03060         INC     A
51C4 18F8      03070         JR      BNLP1
51C6 19        03080 BNSKP1  ADD     HL,DE
51C7 C630      03090         ADD     A,30H
51C9 DD7700    03100         LD      (IX),A
51CC DD23      03110         INC     IX
51CE FD23      03120         INC     IY
51D0 FD23      03130         INC     IY
51D2 7B        03140         LD      A,E
51D3 FE01      03150         CP      1
51D5 20E0      03160         JR      NZ,BNLP0
51D7 DDE1      03170         POP     IX
51D9 0604      03180         LD      B,4
51DB DD7E00    03190 BXLP2   LD      A,(IX)
51DE FE30      03200         CP      '0'
51E0 2009      03210         JR      NZ,BXOUT3
51E2 DD360080  03220         LD      (IX),128
51E6 DD23      03230         INC     IX
51E8 10F1      03240         DJNZ    BXLP2
51EA C9        03250         RET
51EB DD23      03260 BXOUT3  INC     IX
51ED 10FC      03270         DJNZ    BXOUT3
51EF C9        03280         RET
51F0 C9        03290         RET
               03300 ;
51F1 1027      03310 PTAB1   DW      10000
51F3 E803      03320         DW      1000
51F5 6400      03330         DW      100
51F7 0A00      03340         DW      10
51F9 0100      03350         DW      1
               03360 ;
51FB 0000      03370 PL1SCL  DW      00               ;PLAYER 1 SCORE (BINARY)
51FD 00        03380 PL1SCH  DB      0
               03390 ;
51FE 0000      03400 PL2SCL  DW      00               ;PLAYER 2 SCORE (BINARY)
5200 00        03410 PL2SCH  DB      0
               03420 ;
```

```
5201 00        03430 PLFLG   DB      0               ;PLAYER UP
               03440 ;
               03450 ;       *****  TIMER FOR CRUISER/BOMBER/POD  *****
               03460 ;
5202 2A8052    03470 TIMER   LD      HL,(TIME)       ;INCREMENT TIMER
5205 23        03480         INC     HL
5206 228052    03490         LD      (TIME),HL
5209 7D        03500         LD      A,L
520A FEC8      03510         CP      200             ;BRING ON BOMBER ?
520C CA4052    03520         JP      Z,BMRON         ;JUMP IF YES
               03530 ;
520F FE64      03540         CP      100             ;BRING ON POD ?
5211 CA5E52    03550         JP      Z,PODON         ;JUMP IF YES
               03560 ;
5214 7C        03570         LD      A,H
5215 FE02      03580         CP      2               ;TIME FOR CRUISER ?
5217 D8        03590         RET     C               ;RETURN IF NO
               03600 ;
5218 210000    03610         LD      HL,0            ;RESET TIMER
521B 228052    03620         LD      (TIME),HL
               03630 ;
               03640 ;       ***** BRING ON CRUISER *****
               03650 ;
521E 3ABC73    03660         LD      A,(WINDOW)      ;CALCULATE COLUMN
5221 C61E      03670         ADD     A,30
5223 57        03680         LD      D,A             ;COLUMN TO D
               03690 ;
5224 CDBC5C    03700         CALL    ROWRND          ;E GETS RANDOM ROW #
5227 DD21C674  03710         LD      IX,CRUTBL       ;IX => CRUISER TABLE
522B CD365B    03720         CALL    EMPTY           ;FIND EMPTY SPOT
522E D0        03730         RET     NC              ;RETURN IF NONE EMPTY
522F DD360214  03740         LD      (IX+2),20       ;DELAY BEFORE TURN
5233 DD360340  03750         LD      (IX+3),40H      ;SET WARP-IN FLAG
5237 DD360407  03760         LD      (IX+4),7        ;PHASE COUNTER
523B DD36051E  03770         LD      (IX+5),30       ;SHOT DELAY
523F C9        03780         RET
               03790 ;
               03800 ;       ***** BRING ON BOMBER *****
               03810 ;
5240 DD21FE74  03820 BMRON   LD      IX,BMRTBL
5244 CDB55C    03830         CALL    ROWCOL          ;GET RANDOM ROW & COLUMN
5247 CD365B    03840         CALL    EMPTY           ;FIND EMPTY SPOT
524A D0        03850         RET     NC              ;RETURN IF NONE
524B CDCA5C    03860         CALL    WRPCHK          ;CHECK FOR WARP-IN POS.
524E DD7201    03870         LD      (IX+1),D        ;NEW COLUMN
5251 DD36020F  03880         LD      (IX+2),15       ;DELAY
5255 DD360350  03890         LD      (IX+3),50H      ;FLAG WARP-INIT
5259 DD360407  03900         LD      (IX+4),7        ;PHASE COUNT
525D C9        03910         RET
               03920 ;
               03930 ;       ***** BRING ON POD *****
               03940 ;
525E DD21C975  03950 PODON   LD      IX,PODTBL
5262 CDB55C    03960         CALL    ROWCOL          ;GET RANDOM COLUMN #
5265 CD365B    03970         CALL    EMPTY           ;FIND A SPOT
5268 D0        03980         RET     NC              ;RETURN IF NONE
```

```
5269 CDCA5C    03990          CALL    WRPCHK          ;CHECK FOR WARP POS.
526C DD7201    04000          LD      (IX+1),D        ;NEW COLUMN
526F DD360208  04010          LD      (IX+2),8        ;MOVE DELAY
5273 DD360360  04020          LD      (IX+3),60H      ;SET FLAGS
5277 DD360407  04030          LD      (IX+4),7        ;PHASE COUNTER
527B DD360514  04040          LD      (IX+5),20       ;SHOT DELAY
527F C9        04050          RET
               04060 ;
5280 0000      04070 TIME     DW      00              ;TIMER STORAGE
               04080 ;
               04090 ;        ***** BRING ON LANDER *****
               04100 ;
5282 3ABD73    04110 LNDON    LD      A,(LNDTBL-1)    ;GET ACTIVE COUNT
5285 FE07      04120          CP      7               ;LESS THAN 7
5287 D0        04130          RET     NC              ;RETURN IF NO
               04140 ;
5288 2A9973    04150          LD      HL,(DLYCNT)     ;GET DELAY COUNTER
528B 0600      04160          LD      B,0
528D 3A9873    04170          LD      A,(FIRCNT)      ;GET SHOTS FIRED
5290 4F        04180          LD      C,A
5291 ED42      04190          SBC     HL,BC           ;COUNT DOWN
5293 229973    04200          LD      (DLYCNT),HL     ;SAVE RESULT
5296 D0        04210          RET     NC              ;RETURN IF NOT YET
               04220 ;
5297 210020    04230          LD      HL,2000H
529A 229973    04240          LD      (DLYCNT),HL     ;RESET COUNTER
               04250 ;
529D FD21BE73  04260          LD      IY,LNDTBL
52A1 CDB55C    04270          CALL    ROWCOL          ;GET RANDOM ROW & COLUMN
52A4 CD575B    04280          CALL    EMPTY2          ;FIND AN EMPTY SPOT
52A7 D0        04290          RET     NC              ;RETURN IF NONE
               04300 ;
52A8 CDCA5C    04310          CALL    WRPCHK          ;CHECK FOR WARP-IN LIMITS
52AB FD7201    04320          LD      (IY+1),D        ;COLUMN
52AE FD36020F  04330          LD      (IY+2),15       ;DELAY
52B2 FD360340  04340          LD      (IY+3),40H      ;FLAG WARP-ON
52B6 FD360407  04350          LD      (IY+4),7        ;PHASE COUNT
52BA FD360550  04360          LD      (IY+5),80       ;SHOT DELAY
52BE C9        04370          RET
               04380 ;
52BF 30        04390 BONSCR   DM      '00000'
52C4 00        04400          DB      0
52C5 42        04410 BONMSG   DM      'B O N U S  '
52D0 00        04420          DB      0
52D1 20        04430 BNMSG2   DM      '  x  100 = '
52DC 00        04440          DB      0
               04450 ;
52DD 3A0047    04460 STROBE   LD      A,(DSKFLG)      ;CAN WE WRITE TO DISK ?
52E0 B7        04470          OR      A
               04480 ;
               04490          IFEQ    TAPE,0          ;ASSEMBLE IF DISK VERSION
               04500          JR      NZ,STRB2        ;JUMP IF NO.
               04510          LD      A,(3880H)
               04520          OR      A               ;SAVE SCORES ?
               04530          JP      NZ,PUTSCR       ;JUMP IF YES
               04540          ENDIF
```

```
                04550 ;
52E1 3A7F38     04560 STRB2   LD      A,(387FH)
52E4 B7         04570         OR      A                ;START GAME ?
52E5 C8         04580         RET     Z                ;RETURN IF NO.
                04590 ;
52E6 31FA44     04600         LD      SP,MASK-5        ;RESET STACK
52E9 210000     04610         LD      HL,0
52EC E5         04620         PUSH    HL
52ED E5         04630         PUSH    HL
52EE AF         04640         XOR     A                ;DISABLE INTERRUPTS
52EF 32FF44     04650         LD      (MASK),A
52F2 C36947     04660         JP      START2           ;START GAME
                04670 ;
52F5            01850 *GET    EXPLODE/ASM              ;EXPLOSION ROUTINES
                00010 ;**************************************
                00020 ;**     EXPLODE ROUTINE  (ALIEN)     **
                00030 ;**************************************
                00040 ;
                00050 ;       *****   FIRST ENTRY INTO EXPLODE ROUTINE   *****
                00060 ;
52F5 212D53     00070 EXPLOD  LD      HL,EXPLD         ;CHANGE EXPLODE VECTOR
52F8 22E04F     00080         LD      (EXCNG0+1),HL
52FB DD213854   00090         LD      IX,EXTBL+1       ;EXPLOSION TABLE
52FF DD7E01     00100         LD      A,(IX+1)         ;MSB OF EXPLOSION ADDRESS
5302 21445D     00110         LD      HL,AD64          ;ADD 64 ROUTINE ADDRESS
5305 FE3D       00120         CP      3DH              ;EXPLODE UP ?
5307 3803       00130         JR      C,EXDN           ;JUMP IF NO
5309 214B5D     00140         LD      HL,SB64          ;SUBTRACT 64 ROUTINE
                00150 ;
530C DD7E02     00160 EXDN    LD      A,(IX+2)         ;GET EXPLOSION DIRECTION
530F B7         00170         OR      A                ;EXPLODE LEFT OR RIGHT ?
5310 3E23       00180         LD      A,23H            ;"INC HL" INSTRUCTION
5312 2802       00190         JR      Z,EXDN1          ;JUMP IF EXPLODE RIGHT
                00200 ;
5314 3E2B       00210         LD      A,2BH            ;"DEC HL" INSTRUCTION
                00220 ;
5316 226A53     00230 EXDN1   LD      (EXCNG2+1),HL    ;MODIFY CODE IN EXPLOSION
5319 229553     00240         LD      (EXCNG4+1),HL    ;ROUTINES.
531C 22B953     00250         LD      (EXCNG6+1),HL
531F 325753     00260         LD      (EXCNG1),A
5322 328453     00270         LD      (EXCNG3),A
5325 32A853     00280         LD      (EXCNG5),A
                00290 ;
5328 3E04       00300         LD      A,4              ;SETUP PHASE COUNTER
532A 32F355     00310         LD      (EXCNT),A
                00320 ;
                00330 ;       *****   SECOND ENTRY INTO EXPLODE ROUTINE   *****
                00340 ;
532D DD213854   00350 EXPLD   LD      IX,EXTBL+1       ;EXPLOSION TABLE POINTER
5331 DD6E00     00360         LD      L,(IX)           ;GET EXPLOSION ADDRESS
5334 DD6601     00370         LD      H,(IX+1)
5337 0603       00380         LD      B,3              ;ROW COUNT
5339 3AF355     00390         LD      A,(EXCNT)        ;GET PHASE OF EXPLOSION
533C 3D         00400         DEC     A                ;UPDATE
533D 32F355     00410         LD      (EXCNT),A
5340 CA9B53     00420         JP      Z,EXDON1         ;JUMP IF DONE
```

```
               00430 ;
5343 FE03      00440          CP      3              ;FIRST PHASE ?
5345 382A      00450          JR      C,EXCOM1       ;JUMP IF NO
               00460 ;
5347 117B55    00470          LD      DE,EXDAT1      ;EXPLOSION DATA - PHASE 1
534A C5        00480 EXLP1    PUSH    BC             ;SAVE ROW COUNT
534B E5        00490          PUSH    HL             ;SAVE SCREEN ADDRESS
534C 060A      00500          LD      B,10           ;COLUMN COUNT
534E 3E01      00510          LD      A,1
5350 EE03      00520 EXLP2    XOR     3
5352 F5        00530          PUSH    AF
5353 D3FF      00540          OUT     (255),A        ;MAKE NOISE
5355 1A        00550          LD      A,(DE)         ;GET CHARACTER FROM DATA
5356 77        00560          LD      (HL),A         ;PUT ON SCREEN
5357 23        00570 EXCNG1   INC     HL             ;UPDATE POINTERS
               00580                                 ;INSTRUCTION CHANGES
               00590                                 ;BETWEEN INC AND DEC HL.
5358 13        00600          INC     DE
5359 CD3D5C    00610          CALL    SIDCHK         ;CHECK FOR WRAP-AROUND
535C 2808      00620          JR      Z,SID1         ;JUMP IF WRAP-AROUND
535E F1        00630          POP     AF
535F EE03      00640          XOR     3
5361 D3FF      00650          OUT     (255),A        ;MAKE NOISE
5363 10EB      00660          DJNZ    EXLP2          ;LOOP FOR COLUMN COUNT
               00670 ;
5365 F5        00680          PUSH    AF
5366 F1        00690 SID1     POP     AF
5367 E1        00700          POP     HL             ;GET START COLUMN
5368 D5        00710          PUSH    DE             ;SAVE TABLE POS.
5369 CD4B5D    00720 EXCNG2   CALL    SB64           ;UPDATE ROW
               00730                                 ;INSTRUCTION CHANGES
               00740                                 ;BETWEEN CALL SB64 AND
               00750                                 ;CALL AD64.
536C D1        00760          POP     DE             ;GET TABLE POS.
536D C1        00770          POP     BC             ;GET ROW COUNT
536E 10DA      00780          DJNZ    EXLP1          ;LOOP FOR COLUMN COUNT
5370 C9        00790          RET
               00800 ;
               00810 ;        *****   PHASES 2 & 3  *****
               00820 ;
5371 C5        00830 EXCOM1   PUSH    BC             ;SAVE ROW COUNT
5372 E5        00840          PUSH    HL             ;SAVE SCREEN ADDRESS
5373 060A      00850          LD      B,10           ;CHARACTER COUNT
5375 3E01      00860          LD      A,1
5377 EE03      00870 EXLP6    XOR     3
5379 F5        00880          PUSH    AF
537A D3FF      00890          OUT     (255),A        ;MAKE NOISE
537C 7E        00900          LD      A,(HL)         ;GET CHAR. FROM SCREEN
537D FE81      00910          CP      129            ;GRAPHIC ?
537F 3803      00920          JR      C,EXCNG3       ;JUMP IF NO
5381 EE3F      00930          XOR     3FH            ;COMPLEMENT IT  (REVERSE)
5383 77        00940          LD      (HL),A         ;PUT BACK ON SCREEN
5384 23        00950 EXCNG3   INC     HL             ;THIS INSTRUCTION CHANGES
               00960                                 ;BETWEEN INC HL & DEC HL.
5385 CD3D5C    00970          CALL    SIDCHK         ;CHECK FOR WRAP AROUND
5388 2808      00980          JR      Z,SID3         ;JUMP IF WRAP
```

```
538A F1         00990          POP     AF
538B EE03       01000          XOR     3
538D D3FF       01010          OUT     (255),A            ;MAKE NOISE
538F 10E6       01020          DJNZ    EXLP6              ;LOOP FOR COLUMN COUNT
                01030 ;
5391 F5         01040          PUSH    AF
5392 F1         01050 SID3     POP     AF
5393 E1         01060          POP     HL                 ;GET START ROW
5394 CD4B5D     01070 EXCNG4   CALL    SB64               ;THIS INSTRUCTION CHANGES
                01080                                     ;BETWEEN CALL SB64 AND
                01090                                     ;CALL AD64.
5397 C1         01100          POP     BC                 ;GET ROW COUNT
5398 10D7       01110          DJNZ    EXCOM1             ;LOOP FOR COLUMN COUNT
539A C9         01120          RET
                01130 ;
                01140 ;        *****   LAST (ERASE) PHASE OF EXPLOSION   *****
                01150 ;
539B C5         01160 EXDON1   PUSH    BC                 ;SAVE ROW COUNT
539C E5         01170          PUSH    HL                 ;SAVE SCREEN ADDRESS
539D 060A       01180          LD      B,10               ;CHARACTER COUNT
539F 3E01       01190          LD      A,1
53A1 EE03       01200 EXPLP2   XOR     3
53A3 F5         01210          PUSH    AF
53A4 D3FF       01220          OUT     (255),A            ;MAKE NOISE
53A6 3680       01230          LD      (HL),128           ;ERASE IT
53A8 23         01240 EXCNG5   INC     HL                 ;THIS INSTRUCTION CHANGES
                01250                                     ;BETWEEN INC HL & DEC HL.
53A9 CD3D5C     01260          CALL    SIDCHK             ;CHECK FOR WRAP AROUND
53AC 2808       01270          JR      Z,SID5             ;JUMP IF WRAP
53AE F1         01280          POP     AF
53AF EE03       01290          XOR     3
53B1 D3FF       01300          OUT     (255),A
53B3 10EC       01310          DJNZ    EXPLP2             ;LOOP FOR COLUMN COUNT
                01320 ;
53B5 F5         01330          PUSH    AF
53B6 F1         01340 SID5     POP     AF
53B7 E1         01350          POP     HL                 ;GET ROW START
53B8 CD4B5D     01360 EXCNG6   CALL    SB64               ;THIS INSTRUCTION CHANGES
                01370                                     ;BETWEEN CALL SB64 AND
                01380                                     ;CALL AD64.
53BB C1         01390          POP     BC                 ;GET ROW COUNT
53BC 10DD       01400          DJNZ    EXDON1             ;LOOP FOR ROW COUNT
                01410 ;
53BE 21F552     01420          LD      HL,EXPLOD          ;CHANGE EXPLOSION VECTOR
53C1 22E04F     01430          LD      (EXCNG0+1),HL
                01440 ;
                01450 ;*********************************
                01460 ;**      MOVE DATA DOWN ON      **
                01470 ;**      EXPLOSION TABLE        **
                01480 ;*********************************
                01490 ;
53C4 DDE5       01500          PUSH    IX                 ;PUT TABLE ADDRESS
53C6 FDE1       01510          POP     IY                 ;IN IY AND MOVE ALL
53C8 DD7EFF     01520          LD      A,(IX-1)           ;ENTRIES DOWN ONE.
53CB 3D         01530          DEC     A                  ;UPDATE ACTIVE COUNT
53CC DD77FF     01540          LD      (IX-1),A
```

```
53CF 3C       01550        INC    A
53D0 110300   01560        LD     DE,3          ;ENTRY OFFSET
              01570 ;
53D3 3D       01580 EXPPLP DEC    A             ;DONE ?
53D4 2818     01590        JR     Z,EXPPOT      ;JUMP IF YES
              01600 ;
53D6 FD19     01610        ADD    IY,DE
53D8 FD4600   01620        LD     B,(IY)        ;MOVE DATA DOWN
53DB DD7000   01630        LD     (IX),B
53DE FD4601   01640        LD     B,(IY+1)
53E1 DD7001   01650        LD     (IX+1),B
53E4 FD4602   01660        LD     B,(IY+2)
53E7 DD7002   01670        LD     (IX+2),B
53EA DD19     01680        ADD    IX,DE
53EC 18E5     01690        JR     EXPPLP
              01700 ;
53EE 3A3754   01710 EXPPOT LD     A,(EXTBL)     ;GET ACTIVE ENTRY COUNT
53F1 B7       01720        OR     A             ;ANY LEFT ON TABLE ?
53F2 C0       01730        RET    NZ            ;RETURN IF YES
              01740 ;
53F3 3AFF44   01750        LD     A,(MASK)
53F6 CB87     01760        RES    0,A           ;CANCEL EXPLOSIONS
53F8 32FF44   01770        LD     (MASK),A
53FB C9       01780        RET
              01790 ;
              01800 ;******************************************
              01810 ;**     PUT EXPLOSION DATA ON TABLE      **
              01820 ;******************************************
              01830 ;
53FC F5       01840 EXPSH  PUSH   AF
53FD DD213854 01850        LD     IX,EXTBL+1    ;EXPLOSION TABLE
5401 DD7EFF   01860        LD     A,(IX-1)      ;GET ACTIVE COUNT
5404 DDBEFE   01870        CP     (IX-2)        ;SAME AS MAX ALLOWED ?
5407 282B     01880        JR     Z,EPHEXT      ;JUMP IF YES
5409 3AA173   01890        LD     A,(SHROW)     ;GET THE ROW #
540C 32D05B   01900        LD     (ROWPAS),A
540F DD7EFF   01910        LD     A,(IX-1)      ;GET ACTIVE COUNT
5412 3C       01920        INC    A             ;ONE MORE EXPLOSION TO DO.
5413 DD77FF   01930        LD     (IX-1),A
5416 110300   01940        LD     DE,3          ;ENTRY OFFSET
              01950 ;
              01960 ;      *****  FIND EMPTY SPOT ON EXPLODE TABLE   *****
              01970 ;
5419 3D       01980 EXPHLP DEC    A             ;DONE ?
541A 2804     01990        JR     Z,EXPHOT      ;JUMP IF YES
541C DD19     02000        ADD    IX,DE
541E 18F9     02010        JR     EXPHLP
              02020 ;
5420 DD7500   02030 EXPHOT LD     (IX),L        ;PUT HIT ADDRESS
5423 DD7401   02040        LD     (IX+1),H      ;ON TABLE
5426 3AAA73   02050        LD     A,(DIRF2)     ;PUT THE DIRECTION
5429 DD7702   02060        LD     (IX+2),A      ;ON THE TABLE
542C 3AFF44   02070        LD     A,(MASK)
542F CBC7     02080        SET    0,A           ;SET EXPLODE FLAG
5431 32FF44   02090        LD     (MASK),A
5434 F1       02100 EPHEXT POP    AF
```

```
5435 C9       02110         RET
              02120 ;
              02130 ;********************************
              02140 ;**      EXPLOSION TABLE      **
              02150 ;********************************
              02160 ;
5436 14       02170         DB      20      ;MAX ALLOWED ON TABLE
5437 00       02180 EXTBL   DB      0       ;# OF ACTIVE ENTRIES ON TABLE
              02190 ;
5438 00       02200         DB      0       ;LSB OF EXPLODE ADDRESS
5439 00       02210         DB      0       ;MSB OF EXPLODE ADDRESS
543A 00       02220         DB      0       ;DIRECTION OF EXPLOSION
              02230 ;
543B 00       02240         DB      0,0,0   ;2
543E 00       02250         DB      0,0,0   ;3
5441 00       02260         DB      0,0,0   ;4
5444 00       02270         DB      0,0,0   ;5
5447 00       02280         DB      0,0,0   ;6
544A 00       02290         DB      0,0,0   ;7
544D 00       02300         DB      0,0,0   ;8
5450 00       02310         DB      0,0,0   ;9
5453 00       02320         DB      0,0,0   ;10
5456 00       02330         DB      0,0,0   ;11
5459 00       02340         DB      0,0,0   ;12
545C 00       02350         DB      0,0,0   ;13
545F 00       02360         DB      0,0,0   ;14
5462 00       02370         DB      0,0,0   ;15
5465 00       02380         DB      0,0,0   ;16
5468 00       02390         DB      0,0,0   ;17
546B 00       02400         DB      0,0,0   ;18
546E 00       02410         DB      0,0,0   ;19
5471 00       02420         DB      0,0,0   ;20
5474 00       02430         DB      0,0,0   ;DUMMY
              02440 ;
              02450 ;***************************
              02460 ;** SHIP EXPLODE ROUTINE **
              02470 ;***************************
              02480 ;
5477 AF       02490 SHIP    XOR     A                ;CANCEL ALL PENDING
5478 32FF44   02500         LD      (MASK),A         ;INTERRUPT PROCESSING.
547B CDCB50   02510         CALL    SCORE            ;UPDATE SCORE DISPLAY
547E 2AA773   02520         LD      HL,(SHPOS)       ;GET SHIP POSITION.
5481 E5       02530         PUSH    HL               ;SAVE START ADDRESS
5482 112B55   02540         LD      DE,SHDAT1        ;FIRST EXPLODE DATA
5485 010402   02550         LD      BC,0204H         ;B=ROW C=COLUMN COUNT
5488 CDCB54   02560         CALL    SHDRW            ;EXPLODE & COMPLEMENT
              02570 ;
548B E1       02580         POP     HL               ;GET START ADDRESS
548C CD4B5D   02590         CALL    SB64             ;UP ONE ROW
548F 2B       02600         DEC     HL               ;BACK TWO COLUMNS
5490 2B       02610         DEC     HL
5491 E5       02620         PUSH    HL               ;SAVE IT
5492 113355   02630         LD      DE,SHDAT2        ;SECOND EXPLODE DATA
5495 010804   02640         LD      BC,0408H         ;ROW/COL COUNT
5498 CDCB54   02650         CALL    SHDRW            ;DRAW & COMPLEMENT
              02660 ;
```

```
549B E1         02670          POP     HL                ;GET START ADDRESS
549C 2B         02680          DEC     HL                ;BACK UP ONE COLUMN
549D E5         02690          PUSH    HL                ;SAVE IT
549E 010A04     02700          LD      BC,040AH          ;ROW/COL COUNT
54A1 C5         02710          PUSH    BC                ;SAVE IT
54A2 115355     02720          LD      DE,SHDAT3         ;THIRD EXPLODE
54A5 CDCB54     02730          CALL    SHDRW             ;DRAW AND COMPLEMENT
                02740 ;
54A8 C1         02750          POP     BC                ;GET ROW/COL COUNT
54A9 E1         02760          POP     HL                ;GET START ADDRESS
54AA E5         02770          PUSH    HL                ;SAVE IT
54AB C5         02780          PUSH    BC                ;SAVE ROW/COL COUNT
54AC CDEC54     02790          CALL    SCOP1             ;COMPLEMENT AGAIN
                02800 ;
54AF CD0A55     02810          CALL    SO5ND             ;MAKE A NOISE
54B2 C1         02820          POP     BC                ;GET ROW/COL COUNT
54B3 E1         02830          POP     HL                ;GET START ADDRESS
54B4 C5         02840 SCRLP    PUSH    BC                ;SAVE ROW COUNT
54B5 E5         02850          PUSH    HL                ;SAVE ADDRES
54B6 41         02860          LD      B,C               ;GET ROW COUNT
                02870 ;
54B7 3680       02880 SCRLP2   LD      (HL),128          ;CLEAR CHARACTER
54B9 23         02890          INC     HL                ;BUMP POINTER
54BA 10FB       02900          DJNZ    SCRLP2            ;LOOP FOR COL. COUNT
                02910 ;
54BC E1         02920          POP     HL                ;GET ROW START
54BD CD445D     02930          CALL    AD64              ;DROP ONE ROW
54C0 C1         02940          POP     BC                ;GET ROW COUNT
54C1 7C         02950          LD      A,H
54C2 FE40       02960          CP      40H               ;OFF THE SCREEN ?
54C4 3002       02970          JR      NC,DRSKP3         ;JUMP IF YES
54C6 10EC       02980          DJNZ    SCRLP             ;LOOP FOR ROW COUNT
                02990 ;
54C8 C3B14B     03000 DRSKP3   JP      WRMSRT
                03010 ;
54CB C5         03020 SHDRW    PUSH    BC                ;SAVE ROW/COL
54CC E5         03030          PUSH    HL                ;SAVE SCREEN POS.
                03040 ;
54CD C5         03050 SHDRW1   PUSH    BC                ;SAVE ROW COUNT
54CE E5         03060          PUSH    HL                ;SAVE SCREEN POS.
54CF 41         03070          LD      B,C               ;GET COL. COUNT
                03080 ;
54D0 1A         03090 SHDRW2   LD      A,(DE)            ;GET DATA
54D1 77         03100          LD      (HL),A            ;PUT ON SCREEN
54D2 13         03110          INC     DE                ;BUMP POINTERS
54D3 23         03120          INC     HL
54D4 10FA       03130          DJNZ    SHDRW2            ;LOOP FOR COL. COUNT
                03140 ;
54D6 E1         03150          POP     HL                ;GET ROW START
54D7 D5         03160          PUSH    DE                ;SAVE DATA POS.
54D8 CD445D     03170          CALL    AD64              ;DROP ONE ROW
54DB D1         03180          POP     DE                ;GET DATA POS.
54DC C1         03190          POP     BC                ;GET ROW COUNT
54DD 7C         03200          LD      A,H
54DE FE40       03210          CP      40H               ;OFF THE SCREEN ?
54E0 3002       03220          JR      NC,DRSKP1         ;JUMP IF YES
```

```
54E2 10E9      03230          DJNZ    SHDRW1          ;LOOP FOR ROW COUNT
54E4 CD0A55    03240 DRSKP1   CALL    SO5ND           ;MAKE A NOISE
54E7 CD1E55    03250          CALL    DELAY
               03260 ;
54EA E1        03270          POP     HL              ;GET FIRST SCREEN POS.
54EB C1        03280          POP     BC              ;GET FIRST ROW/COL COUNT
               03290 ;
54EC C5        03300 SCOP1    PUSH    BC              ;SAVE ROW COUNT
54ED E5        03310          PUSH    HL              ;SAVE SCREEN POS.
54EE 41        03320          LD      B,C             ;GET COL. COUNT
               03330 ;
54EF 7E        03340 SCOP2    LD      A,(HL)          ;GET SCREEN DATA
54F0 FE81      03350          CP      129             ;NON GRAPHIC ?
54F2 3803      03360          JR      C,SCOSKP        ;JUMP IF YES
54F4 EE3F      03370          XOR     3FH             ;COMPLEMENT
54F6 77        03380          LD      (HL),A          ;PUT BACK ON SCREEN
54F7 23        03390 SCOSKP   INC     HL              ;BUMP SCREEN POS.
54F8 10F5      03400          DJNZ    SCOP2           ;LOOP FOR COL. COUNT
               03410 ;
54FA E1        03420          POP     HL              ;GET ROW START
54FB CD445D    03430          CALL    AD64            ;DROP ONE ROW
54FE C1        03440          POP     BC              ;GET ROW COUNT
54FF 7C        03450          LD      A,H
5500 FE40      03460          CP      40H             ;OFF THE SCREEN ?
5502 3002      03470          JR      NC,DRSKP2       ;JUMP IF YES
5504 10E6      03480          DJNZ    SCOP1           ;LOOP TIL DONE
5506 CD0A55    03490 DRSKP2   CALL    SO5ND           ;MAKE A NOISE
5509 C9        03500          RET
               03510 ;
550A C5        03520 SO5ND    PUSH    BC
550B 06C8      03530          LD      B,200
550D 3E01      03540          LD      A,1
550F EE03      03550 SO5LP    XOR     3
5511 D3FF      03560          OUT     (255),A
5513 C5        03570          PUSH    BC
5514 10FE      03580          DJNZ    $
5516 C1        03590          POP     BC
5517 05        03600          DEC     B
5518 05        03610          DEC     B
5519 05        03620          DEC     B
551A 10F3      03630          DJNZ    SO5LP
551C C1        03640          POP     BC
551D C9        03650          RET
               03660 ;
551E C5        03670 DELAY    PUSH    BC              ;PAUSE ROUTINE
551F F5        03680          PUSH    AF
5520 015003    03690          LD      BC,350H
5523 0B        03700 DLYLP    DEC     BC
5524 78        03710          LD      A,B
5525 B1        03720          OR      C
5526 20FB      03730          JR      NZ,DLYLP
5528 F1        03740          POP     AF
5529 C1        03750          POP     BC
552A C9        03760          RET
               03770 ;
               03780 ;********* SHIP EXPLODE DATA *********
```

```
                03790 ;
552B 98         03800 SHDAT1  DB      152,161,133,134
552F A6         03810         DB      166,136,177,137
                03820 ;
5533 80         03830 SHDAT2  DB      128,144,144,146,164,128,144,128
553B 80         03840         DB      128,132,132,132,132,154,160,128
5543 81         03850         DB      129,137,132,137,134,152,153,128
554B 80         03860         DB      128,128,130,161,153,128,132,128
                03870 ;
5553 A0         03880 SHDAT3  DB      160,132,134,152,146,161,132,137,168,160
555D 99         03890         DB      153,130,132,128,128,128,130,130,144,166
5567 A4         03900         DB      164,134,132,144,128,128,128,132,132,173
5571 81         03910         DB      129,132,161,164,161,152,160,129,146,136
                03920 ;
                03930 ;********** ALIEN EXPLODE DATA **********
                03940 ;
557B 92         03950 EXDAT1  DB      146,136,160,162,136,130,144,132,132,128
5585 80         03960         DB      128,160,160,132,145,132,161,160,136,136
558F 80         03970         DB      128,128,128,128,128,128,128,144,152,146
                03980 ;
                03990 ;********** WARP-IN DATA **********
                04000 ;
5599 80         04010 WARP1   DB      128,128,128,128,128,128,128,128,128,128,166,166,166,166,128
55A8 80         04020 WARP2   DB      128,128,128,128,128,128,128,166,166,166,166,166,166,128,128
55B7 80         04030 WARP3   DB      128,128,128,166,166,166,166,166,166,166,166,128,128,128,128
55C6 80         04040 WARP4   DB      128,128,166,166,166,166,166,166,128,128,128,128,128,128,128
55D5 80         04050 WARP5   DB      128,128,166,166,166,166,128,128,128,128,128,128,128,128,128
55E4 80         04060 WARP6   DB      128,166,140,153,128,128,128,128,128,128,128,128,128,128,128
                04070 ;
55F3 04         04080 EXCNT   DB      4                 ;EXPLOSION PHASE COUNTER
                04090 ;
55F4            01860 *GET    POLL/ASM                  ;KEYBOARD & SHIP CONTROL
                00010 ;**************************************
                00020 ;**     POLL KEYBOARD ROUTINE       **
                00030 ;**************************************
                00040 ;
55F4 3A1038     00050 POLL    LD      A,(3810H)         ;KEYS 1,2 & 3
55F7 CB4F       00060         BIT     1,A               ;CHANGE DIRECTION ?
55F9 200F       00070         JR      NZ,C1DO
55FB 3AAA73     00080         LD      A,(DIRF2)         ;CURRENT SHIP DIRECTION
55FE 3C         00090         INC     A                 ;1= RIGHT : 2= LEFT
55FF 47         00100         LD      B,A
5600 DB00       00110         IN      A,(0)             ;GET JOYSTICK
5602 2F         00120         CPL
5603 CB3F       00130         SRL     A
5605 CB3F       00140         SRL     A
5607 E603       00150         AND     3                 ;MASK OUT UNWANTED BITS
5609 A0         00160         AND     B                 ;CHANGE DIRECTION ?
560A F5         00170 C1DO    PUSH    AF                ;SAVE FLAGS
560B C45357     00180         CALL    NZ,CNGDIR         ;CALL IF YES
560E F1         00190         POP     AF
560F 2004       00200         JR      NZ,CNGSKP         ;JUMP IF YES
                00210 ;
5611 AF         00220         XOR     A                 ;CANCEL CHANGE DIRECTION
5612 32B473     00230         LD      (CNGFLG),A        ;FLAG
                00240 ;
```

```
5615 21AB73    00250 CNGSKP  LD      HL,UPDNF         ;HL -> UP/DOWN MOVE DELAY
5618 4E        00260         LD      C,(HL)           ;GET UP/DOWN COUNTER
5619 0D        00270         DEC     C                ;CAN WE MOVE YET ?
561A 71        00280         LD      (HL),C
561B 2023      00290         JR      NZ,UDSKP         ;JUMP IF NOT YET
561D 0E03      00300         LD      C,3
561F 71        00310         LD      (HL),C
               00320 ;
5620 3A1038    00330         LD      A,(3810H)        ;UP/DOWN - BIT 2/3
5623 E60C      00340         AND     0CH              ;MASK OUT UNWANTED BITS
5625 47        00350         LD      B,A
5626 DB00      00360         IN      A,(0)            ;GET JOYSTICK
5628 2F        00370         CPL
5629 E603      00380         AND     3                ;MASK OUT UNWANTED BITS
562B B0        00390         OR      B                ;COMBINE KEYBORD/JOYSTICK
562C CB57      00400         BIT     2,A              ;MOVE UP ?
562E 2002      00410         JR      NZ,U1DO          ;JUMP IF YES
5630 CB47      00420         BIT     0,A              ;MOVE UP ?
5632 F5        00430 U1DO    PUSH    AF
5633 C44658    00440         CALL    NZ,UP            ;CALL IF YES
5636 F1        00450         POP     AF
               00460 ;
5637 CB5F      00470         BIT     3,A              ;MOVE DOWN ?
5639 2002      00480         JR      NZ,DWNCNG        ;JUMP IF YES
563B CB4F      00490         BIT     1,A              ;MOVE DOWN ?
563D C47258    00500 DWNCNG  CALL    NZ,DOWN          ;CALL IF YES
               00510                                  ;THIS CALL CHANGES WHEN
               00520                                  ;THE SHIP IS CARRYING A MAN
               00530 ;
5640 CD885A    00540 UDSKP   CALL    SHTCLR           ;CLEAR ANY SHOTS FROM
               00550                                  ;THE SCREEN.
               00560 ;
5643 3A2038    00570         LD      A,(3820H)        ;8/9  BIT 0/1 THRUST/FIRE
5646 E603      00580         AND     3                ;MASK OUT UNWANTED BITS
5648 47        00590         LD      B,A
5649 DB00      00600         IN      A,(0)            ;GET JOYSTICK
564B 2F        00610         CPL
564C E61C      00620         AND     1CH              ;BITS 4,3,2  FIRE/MOVE
564E B0        00630         OR      B
               00640 ;
564F CB4F      00650         BIT     1,A              ;FIRE ?
5651 2002      00660         JR      NZ,F1DO          ;JUMP IF YES
5653 CB67      00670         BIT     4,A              ;FIRE ?
5655 F5        00680 F1DO    PUSH    AF
5656 C43859    00690         CALL    NZ,FIRE          ;CALL IF YES
5659 F1        00700         POP     AF
565A 2006      00710         JR      NZ,MVSKP         ;JUMP IF YES
565C F5        00720         PUSH    AF
565D AF        00730         XOR     A                ;CLEAR SHOT FLAG
565E 32D15B    00740         LD      (SHTF1),A
5661 F1        00750         POP     AF
               00760 ;
5662 CB47      00770 MVSKP   BIT     0,A              ;THRUST ?
5664 2002      00780         JR      NZ,T1DO          ;JUMP IF YES
5666 E60C      00790         AND     0CH              ;THRUST ?
5668 F5        00800 T1DO    PUSH    AF
```

```
5669 C49D56     00810 MVCNG   CALL    NZ,SPEEDU         ;CALL IF YES
                00820                                   ;THIS CALL CHANGES FOR
                00830                                   ;SPEED UP - SLOW DOWN
                00840                                   ;FULL SPEED - CHANGE DIRECTION
566C F1         00850         POP     AF
566D C0         00860         RET     NZ                ;RETURN IF YES
                00870 ;
566E 3AB673     00880         LD      A,(MVFLG3)        ;MOVING ?
5671 B7         00890         OR      A
5672 C8         00900         RET     Z                 ;RETURN IF NO
                00910 ;
5673 CD2157     00920 MVCNG2  CALL    STOP              ;CALL IF YES
5676 C9         00930         RET
                00940 ;
                00950 ;************************************
                00960 ;**        MOTION ROUTINES        **
                00970 ;************************************
                00980 ;
                00990 ;       *****   FULL SPEED   *****
                01000 ;
5677 3AA973     01010 MOVE    LD      A,(DIRFLG)        ;GET DIRECTION OF MOVE
567A ED5BBA73   01020         LD      DE,(DPOINT)       ;MOUNTAIN DATA POINTER
567E B7         01030         OR      A
567F 3ABC73     01040         LD      A,(WINDOW)        ;SCREEN POSITION ON
                01050                                   ;PLAY GRID.
5682 C29456     01060         JP      NZ,MINUS          ;JUMP IF MOVE RIGHT
                01070 ;
                01080 ;       ** MOVE MOUNTAINS LEFT **
                01090 ;
5685 3C         01100         INC     A                 ;MOVE SCREEN POSITION
5686 F5         01110         PUSH    AF
5687 CD0C57     01120         CALL    SCOL1             ;UPDATE SHIP COLUMN
568A F1         01130         POP     AF
568B 1C         01140         INC     E                 ;UPDATE MOUNTAIN DATA POINTER
568C 32BC73     01150 PLSKP   LD      (WINDOW),A
568F ED53BA73   01160         LD      (DPOINT),DE
5693 C9         01170         RET
                01180 ;
                01190 ;       ** MOVE MOUNTAINS RIGHT **
                01200 ;
5694 3D         01210 MINUS   DEC     A                 ;UPDATE SCREEN POSITION
5695 F5         01220         PUSH    AF
5696 CD1457     01230         CALL    SCOL2             ;UPDATE SHIP COLUMN
5699 F1         01240         POP     AF
569A 1D         01250         DEC     E                 ;UPDATE MOUNTAIN DATA POINTER
569B 18EF       01260         JR      PLSKP
                01270 ;
                01280 ;       *****   ACCELERATION ROUTINE    *****
                01290 ;
569D 21B956     01300 SPEEDU  LD      HL,SPDUP          ;CHANGE MOVE VECTOR
56A0 226A56     01310         LD      (MVCNG+1),HL
56A3 212157     01320         LD      HL,STOP           ;CHANGE STOP VECTOR
56A6 227456     01330         LD      (MVCNG2+1),HL
56A9 3E01       01340         LD      A,1
56AB 32B673     01350         LD      (MVFLG3),A        ;SET MOTION FLAG
56AE 3E02       01360         LD      A,2
```

```
56B0 32B773   01370          LD     (MVFLG4),A       ;SHIP MOVE DELAY
56B3 3AAA73   01380          LD     A,(DIRF2)        ;GET SHIP DIRECTION
56B6 32A973   01390          LD     (DIRFLG),A       ;MOUNTAIN DIRECTION
              01400 ;
56B9 ED4BB873 01410 SPDUP    LD     BC,(SPDCNT)      ;GET DELAY COUNT
56BD 0D       01420          DEC    C                ;MOVE ?
56BE 200C     01430          JR     NZ,SPEXT1        ;JUMP IF NO
56C0 05       01440          DEC    B                ;SHORTER DELAY NEXT TIME
56C1 2801     01450          JR     Z,SPDUP2
56C3 05       01460          DEC    B
56C4 48       01470 SPDUP2   LD     C,B
56C5 280A     01480          JR     Z,SPEXT2         ;JUMP IF NO MORE DELAY
56C7 C5       01490          PUSH   BC               ;SAVE COUNT
56C8 CD7756   01500          CALL   MOVE             ;DO IT
56CB C1       01510          POP    BC               ;GET COUNT
              01520 ;
56CC ED43B873 01530 SPEXT1   LD     (SPDCNT),BC      ;SAVE FOR NEXT TIME
56D0 C9       01540          RET
              01550 ;
56D1 21D956   01560 SPEXT2   LD     HL,MOVE3         ;CHANGE MOVE VECTOR
56D4 226A56   01570          LD     (MVCNG+1),HL
56D7 18F3     01580          JR     SPEXT1
              01590 ;
              01600 ;        *****  SLIDE BACK ROUTINE   *****
              01610 ;
56D9 CD7756   01620 MOVE3    CALL   MOVE             ;DO IT
56DC 3AB773   01630          LD     A,(MVFLG4)       ;MOVE SHIP FLAG
56DF 3D       01640          DEC    A
56E0 32B773   01650          LD     (MVFLG4),A
56E3 C0       01660          RET    NZ               ;RETURN IF NOT YET
56E4 3E03     01670          LD     A,3
56E6 32B773   01680          LD     (MVFLG4),A
              01690 ;
56E9 2AA773   01700          LD     HL,(SHPOS)       ;GET SHIP POSITION
56EC 3AA973   01710          LD     A,(DIRFLG)       ;GET DIRECTION
56EF B7       01720          OR     A
56F0 280D     01730          JR     Z,PL1            ;JUMP IF MOVE LEFT
              01740 ;
56F2 7D       01750          LD     A,L              ;MOVE SHIP RIGHT
56F3 E63F     01760          AND    3FH              ;ON SCREEN
56F5 FE2D     01770          CP     2DH              ;ALL THE WAY OVER ?
56F7 2821     01780          JR     Z,MVCMD1         ;JUMP IF YES
56F9 2C       01790          INC    L                ;UPDATE SHIP POSITION
56FA 22A773   01800          LD     (SHPOS),HL       ;SAVE NEW POSITION
56FD 180D     01810          JR     SCOL1            ;GET NEW SHIP COLUMN
              01820 ;
56FF 7D       01830 PL1      LD     A,L              ;MOVE SHIP LEFT ON SCREEN
5700 E63F     01840          AND    3FH
5702 FE0F     01850          CP     0FH              ;ALL THE WAY BACK ?
5704 2814     01860          JR     Z,MVCMD1         ;JUMP IF YES
5706 2D       01870          DEC    L                ;UPDATE SHIP POSITION
5707 22A773   01880          LD     (SHPOS),HL       ;SAVE NEW POSITION
570A 1808     01890          JR     SCOL2            ;UPDATE SHIP COLUMN
              01900 ;
570C 3AA273   01910 SCOL1    LD     A,(SHCOL)        ;GET CURRENT SHIP COLUMN
570F 3C       01920          INC    A                ;UPDATE
```

```
5710 32A273    01930 PLSKP5  LD      (SHCOL),A       ;SAVE
5713 C9        01940         RET
               01950 ;
5714 3AA273    01960 SCOL2   LD      A,(SHCOL)       ;GET CURRENT SHIP COLUMN
5717 3D        01970         DEC     A               ;UPDATE
5718 18F6      01980         JR      PLSKP5
               01990 ;
               02000 ;       *****   CHANGE VECTOR TO FULL SPEED   *****
               02010 ;
571A 217756    02020 MVCMD1  LD      HL,MOVE         ;CHANGE MOVE VECTOR
571D 226A56    02030         LD      (MVCNG+1),HL
5720 C9        02040         RET
               02050 ;
               02060 ;       *****   CHANGE TO SLOW DOWN ROUTINE   *****
               02070 ;
5721 219D56    02080 STOP    LD      HL,SPEEDU       ;CHANGE MOVE VECTOR
5724 226A56    02090         LD      (MVCNG+1),HL
5727 212D57    02100         LD      HL,SLODWN       ;CHANGE STOP VECTOR
572A 227456    02110         LD      (MVCNG2+1),HL
               02120 ;
572D ED4BB873  02130 SLODWN  LD      BC,(SPDCNT)     ;GET DELAY COUNT
5731 0D        02140         DEC     C               ;MOVE ?
5732 FA3757    02150         JP      M,SLD1          ;JUMP IF NO
5735 2095      02160         JR      NZ,SPEXT1       ;JUMP IF NO
5737 04        02170 SLD1    INC     B               ;LONGER DELAY NEXT TIME
5738 48        02180         LD      C,B
5739 78        02190         LD      A,B
573A FE0A      02200         CP      10              ;MAX DELAY ?
573C 3008      02210         JR      NC,SLEXT2       ;JUMP IF YES
573E C5        02220         PUSH    BC              ;SAVE COUNT
573F CD7756    02230         CALL    MOVE            ;DO IT
5742 C1        02240         POP     BC
5743 C3CC56    02250         JP      SPEXT1
               02260 ;
5746 212157    02270 SLEXT2  LD      HL,STOP         ;CHANGE STOP VECTOR
5749 227456    02280         LD      (MVCNG2+1),HL
574C AF        02290         XOR     A
574D 32B673    02300         LD      (MVFLG3),A      ;CLEAR MOTION FLAG
5750 C3CC56    02310         JP      SPEXT1
               02320 ;
               02330 ;       *****   CHANGE DIRECTION ROUTINES   *****
               02340 ;
5753 3AB473    02350 CNGDIR  LD      A,(CNGFLG)      ;KEY MUST BE RELEASED
5756 B7        02360         OR      A               ;FOR ONE STROBE TO
5757 C0        02370         RET     NZ              ;CHANGE AGAIN
5758 3C        02380         INC     A
5759 32B473    02390         LD      (CNGFLG),A      ;SET CHANGE FLAG
575C 3AAA73    02400         LD      A,(DIRF2)       ;GET SHIP DIRECTION
575F EE01      02410         XOR     1               ;CHANGE IT
5761 32AA73    02420         LD      (DIRF2),A
5764 3AB673    02430         LD      A,(MVFLG3)      ;MOTION FLAG
5767 B7        02440         OR      A
5768 2007      02450         JR      NZ,CNGD2        ;JUMP IF MOVING
               02460 ;
576A 3AAA73    02470         LD      A,(DIRF2)       ;MAKE MOTION DIRECTION
576D 32A973    02480         LD      (DIRFLG),A      ;SAME AS SHIP DIRECTION
```

```
5770 C9       02490          RET
              02500 ;
5771 217857   02510 CNGD2    LD      HL,CNGD3        ;CHANGE MOVE VECTOR
5774 226A56   02520          LD      (MVCNG+1),HL    ;TO THIS ROUTINE
5777 C9       02530          RET
              02540 ;
5778 CD2D57   02550 CNGD3    CALL    SLODWN          ;DECELERATE
577B 3AB673   02560          LD      A,(MVFLG3)
577E B7       02570          OR      A               ;STOPPED ?
577F C0       02580          RET     NZ              ;RETURN IF NO
              02590 ;
5780 219D56   02600          LD      HL,SPEEDU       ;CHANGE MOVE VECTOR
5783 226A56   02610          LD      (MVCNG+1),HL    ;TO SPEED UP ROUTINE
5786 C9       02620          RET
              02630 ;
              02640 ;        *****   SHIP ERASE (NOT CARRYING MAN)    *****
              02650 ;
5787 EB       02660 ERASE    EX      DE,HL           ;HL => OLD POS.
5788 0604     02670          LD      B,4             ;OF SHIP
578A 3680     02680 ERLP     LD      (HL),128        ;ERASE IT
578C 23       02690          INC     HL
578D 10FB     02700          DJNZ    ERLP
578F C9       02710          RET
              02720 ;
              02730 ;        *****   SHIP ERASE (CARRYING MAN)    *****
              02740 ;
5790 EB       02750 ERASE2   EX      DE,HL           ;HL => OLD POS.
5791 D5       02760          PUSH    DE
5792 CD445D   02770          CALL    AD64
5795 0604     02780          LD      B,4
5797 3E80     02790          LD      A,128
5799 77       02800 ERLP2    LD      (HL),A          ;ERASE MAN
579A 12       02810          LD      (DE),A          ;ERASE SHIP
579B 13       02820          INC     DE
579C 23       02830          INC     HL
579D 10FA     02840          DJNZ    ERLP2
579F D1       02850          POP     DE
57A0 C9       02860          RET
              02870 ;
              02880 ;**********************************
              02890 ;**     DRAW SHIP ROUTINES       **
              02900 ;**********************************
              02910 ;
57A1 CD0358   02920 UPDATE   CALL    UPDT1           ;THIS VECTOR CHANGES WITH
              02930                                  ;THE STATUS OF THE SHIP
57A4 C9       02940          RET
              02950 ;
              02960 ;        *****   DRAW SHIP (LOOKING FOR MAN)    *****
              02970 ;
57A5 CD0358   02980 UPDT3    CALL    UPDT1           ;DRAW SHIP
57A8 D5       02990          PUSH    DE
57A9 EB       03000          EX      DE,HL           ;HL => SCREEN POS.
57AA CD445D   03010          CALL    AD64
57AD 0602     03020          LD      B,2
57AF 23       03030 MNFLP    INC     HL              ;COMPENSATE FOR SPACE
57B0 7E       03040          LD      A,(HL)
```

```
57B1 FEFD     03050         CP      253             ;MAN ?
57B3 2804     03060         JR      Z,MNFND         ;JUMP IF YES
57B5 10F8     03070         DJNZ    MNFLP
57B7 1848     03080         JR      UPDEXO
              03090 ;
57B9 FD210274 03100 MNFND   LD      IY,MANTBL       ;MAN TABLE
57BD 3680     03110         LD      (HL),128
57BF CD065B   03120         CALL    SPSER1          ;SEARCH ON COLUMN
57C2 FDCB0346 03130         BIT     0,(IY+3)        ;IN AIR ?
57C6 2839     03140         JR      Z,UPDEXO        ;JUMP IF NO
57C8 FD360342 03150         LD      (IY+3),42H      ;SET LANDER CARRY
57CC 3A0274   03160         LD      A,(MANTBL)
57CF 3C       03170         INC     A
57D0 90       03180         SUB     B               ;GET ENTRY #
57D1 32A373   03190         LD      (CRYENT),A      ;SAVE IT.
57D4 21AE58   03200         LD      HL,DOWN3        ;CHANGE MOVE DOWN VECTOR
57D7 223E56   03210         LD      (DWNCNG+1),HL
57DA 219057   03220         LD      HL,ERASE2       ;CHANGE SHIP ERASE VECTOR
57DD 226B58   03230         LD      (UPOT1+1),HL
57E0 21F157   03240         LD      HL,UPDT2        ;CHANGE SHIP DRAW VECTOR
57E3 22A257   03250         LD      (UPDATE+1),HL
57E6 01F401   03260         LD      BC,500          ;500 POINTS FOR CATCHING MAN
57E9 CDED4F   03270         CALL    HIT             ;ADD TO PLAYERS SCORE
57EC CD934F   03280         CALL    BINK
57EF 1810     03290         JR      UPDEXO
              03300 ;
              03310 ;       *****   DRAW SHIP (CARRYING MAN)   *****
              03320 ;
57F1 CD0358   03330 UPDT2   CALL    UPDT1           ;DRAW SHIP
57F4 D5       03340         PUSH    DE              ;SAVE SHIP POSITION
57F5 EB       03350         EX      DE,HL
57F6 CD445D   03360         CALL    AD64
57F9 3680     03370         LD      (HL),128        ;DRAW MAN
57FB 23       03380         INC     HL
57FC 36FD     03390         LD      (HL),253
57FE 23       03400         INC     HL
57FF 3680     03410         LD      (HL),128
              03420 ;
5801 D1       03430 UPDEXO  POP     DE
5802 C9       03440         RET
              03450 ;
              03460 ;       *****   DRAW SHIP   *****
              03470 ;
5803 3AAA73   03480 UPDT1   LD      A,(DIRF2)       ;GET SHIP DIRECTION
5806 D5       03490         PUSH    DE              ;DE => SHIP SCREEN POS.
5807 D5       03500         PUSH    DE
5808 DDE1     03510         POP     IX              ;PUT IN IX
580A 21AC73   03520         LD      HL,SHIPF        ;HL GETS SHIP DATA
580D B7       03530         OR      A               ;WHICH DIRECTION ?
580E F5       03540         PUSH    AF
580F 2803     03550         JR      Z,BKSKP         ;JUMP IF POINT RIGHT
5811 21B073   03560         LD      HL,SHIPB
              03570 ;
5814 0604     03580 BKSKP   LD      B,4             ;LENGTH OF SHIP DATA
5816 1A       03590 SUPDLP  LD      A,(DE)          ;SCREEN CHAR
5817 FEC0     03600         CP      192             ;ALIEN OR SHOT ?
```

```
5819 D44058   03610          CALL    NC,SPEXLP        ;CALL IF YES
581C 7E       03620          LD      A,(HL)           ;GET SHIP CHARACTER
581D 12       03630          LD      (DE),A           ;PUT ON SCREEN
581E 13       03640          INC     DE
581F 23       03650          INC     HL
5820 10F4     03660          DJNZ    SUPDLP
              03670 ;
5822 3AB673   03680          LD      A,(MVFLG3)
5825 B7       03690          OR      A                ;MOVING ?
5826 2815     03700          JR      Z,UPDEXT         ;JUMP IF NO
5828 3AB573   03710          LD      A,(MVCHR)        ;GET FLAME CHARACTER
582B EE3F     03720          XOR     3FH              ;CHANGE IT
582D 32B573   03730          LD      (MVCHR),A
5830 4F       03740          LD      C,A
5831 F1       03750          POP     AF               ;GET FLAGS
5832 F5       03760          PUSH    AF
5833 2005     03770          JR      NZ,MVBK          ;JUMP IF MOVING LEFT
5835 DD7100   03780          LD      (IX),C
5838 1803     03790          JR      UPDEXT
              03800 ;
583A DD7103   03810 MVBK     LD      (IX+3),C
583D F1       03820 UPDEXT   POP     AF
583E D1       03830          POP     DE               ;DE => SHIP POS.
583F C9       03840          RET
              03850 ;
5840 FEFD     03860 SPEXLP   CP      253              ;MAN ?
5842 C8       03870          RET     Z                ;RETURN IF YES
5843 C37754   03880          JP      SHIP             ;EXPLODE SHIP
              03890 ;
              03900 ;        *****   MOVE SHIP UP   *****
              03910 ;
5846 2AA773   03920 UP       LD      HL,(SHPOS)       ;GET SHIP SCREEN POS.
5849 CD4B5D   03930          CALL    SB64             ;DE => OLD POS.
584C 3AA173   03940          LD      A,(SHROW)        ;GET SHIP ROW
584F 3D       03950          DEC     A
5850 FE02     03960          CP      2                ;TOO HIGH ?
5852 D8       03970          RET     C                ;RETURN IF YES
5853 32A173   03980          LD      (SHROW),A
              03990 ;
5856 3AB673   04000          LD      A,(MVFLG3)
5859 B7       04010          OR      A                ;MOVING ?
585A 280E     04020          JR      Z,UPOT1          ;JUMP IF NO
585C 3AAA73   04030          LD      A,(DIRF2)
585F B7       04040          OR      A                ;WHICH DIRECTION ?
5860 2005     04050          JR      NZ,UPOT3         ;JUMP IF FACING LEFT
5862 CDA358   04060          CALL    PL2
5865 1803     04070          JR      UPOT1
              04080 ;
5867 CD9858   04090 UPOT3    CALL    MI2
              04100 ;
586A CD8757   04110 UPOT1    CALL    ERASE            ;THIS VECTOR CHANGES WITH
              04120                                   ;THE STATUS OF THE SHIP
586D ED53A773 04130          LD      (SHPOS),DE       ;SAVE NEW SHIP POSITION
5871 C9       04140          RET
              04150 ;
              04160 ;        *****   MOVE SHIP DOWN (NOT CARRYING MAN)   *****
```

```
            04170 ;
5872 2AA773 04180 DOWN    LD      HL,(SHPOS)      ;GET SHIP POS.
5875 CD445D 04190         CALL    AD64            ;DE => OLD POS.
5878 3AA173 04200         LD      A,(SHROW)       ;GET GRID ROW
587B 3C     04210         INC     A               ;DOWN ONE
587C FE10   04220         CP      16              ;TOO LOW ?
587E D0     04230         RET     NC              ;RETURN IF YES
587F 32A173 04240         LD      (SHROW),A
            04250 ;
5882 3AB673 04260         LD      A,(MVFLG3)      ;MOVING ?
5885 B7     04270         OR      A
5886 28E2   04280         JR      Z,UPOT1         ;JUMP IF NO
5888 3AAA73 04290         LD      A,(DIRF2)
588B B7     04300         OR      A               ;WHICH DIRECTION ?
588C 2005   04310         JR      NZ,DWNOT2       ;JUMP IF FACING LEFT
588E CD9858 04320         CALL    MI2
5891 18D7   04330         JR      UPOT1
            04340 ;
5893 CDA358 04350 DWNOT2  CALL    PL2
5896 18D2   04360         JR      UPOT1
            04370 ;
5898 7D     04380 MI2     LD      A,L             ;CHECK FOR SCREEN LIMIT
5899 E63F   04390         AND     3FH
589B FE2D   04400         CP      2DH
589D C8     04410         RET     Z               ;RETURN IF AT LIMIT
589E 23     04420         INC     HL              ;MOVE SHIP RIGHT
589F CD0C57 04430         CALL    SCOL1           ;UPDATE SHIP COLUMN
58A2 C9     04440         RET
            04450 ;
58A3 7D     04460 PL2     LD      A,L             ;CHECK FOR SCREEN LIMIT
58A4 E63F   04470         AND     3FH
58A6 FE0F   04480         CP      0FH
58A8 C8     04490         RET     Z               ;RETURN IF AT LIMIT
58A9 2B     04500         DEC     HL              ;MOVE SHIP LEFT
58AA CD1457 04510         CALL    SCOL2           ;UPDATE SHIP COLUMN
58AD C9     04520         RET
            04530 ;
            04540 ;       *****   MOVE SHIP DOWN (CARRYING MAN)   *****
            04550 ;
58AE 2AA773 04560 DOWN3   LD      HL,(SHPOS)      ;GET SHIP POSITION
58B1 CD445D 04570         CALL    AD64            ;MOVE DOWN
58B4 3AA173 04580         LD      A,(SHROW)       ;CHECK FOR ON THE
58B7 3C     04590         INC     A               ;BOTTOM ROW
58B8 32A173 04600         LD      (SHROW),A
58BB FE0E   04610         CP      14              ;ON BOTTOM ?
58BD 38AB   04620         JR      C,UPOT1         ;JUMP IF NO
            04630 ;
58BF E5     04640         PUSH    HL              ;SAVE NEW SHIP POS.
58C0 D5     04650         PUSH    DE              ;SAVE OLD SHIP POS.
58C1 218757 04660         LD      HL,ERASE        ;RESTORE "ERASE" VECTOR
58C4 226B58 04670         LD      (UPOT1+1),HL
58C7 217258 04680         LD      HL,DOWN         ;RESTORE "DOWN" VECTOR
58CA 223E56 04690         LD      (DWNCNG+1),HL
58CD 210358 04700         LD      HL,UPDT1        ;RESTORE "UPDATE" VECTOR
58D0 22A257 04710         LD      (UPDATE+1),HL
58D3 01F401 04720         LD      BC,500          ;SCORE VALUE
```

```
58D6 CDED4F   04730          CALL    HIT             ;UPDATE SCORE
58D9 CD934F   04740          CALL    BINK
              04750 ;
58DC FD210274 04760          LD      IY,MANTBL       ;FIND AND UPDATE
58E0 3AA373   04770          LD      A,(CRYENT)      ;MAN ENTRY
58E3 CD2B5B   04780          CALL    SPSER2          ;SEARCH ON ENTRY #
58E6 FD360300 04790          LD      (IY+3),0        ;RESET FLAGS
58EA FD36000F 04800          LD      (IY),15         ;ROW #
58EE 3AA273   04810          LD      A,(SHCOL)
58F1 FD7701   04820          LD      (IY+1),A        ;COLUMN
              04830 ;
58F4 D1       04840          POP     DE              ;GET OLD SHIP POS.
58F5 E1       04850          POP     HL              ;GET NEW SHIP POS.
58F6 C36A58   04860          JP      UPOT1
              04870 ;
              04880 ;************************************
              04890 ;**     DRAW MOUNTAINS ROUTINE      **
              04900 ;************************************
              04910 ;
58F9 3AA073   04920 DRAW     LD      A,(DRWFLG)
58FC B7       04930          OR      A               ;DRAW MOUNTAINS ?
58FD 202E     04940          JR      NZ,MNDLY        ;JUMP IF NO
              04950 ;
58FF 21003F   04960          LD      HL,3F00H        ;START ROW
5902 ED5BBA73 04970          LD      DE,(DPOINT)     ;START POSITION ON
              04980                                  ;MOUNTAIN ROW OFFSET TABLE
5906 0640     04990          LD      B,64            ;COLUMN COUNT
              05000 ;
5908 E5       05010 DRAW2    PUSH    HL              ;SAVE COLUMN TOP
5909 C5       05020          PUSH    BC              ;SAVE COLUMN COUNT
590A CD1B59   05030          CALL    CLEAR           ;ERASE THIS COLUMN (4 ROWS)
590D 1A       05040          LD      A,(DE)          ;GET ROW OFFSET
590E 85       05050          ADD     A,L             ;ADD TO SCREEN POS.
590F 6F       05060          LD      L,A
5910 15       05070          DEC     D               ;CHANGE TO DATA TABLE
5911 1A       05080          LD      A,(DE)          ;GET DISPLAY DATA
5912 14       05090          INC     D               ;BACK TO ROW OFFSET TABLE
5913 77       05100          LD      (HL),A          ;PUT ON SCREEN
5914 1C       05110          INC     E               ;UPDATE DATA POINTER
5915 C1       05120          POP     BC
5916 E1       05130          POP     HL              ;GET TOP OF COLUMN
5917 23       05140          INC     HL              ;NEXT COLUMN
5918 10EE     05150          DJNZ    DRAW2
591A C9       05160          RET                     ;ALL DONE
              05170 ;
591B E5       05180 CLEAR    PUSH    HL              ;SAVE SCREEN POINTER
591C C5       05190          PUSH    BC              ;SAVE COUNTER
591D D5       05200          PUSH    DE              ;SAVE DATA POINTER
591E 114000   05210          LD      DE,64           ;ROW OFFSET
5921 0604     05220          LD      B,4             ;ROW COUNT
5923 3E80     05230          LD      A,128
5925 77       05240 CLRLP    LD      (HL),A
5926 19       05250          ADD     HL,DE
5927 10FC     05260          DJNZ    CLRLP
5929 D1       05270          POP     DE
592A C1       05280          POP     BC
```

```
592B E1         05290           POP     HL
592C C9         05300           RET
                05310 ;
                05320 ;         *****   EQUALIZING DELAY FOR SHIP IN SPACE   *****
                05330 ;
592D C5         05340 MNDLY     PUSH    BC
592E 010004     05350           LD      BC,400H
5931 0B         05360 MNDLP     DEC     BC
5932 78         05370           LD      A,B
5933 B1         05380           OR      C
5934 20FB       05390           JR      NZ,MNDLP
5936 C1         05400           POP     BC
5937 C9         05410           RET
                05420 ;
5938            01870 *GET      FIRE/ASM                ;FIRE & HIT DETECT
                00010 ;******************
                00020 ;**       FIRE     **
                00030 ;******************
                00040 ;
5938 3AD15B     00050 FIRE      LD      A,(SHTF1)       ;CAN WE FIRE YET ?
593B B7         00060           OR      A
593C C0         00070           RET     NZ              ;RETURN IF NO
593D 3C         00080           INC     A
593E 32D15B     00090           LD      (SHTF1),A       ;SET FLAG
                00100 ;
5941 3A9873     00110           LD      A,(FIRCNT)      ;UPDATE SHOTS FIRED
5944 3C         00120           INC     A
5945 329873     00130           LD      (FIRCNT),A
                00140 ;
5948 CD2D4F     00150           CALL    DOWN4           ;MAKE SOME NOISE
594B 2AA773     00160           LD      HL,(SHPOS)      ;FIND SHIP POSITION
594E 3AAA73     00170           LD      A,(DIRF2)       ;FIND DIRECTION
5951 B7         00180           OR      A
5952 202B       00190           JR      NZ,FIREB        ;JUMP IF RIGHT => LEFT
                00200 ;
5954 3C         00210           INC     A
5955 32D35B     00220           LD      (SHTDIR),A      ;STORE SHOT DIRECTION
5958 23         00230           INC     HL
5959 23         00240           INC     HL
595A 22D55B     00250           LD      (SHTBL),HL      ;WHERE TO START THE SHOT
595D 0600       00260           LD      B,0             ;SHOT LENGTH COUNTER
                00270 ;
595F 23         00280 FIRLP1    INC     HL
5960 7E         00290           LD      A,(HL)          ;CHECK FOR HIT
5961 FEEB       00300           CP      235             ;SHOT ?
5963 2804       00310           JR      Z,FIRSK1        ;JUMP IF YES
5965 FEC0       00320           CP      192             ;ALIEN ?
5967 300F       00330           JR      NC,EXPL1        ;JUMP IF YES
                00340 ;
5969 365F       00350 FIRSK1    LD      (HL),95         ;DRAW LINE
596B 04         00360           INC     B               ;UPDATE LENGTH COUNTER
596C 7D         00370           LD      A,L
596D E63F       00380           AND     3FH             ;WRAP AROUND ?
596F FE3F       00390           CP      3FH
5971 20EC       00400           JR      NZ,FIRLP1       ;JUMP IF NO
                00410 ;
```

```
5973 78        00420 FIREXT  LD      A,B             ;GET THE LENGTH
5974 32D45B    00430         LD      (LENCNT),A      ;SAVE IT
5977 C9        00440         RET
               00450 ;
5978 C5        00460 EXPL1   PUSH    BC              ;SAVE LENGTH COUNT
5979 CD9F59    00470         CALL    EXPL2
597C C1        00480         POP     BC              ;GET LENGTH COUNT
597D 18F4      00490         JR      FIREXT
               00500 ;
597F 3C        00510 FIREB   INC     A
5980 32D35B    00520         LD      (SHTDIR),A      ;STORE SHOT DIRECTION
5983 22D55B    00530         LD      (SHTBL),HL      ;SAVE SHOT ADDRESS
5986 2B        00540         DEC     HL
5987 0600      00550         LD      B,0             ;SHOT LENGTH COUNTER
               00560 ;
5989 04        00570 FIRLP2  INC     B
598A 7E        00580         LD      A,(HL)          ;CHECK FOR HIT
598B FEEB      00590         CP      235             ;SHOT ?
598D 2804      00600         JR      Z,FIRSK2        ;JUMP IF YES
598F FEC0      00610         CP      192             ;ALIEN ?
5991 30E5      00620         JR      NC,EXPL1        ;JUMP IF YES
5993 365F      00630 FIRSK2  LD      (HL),95         ;DRAW LINE
5995 2B        00640         DEC     HL              ;UPDATE POINTER
5996 7D        00650         LD      A,L
5997 E63F      00660         AND     3FH             ;WRAP AROUND ?
5999 FE3F      00670         CP      3FH
599B 20EC      00680         JR      NZ,FIRLP2       ;JUMP IF NO
599D 18D4      00690         JR      FIREXT
               00700 ;
599F 3680      00710 EXPL2   LD      (HL),128        ;ERASE CHARACTER
59A1 CDFC53    00720         CALL    EXPSH           ;SAVE INFO TO EXPLODE
               00730                                 ;TABLE.
               00740 ;
59A4 32A473    00750         LD      (CHRPAS),A      ;SAVE CHARACTER TYPE
59A7 FEE0      00760         CP      224             ;LANDER ?
59A9 204D      00770         JR      NZ,NXT1         ;JUMP IF NO
               00780 ;
               00790 ;       **********  LANDER HIT  **********
               00800 ;
59AB DD21BE73  00810         LD      IX,LNDTBL       ;LANDER TABLE
59AF DD35FF    00820         DEC     (IX-1)          ;DECREMENT ACTIVE COUNT
59B2 CDC35A    00830         CALL    SEARCH          ;SEARCH ON SCREEN ADDRESS
               00840 ;
59B5 016400    00850         LD      BC,100          ;SCORE VALUE
59B8 DDCB036E  00860         BIT     5,(IX+3)        ;LOCKED ON MAN ?
59BC CA7D5A    00870         JP      Z,SERSKP        ;JUMP IF NO
               00880 ;
59BF C5        00890         PUSH    BC              ;SAVE SCORE
59C0 FD210274  00900         LD      IY,MANTBL       ;MAN TABLE
59C4 DDCB0366  00910         BIT     4,(IX+3)        ;MAN IN AIR ?
59C8 2013      00920         JR      NZ,LNHIT2       ;JUMP IF YES
59CA FD23      00930         INC     IY
59CC FD46FF    00940         LD      B,(IY-1)        ;GET ACTIVE COUNT
59CF DD6604    00950         LD      H,(IX+4)        ;GET LANDER COLUMN
59D2 CD125B    00960         CALL    SPSER3          ;SEARCH ON COLUMN
59D5 FD360300  00970         LD      (IY+3),0        ;RESET ALL MAN FLAGS
```

```
59D9 C1         00980          POP     BC              ;GET SCORE
59DA C37D5A     00990          JP      SERSKP
                01000 ;
59DD DD7E04     01010 LNHIT2   LD      A,(IX+4)        ;ENTRY # OF MAN
59E0 CD2B5B     01020          CALL    SPSER2          ;SEARCH ON ENTRY #
59E3 C1         01030          POP     BC              ;GET SCORE
                01040 ;
59E4 FD360303   01050          LD      (IY+3),3        ;SET TAKEN & DOWN FLAGS
59E8 DD7E00     01060          LD      A,(IX)          ;ROW FROM LANDER
59EB 3C         01070          INC     A               ;ONE ROW DOWN
59EC FD7700     01080          LD      (IY),A          ;PUT ON MAN TABLE
                01090 ;
59EF 21A557     01100          LD      HL,UPDT3        ;CHANGE UPDATE VECTOR
59F2 22A257     01110          LD      (UPDATE+1),HL
59F5 C37D5A     01120          JP      SERSKP
                01130 ;
59F8 FEFD       01140 NXT1     CP      253             ;MAN ?
59FA 2026       01150          JR      NZ,NXT2         ;JUMP IF NO
                01160 ;
                01170 ;        **********  MAN HIT  **********
                01180 ;
59FC FD210274   01190          LD      IY,MANTBL       ;MAN TABLE
5A00 FD35FF     01200          DEC     (IY-1)          ;DECREMENT ACTIVE COUNT
5A03 CD065B     01210          CALL    SPSER1          ;SEARCH ON COLUMN
5A06 FDCB03FE   01220          SET     7,(IY+3)        ;KILL OFF MAN
5A0A FDCB034E   01230          BIT     1,(IY+3)        ;LANDER LOCKED IN ?
5A0E C8         01240          RET     Z               ;RETURN IF NO
                01250 ;
5A0F FD7E04     01260          LD      A,(IY+4)        ;ENTRY # OF LANDER
5A12 FD21BE73   01270          LD      IY,LNDTBL       ;LANDER TABLE
5A16 CD2B5B     01280          CALL    SPSER2          ;SEARCH ON ENTRY #
5A19 FD7E03     01290          LD      A,(IY+3)        ;GET FLAGS
5A1C E680       01300          AND     80H             ;CLEAR ALL FLAGS EXCEPT
                01310                                  ;THE ACTIVE FLAG
5A1E FD7703     01320          LD      (IY+3),A        ;FOR LANDER ENTRY
5A21 C9         01330          RET
                01340 ;
5A22 FEC5       01350 NXT2     CP      197             ;MUTANT ?
5A24 200C       01360          JR      NZ,NXT3         ;JUMP IF NO
                01370 ;
                01380 ;        **********  MUTANT HIT  **********
                01390 ;
5A26 DD216474   01400          LD      IX,MUTBL        ;MUTANT TABLE
5A2A CDC35A     01410          CALL    SEARCH
5A2D 019600     01420          LD      BC,150          ;SCORE VALUE
5A30 1852       01430          JR      SERSK1
                01440 ;
5A32 FEDC       01450 NXT3     CP      220             ;CRUISER HIT ?
5A34 200C       01460          JR      NZ,NXT4         ;JUMP IF NO
                01470 ;
                01480 ;        **********  CRUISER HIT  **********
                01490 ;
5A36 DD21C674   01500          LD      IX,CRUTBL       ;CRUISER TABLE
5A3A CDC35A     01510          CALL    SEARCH
5A3D 01C800     01520          LD      BC,200          ;SCORE VALUE
5A40 183B       01530          JR      SERSKP
```

```
                01540 ;
5A42 FEF0       01550 NXT4     CP      240             ;BOMBER HIT ?
5A44 200C       01560          JR      NZ,NXT5         ;JUMP IF NO
                01570 ;
                01580 ;        **********  BOMBER HIT **********
                01590 ;
5A46 DD21FE74   01600          LD      IX,BMRTBL       ;BOMBER TABLE
5A4A CDC35A     01610          CALL    SEARCH
5A4D 01FA00     01620          LD      BC,250          ;SCORE VALUE
5A50 182B       01630          JR      SERSKP
                01640 ;
5A52 FED6       01650 NXT5     CP      214             ;BOMB HIT ?
5A54 2008       01660          JR      NZ,NXT6         ;JUMP IF NO
                01670 ;
                01680 ;        ********** BOMB HIT **********
                01690 ;
5A56 DD211E75   01700          LD      IX,BMBTBL       ;BOMB TABLE
5A5A CDC35A     01710          CALL    SEARCH
5A5D C9         01720          RET
                01730 ;
5A5E FEC2       01740 NXT6     CP      194             ;POD HIT ?
5A60 200C       01750          JR      NZ,NXT7         ;JUMP IF NO
                01760 ;
                01770 ;        ********** POD HIT **********
                01780 ;
5A62 DD21C975   01790          LD      IX,PODTBL       ;POD TABLE
5A66 CDC35A     01800          CALL    SEARCH
5A69 01E803     01810          LD      BC,1000         ;SCORE VALUE
5A6C 180F       01820          JR      SERSKP
                01830 ;
5A6E FEE5       01840 NXT7     CP      229             ;SWARMER HIT ?
5A70 C0         01850          RET     NZ              ;RETURN IF NO
                01860 ;
                01870 ;        ********** SWARMER HIT **********
                01880 ;
5A71 DD21DD75   01890          LD      IX,SRMTBL       ;SWARMER TABLE
5A75 CDC35A     01900          CALL    SEARCH
5A78 019600     01910          LD      BC,150          ;SCORE VALUE
5A7B 1807       01920          JR      SERSK1
                01930 ;
5A7D 3A9D73     01940 SERSKP   LD      A,(WAVCNT)      ;UPDATE HIT COUNTER
5A80 3C         01950          INC     A
5A81 329D73     01960          LD      (WAVCNT),A
                01970 ;
5A84 CDED4F     01980 SERSK1   CALL    HIT             ;UPDATE PLAYERS SCORE
5A87 C9         01990          RET
                02000 ;
                02010 ;        *****   CLEAR SHOT FROM SCREEN   *****
                02020 ;
5A88 3AD25B     02030 SHTCLR   LD      A,(SHTF2)       ;PAUSE BEFORE CLEARING
5A8B EE01       02040          XOR     1               ;SHOT
5A8D 32D25B     02050          LD      (SHTF2),A
5A90 C8         02060          RET     Z
                02070 ;
5A91 2AD55B     02080          LD      HL,(SHTBL)      ;GET SHOT ADDRESS
5A94 3AD35B     02090          LD      A,(SHTDIR)      ;GET SHOT DIRECTION
```

```
5A97 3D       02100          DEC     A
5A98 FAAE5A   02110          JP      M,CLEXT          ;JUMP IF INVALID
5A9B 2816     02120          JR      Z,CLLFT          ;JUMP IF LEFT => RIGHT
              02130 ;
5A9D 3D       02140          DEC     A
5A9E 200E     02150          JR      NZ,CLEXT         ;JUMP IF INVALID
              02160 ;
5AA0 3AD45B   02170          LD      A,(LENCNT)       ;GET LENGTH OF SHOT
5AA3 47       02180          LD      B,A
              02190 ;
5AA4 2B       02200 CLLP1    DEC     HL
5AA5 7E       02210          LD      A,(HL)           ;DONT ERASE UNLESS IT
5AA6 FE5F     02220          CP      95               ;IS A SHOT CHARACTER.
5AA8 2002     02230          JR      NZ,CLSKP1        ;JUMP IF IT IS NOT
5AAA 3680     02240          LD      (HL),128         ;CLEAR
5AAC 10F6     02250 CLSKP1   DJNZ    CLLP1
              02260 ;
5AAE AF       02270 CLEXT    XOR     A                ;CLEAR DIRECTION
5AAF 32D35B   02280          LD      (SHTDIR),A
5AB2 C9       02290          RET
              02300 ;
5AB3 3AD45B   02310 CLLFT    LD      A,(LENCNT)       ;GET SHOT LENGTH
5AB6 47       02320          LD      B,A
              02330 ;
5AB7 23       02340 CLLP2    INC     HL
5AB8 7E       02350          LD      A,(HL)           ;DONT ERASE UNLESS IT IS
5AB9 FE5F     02360          CP      95               ;A SHOT CHARACTER.
5ABB 2002     02370          JR      NZ,CLSKP2        ;JUMP IF IT IS NOT
5ABD 3680     02380          LD      (HL),128         ;CLEAR
5ABF 10F6     02390 CLSKP2   DJNZ    CLLP2
              02400 ;
5AC1 18EB     02410          JR      CLEXT
              02420 ;
              02430 ;************************************
              02440 ;**      SEARCH DEVICE TABLE       **
              02450 ;** IX MUST POINT TO TABLE START   **
              02460 ;**      HL MUST = SCREEN POS.     **
              02470 ;**      (ROWPAS) MUST = ROW #     **
              02480 ;************************************
              02490 ;
              02500 ;        *****   SEARCH AND DESTROY ENTRY   *****
              02510 ;
5AC3 CDCC5A   02520 SEARCH   CALL    SERCH1           ;FIND TABLE ENTRY
5AC6 C0       02530          RET     NZ               ;RETURN IF NOT FOUND
5AC7 DDCB03FE 02540          SET     7,(IX+3)         ;SET INACTIVE FLAG
5ACB C9       02550          RET
              02560 ;
              02570 ;        *****   SEARCH WITH NO DESTROY ENTRY   *****
              02580 ;
5ACC DD7E00   02590 SERCH1   LD      A,(IX)           ;GET COUNT
5ACF DD23     02600          INC     IX
5AD1 B7       02610          OR      A                ;ANYTHING THERE ?
5AD2 CAFA5A   02620          JP      Z,NONE           ;JUMP IF NO
              02630 ;
5AD5 47       02640          LD      B,A              ;ENTRY COUNT
5AD6 CDFC5A   02650          CALL    COLCVT           ;FIND COLUMN # FROM
```

```
               02660                                ;SCREEN ADDRESS
5AD9 3AD05B    02670         LD      A,(ROWPAS)     ;GET THE ROW #
5ADC 6F        02680         LD      L,A            ;H = COLUMN #
5ADD 110000    02690         LD      DE,0           ;CLEAR ENTRY OFFSET
               02700 ;
5AE0 DD19      02710 SERLP1  ADD     IX,DE          ;NEXT ENTRY
5AE2 DDCB037E  02720         BIT     7,(IX+3)       ;ACTIVE ENTRY ?
5AE6 200D      02730         JR      NZ,NOMAT       ;JUMP IF NO
5AE8 DD7E01    02740         LD      A,(IX+1)       ;GET COLUMN
5AEB BC        02750         CP      H              ;SAME ?
5AEC 2007      02760         JR      NZ,NOMAT       ;JUMP IF NO
5AEE DD7E00    02770         LD      A,(IX)         ;GET ROW
5AF1 BD        02780         CP      L              ;SAME ?
5AF2 2001      02790         JR      NZ,NOMAT       ;JUMP IF NO
5AF4 C9        02800         RET                    ;RETURN IF MATCH.
               02810 ;
5AF5 110600    02820 NOMAT   LD      DE,6           ;ENTRY OFFSET
5AF8 10E6      02830         DJNZ    SERLP1         ;LOOP FOR ENTRY COUNT
               02840 ;
5AFA 04        02850 NONE    INC     B              ;FLAG NOT FOUND
5AFB C9        02860         RET
               02870 ;
               02880 ;***********************************
               02890 ;**     COLUMN CONVERT            **
               02900 ;**  HL MUST - SCREEN POSITION    **
               02910 ;**  H COMES BACK WITH COLUMN #  **
               02920 ;***********************************
               02930 ;
5AFC 3E3F      02940 COLCVT  LD      A,3FH          ;GET SCREEN OFFSET
5AFE A5        02950         AND     L
5AFF 4F        02960         LD      C,A
5B00 3ABC73    02970         LD      A,(WINDOW)     ;GET GRID START
5B03 81        02980         ADD     A,C
5B04 67        02990         LD      H,A            ;SAVE COLUMN
5B05 C9        03000         RET
               03010 ;
               03020 ;*******************************************
               03030 ;**     SEARCH ON COLUMN # ONLY           **
               03040 ;**     IY MUST = TABLE START             **
               03050 ;**     HL MUST = SCREEN POSITION         **
               03060 ;*******************************************
               03070 ;
5B06 FD7E00    03080 SPSER1  LD      A,(IY)         ;GET ENTRY COUNT
5B09 FD23      03090         INC     IY
5B0B B7        03100         OR      A
5B0C 281B      03110         JR      Z,SREXT2       ;JUMP IF NONE
5B0E 47        03120         LD      B,A
5B0F CDFC5A    03130         CALL    COLCVT         ;FIND COLUMN #
5B12 110000    03140 SPSER3  LD      DE,0           ;ENTRY FOR COLUMN IN H
5B15 FD19      03150 SERLP2  ADD     IY,DE          ;NEXT ENTRY
5B17 FDCB037E  03160         BIT     7,(IY+3)       ;ACTIVE ENTRY ?
5B1B 2007      03170         JR      NZ,NOMAT2      ;JUMP IF NO
5B1D FD7E01    03180         LD      A,(IY+1)       ;GET COLUMN #
5B20 BC        03190         CP      H              ;SAME ?
5B21 2001      03200         JR      NZ,NOMAT2      ;JUMP IF NO
5B23 C9        03210         RET
```

```
                 03220 ;
5B24 110600      03230 NOMAT2  LD      DE,6            ;ENTRY OFFSET
5B27 10EC        03240         DJNZ    SERLP2
                 03250 ;
5B29 04          03260 SREXT2  INC     B               ;FLAG NOT FOUND
5B2A C9          03270         RET
                 03280 ;
                 03290 ;****************************************
                 03300 ;**      SEARCH ON ENTRY NUMBER        **
                 03310 ;**       IY MUST = TABLE START        **
                 03320 ;**       A MUST = ENTRY #             **
                 03330 ;****************************************
                 03340 ;
5B2B FD23        03350 SPSER2  INC     IY
5B2D 110600      03360         LD      DE,6            ;ENTRY OFFSET
5B30 3D          03370 SERLP4  DEC     A               ;THIS ENTRY ?
5B31 C8          03380         RET     Z               ;RETURN IF YES
5B32 FD19        03390         ADD     IY,DE           ;NEXT ENTRY
5B34 18FA        03400         JR      SERLP4
                 03410 ;
                 03420 ;*********************************
                 03430 ;**         EMPTY SEARCH        **
                 03440 ;**     FINDS EMPTY ENTRY AT    **
                 03450 ;**     THE END OF THE TABLE    **
                 03460 ;**                             **
                 03470 ;** IX MUST POINT TO TABLE START **
                 03480 ;** DE MUST = ROW/COLUMN #'S    **
                 03490 ;*********************************
                 03500 ;
5B36 DD7E00      03510 EMPTY   LD      A,(IX)          ;GET ACTIVE COUNT
5B39 DDBEFF      03520         CP      (IX-1)          ;SAME AS MAX ?
5B3C D0          03530         RET     NC              ;RETURN IF YES
5B3D 3C          03540         INC     A
5B3E DD7700      03550         LD      (IX),A          ;INCREMENT ACTIVE COUNT
5B41 DD23        03560         INC     IX
5B43 D5          03570         PUSH    DE              ;SAVE ROW/COLUMN
5B44 110600      03580         LD      DE,6            ;ENTRY OFFSET
                 03590 ;
5B47 3D          03600 EMPLP   DEC     A
5B48 2804        03610         JR      Z,EMDON         ;JUMP IF DONE
5B4A DD19        03620         ADD     IX,DE           ;NEXT ENTRY
5B4C 18F9        03630         JR      EMPLP
                 03640 ;
5B4E D1          03650 EMDON   POP     DE              ;GET ROW/COLUMN
5B4F DD7300      03660         LD      (IX),E          ;ROW
5B52 DD7201      03670         LD      (IX+1),D        ;COLUMN
5B55 37          03680         SCF                     ;FLAG FOUND
5B56 C9          03690         RET
                 03700 ;
                 03710 ;*********************************
                 03720 ;**         EMPTY SEARCH        **
                 03730 ;**   FINDS AN EMPTY SPOT IN    **
                 03740 ;**   THE MIDDLE OF THE TABLE   **
                 03750 ;**                             **
                 03760 ;** IY MUST POINT TO TABLE START **
                 03770 ;** DE MUST = ROW/COLUMN #'S    **
```

```
               03780 ;***********************************
               03790 ;
5B57 FD7EFF    03800 EMPTY2 LD      A,(IY-1)          ;GET ACTIVE COUNT
5B5A FDBE00    03810        CP      (IY)              ;COMPARE WITH MAX
5B5D D0        03820        RET     NC                ;RETURN IF SAME
5B5E 3C        03830        INC     A
5B5F FD77FF    03840        LD      (IY-1),A          ;UPDATE ACTIVE
5B62 FD7E00    03850        LD      A,(IY)            ;GET COUNT
5B65 FD23      03860        INC     IY
5B67 010600    03870        LD      BC,6              ;ENTRY OFFSET
               03880 ;
5B6A FDCB037E  03890 EMPLP2 BIT     7,(IY+3)          ;ACTIVE ?
5B6E 2007      03900        JR      NZ,EMDON2         ;JUMP IF NO
5B70 FD09      03910        ADD     IY,BC             ;NEXT ENTRY
5B72 3D        03920        DEC     A                 ;END OF TABLE ?
5B73 20F5      03930        JR      NZ,EMPLP2         ;JUMP IF NO
5B75 B7        03940        OR      A                 ;FLAG NOT FOUND
5B76 C9        03950        RET
               03960 ;
5B77 FD7300    03970 EMDON2 LD      (IY),E
5B7A FD7201    03980        LD      (IY+1),D
5B7D 37        03990        SCF                       ;FLAG FOUND
5B7E C9        04000        RET
               04010 ;
               04020 ;========================================
               04030 ;**            CRUNCH                  **
               04040 ;**     MOVES ALL ENTRIES BACK ONE      **
               04050 ;**     POSITION STARTING WITH THE      **
               04060 ;**     ENTRY FOLLOWING (IX)            **
               04070 ;**     B = COUNT TO MOVE + 1           **
               04080 ;****************************************
               04090 ;
5B7F 3AFF44    04100 CRUNCH LD      A,(MASK)
5B82 CBBF      04110        RES     7,A               ;DISABLE INTERRUPTS
5B84 32FF44    04120        LD      (MASK),A
5B87 DDE5      04130        PUSH    IX                ;PUT TABLE POS. IN 'IY'
5B89 FDE1      04140        POP     IY
5B8B DDE5      04150        PUSH    IX                ;SAVE CURRENT POS.
5B8D C5        04160        PUSH    BC                ;SAVE COUNT
5B8E 110600    04170        LD      DE,6              ;ENTRY OFFSET
               04180 ;
5B91 FD19      04190 CRLOP  ADD     IY,DE             ;IY => NEXT ENTRY
5B93 FD7E00    04200        LD      A,(IY)            ;MOVE DATA BACK ONE
5B96 DD7700    04210        LD      (IX),A            ;ENTRY POSITION
5B99 FD7E01    04220        LD      A,(IY+1)
5B9C DD7701    04230        LD      (IX+1),A
5B9F FD7E02    04240        LD      A,(IY+2)
5BA2 DD7702    04250        LD      (IX+2),A
5BA5 FD7E03    04260        LD      A,(IY+3)
5BA8 DD7703    04270        LD      (IX+3),A
5BAB FD7E04    04280        LD      A,(IY+4)
5BAE DD7704    04290        LD      (IX+4),A
5BB1 FD7E05    04300        LD      A,(IY+5)
5BB4 DD7705    04310        LD      (IX+5),A
5BB7 DD19      04320        ADD     IX,DE             ;IX = IY
5BB9 10D6      04330        DJNZ    CRLOP
```

```
              04340 ;
5BBB C1       04350         POP     BC                ;GET COUNT
5BBC DDE1     04360         POP     IX                ;GET START ENTRY
5BBE 2ACE5B   04370         LD      HL,(TBLPAS)       ;ONE LESS ENTRY ON
5BC1 35       04380         DEC     (HL)              ;THE TABLE
5BC2 110000   04390         LD      DE,0
5BC5 3AFF44   04400         LD      A,(MASK)
5BC8 CBFF     04410         SET     7,A               ;ENABLE INTERRUPTS
5BCA 32FF44   04420         LD      (MASK),A
5BCD C9       04430         RET
              04440 ;
5BCE 0000     04450 TBLPAS  DW      0
5BD0 00       04460 ROWPAS  DB      0                 ;PASSES ROW # FROM CALLER
5BD1 00       04470 SHTF1   DB      0                 ;SHOT FLAG
5BD2 00       04480 SHTF2   DB      0                 ;PAUSE FLAG
5BD3 00       04490 SHTDIR  DB      0                 ;DIRECTION OF SHOT
              04500                                   ;1=L->R  2=R->L
5BD4 00       04510 LENCNT  DB      0                 ;LENGTH OF SHOT
5BD5 0000     04520 SHTBL   DW      00                ;ADDRESS OF SHOT
              04530 ;
              04540 ;*****************************************
              04550 ;**     CHECK FOR CHARACTER ON-SCREEN   **
              04560 ;**     IX MUST POINT TO ENTRY          **
              04570 ;*****************************************
              04580 ;
5BD7 DD5E00   04590 ONCHK   LD      E,(IX)            ;GET ROW
5BDA DD5601   04600         LD      D,(IX+1)          ;GET COLUMN
5BDD DDCB0396 04610         RES     2,(IX+3)          ;CLEAR ON SCREEN FLAG
5BE1 3ABC73   04620         LD      A,(WINDOW)        ;CURRENT SCREEN POSITION
5BE4 4F       04630         LD      C,A
5BE5 7A       04640         LD      A,D               ;GET CHARACTER COL.
5BE6 91       04650         SUB     C                 ;CALCULATE OFFSET
5BE7 FE3F     04660         CP      63                ;ON SCREEN ?
5BE9 D0       04670         RET     NC                ;RETURN IF NO
              04680 ;
5BEA 32C660   04690         LD      (OFFSET),A
5BED 21003C   04700         LD      HL,3C00H          ;SCREEN ROW 0 - COL 0
5BF0 014000   04710         LD      BC,64             ;ROW OFFSET
5BF3 7B       04720         LD      A,E               ;ROW # OF CHARACTER
              04730 ;
5BF4 09       04740 ONLP1   ADD     HL,BC
5BF5 3D       04750         DEC     A
5BF6 20FC     04760         JR      NZ,ONLP1
              04770 ;
5BF8 3AC660   04780         LD      A,(OFFSET)
5BFB 85       04790         ADD     A,L
5BFC 6F       04800         LD      L,A
5BFD E63F     04810         AND     3FH
5BFF FE3D     04820         CP      3DH               ;2 FROM EDGE ?
5C01 3009     04830         JR      NC,CHRERS         ;JUMP IF YES
              04840 ;
5C03 FE02     04850         CP      2                 ;2 FROM OTHER EDGE ?
5C05 3805     04860         JR      C,CHRERS          ;JUMP IF YES
              04870 ;
5C07 DDCB03D6 04880         SET     2,(IX+3)          ;FLAG ON SCREEN
5C0B C9       04890         RET
```

```
                04900 ;
                04910 ;*********************************
                04920 ;**      ERASE IF AT EDGES       **
                04930 ;*********************************
                04940 ;
5C0C 2B         04950 CHRERS  DEC     HL              ;SAVE SCREEN ADDRESS
5C0D D5         04960         PUSH    DE              ;SAVE ROW/COLUMN
5C0E CD445D     04970         CALL    AD64            ;DROP 1 ROW
5C11 0607       04980         LD      B,7
5C13 3E80       04990         LD      A,128
5C15 77         05000 ERSLP   LD      (HL),A          ;ERASE MAN IF CARRIED BY
                05010                                 ;LANDER.
5C16 12         05020         LD      (DE),A          ;ERASE CHARACTER
5C17 23         05030         INC     HL
5C18 13         05040         INC     DE
5C19 10FA       05050         DJNZ    ERSLP
5C1B D1         05060         POP     DE
5C1C C9         05070         RET
                05080 ;
                05090 ;*********************************
                05100 ;**      NEW SHOT TO TABLE       **
                05110 ;*********************************
                05120 ;
5C1D DD5601     05130 SHOCHK  LD      D,(IX+1)        ;COLUMN
5C20 DD5E00     05140         LD      E,(IX)          ;ROW
5C23 DDE5       05150         PUSH    IX
5C25 DD219D75   05160         LD      IX,SHOTBL
5C29 CD365B     05170         CALL    EMPTY           ;FIND A SPOT
5C2C 300C       05180         JR      NC,SHEX         ;JUMP IF NONE
5C2E DD360201   05190         LD      (IX+2),1        ;DELAY VALUE
5C32 DD360508   05200         LD      (IX+5),8        ;DELAY RESET
5C36 DD360320   05210         LD      (IX+3),20H      ;FLAG SEARCH FOR SHIP
5C3A DDE1       05220 SHEX    POP     IX
5C3C C9         05230         RET
                05240 ;
                05250 ;*****************************************
                05260 ;**      CHECK FOR EDGE OF SCREEN        **
                05270 ;*****************************************
                05280 ;
5C3D 7D         05290 SIDCHK  LD      A,L
5C3E E63F       05300         AND     3FH             ;ALL THE WAY OVER ?
5C40 FE3F       05310         CP      3FH
5C42 C8         05320         RET     Z               ;RETURN IF YES
5C43 B7         05330         OR      A
5C44 C9         05340         RET
                05350 ;
                05360 ;*********************************
                05370 ;**      CLEAR ENTIRE SCREEN     **
                05380 ;*********************************
                05390 ;
5C45 21003C     05400 CLS     LD      HL,3C00H        ;START POSITION
5C48 E5         05410         PUSH    HL
5C49 D1         05420         POP     DE
5C4A 13         05430         INC     DE
5C4B 3680       05440         LD      (HL),128
5C4D 01FF03     05450         LD      BC,1023         ;CHARACTER COUNT
```

```
5C50 EDB0     05460         LDIR
5C52 C9       05470         RET
              05480 ;
              05490 ;*********************************
              05500 ;**     CLEAR PLAYING FIELD     **
              05510 ;*********************************
              05520 ;
5C53 3AFF44   05530 GMECLR  LD      A,(MASK)
5C56 F5       05540         PUSH    AF
5C57 CBBF     05550         RES     7,A             ;DISABLE INTERRUPTS
5C59 32FF44   05560         LD      (MASK),A
5C5C 21403C   05570         LD      HL,3C40H        ;START POSITION
5C5F 01BF03   05580         LD      BC,03BFH        ;BYTE COUNT
5C62 E5       05590         PUSH    HL
5C63 D1       05600         POP     DE
5C64 13       05610         INC     DE
5C65 3680     05620         LD      (HL),128
5C67 EDB0     05630         LDIR
5C69 F1       05640         POP     AF
5C6A CBCF     05650         SET     1,A             ;SET SCORE UPDATE
5C6C 32FF44   05660         LD      (MASK),A
5C6F C9       05670         RET
              05680 ;
              05690 ;*************************************
              05700 ;**     FILL BUFFER WITH RANDOM     **
              05710 ;**         COLUMN NUMBERS          **
              05720 ;*************************************
              05730 ;
5C70 48       05740 RNDCOL  LD      C,B             ;RANDOM NUMBER COUNT
5C71 219C5C   05750         LD      HL,BUFFER       ;BUFFER POINTER
5C74 D5       05760         PUSH    DE              ;SAVE FOR RETURN
5C75 C5       05770         PUSH    BC              ;SAVE FOR RETURN
5C76 E5       05780         PUSH    HL              ;SAVE FOR RETURN
5C77 C5       05790         PUSH    BC
5C78 E5       05800         PUSH    HL
              05810 ;
5C79 36FF     05820 CMLP    LD      (HL),255        ;CLEAR TABLE
5C7B 23       05830         INC     HL
5C7C 10FB     05840         DJNZ    CMLP
              05850 ;
5C7E E1       05860 TRYAGN  POP     HL              ;TABLE POS.
5C7F C1       05870         POP     BC              ;COUNT
5C80 C5       05880 NXTMN1  PUSH    BC
5C81 E5       05890         PUSH    HL
              05900 ;
5C82 ED5F     05910         LD      A,R             ;GET RANDOM
5C84 57       05920         LD      D,A
5C85 ED5F     05930         LD      A,R
5C87 82       05940         ADD     A,D
5C88 41       05950         LD      B,C             ;GET MAX COUNT
5C89 219C5C   05960         LD      HL,BUFFER       ;STORAGE FOR #'S PICKED
5C8C BE       05970 CMLP2   CP      (HL)            ;ALREADY USED ?
5C8D 28EF     05980         JR      Z,TRYAGN        ;JUMP IF YES
5C8F 23       05990         INC     HL
5C90 10FA     06000         DJNZ    CMLP2
              06010 ;
```

```
5C92 E1         06020           POP     HL              ;CURRENT POS. IN TABLE
5C93 77         06030           LD      (HL),A          ;SAVE #
5C94 23         06040           INC     HL              ;NEXT POS.
                06050 ;
5C95 C1         06060           POP     BC              ;COUNT
5C96 10E8       06070           DJNZ    NXTMN1
5C98 E1         06080           POP     HL              ;GET BUFFER START
5C99 C1         06090           POP     BC              ;GET COUNT
5C9A D1         06100           POP     DE
5C9B C9         06110           RET
                06120 ;
0019            06130 BUFFER    DS      25
                06140 ;
                06150 ;*****************************************
                06160 ;**       GET RANDOM ROW & COLUMN IN DE   **
                06170 ;*****************************************
                06180 ;
5CB5 ED5F       06190 ROWCOL    LD      A,R             ;ENTRY FOR COLUMN & ROW
5CB7 57         06200           LD      D,A
5CB8 ED5F       06210           LD      A,R
5CBA 82         06220           ADD     A,D
5CBB 57         06230           LD      D,A
5CBC ED5F       06240 ROWRND    LD      A,R             ;ENTRY FOR ROW ONLY
5CBE E607       06250           AND     7               ;GETS ROW # BETWEEN
5CC0 FE05       06260           CP      5               ;2 & 4 INCLUSIVE
5CC2 30F8       06270           JR      NC,ROWRND
5CC4 FE02       06280           CP      2
5CC6 38F4       06290           JR      C,ROWRND
5CC8 5F         06300           LD      E,A
5CC9 C9         06310           RET
                06320 ;
                06330 ;*****************************************
                06340 ;**       CHECK LIMITS FOR WARP-IN        **
                06350 ;*****************************************
                06360 ;
5CCA 7A         06370 WRPCHK    LD      A,D             ;GET COLUMN
5CCB 21BC73     06380           LD      HL,WINDOW
5CCE 96         06390           SUB     (HL)
5CCF FE0C       06400           CP      12
5CD1 D0         06410           RET     NC
5CD2 3E0B       06420           LD      A,11
5CD4 82         06430           ADD     A,D
5CD5 57         06440           LD      D,A
5CD6 C9         06450           RET
                06460 ;
                06470 ;*********************************
                06480 ;**       CLEAR SCREEN CHARACTER  **
                06490 ;*********************************
                06500 ;
5CD7 E5         06510 MUCLR     PUSH    HL              ;SAVE SCREEN ADDRESS
5CD8 2B         06520           DEC     HL
5CD9 3E40       06530           LD      A,40H           ;LIMIT CHECK VALUE
5CDB 0603       06540           LD      B,3
                06550 ;
5CDD 3680       06560 MUCLRP    LD      (HL),128
5CDF 23         06570           INC     HL
```

```
5CE0 BC        06580         CP      H
5CE1 2802      06590         JR      Z,MUCLSK
5CE3 10F8      06600         DJNZ    MUCLRP
               06610 ;
5CE5 E1        06620 MUCLSK  POP     HL
5CE6 C9        06630         RET
               06640 ;
               06650 ;*********************************
               06660 ;**      WARP ONTO SCREEN       **
               06670 ;*********************************
               06680 ;
5CE7 DD7E04    06690 WARPIN  LD      A,(IX+4)          ;GET PHASE COUNTER
5CEA 3D        06700         DEC     A                 ;UPDATE
5CEB DD7704    06710         LD      (IX+4),A
5CEE 201B      06720         JR      NZ,WRPXT1         ;JUMP IF NOT DONE
               06730 ;
5CF0 DDCB03B6  06740         RES     6,(IX+3)          ;RESET WARP-IN
5CF4 060F      06750         LD      B,15              ;CLEAR CHARACTERS
5CF6 3E01      06760         LD      A,1
5CF8 3680      06770 WRPLP0  LD      (HL),128
5CFA 23        06780         INC     HL
5CFB EE03      06790         XOR     3
5CFD D3FF      06800         OUT     (255),A           ;MAKE NOISE
5CFF F5        06810         PUSH    AF
5D00 CD3D5C    06820         CALL    SIDCHK            ;DO LIMIT CHECK
5D03 2804      06830         JR      Z,WEXTX           ;JUMP IF OUT OF LIMIT
5D05 F1        06840         POP     AF
5D06 10F0      06850         DJNZ    WRPLP0
5D08 C9        06860         RET
               06870 ;
5D09 F1        06880 WEXTX   POP     AF
5D0A C9        06890         RET
               06900 ;
5D0B 11E455    06910 WRPXT1  LD      DE,WARP6          ;LAST PHASE ?
5D0E 3D        06920         DEC     A
5D0F 281B      06930         JR      Z,WRPXT2          ;JUMP IF YES
               06940 ;
5D11 11D555    06950         LD      DE,WARP5          ;FIFTH PHASE ?
5D14 3D        06960         DEC     A
5D15 2815      06970         JR      Z,WRPXT2          ;JUMP IF YES ?
               06980 ;
5D17 11C655    06990         LD      DE,WARP4          ;FOURTH PHASE ?
5D1A 3D        07000         DEC     A
5D1B 280F      07010         JR      Z,WRPXT2          ;JUMP IF YES
               07020 ;
5D1D 11B755    07030         LD      DE,WARP3          ;THIRD PHASE ?
5D20 3D        07040         DEC     A
5D21 2809      07050         JR      Z,WRPXT2          ;JUMP IF YES
               07060 ;
5D23 11A855    07070         LD      DE,WARP2          ;SECOND PHASE ?
5D26 3D        07080         DEC     A
5D27 2803      07090         JR      Z,WRPXT2          ;JUMP IF YES
               07100 ;
5D29 119955    07110         LD      DE,WARP1          ;FIRST PHASE
               07120 ;
5D2C 060F      07130 WRPXT2  LD      B,15              ;15 CHARACTERS
```

```
5D2E 3E01     07140         LD      A,1
5D30 EE03     07150 WRPLP1  XOR     3
5D32 F5       07160         PUSH    AF
5D33 D3FF     07170         OUT     (255),A         ;MAKE NOISE
5D35 1A       07180         LD      A,(DE)          ;GET CHARACTER
5D36 77       07190         LD      (HL),A          ;PUT ON SCREEN
5D37 23       07200         INC     HL
5D38 13       07210         INC     DE
5D39 CD3D5C   07220         CALL    SIDCHK          ;CHECK LIMIT
5D3C 2804     07230         JR      Z,WEXTX2        ;JUMP IF AT LIMIT
5D3E F1       07240         POP     AF
5D3F 10EF     07250         DJNZ    WRPLP1
5D41 C9       07260         RET
              07270 ;
5D42 F1       07280 WEXTX2  POP     AF
5D43 C9       07290         RET
              07300 ;
              07310 ;*************************************************
              07320 ;**     ADD 64 TO HL - DE GETS OLD VALUE        **
              07330 ;*************************************************
              07340 ;
5D44 014000   07350 AD64    LD      BC,64           ;ADD 64 TO HL
5D47 E5       07360         PUSH    HL              ;DE RETURNS WITH
5D48 09       07370         ADD     HL,BC           ;OLD VALUE
5D49 D1       07380         POP     DE
5D4A C9       07390         RET
              07400 ;
              07410 ;*************************************************
              07420 ;**   SUBTRACT 64 FROM HL - DE GETS OLD VALUE   **
              07430 ;*************************************************
              07440 ;
5D4B 01C0FF   07450 SB64    LD      BC,-64          ;SUBTRACT 64 FROM HL
5D4E E5       07460         PUSH    HL              ;DE RETURNS WITH OLD
5D4F 09       07470         ADD     HL,BC           ;VALUE
5D50 D1       07480         POP     DE
5D51 C9       07490         RET
              07500 ;
              07510 ;*****************************************
              07520 ;**   PRINT MESSAGE FROM (HL) TO (DE)   **
              07530 ;*****************************************
              07540 ;
5D52 11003C   07550 PRINT   LD      DE,3C00H
5D55 7E       07560 PRNT    LD      A,(HL)          ;DISPLAY ROUTINE
5D56 23       07570         INC     HL              ;PRINTS TO 0
5D57 B7       07580         OR      A
5D58 C8       07590         RET     Z
5D59 12       07600         LD      (DE),A
5D5A 13       07610         INC     DE
5D5B 18F8     07620         JR      PRNT
              07630 ;
5D5D          01880 *GET    SHOT/ASM                ;ALIEN SHOT ROUTINES
              00010 ;*************************************************
              00020 ;**             SHOT PROCESSER                  **
              00030 ;**     CHARACTER = 235                         **
              00040 ;**     IX    = ROW # (GRID)                    **
              00050 ;**     IX+1  = COLUMN # (GRID)                 **
```

```
                   00060 ;**      IX+2 = DELAY BEFORE MOVE                     **
                   00070 ;**      IX+3 = FLAGS -- BIT 0 = MOVE UP              **
                   00080 ;**                          1 = MOVE DOWN            **
                   00090 ;**                          2 = ON SCREEN            **
                   00100 ;**                          3 = MOVE RIGHT           **
                   00110 ;**                          4 = UNUSED               **
                   00120 ;**                          5 = FIND DIRECTION       **
                   00130 ;**                          6 = MOVE LEFT            **
                   00140 ;**                          7 = ACTIVE/INACTIVE      **
                   00150 ;**      IX+4 = UP/DOWN TO LEFT/RIGHT RATIO           **
                   00160 ;**      IX+5 = DELAY RESET                           **
                   00170 ;*************************************************
                   00180 ;
5D5D DD219D75      00190 SHOT    LD      IX,SHOTBL       ;SHOT TABLE
5D61 DD22CE5B      00200         LD      (TBLPAS),IX
5D65 DD7E00        00210         LD      A,(IX)          ;GET ACTIVE ENTRY COUNT
5D68 B7            00220         OR      A               ;ANY THERE ?
5D69 C8            00230         RET     Z               ;RETURN IF NO
                   00240 ;
5D6A 47            00250         LD      B,A             ;GET COUNT
5D6B DD23          00260         INC     IX
5D6D 110000        00270         LD      DE,0            ;CLEAR ENTRY OFFSET
                   00280 ;
5D70 DD19          00290 SHOLOP  ADD     IX,DE           ;NEXT SHOT DATA
5D72 DDCB037E      00300         BIT     7,(IX+3)        ;ACTIVE ?
5D76 2806          00310         JR      Z,CNSK9         ;JUMP IF YES
5D78 CD7F5B        00320         CALL    CRUNCH          ;MOVE ENTRIES DOWN 1
5D7B 10F3          00330         DJNZ    SHOLOP          ;LOOP FOR ACTIVE COUNT
5D7D C9            00340         RET
                   00350 ;
5D7E C5            00360 CNSK9   PUSH    BC
5D7F DDCB036E      00370         BIT     5,(IX+3)        ;NEED DIRECTION ?
5D83 C41F5E        00380         CALL    NZ,DIRFND       ;CALL IF YES
5D86 DDCB037E      00390         BIT     7,(IX+3)        ;STILL ACTIVE ?
5D8A 202D          00400         JR      NZ,SHOEX8       ;JUMP IF NO
5D8C CDD75B        00410         CALL    ONCHK           ;FIND POSITION ON SCREEN
5D8F DDCB0356      00420         BIT     2,(IX+3)        ;ON SCREEN ?
5D93 282B          00430         JR      Z,SHOKIL        ;JUMP IF NO
                   00440 ;
5D95 CDD75C        00450         CALL    MUCLR           ;CLEAR OLD CHARACTER
5D98 DD3502        00460         DEC     (IX+2)          ;MOVE YET ?
5D9B 201A          00470         JR      NZ,SHODRW       ;JUMP IF NO
                   00480 ;
5D9D DD7E05        00490         LD      A,(IX+5)        ;SPEED UP SHOT MOVE
5DA0 3D            00500         DEC     A
5DA1 DD7705        00510         LD      (IX+5),A
5DA4 2005          00520         JR      NZ,SCNTSK       ;JUMP IF NOT FASTEST
5DA6 3E01          00530         LD      A,1
5DA8 DD7705        00540         LD      (IX+5),A
                   00550 ;
5DAB DD7702        00560 SCNTSK  LD      (IX+2),A        ;RESET MOVE DELAY COUNT
5DAE CDC65D        00570         CALL    SHOMOV          ;MOVE SHOT
5DB1 DDCB0356      00580         BIT     2,(IX+3)        ;STILL ON SCREEN ?
5DB5 2809          00590         JR      Z,SHOKIL        ;JUMP IF NO
                   00600 ;
5DB7 36EB          00610 SHODRW  LD      (HL),235        ;DISPLAY CHARACTER
```

```
                00620 ;
5DB9 C1         00630 SHOEX8  POP    BC                ;GET COUNT
5DBA 110600     00640         LD     DE,6              ;ENTRY OFFSET
5DBD 10B1       00650         DJNZ   SHOLOP
5DBF C9         00660         RET
                00670 ;
5DC0 DDCB03FE   00680 SHOKIL  SET    7,(IX+3)          ;KILL OFF SHOT
5DC4 18F3       00690         JR     SHOEX8
                00700 ;
5DC6 DD7E04     00710 SHOMOV  LD     A,(IX+4)          ;GET RATIO FLAG
5DC9 B7         00720         OR     A                 ;UP/DOWN ALWAYS ?
5DCA FAD45D     00730         JP     M,SHOMV2          ;JUMP IF YES
5DCD EE01       00740         XOR    1                 ;TOGGLE FLAG
5DCF DD7704     00750         LD     (IX+4),A          ;SAVE IT
5DD2 2831       00760         JR     Z,SHRICK          ;SKIP UP/DOWN IF ZERO
                00770 ;
5DD4 DD7E00     00780 SHOMV2  LD     A,(IX)            ;GET ROW
5DD7 DDCB0346   00790         BIT    0,(IX+3)          ;MOVE UP ?
5DDB 2812       00800         JR     Z,SHDNCK          ;JUMP IF NO
                00810 ;
5DDD 3D         00820         DEC    A
5DDE DD7700     00830         LD     (IX),A
5DE1 FE02       00840         CP     2                 ;OFF SCREEN ?
5DE3 3005       00850         JR     NC,SHOK1          ;JUMP IF NO
                00860 ;
5DE5 DDCB0396   00870         RES    2,(IX+3)          ;FLAG OFF SCREEN
5DE9 C9         00880         RET
                00890 ;
5DEA CD4B5D     00900 SHOK1   CALL   SB64              ;MOVE ON SCREEN
5DED 1816       00910         JR     SHRICK
                00920 ;
5DEF DDCB034E   00930 SHDNCK  BIT    1,(IX+3)          ;MOVE DOWN ?
5DF3 2810       00940         JR     Z,SHRICK          ;JUMP IF NO
                00950 ;
5DF5 3C         00960         INC    A
5DF6 DD7700     00970         LD     (IX),A
5DF9 FE10       00980         CP     16                ;OFF SCREEN ?
5DFB 3805       00990         JR     C,SHOK2           ;JUMP IF NO
                01000 ;
5DFD DDCB0396   01010         RES    2,(IX+3)          ;FLAG OFF SCREEN
5E01 C9         01020         RET
                01030 ;
5E02 CD445D     01040 SHOK2   CALL   AD64
5E05 DD7E01     01050 SHRICK  LD     A,(IX+1)          ;GET COLUMN
5E08 DDCB035E   01060         BIT    3,(IX+3)          ;MOVE RIGHT ?
5E0C 2806       01070         JR     Z,SHLFT           ;JUMP IF NO
                01080 ;
5E0E 23         01090         INC    HL                ;MOVE ON SCREEN
5E0F 3C         01100         INC    A                 ;MOVE ON GRID
5E10 DD7701     01110         LD     (IX+1),A          ;SAVE NEW POS.
5E13 C9         01120         RET
                01130 ;
5E14 DDCB0376   01140 SHLFT   BIT    6,(IX+3)          ;MOVE LEFT ?
5E18 C8         01150         RET    Z                 ;RETURN IF NO
                01160 ;
5E19 2B         01170         DEC    HL                ;MOVE ON SCREEN
```

```
5E1A 3D         01180           DEC     A                   ;MOVE ON GRID
5E1B DD7701     01190           LD      (IX+1),A
5E1E C9         01200           RET
                01210 ;
5E1F DD7E03     01220 DIRFND    LD      A,(IX+3)            ;GET FLAGS
5E22 E604       01230           AND     4                   ;CANCEL MOVES
5E24 DD7703     01240           LD      (IX+3),A
                01250 ;
5E27 3AA173     01260           LD      A,(SHROW)           ;GET SHIP ROW
5E2A DD9600     01270           SUB     (IX)                ;SUBTRACT SHOT ROW
5E2D 280C       01280           JR      Z,SHNOUD            ;NO UP/DOWN IF SAME
                01290 ;
5E2F 3006       01300           JR      NC,SHDOWN           ;MOVE DOWN IF LESS THAN
                01310 ;
5E31 DDCB03C6   01320           SET     0,(IX+3)            ;SET MOVE UP FLAG
5E35 1804       01330           JR      SHNOUD
                01340 ;
5E37 DDCB03CE   01350 SHDOWN    SET     1,(IX+3)            ;SET MOVE DOWN FLAG
                01360 ;
5E3B DD360480   01370 SHNOUD    LD      (IX+4),80H          ;SET UP/DOWN ONLY
5E3F 3AA273     01380           LD      A,(SHCOL)           ;GET SHIP COLUMN
5E42 DD9601     01390           SUB     (IX+1)              ;FIND DIFFERENCE
5E45 C8         01400           RET     Z                   ;RETURN IF SAME
                01410 ;
5E46 300B       01420           JR      NC,SHRT1            ;JUMP IF MAYBE MOVE RIGHT
                01430 ;
5E48 2F         01440           CPL
5E49 FE80       01450           CP      128                 ;HALF WAY POINT
5E4B 300A       01460           JR      NC,SHRT2            ;MOVE RIGHT
                01470 ;
5E4D DDCB03F6   01480 SHLF2     SET     6,(IX+3)            ;SET MOVE LEFT
5E51 1808       01490           JR      RATFND
                01500 ;
5E53 FE80       01510 SHRT1     CP      128                 ;MOVE LEFT ?
5E55 30F6       01520           JR      NC,SHLF2            ;JUMP IF YES
                01530 ;
5E57 DDCB03DE   01540 SHRT2     SET     3,(IX+3)            ;SET MOVE RIGHT
5E5B FE1F       01550 RATFND    CP      31                  ;CLOSE ENOUGH ?
5E5D 300B       01560           JR      NC,NOKOFF           ;JUMP IF NO
5E5F DD360400   01570           LD      (IX+4),0
5E63 FE0A       01580           CP      10                  ;REAL CLOSE ?
5E65 D0         01590           RET     NC                  ;RETURN IF NO
5E66 DD3504     01600           DEC     (IX+4)              ;SET MINUS
5E69 C9         01610           RET
                01620 ;
5E6A DDCB03FE   01630 NOKOFF    SET     7,(IX+3)            ;KILL OFF SHOT
5E6E C9         01640           RET
                01650 ;
5E6F            01890 *GET      LANDER/ASM                  ;LANDER/MAN/MUTANT ROUTINES
                00010 ;*****************************************************
                00020 ;**                  LANDER                          **
                00030 ;**       CHARACTER = 224                             **
                00040 ;**       IX    = ROW #  (GRID)                       **
                00050 ;**       IX+1 = COLUMN # (GRID)                      **
                00060 ;**       IX+2 = DELAY BEFORE MOVE                    **
                00070 ;**       IX+3 = FLAGS -- BIT 0 = UP/DOWN             **
```

```
               00080 ;**                        1 = LEFT/RIGHT      **
               00090 ;**                        2 = ON SCREEN       **
               00100 ;**                        3 = OVER MAN        **
               00110 ;**                        4 = MAN IN AIR      **
               00120 ;**                        5 = SEARCH FOR MAN  **
               00130 ;**                        6 = WARP-IN         **
               00140 ;**                        7 = ACTIVE/INACTIVE **
               00150 ;**      IX+4 = ENTRY # OF MAN CARRIED         **
               00160 ;**           = COLUMN # TO GO TO              **
               00170 ;**           = WARP-IN PHASE COUNTER          **
               00180 ;**      IX+5 = SHOT DELAY                     **
               00190 ;**********************************************
               00200 ;
5E6F DD21BE73  00210 LANDER  LD      IX,LNDTBL       ;LANDER TABLE
5E73 DD7EFF    00220         LD      A,(IX-1)        ;ACTIVE ENTRIES
5E76 B7        00230         OR      A               ;ANY THERE ?
5E77 C8        00240         RET     Z               ;RETURN IF NO
5E78 DD4600    00250         LD      B,(IX)          ;GET ENTRY COUNT
5E7B DD23      00260         INC     IX              ;IX=> ROW
5E7D 110000    00270         LD      DE,0            ;CLEAR ENTRY OFFSET
               00280 ;
5E80 DD19      00290 LNDLOP  ADD     IX,DE           ;NEXT LANDER DATA
5E82 C5        00300         PUSH    BC              ;SAVE ENTRY COUNT
5E83 DDCB037E  00310         BIT     7,(IX+3)        ;ACTIVE ENTRY ?
5E87 2062      00320         JR      NZ,LNDEX9       ;JUMP IF NO
               00330 ;
5E89 CDD75B    00340         CALL    ONCHK
5E8C DDCB0376  00350         BIT     6,(IX+3)        ;WARP IN ?
5E90 2811      00360         JR      Z,WRSK2         ;JUMP IF NO
               00370 ;
5E92 DDCB0356  00380         BIT     2,(IX+3)        ;ON SCREEN ?
5E96 2006      00390         JR      NZ,WRSK3        ;JUMP IF YES
               00400 ;
5E98 DDCB03B6  00410         RES     6,(IX+3)        ;RESET WARP-IN
5E9C 1805      00420         JR      WRSK2
               00430 ;
5E9E CDE75C    00440 WRSK3   CALL    WARPIN          ;DO WARP-IN PHASE
5EA1 1841      00450         JR      LNDEX8
               00460 ;
5EA3 DDCB036E  00470 WRSK2   BIT     5,(IX+3)        ;SEARCH FOR MAN ?
5EA7 CCF75F    00480         CALL    Z,MSRCH         ;CALL IF YES
5EAA DDCB0356  00490         BIT     2,(IX+3)        ;ON SCREEN ?
5EAE C4FF5E    00500         CALL    NZ,LNCLR        ;CALL IF YES
5EB1 CD445D    00510         CALL    AD64            ;GET POSSIBLE MAN POS.
5EB4 22C760    00520         LD      (LNMAN),HL      ;SAVE IT
5EB7 EB        00530         EX      DE,HL           ;HL => LANDER POSITION
5EB8 DD3502    00540         DEC     (IX+2)          ;MOVE YET ?
5EBB 2014      00550         JR      NZ,LNDRW        ;JUMP IF NO
               00560 ;
5EBD DD36020F  00570         LD      (IX+2),15       ;RESET MOVE DELAY COUNT
5EC1 DDCB036E  00580         BIT     5,(IX+3)        ;LOCKED ON MAN ?
5EC5 2804      00590         JR      Z,LNEX5         ;JUMP IF NO
               00600 ;
5EC7 DDCB035E  00610         BIT     3,(IX+3)        ;OVER MAN ?
5ECB CC205F    00620 LNEX5   CALL    Z,LRMV          ;MOVE TOWARD MAN IF NO
5ECE CD4F5F    00630         CALL    UDMV            ;MOVE UP/DOWN
```

```
                 00640 ;
5ED1 DDCB0356    00650 LNDRW    BIT     2,(IX+3)        ;ON SCREEN ?
5ED5 2814        00660          JR      Z,LNDEX9        ;JUMP IF NO
                 00670 ;
5ED7 36E0        00680          LD      (HL),224        ;DRAW LANDER
                 00690 ;
5ED9 DDCB0366    00700          BIT     4,(IX+3)        ;MAN IN AIR ?
5EDD 2805        00710          JR      Z,LNDEX8        ;JUMP IF NO
                 00720 ;
5EDF 2AC760      00730          LD      HL,(LNMAN)      ;MAN POS.
5EE2 36FD        00740          LD      (HL),253        ;DRAW MAN
                 00750 ;
5EE4 DDCB0356    00760 LNDEX8   BIT     2,(IX+3)        ;ON SCREEN ?
5EE8 C4F45E      00770          CALL    NZ,LNSHOT       ;DO SHOT IF YES
5EEB C1          00780 LNDEX9   POP     BC              ;GET ENTRY COUNTER
5EEC 110600      00790          LD      DE,6            ;ENTRY OFFSET
5EEF 05          00800          DEC     B
5EF0 C2805E      00810          JP      NZ,LNDLOP       ;LOOP TIL DONE
5EF3 C9          00820          RET
                 00830 ;
                 00840 ;        *****   PROCESS LANDER SHOT   *****
                 00850 ;
5EF4 DD3505      00860 LNSHOT   DEC     (IX+5)          ;SHOT DELAY COUNT
5EF7 C0          00870          RET     NZ              ;RETURN IF NOT YET
                 00880 ;
5EF8 DD360564    00890          LD      (IX+5),100      ;RESET COUNTER
5EFC C31D5C      00900          JP      SHOCHK          ;SHOOT
                 00910 ;
                 00920 ;        *****   CLEAR LANDER FROM SCREEN   *****
                 00930 ;
5EFF E5          00940 LNCLR    PUSH    HL              ;SAVE SCREEN POS.
5F00 2B          00950          DEC     HL
5F01 0603        00960          LD      B,3
5F03 3680        00970 LNCLP1   LD      (HL),128        ;CLEAR LANDER
5F05 23          00980          INC     HL
5F06 10FB        00990          DJNZ    LNCLP1
5F08 E1          01000          POP     HL
5F09 DDCB0366    01010          BIT     4,(IX+3)        ;MAN IN AIR ?
5F0D C8          01020          RET     Z               ;RETURN IF NO
5F0E CD445D      01030          CALL    AD64
5F11 2B          01040          DEC     HL
5F12 0603        01050          LD      B,3
5F14 3E40        01060          LD      A,40H
5F16 3680        01070 LNCLP2   LD      (HL),128        ;CLEAR MAN
5F18 23          01080          INC     HL
5F19 BC          01090          CP      H
5F1A 2802        01100          JR      Z,LNCLSK
5F1C 10F8        01110          DJNZ    LNCLP2
                 01120 ;
5F1E EB          01130 LNCLSK   EX      DE,HL           ;HL-> LANDER POSITION
5F1F C9          01140          RET
                 01150 ;
                 01160 ;        *****   MOVE LANDER ROUTINES   *****
                 01170 ;
5F20 DD7E04      01180 LRMV     LD      A,(IX+4)        ;GET COLUMN OF MAN
5F23 DD9601      01190          SUB     (IX+1)          ;FIND DISTANCE
```

```
5F26 381A      01200         JR     C,LNRT1          ;JUMP IF MOVE RIGHT CHECK
               01210 ;
5F28 2F        01220         CPL
5F29 FE80      01230         CP     128              ;HALFWAY ?
5F2B 3019      01240         JR     NC,LNRT2         ;MOVE RIGHT IF OVER
               01250 ;
5F2D DD3501    01260 LNLF2   DEC    (IX+1)           ;MOVE LEFT
5F30 DD7E01    01270         LD     A,(IX+1)
5F33 2B        01280         DEC    HL               ;MOVE LEFT ON SCREEN
5F34 DDBE04    01290 LLSK2   CP     (IX+4)           ;SAME COLUMN AS MAN ?
5F37 C0        01300         RET    NZ               ;RETURN IF NO
5F38 DDCB036E  01310         BIT    5,(IX+3)         ;LOCKED ON MAN ?
5F3C C8        01320         RET    Z                ;RETURN IF NO
5F3D DDCB03DE  01330         SET    3,(IX+3)         ;FLAG SAME COLUMN
5F41 C9        01340         RET
               01350 ;
5F42 FE80      01360 LNRT1   CP     128              ;HALF WAY ?
5F44 30E7      01370         JR     NC,LNLF2         ;JUMP IF OVER
               01380 ;
5F46 DD3401    01390 LNRT2   INC    (IX+1)           ;MOVE RIGHT
5F49 DD7E01    01400         LD     A,(IX+1)
5F4C 23        01410         INC    HL               ;MOVE RIGHT ON SCREEN
5F4D 18E5      01420         JR     LLSK2
               01430 :
5F4F DDCB0366  01440 UDMV    BIT    4,(IX+3)         ;MAN IN AIR ?
5F53 2048      01450         JR     NZ,LNDUP         ;JUMP IF YES
5F55 DDCB035E  01460         BIT    3,(IX+3)         ;OVER MAN ?
5F59 2008      01470         JR     NZ,LNDWN         ;JUMP IF YES
               01480 ;
5F5B ED5F      01490         LD     A,R              ;RANDOM UP/DOWN MOVEMENT
5F5D E60F      01500         AND    0FH
5F5F FE08      01510         CP     8
5F61 303A      01520         JR     NC,LNDUP
               01530 ;
5F63 DD7E00    01540 LNDWN   LD     A,(IX)           ;GET COLUMN #
5F66 3C        01550         INC    A                ;MOVE IT DOWN
5F67 FE0F      01560         CP     15               ;TOO FAR ?
5F69 382B      01570         JR     C,LDWNOK         ;JUMP IF NO
               01580 :
5F6B DDCB035E  01590         BIT    3,(IX+3)         ;OVER MAN ?
5F6F C8        01600         RET    Z                ;RETURN IF NO
               01610 ;
5F70 E5        01620         PUSH   HL               ;SAVE SCREEN POS.
5F71 CD445D    01630         CALL   AD64             ;GET SCREEN ADDRESS OF MAN
5F74 22C760    01640         LD     (LNMAN),HL       ;MAN POS.
5F77 FD210374  01650         LD     IY,MANTBL+1      ;MAN TABLE
5F7B FD46FF    01660         LD     B,(IY-1)         ;GET COUNT
5F7E DD6604    01670         LD     H,(IX+4)         ;COLUMN #
5F81 CD125B    01680         CALL   SPSER3           ;SEARCH ON COLUMN #
5F84 3A0274    01690         LD     A,(MANTBL)
5F87 3C        01700         INC    A
5F88 90        01710         SUB    B                ;A= ENTRY # OF MAN
5F89 FDCB03EE  01720         SET    5,(IY+3)         ;SET CARRIED BY LANDER
5F8D DD7704    01730         LD     (IX+4),A         ;ENTRY # OF MAN
5F90 DDCB03E6  01740         SET    4,(IX+3)         ;SET MAN IN AIR
5F94 E1        01750         POP    HL               ;LANDER POSITION
```

```
5F95 C9         01760          RET
                01770 ;
5F96 DD7700     01780 LDWNOK   LD      (IX),A          ;UPDATE ROW
5F99 CD445D     01790          CALL    AD64            ;HL => NEW POS.
5F9C C9         01800          RET
                01810 ;
5F9D DD7E00     01820 LNDUP    LD      A,(IX)          ;GET ROW
5FA0 3D         01830          DEC     A               ;MOVE IT UP
5FA1 FE02       01840          CP      2               ;AT THE TOP ?
5FA3 380B       01850          JR      C,MOVLND        ;JUMP IF YES
                01860 ;
5FA5 DD7700     01870          LD      (IX),A          ;UPDATE ROW
5FA8 CD4B5D     01880          CALL    SB64            ;GET NEW SCREEN POS.
5FAB ED53C760   01890          LD      (LNMAN),DE      ;NEW MAN POS.
5FAF C9         01900          RET
                01910 ;
                01920 ;        *****   CHANGE LANDER TO MUTANT   *****
                01930 ;
5FB0 DDCB0366   01940 MOVLND   BIT     4,(IX+3)        ;MAN IN AIR ?
5FB4 C8         01950          RET     Z               ;RETURN IF NO
                01960 ;
5FB5 3ABD73     01970          LD      A,(LNDTBL-1)    ;UPDATE ACTIVE ENTRIES
5FB8 3D         01980          DEC     A               ;FOR LANDER
5FB9 32BD73     01990          LD      (LNDTBL-1),A
5FBC DD7E03     02000          LD      A,(IX+3)        ;GET LANDER FLAGS
5FBF E604       02010          AND     4               ;MASK FOR ON SCREEN
5FC1 F680       02020          OR      80H             ;SET INACTIVE
5FC3 DD7703     02030          LD      (IX+3),A        ;KILL OFF LANDER
5FC6 DD7E04     02040          LD      A,(IX+4)        ;MAN ENTRY #
                02050 ;
5FC9 FD210374   02060          LD      IY,MANTBL+1
5FCD FD35FE     02070          DEC     (IY-2)          ;ONE LESS MAN
5FD0 CD2B5B     02080          CALL    SPSER2          ;SEARCH ON ENTRY #
5FD3 FDCB03FE   02090          SET     7,(IY+3)        ;KILL OFF MAN
                02100 ;
5FD7 DD5E00     02110          LD      E,(IX)          ;GET LANDER ROW
5FDA DD5601     02120          LD      D,(IX+1)        ;GET LANDER COLUMN
5FDD DDE5       02130          PUSH    IX
5FDF DD216474   02140          LD      IX,MUTBL        ;MUTANT TABLE
5FE3 CD365B     02150          CALL    EMPTY           ;FIND EMPTY ENTRY
5FE6 3008       02160          JR      NC,NOSP         ;JUMP IF NONE
                02170 ;
5FE8 DD360205   02180          LD      (IX+2),5
5FEC DD360300   02190          LD      (IX+3),0        ;CLEAR MUTANT FLAGS
                02200 ;
5FF0 DDE1       02210 NOSP     POP     IX              ;GET LANDER TABLE POS.
5FF2 DDCB0396   02220          RES     2,(IX+3)        ;CLEAR ON SCREEN FLAG
5FF6 C9         02230          RET
                02240 ;
                02250 ;        *****   SEARCH FOR MAN TO LOCK ON   *****
                02260 ;
5FF7 FD210374   02270 MSRCH    LD      IY,MANTBL+1     ;START OF ENTRIES
5FFB FD46FF     02280          LD      B,(IY-1)        ;MAX ENTRIES
5FFE 110600     02290          LD      DE,6            ;DATA COUNT/ENTRY
                02300 ;
6001 FDCB037E   02310 MSRLP    BIT     7,(IY+3)        ;ACTIVE ENTRY ?
```

```
6005 2006      02320          JR      NZ,MSRSKP        ;JUMP IF NO
               02330 ;
6007 FDCB034E  02340          BIT     1,(IY+3)         ;TAKEN ?
600B 280B      02350          JR      Z,MSFND          ;JUMP IF NO
               02360 ;
600D FD19      02370 MSRSKP   ADD     IY,DE            ;ADD OFFSET
600F 10F0      02380          DJNZ    MSRLP
               02390 ;
6011 3AA273    02400          LD      A,(SHCOL)        ;MAKE 'EM MOVE TO SHIP
6014 DD7704    02410          LD      (IX+4),A
6017 C9        02420          RET
               02430 ;
6018 FDCB03CE  02440 MSFND    SET     1,(IY+3)         ;SET TAKEN FLAG
601C FD7E01    02450          LD      A,(IY+1)         ;GET COLUMN #
601F DD7704    02460          LD      (IX+4),A
6022 DDCB03EE  02470          SET     5,(IX+3)         ;SET LOCKED ON FLAG
6026 D1        02480          POP     DE               ;RETURN ADDRESS
6027 C1        02490          POP     BC               ;ENTRY COUNTER
6028 3ABE73    02500          LD      A,(LNDTBL)
602B 3C        02510          INC     A
602C 90        02520          SUB     B                ;GET ENTRY # OF LANDER
602D FD7704    02530          LD      (IY+4),A         ;MAN POINTS BACK TO LNDER
6030 C5        02540          PUSH    BC               ;FIX STACK
6031 D5        02550          PUSH    DE
6032 C9        02560          RET
               02570 ;
               02580 ;*************************************************
               02590 ;**               MAN ON BOTTOM ROUTINE           **
               02600 ;**      CHARACTER = 253                          **
               02610 ;**      IX    = ROW                              **
               02620 ;**      IX+1  = COLUMN                           **
               02630 ;**      IX+2  = DELAY WHEN MOVING DOWN           **
               02640 ;**      IX+3  = FLAGS -- BIT 0 = MOVE DOWN       **
               02650 ;**                           1 = TAKEN           **
               02660 ;**                           2 = ON SCREEN       **
               02670 ;**                           3 = UNUSED          **
               02680 ;**                           4 = UNUSED          **
               02690 ;**                           5 = LANDER CARRY    **
               02700 ;**                           6 = SHIP CARRY      **
               02710 ;**                           7 = ACTIVE/INACTIVE **
               02720 ;**      IX+4  = ENTRY # OF LANDER                **
               02730 ;**      IX+5  = UNUSED                           **
               02740 ;*************************************************
               02750 ;
6033 3AA073    02760 MAN      LD      A,(DRWFLG)       ;MOUNTAINS ACTIVE ?
6036 B7        02770          OR      A
6037 C0        02780          RET     NZ               ;RETURN IF NO
               02790 ;
6038 DD210274  02800          LD      IX,MANTBL        ;MAN TABLE
603C DD7EFF    02810          LD      A,(IX-1)         ;ANY LEFT ?
603F B7        02820          OR      A
6040 CA304D    02830          JP      Z,SPACE          ;JUMP IF NO
6043 FA304D    02840          JP      M,SPACE          ; ''  '' ''
               02850 ;
6046 DD4600    02860          LD      B,(IX)           ;GET MAX COUNT
6049 DD23      02870          INC     IX
```

```
604B 110000   02880         LD      DE,0            ;CLEAR ENTRY OFFSET
              02890 ;
604E DD19     02900 MANLOP  ADD     IX,DE           ;NEXT ENTRY
6050 C5       02910         PUSH    BC              ;SAVE COUNT
6051 DDCB037E 02920         BIT     7,(IX+3)        ;ACTIVE ?
6055 2029     02930         JR      NZ,MANEXT       ;JUMP IF NO
              02940 ;
6057 DDCB036E 02950         BIT     5,(IX+3)        ;CARRIED BY LANDER ?
605B 2023     02960         JR      NZ,MANEXT       ;JUMP IF YES
              02970 ;
605D DDCB0376 02980         BIT     6,(IX+3)        ;CARRIED BY SHIP ?
6061 201D     02990         JR      NZ,MANEXT       ;JUMP IF YES
              03000 ;
6063 CDD75B   03010         CALL    ONCHK
6066 DDCB0356 03020         BIT     2,(IX+3)        ;ON SCREEN ?
606A 281B     03030         JR      Z,MANEX8        ;JUMP IF NO
              03040 ;
606C DDCB0346 03050         BIT     0,(IX+3)        ;MOVING DOWN ?
6070 280C     03060         JR      Z,MANDRW        ;JUMP IF NO
              03070 ;
6072 CDD75C   03080         CALL    MUCLR           ;CLEAR CHARACTER
6075 CD9260   03090         CALL    MANMOV          ;MOVE THE MAN
6078 DDCB0356 03100         BIT     2,(IX+3)        ;STILL ON SCREEN ?
607C 2802     03110         JR      Z,MANEXT        ;JUMP IF NO
              03120 ;
607E 36FD     03130 MANDRW  LD      (HL),253        ;DRAW MAN
              03140 ;
6080 C1       03150 MANEXT  POP     BC              ;GET COUNT
6081 110600   03160         LD      DE,6            ;ENTRY OFFSET
6084 10C8     03170         DJNZ    MANLOP
6086 C9       03180         RET
              03190 ;
6087 DDCB0346 03200 MANEX8  BIT     0,(IX+3)        ;MOVING DOWN ?
608B 28F3     03210         JR      Z,MANEXT        ;JUMP IF NO
608D CD9260   03220         CALL    MANMOV          ;MOVE MAN
6090 18EE     03230         JR      MANEXT
              03240 ;
6092 DD3502   03250 MANMOV  DEC     (IX+2)          ;MOVE YET ?
6095 C0       03260         RET     NZ              ;RETURN IF NO
6096 DD36020F 03270         LD      (IX+2),15       ;RESET DELAY COUNTER
609A 7B       03280         LD      A,E             ;GET ROW
609B 3C       03290         INC     A               ;UPDATE
609C FE10     03300         CP      16              ;AT BOTTOM ?
609E 381A     03310         JR      C,KILSKP        ;JUMP IF NO
60A0 DD360380 03320         LD      (IX+3),80H      ;MAKE INACTIVE
60A4 3A0174   03330         LD      A,(MANTBL-1)    ;GET COUNT
60A7 3D       03340         DEC     A               ;UPDATE
60A8 320174   03350         LD      (MANTBL-1),A
60AB E5       03360         PUSH    HL              ;SAVE SCREEN POSITION
60AC 210358   03370         LD      HL,UPDT1        ;CHANGE SHIP DRAW VECTOR
60AF 22A257   03380         LD      (UPDATE+1),HL
60B2 218757   03390         LD      HL,ERASE        ;CHANGE SHIP ERASE VECTOR
60B5 226B58   03400         LD      (UPOT1+1),HL
60B8 E1       03410         POP     HL
60B9 C9       03420         RET
              03430 ;
```

```
60BA DD7700   03440 KILSKP  LD      (IX),A          ;UPDATE ROW
60BD DDCB0356 03450         BIT     2,(IX+3)        ;ON SCREEN ?
60C1 C8       03460         RET     Z               ;RETURN IF NO
60C2 CD445D   03470         CALL    AD64
60C5 C9       03480         RET
              03490 ;
60C6 00       03500 OFFSET  DB      0
60C7 0000     03510 LNMAN   DW      00              ;ERASE POS. OF MAN
              03520 ;
              03530 ;*****************************************************
              03540 ;**                MUTANT                           **
              03550 ;**         CHARACTER = 197                          **
              03560 ;**         IX     = ROW # (GRID)                    **
              03570 ;**         IX+1   = COLUMN # (GRID)                 **
              03580 ;**         IX+2   = DELAY BEFORE MOVE               **
              03590 ;**         IX+3   = FLAGS   BIT 0 = UP/DOWN         **
              03600 ;**                              1 = LEFT/RIGHT      **
              03610 ;**                              2 = ON SCREEN       **
              03620 ;**                              3 = LFT/RGHT CANCEL **
              03630 ;**                              4 = UNUSED          **
              03640 ;**                              5 = MOVE FLAG       **
              03650 ;**                              6 = UNUSED          **
              03660 ;**                              7 = ACTIVE/INACTIVE **
              03670 ;**         IX+4 = UNUSED                            **
              03680 ;**         IX+5 = UNUSED                            **
              03690 ;*****************************************************
              03700 ;
60C9 DD216474 03710 MUTANT  LD      IX,MUTBL        ;GET MUTANT TABLE
60CD DD22CE5B 03720         LD      (TBLPAS),IX     ;PASS IT TO "CRUNCH"
60D1 DD7E00   03730         LD      A,(IX)          ;GET MUTANT COUNT
60D4 B7       03740         OR      A               ;ANY THERE ?
60D5 C8       03750         RET     Z               ;RETURN IF NO
60D6 47       03760         LD      B,A
60D7 DD23     03770         INC     IX              ;IX => FIRST ENTRY
60D9 110000   03780         LD      DE,0            ;CLEAR ENTRY OFFSET
              03790 ;
60DC DD19     03800 MUTLOP  ADD     IX,DE           ;NEXT ENTRY
60DE DDCB037E 03810         BIT     7,(IX+3)        ;ACTIVE ENTRY ?
60E2 2806     03820         JR      Z,CNSKP2        ;JUMP IF YES
              03830 ;
60E4 CD7F5B   03840         CALL    CRUNCH          ;MOVE EVERYTHING DOWN
60E7 10F3     03850         DJNZ    MUTLOP
60E9 C9       03860         RET
              03870 ;
60EA C5       03880 CNSKP2  PUSH    BC              ;SAVE COUNT
60EB CDD75B   03890         CALL    ONCHK
60EE DDCB0356 03900         BIT     2,(IX+3)        ;ON SCREEN ?
60F2 2838     03910         JR      Z,MUEX8         ;JUMP IF NO
              03920 ;
60F4 CDD75C   03930         CALL    MUCLR           ;CLEAR OLD CHARACTER
60F7 DDCB039E 03940         RES     3,(IX+3)        ;CLEAR LFT/RIGHT CANCEL
60FB DDCB03AE 03950         RES     5,(IX+3)        ;CLEAR MOVE FLAG
60FF CD3161   03960         CALL    MUMOV           ;MOVE ON GRID
6102 DDCB036E 03970         BIT     5,(IX+3)        ;MOVE ?
6106 281B     03980         JR      Z,MUDRW         ;JUMP IF NO
              03990 ;
```

```
6108 DDCB0346 04000        BIT    0,(IX+3)        ;MOVE UP ?
610C F5       04010        PUSH   AF
610D CC4B5D   04020        CALL   Z,SB64          ;CALL IF YES
6110 F1       04030        POP    AF
6111 C4445D   04040        CALL   NZ,AD64         ;CALL IF NO
              04050 ;
6114 DDCB035E 04060        BIT    3,(IX+3)        ;MOVE LEFT/RIGHT ?
6118 2809     04070        JR     Z,MUDRW         ;JUMP IF NO
              04080 ;
611A 23       04090        INC    HL              ;MOVE RIGHT
611B DDCB034E 04100        BIT    1,(IX+3)        ;MOVE RIGHT ?
611F 2802     04110        JR     Z,MUDRW         ;JUMP IF YES
6121 2B       04120        DEC    HL              ;CANCEL MOVE RIGHT
6122 2B       04130        DEC    HL              ;MOVE LEFT
              04140 ;
6123 36C5     04150 MUDRW  LD     (HL),197        ;DRAW MUTANT
              04160 ;
6125 C1       04170 MUTEXT POP    BC              ;COUNT
6126 110600   04180        LD     DE,6            ;ENTRY OFFSET
6129 10B1     04190        DJNZ   MUTLOP          ;LOOP TIL END OF TABLE
612B C9       04200        RET
              04210 ;
612C CD3161   04220 MUEX8  CALL   MUMOV           ;MOVE WITHOUT DISPLAY
612F 18F4     04230        JR     MUTEXT
              04240 ;
6131 DD3502   04250 MUMOV  DEC    (IX+2)          ;MOVE YET ?
6134 C0       04260        RET    NZ              ;RETURN IF NO
6135 DD360203 04270        LD     (IX+2),3        ;RESET MOVE DELAY
6139 DDCB03EE 04280        SET    5,(IX+3)        ;SET MOVE FLAG
              04290 ;
613D ED5F     04300        LD     A,R             ;RANDOM UP/DOWN MOVEMENT
613F E60F     04310        AND    0FH
6141 FE08     04320        CP     8
6143 7B       04330        LD     A,E             ;GET ROW #
6144 300C     04340        JR     NC,MUDWN
              04350 ;
6146 3D       04360 MUUP   DEC    A               ;MOVE IT UP
6147 FE02     04370        CP     2               ;OFF SCREEN
6149 3806     04380        JR     C,MUDWN1        ;JUMP IF YES
              04390 ;
614B DDCB0386 04400        RES    0,(IX+3)        ;FLAG UP
614F 180F     04410        JR     MUDOK
              04420 ;
6151 3C       04430 MUDWN1 INC    A               ;CANCEL UP
6152 3C       04440 MUDWN  INC    A               ;MOVE DOWN
6153 FE10     04450        CP     16              ;TOO FAR ?
6155 3006     04460        JR     NC,MUUP1        ;JUMP IF YES
              04470 ;
6157 DDCB03C6 04480        SET    0,(IX+3)        ;FLAG DOWN
615B 1803     04490        JR     MUDOK
              04500 ;
615D 3D       04510 MUUP1  DEC    A               ;CANCEL DOWN
615E 18E6     04520        JR     MUUP
              04530 ;
6160 DD7700   04540 MUDOK  LD     (IX),A          ;PUT NEW ROW IN TABLE
              04550 ;
```

```
6163 3AA273   04560        LD      A,(SHCOL)          ;GET SHIP GRID COLUMN
6166 BA       04570        CP      D                  ;SAME ?
6167 C8       04580        RET     Z                  ;RETURN IF YES
              04590 ;
6168 3AAA73   04600        LD      A,(DIRF2)          ;GET SHIP DIRECTION
616B B7       04610        OR      A
616C 2809     04620        JR      Z,MUSKP5           ;JUMP RIGHT TO LEFT
616E DDCB038E 04630        RES     1,(IX+3)           ;FLAG RIGHT
6172 DD3401   04640        INC     (IX+1)             ;MOVE RIGHT
6175 1807     04650        JR      MUSKP7
              04660 ;
6177 DDCB03CE 04670 MUSKP5 SET     1,(IX+3)           ;FLAG LEFT
617B DD3501   04680        DEC     (IX+1)             ;GO LEFT
              04690 ;
617E DDCB03DE 04700 MUSKP7 SET     3,(IX+3)           ;SET MOVE FLAG
6182 C9       04710        RET
              04720 ;
6183          01900 *GET   CRUISER/ASM                ;CRUISER ROUTINES
              00010 ;*****************************************************
              00020 ;**              CRUISER                            **
              00030 ;**        CHARACTER = 220                          **
              00040 ;**        IX   = ROW # (GRID)                      **
              00050 ;**        IX+1 = COLUMN # (GRID)                   **
              00060 ;**        IX+2 = DELAY BEFORE TURN                 **
              00070 ;**        IX+3 = FLAGS -- BIT 0 = UP/DOWN          **
              00080 ;**                            1 = LEFT/RIGHT       **
              00090 ;**                            2 = ON SCREEN        **
              00100 ;**                            3 = UP/DOWN CANCEL   **
              00110 ;**                            4 = UNUSED           **
              00120 ;**                            5 = UP/DN DELAY      **
              00130 ;**                            6 = WARP IN          **
              00140 ;**                            7 = ACTIVE/INACTIVE  **
              00150 ;**        IX+4 = PHASE COUNTER FOR WARP IN         **
              00160 ;**        IX+5 = SHOT DELAY                        **
              00170 ;*****************************************************
              00180 ;
6183 DD21C674 00190 CRUISE LD      IX,CRUTBL          ;GET CRUISER TABLE
6187 DD22CE5B 00200        LD      (TBLPAS),IX        ;PASS TO CRUNCH
618B DD7E00   00210        LD      A,(IX)
618E B7       00220        OR      A                  ;ANY CRUISERS ?
618F C8       00230        RET     Z                  ;RETURN IF NO
6190 47       00240        LD      B,A
6191 DD23     00250        INC     IX                 ;IX => FIRST ENTRY
6193 110000   00260        LD      DE,0               ;CLEAR ENTRY OFFSET
              00270 ;
6196 DD19     00280 CURLOP ADD     IX,DE              ;NEXT ENTRY
6198 DDCB037E 00290        BIT     7,(IX+3)           ;ACTIVE ENTRY ?
619C 2806     00300        JR      Z,CNSKP3           ;JUMP IF YES
619E CD7F5B   00310        CALL    CRUNCH             ;MOVE 'EM BACK ONE
61A1 10F3     00320        DJNZ    CURLOP
61A3 C9       00330        RET
              00340 ;
61A4 C5       00350 CNSKP3 PUSH    BC                 ;SAVE COUNT
61A5 CDD75B   00360        CALL    ONCHK
61A8 DDCB0356 00370        BIT     2,(IX+3)           ;ON SCREEN ?
61AC 2847     00380        JR      Z,CUEX8            ;JUMP IF NOT ON SCREEN
```

```
                00390 ;
61AE DDCB0376 00400         BIT     6,(IX+3)        ;WARPING IN ?
61B2 2805     00410         JR      Z,WRSK1         ;JUMP IF NO
              00420 ;
61B4 CDE75C   00430         CALL    WARPIN
61B7 1823     00440         JR      CUREXT
              00450 ;
61B9 CDD75C   00460 WRSK1   CALL    MUCLR           ;CLEAR OLD CHARACTER
61BC CDFE61   00470         CALL    POSCHK          ;FIND POS
61BF DDCB035E 00480         BIT     3,(IX+3)        ;MOVE UP/DOWN ?
61C3 280C     00490         JR      Z,CUDRW         ;JUMP IF NO
61C5 DDCB0346 00500         BIT     0,(IX+3)        ;MOVE UP ?
61C9 F5       00510         PUSH    AF
61CA CC4B5D   00520         CALL    Z,SB64          ;CALL IF YES
61CD F1       00530         POP     AF
61CE C4445D   00540         CALL    NZ,AD64         ;CALL IF NO
              00550 ;
61D1 23       00560 CUDRW   INC     HL              ;MOVE RIGHT
61D2 DDCB034E 00570         BIT     1,(IX+3)        ;MOVE RIGHT ?
61D6 2802     00580         JR      Z,CUDRW2        ;JUMP IF YES
              00590 ;
61D8 2B       00600         DEC     HL              ;CANCEL MOVE RIGHT
61D9 2B       00610         DEC     HL              ;MOVE LEFT
61DA 36DC     00620 CUDRW2  LD      (HL),220        ;DRAW CRUISER
              00630 ;
61DC DDCB0356 00640 CUREXT  BIT     2,(IX+3)        ;ON SCREEN ?
61E0 C4EA61   00650         CALL    NZ,CUSHOT       ;DO SHOT IF YES
61E3 C1       00660         POP     BC
61E4 110600   00670         LD      DE,6            ;ENTRY OFFSET
61E7 10AD     00680         DJNZ    CURLOP
61E9 C9       00690         RET
              00700 ;
61EA DD3505   00710 CUSHOT  DEC     (IX+5)          ;SHOOT YET ?
61ED C0       00720         RET     NZ              ;RETURN IF NO
              00730 ;
61EE DD36051E 00740         LD      (IX+5),30       ;RESET SHOT DELAY
61F2 C31D5C   00750         JP      SHOCHK
              00760 ;
61F5 CDFE61   00770 CUEX8   CALL    POSCHK          ;MOVE OFF OF SCREEN
61F8 DDCB03B6 00780         RES     6,(IX+3)        ;RESET WARP IN
61FC 18DE     00790         JR      CUREXT
              00800 ;
61FE 3AA273   00810 POSCHK  LD      A,(SHCOL)       ;GET SHIP COLUMN
6201 92       00820         SUB     D               ;FIND DISTANCE
6202 3010     00830         JR      NC,CURT1        ;JUMP IF MOVE RIGHT CHK
              00840 ;
6204 2F       00850         CPL
6205 FE80     00860         CP      128             ;HALF WAY POINT
6207 300F     00870         JR      NC,CURT2        ;MOVE RIGHT IF OVER
              00880 ;
6209 DDCB034E 00890 CULF2   BIT     1,(IX+3)        ;ALREADY MOVING LEFT ?
620D 2814     00900         JR      Z,CUDCK         ;JUMP IF NO
              00910 ;
620F DD3501   00920 CULF3   DEC     (IX+1)          ;MOVE LEFT (ROW #)
6212 1830     00930         JR      CUPCK
              00940 ;
```

```
6214 FE80     00950 CURT1   CP      128             ;HALF WAY ?
6216 30F1     00960         JR      NC,CULF2        ;JUMP IF OVER
              00970 ;
6218 DDCB034E 00980 CURT2   BIT     1,(IX+3)        ;ALREADY MOVING RIGHT ?
621C 2005     00990         JR      NZ,CUDCK        ;JUMP IF NO
              01000 ;
621E DD3401   01010 CURT3   INC     (IX+1)          ;MOVE RIGHT (ROW #)
6221 1821     01020         JR      CUPCK
              01030 ;
6223 DD3502   01040 CUDCK   DEC     (IX+2)          ;CHANGE DIRECTION YET ?
6226 2808     01050         JR      Z,CUDIR         ;JUMP IF YES
6228 DDCB034E 01060         BIT     1,(IX+3)        ;MOVING RIGHT ?
622C 28F0     01070         JR      Z,CURT3         ;JUMP IF YES
622E 18DF     01080         JR      CULF3           ;MOVE LEFT
              01090 ;
6230 ED5F     01100 CUDIR   LD      A,R             ;RESET CHANGE DIRECTION
              01110                                 ;DELAY (RANDOM)
6232 3C       01120         INC     A
6233 DD7702   01130         LD      (IX+2),A
              01140 ;
6236 DD7E03   01150         LD      A,(IX+3)        ;GET FLAGS
6239 EE02     01160         XOR     2               ;TOGGLE LEFT/RIGHT
623B DD7703   01170         LD      (IX+3),A
623E CB4F     01180         BIT     1,A
6240 28DC     01190         JR      Z,CURT3         ;MOVE RIGHT
6242 18CB     01200         JR      CULF3           ;MOVE LEFT
              01210 ;
6244 DDCB039E 01220 CUPCK   RES     3,(IX+3)        ;RESET UP/DOWN CANCEL
6248 DD7E03   01230         LD      A,(IX+3)
624B EE20     01240         XOR     20H             ;TOGGLE MULTIPLYER
624D DD7703   01250         LD      (IX+3),A
6250 CB6F     01260         BIT     5,A             ;MOVE UP/DOWN ?
6252 C0       01270         RET     NZ              ;RETURN IF NO
              01280 ;
6253 3A8962   01290         LD      A,(UDDLY)       ;GET INTER ACTIVE DELAY
              01300                                 ;COUNT
6256 3D       01310         DEC     A
6257 328962   01320         LD      (UDDLY),A
625A C0       01330         RET     NZ              ;RETURN IF NOT YET
              01340 ;
625B 3E05     01350         LD      A,5             ;RESET DELAY
625D 328962   01360         LD      (UDDLY),A
6260 DDCB03DE 01370         SET     3,(IX+3)        ;SET UP/DOWN MOVE
              01380 ;
6264 ED5F     01390         LD      A,R             ;RANDOM UP DOWN MOVE
6266 E60F     01400         AND     0FH
6268 FE08     01410         CP      8
626A 7B       01420         LD      A,E
626B 380E     01430         JR      C,CDWN1         ;JUMP IF MOVE DOWN
              01440 ;
626D 3D       01450 CUP1    DEC     A               ;MOVE UP
626E FE02     01460         CP      2               ;TOO FAR ?
6270 3808     01470         JR      C,CDWN2         ;JUMP IF YES
6272 DDCB0386 01480         RES     0,(IX+3)        ;FLAG UP
6276 DD7700   01490 CUPEXT  LD      (IX),A
6279 C9       01500         RET
```

```
              01510 ;
627A 3C       01520 CDWN2   INC     A               ;CANCEL UP MOVE
627B 3C       01530 CDWN1   INC     A               ;MOVE DOWN
627C FE10     01540         CP      16              ;TOO FAR ?
627E 3006     01550         JR      NC,CUP2         ;JUMP IF YES
6280 DDCB03C6 01560         SET     0,(IX+3)        ;FLAG DOWN
6284 18F0     01570         JR      CUPEXT
              01580 ;
6286 3D       01590 CUP2    DEC     A               ;CANCEL DOWN
6287 18E4     01600         JR      CUP1
              01610 ;
6289 05       01620 UDDLY   DB      5
              01630 ;
628A          01910 *GET    BOMBER/ASM              ;BOMBER/BOMBS ROUTINES
              00010 ;*************************************************
              00020 ;**             BOMBER                          **
              00030 ;**     CHARACTER = 240                         **
              00040 ;**     IX   = ROW # (GRID)                     **
              00050 ;**     IX+1 = COLUMN # (GRID)                  **
              00060 ;**     IX+2 = DELAY BEFORE MOVE                **
              00070 ;**     IX+3 = FLAGS -- BIT 0 = MOVE UP/DOWN    **
              00080 ;**                         1 = DROP SWITCH     **
              00090 ;**                         2 = ON SCREEN       **
              00100 ;**                         3 = UP/DOWN DELAY   **
              00110 ;**                         4 = INIT FLAG       **
              00120 ;**                         5 = BOMB DROP FLAG  **
              00130 ;**                         6 = WARP ON         **
              00140 ;**                         7 = ACTIVE/INACTIVE **
              00150 ;**     IX+4 = PHASE COUNTER FOR WARP IN        **
              00160 ;**          = DELAY BEFORE DROPPING BOMBS      **
              00170 ;**          = BOMB COUNT DURING DROP           **
              00180 ;**     IX+5 = UNUSED                           **
              00190 ;*************************************************
              00200 ;
628A DD21FE74 00210 BOMBER  LD      IX,BMRTBL       ;BOMBER TABLE
628E DD22CE5B 00220         LD      (TBLPAS),IX
6292 DD7E00   00230         LD      A,(IX)
6295 B7       00240         OR      A               ;ANY ACTIVE ?
6296 C8       00250         RET     Z               ;RETURN IF NO
6297 47       00260         LD      B,A             ;GET COUNT
6298 DD23     00270         INC     IX
629A 110000   00280         LD      DE,0            ;CLEAR ENTRY OFFSET
              00290 ;
629D DD19     00300 BMRLOP  ADD     IX,DE           ;NEXT ENTRY
629F DDCB037E 00310         BIT     7,(IX+3)        ;ALIVE ?
62A3 2806     00320         JR      Z,CSKP11        ;JUMP IF YES
62A5 CD7F5B   00330         CALL    CRUNCH          ;MOVE 'EM BACK ONE
62A8 10F3     00340         DJNZ    BMRLOP
62AA C9       00350         RET
              00360 ;
62AB C5       00370 CSKP11  PUSH    BC              ;SAVE COUNT
62AC CDD75B   00380         CALL    ONCHK
62AF DDCB0376 00390         BIT     6,(IX+3)        ;WARP IN ?
62B3 2811     00400         JR      Z,WRSK4         ;JUMP IF NO
              00410 ;
62B5 DDCB0356 00420         BIT     2,(IX+3)        ;ON SCREEN ?
```

```
62B9 2006     00430        JR      NZ,WRSK5        ;JUMP IF YES
              00440 ;
62BB DDCB03B6 00450        RES     6,(IX+3)        ;CANCEL WARP IN
62BF 1805     00460        JR      WRSK4
              00470 ;
62C1 CDE75C   00480 WRSK5  CALL    WARPIN
62C4 1815     00490        JR      BMREXT
              00500 ;
62C6 DDCB0356 00510 WRSK4  BIT     2,(IX+3)        ;ON SCREEN ?
62CA 2816     00520        JR      Z,BMREX8        ;JUMP IF NO
              00530 ;
62CC DDCB0366 00540        BIT     4,(IX+3)        ;INIT ?
62D0 C48263   00550        CALL    NZ,BRINIT       ;CALL IF YES
62D3 CDD75C   00560        CALL    MUCLR           ;CLEAR OLD POS.
62D6 CDE762   00570        CALL    BMRMOV          ;DO MOVE
62D9 36F0     00580        LD      (HL),240        ;DRAW BOMBER
              00590 ;
62DB C1       00600 BMREXT POP     BC              ;COUNT
62DC 110600   00610        LD      DE,6            ;ENTRY OFFSET
62DF 10BC     00620        DJNZ    BMRLOP
62E1 C9       00630        RET
              00640 ;
62E2 CDE762   00650 BMREX8 CALL    BMRMOV          ;MOVE OFF SCREEN
62E5 18F4     00660        JR      BMREXT
              00670 ;
62E7 DD3502   00680 BMRMOV DEC     (IX+2)          ;MOVE YET ?
62EA C0       00690        RET     NZ              ;RETURN IF NO
              00700 ;
62EB DD36020F 00710        LD      (IX+2),15       ;RESET DELAY
62EF DD3501   00720        DEC     (IX+1)          ;MOVE LEFT
62F2 2B       00730        DEC     HL              ;DO IT ON SCREEN
62F3 DD7E03   00740        LD      A,(IX+3)
62F6 EE08     00750        XOR     08
62F8 DD7703   00760        LD      (IX+3),A
62FB CB5F     00770        BIT     3,A             ;MOVE UP/DOWN ?
62FD 282E     00780        JR      Z,BUDSKP        ;JUMP IF NO
              00790 ;
62FF DD7E00   00800        LD      A,(IX)          ;GET ROW
6302 DDCB0346 00810        BIT     0,(IX+3)        ;MOVE UP ?
6306 200A     00820        JR      NZ,BRDWN1       ;JUMP IF NO
              00830 ;
6308 3D       00840 BRUP1  DEC     A               ;MOVE UP
6309 FE02     00850        CP      2               ;TOO FAR ?
630B 3011     00860        JR      NC,BROK1        ;JUMP IF NO
              00870 ;
630D DDCB03C6 00880        SET     0,(IX+3)        ;SET MOVE DOWN
6311 3C       00890        INC     A               ;CANCEL UP MOVE
6312 3C       00900 BRDWN1 INC     A               ;MOVE DOWN
6313 FE0C     00910        CP      12              ;TOO FAR
6315 3807     00920        JR      C,BROK1         ;JUMP IF NO
              00930 ;
6317 DDCB0386 00940        RES     0,(IX+3)        ;FLAG MOVE UP
631B 3D       00950        DEC     A               ;CANCEL DOWN MOVE
631C 18EA     00960        JR      BRUP1
              00970 ;
631E DD7700   00980 BROK1  LD      (IX),A          ;SAVE NEW ROW
```

```
6321 DDCB0346 00990          BIT     0,(IX+3)        ;UP OR DOWN ?
6325 F5       01000          PUSH    AF              ;SAVE FLAGS
6326 CC4B5D   01010          CALL    Z,SB64          ;CALL IF UP
6329 F1       01020          POP     AF
632A C4445D   01030          CALL    NZ,AD64         ;CALL IF DOWN
632D DD7E03   01040 BUDSKP   LD      A,(IX+3)        ;GET FLAGS
6330 EE02     01050          XOR     2               ;TOGGLE DROP SWITCH
6332 DD7703   01060          LD      (IX+3),A
              01070 ;
6335 CB6F     01080          BIT     5,A             ;DROPPING BOMBS ALREADY ?
6337 2011     01090          JR      NZ,DRPIT        ;JUMP IF YES
              01100 ;
6339 DD3504   01110          DEC     (IX+4)          ;START DROPPING BOMBS ?
633C C0       01120          RET     NZ              ;RETURN IF NOT YET
633D DDCB03EE 01130          SET     5,(IX+3)        ;SET BOMBS IN PROGRESS
6341 ED5F     01140          LD      A,R
6343 E607     01150          AND     7
6345 3C       01160          INC     A
6346 DD7704   01170          LD      (IX+4),A        ;RANDOM BOMB COUNT
6349 C9       01180          RET
              01190 ;
634A DD7E04   01200 DRPIT    LD      A,(IX+4)        ;GET BOMB COUNTER
634D B7       01210          OR      A               ;MORE TO DROP ?
634E 2009     01220          JR      NZ,DRPIT2       ;JUMP IF YES
              01230 ;
6350 DD36042D 01240 BMCAN    LD      (IX+4),45       ;RESET DELAY BEFORE DROP
6354 DDCB03AE 01250          RES     5,(IX+3)        ;CANCEL DROP
6358 C9       01260          RET
              01270 ;
6359 DDCB034E 01280 DRPIT2   BIT     1,(IX+3)        ;DROP YET ?
635D C8       01290          RET     Z               ;RETURN IF NO
              01300 ;
635E 3D       01310          DEC     A               ;UPDATE BOMB COUNT
635F DD7704   01320          LD      (IX+4),A
              01330 ;
6362 DD5E00   01340          LD      E,(IX)          ;GET ROW #
6365 DD7E01   01350          LD      A,(IX+1)        ;GET COLUMN #
6368 3C       01360          INC     A
6369 57       01370          LD      D,A
636A DDE5     01380          PUSH    IX
636C DD211E75 01390          LD      IX,BMBTBL       ;BOMB TABLE
6370 CD365B   01400          CALL    EMPTY
6373 3008     01410          JR      NC,DRPNO        ;JUMP IF NO EMPTY FOUND
              01420 ;
6375 DD3602FF 01430          LD      (IX+2),255      ;ALIVE COUNT
6379 DD360300 01440          LD      (IX+3),0        ;CLEAR FLAGS
637D DDE1     01450 DRPNO    POP     IX
637F 30CF     01460          JR      NC,BMCAN
6381 C9       01470          RET
              01480 ;
6382 DDCB03A6 01490 BRINIT   RES     4,(IX+3)
6386 DD360432 01500          LD      (IX+4),50       ;DELAY BEFORE DROP
638A C9       01510          RET
              01520 ;
              01530 ;**********************************************
              01540 ;**               BOMB ROUTINES            **
```

```
               01550 ;**      CHARACTER = 214                             **
               01560 ;**      IX   = ROW # (GRID)                         **
               01570 ;**      IX+1 = COLUMN # (GRID)                      **
               01580 ;**      IX+2 = ALIVE COUNT                          **
               01590 ;**      IX+3 = FLAGS -- BIT 0 = UNUSED              **
               01600 ;**                          1 = UNUSED              **
               01610 ;**                          2 = ON SCREEN           **
               01620 ;**                          3 = UNUSED              **
               01630 ;**                          4 = UNUSED              **
               01640 ;**                          5 = UNUSED              **
               01650 ;**                          6 = UNUSED              **
               01660 ;**                          7 = ACTIVE/INACTIVE     **
               01670 ;**      IX+4 = UNUSED                               **
               01680 ;**      IX+5 = UNUSED                               **
               01690 ;****************************************************
               01700 ;
638B DD211E75  01710 BOMB     LD      IX,BMBTBL       ;BOMB TABLE
638F DD22CE5B  01720          LD      (TBLPAS),IX
6393 DD7E00    01730          LD      A,(IX)  ;ACTIVE COUNT
6396 B7        01740          OR      A
6397 C8        01750          RET     Z               ;RETURN IF NONE
6398 47        01760          LD      B,A             ;COUNTER
6399 DD23      01770          INC     IX
639B 110000    01780          LD      DE.0            ;CLEAR ENTRY OFFSET
               01790 ;
639E DD19      01800 BMBLOP   ADD     IX,DE           ;NEXT ENTRY
63A0 DDCB037E  01810          BIT     7,(IX+3)        ;ACTIVE ?
63A4 2806      01820          JR      Z,CSKP10        ;JUMP IF YES
               01830 ;
63A6 CD7F5B    01840          CALL    CRUNCH          ;BACK IT UP
63A9 10F3      01850          DJNZ    BMBLOP
63AB C9        01860          RET
               01870 ;
63AC C5        01880 CSKP10   PUSH    BC              ;COUNT
63AD CDD75B    01890          CALL    ONCHK
63B0 DDCB0356  01900          BIT     2,(IX+3)        ;ON SCREEN ?
63B4 280A      01910          JR      Z,BMBEXT        ;JUMP IF NO
               01920 ;
63B6 CDD75C    01930          CALL    MUCLR           ;CLEAR OLD POS.
63B9 DD3502    01940          DEC     (IX+2)          ;CANCEL BOMB YET ?
63BC 2809      01950          JR      Z,BMBKIL        ;JUMP IF YES
               01960 ;
63BE 36D6      01970          LD      (HL),214        ;DRAW BOMB
               01980 ;
63C0 C1        01990 BMBEXT   POP     BC              ;COUNT
63C1 110600    02000          LD      DE,6            ;ENTRY OFFSET
63C4 10D8      02010          DJNZ    BMBLOP
63C6 C9        02020          RET
               02030 ;
63C7 DDCB03FE  02040 BMBKIL   SET     7,(IX+3)        ;KILL OFF BOMB
63CB 18F3      02050          JR      BMBEXT
               02060 ;
63CD           01920 *GET     POD/ASM                 ;POD/SWARMER ROUTINES
               00010 ;****************************************************
               00020 ;**               POD ROUTINES                     **
               00030 ;**      CHARACTER = 194                             **
               00040 ;**      IX   = ROW # (GRID)                           **
               00050 ;**      IX+1 = COLUMN # (GRID)                        **
               00060 ;**      IX+2 = DELAY BEFORE MOVE                      **
               00070 ;**      IX+3 = FLAGS -- BIT 0 = MOVE UP               **
               00080 ;**                          1 = MOVE DOWN             **
               00090 ;**                          2 = ON SCREEN             **
               00100 ;**                          3 = RIGHT/LEFT            **
               00110 ;**                          4 = UNUSED                **
               00120 ;**                          5 = INIT FLAG             **
               00130 ;**                          6 = WARP-IN               **
               00140 ;**                          7 = ACTIVE/INACTIVE **
               00150 ;**      IX+4 = PHASE COUNT FOR WARP IN                **
               00160 ;**           = DIRECTION COUNTER                      **
               00170 ;**      IX+5 = SHOT DELAY                             **
               00180 ;*******************************************************
               00190 ;
63CD DD21C975 00200 POD     LD      IX,PODTBL       ;POD TABLE
63D1 DD22CE5B 00210         LD      (TBLPAS),IX
63D5 DD7E00   00220         LD      A,(IX)          ;ACTIVE COUNT
63D8 B7       00230         OR      A               ;ANY THERE ?
63D9 C8       00240         RET     Z               ;RETURN IF NO
63DA 47       00250         LD      B,A
63DB DD23     00260         INC     IX
63DD 110000   00270         LD      DE,0            ;CLEAR ENTRY OFFSET
              00280 ;
63E0 DD19     00290 PODLOP  ADD     IX,DE           ;NEXT ENTRY
63E2 DDCB037E 00300         BIT     7,(IX+3)        ;ACTIVE ENTRY ?
63E6 2809     00310         JR      Z,PODEX1        ;JUMP IF YES
              00320 ;
63E8 CDD564   00330         CALL    SWMON           ;PUT ON SWARMERS
63EB CD7F5B   00340         CALL    CRUNCH          ;MOVE IT BACK
63EE 10F0     00350         DJNZ    PODLOP
63F0 C9       00360         RET
              00370 ;
63F1 C5       00380 PODEX1  PUSH    BC              ;SAVE COUNT
63F2 CDD75B   00390         CALL    ONCHK
63F5 DDCB0376 00400         BIT     6,(IX+3)        ;WARP-IN ?
63F9 2811     00410         JR      Z,POSK4         ;JUMP IF NO
              00420 ;
63FB DDCB0356 00430         BIT     2,(IX+3)        ;ON SCREEN ?
63FF 2006     00440         JR      NZ,POSK5        ;JUMP IF YES
              00450 ;
6401 DDCB03B6 00460         RES     6,(IX+3)        ;RESET WARP-IN
6405 1805     00470         JR      POSK4
              00480 ;
6407 CDE75C   00490 POSK5   CALL    WARPIN
640A 1835     00500         JR      PODEXT
              00510 ;
640C DDCB036E 00520 POSK4   BIT     5,(IX+3)        ;INITIALIZE ?
6410 C46864   00530         CALL    NZ,PDINIT       ;CALL IF YES
6413 DDCB0356 00540         BIT     2,(IX+3)        ;ON SCREEN ?
6417 2836     00550         JR      Z,PODEX8        ;JUMP IF NO
              00560 ;
6419 CDD75C   00570         CALL    MUCLR           ;CLEAR OLD CHARACTER
641C DD3502   00580         DEC     (IX+2)          ;MOVE YET ?
641F 201E     00590         JR      NZ,PODRW        ;JUMP IF NO
```

```
                00600 ;
6421 DD360208 00610          LD      (IX+2),8          ;RESET DELAY
6425 CD8064   00620          CALL    PODMOV            ;MOVE POD
              00630 ;
6428 DDCB0346 00640          BIT     0,(IX+3)          ;MOVE UP ?
642C C44B5D   00650          CALL    NZ,SB64           ;CALL IF YES
642F DDCB034E 00660          BIT     1,(IX+3)          ;MOVE DOWN ?
6433 C4445D   00670          CALL    NZ,AD64           ;CALL IF YES
              00680 ;
6436 23       00690          INC     HL                ;MOVE RIGHT
6437 DDCB035E 00700          BIT     3,(IX+3)          ;MOVE RIGHT ?
643B 2802     00710          JR      Z,PODRW           ;JUMP IF YES
              00720 ;
643D 2B       00730          DEC     HL                ;CANCEL RIGHT MOVE
643E 2B       00740          DEC     HL                ;MOVE LEFT
643F 36C2     00750 PODRW    LD      (HL),194          ;DRAW POD
              00760 ;
6441 DDCB0356 00770 PODEXT   BIT     2,(IX+3)          ;ON SCREEN ?
6445 C45D64   00780          CALL    NZ,PDSHOT         ;DO SHOT IF YES
6448 C1       00790          POP     BC                ;COUNT
6449 110600   00800          LD      DE,6              ;ENTRY OFFSET
644C 1092     00810          DJNZ    PODLOP
644E C9       00820          RET
              00830 ;
644F DD3502   00840 PODEX8   DEC     (IX+2)            ;MOVE YET ?
6452 20ED     00850          JR      NZ,PODEXT         ;JUMP IF NO
6454 DD360208 00860          LD      (IX+2),8          ;RESET DELAY
6458 CD8064   00870          CALL    PODMOV
645B 18E4     00880          JR      PODEXT
              00890 ;
645D DD3505   00900 PDSHOT   DEC     (IX+5)            ;SHOOT YET ?
6460 C0       00910          RET     NZ                ;RETURN IF NO
              00920 ;
6461 DD360514 00930          LD      (IX+5),20         ;RESET COUNTER
6465 C31D5C   00940          JP      SHOCHK
              00950 ;
6468 DDCB03AE 00960 PDINIT   RES     5,(IX+3)          ;RESET INIT FLAG
646C DD360480 00970          LD      (IX+4),80H        ;DIRECTION DELAY
6470 DDCB039E 00980          RES     3,(IX+3)          ;FLAG RIGHT
6474 ED5F     00990          LD      A,R
6476 E60F     01000          AND     0FH
6478 FE08     01010          CP      8
647A D0       01020          RET     NC
647B DDCB03DE 01030          SET     3,(IX+3)
647F C9       01040          RET
              01050 ;
6480 DD7E03   01060 PODMOV   LD      A,(IX+3)          ;GET FLAGS
6483 E6FC     01070          AND     0FCH              ;CLEAR UP/DOWN
6485 DD7703   01080          LD      (IX+3),A
              01090 ;
6488 ED5F     01100          LD      A,R               ;RANDOM UP/DOWN
648A E60F     01110          AND     0FH
648C FE05     01120          CP      5                 ;MOVE UP/DOWN ?
648E 3821     01130          JR      C,PUDEXT          ;JUMP IF NO
              01140 ;
6490 FE0A     01150          CP      10                ;MOVE UP ?
```

```
6492 DD7E00   01160          LD      A,(IX)          ;GET ROW
6495 300E     01170          JR      NC,PDWN         ;JUMP IF NO
              01180 ;
6497 3D       01190          DEC     A               ;MOVE UP
6498 FE02     01200          CP      2               ;TOO FAR
649A 3815     01210          JR      C,PUDEXT        ;JUMP IF YES
              01220 ;
649C DD7700   01230          LD      (IX),A          ;NEW ROW
649F DDCB03C6 01240          SET     0,(IX+3)        ;SET UP
64A3 180C     01250          JR      PUDEXT
              01260 ;
64A5 3C       01270 PDWN     INC     A               ;MOVE DOWN
64A6 FE10     01280          CP      16              ;TOO FAR ?
64A8 3007     01290          JR      NC,PUDEXT       ;JUMP IF YES
              01300 ;
64AA DD7700   01310          LD      (IX),A          ;NEW ROW
64AD DDCB03CE 01320          SET     1,(IX+3)        ;SET DOWN
              01330 ;
64B1 DD3504   01340 PUDEXT   DEC     (IX+4)          ;CHANGE DIRECTION ?
64B4 2011     01350          JR      NZ,PLREX1       ;JUMP IF NO
              01360 ;
64B6 ED5F     01370          LD      A,R             ;GET NEW DELAY VALUE
64B8 4F       01380          LD      C,A
64B9 ED5F     01390          LD      A,R
64BB 81       01400          ADD     A,C
64BC DD7704   01410          LD      (IX+4),A
64BF DD7E03   01420          LD      A,(IX+3)        ;GET FLAGS
64C2 EE08     01430          XOR     8               ;TOGGLE LEFT/RIGHT
64C4 DD7703   01440          LD      (IX+3),A
              01450 ;
64C7 DDCB035E 01460 PLREX1   BIT     3,(IX+3)        ;RIGHT OR LEFT ?
64CB 2004     01470          JR      NZ,PLFT         ;JUMP IF LEFT
              01480 ;
64CD DD3401   01490          INC     (IX+1)          ;MOVE RIGHT
64D0 C9       01500          RET
              01510 ;
64D1 DD3501   01520 PLFT     DEC     (IX+1)          ;MOVE LEFT
64D4 C9       01530          RET
              01540 ;
              01550 ;        *****   BRING ON SWARMERS   *****
              01560 ;
64D5 DDE5     01570 SWMON    PUSH    IX              ;SAVE POD TABLE POINTER
64D7 CDDD64   01580          CALL    SWMON2
64DA DDE1     01590 SWMNEX   POP     IX
64DC C9       01600          RET
              01610 ;
64DD DD5E00   01620 SWMON2   LD      E,(IX)          ;GET ROW/COLUMN
64E0 DD5601   01630          LD      D,(IX+1)
64E3 DD21DD75 01640          LD      IX,SRMTBL       ;SWARMER TABLE
64E7 DDE5     01650          PUSH    IX
64E9 CD365B   01660          CALL    EMPTY           ;FIND EMPTY SPOT
64EC 30EC     01670          JR      NC,SWMNEX       ;JUMP IF NONE
              01680 ;
64EE DD360311 01690          LD      (IX+3),11H      ;MOVE UP/LEFT
64F2 DD360209 01700          LD      (IX+2),9        ;SEEK DELAY
64F6 DDE1     01710          POP     IX
```

```
64F8 DDE5     01720          PUSH    IX
64FA CD365B   01730          CALL    EMPTY           ;FIND EMPTY SPOT
64FD 30DB     01740          JR      NC,SWMNEX       ;JUMP IF NONE
              01750 ;
64FF DD360312 01760          LD      (IX+3),12H      ;MOVE DOWN/LEFT
6503 DD360208 01770          LD      (IX+2),8        ;SEEK DELAY
6507 DDE1     01780          POP     IX
6509 DDE5     01790          PUSH    IX
650B CD365B   01800          CALL    EMPTY           ;FIND EMPTY SPOT
650E 30CA     01810          JR      NC,SWMNEX       ;JUMP IF NONE
              01820 ;
6510 DD360301 01830          LD      (IX+3),1        ;MOVE UP
6514 DD36020A 01840          LD      (IX+2),10       ;SEEK DELAY
6518 DDE1     01850          POP     IX
651A DDE5     01860          PUSH    IX
651C CD365B   01870          CALL    EMPTY           ;FIND EMPTY SPOT
651F 30B9     01880          JR      NC,SWMNEX       ;JUMP IF NONE
              01890 ;
6521 DD360302 01900          LD      (IX+3),2        ;MOVE DOWN
6525 DD36020A 01910          LD      (IX+2),10       ;SEEK DELAY
6529 DDE1     01920          POP     IX
652B DDE5     01930          PUSH    IX
652D CD365B   01940          CALL    EMPTY           ;FIND EMPTY SPOT
6530 30A8     01950          JR      NC,SWMNEX       ;JUMP IF NONE
              01960 ;
6532 DD360309 01970          LD      (IX+3),9        ;MOVE UP/RIGHT
6536 DD360208 01980          LD      (IX+2),8        ;SEEK DELAY
653A DDE1     01990          POP     IX
653C CD365B   02000          CALL    EMPTY           ;FIND EMPTY SPOT
653F D0       02010          RET     NC              ;RETURN IF NONE
              02020 ;
6540 DD36030A 02030          LD      (IX+3),0AH      ;MOVE DOWN/RIGHT
6544 DD360209 02040          LD      (IX+2),9        ;SEEK DELAY
6548 C9       02050          RET
              02060 ;
              02070 ;**************************************************
              02080 ;**              SWARMER ROUTINES                **
              02090 ;**      CHARACTER = 229                         **
              02100 ;**      IX   = ROW # (GRID)                     **
              02110 ;**      IX+1 = COLUMN # (GRID)                  **
              02120 ;**      IX+2 = DELAY BEFORE SEEK                **
              02130 ;**      IX+3 = FLAGS -- BIT 0 = MOVE UP         **
              02140 ;**                          1 = MOVE DOWN       **
              02150 ;**                          2 = ON SCREEN       **
              02160 ;**                          3 = MOVE RIGHT      **
              02170 ;**                          4 = MOVE LEFT       **
              02180 ;**                          5 = SEEK            **
              02190 ;**                          6 = UP/DOWN DELAY   **
              02200 ;**                          7 = ACTIVE/INACTIVE **
              02210 ;**      IX+4 = UNUSED                           **
              02220 ;**      IX+5 = UNUSED                           **
              02230 ;**************************************************
              02240 ;
6549 DD21DD75 02250 SWARM    LD      IX,SRMTBL       ;SWARMER TABLE
654D DD22CE5B 02260          LD      (TBLPAS),IX
6551 DD7E00   02270          LD      A,(IX)
```

```
6554 B7         02280          OR      A               ;ANY ACTIVE ?
6555 C8         02290          RET     Z               ;RETURN IF NO
6556 DD23       02300          INC     IX
6558 47         02310          LD      B,A
6559 110000     02320          LD      DE,0            ;CLEAR ENTRY OFFSET
                02330 ;
655C DD19       02340 SWMLOP   ADD     IX,DE           ;NEXT ENTRY
655E DDCB037E   02350          BIT     7,(IX+3)        ;ACTIVE ?
6562 2806       02360          JR      Z,CNSKP8        ;JUMP IF YES
6564 CD7F5B     02370          CALL    CRUNCH          ;MOVE IT BACK
6567 10F3       02380          DJNZ    SWMLOP
6569 C9         02390          RET
                02400 ;
656A C5         02410 CNSKP8   PUSH    BC              ;SAVE COUNT
656B CDD75B     02420          CALL    ONCHK
656E DDCB0356   02430          BIT     2,(IX+3)        ;ON SCREEN ?
6572 282C       02440          JR      Z,SWMEX8        ;JUMP IF NO
                02450 ;
6574 CDD75C     02460          CALL    MUCLR           ;CLEAR OLD CHARACTER
6577 CDA565     02470          CALL    SWMMOV          ;MOVE SWARMER
657A DDCB0346   02480          BIT     0,(IX+3)        ;MOVE UP ?
657E C44B5D     02490          CALL    NZ,SB64         ;CALL IF YES
6581 DDCB034E   02500          BIT     1,(IX+3)        ;MOVE DOWN ?
6585 C4445D     02510          CALL    NZ,AD64         ;CALL IF YES
6588 23         02520          INC     HL              ;MOVE RIGHT
6589 DDCB035E   02530          BIT     3,(IX+3)        ;MOVE RIGHT ?
658D 2008       02540          JR      NZ,SWMDRW       ;JUMP IF YES
                02550 ;
658F 2B         02560          DEC     HL              ;CANCEL MOVE RIGHT
6590 DDCB0366   02570          BIT     4,(IX+3)        ;MOVE LEFT ?
6594 2801       02580          JR      Z,SWMDRW        ;JUMP IF NO
                02590 ;
6596 2B         02600          DEC     HL              ;MOVE LEFT
6597 36E5       02610 SWMDRW   LD      (HL),229        ;DRAW SWARMER
                02620 ;
6599 C1         02630 SWMEXT   POP     BC              ;COUNT
659A 110600     02640          LD      DE,6            ;ENTRY OFFSET
659D 10BD       02650          DJNZ    SWMLOP
659F C9         02660          RET
                02670 ;
65A0 CDA565     02680 SWMEX8   CALL    SWMMOV
65A3 18F4       02690          JR      SWMEXT
                02700 ;
65A5 DDCB036E   02710 SWMMOV   BIT     5,(IX+3)        ;SEEK ?
65A9 286F       02720          JR      Z,SWEKSK        ;JUMP IF NO
                02730 ;
65AB DD7E03     02740          LD      A,(IX+3)        ;GET FLAGS
65AE E6E4       02750          AND     0E4H            ;CANCEL ALL MOTION
65B0 DD7703     02760          LD      (IX+3),A
65B3 3AA273     02770          LD      A,(SHCOL)       ;GET SHIP COLUMN
65B6 DD9601     02780          SUB     (IX+1)          ;SAME ?
65B9 281B       02790          JR      Z,SWMUD         ;JUMP IF YES
                02800 ;
65BB 300E       02810          JR      NC,SWRT1        ;JUMP IF POSSIBLE MOVE RIGHT
65BD 2F         02820          CPL
65BE FE80       02830          CP      128
```

```
65C0 300D     02840          JR     NC,SWRT2        ;JUMP IF MOVE RIGHT
              02850 ;
65C2 DD3501   02860 SWLF3    DEC    (IX+1)          ;MOVE LEFT
65C5 DDCB03E6 02870          SET    4,(IX+3)        ;SET MOVE LEFT
65C9 180B     02880          JR     SWMUD
              02890 ;
65CB FE80     02900 SWRT1    CP     128
65CD 30F3     02910          JR     NC,SWLF3        ;JUMP IF MOVE LEFT
65CF DD3401   02920 SWRT2    INC    (IX+1)          ;MOVE RIGHT
65D2 DDCB03DE 02930          SET    3,(IX+3)        ;SET MOVE RIGHT
              02940 ;
65D6 DD7E03   02950 SWMUD    LD     A,(IX+3)        ;GET FLAGS
65D9 EE40     02960          XOR    40H             ;TOGGLE UP DOWN DELAY
65DB DD7703   02970          LD     (IX+3),A
65DE CB77     02980          BIT    6,A             ;MOVE UP/DOWN ?
65E0 C0       02990          RET    NZ              ;RETURN IF NO
              03000 ;
65E1 3AA173   03010          LD     A,(SHROW)       ;GET SHIP ROW
65E4 5F       03020          LD     E,A
65E5 DD7E00   03030          LD     A,(IX)
65E8 BB       03040          CP     E               ;SAME ROW ?
65E9 200F     03050          JR     NZ,SWXYZ        ;JUMP IF NO
              03060 ;
65EB 3A5D66   03070          LD     A,(FOOL)        ;KEEP SWARMERS JUMPING
65EE EE01     03080          XOR    1               ;AROUND TO MAKE THEM
65F0 325D66   03090          LD     (FOOL),A        ;HARDER TO HIT.
65F3 DD7E00   03100          LD     A,(IX)          ;GET ROW
65F6 2015     03110          JR     NZ,SWUP         ;JUMP IF MOVE UP
65F8 1802     03120          JR     SWDN
              03130 ;
65FA 3011     03140 SWXYZ    JR     NC,SWUP         ;JUMP IF MOVE UP
65FC 3C       03150 SWDN     INC    A               ;MOVE DOWN
65FD FE10     03160          CP     16              ;TOO FAR ?
65FF 300B     03170          JR     NC,SWUP1        ;JUMP IF YES
              03180 ;
6601 DD7700   03190          LD     (IX),A          ;SAVE NEW ROW
6604 DDCB03CE 03200          SET    1,(IX+3)        ;SET MOVE DOWN
6608 C9       03210          RET
              03220 ;
6609 3C       03230 SWDN1    INC    A               ;CANCEL UP
660A 18F0     03240          JR     SWDN
              03250 ;
660C 3D       03260 SWUP1    DEC    A               ;CANCEL DOWN
660D 3D       03270 SWUP     DEC    A               ;MOVE UP
660E FE02     03280          CP     2               ;TOO FAR ?
6610 38F7     03290          JR     C,SWDN1         ;JUMP IF YES
              03300 ;
6612 DD7700   03310          LD     (IX),A          ;SAVE NEW ROW
6615 DDCB03C6 03320          SET    0,(IX+3)        ;SET MOVE UP
6619 C9       03330          RET
              03340 ;
661A DD3502   03350 SWEKSK   DEC    (IX+2)          ;SEEK YET ?
661D 2004     03360          JR     NZ,SWSKP2       ;JUMP IF NO
661F DDCB03EE 03370          SET    5,(IX+3)        ;SET SEEK
              03380 ;
6623 DDCB035E 03390 SWSKP2   BIT    3,(IX+3)        ;MOVE RIGHT ?
```

```
6627 2805      03400          JR      Z,SWNXT1        ;JUMP IF NO
6629 DD3401    03410          INC     (IX+1)          ;MOVE RIGHT
662C 1809      03420          JR      SWNXT2
               03430 ;
662E DDCB0366  03440 SWNXT1   BIT     4,(IX+3)        ;MOVE LEFT ?
6632 2803      03450          JR      Z,SWNXT2        ;JUMP IF NO
6634 DD3501    03460          DEC     (IX+1)          ;MOVE LEFT
6637 DD7E00    03470 SWNXT2   LD      A,(IX)          ;GET ROW
663A DDCB0346  03480          BIT     0,(IX+3)        ;MOVE UP ?
663E 280A      03490          JR      Z,SWNXT3        ;JUMP IF NO
6640 3D        03500          DEC     A
6641 FE02      03510          CP      2               ;TOO FAR ?
6643 3014      03520          JR      NC,SWNXT4       ;JUMP IF NO
6645 DDCB0386  03530          RES     0,(IX+3)        ;CANCEL UP
6649 C9        03540          RET
               03550 ;
664A DDCB034E  03560 SWNXT3   BIT     1,(IX+3)        ;MOVE DOWN ?
664E C8        03570          RET     Z               ;RETURN IF NO
664F 3C        03580          INC     A
6650 FE10      03590          CP      16              ;TOO FAR ?
6652 3805      03600          JR      C,SWNXT4        ;JUMP IF NO
6654 DDCB038E  03610          RES     1,(IX+3)        ;CANCEL DOWN
6658 C9        03620          RET
               03630 ;
6659 DD7700    03640 SWNXT4   LD      (IX),A          ;NEW ROW
665C C9        03650          RET
               03660 ;
665D 00        03670 FOOL     DB      0
               03680 ;
665E           01930 *GET     WHO/ASM                 ;SSM/ALIEN DEFENSE DATA & DISPLAY ROUTINES
               00010 ;*************************
               00020 ;**      WHO IT IS      **
               00030 ;*************************
               00040 ;
665E 21FA66    00050 WHO      LD      HL,SSM          ;SSM GRAPHICS DATA
6661 11003C    00060          LD      DE,3C00H        ;SCREEN POSITION
6664 0603      00070          LD      B,3             ;THREE BLOCKS OF DATA
6666 C5        00080 WHOLP    PUSH    BC
6667 CD816B    00090          CALL    PRINT3          ;PRINTS ROWS OF BLANKS
666A CDBE6B    00100          CALL    PRINT4          ;PRINTS GRAPHICS
666D C1        00110          POP     BC
666E 10F6      00120          DJNZ    WHOLP
               00130 ;
6670 CD816B    00140          CALL    PRINT3          ;ROWS OF BLANKS
               00150 ;
               00160          IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
               00170          LD      HL,SOFTD        ;TALK DATA
               00180          CALL    TALK            ;MAKE NOISE
               00190          ENDIF
               00200 ;
6673 0603      00210          LD      B,3
6675 CD146C    00220 SSMLOP   CALL    DELAY6          ;WAIT AWHILE
6678 10FB      00230          DJNZ    SSMLOP
               00240 ;
667A CDB266    00250          CALL    COMP2           ;COMPLEMENT SCREEN
667D 0605      00260          LD      B,5
```

```
667F CD146C     00270 SSLOP2  CALL     DELAY6           ;WAIT AWHILE
6682 10FB       00280         DJNZ     SSLOP2
6684 C9         00290         RET
                00300 ;
6685 214169     00310 ASK     LD       HL,ASKMES        ;"1 OR 2 PLAYERS ?"
6688 CDA666     00320         CALL     PRINT2
668B 13         00330         INC      DE
668C 13         00340         INC      DE
                00350 ;
668D 3A1038     00360 ASKLOP  LD       A,(3810H)        ;"1 OR 2"
6690 CB4F       00370         BIT      1,A              ;"1" ?
6692 2006       00380         JR       NZ,ASKONE        ;JUMP IF YES
6694 CB57       00390         BIT      2,A              ;"2" ?
6696 2002       00400         JR       NZ,ASKONE        ;JUMP IF YES
6698 18F3       00410         JR       ASKLOP
                00420 ;
669A CB3F       00430 ASKONE  SRL      A
669C 32C14C     00440         LD       (PLYCNT),A
669F C630       00450         ADD      A,30H
66A1 12         00460         LD       (DE),A
66A2 CD146C     00470         CALL     DELAY6
66A5 C9         00480         RET
                00490 ;
66A6 5E         00500 PRINT2  LD       E,(HL)           ;GET SCREEN ADDRESS
66A7 23         00510         INC      HL
66A8 56         00520         LD       D,(HL)
                00530 ;
66A9 23         00540         INC      HL
66AA 7E         00550 PR2LP   LD       A,(HL)           ;GET CHARACTER
66AB 23         00560         INC      HL
66AC B7         00570         OR       A                ;DONE ?
66AD C8         00580         RET      Z                ;RETURN IF YES
66AE 12         00590         LD       (DE),A           ;PUT ON SCREEN
66AF 13         00600         INC      DE
66B0 18F8       00610         JR       PR2LP            ;LOOP TIL DONE
                00620 ;
66B2 21003C     00630 COMP2   LD       HL,3C00H         ;SCREEN ADDRESS
66B5 014010     00640         LD       BC,1040H         ;B=ROWS C=COLUMNS
                00650 ;
66B8 C5         00660 COM2LP  PUSH     BC               ;SAVE ROW/COLUMN
66B9 E5         00670         PUSH     HL               ;SAVE SCREEN ADDRESS
66BA 41         00680         LD       B,C              ;GET COLUMN COUNT
                00690 ;
66BB 7E         00700 CM2LP   LD       A,(HL)           ;GET CHARACTER
66BC FE80       00710         CP       128              ;GRAPHICS ?
66BE 3807       00720         JR       C,CM2SK2         ;JUMP IF NO
66C0 FEC0       00730         CP       192              ;GRAPHICS ?
66C2 3003       00740         JR       NC,CM2SK2        ;JUMP IF NO
66C4 EE03       00750         XOR      3                ;COMPLEMENT TOP THIRD
66C6 77         00760         LD       (HL),A
66C7 23         00770 CM2SK2  INC      HL
66C8 10F1       00780         DJNZ     CM2LP
                00790 ;
66CA CD076C     00800         CALL     DELAY5           ;PAUSE
66CD E1         00810         POP      HL               ;SCREEN ADDRESS
66CE E5         00820         PUSH     HL
```

```
66CF 41        00830          LD      B,C               ;COLUMN COUNT
               00840 ;
66D0 7E        00850 CM3LP    LD      A,(HL)            ;GET CHARACTER
66D1 FE80      00860          CP      128               ;GRAPHICS ?
66D3 3807      00870          JR      C,CM3SK3          ;JUMP IF NO
66D5 FEC0      00880          CP      192               ;GRAPHICS ?
66D7 3003      00890          JR      NC,CM3SK3         ;JUMP IF NO
66D9 EE0C      00900          XOR     0CH               ;COMPLEMENT MIDDLE THIRD
66DB 77        00910          LD      (HL),A
66DC 23        00920 CM3SK3   INC     HL
66DD 10F1      00930          DJNZ    CM3LP
               00940 ;
66DF CD076C    00950          CALL    DELAY5            ;PAUSE
66E2 E1        00960          POP     HL
66E3 41        00970          LD      B,C               ;COLUMN COUNT
               00980 ;
66E4 7E        00990 CM4LP    LD      A,(HL)            ;GET CHARACTER
66E5 FE80      01000          CP      128               ;GRAPHICS ?
66E7 3807      01010          JR      C,CM4SK4          ;JUMP IF NO
66E9 FEC0      01020          CP      192               ;GRAPHICS ?
66EB 3003      01030          JR      NC,CM4SK4         ;JUMP IF NO
66ED EE30      01040          XOR     30H               ;COMPLEMENT BOTTOM THIRD
66EF 77        01050          LD      (HL),A
66F0 23        01060 CM4SK4   INC     HL
66F1 10F1      01070          DJNZ    CM4LP
               01080 ;
66F3 CD076C    01090          CALL    DELAY5            ;PAUSE
66F6 C1        01100          POP     BC                ;GET ROW/COLUMN COUNT
66F7 10BF      01110          DJNZ    COM2LP            ;LOOP FOR ROW COUNT
66F9 C9        01120          RET
               01130 ;
               01140 ;*********************************
               01150 ;**      SSM INC. GRAPHICS      **
               01160 ;*********************************
               01170 ;
66FA 03        01180 SSM      DB      3                 ;3 ROWS OF BLANKS
               01190 ;
66FB 07        01200          DB      7                 ;7 ROWS OF DATA
               01210 ;
66FC 80        01220          DB      128,128,128,128,128,128
6702 B8        01230          DB      184,191,191,191,191,191,191,191 ;S1
670A 80        01240          DB      128,128,128,128,128,128
6710 B8        01250          DB      184,191,191,191,191,191,191,191 ;S1
6718 80        01260          DB      128,128,128,128,128,128
671E BF        01270          DB      191,191,144,128,128,160,191,191 ;M1
6726 80        01280          DB      128,243,128,128,128,128
672C 80        01290          DB      128,128,128,128,128,128,128,128 ;INC1
6734 80        01300          DB      128,128,128,128,128,128,128,128 ;INC1
               01310 ;
673C 80        01320          DB      128,128,128,128,128,128
6742 BF        01330          DB      191,191,128,128,128,128,128,128 ;S2
674A 80        01340          DB      128,128,128,128,128,128
6750 BF        01350          DB      191,191,128,128,128,128,128,128 ;S2
6758 80        01360          DB      128,128,128,128,128,128
675E BF        01370          DB      191,191,175,180,184,159,191,191 ;M2
6766 80        01380          DB      128,128,128,128,128,128
```

```
676C 80         01390           DB      128,128,128,128,128,128,128,128 ;INC2
6774 80         01400           DB      128,128,128,128,128,128,128,128 ;INC2
                01410 ;
677C 80         01420           DB      128,128,128,128,128,128
6782 BF         01430           DB      191,191,128,128,128,128,128,128 ;S3
678A 80         01440           DB      128,128,128,128,128,128
6790 BF         01450           DB      191,191,128,128,128,128,128,128 ;S3
6798 80         01460           DB      128,128,128,128,128,128
679E BF         01470           DB      191,191,128,139,135,128,191,191 ;M3
67A6 80         01480           DB      128,128,128,128,128,128
67AC 80         01490           DB      128,128,128,128,128,128,128,128 ;INC3
67B4 80         01500           DB      128,128,128,128,128,128,128,128 ;INC3
                01510 ;
67BC 80         01520           DB      128,128,128,128,128,128
67C2 8B         01530           DB      139,191,191,191,191,191,191,180 ;S4
67CA 80         01540           DB      128,128,128,128,128,128
67D0 8B         01550           DB      139,191,191,191,191,191,191,180 ;S4
67D8 80         01560           DB      128,128,128,128,128,128
67DE BF         01570           DB      191,191,128,128,128,128,191,191 ;M4
67E6 80         01580           DB      128,128,128,128,128,128
67EC 80         01590           DB      128,128,128,128,128,128,128,128 ;INC4
67F4 80         01600           DB      128,128,128,128,128,128,128,128 ;INC4
                01610 ;
67FC 80         01620           DB      128,128,128,128,128,128
6802 80         01630           DB      128,128,128,128,128,128,191,191 ;S5
680A 80         01640           DB      128,128,128,128,128,128
6810 80         01650           DB      128,128,128,128,128,128,191,191 ;S5
6818 80         01660           DB      128,128,128,128,128,128
681E BF         01670           DB      191,191,128,128,128,128,191,191 ;M5
6826 80         01680           DB      128,128,128,128,128,128
682C 8C         01690           DB      140,128,176,128,128,128,128,128 ;INC5
6834 80         01700           DB      128,128,128,128,128,128,128,128 ;INC5
                01710 ;
683C 80         01720           DB      128,128,128,128,128,128
6842 80         01730           DB      128,128,128,128,128,128,191,191 ;S6
684A 80         01740           DB      128,128,128,128,128,128
6850 80         01750           DB      128,128,128,128,128,128,191,191 ;S6
6858 80         01760           DB      128,128,128,128,128,128
685E BF         01770           DB      191,191,128,128,128,128,191,191 ;M6
6866 80         01780           DB      128,128,128,128,128,128
686C BF         01790           DB      191,128,191,143,189,128,190,143 ;INC6
6874 8F         01800           DB      143,128,128,128,128,128,128,128 ;INC6
                01810 ;
687C 80         01820           DB      128,128,128,128,128,128
6882 BF         01830           DB      191,191,191,191,191,191,191,135 ;S7
688A 80         01840           DB      128,128,128,128,128,128
6890 BF         01850           DB      191,191,191,191,191,191,191,135 ;S7
6898 80         01860           DB      128,128,128,128,128,128
689E BF         01870           DB      191,191,128,128,128,128,191,191 ;M7
68A6 80         01880           DB      128,128,128,128,128,128
68AC BF         01890           DB      191,128,191,128,191,128,175,188 ;INC7
68B4 BC         01900           DB      188,128,176,128,128,128,128,128 ;INC7
                01910 ;
68BC 01         01920           DB      1                   ;1 ROW OF BLANKS
                01930 ;
68BD 01         01940           DB      1                   ;1 ROW OF DATA
```

```
                01950 ;
68BE 80         01960         DB      128,128,128,128,128,128,128,128
68C6 80         01970         DB      128,128,128,128,128,128,128
                01980 ;
68CD 53         01990         DM      'Soft Sector Marketing Incorporated'
                02000 ;
68EF 80         02010         DB      128,128,128,128,128,128,128,128
68F7 80         02020         DB      128,128,128,128,128,128,128
                02030 ;
68FE 01         02040         DB      1               ;1 ROW OF BLANKS
                02050 ;
68FF 01         02060         DB      1               ;1 ROW OF DATA
                02070 ;
6900 80         02080         DB      128,128,128,128,128,128,128,128
6908 80         02090         DB      128,128,128,128,128,128,128,128
6910 80         02100         DB      128,128,128,128,128,128,128,128
                02110 ;
6918 50         02120         DM      'P R E S E N T S'
                02130 ;
6927 80         02140         DB      128,128,128,128,128,128,128,128
692F 80         02150         DB      128,128,128,128,128,128,128,128
6937 80         02160         DB      128,128,128,128,128,128,128,128
693F 80         02170         DB      128
                02180 ;
6940 02         02190         DB      2               ;2 ROWS OF BLANKS
                02200 ;
6941 D83D       02210 ASKMES  DW      3DD8H
6943 31         02220         DM      '1 or 2 Players ?'
6953 00         02230         DB      0
                02240 ;
                02250 ;**************************************
                02260 ;**     ALIEN DEFENSE GRAPHICS      **
                02270 ;**************************************
                02280 ;
6954 02         02290 WHTDAT  DB      2               ;TWO ROWS OF BLANKS
                02300 ;
                02310 ;********** ALIEN **********
                02320 ;
6955 03         02330         DB      3               ;3 ROWS OF DATA
                02340 ;
6956 80         02350         DB      128,128,128,128,128,128,128,128
695E 80         02360         DB      128,128,128,128,128
6963 80         02370         DB      128,168,191,191,148,128,128,128 ;A1
696B BF         02380         DB      191,191,128,128,128,128,128,128 ;L1
6973 80         02390         DB      128,143,191,191,143,128,128,128 ;I1
697B BF         02400         DB      191,191,143,143,143,143,128,128 ;E1
6983 BF         02410         DB      191,191,180,128,191,191         ;N1
                02420 ;
6989 80         02430         DB      128,128,128,128,128,128,128,128
6991 80         02440         DB      128,128,128,128,128
                02450 ;
6996 80         02460         DB      128,128,128,128,128,128,128,128
699E 80         02470         DB      128,128,128,128,128
69A3 A0         02480         DB      160,191,189,190,191,144,128,128 ;A2
69AB BF         02490         DB      191,191,128,128,128,128,128,128 ;L2
69B3 80         02500         DB      128,128,191,191,128,128,128,128 ;I2
```

```
69BB BF       02510          DB       191,191,143,143,128,128,128,128 ;E2
69C3 BF       02520          DB       191,191,175,189,191,191         ;N2
69C9 80       02530          DB       128,128,128,128,128,128,128,128
69D1 80       02540          DB       128,128,128,128,128
              02550 ;
69D6 80       02560          DB       128,128,128,128,128,128,128,128
69DE 80       02570          DB       128,128,128,128,128
69E3 BE       02580          DB       190,191,129,130,191,189,128,128 ;A3
69EB BF       02590          DB       191,191,188,188,188,188,128,128 ;L3
69F3 80       02600          DB       128,188,191,191,188,128,128,128 ;I3
69FB BF       02610          DB       191,191,188,188,188,188,128,128 ;E3
6A03 BF       02620          DB       191,191,128,139,191,191         ;N3
6A09 80       02630          DB       128,128,128,128,128,128,128,128
6A11 80       02640          DB       128,128,128,128,128
              02650 ;
6A16 02       02660          DB       2                   ;2 ROWS OF BLANKS
              02670 ;
              02680 ;********** DEFENSE **********
              02690 ;
6A17 03       02700          DB       3                   ;3 ROWS OF DATA
6A18 80       02710          DB       128,128,128,128,128
6A1D BF       02720          DB       191,191,143,143,191,180,128,128 ;D1
6A25 BF       02730          DB       191,191,143,143,143,143,128,128 ;E1
6A2D BF       02740          DB       191,191,143,143,143,143,128,128 ;F1
6A35 BF       02750          DB       191,191,143,143,143,143,128,128 ;E1
6A3D BF       02760          DB       191,191,180,128,191,191,128,128 ;N1
6A45 BE       02770          DB       190,191,143,143,175,189,128,128 ;S1
6A4D BF       02780          DB       191,191,143,143,143,143         ;E1
6A53 80       02790          DB       128,128,128,128,128
              02800 ;
6A58 80       02810          DB       128,128,128,128,128
6A5D BF       02820          DB       191,191,128,128,170,191,128,128 ;D2
6A65 BF       02830          DB       191,191,143,143,128,128,128,128 ;E2
6A6D BF       02840          DB       191,191,143,143,128,128,128,128 ;F2
6A75 BF       02850          DB       191,191,143,143,128,128,128,128 ;E2
6A7D BF       02860          DB       191,191,175,189,191,191,128,128 ;N2
6A85 8B       02870          DB       139,143,191,191,188,180,128,128 ;S2
6A8D BF       02880          DB       191,191,143,143,128,128         ;E2
6A93 80       02890          DB       128,128,128,128,128
              02900 ;
6A98 80       02910          DB       128,128,128,128,128
6A9D BF       02920          DB       191,191,188,188,191,135,128,128 ;D3
6AA5 BF       02930          DB       191,191,188,188,188,188,128,128 ;E3
6AAD BF       02940          DB       191,191,128,128,128,128,128,128 ;F3
6AB5 BF       02950          DB       191,191,188,188,188,188,128,128 ;E3
6ABD BF       02960          DB       191,191,128,139,191,191,128,128 ;N3
6AC5 AF       02970          DB       175,189,188,188,191,159,128,128 ;S3
6ACD BF       02980          DB       191,191,188,188,188,188         ;E3
6AD3 80       02990          DB       128,128,128,128,128
              03000 ;
6AD8 03       03010          DB       3                   ;3 ROWS OF BLANKS
              03020 ;
6AD9 02       03030          DB       2                   ;2 ROWS OF DATA
6ADA 80       03040          DB       128,128,128,128,128,128,128,128
6AE2 80       03050          DB       128,128,128,128,128,128,128,128
6AEA 80       03060          DB       128,128,128,128,128,128,128,128
```

```
6AF2 43       03070        DB      'Copyright ',239,' 1981'
6B02 80       03080        DB      128,128,128,128,128,128,128,128
6B0A 80       03090        DB      128,128,128,128,128,128,128,128
6B12 80       03100        DB      128,128,128,128,128,128,128,128
              03110 ;
6B1A 80       03120        DB      128,128,128,128,128,128,128,128
6B22 80       03130        DB      128,128,128,128,128,128,128,128
6B2A 80       03140        DB      128,128,128
6B2D 53       03150        DM      'Soft Sector Marketing Inc.'
6B47 80       03160        DB      128,128,128,128,128,128,128,128
6B4F 80       03170        DB      128,128,128,128,128,128,128,128
6B57 80       03180        DB      128,128,128
              03190 ;
6B5A 01       03200        DB      1               ;1 ROW OF BLANKS
              03210 ;
6B5B 215469   03220 WHAT   LD      HL,WHTDAT       ;ALIEN DEFENSE DATA
6B5E 11003C   03230        LD      DE,3C00H        ;SCREEN ADDRESS
6B61 0603     03240        LD      B,3             ;THREE BLOCKS OF DATA
              03250 ;
6B63 C5       03260 WHTLP1 PUSH    BC
6B64 CD816B   03270        CALL    PRINT3          ;ROWS OF BLANKS
6B67 CDBE6B   03280        CALL    PRINT4          ;DISPLAY DATA
6B6A C1       03290        POP     BC
6B6B 10F6     03300        DJNZ    WHTLP1
              03310 ;
6B6D CD816B   03320        CALL    PRINT3          ;BLANKS
              03330 ;
              03340        IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
              03350        LD      HL,ALIEND       ;TALK DATA
              03360        CALL    TALK            ;MAKE NOISE
              03370        ENDIF
              03380 ;
6B70 CD146C   03390        CALL    DELAY6          ;PAUSE
6B73 CD146C   03400        CALL    DELAY6
6B76 CDB266   03410        CALL    COMP2           ;COMPLEMENT SCREEN
              03420 ;
              03430        IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
              03440        LD      HL,LARRYD       ;TALK DATA
              03450        CALL    TALK            ;MAKE NOISE
              03460        ENDIF
              03470 ;
6B79 0604     03480        LD      B,4
6B7B CD146C   03490 AILOP  CALL    DELAY6          ;WAIT AWHILE
6B7E 10FB     03500        DJNZ    AILOP
6B80 C9       03510        RET
              03520 ;
6B81 46       03530 PRINT3 LD      B,(HL)          ;ROW COUNT FOR BLANKS
6B82 23       03540        INC     HL
6B83 0E40     03550        LD      C,64            ;COLUMN COUNT
              03560 ;
6B85 C5       03570 PR3LP  PUSH    BC              ;SAVE ROW/COLUMN COUNT
6B86 D5       03580        PUSH    DE              ;SCREEN ADDRESS
6B87 41       03590        LD      B,C             ;GET COLUMN COUNT
              03600 ;
6B88 1A       03610 PR3LP1 LD      A,(DE)          ;GET DATA FROM SCREEN
6B89 FE80     03620        CP      128             ;GRAPHICS ?
```

```
6B8B 3807       03630           JR      C,P3SK1         ;JUMP IF NO
6B8D FEC0       03640           CP      192             ;GRAPHICS ?
6B8F 3003       03650           JR      NC,P3SK1        ;JUMP IF NO
6B91 E6FC       03660           AND     0FCH            ;MERGE IN TOP THIRD
6B93 12         03670           LD      (DE),A
6B94 13         03680 P3SK1     INC     DE
6B95 10F1       03690           DJNZ    PR3LP1
                03700 ;
6B97 CD076C     03710           CALL    DELAY5          ;PAUSE
6B9A D1         03720           POP     DE              ;SCREEN ADDRESS
6B9B D5         03730           PUSH    DE
6B9C 41         03740           LD      B,C             ;GET COLUMN COUNT
                03750 ;
6B9D 1A         03760 PR3LP2    LD      A,(DE)          ;GET SCREEN CHARACTER
6B9E FE80       03770           CP      128             ;GRAPHICS ?
6BA0 3807       03780           JR      C,P3SK2         ;JUMP IF NO
6BA2 FEC0       03790           CP      192             ;GRAPHICS ?
6BA4 3003       03800           JR      NC,P3SK2        ;JUMP IF NO
6BA6 E6F0       03810           AND     0F0H            ;MERGE IN MIDDLE THIRD
6BA8 12         03820           LD      (DE),A
6BA9 13         03830 P3SK2     INC     DE
6BAA 10F1       03840           DJNZ    PR3LP2
                03850 ;
6BAC CD076C     03860           CALL    DELAY5          ;PAUSE
6BAF D1         03870           POP     DE              ;SCREEN POSITION
6BB0 41         03880           LD      B,C             ;COLUMN COUNT
6BB1 3E80       03890           LD      A,128           ;GRAPHICS SPACE
6BB3 12         03900 PR3LP3    LD      (DE),A
6BB4 13         03910           INC     DE
6BB5 10FC       03920           DJNZ    PR3LP3
                03930 ;
6BB7 CD076C     03940           CALL    DELAY5          ;PAUSE
6BBA C1         03950           POP     BC              ;GET ROW COUNT
6BBB 10C8       03960           DJNZ    PR3LP           ;LOOP FOR ROW COUNT
6BBD C9         03970           RET
                03980 ;
6BBE 46         03990 PRINT4    LD      B,(HL)          ;ROW COUNT FOR GRAPHICS
6BBF 23         04000           INC     HL
6BC0 0E40       04010           LD      C,64            ;COLUMN COUNT
6BC2 EB         04020           EX      DE,HL           ;HL=>SCREEN DE=>DATA
                04030 ;
6BC3 C5         04040 PR4LP     PUSH    BC              ;SAVE ROW/COLUMN COUNT
6BC4 E5         04050           PUSH    HL              ;SAVE SCREEN ADDRESS
6BC5 D5         04060           PUSH    DE              ;SAVE DATA POSITION
6BC6 41         04070           LD      B,C             ;COLUMN COUNT
                04080 ;
6BC7 7E         04090 PR4LP1    LD      A,(HL)          ;GET SCREEN CHARACTER
6BC8 FE80       04100           CP      128             ;GRAPHICS ?
6BCA 3808       04110           JR      C,P4SK1         ;JUMP IF NO
6BCC FEC0       04120           CP      192             ;GRAPHICS ?
6BCE 3004       04130           JR      NC,P4SK1        ;JUMP IF NO
6BD0 1A         04140           LD      A,(DE)          ;GET DATA FROM TABLE
6BD1 E603       04150           AND     03H             ;MERGE TOP THIRD WITH
6BD3 B6         04160           OR      (HL)            ;SCREEN DATA.
                04170 ;
6BD4 77         04180 P4SK1     LD      (HL),A
```

```
6BD5 23         04190           INC     HL
6BD6 13         04200           INC     DE
6BD7 10EE       04210           DJNZ    PR4LP1
                04220 ;
6BD9 CD076C     04230           CALL    DELAY5          ;PAUSE
6BDC D1         04240           POP     DE              ;DATA POINTER
6BDD E1         04250           POP     HL              ;SCREEN POINTER
6BDE E5         04260           PUSH    HL
6BDF D5         04270           PUSH    DE
6BE0 41         04280           LD      B,C             ;COLUMN COUNT
                04290 ;
6BE1 7E         04300 PR4LP2    LD      A,(HL)          ;GET SCREEN CHARACTER
6BE2 FE80       04310           CP      128             ;GRAPHICS ?
6BE4 3808       04320           JR      C,P4SK2         ;JUMP IF NO
6BE6 FEC0       04330           CP      192             ;GRAPHICS CHARACTER ?
6BE8 3004       04340           JR      NC,P4SK2        ;JUMP IF NO
6BEA 1A         04350           LD      A,(DE)          ;GET DATA FROM TABLE
6BEB E60C       04360           AND     0CH             ;MERGE WITH MIDDLE THIRD
6BED B6         04370           OR      (HL)            ;OF SCREEN DATA
                04380 ;
6BEE 77         04390 P4SK2     LD      (HL),A
6BEF 23         04400           INC     HL
6BF0 13         04410           INC     DE
6BF1 10EE       04420           DJNZ    PR4LP2
                04430 ;
6BF3 CD076C     04440           CALL    DELAY5          ;PAUSE
6BF6 D1         04450           POP     DE              ;DATA POINTER
6BF7 E1         04460           POP     HL              ;SCREEN POINTER
6BF8 41         04470           LD      B,C             ;COLUMN COUNT
                04480 ;
6BF9 1A         04490 PR4LP3    LD      A,(DE)          ;GET DATA
6BFA 77         04500           LD      (HL),A          ;PUT ON SCREEN
6BFB 13         04510           INC     DE
6BFC 23         04520           INC     HL
6BFD 10FA       04530           DJNZ    PR4LP3
                04540 ;
6BFF CD076C     04550           CALL    DELAY5          ;PAUSE
6C02 C1         04560           POP     BC              ;ROW COUNTER
6C03 10BE       04570           DJNZ    PR4LP           ;LOOP FOR ROW COUNT
6C05 EB         04580           EX      DE,HL
6C06 C9         04590           RET
                04600 ;
6C07 F5         04610 DELAY5    PUSH    AF
6C08 C5         04620           PUSH    BC
6C09 015003     04630           LD      BC,350H
6C0C 0B         04640 DL5LP     DEC     BC
6C0D 78         04650           LD      A,B
6C0E B1         04660           OR      C
6C0F 20FB       04670           JR      NZ,DL5LP
6C11 C1         04680           POP     BC
6C12 F1         04690           POP     AF
6C13 C9         04700           RET
                04710 ;
6C14 F5         04720 DELAY6    PUSH    AF
6C15 C5         04730           PUSH    BC
6C16 010030     04740           LD      BC,3000H
```

```
6C19 0B         04750 DL6LP    DEC      BC
6C1A 78         04760          LD       A,B
6C1B B1         04770          OR       C
6C1C 20FB       04780          JR       NZ,DL6LP
6C1E C1         04790          POP      BC
6C1F F1         04800          POP      AF
6C20 C9         04810          RET
                04820 ;
6C21            01940 *GET     POINTS/ASM                  ;POINTS/TOPTEN DISPLAY
                00010 ;*************************
                00020 ;**   POINTS ROUTINE    **
                00030 ;*************************
                00040 ;
6C21 033E       00050 LNPTM    DW       3E03H              ;SCREEN ADDRESS
6C23 20         00060          DM       ' LANDER '
6C2B 00         00070          DB       0
6C2C 453E       00080          DW       3E45H              ;SCREEN ADDRESS
6C2E 20         00090          DB       ' ',224,' '        ;CHARACTER
6C31 00         00100          DB       0
6C32 843E       00110          DW       3E84H              ;SCREEN ADDRESS
6C34 20         00120          DM       ' 100 '            ;POINTS
6C39 00         00130          DB       0
                00140 ;
6C3A 133E       00150 MUPTM    DW       3E13H              ;SCREEN ADDRESS
6C3C 20         00160          DM       ' MUTANT '
6C44 00         00170          DB       0
6C45 553E       00180          DW       3E55H              ;SCREEN ADDRESS
6C47 20         00190          DB       ' ',197,' '        ;CHARACTER
6C4A 00         00200          DB       0
6C4B 943E       00210          DW       3E94H              ;SCREEN ADDRESS
6C4D 20         00220          DM       ' 150 '            ;POINTS
6C52 00         00230          DB       0
                00240 ;
6C53 213E       00250 CRPTM    DW       3E21H              ;SCREEN ADDRESS
6C55 20         00260          DM       ' CRUISER '
6C5E 00         00270          DB       0
6C5F 643E       00280          DW       3E64H              ;SCREEN ADDRESS
6C61 20         00290          DB       ' ',220,' '        ;CHARACTER
6C64 00         00300          DB       0
6C65 A33E       00310          DW       3EA3H              ;SCREEN ADDRESS
6C67 20         00320          DM       ' 200 '            ;POINTS
6C6C 00         00330          DB       0
                00340 ;
6C6D 033F       00350 BMPTM    DW       3F03H              ;SCREEN ADDRESS
6C6F 20         00360          DM       ' BOMBER '
6C77 00         00370          DB       0
6C78 453F       00380          DW       3F45H              ;SCREEN ADDRESS
6C7A 20         00390          DB       ' ',240,' '        ;CHARACTER
6C7D 00         00400          DB       0
6C7E 843F       00410          DW       3F84H              ;SCREEN ADDRESS
6C80 20         00420          DM       ' 250 '            ;POINTS
6C85 00         00430          DB       0
                00440 ;
6C86 143F       00450 PDPTM    DW       3F14H              ;SCREEN ADDRESS
6C88 20         00460          DM       ' POD '
6C8D 00         00470          DB       0
```

```
6C8E 553F     00480           DW      3F55H           ;SCREEN ADDRESS
6C90 20       00490           DB      ' ',194,' '     ;CHARACTER
6C93 00       00500           DB      0
6C94 933F     00510           DW      3F93H           ;SCREEN ADDRESS
6C96 20       00520           DM      ' 1000 '        ;POINTS
6C9C 00       00530           DB      0
              00540 ;
6C9D 213F     00550 SWPTM     DW      3F21H           ;SCREEN ADDRESS
6C9F 20       00560           DM      ' SWARMER '
6CA8 00       00570           DB      0
6CA9 643F     00580           DW      3F64H           ;SCREEN ADDRESS
6CAB 20       00590           DB      ' ',229,' '     ;CHARACTER
6CAE 00       00600           DB      0
6CAF A33F     00610           DW      3FA3H           ;SCREEN ADDRESS
6CB1 20       00620           DM      ' 150 '         ;POINTS
6CB6 00       00630           DB      0
              00640 ;
6CB7 20       00650 BO5M      DM      ' 500 '         ;BONUS
              00660 ;
6CBC 00       00670 PNTTBL    DB      0,0,0,0,0,0
              00680 ;
6CC2 CD146C   00690 POINTS    CALL    DELAY6          ;PAUSE
6CC5 CD146C   00700           CALL    DELAY6
6CC8 210358   00710           LD      HL,UPDT1        ;CHANGE SHIP DRAW VECTOR
6CCB 22A257   00720           LD      (UPDATE+1),HL
6CCE 210000   00730           LD      HL,0            ;INIT VARIABLES FOR
              00740                                   ;ATTRACT MODE.
6CD1 22A973   00750           LD      (DIRFLG),HL
6CD4 22FB51   00760           LD      (PL1SCL),HL
6CD7 22D15B   00770           LD      (SHTF1),HL
6CDA 229B73   00780           LD      (CMEAGN),HL
6CDD AF       00790           XOR     A
6CDE 32FD51   00800           LD      (PL1SCH),A
6CE1 32B673   00810           LD      (MVFLG3),A
6CE4 320152   00820           LD      (PLFLG),A
6CE7 323754   00830           LD      (EXTBL),A
6CEA 329D73   00840           LD      (WAVCNT),A
6CED 3E06     00850           LD      A,6
6CEF 32A173   00860           LD      (SHROW),A
6CF2 21893C   00870           LD      HL,3C89H
6CF5 22A773   00880           LD      (SHPOS),HL
              00890 ;
6CF8 CD455C   00900           CALL    CLS             ;CLEAR THE SCREEN
6CFB 211F4B   00910           LD      HL,STRTMS
6CFE CD525D   00920           CALL    PRINT
6D01 CDCB50   00930           CALL    SCORE
6D04 ED5BA773 00940           LD      DE,(SHPOS)      ;GET SHIP POSITION
6D08 CDA157   00950           CALL    UPDATE          ;DRAW SHIP
6D0B CDF958   00960           CALL    DRAW            ;DRAW MOUNTAINS
6D0E CD146C   00970           CALL    DELAY6          ;PAUSE
6D11 CD146C   00980           CALL    DELAY6
              00990 ;
              01000           IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
              01010           LD      HL,HOWSD        ;TALK DATA
              01020           CALL    TALK            ;MAKE NOISE
              01030           ENDIF
```

```
               01040 ;
6D14 DD21BC6C  01050          LD       IX,PNTTBL       ;DUMMY TABLE
6D18 DD3600E0  01060          LD       (IX),224        ;PROCESS LANDER
6D1C DDE5      01070          PUSH     IX
6D1E CDEA6D    01080          CALL     CHRON           ;BRING ON CHARACTER
               01090                                   ;MOVE LANDER & FIRE
6D21 21216C    01100          LD       HL,LNPTM        ;DISPLAY LANDER MESSAGE
6D24 CDD66D    01110          CALL     PRINT5
6D27 CD776E    01120          CALL     MOSUBB          ;MOVE LANDER TO HOME POSITION
6D2A CD146C    01130          CALL     DELAY6
6D2D CD146C    01140          CALL     DELAY6
6D30 DDE1      01150          POP      IX
               01160 ;
6D32 DD3600C5  01170          LD       (IX),197        ;PROCESS MUTANT
6D36 DDE5      01180          PUSH     IX
6D38 CDEA6D    01190          CALL     CHRON
6D3B 213A6C    01200          LD       HL,MUPTM
6D3E CDD66D    01210          CALL     PRINT5
6D41 CD776E    01220          CALL     MOSUBB
6D44 CD146C    01230          CALL     DELAY6
6D47 CD146C    01240          CALL     DELAY6
6D4A DDE1      01250          POP      IX
               01260 ;
6D4C DD3600DC  01270          LD       (IX),220        ;PROCESS CRUISER
6D50 DDE5      01280          PUSH     IX
6D52 CDEA6D    01290          CALL     CHRON
6D55 21536C    01300          LD       HL,CRPTM
6D58 CDD66D    01310          CALL     PRINT5
6D5B CD776E    01320          CALL     MOSUBB
6D5E CD146C    01330          CALL     DELAY6
6D61 CD146C    01340          CALL     DELAY6
6D64 DDE1      01350          POP      IX
               01360 ;
6D66 DD3600F0  01370          LD       (IX),240        ;PROCESS BOMBER
6D6A DDE5      01380          PUSH     IX
6D6C CDEA6D    01390          CALL     CHRON
6D6F 216D6C    01400          LD       HL,BMPTM
6D72 CDD66D    01410          CALL     PRINT5
6D75 CD776E    01420          CALL     MOSUBB
6D78 CD146C    01430          CALL     DELAY6
6D7B CD146C    01440          CALL     DELAY6
6D7E DDE1      01450          POP      IX
               01460 ;
6D80 DD3600C2  01470          LD       (IX),194        ;PROCESS POD
6D84 DDE5      01480          PUSH     IX
6D86 CDEA6D    01490          CALL     CHRON
6D89 21866C    01500          LD       HL,PDPTM
6D8C CDD66D    01510          CALL     PRINT5
6D8F CD776E    01520          CALL     MOSUBB
6D92 CD146C    01530          CALL     DELAY6
6D95 CD146C    01540          CALL     DELAY6
6D98 DDE1      01550          POP      IX
               01560 ;
6D9A DD3600E5  01570          LD       (IX),229        ;PROCESS SWARMER
6D9E DDE5      01580          PUSH     IX
6DA0 CDEA6D    01590          CALL     CHRON
```

```
6DA3 219D6C   01600          LD      HL,SWPTM
6DA6 CDD66D   01610          CALL    PRINT5
6DA9 CD776E   01620          CALL    MOSUBB
6DAC CD146C   01630          CALL    DELAY6
6DAF CD146C   01640          CALL    DELAY6
              01650 ;
6DB2 21B53F   01660          LD      HL,3FB5H          ;SCREEN POSITION
6DB5 36E0     01670          LD      (HL),224          ;DRAW LANDER
6DB7 CD445D   01680          CALL    AD64
6DBA 36FD     01690          LD      (HL),253          ;DRAW MAN
6DBC EB       01700          EX      DE,HL
6DBD CD146C   01710          CALL    DELAY6
6DC0 CDBE6E   01720          CALL    FIVE              ;MOVE LANDER/MAN UP
6DC3 CD086E   01730          CALL    MOVF              ;MOVE SHIP & FIRE
6DC6 CDD66E   01740          CALL    MOSUBF            ;CATCH MAN,MOVE DOWN,MOVE UP
6DC9 CD776E   01750          CALL    MOSUBB            ;MOVE TO HOME POSITION
              01760 ;
6DCC 0605     01770          LD      B,5
6DCE CD146C   01780 DELP     CALL    DELAY6
6DD1 10FB     01790          DJNZ    DELP
6DD3 DDE1     01800          POP     IX
6DD5 C9       01810          RET
              01820 ;
6DD6 0603     01830 PRINT5   LD      B,3
6DD8 CDA666   01840 PTLP1    CALL    PRINT2
6DDB 10FB     01850          DJNZ    PTLP1
6DDD 3E01     01860          LD      A,1
6DDF 32AA73   01870          LD      (DIRF2),A         ;SHIP DIRECTION
6DE2 ED5BA773 01880          LD      DE,(SHPOS)        ;SHIP POSITION
6DE6 CDA157   01890          CALL    UPDATE            ;DRAW SHIP
6DE9 C9       01900          RET
              01910 ;
6DEA 0607     01920 CHRON    LD      B,7               ;WARP-IN PHASE COUNT
6DEC DD7004   01930          LD      (IX+4),B
6DEF 21B03D   01940          LD      HL,3DB0H          ;SCREEN POSITION
              01950 ;
6DF2 C5       01960 CONLP1   PUSH    BC
6DF3 E5       01970          PUSH    HL
6DF4 CDE75C   01980          CALL    WARPIN            ;BRING ON CHARACTER
6DF7 CD1E55   01990          CALL    DELAY             ;PAUSE
6DFA E1       02000          POP     HL
6DFB C1       02010          POP     BC
6DFC 10F4     02020          DJNZ    CONLP1
              02030 ;
6DFE DD7E00   02040          LD      A,(IX)            ;GET CHARACTER TYPE
6E01 77       02050          LD      (HL),A            ;DISPLAY
6E02 CD146C   02060          CALL    DELAY6            ;PAUSE
6E05 CD146C   02070          CALL    DELAY6
              02080 ;
6E08 010604   02090 MOVF     LD      BC,0406H          ;B=ROWS : C=COLUMNS
6E0B 3E01     02100          LD      A,1
6E0D 32B673   02110          LD      (MVFLG3),A        ;FLAG MOTION
              02120 ;
6E10 C5       02130 MOSLP1   PUSH    BC                ;ROW/COLUMN
6E11 41       02140          LD      B,C               ;GET COLUMN COUNT
6E12 ED5BA773 02150          LD      DE,(SHPOS)        ;GET SHIP POSITION
```

```
                02160 ;
6E16 D5         02170 MOSLP2  PUSH    DE
6E17 E1         02180         POP     HL
6E18 C5         02190         PUSH    BC
6E19 CD8757     02200         CALL    ERASE           ;DE COMES BACK W/OLD
6E1C 13         02210         INC     DE
6E1D CDA157     02220         CALL    UPDATE          ;DRAW SHIP
6E20 CD5D6E     02230         CALL    DELAY7          ;PAUSE
6E23 C1         02240         POP     BC
6E24 10F0       02250         DJNZ    MOSLP2
                02260 ;
6E26 EB         02270         EX      DE,HL
6E27 CD445D     02280         CALL    AD64            ;DE=>OLD :HL=>NEW
6E2A CD8757     02290         CALL    ERASE           ;ERASE SHIP
6E2D ED53A773   02300         LD      (SHPOS),DE      ;DE=>NEW
6E31 CDA157     02310         CALL    UPDATE          ;DRAW SHIP
6E34 CD5D6E     02320         CALL    DELAY7          ;PAUSE
6E37 C1         02330         POP     BC
6E38 10D6       02340         DJNZ    MOSLP1          ;LOOP FOR ROW COUNT
                02350 ;
6E3A CD3859     02360         CALL    FIRE            ;FIRE A SHOT
6E3D CD5D6E     02370         CALL    DELAY7          ;PAUSE
6E40 CD885A     02380         CALL    SHTCLR          ;CLEAR THE SHOT
6E43 CDCB50     02390         CALL    SCORE           ;UPDATE SCORE
6E46 CDF552     02400         CALL    EXPLOD          ;START EXPLOSION
6E49 0603       02410         LD      B,3             ;PHASE COUNT
                02420 ;
6E4B C5         02430 MOSLP3  PUSH    BC
6E4C CD5D6E     02440         CALL    DELAY7
6E4F CD2D53     02450         CALL    EXPLD           ;PROCESS EXPLOSION
6E52 C1         02460         POP     BC
6E53 10F6       02470         DJNZ    MOSLP3
                02480 ;
6E55 AF         02490         XOR     A
6E56 32D15B     02500         LD      (SHTF1),A       ;CLEAR SHOT
6E59 32D25B     02510         LD      (SHTF2),A
6E5C C9         02520         RET
                02530 ;
6E5D F5         02540 DELAY7  PUSH    AF
6E5E C5         02550         PUSH    BC
6E5F 010007     02560         LD      BC,700H
6E62 0B         02570 DL7LP   DEC     BC
6E63 78         02580         LD      A,B
6E64 B1         02590         OR      C
6E65 20FB       02600         JR      NZ,DL7LP
6E67 C1         02610         POP     BC
6E68 F1         02620         POP     AF
6E69 C9         02630         RET
                02640 ;
6E6A F5         02650 DELAY8  PUSH    AF
6E6B C5         02660         PUSH    BC
6E6C 010015     02670         LD      BC,1500H
6E6F 0B         02680 DL8LP   DEC     BC
6E70 78         02690         LD      A,B
6E71 B1         02700         OR      C
6E72 20FB       02710         JR      NZ,DL8LP
```

```
6E74 C1       02720          POP     BC
6E75 F1       02730          POP     AF
6E76 C9       02740          RET
              02750 ;
              02760 ;        *****   MOVE SHIP TO HOME POSITION   *****
              02770 ;
6E77 ED5BA773 02780 MOSUBB   LD      DE,(SHPOS)       ;SHIP POSITION
6E7B D5       02790          PUSH    DE
6E7C E1       02800          POP     HL
6E7D 0618     02810          LD      B,24             ;MOVE 24 COLUMNS
              02820 ;
6E7F C5       02830 MOBLP1   PUSH    BC
6E80 CD8757   02840          CALL    ERASE            ;ERASE SHIP
6E83 1B       02850          DEC     DE               ;UPDATE POSITION
6E84 CDA157   02860          CALL    UPDATE           ;DRAW SHIP
6E87 CD5D6E   02870          CALL    DELAY7           ;PAUSE
6E8A D5       02880          PUSH    DE
6E8B E1       02890          POP     HL
6E8C C1       02900          POP     BC
6E8D 10F0     02910          DJNZ    MOBLP1           ;LOOP FOR COLUMN COUNT
6E8F AF       02920          XOR     A
6E90 32AA73   02930          LD      (DIRF2),A        ;CHANGE DIRECTION
6E93 CDA157   02940          CALL    UPDATE           ;REDRAW SHIP
6E96 0604     02950          LD      B,4              ;UP 4 ROWS
6E98 ED53A773 02960          LD      (SHPOS),DE       ;SAVE CURRENT SHIP POS.
              02970 ;
6E9C C5       02980 MOBLP2   PUSH    BC
6E9D CD5D6E   02990          CALL    DELAY7           ;PAUSE
6EA0 2AA773   03000          LD      HL,(SHPOS)       ;GET SHIP POSITION
6EA3 CD4B5D   03010          CALL    SB64             ;MOVE UP 1 ROW
6EA6 CD8757   03020          CALL    ERASE            ;ERASE OLD
6EA9 CDA157   03030          CALL    UPDATE           ;DRAW NEW
6EAC ED53A773 03040          LD      (SHPOS),DE       ;SAVE POSITION
6EB0 C1       03050          POP     BC
6EB1 10E9     03060          DJNZ    MOBLP2           ;LOOP FOR ROW COUNT
              03070 ;
6EB3 AF       03080          XOR     A
6EB4 32B673   03090          LD      (MVFLG3),A       ;CLEAR MOTION
6EB7 CDA157   03100          CALL    UPDATE           ;REDRAW SHIP
6EBA CD146C   03110          CALL    DELAY6           ;PAUSE
6EBD C9       03120          RET
              03130 ;
6EBE 0608     03140 FIVE     LD      B,8              ;ROW COUNT
6EC0 C5       03150 FIVLP1   PUSH    BC
6EC1 CD5D6E   03160          CALL    DELAY7           ;PAUSE
6EC4 CD5D6E   03170          CALL    DELAY7
6EC7 3E80     03180          LD      A,128            ;CLEAR LAST POSITION
6EC9 12       03190          LD      (DE),A
6ECA CD4B5D   03200          CALL    SB64             ;CALCULATE NEW POSITION
6ECD 3EFD     03210          LD      A,253
6ECF 36E0     03220          LD      (HL),224         ;DRAW LANDER
6ED1 12       03230          LD      (DE),A           ;DRAW MAN
6ED2 C1       03240          POP     BC
6ED3 10EB     03250          DJNZ    FIVLP1           ;LOOP FOR ROW COUNT
6ED5 C9       03260          RET
              03270 ;
```

```
6ED6 ED5BA773 03280 MOSUBF  LD      DE,(SHPOS)      ;GET SHIP POSITION
6EDA 0612     03290         LD      B,18            ;COLUMN COUNT
              03300 ;
6EDC D5       03310 MOF1    PUSH    DE              ;MOVE FORWARD 18 COLUMNS
6EDD E1       03320         POP     HL
6EDE C5       03330         PUSH    BC
6EDF CD8757   03340         CALL    ERASE           ;ERASE SHIP
6EE2 13       03350         INC     DE
6EE3 CDA157   03360         CALL    UPDATE          ;DRAW SHIP
6EE6 CD5D6E   03370         CALL    DELAY7          ;PAUSE
6EE9 C1       03380         POP     BC
6EEA 10F0     03390         DJNZ    MOF1            ;LOOP FOR COLUMN COUNT
              03400 ;
6EEC 3E01     03410         LD      A,1             ;CHANGE DIRECTION
6EEE 32AA73   03420         LD      (DIRF2),A
6EF1 CDA157   03430         CALL    UPDATE          ;TURN AROUND
6EF4 ED53A773 03440         LD      (SHPOS),DE      ;SAVE POSITION
6EF8 CD506F   03450         CALL    BFIV            ;"500 POINTS"
6EFB CD5D6E   03460         CALL    DELAY7          ;PAUSE
              03470 ;
6EFE 0608     03480         LD      B,8             ;ROW COUNT
6F00 C5       03490 MOF2    PUSH    BC              ;MOVE DOWN 8 ROWS W/MAN
6F01 CD5D6E   03500         CALL    DELAY7          ;PAUSE
6F04 CD5D6E   03510         CALL    DELAY7
6F07 2AA773   03520         LD      HL,(SHPOS)      ;GET SHIP POSITION
6F0A CD445D   03530         CALL    AD64
6F0D CD9057   03540         CALL    ERASE2          ;ERASE OLD
6F10 CDF157   03550         CALL    UPDT2           ;DRAW NEW
6F13 ED53A773 03560         LD      (SHPOS),DE      ;SAVE POSITION
6F17 C1       03570         POP     BC
6F18 10E6     03580         DJNZ    MOF2            ;LOOP FOR ROW COUNT
6F1A CD506F   03590         CALL    BFIV            ;"500 POINTS"
              03600 ;
6F1D 0608     03610         LD      B,8             ;MOVE UP 8
6F1F C5       03620 MOF3    PUSH    BC              ;W/OUT MAN
6F20 CD5D6E   03630         CALL    DELAY7          ;PAUSE
6F23 CD5D6E   03640         CALL    DELAY7
6F26 2AA773   03650         LD      HL,(SHPOS)      ;GET SHIP POSITION
6F29 CD4B5D   03660         CALL    SB64
6F2C CD8757   03670         CALL    ERASE           ;ERASE OLD
6F2F CDA157   03680         CALL    UPDATE          ;DRAW NEW
6F32 ED53A773 03690         LD      (SHPOS),DE      ;SAVE POSITION
6F36 C1       03700         POP     BC
6F37 10E6     03710         DJNZ    MOF3            ;LOOP FOR ROW COUNT
              03720 ;
6F39 0612     03730         LD      B,18            ;MOVE RIGHT 18
6F3B D5       03740 MOB4    PUSH    DE
6F3C E1       03750         POP     HL
6F3D C5       03760         PUSH    BC
6F3E CD8757   03770         CALL    ERASE           ;ERASE OLD
6F41 1B       03780         DEC     DE
6F42 CDA157   03790         CALL    UPDATE          ;DRAW NEW
6F45 CD5D6E   03800         CALL    DELAY7          ;PAUSE
6F48 C1       03810         POP     BC
6F49 10F0     03820         DJNZ    MOB4            ;LOOP FOR COLUMN COUNT
6F4B ED53A773 03830         LD      (SHPOS),DE      ;SAVE POSITION
```

```
6F4F C9         03840          RET
                03850 ;
6F50 E5         03860 BFIV     PUSH    HL                 ;DISPLAY "500 POINTS"
6F51 D5         03870          PUSH    DE
6F52 EB         03880          EX      DE,HL
6F53 23         03890          INC     HL
6F54 23         03900          INC     HL
6F55 23         03910          INC     HL
6F56 23         03920          INC     HL
6F57 11B76C     03930          LD      DE,BO5M
6F5A 0605       03940          LD      B,5
6F5C 1A         03950 BFIVLP   LD      A,(DE)
6F5D 77         03960          LD      (HL),A
6F5E 23         03970          INC     HL
6F5F 13         03980          INC     DE
6F60 10FA       03990          DJNZ    BFIVLP
6F62 01F401     04000          LD      BC,500
6F65 CDED4F     04010          CALL    HIT                ;ADD TO SCORE
6F68 CDCB50     04020          CALL    SCORE              ;DISPLAY SCORE
6F6B D1         04030          POP     DE
6F6C E1         04040          POP     HL
6F6D CD934F     04050          CALL    BINK
6F70 C9         04060          RET
                04070 ;
                04080 ;*************************
                04090 ;**      TOPTEN SCORES  **
                04100 ;*************************
                04110 ;
6F71 00         04120 TENTBL   DB      0,0,0              ;SCORE 1
6F74 20         04130          DM      '   '              ;INITIALS 1
6F77 00         04140          DB      0,0,0              ;2
6F7A 20         04150          DM      '   '
6F7D 00         04160          DB      0,0,0              ;3
6F80 20         04170          DM      '   '
6F83 00         04180          DB      0,0,0              ;4
6F86 20         04190          DM      '   '
6F89 00         04200          DB      0,0,0              ;5
6F8C 20         04210          DM      '   '
6F8F 00         04220          DB      0,0,0              ;6
6F92 20         04230          DM      '   '
6F95 00         04240          DB      0,0,0              ;7
6F98 20         04250          DM      '   '
6F9B 00         04260          DB      0,0,0              ;8
6F9E 20         04270          DM      '   '
6FA1 00         04280          DB      0,0,0              ;9
6FA4 20         04290          DM      '   '
6FA7 00         04300 TENEND   DB      0,0,0              ;10
6FAA 20         04310          DM      '   '
                04320 ;
6FAD 00         04330 TCHK     DB      0,0,0              ;SCORE TO CHECK
                04340 ;
003C            04350 TENBUF   DS      60                 ;BUFFER FOR COMPARE/SORT
                04360 ;
6FEC 3A0152     04370 TOPCHK   LD      A,(PLFLG)          ;WHO'S UP
6FEF B7         04380          OR      A
6FF0 ED5BFB51   04390          LD      DE,(PL1SCL)        ;PLAYER 1 SCORE
```

```
6FF4 3AFD51    04400         LD      A,(PL1SCH)
6FF7 2807      04410         JR      Z,TOPNXT         ;JUMP IF PLAYER 1
               04420 ;
6FF9 ED5BFE51  04430         LD      DE,(PL2SCL)      ;PLAYER 2 SCORE
6FFD 3A0052    04440         LD      A,(PL2SCH)
               04450 ;
7000 FD21AD6F  04460 TOPNXT  LD      IY,TCHK          ;STORAGE FOR CHECK SCORE
7004 FD7300    04470         LD      (IY),E           ;LOW BYTE
7007 FD7201    04480         LD      (IY+1),D         ;MIDDLE BYTE
700A FD7702    04490         LD      (IY+2),A         ;HIGH BYTE
               04500 ;
700D DD21716F  04510         LD      IX,TENTBL        ;TOP TEN SCORES TABLE
7011 060A      04520         LD      B,10             ;# OF ENTRIES
7013 110600    04530         LD      DE,6             ;ENTRY OFFSET
               04540 ;
7016 B7        04550 TENLOP  OR      A                ;CLEAR CARRY
7017 DD7E00    04560         LD      A,(IX)           ;LO BYTE - TABLE
701A FD9E00    04570         SBC     A,(IY)           ;PLAYER LO BYTE
701D DD7E01    04580         LD      A,(IX+1)         ;MID BYTE TABLE
7020 FD9E01    04590         SBC     A,(IY+1)         ;PLAYER MID BYTE
7023 DD7E02    04600         LD      A,(IX+2)         ;HI BYTE TABLE
7026 FD9E02    04610         SBC     A,(IY+2)         ;PLAYER HI BYTE
7029 3805      04620         JR      C,REPLCE         ;JUMP IF GREATER
               04630 ;
702B DD19      04640         ADD     IX,DE            ;NEXT ENTRY
702D 10E7      04650         DJNZ    TENLOP
702F C9        04660         RET                      ;BETTER LUCK NEXT TIME
               04670 ;
7030 DD21A76F  04680 REPLCE  LD      IX,TENEND
7034 FD7E00    04690         LD      A,(IY)           ;TRANSFER SCORE TO
7037 DD7700    04700         LD      (IX),A           ;LOWEST POSITION ON
703A FD7E01    04710         LD      A,(IY+1)         ;TOP TEN TABLE
703D DD7701    04720         LD      (IX+1),A
7040 FD7E02    04730         LD      A,(IY+2)
7043 DD7702    04740         LD      (IX+2),A
               04750 ;
               04760         IFEQ    TLK,1            ;ASSEMBLE IF TALK VER.
               04770         LD      HL,GREATD        ;TALK DATA
               04780         CALL    TALK             ;MAKE NOISE
               04790         ENDIF
               04800 ;
7046 212A71    04810         LD      HL,REMES         ;"YOU'RE IN THE TOP TEN"
7049 CDA666    04820         CALL    PRINT2
704C CDA666    04830         CALL    PRINT2
704F CDE670    04840         CALL    INPUT            ;GET INITIALS
7052 7E        04850         LD      A,(HL)           ;TRANSFER TO TOP TEN TABLE
7053 DD7703    04860         LD      (IX+3),A
7056 23        04870         INC     HL
7057 7E        04880         LD      A,(HL)
7058 DD7704    04890         LD      (IX+4),A
705B 23        04900         INC     HL
705C 7E        04910         LD      A,(HL)
705D DD7705    04920         LD      (IX+5),A
               04930 ;
7060 013C00    04940         LD      BC,60            ;MOVE SCORE TABLE
7063 21716F    04950         LD      HL,TENTBL        ;TO SCORE BUFFER
```

```
7066 E5        04960          PUSH    HL
7067 11B06F    04970          LD      DE,TENBUF
706A EDB0      04980          LDIR
               04990 ;
706C E1        05000          POP     HL                ;BEGIN DECENDING SORT
706D 060A      05010          LD      B,10              ;# OF ITEMS
               05020 ;
706F C5        05030 SORLOP   PUSH    BC                ;SAVE COUNT
7070 CD7E70    05040          CALL    SORT
7073 C1        05050          POP     BC
7074 10F9      05060          DJNZ    SORLOP            ;LOOP FOR COUNT
               05070 ;
7076 0603      05080          LD      B,3
7078 CD146C    05090 SRLOP1   CALL    DELAY6            ;WAIT AWHILE
707B 10FB      05100          DJNZ    SRLOP1
707D C9        05110          RET
               05120 ;
707E E5        05130 SORT     PUSH    HL                ;SAVE TABLE POSITION
707F DD21B06F  05140          LD      IX,TENBUF         ;TOP TEN TABLE
7083 DDE5      05150          PUSH    IX
7085 FDE1      05160          POP     IY
7087 110000    05170          LD      DE,0              ;CLEAR ENTRY OFFSET
708A 060A      05180          LD      B,10              ;ENTRY COUNT
               05190 ;
708C DD19      05200 HILP1    ADD     IX,DE             ;NEXT ENTRY
708E B7        05210          OR      A                 ;CLEAR CARRY
708F FD7E00    05220          LD      A,(IY)            ;LO BYTE
7092 DD9E00    05230          SBC     A,(IX)
7095 FD7E01    05240          LD      A,(IY+1)          ;MID BYTE
7098 DD9E01    05250          SBC     A,(IX+1)
709B FD7E02    05260          LD      A,(IY+2)          ;HI BYTE
709E DD9E02    05270          SBC     A,(IX+2)
               05280 ;
70A1 DCE170    05290          CALL    C,HIGHER          ;CALL IF (IX) HIGHER
70A4 110600    05300          LD      DE,6              ;ENTRY OFFSET
70A7 10E3      05310          DJNZ    HILP1             ;LOOP FOR ENTRY COUNT
               05320 ;
70A9 E1        05330          POP     HL                ;GET TABLE POS.
70AA FD7E00    05340          LD      A,(IY)            ;LO BYTE
70AD 77        05350          LD      (HL),A
70AE 23        05360          INC     HL
70AF FD360000  05370          LD      (IY),0            ;CLEAR BUFFER
70B3 FD7E01    05380          LD      A,(IY+1)          ;MID BYTE
70B6 77        05390          LD      (HL),A
70B7 23        05400          INC     HL
70B8 FD360100  05410          LD      (IY+1),0
70BC FD7E02    05420          LD      A,(IY+2)          ;HI BYTE
70BF 77        05430          LD      (HL),A
70C0 23        05440          INC     HL
70C1 FD360200  05450          LD      (IY+2),0
70C5 FD7E03    05460          LD      A,(IY+3)          ;INITIAL 1
70C8 77        05470          LD      (HL),A
70C9 23        05480          INC     HL
70CA FD360380  05490          LD      (IY+3),128
70CE FD7E04    05500          LD      A,(IY+4)          ;2
70D1 77        05510          LD      (HL),A
```

```
70D2 23        05520            INC      HL
70D3 FD360480  05530            LD       (IY+4),128
70D7 FD7E05    05540            LD       A,(IY+5)           ;3
70DA 77        05550            LD       (HL),A
70DB 23        05560            INC      HL
70DC FD360580  05570            LD       (IY+5),128
70E0 C9        05580            RET
               05590 ;
70E1 DDE5      05600 HIGHER     PUSH     IX
70E3 FDE1      05610            POP      IY
70E5 C9        05620            RET
               05630 ;
70E6 0603      05640 INPUT      LD       B,3                ;INPUT COUNT
70E8 218A71    05650            LD       HL,BUFF            ;KEY BUFFER
70EB 13        05660            INC      DE                 ;SCREEN POSITION
70EC C5        05670            PUSH     BC                 ;SAVE COUNT
70ED E5        05680            PUSH     HL                 ;SAVE BUFFER POINTER
70EE 04        05690            INC      B                  ;COUNT + 1
               05700 ;
70EF 3680      05710 INLOP      LD       (HL),128           ;CLEAR BUFFER
70F1 23        05720            INC      HL
70F2 10FB      05730            DJNZ     INLOP
               05740 ;
70F4 E1        05750            POP      HL                 ;BUFFER
70F5 C1        05760            POP      BC                 ;COUNT
               05770 ;
70F6 D5        05780 WAIT       PUSH     DE                 ;SAVE SCREEN POSITION
70F7 CD4900    05790            CALL     49H                ;WAIT FOR KEY PRESSED
70FA D1        05800            POP      DE                 ;SCREEN POSITION
70FB FE0D      05810            CP       13                 ;ENTER KEY ?
70FD 2819      05820            JR       Z,INEXT            ;JUMP IF YES
               05830 ;
70FF FE08      05840            CP       8                  ;BACKSPACE ?
7101 2819      05850            JR       Z,BKSPC            ;JUMP IF YES
               05860 ;
7103 FE41      05870            CP       'A'                ;VALID KEY ?
7105 38EF      05880            JR       C,WAIT             ;JUMP IF NO
               05890 ;
7107 FE5B      05900            CP       91                 ;VALID KEY ?
7109 30EB      05910            JR       NC,WAIT            ;JUMP IF NO
               05920 ;
710B 4F        05930            LD       C,A                ;SAVE KEY PRESSED
710C 78        05940            LD       A,B                ;GET COUNT
710D B7        05950            OR       A                  ;MAX INPUT ALREADY ?
710E 28E6      05960            JR       Z,WAIT             ;JUMP IF YES
               05970 ;
7110 05        05980            DEC      B                  ;UPDATE COUNT
7111 79        05990            LD       A,C                ;GET KEY PRESSED
7112 13        06000            INC      DE                 ;UPDATE SCREEN POSITION
7113 12        06010            LD       (DE),A             ;PUT ON SCREEN
7114 77        06020            LD       (HL),A             ;PUT IN BUFFER
7115 23        06030            INC      HL                 ;UPDATE BUFFER POINTER
7116 18DE      06040            JR       WAIT               ;LOOP TIL ENTER PRESSED
               06050 ;
7118 218A71    06060 INEXT      LD       HL,BUFF            ;START OF KEY BUFFER
711B C9        06070            RET
```

```
                 06080 ;
711C 78          06090 BKSPC    LD      A,B               ;GET COUNT
711D FE03        06100          CP      3                 ;ALREADY ALL THE WAY BACK ?
711F 3E80        06110          LD      A,128
7121 12          06120          LD      (DE),A            ;ERASE SCREEN
7122 30D2        06130          JR      NC,WAIT           ;JUMP IF YES
                 06140 ;
7124 04          06150          INC     B                 ;UPDATE COUNT
7125 1B          06160          DEC     DE                ;UPDATE SCREEN POSITION
7126 2B          06170          DEC     HL                ;UPDATE BUFFER
7127 77          06180          LD      (HL),A            ;ERASE BUFFER
7128 18CC        06190          JR      WAIT
                 06200 ;
712A 503D        06210 REMES    DW      3D50H             ;SCREEN ADDRESS
712C 59          06220          DM      'Your score is in the TOP TEN !!!'
714C 00          06230          DB      0
714D C93D        06240          DW      3DC9H             ;SCREEN ADDRESS
714F 50          06250          DM      'Please type  your initials and press ENTER '
717A 00          06260          DB      0
                 06270 ;
717B 48          06280 TENMSG   DM      'HIGH SCORES  '
7188 00          06290          DB      0
7189 00          06300 TENFLG   DB      0
0004             06310 BUFF     DS      4
                 06320 ;
718E 3A8971      06330 TENPNT   LD      A,(TENFLG)        ;ANYTHING THERE ?
7191 B7          06340          OR      A
7192 C8          06350          RET     Z                 ;RETURN IF NO
                 06360 ;
7193 CD455C      06370          CALL    CLS               ;CLEAR THE SCREEN
7196 FD21716F    06380          LD      IY,TENTBL         ;TOP TEN SCORE TABLE
719A 21CD3C      06390          LD      HL,3CCDH          ;SCREEN POSITION
719D CDFF71      06400          CALL    IPNT              ;1
                 06410 ;
71A0 21263D      06420          LD      HL,3D26H
71A3 CDFF71      06430          CALL    IPNT              ;2
                 06440 ;
71A6 214D3D      06450          LD      HL,3D4DH
71A9 CDFF71      06460          CALL    IPNT              ;3
                 06470 ;
71AC 21A63D      06480          LD      HL,3DA6H
71AF CDFF71      06490          CALL    IPNT              ;4
                 06500 ;
71B2 21CD3D      06510          LD      HL,3DCDH
71B5 CDFF71      06520          CALL    IPNT              ;5
                 06530 ;
71B8 21263E      06540          LD      HL,3E26H
71BB CDFF71      06550          CALL    IPNT              ;6
                 06560 ;
71BE 214D3E      06570          LD      HL,3E4DH
71C1 CDFF71      06580          CALL    IPNT              ;7
                 06590 ;
71C4 21A63E      06600          LD      HL,3EA6H
71C7 CDFF71      06610          CALL    IPNT              ;8
                 06620 ;
71CA 21CD3E      06630          LD      HL,3ECDH
```

```
71CD CDFF71    06640          CALL     IPNT              ;9
               06650 ;
71D0 21263F    06660          LD       HL,3F26H
71D3 CDFF71    06670          CALL     IPNT              ;10
               06680 ;
               06690          IFEQ     TLK,1             ;ASSEMBLE IF TALK VER.
               06700          LD       HL,TRYTD          ;TALK DATA
               06710          CALL     TALK              ;MAKE NOISE
               06720          ENDIF
               06730 ;
71D6 06C8      06740          LD       B,200             ;LOOP COUNTER
71D8 C5        06750 IPLOP    PUSH     BC                ;DO TRAVELING "HIGH SCORES"
71D9 ED5B7C72  06760          LD       DE,(SCPNT)        ;DATA POSITION
71DD D5        06770          PUSH     DE
71DE 21003C    06780          LD       HL,3C00H          ;START SCREEN POSITION
71E1 CD2872    06790          CALL     SCRITE            ;ACROSS THE TOP
71E4 CD5272    06800          CALL     SCDOWN            ;DOWN THE RIGHT SIDE
71E7 CD3D72    06810          CALL     SCLEFT            ;ACROSS TH BOTTOM
71EA CD6772    06820          CALL     SCUP              ;UP THE LEFT SIDE
71ED D1        06830          POP      DE
71EE 13        06840          INC      DE                ;NEW DATA START
71EF 1A        06850          LD       A,(DE)
71F0 B7        06860          OR       A                 ;END OF DATA ?
71F1 CC7E72    06870          CALL     Z,SCFIX           ;CALL IF YES
71F4 ED537C72  06880          LD       (SCPNT),DE        ;SAVE NEW START
71F8 CD6A6E    06890          CALL     DELAY8            ;PAUSE
71FB C1        06900          POP      BC                ;GET LOOP COUNT
71FC 10DA      06910          DJNZ     IPLOP
71FE C9        06920          RET
               06930 ;
71FF FD7E03    06940 IPNT     LD       A,(IY+3)          ;GET FIRST INITIAL
7202 77        06950          LD       (HL),A            ;ON SCREEN
7203 23        06960          INC      HL
7204 FD7E04    06970          LD       A,(IY+4)          ;SECOND INITIAL
7207 77        06980          LD       (HL),A
7208 23        06990          INC      HL
7209 FD7E05    07000          LD       A,(IY+5)          ;THIRD INITIAL
720C 77        07010          LD       (HL),A
720D 23        07020          INC      HL
720E 23        07030          INC      HL
720F E5        07040          PUSH     HL
7210 DDE1      07050          POP      IX
7212 FDE5      07060          PUSH     IY
7214 FD6E00    07070          LD       L,(IY)            ;GET SCORE (BINARY)
7217 FD6601    07080          LD       H,(IY+1)
721A FD7E02    07090          LD       A,(IY+2)
721D CD3951    07100          CALL     BXDEC2            ;CONVERT TO DECIMAL ASCII
               07110                                     ;AND PUT ON SCREEN
7220 FDE1      07120          POP      IY
7222 010600    07130          LD       BC,6              ;ENTRY OFFSET
7225 FD09      07140          ADD      IY,BC             ;NEXT ENTRY
7227 C9        07150          RET
               07160 ;
7228 0640      07170 SCRITE   LD       B,64              ;64 COLUMNS
722A 1A        07180 SCRLOP   LD       A,(DE)            ;GET DATA
722B B7        07190          OR       A                 ;END OF DATA ?
```

```
722C CC7E72    07200          CALL    Z,SCFIX         ;CALL IF YES
722F 28F9      07210          JR      Z,SCRLOP        ;JUMP IF YES
               07220 ;
7231 77        07230          LD      (HL),A          ;PUT ON SCREEN
7232 23        07240          INC     HL              ;UPDATE SCREEN POINTER
7233 13        07250          INC     DE              ;UPDATE DATA POINTER
7234 10F4      07260          DJNZ    SCRLOP          ;LOOP FOR COLUMN COUNT
7236 D5        07270          PUSH    DE
7237 2B        07280          DEC     HL
7238 CD445D    07290          CALL    AD64
723B D1        07300          POP     DE
723C C9        07310          RET
               07320 ;
723D 0640      07330 SCLEFT   LD      B,64            ;64 COLUMNS
723F 1A        07340 SCLLOP   LD      A,(DE)          ;GET DATA
7240 B7        07350          OR      A               ;END OF DATA ?
7241 CC7E72    07360          CALL    Z,SCFIX         ;CALL IF YES
7244 28F9      07370          JR      Z,SCLLOP        ;JUMP IF YES
               07380 ;
7246 77        07390          LD      (HL),A          ;PUT ON SCREEN
7247 2B        07400          DEC     HL
7248 13        07410          INC     DE
7249 10F4      07420          DJNZ    SCLLOP
724B D5        07430          PUSH    DE
724C 23        07440          INC     HL
724D CD4B5D    07450          CALL    SB64
7250 D1        07460          POP     DE
7251 C9        07470          RET
               07480 ;
7252 060E      07490 SCDOWN   LD      B,14            ;ROW COUNT
7254 C5        07500 SCDLOP   PUSH    BC
7255 1A        07510 SCDLP2   LD      A,(DE)          ;GET DATA
7256 B7        07520          OR      A               ;END OF DATA ?
7257 CC7E72    07530          CALL    Z,SCFIX         ;CALL IF YES
725A 28F9      07540          JR      Z,SCDLP2        ;JUMP IF YES
               07550 ;
725C 77        07560          LD      (HL),A          ;PUT ON SCREEN
725D 13        07570          INC     DE
725E D5        07580          PUSH    DE
725F CD445D    07590          CALL    AD64
7262 D1        07600          POP     DE
7263 C1        07610          POP     BC
7264 10EE      07620          DJNZ    SCDLOP
7266 C9        07630          RET
               07640 ;
7267 060E      07650 SCUP     LD      B,14            ;ROW COUNT
7269 C5        07660 SCULOP   PUSH    BC
726A 1A        07670 SCULP2   LD      A,(DE)          ;GET DATA
726B B7        07680          OR      A               ;END OF DATA ?
726C CC7E72    07690          CALL    Z,SCFIX         ;CALL IF YES
726F 28F9      07700          JR      Z,SCULP2        ;JUMP IF YES
               07710 ;
7271 77        07720          LD      (HL),A          ;PUT ON SCREEN
7272 13        07730          INC     DE
7273 D5        07740          PUSH    DE
7274 CD4B5D    07750          CALL    SB64
```

```
7277 D1         07760          POP     DE
7278 C1         07770          POP     BC
7279 10EE       07780          DJNZ    SCULOP
727B C9         07790          RET
                07800 ;
727C 7B71       07810 SCPNT    DW      TENMSG          ;DATA POINTER STORAGE
                07820 ;
727E 117B71     07830 SCFIX    LD      DE,TENMSG
7281 C9         07840          RET
                07850 ;
                07860 ;*************************
                07870 ;**      KEYS ROUTINE    **
                07880 ;*************************
                07890 ;
7282 143C       07900 KEYMSG   DW      3C14H           ;SCREEN ADDRESS
7284 47         07910          DM      'G A M E   C O N T R O L S'
729D 00         07920          DB      0               ;END OF MESSAGE FLAG
                07930 ;
729E 933C       07940          DW      3C93H
72A0 4B         07950          DM      'Key       Function'
72B2 00         07960          DB      0
                07970 ;
72B3 D33C       07980          DW      3CD3H
72B5 3D         07990          DB      '===       ========'
72C7 00         08000          DB      0
                08010 ;
72C8 543D       08020          DW      3D54H
72CA 31         08030          DM      '1.......Change Direction'
72E2 00         08040          DB      0
                08050 ;
72E3 943D       08060          DW      3D94H
72E5 32         08070          DM      '2.......Move Up'
72F4 00         08080          DB      0
                08090 ;
72F5 D43D       08100          DW      3DD4H
72F7 33         08110          DM      '3.......Move Down'
7308 00         08120          DB      0
                08130 ;
7309 143E       08140          DW      3E14H
730B 38         08150          DM      '8.......Move Forward'
731F 00         08160          DB      0
                08170 ;
7320 543E       08180          DW      3E54H
7322 39         08190          DM      '9.......Fire'
732E 00         08200          DB      0
                08210 ;
732F D03E       08220          DW      3ED0H
7331 53         08230          DM      'SPACE BAR...Smart Bomb'
7347 00         08240          DB      0
                08250 ;
7348 503F       08260          DW      3F50H
734A 45         08270          DM      'ENTER KEY...Hyper Space'
7361 00         08280          DB      0
                08290 ;
7362 CD455C     08300 KEYS     CALL    CLS             ;CLEAR THE SCREEN
                08310 ;
```

```
              08320         IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
              08330         LD      HL,CONTD        ;TALK DATA
              08340         CALL    TALK            ;MAKE NOISE
              08350         ENDIF
              08360 ;
7365 218272   08370         LD      HL,KEYMSG       ;FIRST MESSAGE
7368 060A     08380         LD      B,10            ;MESSAGE COUNT
              08390 ;
736A CDA666   08400 KEYLOP  CALL    PRINT2          ;PRINT MESSAGE
736D CD146C   08410         CALL    DELAY6          ;PAUSE
7370 10F8     08420         DJNZ    KEYLOP          ;LOOP FOR MESSAGE COUNT
              08430 ;
              08440         IFEQ    TLK,1           ;ASSEMBLE IF TALK VER.
              08450         LD      HL,PRESSD       ;TALK DATA
              08460         CALL    TALK            ;MAKE NOISE
              08470         ENDIF
              08480 ;
7372 063C     08490         LD      B,60
7374 CD146C   08500 KLOP1   CALL    DELAY6          ;WAIT AWHILE
7377 10FB     08510         DJNZ    KLOP1
7379 C9       08520         RET
              08530 ;
              01950 ;
              01960         IFEQ    TAPE,0          ;ASSEMBLE IF DISK VERSION
737A          01970 *GET    DISK/ASM                ;DISK I/O ROUTINES
              01980         ENDIF
              01990 ;
737A          02000 *GET    TABLES/ASM              ;GAME/PLAYER VARIABLE STORAGE AREA
              00010 ;*********************************
              00020 ;**     CHARACTER TABLES        **
              00030 ;*********************************
              00040 ;
              00050 ;*********************************
              00060 ;**     PLAYER 1 TABLES         **
              00070 ;*********************************
              00080 ;
737A 0A       00090 TABLE1  DB      10              ;ACTIVE LANDERS
737B 0F       00100         DB      15              ;ACTIVE MEN
737C 00       00110         DB      0               ;ACTIVE MUTANTS
737D 04       00120         DB      4               ;MAX CRUISERS
737E 00       00130         DB      0               ;ACTIVE CRUISERS
737F 00       00140         DB      0               ;ACTIVE BOMBERS
              00150 ;
7380 00       00160 FI1CNT  DB      0               ;SHOTS FIRED
7381 0000     00170         DW      00              ;DELAY VALUE
7383 0000     00180         DW      00              ;10,000 COUNTER
7385 00       00190         DB      0               ;WAVE COUNTER
7386 03       00200 S1PCNT  DB      3               ;SHIP COUNTER
7387 03       00210 S1TCNT  DB      3               ;SMART BOMB COUNTER
7388 00       00220         DB      0               ;MOUNTAINS ACTIVE FLAG
              00230 ;
              00240 ;*********************************
              00250 ;**     PLAYER 2 TABLES         **
              00260 ;*********************************
              00270 ;
7389 0A       00280 TABLE2  DB      10              ;ACTIVE LANDERS
```

```
738A 0F       00290          DB      15              ;ACTIVE MEN
738D 00       00300          DB      0               ;ACTIVE MUTANTS
738C 04       00310          DB      4               ;MAX CRUISERS
738D 00       00320          DB      0               ;ACTIVE CRUISERS
738E 00       00330          DB      0               ;ACTIVE BOMBERS
              00340 ;
738F 00       00350 FI2CNT   DB      0               ;SHOTS FIRED
7390 0000     00360          DW      00              ;DELAY VALUE
7392 0000     00370          DW      00              ;10,000 COUNTER
7394 00       00380          DB      0               ;WAVE COUNTER
7395 03       00390 S2PCNT   DB      3               ;SHIP COUNTER
7396 03       00400 S2TCNT   DB      3               ;SMART BOMB COUNTER
7397 00       00410          DB      0               ;MOUNTAINS ACTIVE FLAG
              00420 ;
              00430 ;*************************
              00440 ;**      GAME TABLES     **
              00450 ;*************************
              00460 ;
7398 00       00470 FIRCNT   DB      0               ;SHOTS FIRED
7399 0000     00480 DLYCNT   DW      00              ;DELAY VALUE
739B 0000     00490 CMEAGN   DW      00              ;10,000 COUNTER
739D 00       00500 WAVCNT   DB      0               ;WAVE COUNTER
739E 03       00510 SHPCNT   DB      3               ;SHIP COUNTER
739F 03       00520 SMTCNT   DB      3               ;SMART BOMB COUNTER
73A0 00       00530 DRWFLG   DB      0               ;MOUNTAINS ACTIVE FLAG
              00540                                  ;9 BYTES FOR "LDIR"
              00550 ;
              00560 ;*************************
              00570 ;**    GAME VARIABLES    **
              00580 ;
73A1 05       00590 SHROW    DB      5               ;SHIP POS. GRID ROW #
73A2 1F       00600 SHCOL    DB      31              ;SHIP POS. GRID COL #
73A3 00       00610 CRYENT   DB      0               ;ENTRY # OF MAN CARRIED
73A4 00       00620 CHRPAS   DB      0               ;CHARACTER HIT
73A5 00       00630 ROWSAV   DB      0               ;ROW NUMBER OF SHOT
73A6 00       00640 SPFLG    DB      0               ;SPACE FLAG
73A7 5F3D     00650 SHPOS    DW      3D5FH           ;SHIP SCREEN POS.
73A9 00       00660 DIRFLG   DB      0               ;MOTION DIRECTION #
73AA 00       00670 DIRF2    DB      0               ;SHIP DIRECTION   #
73AB 04       00680 UPDNF    DB      4               ;SHIP UP/DOWN DELAY
73AC 80       00690 SHIPF    DB      128,153,140,128 ;FORWARD SHIP DATA
73B0 80       00700 SHIPB    DB      128,140,166,128 ;BACKWARD SHIP DATA
73B4 00       00710 CNGFLG   DB      0               ;CHANGE DIRECTION FLAG
73B5 8C       00720 MVCHR    DB      140             ;FLAME CHARACTER
73B6 00       00730 MVFLG3   DB      0               ;MOTION FLAG
73B7 03       00740 MVFLG4   DB      3               ;SHIP MOTION DELAY
73B8 0A       00750 SPDCNT   DB      10,10           ;SPEED COUNTER
73BA 0046     00760 DPOINT   DW      MOUNT           ;MOUNTAIN DATA POINTER STORAGE
73BC 00       00770 WINDOW   DB      00              ;SCREEN POSITION ON GRID
              00780 ;
              00790 ;*************************
              00800 ;**      LANDER TABLE    **
              00810 ;
73BD 00       00820          DB      0               ;ACTIVE ENTRIES
73BE 0A       00830 LNDTBL   DB      10              ;MAX ENTRIES
              00840 ;
```

```
73BF 00        00850            DB      0                   ;ROW #
73C0 00        00860            DB      0                   ;COLUMN #
73C1 0F        00870            DB      15                  ;MOVE DELAY
73C2 80        00880            DB      80H                 ;FLAGS
73C3 00        00890            DB      0                   ;WARP-IN PHASE COUNTER
73C4 00        00900            DB      0                   ;SHOT DELAY
               00910 ;
73C5 00        00920            DB      0,0,15,80H,0,0      ;2
73CB 00        00930            DB      0,0,15,80H,0,0      ;3
73D1 00        00940            DB      0,0,15,80H,0,0      ;4
73D7 00        00950            DB      0,0,15,80H,0,0      ;5
73DD 00        00960            DB      0,0,15,80H,0,0      ;6
73E3 00        00970            DB      0,0,15,80H,0,0      ;7
73E9 00        00980            DB      0,0,15,80H,0,0      ;8
73EF 00        00990            DB      0,0,15,80H,0,0      ;9
73F5 00        01000            DB      0,0,15,80H,0,0      ;10
73FB 00        01010            DB      0,0,15,80H,0,0      ;DUMMY  - 68 BYTES
               01020 ;
               01030 ;**************************
               01040 ;**     MAN TABLE        **
               01050 ;
7401 0A        01060            DB      10                  ;# ACTIVE ENTRIES
7402 0F        01070 MANTBL     DB      15                  ;MAX ENTRIES ALLOWED
               01080 ;
7403 00        01090            DB      0                   ;ROW #
7404 00        01100            DB      0                   ;COLUMN #
7405 0F        01110            DB      15                  ;DELAY BEFORE MOVE DOWN
7406 80        01120            DB      80H                 ;FLAGS
7407 00        01130            DB      0                   ;ENTRY # OF LANDER
7408 00        01140            DB      0                   ;UNUSED
               01150 ;
7409 00        01160            DB      0,0,15,80H,0,0      ;2
740F 00        01170            DB      0,0,15,80H,0,0      ;3
7415 00        01180            DB      0,0,15,80H,0,0      ;4
741B 00        01190            DB      0,0,15,80H,0,0      ;5
7421 00        01200            DB      0,0,15,80H,0,0      ;6
7427 00        01210            DB      0,0,15,80H,0,0      ;7
742D 00        01220            DB      0,0,15,80H,0,0      ;8
7433 00        01230            DB      0,0,15,80H,0,0      ;9
7439 00        01240            DB      0,0,15,80H,0,0      ;10
743F 00        01250            DB      0,0,15,80H,0,0      ;11
7445 00        01260            DB      0,0,15,80H,0,0      ;12
744B 00        01270            DB      0,0,15,80H,0,0      ;13
7451 00        01280            DB      0,0,15,80H,0,0      ;14
7457 00        01290            DB      0,0,15,80H,0,0      ;15
745D 00        01300            DB      0,0,15,80H,0,0      ;DUMMY  - 98 BYTES
               01310 ;
               01320 ;**************************
               01330 ;**     MUTANT TABLE     **
               01340 ;
7463 0F        01350            DB      15                  ;MAX ENTRIES ALLOWED
7464 00        01360 MUTBL      DB      0                   ;# OF ACTIVE ENTRIES
               01370 ;
7465 00        01380            DB      0                   ;ROW #
7466 00        01390            DB      0                   ;COLUMN #
7467 03        01400            DB      3                   ;MOVE DELAY
```

```
7468 80     01410          DB     80H             ;FLAGS
7469 00     01420          DB     0               ;UNUSED
746A 00     01430          DB     0               ;UNUSED
            01440 ;
746B 00     01450          DB     0,0,3,80H,0,0   ;2
7471 00     01460          DB     0,0,3,80H,0,0   ;3
7477 00     01470          DB     0,0,3,80H,0,0   ;4
747D 00     01480          DB     0,0,3,80H,0,0   ;5
7483 00     01490          DB     0,0,3,80H,0,0   ;6
7489 00     01500          DB     0,0,3,80H,0,0   ;7
748F 00     01510          DB     0,0,3,80H,0,0   ;8
7495 00     01520          DB     0,0,3,80H,0,0   ;9
749B 00     01530          DB     0,0,3,80H,0,0   ;10
74A1 00     01540          DB     0,0,3,80H,0,0   ;11
74A7 00     01550          DB     0,0,3,80H,0,0   ;12
74AD 00     01560          DB     0,0,3,80H,0,0   ;13
74B3 00     01570          DB     0,0,3,80H,0,0   ;14
74B9 00     01580          DB     0,0,3,80H,0,0   ;15
74BF 00     01590          DB     0,0,3,80H,0,0   ;DUMMY  - 98 BYTES
            01600 ;
            01610 ;*************************
            01620 ;**      CRUISER TABLE    **    STARTS WITH 4 MAX
            01630 ;                               INCREMENTS TO 8
74C5 04     01640          DB     4               ;MAX ENTRIES ALLOWED
74C6 00     01650 CRUTBL   DB     0               ;ACTIVE ENTRIES
            01660 ;
74C7 00     01670          DB     0               ;ROW #
74C8 00     01680          DB     0               ;COLUMN #
74C9 14     01690          DB     20              ;DELAY BEFORE CHANGE DIR.
74CA 80     01700          DB     80H             ;FLAGS
74CB 00     01710          DB     0               ;WARP-IN PHASE COUNTER
74CC 00     01720          DB     0               ;SHOT DELAY
            01730 ;
74CD 00     01740          DB     0,0,20,80H,0,0  ;2
74D3 00     01750          DB     0,0,20,80H,0,0  ;3
74D9 00     01760          DB     0,0,20,80H,0,0  ;4
74DF 00     01770          DB     0,0,20,80H,0,0  ;5
74E5 00     01780          DB     0,0,20,80H,0,0  ;6
74EB 00     01790          DB     0,0,20,80H,0,0  ;7
74F1 00     01800          DB     0,0,20,80H,0,0  ;8
74F7 00     01810          DB     0,0,20,80H,0,0  ;DUMMY  - 32 BYTES
            01820 ;
            01830 ;*************************
            01840 ;**      BOMBER TABLE     **
            01850 ;
74FD 04     01860          DB     4               ;MAX ALLOWED
74FE 00     01870 BMRTBL   DB     0               ;ACTIVE COUNT
            01880 ;
74FF 00     01890          DB     0               ;ROW #
7500 00     01900          DB     0               ;COLUMN #
7501 0F     01910          DB     15              ;DELAY
7502 80     01920          DB     80H             ;FLAGS
7503 00     01930          DB     0               ;COUNTER
7504 00     01940          DB     0               ;UNUSED
            01950 ;
7505 00     01960          DB     0,0,15,80H,0,0  ;2
```

```
750B 00         01970          DB      0,0,15,80H,0,0  ;3
7511 00         01980          DB      0,0,15,80H,0,0  ;4
7517 00         01990          DB      0,0,15,80H,0,0  ;DUMMY  - 32 BYTES
                02000 ;
                02010 ;**************************
                02020 ;**     BOMB TABLE      **
                02030 ;
751D 14         02040          DB      20              ;MAX ALLOWED
751E 00         02050 BMBTBL   DB      0               ;ACTIVE COUNT
                02060 ;
751F 00         02070          DB      0               ;ROW #
7520 00         02080          DB      0               ;COLUMN #
7521 14         02090          DB      20              ;ON SCREEN DELAY
7522 80         02100          DB      80H             ;FLAGS
7523 00         02110          DB      0               ;UNUSED
7524 00         02120          DB      0               ;UNUSED
                02130 ;
7525 00         02140          DB      0,0,20,80H,0,0  ;2
752B 00         02150          DB      0,0,20,80H,0,0  ;3
7531 00         02160          DB      0,0,20,80H,0,0  ;4
7537 00         02170          DB      0,0,20,80H,0,0  ;5
753D 00         02180          DB      0,0,20,80H,0,0  ;6
7543 00         02190          DB      0,0,20,80H,0,0  ;7
7549 00         02200          DB      0,0,20,80H,0,0  ;8
754F 00         02210          DB      0,0,20,80H,0,0  ;9
7555 00         02220          DB      0,0,20,80H,0,0  ;10
755B 00         02230          DB      0,0,20,80H,0,0  ;11
7561 00         02240          DB      0,0,20,80H,0,0  ;12
7567 00         02250          DB      0,0,20,80H,0,0  ;13
756D 00         02260          DB      0,0,20,80H,0,0  ;14
7573 00         02270          DB      0,0,20,80H,0,0  ;15
7579 00         02280          DB      0,0,20,80H,0,0  ;16
757F 00         02290          DB      0,0,20,80H,0,0  ;17
7585 00         02300          DB      0,0,20,80H,0,0  ;18
758B 00         02310          DB      0,0,20,80H,0,0  ;19
7591 00         02320          DB      0,0,20,80H,0,0  ;20
7597 00         02330          DB      0,0,20,80H,0    ;DUMMY  - 128 BYTES
                02340 ;
                02350 ;**************************
                02360 ;**     SHOT TABLE      **
                02370 ;
759C 06         02380          DB      6               ;MAX ALLOWED
759D 00         02390 SHOTBL   DB      0               ;ACTIVE COUNT
                02400 ;
759E 00         02410          DB      0               ;ROW #
759F 00         02420          DB      0               ;COLUMN #
75A0 0A         02430          DB      10              ;MOVE DELAY
75A1 80         02440          DB      80H             ;FLAGS
75A2 00         02450          DB      0               ;UP/DOWN TO LEFT/RIGHT RATIO
75A3 00         02460          DB      0               ;UNUSED
                02470 ;
75A4 00         02480          DB      0,0,10,80H,0,0  ;2
75AA 00         02490          DB      0,0,10,80H,0,0  ;3
75B0 00         02500          DB      0,0,10,80H,0,0  ;4
75B6 00         02510          DB      0,0,10,80H,0,0  ;5
75BC 00         02520          DB      0,0,10,80H,0,0  ;6
```

```
75C2 00         02530           DB      0,0,10,80H,0,0  ;DUMMY  - 44 BYTES
                02540 ;
                02550 ;*************************
                02560 ;**       POD TABLE       **
                02570 ;
75C8 02         02580           DB      2               ;MAX ALLOWED
75C9 00         02590 PODTBL    DB      0               ;ACTIVE COUNT
                02600 ;
75CA 00         02610           DB      0               ;ROW #
75CB 00         02620           DB      0               ;COLUMN #
75CC 03         02630           DB      3               ;MOVE DELAY
75CD 80         02640           DB      80H             ;FLAGS
75CE 00         02650           DB      0               ;WARP-IN PHASE COUNTER
75CF 00         02660           DB      0               ;SHOT DELAY
                02670 ;
75D0 00         02680           DB      0,0,3,80H,0,0   ;2
75D6 00         02690           DB      0,0,3,80H,0,0   ;DUMMY  - 20 BYTES
                02700 ;
                02710 ;*************************
                02720 ;**       SWARMER TABLE   **
                02730 ;
75DC 0C         02740           DB      12              ;MAX ALLOWED
75DD 00         02750 SRMTBL    DB      0               ;ACTIVE ENTRIES
                02760 ;
75DE 00         02770           DB      0               ;ROW #
75DF 00         02780           DB      0               ;COLUMN #
75E0 03         02790           DB      3               ;DELAY BEFORE SEEK SHIP
75E1 80         02800           DB      80H             ;FLAGS
75E2 00         02810           DB      0               ;UNUSED
75E3 00         02820           DB      0               ;UNUSED
                02830 ;
75E4 00         02840           DB      0,0,3,80H,0,0   ;2
75EA 00         02850           DB      0,0,3,80H,0,0   ;3
75F0 00         02860           DB      0,0,3,80H,0,0   ;4
75F6 00         02870           DB      0,0,3,80H,0,0   ;5
75FC 00         02880           DB      0,0,3,80H,0,0   ;6
7602 00         02890           DB      0,0,3,80H,0,0   ;7
7608 00         02900           DB      0,0,3,80H,0,0   ;8
760E 00         02910           DB      0,0,3,80H,0,0   ;9
7614 00         02920           DB      0,0,3,80H,0,0   ;10
761A 00         02930           DB      0,0,3,80H,0,0   ;11
7620 00         02940           DB      0,0,3,80H,0,0   ;12
7626 00         02950           DB      0,0,3,80H,0,0   ;DUMMY  - 80 BYTES
                02960                                   ;1205 BYTES TOTAL
                02970 ;
                02010 ;
                02020           IFEQ    TLK,1           ;ASSEMBLE IF TALK VERSION
                02030 SOFTD     DS      1024            ;TALK DATA STORAGE AREA
                02040 ALIEND    DS      768
                02050 LARRYD    DS      768
                02060 BONUSD    DS      512
                02070 COWARD    DS      256
                02080 YUP1D     DS      512
                02090 YUP2D     DS      512
                02100 GREATD    DS      512
                02110 YINTD     DS      512
```

```
              02120 BOYD    DS      768
              02130 PRESSD  DS      768
              02140 HOWSD   DS      1024
              02150 CONTD   DS      1024
              02160 TRYTD   DS      1024
              02170         ENDIF
              02180 ;
312D          02190 ZBYTE   EQU     $-MASK          ;GIVES PROGRAM SIZE
4703          02200         END     START
00000 Total errors
```

```
MISOSYS XREF-1.0 00/00/00 00:23:46                                   Page 00001
Origin   Symbolic Label Value Line# Usage    Line#'s of References
FIRE     AD64           5D44 07350 EXPLODE  00110 02930 03170 03430
                                   POLL     02770 03010 03360 04190 04570
                                   FIRE     04970
                                   SHOT     01040
                                   LANDER   00510 01030 01630 01790 03470 04040
                                   CRUISER  00540
                                   BOMBER   01030
                                   POD      00670 02510
                                   POINTS   01680 02280 03530 07290 07590
INTERR   ADDLP          5075 01260 INTERR   01270
WHO      AILOP          6B7B 03490 WHO      03500
WHO      ASK            6685 00310 $MAIN    01790
WHO      ASKLOP         668D 00360 WHO      00410
WHO      ASKMES         6941 02210 WHO      00310
WHO      ASKONE         669A 00430 WHO      00380 00400
POINTS   BFIV           6F50 03860 POINTS   03450 03590
POINTS   BFIVLP         6F5C 03950 POINTS   03990
GAME     BIGLP          4AD2 00300 GAME     00550
INTERR   BIN1           51B1 02980 INTERR   01350
HYPER    BINK           4F93 02590 POLL     03280 04740
                                   POINTS   04050
POLL     BKSKP          5814 03580 POLL     03550
POINTS   BKSPC          711C 06090 POINTS   05850
BOMBER   BMBEXT         63C0 01990 BOMBER   01910 02050
BOMBER   BMBKIL         63C7 02040 BOMBER   01950
BOMBER   BMBLOP         639E 01800 BOMBER   01850 02010
INIT     BMBLP          485D 00990 INIT     01020
TABLES   BMBTBL         751E 02050 INIT     00950
                                   FIRE     01700
                                   BOMBER   01390 01710
BOMBER   BMCAN          6350 01240 BOMBER   01460
POINTS   BMPTM          6C6D 00350 POINTS   01400
BOMBER   BMREX8         62E2 00650 BOMBER   00520
BOMBER   BMREXT         62DB 00600 BOMBER   00490 00660
INIT     BMRIT2         4A7B 03530 INIT     02620
INIT     BMRIT9         49B8 02620 INIT     02600
BOMBER   BMRLOP         629D 00300 BOMBER   00340 00620
INIT     BMRLP          4824 00730 INIT     00760
INIT     BMRLP2         4A8F 03610 INIT     03690
BOMBER   BMRMOV         62E7 00680 BOMBER   00570 00650
INTERR   BMRON          5240 03820 INTERR   03520
TABLES   BMRTBL         74FE 01870 INIT     00690 01990 03530
                                   INTERR   03820
                                   FIRE     01600
                                   BOMBER   00210
HYPER    BNKEXT         4FAE 02780 HYPER    02720
HYPER    BNKLP1         4F9D 02650 HYPER    02760
HYPER    BNKLP2         4FA0 02670 HYPER    02740
INTERR   BNLP0          51B7 03000 INTERR   03160
INTERR   BNLP1          51BE 03030 INTERR   03070
INTERR   BNMSG2         52D1 04430 INTERR   01450
INTERR   BNSKP1         51C6 03080 INTERR   03050
POINTS   BO5M           6CB7 00650 POINTS   03930
```

```
Origin   Symbolic Label Value Line# Usage    Line#'s of References
BOMBER   BOMB            638B 01710 GAME     00620
BOMBER   BOMBER          628A 00210 GAME     00630
INTERR   BONEXT          5078 01290 INTERR   01240
INTERR   BONLP1          5092 01410 INTERR   01440
INTERR   BONLP3          50A9 01580 INTERR   01610
INTERR   BONLP4          50A5 01560 INTERR   01630
INTERR   BONMSG          52C5 04410 INTERR   01360
INTERR   BONSCR          52BF 04390 INTERR   01340 01470
BOMBER   BRDWN1          6312 00900 BOMBER   00820
BOMBER   BRINIT          6382 01490 BOMBER   00550
BOMBER   BROK1           631E 00980 BOMBER   00860 00920
BOMBER   BRUP1           6308 00840 BOMBER   00960
BOMBER   BUDSKP          632D 01040 BOMBER   00780
POINTS   BUFF            718A 06310 POINTS   05650 06060
FIRE     BUFFER          5C9C 06130 FIRE     05750 05960
INTERR   BXDEC2          5139 02360 INTERR   01800 01850
                                    POINTS   07100
INTERR   BXLP1           5178 02720 INTERR   02780
INTERR   BXLP2           51DB 03190 INTERR   03240
INTERR   BXOUT           5172 02680 INTERR   02640
INTERR   BXOUT2          5187 02800 INTERR   02740 02840
INTERR   BXOUT3          51EB 03260 INTERR   03210 03270
POLL     C1DO            560A 00170 POLL     00070
GAME     CANSKP          4BCF 00910 GAME     00790
CRUISER  CDWN1           627B 01530 CRUISER  01430
CRUISER  CDWN2           627A 01520 CRUISER  01470
GAME     CHARAC          4B03 00570 GAME     00310 02080
                                    HYPER    01120
FIRE     CHRERS          5C0C 04950 FIRE     04830 04860
POINTS   CHRON           6DEA 01920 POINTS   01080 01190 01290 01390 01490 01590
TABLES   CHRPAS          73A4 00620 FIRE     00750
$MAIN    CLDSRT          4732 01400 GAME     00850
POLL     CLEAR           591B 05180 POLL     05030
FIRE     CLEXT           5AAE 02270 FIRE     02110 02150 02410
FIRE     CLLFT           5AB3 02310 FIRE     02120
FIRE     CLLP1           5AA4 02200 FIRE     02250
FIRE     CLLP2           5AB7 02340 FIRE     02390
POLL     CLRLP           5925 05240 POLL     05260
FIRE     CLS             5C45 05400 $MAIN    01540 01680
                                    POINTS   00900 06370 08300
FIRE     CLSKP1          5AAC 02250 FIRE     02230
FIRE     CLSKP2          5ABF 02390 FIRE     02370
WHO      CM2LP           66BB 00700 WHO      00780
WHO      CM2SK2          66C7 00770 WHO      00720 00740
WHO      CM3LP           66D0 00850 WHO      00930
WHO      CM3SK3          66DC 00920 WHO      00870 00890
WHO      CM4LP           66E4 00990 WHO      01070
WHO      CM4SK4          66F0 01060 WHO      01010 01030
TABLES   CMEAGN          739B 00490 INIT     00150
                                    INTERR   00710 00730 00790
                                    POINTS   00780
FIRE     CMLP            5C79 05820 FIRE     05840
FIRE     CMLP2           5C8C 05970 FIRE     06000
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
GAME     CMPLP1         4D49 03110 GAME     03190
GAME     CMPSKP         4D29 02800 GAME     02770
POLL     CNGD2          5771 02510 POLL     02450
POLL     CNGD3          5778 02550 POLL     02510
POLL     CNGDIR         5753 02350 POLL     00180
TABLES   CNGFLG         73B4 00710 POLL     00230 02350 02390
POLL     CNGSKP         5615 00250 POLL     00200
SHOT     CNSK9          5D7E 00360 SHOT     00310
LANDER   CNSKP2         60EA 03880 LANDER   03820
CRUISER  CNSKP3         61A4 00350 CRUISER  00300
POD      CNSKP8         656A 02410 POD      02360
FIRE     COLCVT         5AFC 02940 FIRE     02650 03130
WHO      COM2LP         66B8 00660 WHO      01110
GAME     COMP           4D21 02750 GAME     02850 03140
                                   HYPER    00590
WHO      COMP2          66B2 00630 WHO      00250 03410
POINTS   CONLP1         6DF2 01960 POINTS   02020
FIRE     CRLOP          5B91 04190 FIRE     04330
POINTS   CRPTM          6C53 00250 POINTS   01300
INIT     CRUILP         4808 00610 INIT     00650
CRUISER  CRUISE         6183 00190 GAME     00610
INIT     CRUIT2         4A4A 03310 INIT     02570
INIT     CRUIT9         49AD 02570 INIT     02540
INIT     CRULP2         4A5D 03390 INIT     03480
FIRE     CRUNCH         5B7F 04100 SHOT     00320
                                   LANDER   03840
                                   CRUISER  00310
                                   BOMBER   00330 01840
                                   POD      00340 02370
TABLES   CRUTBL         74C6 01650 INIT     00570 01960 03310
                                   GAME     03220
                                   INTERR   03710
                                   FIRE     01500
                                   CRUISER  00190
TABLES   CRYENT         73A3 00610 POLL     03190 04770
BOMBER   CSKP10         63AC 01880 BOMBER   01820
BOMBER   CSKP11         62AB 00370 BOMBER   00320
CRUISER  CUDCK          6223 01040 CRUISER  00900 00990
CRUISER  CUDIR          6230 01100 CRUISER  01050
CRUISER  CUDRW          61D1 00560 CRUISER  00490
CRUISER  CUDRW2         61DA 00620 CRUISER  00580
CRUISER  CUEX8          61F5 00770 CRUISER  00380
CRUISER  CULF2          6209 00890 CRUISER  00960
CRUISER  CULF3          620F 00920 CRUISER  01080 01200
CRUISER  CUP1           626D 01450 CRUISER  01600
CRUISER  CUP2           6286 01590 CRUISER  01550
CRUISER  CUPCK          6244 01220 CRUISER  00930 01020
CRUISER  CUPEXT         6276 01490 CRUISER  01570
CRUISER  CUREXT         61DC 00640 CRUISER  00440 00790
CRUISER  CURLOP         6196 00280 CRUISER  00320 00680
CRUISER  CURT1          6214 00950 CRUISER  00830
CRUISER  CURT2          6218 00980 CRUISER  00870
CRUISER  CURT3          621E 01010 CRUISER  01070 01190
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
CRUISER  CUSHOT          61EA 00710 CRUISER  00650
EXPLODE  DELAY           551E 03670 HYPER    01350
                                   EXPLODE  03250
                                   POINTS   01990
WHO      DELAY5          6C07 04610 WHO      00800 00950 01090 03710 03860 03940
                                   WHO      04230 04440 04550
WHO      DELAY6          6C14 04720 GAME     01460 01530 02570 02630
                                   WHO      00220 00270 00470 03390 03400 03490
                                   POINTS   00690
                                   POINTS   00700 00970 00980 01130 01140 01230
                                   POINTS   01240 01330 01340 01430 01440 01530
                                   POINTS   01540 01630 01640 01710 01780 02060
                                   POINTS   02070 03110 05090 08410 08500
POINTS   DELAY7          6E5D 02540 POINTS   02230 02320 02370 02440 02870 02990
                                   POINTS   03160 03170 03370 03460 03500 03510
                                   POINTS   03630 03640 03800
POINTS   DELAY8          6E6A 02650 POINTS   06890
POINTS   DELP            6DCE 01780 POINTS   01790
TABLES   DIRF2           73AA 00670 INIT     01620
                                   EXPLODE  02050
                                   POLL     00080 01380 02400 02420 02470 03480
                                   POLL     04030 04290
                                   FIRE     00170
                                   LANDER   04600
                                   POINTS   01870 02930 03420
TABLES   DIRFLG          73A9 00660 INIT     01610
                                   POLL     01010 01390 01710 02480
                                   POINTS   00750
SHOT     DIRFND          5E1F 01220 SHOT     00380
WHO      DL5LP           6C0C 04640 WHO      04670
WHO      DL6LP           6C19 04750 WHO      04780
POINTS   DL7LP           6E62 02570 POINTS   02600
POINTS   DL8LP           6E6F 02680 POINTS   02710
TABLES   DLYCNT          7399 00480 INIT     00170
                                   INTERR   04150 04200 04240
EXPLODE  DLYLP           5523 03700 EXPLODE  03730
POLL     DOWN            5872 04180 INIT     01520
                                   POLL     00500 04680
POLL     DOWN3           58AE 04560 POLL     03200
HYPER    DOWN4           4F2D 01910 FIRE     00150
TABLES   DPOINT          73BA 00760 INIT     01400
                                   HYPER    00780
                                   POLL     01020 01160 04970
POLL     DRAW            58F9 04920 INIT     01690
                                   GAME     00300
                                   HYPER    00860
                                   POINTS   00960
POLL     DRAW2           5908 05010 POLL     05150
BOMBER   DRPIT           634A 01200 BOMBER   01090
BOMBER   DRPIT2          6359 01280 BOMBER   01220
BOMBER   DRPNO           637D 01450 BOMBER   01410
EXPLODE  DRSKP1          54E4 03240 EXPLODE  03220
EXPLODE  DRSKP2          5506 03490 EXPLODE  03470
```

```
Origin   Symbolic Label Value Line# Usage    Line#'s of References
EXPLODE  DRSKP3          54C8 03000 EXPLODE  02970
TABLES   DRWFLG          73A0 00530 $MAIN    01510
                                    INIT     00080 02330
                                    GAME     02910
                                    INTERR   00990
                                    POLL     04920
                                    LANDER   02760
$MAIN    DSKFLG          4700 00980 $MAIN    01070 01730
                                    INTERR   04460
HYPER    DSWNP           4F32 01940 HYPER    02010
POLL     DWNCNG          563D 00500 INIT     01530
                                    POLL     00480 03210 04690
POLL     DWNOT2          5893 04350 POLL     04310
FIRE     EMDON           5B4E 03650 FIRE     03610
FIRE     EMDON2          5B77 03970 FIRE     03900
FIRE     EMPLP           5B47 03600 FIRE     03630
FIRE     EMPLP2          5B6A 03890 FIRE     03930
FIRE     EMPTY           5B36 03510 INTERR   03720 03840 03970
                                    FIRE     05170
                                    LANDER   02150
                                    BOMBER   01400
                                    POD      01660 01730 01800 01870 01940 02000
FIRE     EMPTY2          5B57 03800 INTERR   04280
EXPLODE  EPHEXT          5434 02100 EXPLODE  01880
POLL     ERASE           5787 02660 INIT     01440
                                    POLL     04110 04660
                                    LANDER   03390
                                    POINTS   02200 02290 02840 03020 03340 03670
                                    POINTS   03770
POLL     ERASE2          5790 02750 POLL     03220
                                    POINTS   03540
POLL     ERLP            578A 02680 POLL     02700
POLL     ERLP2           5799 02800 POLL     02840
FIRE     ERSLP           5C15 05000 FIRE     05050
INTERR   EXCNG0          4FDF 00370 INIT     01550
                                    EXPLODE  00080 01430
EXPLODE  EXCNG1          5357 00570 EXPLODE  00260
EXPLODE  EXCNG2          5369 00720 EXPLODE  00230
EXPLODE  EXCNG3          5384 00950 EXPLODE  00270 00920
EXPLODE  EXCNG4          5394 01070 EXPLODE  00240
EXPLODE  EXCNG5          53A8 01240 EXPLODE  00280
EXPLODE  EXCNG6          53B8 01360 EXPLODE  00250
EXPLODE  EXCNT           55F3 04080 INIT     01670
                                    EXPLODE  00310 00390 00410
EXPLODE  EXCOM1          5371 00830 EXPLODE  00450 01110
EXPLODE  EXDAT1          557B 03950 EXPLODE  00470
EXPLODE  EXDN            530C 00160 EXPLODE  00130
EXPLODE  EXDN1           5316 00230 EXPLODE  00190
EXPLODE  EXDON1          539B 01160 EXPLODE  00420 01400
EXPLODE  EXLP1           534A 00480 EXPLODE  00780
EXPLODE  EXLP2           5350 00520 EXPLODE  00660
EXPLODE  EXLP6           5377 00870 EXPLODE  01020
EXPLODE  EXPHLP          5419 01980 EXPLODE  02010
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
EXPLODE  EXPHOT         5420 02030 EXPLODE  01990
FIRE     EXPL1          5978 00460 GAME     02350
                                   FIRE     00330 00620
FIRE     EXPL2          599F 00710 FIRE     00470
EXPLODE  EXPLD          532D 00350 EXPLODE  00070
                                   POINTS   02450
EXPLODE  EXPLOD         52F5 00070 INIT     01540
                                   INTERR   00370
                                   EXPLODE  01420
                                   POINTS   02400
EXPLODE  EXPLP2         53A1 01200 EXPLODE  01310
EXPLODE  EXPPLP         53D3 01580 EXPLODE  01690
EXPLODE  EXPPOT         53EE 01710 EXPLODE  01590
EXPLODE  EXPSH          53FC 01840 FIRE     00720
EXPLODE  EXTBL          5437 02180 INIT     01590
                                   EXPLODE  00090 00350 01710 01850
                                   POINTS   00830
POLL     F1DO           5655 00680 POLL     00660
TABLES   FI1CNT         7380 00160 INIT     01800 02080
TABLES   FI2CNT         738F 00350 INIT     01840 02120
TABLES   FIRCNT         7398 00470 INIT     00070 01870 02150
                                   INTERR   04170
                                   FIRE     00110 00130
FIRE     FIRE           5938 00050 POLL     00690
                                   POINTS   02360
FIRE     FIREB          597F 00510 FIRE     00190
FIRE     FIREXT         5973 00420 FIRE     00490 00690
FIRE     FIRLP1         595F 00280 FIRE     00400
FIRE     FIRLP2         5989 00570 FIRE     00680
FIRE     FIRSK1         5969 00350 FIRE     00310
FIRE     FIRSK2         5993 00630 FIRE     00600
POINTS   FIVE           6EBE 03140 POINTS   01720
POINTS   FIVLP1         6EC0 03150 POINTS   03250
GAME     FLASH          4C05 01200 GAME     01160
                                   INTERR   01710
GAME     FLASH2         4C17 01400 GAME     01360
GAME     FLCLR          4BA0 00740 GAME     01500
GAME     FLSHLP         4C19 01410 GAME     01580
POD      FOOL           665D 03670 POD      03070 03090
GAME     GAME           4AB8 00150 GAME     00200
                                   INTERR   01730
GAME     GAME2          4AC2 00220 GAME     00130 03310
                                   HYPER    01170
GAME     GAMES          4AA9 00050 GAME     00100 01180
FIRE     GMECLR         5C53 05530 INIT     00810
                                   GAME     02110 02440 03270
                                   HYPER    00840
                                   INTERR   01160
POINTS   HIGHER         70E1 05600 POINTS   05290
POINTS   HILP1          708C 05200 POINTS   05310
INTERR   HIT            4FED 00510 INTERR   01320
                                   POLL     03270 04730
                                   FIRE     01980
```

```
Origin  Symbolic Label Value Line# Usage   Line#'s of References
                                   POINTS   04010
HYPER    HPCP1          4E53 01330 HYPER    01300
HYPER    HPCP2          4E49 01260 HYPER    00500
HYPER    HPLLP          4F1C 01720 HYPER    01800
HYPER    HPLP1          4DC7 00560 HYPER    00640
HYPER    HPLPCP         4E4B 01280 HYPER    01340
GAME     HPSKP          4AF1 00460 GAME     00400
HYPER    HYPDRW         4E3D 01190 HYPER    00300 00330 00360 00390 00420 00460
                                   HYPER    00950 00980 01010 01040 01070 01100
HYPER    HYPDT1         4E5B 01410 HYPER    00290 01090
HYPER    HYPDT2         4E79 01450 HYPER    00320 01060
HYPER    HYPDT3         4E97 01490 HYPER    00350 01030
HYPER    HYPDT4         4EB5 01530 HYPER    00380 01000
HYPER    HYPDT5         4ED3 01570 HYPER    00410 00970
HYPER    HYPDT6         4EF1 01610 HYPER    00440
HYPER    HYPER          4D74 00050 GAME     00380
GAME     HYPFLG         4CC0 02410 GAME     00430
                                   HYPER    00050 00090
HYPER    HYPTNE         4F0F 01650 HYPER    01230
$MAIN    ILOOP          4758 01580 $MAIN    01630
POINTS   INEXT          7118 06060 POINTS   05820
INIT     INIT2          4830 00810 GAME     01150
                                   INTERR   01700
INIT     INIT3          4943 02030 GAME     00990
                                   INTERR   01680
POINTS   INLOP          70EF 05710 POINTS   05730
POINTS   INPUT          70E6 05640 POINTS   04840
INTERR   INTERR         4FB4 00120 $MAIN    01130
INTERR   INTEX2         4FEA 00450 INTERR   00170
INTERR   INTEXT         4FE3 00400 INTERR   00230
POINTS   IPLOP          71D8 06750 POINTS   06910
POINTS   IPNT           71FF 06940 POINTS   06400 06430 06460 06490 06520 06550
                                   POINTS   06580 06610 06640 06670
INTERR   JUMP2          5159 02510 INTERR   02470
POINTS   KEYLOP         736A 08400 POINTS   08420
POINTS   KEYMSG         7282 07900 POINTS   08370
POINTS   KEYS           7362 08300 $MAIN    01620
LANDER   KILSKP         60BA 03440 LANDER   03310
POINTS   KLOP1          7374 08500 POINTS   08510
LANDER   LANDER         5E6F 00210 GAME     00590
LANDER   LDWNOK         5F96 01780 LANDER   01570
FIRE     LENCNT         5BD4 04510 FIRE     00430 02170 02310
INIT     LLP3           49D0 02750 INIT     02840
LANDER   LLSK2          5F34 01290 LANDER   01420
LANDER   LNCLP1         5F03 00970 LANDER   00990
LANDER   LNCLP2         5F16 01070 LANDER   01110
LANDER   LNCLR          5EFF 00940 LANDER   00500
LANDER   LNCLSK         5F1E 01130 LANDER   01100
LANDER   LNDEX8         5EE4 00760 LANDER   00450 00710
LANDER   LNDEX9         5EEB 00780 LANDER   00320 00660
INIT     LNDIT          49BC 02670 INIT     00210 00340 02390
                                   GAME     03260
LANDER   LNDLOP         5E80 00290 LANDER   00810
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
INTERR   LNDON          5282 04110 INTERR   00300
LANDER   LNDRW          5ED1 00650 LANDER   00550
TABLES   LNDTBL         73BE 00830 INIT     01900 02180 02670
                                   INTERR   01660 04110 04260
                                   FIRE     00810 01270
                                   LANDER   00210 01970 01990 02500
LANDER   LNDUP          5F9D 01820 LANDER   01450 01520
LANDER   LNDWN          5F63 01540 LANDER   01470
LANDER   LNEX5          5ECB 00620 LANDER   00590
FIRE     LNHIT2         59DD 01010 FIRE     00920
INIT     LNIT9          498C 02390 INIT     02360
LANDER   LNLF2          5F2D 01260 LANDER   01370
LANDER   LNMAN          60C7 03510 LANDER   00520 00730 01640 01890
POINTS   LNPTM          6C21 00050 POINTS   01100
LANDER   LNRT1          5F42 01360 LANDER   01200
LANDER   LNRT2          5F46 01390 LANDER   01240
LANDER   LNSHOT         5EF4 00860 LANDER   00770
INTERR   LOOP2          5145 02400 INTERR   02660
INTERR   LOOP3          5150 02440 INTERR   02490
LANDER   LRMV           5F20 01180 LANDER   00620
INIT     LXLP           496C 02220 INIT     02240
INIT     MALP1          4A02 02970 INIT     03060
LANDER   MAN            6033 02760 GAME     00600
LANDER   MANDRW         607E 03130 LANDER   03060
LANDER   MANEX8         6087 03200 LANDER   03030
LANDER   MANEXT         6080 03150 LANDER   02930 02960 02990 03110 03210 03230
INIT     MANIT          49EE 02890 INIT     00230 00360 02410
LANDER   MANLOP         604E 02900 LANDER   03170
LANDER   MANMOV         6092 03250 LANDER   03090 03220
TABLES   MANTBL         7402 01070 INIT     01920 02260 02890
                                   INTERR   01100
                                   POLL     03100 03160 04760
                                   FIRE     00900 01190
                                   LANDER   01650 01690 02060 02270 02800 03330
                                   LANDER   03350
$MAIN    MASK           44FF 00250 $MAIN    01050 01400 01530
                                   INIT     01580
                                   GAME     00220 00260 00280 01750 01770 02260
                                   GAME     02280 02930 02950
                                   HYPER    00110
                                   INTERR   00150 00950 00970 01080 02000 02020
                                   INTERR   04600 04650
                                   EXPLODE  01750 01770 02070 02090 02500
                                   FIRE     04100 04120 04400 04420 05530 05560
                                   FIRE     05660
                                   $MAIN    02190
HYPER    MCNGSND        4F7E 02450 HYPER    02350 02420
HYPER    MDO1           4F5C 02220 HYPER    02380
HYPER    MDO2           4F64 02260 HYPER    02310
HYPER    MDOWN1         4F74 02370 HYPER    02280
HYPER    MDOWN2         4F6C 02300 HYPER    02200 02240 02430
POLL     MI2            5898 04380 POLL     04090 04320
POLL     MINUS          5694 01210 POLL     01060
```

```
Origin    Symbolic Label Value Line# Usage    Line#'s of References
POLL      MNDLP          5931 05360 POLL     05390
POLL      MNDLY          592D 05340 POLL     04940
POLL      MNFLP          57AF 03030 POLL     03070
POLL      MNFND          57B9 03100 POLL     03060
POINTS    MOB4           6F3B 03740 POINTS   03820
POINTS    MOBLP1         6E7F 02830 POINTS   02910
POINTS    MOBLP2         6E9C 02980 POINTS   03060
POINTS    MOF1           6EDC 03310 POINTS   03390
POINTS    MOF2           6F00 03490 POINTS   03580
POINTS    MOF3           6F1F 03620 POINTS   03710
INIT      MOGO           4919 01860 INIT     01810
INIT      MOGO2          4959 02140 INIT     02090
HYPER     MOOSE          4F41 02070 GAME     00120
HYPER     MOOSELP        4F49 02140 HYPER    02510
POINTS    MOSLP1         6E10 02130 POINTS   02340
POINTS    MOSLP2         6E16 02170 POINTS   02250
POINTS    MOSLP3         6E4B 02430 POINTS   02470
POINTS    MOSUBB         6E77 02780 POINTS   01120 01220 01320 01420 01520 01620
                                    POINTS   01750
POINTS    MOSUBF         6ED6 03280 POINTS   01740
$MAIN     MOUNT          4600 00700 INIT     01390
                                    HYPER    00750
                                    TABLES   00760
POLL      MOVE           5677 01010 POLL     01500 01620 02020 02230
POLL      MOVE3          56D9 01620 POLL     01560
POINTS    MOVF           6E08 02090 POINTS   01730
LANDER    MOVLND         5FB0 01940 LANDER   01850
LANDER    MSFND          6018 02440 LANDER   02350
GAME      MSGEND         4CF9 02710 GAME     02500
LANDER    MSRCH          5FF7 02270 LANDER   00480
LANDER    MSRLP          6001 02310 LANDER   02380
LANDER    MSRSKP         600D 02370 LANDER   02320
HYPER     MTONE          4F8C 02530 HYPER    02190 02230 02270
HYPER     MTONE1         4F91 02570 HYPER    02090
FIRE      MUCLR          5CD7 06510 SHOT     00450
                                    LANDER   03080 03930
                                    CRUISER  00460
                                    BOMBER   00560 01930
                                    POD      00570 02460
FIRE      MUCLRP         5CDD 06560 FIRE     06600
FIRE      MUCLSK         5CE5 06620 FIRE     06590
LANDER    MUDOK          6160 04540 LANDER   04410 04490
LANDER    MUDRW          6123 04150 LANDER   03980 04070 04110
LANDER    MUDWN          6152 04440 LANDER   04340
LANDER    MUDWN1         6151 04430 LANDER   04380
LANDER    MUEX8          612C 04220 LANDER   03910
INIT      MUILP          47F0 00500 INIT     00530
INIT      MUILP2         4A34 03190 INIT     03260
LANDER    MUMOV          6131 04250 LANDER   03960 04220
POINTS    MUPTM          6C3A 00150 POINTS   01200
LANDER    MUSKP5         6177 04670 LANDER   04620
LANDER    MUSKP7         617E 04700 LANDER   04650
LANDER    MUTANT         60C9 03710 GAME     00570
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
TABLES   MUTBL          7464 01360 INIT     00430 01940 03110
                                   FIRE     01400
                                   LANDER   02140 03710
LANDER   MUTEXT         6125 04170 LANDER   04230
INIT     MUTIT          47DE 00430 INIT     00190
INIT     MUTIT2         4A20 03110 INIT     02490
                                   GAME     03240
INIT     MUTIT9         499E 02490 INIT     02460
LANDER   MUTLOP         60DC 03800 LANDER   03850 04190
LANDER   MUUP           6146 04360 LANDER   04520
LANDER   MUUP1          615D 04510 LANDER   04460
POLL     MVBK           583A 03810 POLL     03770
TABLES   MVCHR          73B5 00720 POLL     03710 03730
POLL     MVCMD1         571A 02020 POLL     01780 01860
POLL     MVCNG          5669 00810 INIT     01490
                                   POLL     01310 01570 02030 02090 02520 02610
POLL     MVCNG2         5673 00920 INIT     01510
                                   POLL     01330 02110 02280
TABLES   MVFLG3         73B6 00730 INIT     01600
                                   POLL     00880 01350 02300 02430 02560 03680
                                   POLL     04000 04260
                                   POINTS   00810 02110 03090
TABLES   MVFLG4         73B7 00740 POLL     01370 01630 01650 01680
POLL     MVSKP          5662 00770 POLL     00710
INIT     MXLP           4979 02280 INIT     02300
SHOT     NOKOFF         5E6A 01630 SHOT     01560
FIRE     NOMAT          5AF5 02820 FIRE     02730 02760 02790
FIRE     NOMAT2         5B24 03230 FIRE     03170 03200
FIRE     NONE           5AFA 02850 FIRE     02620
LANDER   NOSP           5FF0 02210 LANDER   02160
HYPER    NUM            4F2C 01850 HYPER    00190 01670 01700
FIRE     NXT1           59F8 01140 FIRE     00770
FIRE     NXT2           5A22 01350 FIRE     01150
FIRE     NXT3           5A32 01450 FIRE     01360
FIRE     NXT4           5A42 01550 FIRE     01460
FIRE     NXT5           5A52 01650 FIRE     01560
FIRE     NXT6           5A5E 01740 FIRE     01660
FIRE     NXT7           5A6E 01840 FIRE     01750
FIRE     NXTMN1         5C80 05880 FIRE     06070
LANDER   OFFSET         60C6 03500 FIRE     04690 04780
FIRE     ONCHK          5BD7 04590 SHOT     00410
                                   LANDER   00340 03010 03890
                                   CRUISER  00360
                                   BOMBER   00380 01890
                                   POD      00390 02420
FIRE     ONLP1          5BF4 04740 FIRE     04760
GAME     OVCLR          4D0D 02730 GAME     02600
GAME     OVER           4CC2 02440 GAME     00810
GAME     OVLP1          4CD8 02520 GAME     02670
WHO      P3SK1          6B94 03680 WHO      03630 03650
WHO      P3SK2          6BA9 03830 WHO      03780 03800
WHO      P4SK1          6BD4 04180 WHO      04110 04130
WHO      P4SK2          6BEE 04390 WHO      04320 04340
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
POD     PDINIT          6468 00960 POD      00530
POINTS  PDPTM           6C86 00450 POINTS   01500
POD     PDSHOT          645D 00900 POD      00780
POD     PDWN            64A5 01270 POD      01170
POLL    PL1             56FF 01830 POLL     01730
INTERR  PL1SCH          51FD 03380 $MAIN    01490 01720
                                   INTERR   00590 00610 01790
                                   POINTS   00800 04400
INTERR  PL1SCL          51FB 03370 $MAIN    01440 01700
                                   INTERR   00560 00580 01780
                                   POINTS   00760 04390
POLL    PL2             58A3 04460 POLL     04060 04350
INTERR  PL2SCH          5200 03410 $MAIN    01500
                                   INTERR   00670 00690 01840
                                   POINTS   04440
INTERR  PL2SCL          51FE 03400 $MAIN    01450
                                   INTERR   00640 00660 01830
                                   POINTS   04430
INTERR  PLAY2           5005 00640 INTERR   00540
INTERR  PLFLG           5201 03430 INIT     00060 00380 01750 02030
                                   GAME     00870 00890 00950 00970 01010 01200
                                   GAME     02170 02460
                                   INTERR   00520 00870
                                   POINTS   00820 04370
POD     PLFT            64D1 01520 POD      01470
POD     PLREX1          64C7 01460 POD      01350
POLL    PLSKP           568C 01150 POLL     01260
POLL    PLSKP5          5710 01930 POLL     01980
GAME    PLYCNT          4CC1 02420 INIT     00260
                                   GAME     00820 00840 00910
                                   WHO      00440
GAME    PLYEND          4D0C 02720 GAME     02480
INIT    PLYTWO          47CC 00330 INIT     00280
POINTS  PNTTBL          6CBC 00670 POINTS   01050
POD     POD             63CD 00200 GAME     00640
POD     PODEX1          63F1 00380 POD      00310
POD     PODEX8          644F 00840 POD      00550
POD     PODEXT          6441 00770 POD      00500 00850 00880
POD     PODLOP          63E0 00290 POD      00350 00810
INIT    PODLP           4875 01100 INIT     01130
POD     PODMOV          6480 01060 POD      00620 00870
INTERR  PODON           525E 03950 INTERR   03550
POD     PODRW           643F 00750 POD      00590 00710
TABLES  PODTBL          75C9 02590 INIT     01060
                                   GAME     02050
                                   INTERR   03950
                                   FIRE     01790
                                   POD      00200
POINTS  POINTS          6CC2 00690 $MAIN    01610
POLL    POLL            55F4 00050 GAME     00540
GAME    POLSKP          4AFE 00540 GAME     00500
CRUISER POSCHK          61FE 00810 CRUISER  00470 00770
POD     POSK4           640C 00520 POD      00410 00470
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
POD      POSK5          6407 00490 POD      00440
WHO      PR2LP          66AA 00550 WHO      00610
WHO      PR3LP          6B85 03570 WHO      03960
WHO      PR3LP1         6B88 03610 WHO      03690
WHO      PR3LP2         6B9D 03760 WHO      03840
WHO      PR3LP3         6BB3 03900 WHO      03920
WHO      PR4LP          6BC3 04040 WHO      04570
WHO      PR4LP1         6BC7 04090 WHO      04210
WHO      PR4LP2         6BE1 04300 WHO      04420
WHO      PR4LP3         6BF9 04490 WHO      04530
FIRE     PRINT          5D52 07550 $MAIN    01770
                                   POINTS   00920
WHO      PRINT2         66A6 00500 WHO      00320
                                   POINTS   01840 04820 04830 08400
WHO      PRINT3         6B81 03530 WHO      00090 00140 03270 03320
WHO      PRINT4         6BBE 03990 WHO      00100 03280
POINTS   PRINT5         6DD6 01830 POINTS   01110 01210 01310 01410 01510 01610
FIRE     PRNT           5D55 07560 INTERR   01380 01460 01480
                                   FIRE     07620
INTERR   PTAB1          51F1 03310 INTERR   02980
INTERR   PTAB2          5191 02870 INTERR   02360
POINTS   PTLP1          6DD8 01840 POINTS   01850
POD      PUDEXT         64B1 01340 POD      01130 01210 01250 01290
SHOT     RATFND         5E5B 01550 SHOT     01490
$MAIN    RELEASE        472B 01330 $MAIN    01350
POINTS   REMES          712A 06210 POINTS   04810
POINTS   REPLCE         7030 04680 POINTS   04620
GAME     RESKP2         4BF6 01140 GAME     01080
FIRE     RNDCOL         5C70 05740 INIT     02730 02950 03170 03370 03590
FIRE     ROWCOL         5CB5 06190 INTERR   03830 03960 04270
GAME     ROWLOP         4C63 01860 GAME     01990
FIRE     ROWPAS         5BD0 04460 EXPLODE  01900
                                   FIRE     02670
FIRE     ROWRND         5CBC 06240 INTERR   03700
                                   FIRE     06270 06290
TABLES   ROWSAV         73A5 00630 GAME     01790 02120
TABLES   S1PCNT         7386 00200 $MAIN    01460
                                   GAME     01050 01070
                                   INTERR   00910 01920
TABLES   S1TCNT         7387 00210 GAME     02220
TABLES   S2PCNT         7395 00390 $MAIN    01470
                                   INIT     00300
                                   GAME     01100 01120
                                   INTERR   00940 01960
TABLES   S2TCNT         7396 00400 GAME     02250
FIRE     SB64           5D4B 07450 EXPLODE  00140 00720 01070 01360 02590
                                   POLL     03930
                                   SHOT     00900
                                   LANDER   01880 04020
                                   CRUISER  00520
                                   BOMBER   01010
                                   POD      00650 02490
                                   POINTS   03010 03200 03660 07450 07750
```

```
Origin   Symbolic Label Value Line# Usage    Line#'s of References
POINTS    SCDLOP         7254 07500 POINTS   07620
POINTS    SCDLP2         7255 07510 POINTS   07540
POINTS    SCDOWN         7252 07490 POINTS   06800
POINTS    SCFIX          727E 07830 POINTS   06870 07200 07360 07530 07690
POINTS    SCLEFT         723D 07330 POINTS   06810
POINTS    SCLLOP         723F 07340 POINTS   07370 07420
GAME      SCMSG          4B5F 00690 GAME     01330
                                    INTERR   01870
SHOT      SCNTSK         5DAB 00560 SHOT     00520
POLL      SCOL1          570C 01910 POLL     01120 01810 04430
POLL      SCOL2          5714 01960 POLL     01230 01890 04510
EXPLODE   SCOP1          54EC 03300 EXPLODE  02790 03480
EXPLODE   SCOP2          54EF 03340 EXPLODE  03400
INTERR    SCORE          50CB 01770 $MAIN    01780
                                    INIT     01700
                                    GAME     01170 02450 03280
                                    HYPER    00850
                                    INTERR   00340 01690 01720
                                    EXPLODE  02510
                                    POINTS   00930 02390 04020
GAME      SCORE1         4B69 00700 INTERR   01770
GAME      SCORE2         4B8D 00720 GAME     01380
                                    INTERR   01820
EXPLODE   SCOSKP         54F7 03390 EXPLODE  03360
POINTS    SCPNT          727C 07810 POINTS   06760 06880
POINTS    SCRITE         7228 07170 POINTS   06790
POINTS    SCRLOP         722A 07180 POINTS   07210 07260
EXPLODE   SCRLP          54B4 02840 EXPLODE  02980
EXPLODE   SCRLP2         54B7 02880 EXPLODE  02900
INTERR    SCRSK1         5014 00710 INTERR   00620
INTERR    SCRSKP         5045 00950 INTERR   00770 00920
POINTS    SCULOP         7269 07660 POINTS   07780
POINTS    SCULP2         726A 07670 POINTS   07700
POINTS    SCUP           7267 07650 POINTS   06820
FIRE      SEARCH         5AC3 02520 FIRE     00830 01410 01510 01610 01710 01800
                                    FIRE     01900
FIRE      SERCH1         5ACC 02590 FIRE     02520
FIRE      SERLP1         5AE0 02710 FIRE     02830
FIRE      SERLP2         5B15 03150 FIRE     03240
FIRE      SERLP4         5B30 03370 FIRE     03400
FIRE      SERSK1         5A84 01980 FIRE     01430 01920
FIRE      SERSKP         5A7D 01940 FIRE     00870 00990 01120 01530 01630 01820
TABLES    SHCOL          73A2 00600 INIT     01330
                                    HYPER    00810
                                    POLL     01910 01930 01960 04810
                                    SHOT     01380
                                    LANDER   02400 04560
                                    CRUISER  00810
                                    POD      02770
EXPLODE   SHDAT1         552B 03800 EXPLODE  02540
EXPLODE   SHDAT2         5533 03830 EXPLODE  02630
EXPLODE   SHDAT3         5553 03880 EXPLODE  02720
SHOT      SHDNCK         5DEF 00930 SHOT     00800
```

```
MISOSYS XREF-1.0 00/00/00 00:23:46                              Page 00014
Origin  Symbolic Label Value Line# Usage    Line#'s of References
SHOT     SHDOWN         5E37 01350 SHOT     01300
EXPLODE  SHDRW          54CB 03020 EXPLODE  02560 02650 02730
EXPLODE  SHDRW1         54CD 03050 EXPLODE  03230
EXPLODE  SHDRW2         54D0 03090 EXPLODE  03130
FIRE     SHEX           5C3A 05220 FIRE     05180
EXPLODE  SHIP           5477 02490 POLL     03880
TABLES   SHIPB          73B0 00700 POLL     03560
TABLES   SHIPF          73AC 00690 POLL     03520
SHOT     SHLF2          5E4D 01480 SHOT     01520
SHOT     SHLFT          5E14 01140 SHOT     01070
SHOT     SHNOUD         5E3B 01370 SHOT     01280 01330
FIRE     SHOCHK         5C1D 05130 LANDER   00900
                                   CRUISER  00750
                                   POD      00940
SHOT     SHODRW         5DB7 00610 SHOT     00470
SHOT     SHOEX8         5DB9 00630 SHOT     00400 00690
SHOT     SHOK1          5DEA 00900 SHOT     00850
SHOT     SHOK2          5E02 01040 SHOT     00990
SHOT     SHOKIL         5DC0 00680 SHOT     00430 00590
SHOT     SHOLOP         5D70 00290 SHOT     00330 00650
SHOT     SHOMOV         5DC6 00710 SHOT     00570
SHOT     SHOMV2         5DD4 00780 SHOT     00730
SHOT     SHOT           5D5D 00190 GAME     00580
TABLES   SHOTBL         759D 02390 INIT     00820
                                   GAME     02150
                                   FIRE     05160
                                   SHOT     00190
INIT     SHOTLP         4845 00880 INIT     00910
TABLES   SHPCNT         739E 00510 INIT     00100
                                   GAME     00770 01140
                                   INTERR   00800 00820 00860
TABLES   SHPOS          73A7 00650 INIT     01290 01710
                                   GAME     00320 03290
                                   HYPER    00240 00880 01130
                                   EXPLODE  02520
                                   POLL     01700 01800 01880 03920 04130 04180
                                   POLL     04560
                                   FIRE     00160
                                   POINTS   00880 00940 01880 02150 02300 02780
                                   POINTS   02960 03000 03040 03280 03440 03520
                                   POINTS   03560 03650 03690 03830
SHOT     SHRICK         5E05 01050 SHOT     00760 00910 00940
TABLES   SHROW          73A1 00590 INIT     01310
                                   GAME     01780 01830 01950 01970 02130
                                   HYPER    00830
                                   EXPLODE  01890
                                   POLL     03940 03980 04200 04240 04580 04600
                                   SHOT     01260
                                   POD      03010
                                   POINTS   00860
SHOT     SHRT1          5E53 01510 SHOT     01420
SHOT     SHRT2          5E57 01540 SHOT     01460
FIRE     SHTBL          5BD5 04520 FIRE     00250 00530 02080
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
FIRE     SHTCLR         5A88 02030 POLL     00540
                                   POINTS   02380
FIRE     SHTDIR         5BD3 04490 FIRE     00220 00520 02090 02280
FIRE     SHTF1          5BD1 04470 POLL     00740
                                   FIRE     00050 00090
                                   POINTS   00770 02500
FIRE     SHTF2          5BD2 04480 FIRE     02030 02050
                                   POINTS   02510
EXPLODE  SID1           5366 00690 EXPLODE  00620
EXPLODE  SID3           5392 01050 EXPLODE  00980
EXPLODE  SID5           53B6 01340 EXPLODE  01270
FIRE     SIDCHK         5C3D 05290 EXPLODE  00610 00970 01260
                                   FIRE     06820 07220
POLL     SLD1           5737 02170 POLL     02150
POLL     SLEXT2         5746 02270 POLL     02210
POLL     SLODWN         572D 02130 POLL     02100 02550
GAME     SMART          4C37 01610 GAME     00480
GAME     SMDON          4CAB 02260 GAME     02230
TABLES   SMTCNT         739F 00520 INIT     00110
                                   GAME     01650 01700 02190
                                   INTERR   00830 00850
GAME     SMTEXT         4C8A 02110 GAME     02030
GAME     SMTFLG         4CBF 02400 GAME     00520 01610 02010 02070
GAME     SMTHIT         4CB4 02310 GAME     01910
GAME     SMTLOP         4C65 01890 GAME     01930
GAME     SMTLP          4C58 01810 GAME     02090
EXPLODE  SO5LP          550F 03550 EXPLODE  03630
EXPLODE  SO5ND          550A 03520 EXPLODE  02810 03240 03490
POINTS   SORLOP         706F 05030 POINTS   05060
POINTS   SORT           707E 05130 POINTS   05040
GAME     SPACE          4D30 02870 LANDER   02830 02840
TABLES   SPDCNT         73B8 00750 INIT     01640 01650
                                   POLL     01410 01530 02130
INTERR   SPDRW          510D 02050 INTERR   01940 01980
POLL     SPDUP          56B9 01410 POLL     01300
POLL     SPDUP2         56C4 01470 POLL     01450
POLL     SPEEDU         569D 01300 INIT     01480
                                   POLL     00810 02080 02600
POLL     SPEXLP         5840 03860 POLL     03610
POLL     SPEXT1         56CC 01530 POLL     01430 01580 02160 02250 02310
POLL     SPEXT2         56D1 01560 POLL     01480
TABLES   SPFLG          73A6 00640 GAME     01730 02160 02870
INTERR   SPLP1          511B 02140 INTERR   02170
INTERR   SPLP2          5128 02220 INTERR   02250
INTERR   SPLP3          5132 02300 INTERR   02330
INTERR   SPNEX1         5121 02180 INTERR   02120 02130
INTERR   SPNEX2         512E 02260 INTERR   02200 02210
FIRE     SPSER1         5B06 03080 POLL     03120
                                   FIRE     01210
FIRE     SPSER2         5B2B 03350 POLL     04780
                                   FIRE     01020 01280
                                   LANDER   02080
FIRE     SPSER3         5B12 03140 FIRE     00960
```

```
Origin   Symbolic Label Value Line# Usage    Line#'s of References
                                    LANDER   01680
FIRE      SREXT2         5B29 03260 FIRE     03110
POINTS    SRLOP1         7078 05090 POINTS   05100
INIT      SRMLP          488D 01210 INIT     01240
TABLES    SRMTBL         75DD 02750 INIT     01170
                                    FIRE     01890
                                    POD      01640 02250
WHO       SSLOP2         667F 00270 WHO      00280
WHO       SSM            66FA 01180 WHO      00050
WHO       SSMLOP         6675 00220 WHO      00230
$MAIN     START          4703 01020 $MAIN    02200
$MAIN     START2         4769 01670 INTERR   04660
POLL      STOP           5721 02080 INIT     01500
                                    POLL     00920 01320 02270
INTERR    STRB2          52E1 04560
INTERR    STROBE         52DD 04460 INTERR   00260
GAME      STRTMS         4B1F 00680 $MAIN    01760
                                    POINTS   00910
POLL      SUPDLP         5816 03590 POLL     03660
POD       SWARM          6549 02250 GAME     00650
POD       SWDN           65FC 03150 POD      03120 03240
POD       SWDN1          6609 03230 POD      03290
POD       SWEKSK         661A 03350 POD      02720
POD       SWLF3          65C2 02860 POD      02910
POD       SWMDRW         6597 02610 POD      02540 02580
POD       SWMEX8         65A0 02680 POD      02440
POD       SWMEXT         6599 02630 POD      02690
POD       SWMLOP         655C 02340 POD      02380 02650
POD       SWMMOV         65A5 02710 POD      02470 02680
POD       SWMNEX         64DA 01590 POD      01670 01740 01810 01880 01950
POD       SWMON          64D5 01570 POD      00330
POD       SWMON2         64DD 01620 POD      01580
POD       SWMUD          65D6 02950 POD      02790 02880
POD       SWNXT1         662E 03440 POD      03400
POD       SWNXT2         6637 03470 POD      03420 03450
POD       SWNXT3         664A 03560 POD      03490
POD       SWNXT4         6659 03640 POD      03520 03600
POINTS    SWPTM          6C9D 00550 POINTS   01600
POD       SWRT1          65CB 02900 POD      02810
POD       SWRT2          65CF 02920 POD      02840
POD       SWSKP2         6623 03390 POD      03360
GAME      SWTSKP         4BDD 00990 GAME     00930
POD       SWUP           660D 03270 POD      03110 03140
POD       SWUP1          660C 03260 POD      03170
POD       SWXYZ          65FA 03140 POD      03050
POLL      T1DO           5668 00800 POLL     00780
INIT      TABINT         4903 01750 INIT     00250
                                    GAME     00760
                                    INTERR   01670
TABLES    TABLE1         737A 00090 INIT     01790 02070
TABLES    TABLE2         7389 00280 INIT     01830 02110
$MAIN     TAPE           0001 00100 $MAIN    00150 00210 01200
                                    INTERR   04490
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
                                   $MAIN    01960
FIRE     TBLPAS         5BCE 04450 FIRE     04370
                                   SHOT     00200
                                   LANDER   03720
                                   CRUISER  00200
                                   BOMBER   00220 01720
                                   POD      00210 02260
POINTS   TCHK           6FAD 04330 POINTS   04460
INTERR   TEMP           51A9 02960 INTERR   02380
POINTS   TENBUF         6FB0 04350 POINTS   04970 05140
POINTS   TENEND         6FA7 04300 POINTS   04680
POINTS   TENFLG         7189 06300 $MAIN    01060 01750
                                   POINTS   06330
POINTS   TENLOP         7016 04550 POINTS   04650
POINTS   TENMSG         717B 06280 POINTS   07810 07830
POINTS   TENPNT         718E 06330 $MAIN    01580
POINTS   TENTBL         6F71 04120 POINTS   04510 04950 06380
INTERR   TIME           5280 04070 INTERR   03470 03490 03620
INTERR   TIMER          5202 03470 INTERR   00290
$MAIN    TLK            0000 00130 $MAIN    01290
                                   GAME     01220 02970
                                   HYPER    00130
                                   INTERR   01500
                                   WHO      00160 03340 03430
                                   POINTS   01000 04760 06690 08320 08440
                                   $MAIN    02020
$MAIN    TLKFLG         4701 00990 $MAIN    01080
HYPER    TNECNG         4F15 01690 HYPER    00220 00920
POINTS   TOPCHK         6FEC 04370 GAME     02680
POINTS   TOPNXT         7000 04460 POINTS   04410
FIRE     TRYAGN         5C7E 05860 FIRE     05980
$MAIN    TSPFLG         4702 01000 $MAIN    01090
INIT     TWO            4912 01830 INIT     01770
INIT     TWO2           4952 02110 INIT     02050
GAME     TWO3           4BEF 01100 GAME     01030
GAME     TWO4           4CA8 02250 GAME     02200
POLL     U1DO           5632 00430 POLL     00410
CRUISER  UDDLY          6289 01620 CRUISER  01290 01320 01360
LANDER   UDMV           5F4F 01440 LANDER   00630
POLL     UDSKP          5640 00540 POLL     00290
POLL     UP             5846 03920 POLL     00440
POLL     UPDATE         57A1 02920 INIT     01470 01720
                                   GAME     00330 03300
                                   POLL     03250 04710
                                   FIRE     01110
                                   LANDER   03380
                                   POINTS   00720 00950 01890 02220 02310 02860
                                   POINTS   02940 03030 03100 03360 03430 03680
                                   POINTS   03790
POLL     UPDEX0         5801 03430 POLL     03080 03140 03290
POLL     UPDEXT         583D 03820 POLL     03700 03790
TABLES   UPDNF          73AB 00680 POLL     00250
POLL     UPDT1          5803 03480 INIT     01460
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
                                   POLL     02920 02980 03330 04700
                                   LANDER   03370
                                   POINTS   00710
POLL     UPDT2          57F1 03330 POLL     03240
                                   POINTS   03550
POLL     UPDT3          57A5 02980 FIRE     01100
POLL     UPOT1          586A 04110 INIT     01450
                                   POLL     03230 04020 04070 04280 04330 04360
                                   POLL     04620 04670 04860
                                   LANDER   03400
POLL     UPOT3          5867 04090 POLL     04050
INIT     VECTOR         48B2 01440 HYPER    01160
POINTS   WAIT           70F6 05780 POINTS   05880 05910 05960 06040 06130 06190
EXPLODE  WARP1          5599 04010 FIRE     07110
EXPLODE  WARP2          55A8 04020 FIRE     07070
EXPLODE  WARP3          55B7 04030 FIRE     07030
EXPLODE  WARP4          55C6 04040 FIRE     06990
EXPLODE  WARP5          55D5 04050 FIRE     06950
EXPLODE  WARP6          55E4 04060 FIRE     06910
FIRE     WARPIN         5CE7 06690 LANDER   00440
                                   CRUISER  00430
                                   BOMBER   00480
                                   POD      00490
                                   POINTS   01980
TABLES   WAVCNT         739D 00500 INIT     00130
                                   INTERR   01030 01090
                                   FIRE     01940 01960
                                   POINTS   00840
FIRE     WEXTX          5D09 06880 FIRE     06830
FIRE     WEXTX2         5D42 07280 FIRE     07230
WHO      WHAT           6B5B 03220 $MAIN    01600
WHO      WHO            665E 00050 $MAIN    01590
WHO      WHOLP          6666 00080 WHO      00120
WHO      WHTDAT         6954 02290 WHO      03220
WHO      WHTLP1         6B63 03260 WHO      03300
TABLES   WINDOW         73BC 00770 INIT     01380
                                   HYPER    00710 00730
                                   INTERR   03660
                                   POLL     01040 01150
                                   FIRE     02970 04620 06380
GAME     WRMSRT         4BB1 00760 INIT     00310 00390
                                   EXPLODE  03000
FIRE     WRPCHK         5CCA 06370 INTERR   03860 03990 04310
FIRE     WRPLP0         5CF8 06770 FIRE     06850
FIRE     WRPLP1         5D30 07150 FIRE     07250
FIRE     WRPXT1         5D0B 06910 FIRE     06720
FIRE     WRPXT2         5D2C 07130 FIRE     06930 06970 07010 07050 07090
CRUISER  WRSK1          61B9 00460 CRUISER  00410
LANDER   WRSK2          5EA3 00470 LANDER   00360 00420
LANDER   WRSK3          5E9E 00440 LANDER   00390
BOMBER   WRSK4          62C6 00510 BOMBER   00400 00460
BOMBER   WRSK5          62C1 00480 BOMBER   00430
INTERR   XX             5042 00940 INTERR   00890
```

```
Origin  Symbolic Label Value Line# Usage    Line#'s of References
$MAIN    ZBYTE          312D 02190
00689 Symbols declared  -  01533 References
```