

Escape

You are in the horrible prison of the Warlord and are attempting to escape with some of your fellow prisoners. But to escape, you must leave through the River of Death by passing through the Gates of Destruction.

In this game, you can help 100 prisoners to escape by guiding them through the gates. If you give no help, none will get through. No one has ever got all 100 safely through. You guide each prisoner's boat down the river and through the gates, using the "S" key for left and the "D" key for right. Because the current is so swift, there's no way you can stop the boats. You must maneuver them just right to pass through the gates—if possible.

Watch out for the rocks at the edge of the river. They will also make you lose a prisoner if one touches them. You'll also notice that the shoreline changes as the game progresses. This represents the tide exposing and covering some of the rocks at the edge. Good luck.

Game Program Listing

```

5 P=5
10 PRINTCHR$(147);
20 POKE 56,28:POKE52,28:POKE36879,62
30 FORT=0TO511:POKE7168+T,PEEK(32768+T):
   NEXTT:S=7680:H=22:D=38400
40 FORT=1TO2:READA:FORN=0TO7:READB:
   POKE7168+A*8+N,B:NEXTN,T
50 DATA0,255,255,255,255,255,255,255,255
60 DATA27,24,60,36,36,36,36,60,24
70 POKE36869,255
80 DIM A(31)
90 FOR T=1TO22STEP2
100 A(T)=5+INT(3*SIN(2*T*PI/20)+.5)
110 A(T+1)=17-INT(3*SIN(2*T*PI/20)+.5)
120 NEXT
130 FORT=1TO22STEP2
140 GOSUB1000
200 NEXTT
210 REM INITIAL POSITIONS
220 R=H:C=11:W=5
230 POKES+R*H+C,27:POKED+R*H+C,6
240 M=100
250 FORT=1TO22STEP2
253 CN=CN+1
254 A(T)=5+INT(3*SIN((T+CN)/P+2))
255 A(T+1)=17-A(T)

```



```

256 POKES+R*H+C,32:POKED+R*H+C,3
257 A(T+1)=17-A(T)
258 GOSUB1000
260 FORN=A(T)+1TOA(T+1)-1
270 POKES+R*H+C,32:POKED+R*H+C,3
280 B=PEEK(197)
290 IFB=18THEN330
300 IFB=41THEN380
310 GOTO430
320 REM BOAT GOES RIGHT
330 C=C+1:IFC>20THEN380
350 IFPEEK(S+(R-1)*H+C)=0ORPEEK(S+R*H+C-1)
    =0THEN:C=-1:GOTO740
360 GOTO410
370 REM BOAT GOES LEFT
380 C=C-1:IFC<2THEN330
400 IFPEEK(S+(R-1)*H+C)=0ORPEEK(S+R*H+C+1)
    =0THEN:C=C+1:GOTO740
410 REM WALL AND HERO ADVANCE
430 IFPEEK(S+R*H+C)=0ORPEEK(S+(R-1)*H+C)
    =0THENGOTO740
440 R=R-1
450 POKES+R*H+C,27:POKED+R*H+C,6
456 FOR Q=1TO300:NEXT
458 POKE36878,15:POKE36874,135+5*N:FORQ
    =1TO100:NEXTQ:POKE36874,0:POKE 36878,0
460 POKES+(T+9)*11+N,0:POKED+(T+9)*11+N,0
465 IFPEEK(S+R*H+C)=0THEN740

```

```

470 IF R<5THENPOKES+R*H+C,32:POKED
    +R*H+C,3:GOSUB880:GOTO510
480 NEXT:POKE36878,15:POKE36877,185:
    FORQ=185TO132STEP-1:POKE36877,Q:
    NEXT:POKE36877,0
490 GOSUB 880
500 NEXT:GOTO 250
510 POKE36878,15:POKE36876,176:FORG=
    1TO100:NEXT:POKE36876,194:FORG=1TO
    100:NEXT:POKE36876,238
520 FORG=1TO500:NEXT:POKE36876,0
530 PRINTCHR$(19);CHR$(144);"FREEDOM!!!!"
540 R=21:C=11
550 ES=ES+1
560 PRINT"ESCAPED=";ES
570 M=M-1
580 PRINT"# LEFT=";STR$(M);" "
590 IF M=0THEN620
600 IF T>=22THEN250
610 NEXTT:IFM>0THEN250
620 PRINT"ANOTHER GAME?(Y=YES,N=NO)"
710 B=PEEK(197):IFB=64THEN710
720 IFB=11THENPRINTCHR$(147);:GOTO130
730 END
740 POKE36878,15
745 POKE36875,170:FORG=1TO300:NEXT:
    POKE36875,129:FORG=1TO300:NEXT
760 POKE36875,0:POKE36878,0

```



```

770 PRINTCHR$(19);CHR$(144);"ANOTHER
    LOST!!!":FORG=1TO500:NEXT
780 PRINT"ESCAPED=";ES
790 M=M-1
800 PRINT"# LEFT=";STR$(M);" "
810 IF M=0THEN620
820 IF PEEK(D+R*H+C)=0THENR=R-1
825 F=PEEK(S+R*H+C):F1=PEEK(D+R*H+C)
830 POKES+R*H+C,27:FORB=1TO4:POKED+R*H
    +C,2:FORA=1TO100:NEXTA
840 POKED+R*H+C,6
850 FORA=1TO100:NEXTA,B:POKES+R*H+C,F:
    POKED+R*H+C,F1:GOSUB880
860 IFT>=22THEN250
870 R=H:C=11:NEXTT:IFM>0THEN250
875 GOTO 250
880 FORN=A(T)+1TOA(T+1)-1:POKES+(T+9)*11
    +N,32:POKED+(T+9)*11+N,3:NEXT
890 RETURN
1000 FORN=0TOA(T)
1010 POKES+(T+9)/2*H+N,0:POKED+(T+9)
    /2*H+N,5
1020 NEXTN
1030 FORN=A(T+1)TO21
1040 POKES+(T+9)/2*H+N,0:POKED+(T+9)
    /2*H+N,5
1050 NEXT N
1060 RETURN

```

Variables in this Program Include:

P=A constant used to control the shape of the banks. P is proportional to the period of the sine wave that makes up the banks

T=A variable that stores the current row of the gate

S=The first position on the screen

H=The number of characters/positions per row

D=The first position in color memory

A=Variable used for time delay when boat flashes, changing colors after collision. Increasing A will make the boat flash longer

N=Current position of gate going across the river

B=Used to test the keyboard

R=Row of the boat. Initially, R=22 since the boat is on bottom row

C=Number of columns from the left of boat. Initially, C=11

M=Number of prisoners

CN=Variable that controls change in river shape

Q=Variable used for timing delay

G=Used in timing delay for sound

F=Stores code of character at old position of boat

F1=Stores color of character at old position of boat

How the Program Works

- 5 This is a constant that is used to set the curviness of the river boat. P is proportional to the period of the size wave that makes up the river bank. Try different values of P, such $P=1$ and $P=10$ to see different bank shapes. See what is your best score for different P's
- 10 Clears the screen
- 20 Protects the part of the memory that we will use later to create the monsters and the other special graphic characters
- 30 Copies the VIC-20's character memory to a place where we can change it later. The last part of this line sets up some of the variables that will be used later in the program
- 40 Reads the information contained in the data statements that reconfigure the character set
- 50 Contains the information needed to change
- 60 The character set
- 70 Tells the Vic-20 to use our character set
- 80 Sets aside enough memory to accommodate our
A array
- 90 Sets up the way the river will look
- 100 Defines the left-hand side of the bank
- 110 Defines the right-hand side of the bank
- 120 Says to go on to the next line
- 130 Starts the loop that will determine the banks.
This line should not be changed
- 140 Jumps to line 1000 and returns when finished
- 200 Takes the next value or the next row of bank
- 210 The next section is the beginning of the main
loop
- 220 Initializes a few variables

- 230 Puts the hero's ship on the bottom of the screen
- 240 How many ships we will use to try to get through the river
- 250 This is the main loop that controls the row of the black barrier that crosses the river
- 253 Increases a constant that is used to determine where the bank is drawn (change it if you like)
- 254-255 Used to recalculate the position of the bank
- 256 Blanks the hero's ship from the screen
- 257 Does the same as line 255
- 258 Does the same as line 140
- 260 Controls the gate itself instead of its row
- 270 Makes the hero's ship disappear from the screen to get him ready to advance to his next position
- 280 This is the keyboard test. If a key is pressed, then its code appears in memory location 197
- 290 Tests to see if the key pressed was a "D."
- 300 Tests to see if the key pressed was an "S."
- 310 If no key was pressed, then advance both hero and wall
- 320 This section is for going right
- 330 Adds one to "c" and then tests to see if it is too close to twenty. If it is then goes left
- 350 Tests to see if the hero is going to run into something; if he is, then goes to 7000
- 360 Advances both hero and wall
- 370 This section is for going left
- 380 Subtracts one from "c" and then tests to see if it is too close to two; if it is, then goes right
- 400 Tests to see if the hero is going to run
- 410 Advances both the hero and the wall

- 430 Subtracts one from the hero's current row
- 440 Tests the hero's current position to make sure that he hasn't run into anything
- 450 Puts the hero on the screen in his new position
- 456 This is a delay to slow down the game. You can delete this line if you want the game to go much faster. If you think the game is too fast, then increase this value
- 458 This is the sound routine that sounds as the wall advances (should not be changed)
- 460 Puts the new section of wall on the screen
- 470 Tests to see if the hero has reached the top of the screen; if he has, then the wall will disappear and music will play, (GOSUB 880 and GOTO 740)
- 480 Does the same with the next section of wall
- 490 Clears out the old wall
- 500 Does the same with a new wall
- 510 Sound! The first part turns on the volume. The second part plays a tone on the highest generator. The next part is a delay to hear the tone. The tone is now higher and then another delay. The tone is then changed again
- 520 This is another delay, but this one is going to last longer. But, again the tone generator is turned off after the tone
- 530 Prints "Freedom!!!" at the top of the screen, signifying that the hero has escaped
- 540 Resets the row and column where the hero will start again
- 550 Adds one to the number that have escaped
- 560 Prints "Escaped," then the number
- 570 Subtracts one from the number of attempts left
- 580 Prints "# Left," then the number left

- 590 Tests to see if you have no more attempts left. If not, then GOTO the ending message
- 600 If the "t" loop is already finished, then restarts that loop
- 610 Takes the next row (for the wall)
- 620 Asks if you'd like to play again
- 710 This section tests the keyboard again. If no key is pressed, then the program will wait until one is pressed
- 720 If the "Y" key was pressed, then the program will shift the bank by one position. This will also reset the score
- 730 Ends the program
- 740 Turns on the volume to begin another sound routine
- 745 The first part of this line starts the middle tone generator of the VIC to a medium pitch. The second part is a time delay to hear the note. The third part makes the tone lower in pitch. Then there is another time delay
- 760 Turns off the middle tone generator and the volume
- 770 Prints the captured message on the screen, and then waits a few seconds
- 780 Prints "Escaped" and the current score
- 790 Subtracts one from the number of attempts left
- 800 Prints "# Left" and the number of attempts left
- 810 Tests to see if the number of attempts is zero
- 820 Tests to see if the hero was run aground. If he was, then one will be subtracted from his row. If it is, then the ending message is printed with the score
- 825 Remembers what was at the position where

the boat crashed. After the boat has finished flashing, whatever was there will reappear

830 Flashes the boat on the screen. First, it flashes red and waits

840 Changes the hero's color back to blue

850 Waits a few seconds before going back to red. After the flashing, the characters that were remembered are replaced on the screen

860 If the wall has already reached the last row, then restarts the loop

870 Resets the hero's row and column, then goes back and either takes the next 't' or restarts the 't' loop. Resets the 't' loop

880 Sets up a loop to clear away the previous wall (used many times in the program and shouldn't be changed)

890 Returns from wherever the program was before it came

1000 Starts the loop for the banks to be placed on the screen in a different place. First, the left

1010 Places the left side according to the loop above

1020 Takes the next piece

1030 Now the right side

1040 Places the right side on the screen

1050 Takes the next piece

1060 Goes back to where you came from

How You Can Change the Program

1. Allow multiple gates to make it harder for the prisoners to get through.
2. Allow the prisoners boats to stop temporarily. For example, let them stop on islands. Give extra points if the prisoners reach the islands.