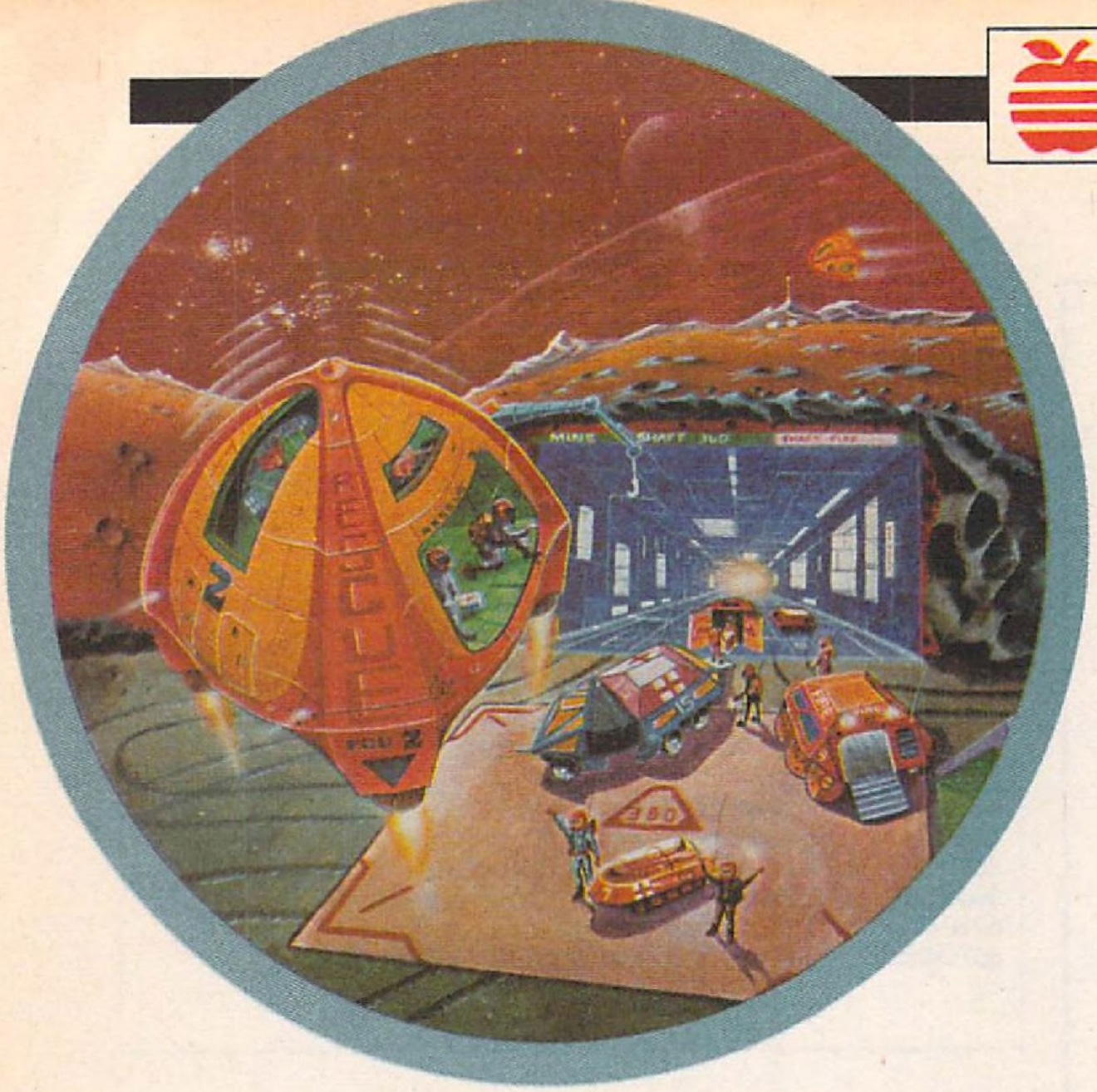


EVACU-POD

by William K. Balthrop
HCM Staff

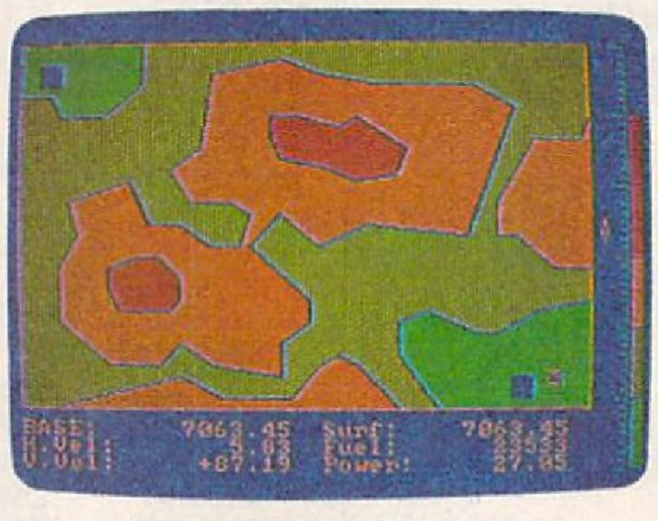
***On another planet,
in another time,
you delicately guide your ship
on a hazardous rescue mission.
Will you make it on time?
Will you make it at all?***

This game is based on an old all-time favorite—*Interplanetary Rescue*—originally published in our forerunner, *99'er Magazine*. Even our old fans of the original will greatly enjoy this much-improved, option-filled version.

The call is out! Send Evacu-Pod!
The time: in the near to distant future.
The place: a neighboring planet.
Mining operations have proven profitable—but risky. Interplanetary construction crews have built full-scale communities on 4 planets, each with its own hospital serviced by crack rescue units. This is where you come in. You've trained back home to fly Evacu-Pod missions, but nothing on the relatively gentle Earth has prepared you for the varying rugged terrain that surrounds each medical center. Evacu-Pod pilots have rated the terrains for their hazards. You may volunteer to fly at any level: beginner, advanced, expert, or professional.

After indicating your skill as a pilot, you can choose the planet to which you would like to be assigned: Earth's moon, Mars, or Venus. A fourth option allows you to designate the size of another planet where you would like to work with gravity ranging from 1 to 9. For comparison purposes, Earth's moon is a 2 and Venus is a 6.

Once you've selected a planet, you will be able to select a time period for your mission. Your choices range from the year 1995 to 2485. As the space program develops over the centuries, more efficient rockets and rocket fuels are invented; thus, the later the year, the easier the assignment will be. This efficiency is incorporated into your rescue vehicle. In 1995, the rockets are fairly inefficient, and so require more fuel to lift the same amount of weight as a rocket built in 2485.



Screen is taken from the IBM version of Evacu-Pod

Winning The Game

To win in *Evacu-Pod*, you must pilot your rescue craft from your home base, over the mountains and valleys of the planet's surface, to the mining camp where you must rescue an injured miner. There you must land at the mining camp, take off again, and safely make it back to your home base. You must do this without crashing, running out of fuel, or running out of time. If you take too long, the injured miner will die while waiting, and you will have failed your mission. The time remaining is not displayed on the screen. This adds a little suspense to the program—you never know when it may be too late . . .

Radar Map

Most of the screen is taken up by the radar map. This is like a topographical map of the local area. Your home base is in the upper-left corner, with the mining camp in the lower-right corner. The radar map is made up of different colors, each color representing a different altitude in 2000-meter increments.

The Radar Altimeter

To the right of the screen is the radar altimeter, which records your altitude in relation to the elevations on the main radar map. The maximum altitude on this altimeter is 10,000 meters. The altimeter serves two purposes: If you are 200 meters or more directly above the planet's surface, the colors of the radar map are displayed vertically, showing their relative altitudes, with a picture of your ship next to the display showing its vertical position. If you are next to one colored portion of the altimeter, you will need to climb before entering the like-colored portion of the radar map. (You will need to be above that color.)

If your ship is within 200 meters of the surface (say, you are at 6150 meters altitude in relation to your base and the terrain you're flying over is at 6000 meters), a different scale will appear on the radar altimeter. The new scale ranges from 0 to 200 meters in relation to the surface directly beneath you. This scale is especially helpful in landing your rescue pod.

Instrumentation

In addition to the radar map and radar altimeter, 6 additional readouts will help you pilot your ship:

BASE — BASE altitude is your altitude in relation to both your home base and the mining camp. Even if you are flying over a 6000-meter mountain top at just 25 meters above the top of the mountain, your BASE altitude would be 6025 meters.

SURF — The SURFace altitude indicates how close you are to the actual surface of the planet. Using our earlier example, if you were flying over a 6000-meter mountain top at a BASE altitude of 6025 meters, your SURFace altitude would be 25 meters. Under this condition, the radar altimeter would be displaying a range of 0 to 200 meters because your surface altitude is below 200 meters.

FUEL — This readout indicates the number of kilograms of fuel that you have remaining. This fuel is added into the weight of your ship. As your fuel is depleted, it will take less power to lift your ship. Once you run out of fuel, your ship will cease to output any thrust and will drop like a rock to the surface.

POWER — The amount of fuel being used by the rockets is displayed here. Three units of power are equal to one kilogram of fuel used every second. The amount of lift generated by the power setting will be determined by the efficiency of your rockets. As mentioned earlier, you can adjust this by selecting a year for the mission. With an efficiency of 1, 1 unit of power would generate 1000 newtons of thrust. The efficiency factor is .66 in 1995 and 2.66 in 2485. This means that only 666 newtons of thrust are generated for a unit of power in 1995, while 2660 newtons of thrust are generated with each unit of power in 2485.

H.VEL — Horizontal VELocity is the speed at which you are traveling across a planet's surface. This speed has no bearing on whether you are going up or down, it's just your speed across the surface. Velocity is measured in meters per second. One meter per second is roughly 2 miles per hour.

V.VEL — Vertical VELocity is the speed at which you are either climbing or descending in altitude. If your vertical velocity is a positive number, you are rising—moving farther *from* the surface. If your vertical velocity is a negative number, you are falling back *toward* the surface. A vertical velocity of 0 would indicate that you are hovering at one altitude. Velocity is measured in meters per second. See H.VEL.

". . . nothing on the relatively gentle Earth has prepared you for such a rugged terrain."

Controls

The Evacu-Pod is like a rocket-powered helicopter—it always flies nose up and has five rocket engines to enable you to maneuver. Four small rockets, one on each side of the pod, give you control over your horizontal direction of travel and velocity. One large rocket beneath the pod supplies all of the lift.

For the keys to use to control your Evacu-Pod, see the Control Capsule.

Landing

Landing the ship is very tricky and difficult. You must touch down at a velocity of −4 meters per second (mps) or less to have a perfect landing. You can land at any location on the map, but if you are not directly over the base and you land at a rate faster than −4 mps, you will crash. The results of landing on a base or mine are a little more bearable:

V. VEL AT TOUCHDOWN	RESULT
0 to −4 mps	Perfect landing.
−4.01 to −10 mps	Landing gear damaged.
−10.01 to −15 mps	Fuel leak—lose half of fuel.
−15.01 to −25 mps	Ship is inoperative. You can't take off again.
−25.01 and up	Crash. You and your crew die in the explosion.

Now get out there, mind your controls, and rescue those miners. Happy Landing!

For your key-in listings, see HCM PROGRAM LISTINGS Contents.

CONTROL CAPSULE
Evacu-Pod

KEY	FUNCTION
U	Increase power light.
I	Increase power medium.
O	Increase power heavy.
J	Decrease power light.
K	Decrease power medium.
L	Decrease power heavy.
E	Thrust ship to the North.
S	Thrust ship to the West.
D	Thrust ship to the East.
X	Thrust ship to the South.
T	Initiates takeoff thrust.
1	Pause game. Continue by pressing any key.

The Apple *II* version of *Evacu-Pod* uses unique character-oriented, machine-language graphics routines, which are located in DATA statements at the end of the program. The two variables, PRNT and REST, contain the addresses used in CALLs. The PRNT routine redirects the output of a PRINT statement from the normal text screen to the high-resolution (hi-res) graphics screen. In addition, the routines change the characters PRINTed into the graphics which appear on the screen in *Evacu-Pod*. The characters are contained in the DATA statements from lines 1540-2370.

After the PRINT to the hi-res screen, the system must have its output routine restored, and this is the purpose of the REST routine.

This program is designed to run under both ProDOS and DOS 3.3. The PRNT routine has to keep the Disk Operating System (DOS) aware of where the screen output is being directed. However, the 2 operating systems have different addresses where this information must be placed, so a special flag is set at zero-page location 6 telling which operating system is in charge.

By PEEKing 2 locations (49505 and 495111) in line 250, the operating system in charge can be discovered. These locations are part of the ProDOS System Global Page. Even in future updates of ProDOS, these locations will contain the same values, so you can feel confident that the program will remain compatible with future versions.

At the same time, we have found that these same locations do *not* contain these values when DOS 3.3 is the DOS, so this program will function properly whether keyed in or converted to either system.

One of the Commodore 64's strengths is its 10-character keyboard buffer. In a game like *Evacu-Pod*, however, the program needs to isolate each keypress, and not keep a queue. Instead of using the GET statement, this program bypasses the buffer and directly PEEKs the keyboard (location 197) for the key presently being pressed.

If no key is pressed, a PEEK of location 197 yields a 64. The following 2-line program allows you to see exactly what values are present when you press a key.

```
10 K=PEEK(197):IF K=64 THEN 10
20 PRINT K:GOTO 10
```

When you press a key while running this program, the value PEEKed from location 197 is printed to the screen. A look at lines 930-1080 will reveal how *Evacu-Pod* uses this same input method.

Quite a bit of mathematical manipulation is involved in *Evacu-Pod*, and the IBM machines make displaying the results to the control panel very easy—the key is the PRINT USING statement. The control panel is displayed in lines 1100-1160. For example, here's line 1100:

```
1100 LOCATE 23,1:PRINT USING "& #####.##";"BASE:
";ABS.ALT;:LOCATE 1,1
```

Here the ABSolute ALTitude from the BASE is being displayed. Because this number depends upon the Evacu-Pod's acceleration, it could have many decimal places. The # symbols in the PRINT USING statement each stand for one digit. They tell the BASIC interpreter to round off the number to the number of digits indicated and display only that number. The & symbol tells the system to display the entire string indicated—in this case, the word BASE.

Evacu-Pod makes excellent use of the TI Extended BASIC DISPLAY AT statement. To depict the surface of the planet, an entire screen of colorful graphics is required. Because 21 rows are needed to display the screen, you might expect to see lines and lines of PRINT statements for each of the 4 possible screens available. Instead, the entire screen is displayed with just one line of code for all 4 screens.

This one line is set up with a series of 4 RESTORE statements in lines 1370-1400. The location RESTOREd to is determined by the skill level chosen by the player. The option is placed on the O(0) array, and that value is used in the ON GOTO statement in line 1360. One of 4 sets of DATA is READ in line 1410 to actually display the screen:

```
1410 CALL CLEAR :: CALL COLOR(9,10,12):: FOR A=1 TO 21 :: READ
A$ :: DISPLAY AT(A,1):A$ :: NEXT A
```

If you look at the DATA in lines 1430-2300, you will see a number of lower-case letters and punctuation marks. These characters were redefined earlier as either solid blocks or spaces, then CALL COLOR was used to set up foreground and background colors. **HCM**

Evacu-Pod (Apple *II* Family) Explanation of the Program.

Line Nos.	
100-200	Program header.
210-410	Initialize program.
420-450	Main menu.
460-590	Main program loop.
600-740	Display subroutines.
750-920	Keyboard input routines.
930-1150	Routines to calculate ship's state.
1160-1210	Update control-panel routine.
1220-1510	Check landing condition.
1520-2370	Terrain display and data.
2380-2470	Screen messages.
2480-2680	Get player options choices.
2690-2830	Message and machine language data.

Evacu-Pod (C-64) Explanation of the Program.

Line Nos.	
100-190	Program header.
200-380	Program initialization.
390-550	Get level input and draw screen.
560-660	Main program loop.
670-790	Display control panel.
800-1370	Display altimeter and update variables.
1380-1800	Landing routines.
1810-2000	Output subroutines.
2010-2920	Radar map print routines.
2930-3250	Sprite DATA.
3260-3420	Sound and message DATA.

Evacu-Pod (IBM PC & IBM PCjr) Explanation of the Program.

Line Nos.	
100-230	Program header.
240-310	Program initialization.
320-540	Get level input and draw screen.
550-740	Main program loop.
750-890	Update variables.
900-1090	Display altimeter.
1100-1190	Display control panel.
1200-1690	Radar map DATA.
1700-1800	Option input routine.
1810-2020	Print messages, end game routines.

Evacu-Pod (TI-99/4A) Explanation of the Program.

Line Nos.	
100-190	Program header.
200-290	Initialize program.
300-370	Main control loop.
380-500	Display routines.
510-670	Get input for ship controls.
680-730	Calculate velocity and altitude.
740-790	Check for a collision.
800-890	Check ship position, update radar.
900-950	Display control-panel values.
960-1010	Landing and crash away from base.
1020-1070	Option to play again. Display score.
1080-1260	Landing and crash at base.
1270-1420	Get options for a new game.
1430-2310	Terrain data and display.
2320	Key input, and pause routine.
2330-2450	Data for option messages.

XB *Evacu-Pod* requires TI Extended BASIC.