DAY 6

Programming the HP-25 — Part II

Where are we?

In preparation for the Day 5 class, you read pp. 73-82, which was an introduction to programming. As your second assignment, you created a program that solves the problem of finding the direction of Mecca. This program used both the $\cos^{-1}$ and $\sin^{-1}$ functions (when reading out loud, you say "arc cos" and "arc sin," and often these are written as arccos and arcsin, or sometimes acos and asin. Inverse functions often do not actually have unique values, so one has to be picked. For $\sin^{-}1$ the calculator will always pick a value between $-90°$ and $90°$. The direction of Mecca (as measured from North) may actually be 180° minus the value the calculator gives.

For our next relatively easy program, let's review the program on pp. 80-82. We have been storing the initial values that we want to feed a program in registers. This program is a nice example of storing the initial value for the program in the X and Y locations in the stack.

Part II of Programming

Our programs have so far had no decision points! They always do the same thing. The exception was Nimb, which you keyed in without really understanding it. It had multiple decision points, including a final decision to go to "I.LOSE" or "BLISS" depending on who had to make the final move.

As we are taking our programming to the next level, we'll start with something easy: the R/S key. Then the PAUSE key. These we will become familiar with together following pp. 83-86 of the manual.

Finally, we get to decision points, which the Hewlett-Packard 25 *Owner's Handbook* calls "Branching."

Unconditional Branching

The types of branching are unconditional branching and conditional branching. Unconditional branching might not seem terribly useful, because it isn't really a decision point. It is at its most powerful when used together with conditional branching, but it is also important on its own.

Conditional Branching

On p. 89, the extremely powerful feature of conditional branching is introduced. Above, it was mentioned that for $\sin^{-}1$ the calculator will always pick a value between $-90°$ and $90°$. The

Preparation for the Next Class

Finish the chapter on Programming (through p. 99 of the *Owner's Handbook*). We skipped a couple of sections (most importantly the section on Statistical), but we will get back to that, and you are now ready to tackle any application in the Applications Programs book.

Also key in the Moon Landing simulator and have it ready to go at the beginning of class. We are going to play it, and we are going to look at how it uses conditional branching. HP's documentation for the Moon Landing Program is on the next three pages.

CHAPTER 3 GAMES

MOON LANDING SIMULATOR

Imagine for a moment the difficulties involved in landing a rocket on the moon with a strictly limited fuel supply. You're coming down tail-first, free-falling toward a hard rock surface. You'll have to ignite your rockets to slow your descent; but if you burn too much too soon, you'll run out of fuel 100 feet up, and then you'll have nothing to look forward to but cold eternal moon dust coming faster every second. The object, clearly, is to space your burns just right so that you will alight on the moon's surface with no downward velocity.

The game starts off with the rocket descending at a velocity of 50 feet/sec from a height of 500 feet. The velocity and height are shown in a combined display as –50.0500, the height appearing to the right of the decimal point and the velocity to the left, with a negative sign on the velocity to indicate downward motion. If a velocity is ever displayed with no fractional part, for example, –15., it means that you have crashed at a speed of 15 feet/sec. In game terms, this means that you have lost; in real-life, it signifies an even less favorable outcome.

You will start the game with 120 units of fuel. You may burn as much or as little of your available fuel as you wish at each step of your descent; burns of zero are quite common. A burn of 5 units will just cancel gravity and hold your speed constant. Any burn over 5 will act to change your speed in an upward direction. You must take care, however, not to burn more fuel than you have; for if you do, no burn at all will take place, and you will free-fall to your doom! The final velocity shown will be your impact velocity (generally rather high). You may display your remaining fuel at any time by recalling R_2.

Equations:

We don't want to get too specific, because that would spoil the fun of the game; but rest assured that the program is solidly based on some old friends from Newtonian physics:

$$x = x_0 + v_0 t + \frac{1}{2} at^2 \qquad v = v_0 + at \qquad v^2 = v_0{}^2 + 2ax$$

where x, v, a, and t are distance, velocity, acceleration, and time.

Notes:

1. If you crash before running out of fuel, the crash velocity shown will be the velocity before the burn, rather than the impact velocity.
2. Use only integer values for burns. Any decimal entry will cause an error in the display for V.X.

Programming Remarks:

An interesting feature of this program is the simultaneous display of both speed and altitude (V.X), as for example, –50.0500. This is accomplished by storing the speed V and the altitude X in their normal form (–50.00, 500.00), then dividing X by 10,000 (10^4) before combining them. An additional subtlety involves the question of the sign of V, and whether ($X/10^4$) is to be added to or subtracted from V. For example, if V = –50 and X = 500, we should subtract: V – ($X/10^4$), in order to generate a display of –50.0500. But if V = 10 and X = 50, we should add: V = V + ($X/10^4$) in order to display 10.0050. Inspection of the program listing, lines 2 through 12, will reveal how a conditional branch was used to resolve the dilemma.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS
00							
01	14 11 04	f FIX 4					Four-place display
02	24 00	RCL 0	X				Form display V.X
03	33	EEX	1. 00	X			
04	04	4	1. 04	X			
05	71	÷	$X/10^4$				Divide X by 10,000
06	24 01	RCL 1	V	$X/10^4$			
07	15 41	g x<0	V	$X/10^4$			Is V negative?
08	13 11	GTO 11	V	$X/10^4$			Yes, branch
09	51	+	$V + X/10^4$				No, add V and X
10	13 13	GTO 13	$V + X/10^4$				
11	21	$x \rightleftarrows y$	$X/10^4$	V			V<0, add V and –X
12	41	–	$V - X/10^4$				
13	74	R/S	V.X				V.X is V ± ($X/10^4$)
14	24 02	RCL 2	F	B			Burn B has been input
15	14 41	f x<y	F	B			Burn > Fuel?
16	13 34	GTO 34	F	B			Yes, prepare to crash
17	22	R↓	B			F	No, update A, X, V
18	23 41 02	STO – 2	B			F	Subtract burn from fuel
19	05	5	5	B			5 units cancels gravity
20	41	–	B – 5				Acceleration = B – 5
21	23 03	STO 3	A				
22	02	2	2	A			
23	71	÷	A/2				
24	24 00	RCL 0	X	A/2			
25	51	+	X + A/2				
26	24 01	RCL 1	V	X + A/2			
27	51	+	X + V + A/2				New altitude: X←X+V+A/2
28	23 00	STO 0	X				
29	15 41	g x<0	X				Is X below ground?
30	13 44	GTO 44	X				Yes, you've crashed
31	24 03	RCL 3	A	X			No, update V
32	23 51 01	STO + 1	A	X			New velocity: V ← V + A
33	13 02	GTO 02	A	X			Display V.X
34	24 01	RCL 1	V				All fuel gone, show
35	15 02	g x^2	V^2				crash velocity as
36	24 00	RCL 0	X	V^2			$V = (V^2 + 2gX)^{1/2}$
37	01	1	1	X	V^2		where g = gravity = 5
38	00	0	10	X	V^2		
39	61	x	10 X	V^2			
40	51	+	V^2 + 10 X				
41	14 02	f $\sqrt{x}$	V				
42	32	CHS	V				Show crash V down
43	23 01	STO 1	V				
44	24 01	RCL 1	V				Come here from line 30
45	14 11 00	f FIX 0	V				Display integer V to
46	13 00	GTO 00	V				show crash
47							
48							
49							

REGISTERS	
R_0	X
R_1	V
R_2	Fuel
R_3	Acceleration
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize	X	500	STO	0		500.00
		V	50	CHS	STO	1	-50.00
		Fuel	120	STO	2		120.00
3	Display initial V.X		f	PRGM	R/S		-50.0500
4	Key in burn, compute new speed						
	and distance	Burn	R/S				V.X
5	Perform step 4 till you land or						
	crash						
6	To see remaining fuel at any						
	time		RCL	2			Fuel
7	To display speed and distance						
	at any time		f	PRGM	R/S		V.X
8	To start a new game, go to step 2.						

Example:

500 STO 0 50 CHS STO 1 120 STO 2

f PRGM R/S → -50.0500

0 R/S → -55.0448

5 R/S → -55.0393

(note constant V when burn = 5)

30 R/S → -30.0350

0 R/S → -35.0318

0 R/S → -40.0280

0 R/S → -45.0238

0 R/S → -50.0190

RCL 2 → 85.0000

(remaining fuel)

f PRGM R/S → -50.0190

(display V.X again)

10 R/S → -45.0143

0 R/S → -50.0095

RCL 2 → 75.0000

10 R/S → -45.0048

25 R/S → -25.0013

20 R/S → -25.