

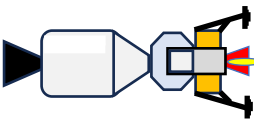


Introduction

The Sinclair QL was launched in 1984 and to celebrate 40 years of continued interest in this home computer and evolved software, hardware spin-offs I thought I'd turn the screws of nostalgia with another icon of computing history. NASA's landing a man on the Moon prompted a high-school student to create the first known Simulation Game of that event and triggered a Lunar Lander Genre.

A QBITS Lunar Lander version of the Text based Game is Part One with Part Two a Lunar Excursion Module (LEM) Simulation in 2D Graphics both coded in QL SuperBASIC.

QBITS Lander -LEM Concepts



The LEM Training Simulation starts by engaging the Auto-Pilot, Release of Docking Clamps and Disengagement from the Orbiter. A long blast of the main Jet follows as the LEM is sent on a Decent path to the Lunar Surface. At some point the Onboard Computer Malfunctions and Manual Override is activated.

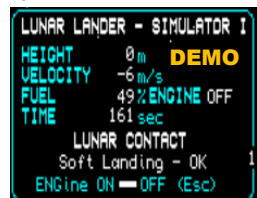
Taking control, use bursts of the Main Jet in conjunction with directional Thrusters to execute a soft landing on the Moon surface with the Horizontal and Vertical speeds of less than 5m/sec, if NOT, the LEM is crippled or destroyed.

QBITS Lander – LEM Controls

Up & Down Cursors Set the amount of Thrust. Left and Right Cursors control the direction of Jet Engine. Toggle Jet On/Off with Spacebar to Decay Orbit and slow Decent. The Monitor at top of screen displays **ALTITUDE**, **FUEL** remaining, **H-VEL** Horizontal velocity and **V-VEL** Vertical velocity, **ENGINE** On/Off Bar shows Red/Green, adjustable **THRUST** setting and a Digital clock **00:00**. Press 'Esc' to abort the Simulation and 'Q' to Quit the program.

QBITS Lander - LEM MoonScape

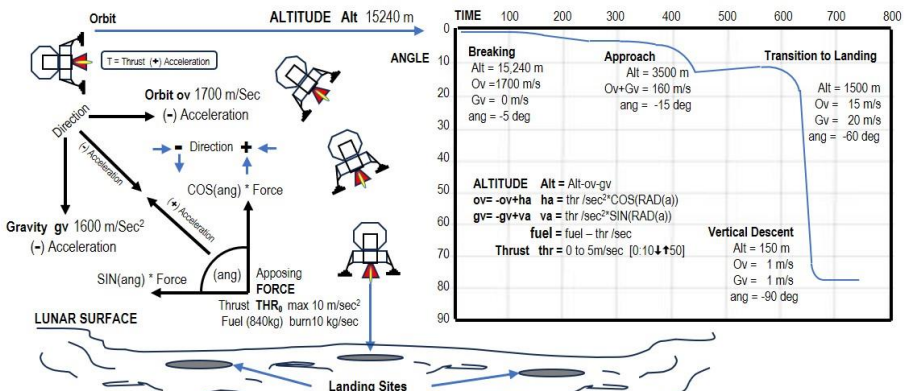
As the LEM's Altitude and Forward Velocity reduces, movement across the Moonscape slows and enlarges to display the surface in more detail. Even a relatively flat surface can still be strewn with boulders and small craters making any landing area potentially more hazardous. A prolonged landing and there is the imminent possibility of running out of Fuel.



QBITS Lander - LEM Simulation

As the disengaged CSM fades away (reduces in size) Auto-Pilot action starts to slow the LEM Orbital speed from 1700m/sec and begins the decent from 16,500m to the Moon surface. The Jet angle is set between 5° and 15° degrees to the horizontal and the LEM moves forward (left to right across the screen). As Orbital (Horizontal velocity) slows Gravity (Vertical velocity) rapidly increases. Change of Jet angle brings the Jet Force to reduce the speed of decent.

As the LEM approaches the moon surface the jet angle can be changed to manoeuvre forward or backward so as to target the landing zone. The aim is for Horizontal and Vertical velocities to be less than 5m/Sec with a final Jet burst just prior to touch down leaving a drop of a less than 5m for a soft touchdown on to the landing site. The execution of the Flight Path taking approx. 13 plus minutes in all.



Note: LEM must keep within Sides & Upper screen boundaries or the Simulation is ABORTED.

QBITS lander - LEM Flight Calculations

The optimum Flight Path is to calculate the minimum time and fuel used to soft land the LEM on the Lunar surface. Orbital Speed and Gravity are viewed as Negative (-) velocities acting on the LEM's mass. The opposing Positive (+) velocity is the Force of Thrust from the Jet Engine. To overcome (-) velocities of Orbit ' ov ' and Gravity ' gv ' the opposing horizontal acceleration ' ha ' and vertical acceleration ' va ' are dependent on Jet Angle ' ang ' and the Rate of Thrust ' thr ' and time ' jt ' the Jet ENGINE is switched ON. As Orbital velocity ' ov ' decreases the action of Gravitational pull ' gv ' accelerates with descent toward the moon's surface.

$$ov = 1700 - ha \quad gv = 1.6 - 1.6 * 1700 / ov$$

$$ha = COS(RAD(ang)) * (thr / 20) * jt : ang [.999 < .001] : thr = 05 < 2.5 \text{ in } 0.5 \text{ steps}$$

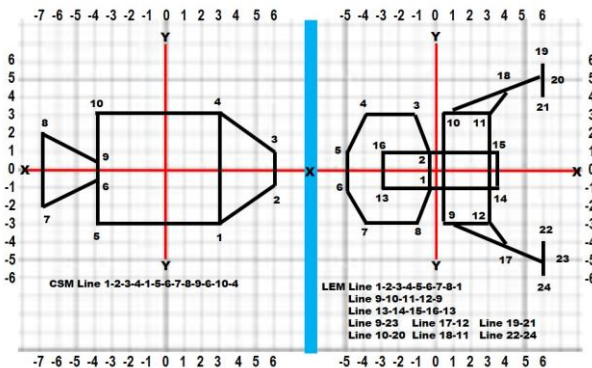
$$va = SIN(RAD(ang)) * (thr / 20) * jt : ang [.001 < .999] : jt = \text{sec/sec count while ENG ON}$$

Gravitational pull on the LEM's mass is reduced and directly related to the LM's Orbital velocity. For simplicity acting Gravity $gv = 1.6 \text{ m/sec}^2 * HVvel_0 / HVvel - ha$ which will be very small to start but increase as Orbital velocity is further reduced by the LEM's Jet. The decent in will increase exponentially as the orbital velocity decreases but controlled by the effects of any vertical acceleration va .

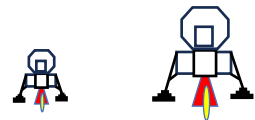
$$HVvel = HVvel - ha :: VVel = -VVel - gv + va : Alt = Alt - VVel$$

QBITS Lander -LEM Graphics

The Command Service Module CSM and Lunar Excursion Module LEM are drawn in vector graphics using SuperBASIC LINE command. As such each x,y point or node can be rotated around a central axis. For the LEM's Flight Path this means turning through 0 to 90 degrees from the Horizontal Orbit into the Vertical Descent for Landing. Then by adding a vector multiplier to the code, the size can also be increased or decreased.



The outline of each design is first mapped on to graph paper. Adding **XX** and **YY** lines with plus and minus values to the grid, and the Object x,y points can then be referenced for use with LINE.



Note: LEM graphics control variables: **ang** angle, **col** colour, **lx ly** screen coordinates, **vz** vector size.

DIMensioned arrays **CM(13,2)** and **LM(40,2)** store the Node x,y coordinates. **PROC**edures **LEM_Ang(ang)** calculates new vector positions with changes in angle. **CSM(col,cx,cy)** and **LEM(col,lx,ly)** draw the Orbiter and Lander, **JET(col1,col2,lx,ly)** draws the Jet exhaust flame, col, col1,col2 are drawn in White Red Yellow or black (background) to clear screen image.

Notes:

The LEM's presented screen image is determined by variables ang, lx, ly, lz and any actions derived from the status of ht, ov, hv, gv, vv, eng, thr, jt

QBITS LEM - MoonScape WIP

Plans for the Moonscape surface at its minimum is seen as just a rough arc across the screen scattered with some craters with the landing area ringed in red. Future development will I hope be able to redrawn an enlarged surface and more recognisable Moonscape as if looking into a large crater with rocky sides and minor craters scattered about. Possibly expand the number of landing areas each with a central and outer area. Landing on the outer area will cause minor damage. Outside of a landing area and the LEM is destroyed.

QBITS LANDER Code

1000 REMark **QBITS_Lander_bas** [QBITS Lander 2024 - QPC2]

1002 dev\$='dos1_':MODE 4:gx=40:gy=40 :REMark Basic settings

1004 **WHEN ERROr** :CONTINUE:**END WHEN**

1006 REMark **Import QBITSConfig Settings**

1007 OPEN _IN#9,dev\$&'QBITSConfig':INPUT#9,gx\gy\dn\$:CLOSE#9

1010 REMark **Arrays**

1011 DIM CM(13,2),LM(40,2),MS(40,2),vx(40),vy(40),x%(8),y%(8)

1012 tm=0

1016 Init_win:Init_LEM:sim=1:Intro_SIM

1018 **DEFine PROCEDURE Init_win**

1019 OPEN#6,scr_ :WINDOW#6,486,98,gx+12,gy+98:CLS#6:SCALE #6,100,0,0

1020 OPEN#5,scr_ :WINDOW#5,508,20,2+gx,220+gy:SCALE#5,20,0,0

1021 OPEN#4,scr_ :WINDOW#4,500,60,6+gx,2+gy

1022 OPEN#3,scr_ :WINDOW#3,500,36,6+gx,2+gy:SCALE#3,20,0,0

1023 WINDOW#2,512,224,gx,gy :BORDER#2,1,3:PAPER#2,0:CLS#2

1024 WINDOW#1,508,212,2+gx,gy+36 :INK 7:SCALE 200,0,0

1025 WINDOW#0,512,32,gx,224+gy :CLS#0:BORDER#0,1,3:PAPER#0,0:CLS#0

1026 OVER#3,1:CSIZE#3,2,1:CLS :FOR a=1 TO 96:POINT#2,RND(5 TO 160),RND(5 TO 85)

1027 INK#3,2:FOR i=0 TO 1:CORSOR#3,2+i,4:PRINT#3,'QBITS LANDER'

1028 INK#3,6:FOR i=0 TO 1:CORSOR#3,4+i,5:PRINT#3,'QBITS LANDER'

1029 OVER#3,0:CORSOR#3,0,0:CSIZE#3,0,1

1030 REMark ***** Display Panel *****

1031 INK#3,4:LINE#3,0,5 TO 60,5 TO 68,2:LINE#3,143,2 TO 151,5 TO 212,5

1032 **RESTORE 1034**:sx=67:LINE#3,sx+4,1

1033 FOR i=1 TO 4:**READ x1,y1,x2,y2**:LINE#3 TO sx+x1,y1:ARC#3 TO sx+x2,y2,PI/2

1034 DATA 72,1,75,4,75,16,72,19,4,19,1,16,1,4,4,1

1035 **RESTORE 1036**:FOR a=1 TO 8:**READ bi,bx,by,str\$**:**QBold 4,bi,1,bx,by,str\$**

1036 DATA 4,168,3,'ALTITUDE',4,168,12,'H-VEL',4,168,22,'V-VEL',4,270,3,'FUEL'

1037 DATA 4,270,13,'ENG',4,270,23,'THR',7,234,12,'← →',7,234,22,'↑ ↓'

1038 BLOCK#4,15,7,242,15,7:BLOCK#4,13,5,243,16,0:BLOCK#4,11,3,244,17,4

1039 INK#3,7:CORSOR#3,300,13:PRINT#3,'00:00':tm=0

1040 **END DEFine**

1042 **DEFine PROCEDURE Init_LEM**

1043 **RESTORE 1044**:FOR a=1 TO 39:**READ LM(a,1),LM(a,2)**

1044 DATA -5,-1,-5,1,-1,3,-4,3,-5,1,-5,-1,-4,-3,-1,-3,-5,-1

1045 DATA -5,-3,5,3,3,3,3,3,-3,5,-3 :REMark Lander Module

1046 DATA -3,-1,3,8,-1,3,8,1,-3,1,-3,-1 :REMark Hatch

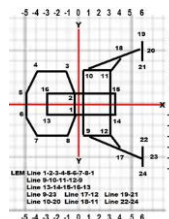
1047 DATA 1,3,6, 5,4,4,3,3,6,4,6,6 :REMark Right Leg/Foot

1048 DATA 1,-3,6,-5,4,-4,3,-3, 6,-4,6,-6 :REMark Left Leg/Foot

1049 DATA 5,5,9,1,5,9,-1,5,5,-5 :REMark Red Jet Flame

1050 DATA 6,0,8,-5,11,0,8,-5,6,0 :REMark White Jet Flame

1051 **END DEFine**



```

1053 DEFINE PROCEDURE QBold(ch,ci,cb,cx,cy,ctr$)
1054 INK#ch,ci:FOR q=0 TO 1:OVER#ch,cb:CURLSOR#ch,cx,cy+q:PRINT#ch,ctr$:OVER#ch,0
1055 END DEFINE

```

```

1057 DEFINE PROCEDURE CSM(col,cx,cy)
1058 RESTORE 1060:INK col:LINE cx+4*cz,cy-3*cz
1059 FOR z=1 TO 12:READ cx1,cy1:LINE TO cx+cx1*cz,cy+cy1*cz
1060 DATA 7,-1,7,1,4,3,4,-3,-4,-3,-4,-5,-7,-2,-7,2,-4,-5,-4,-2,-4,3,4,3
1061 END DEFINE

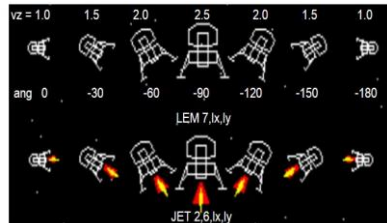
```



```

1063 DEFINE PROCEDURE LEM_Ang(r)
1064 FOR a=1 TO 39
1065 vx(a)=LM(a,1)*COS(RAD(r))-LM(a,2)*SIN(RAD(r))
1066 vy(a)=LM(a,1)*SIN(RAD(r))+LM(a,2)*COS(RAD(r))
1067 LM(a,1)=vx(a):LM(a,2)=vy(a)
1068 END FOR a
1069 END DEFINE

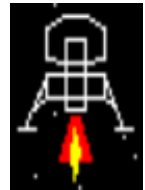
```



```

1071 DEFINE PROCEDURE LEM(col,ix,ly)
1072 INK col:LINE ix+LM(1,1)*lz,ly+LM(1,2)*lz
1073 FOR z=2 TO 9:LINE TO ix+LM(z,1)*lz,ly+LM(z,2)*lz
1074 LINE ix+LM(10,1)*lz,ly+LM(10,2)*lz
1075 FOR z=11 TO 14:LINE TO ix+LM(z,1)*vz,ly+LM(z,2)*lz
1076 LINE ix+LM(15,1)*lz,ly+LM(15,2)*lz
1077 FOR z=16 TO 19:LINE TO ix+LM(z,1)*lz,ly+LM(z,2)*lz
1078 FOR z=20 TO 30 STEP 2
1079 LINE ix+LM(z,1)*vz,ly+LM(z,2)*lz TO ix+LM(z+1,1)*vz,ly+LM(z+1,2)*lz
1080 END FOR
1081 END DEFINE

```



```

1083 DEFINE PROCEDURE JET(col1,col2,ix,ly)
1084 FOR z=1 TO 8:x%(z)=ix+LM(31+z,1)*lz:y%(z)=ly+LM(31+z,2)*lz
1085 INK col1:FILL 1:LINE x%(1),y%(1) TO x%(2),y%(2) TO x%(3),y%(3)
1086 LINE TO x%(4),y%(4) TO x%(1),y%(1):FILL 0
1087 INK col2:FILL 1:LINE x%(5),y%(5) TO x%(6),y%(6) TO x%(7),y%(7)
1088 LINE TO x%(8),y%(8) TO x%(5),y%(5):FILL 0
1089 END DEFINE

```

```

1091 DEFINE PROCEDURE JBEEP
1092 BEEP 20000,40,100,50,5,0,0,0 :REMark Jet Engine Sound
1093 END DEFINE

```

1100 REMark INTRO SIMULATOR I-II

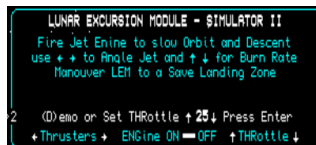
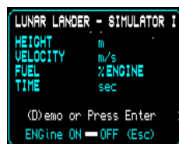
1102 DEFine PROCEDURE Intro_SIM

```

1103 CLS:INK 7:FOR i=1 TO 50:POINT RND(10 TO 340),RND(120 TO 190)
1104 FILL 1:LINE 0,0 TO 0,10:ARC TO 420,10,-PI/12:LINE TO 420,0 TO 0,0:FILL 0
1105 CSIZE 0,0:QBold 1,7,1,140,28,'ASTRONAUT TRAINING':cz=4:vz=4:ang=0
1106 Init_LEM:CSM 7,220,170:LEM 7,270,170:vz=2.5:LEM_Ang -90:LEM 7,66,172
1107 ch=6:QBold ch,7,1,10,6,'LUNAR LANDER - SIMULATOR I':lx=66:ly=172
1108 RESTORE 1110:INK#ch,5:LINE#ch,8,1
1109 FOR i=1 TO 4:READ x1,y1,x2,y2:LINE#ch TO x1,y1:ARC#ch TO x2,y2,PI/2
1110 DATA 124,1,129,6, 129,92,124,97, 8,97,3,92, 3,6,8,1
1111 QBold ch,7,1,220,6,'LUNAR EXCURSION MODULE - SIMULATOR II'
1112 RESTORE 1115:INK#ch,5:LINE#ch,146,1
1113 QBold 6,7,0,204,83,↑ →':QBold 6,7,0,392,83,↑ ↓'
1114 FOR i=1 TO 4:READ x1,y1,x2,y2:LINE#ch TO x1,y1:ARC#ch TO x2,y2,PI/2
1115 DATA 360,1,365,6, 365,92,360,97, 146,97,141,92, 141,6,146,1
1116 FOR i=1 TO 5:READ x,y,str$:QBold ch,5,1,x,y,str$
1117 DATA 10,20,'HEIGHT',10,30,'VELOCITY',10,40,'FUEL',98,40,'ENGINE',10,50,'TIME'
1118 FOR i=1 TO 11:READ x,y,str$:QBold ch,5,0,x,y,str$
1119 DATA 90,20,'m',90,30,'m/s',90,40,'% ',90,50,'sec',20,83,'ENGINE ON OFF (Esc)'
1120 DATA 210,20,'Fire Jet Enine to slow Orbit and Descent'
1121 DATA 204,30,'use ← → to Angle Jet and ↑ ↓ for Burn Rate'
1122 DATA 226,40,'Manouver LEM to a Save Landing Zone'
1123 DATA 212,83,'Thrusters',290,83,'ENGINE ON OFF',400,83,'THRottle'
1124 QBold 3,7,0,360,4,'Select 1<>2 (Q)uit ':QBold 6,7,0,166,70,'1<>2'
1125 BLOCK#ch,12,3,78,87,7:BLOCK#ch,12,3,348,87,7:k=0:Jb=25
1126 REPEAT Intro_lp
1127 IF sim=1
1128 IF k=10:Dem=10:Run_SIM1
1129 IF k=68 OR k=100:Dem=5:Run_SIM1
1130 BLOCK#ch,250,10,210,70,0
1131 QBold ch,7,0,20,70,'(D)emo or Press Enter'
1132 END IF
1133 IF sim=2
1134 IF k= 10: AutoPilot_SIM2:Dem=10:Run_SIM2
1135 IF k=68 OR k=100:AutoPilot_SIM2:Dem=5:Demo_SIM2
1136 IF k=208 AND thr<50:thr=thr+5
1137 IF k=216 AND thr>10:thr=thr-5
1138 QBold ch,7,0,212,70,'(D)emo or Set THRottle ↑ ↓ Press Enter'
1139 BLOCK#ch,150,24,6,60,0:QBold ch,7,1,360,70,Jb
1140 END IF
1141 k=CODE(INKEY$(1)) :IF k=81 OR k=113:LRUN dn$:STOP
1142 IF k=49:sim=1:END IF :IF k=50:sim=2:END IF
1143 END REPEAT Intro_lp
1144 END DEFine

```

Note: DEMO



1150 REMark **SIM1 - LUNAR LANDER**

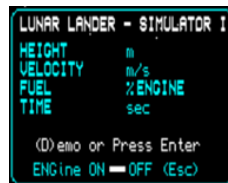
Note: **TIMER** : t0 time zero: lasttime :nowtime ; deltat time in sec : elt elapsed time
Alt Altitude : gv gravitational velocity : eng Engine On/Off :Fuel ; accel acceleration

1152 DEFINE PROCEDURE Run_SIM1

```
1153 BLOCK#ch,148,10,12,70,0 :TIMER=0:t0=Timer : lastime=t0:chk=0
1154 eng=0:gm=-1.62:Alt=10000:Fuel=100:gv=0
1155 REPEAT Sim_ip
1156 CURSOR#ch,58,20:PRINT#ch,FILL$(' ',5-LEN(INT(ht)))&INT(ht)
1157 CURSOR#ch,58,30:PRINT#ch,FILL$(' ',5-LEN(INT(vl)))&INT(vl)
1158 IF eng= 5:QBold ch,7,0,138,40,'ON' :ELSE QBold ch,7,0,138,40,'OFF':BEEP
1159 IF Fuel> 0:QBold ch,7,0,64,40,FILL$(' ',4-LEN(Fuel))&Fuel
1160 IF Fuel<=0:QBold ch,7,0,64,40,' OUT'
1161 CURSOR#ch,64,50:PRINT#ch,FILL$(' ',4-LEN(INT(elt)))&INT(elt)
1162 PAUSE Dem :TIMER=TIMER+1
1163 IF KEYROW(1)=8 OR ht>10000:chk=1:End_SIM1:EXIT Sim_ip
1164 IF Dem=10 AND KEYROW(1)=64:IF eng=0:eng=5:ELSE eng=0:JET 0,0,lx,ly
1165 IF Dem=5 AND Alt<3500 AND Alt>1500:eng=5
1166 IF Dem=5 AND gv>-5:eng=0:BEEP:JET 0,0,lx,ly
1167 IF Dem=5 AND Alt<150 AND gv<-6:eng=5
1168 IF Fuel<=0:eng=0:BEEP:JET 0,0,lx,ly
1169 nowtime=TIMER:elt=nowtime-t0:deltat=nowtime-lastime:lastime=nowtime
1170 accel=gm+eng :REMark Acceleration Gravity -1.6 + Thrust 10
1171 gv=gv+accel*deltat :REMark Velocity (-) or (+)
1172 Alt=alt+gv*deltat+.5*accel*deltat*deltat :REMark Decent New Altitude
1173 Fuel=Fuel-eng*deltat/5 :REMark Fuel used
1174 IF Fuel>0 AND eng=5:JET 2,6,lx,ly:JBEEP :REMark Jet Flame & Sound
1175 IF Alt<0:End_SIM1:EXIT Sim_ip
1176 END REPEAT Sim_ip
1177 REPEAT Ans_ip
1178 IF KEYROW(4)=8:sim=1:EXIT Ans_ip
1179 IF KEYROW(6)=2:sim=2:EXIT Ans_ip
1180 END REPEAT Ans_ip
1181 BLOCK#ch,160,24,6,60,0:k=0
1182 END DEFINE
```

1184 DEFINE PROCEDURE End_SIM1

```
1185 QBold ch,7,0,138,40,'OFF':JET 0,0,lx,ly
1186 IF chk=1: QBold ch,7,1,46,72,'SIM - ABORTED' :RETURN
1187 QBold ch,7,1,46,62,'LUNAR CONTACT'
1188 IF gv>=-5 AND Alt<10:QBold ch,7,0,36,72,'Soft Landing - OK' :RETURN
1189 QBold ch,7,0,10,72,'* Crash Landing - FATAL *'
1190 END DEFINE
```



1200 REMark **SIM2 - LUNAR EXCURSION MODULE**

1202 **DEFine PROCedure Auto_SIM2**

```

1203 BLOCK#1,490,160,10,6,0:FOR s=1 TO 50:POINT#1,RND(10 TO 340),RND(40 TO 180)
1204 Alt=16500:HVel=1700:VVel=0:Fuel=18000:tm=0:eng=4:Init_LEM:LEM_Ang 0
1205 cz=1.2:cx=34:cy=190:CSM 7,cx,cy:lz=1.2:lx=50:ly=190:LEM 7,lx,ly:RESTORE 1207
1206 FOR i=1 TO 3:READ Mes$:CURSOR#3,360,4:PRINT#3,Mes$:CLS#3,4:PAUSE 30
1207 DATA 'Release Docking Clamps' , 'Disengage LEM', 'Engage LEM Auto-Pilot'
1208 FOR i=1 TO 4
1209   CSM 0,cx,cy:LEM 0,lx,ly:cz=cz-.1:lz=lz+.1:cx=cx:cy=cy+2:lx=lx+1:ly=ly-1
1210   CSM 7,cx,cy:LEM 7,lx,ly:PAUSE 5:tm=tm+1:Prt_SIM2
1211 END FOR i
1212 CSM 0,cx,cy:LEM 0,lx,ly:LEM_Ang -5:LEM 7,lx,ly:thr=35
1213 CURSOR#3,360,4:PRINT#3,'Reduce Orbital Speed ' :BLOCK#3,11,3,244,17,2
1214 FOR a=1 TO 25
1215   LEM 0,lx,ly:JET 0,0,lx,ly:Fuel=Fuel-thr*5
1216   IF a>20:ang=ang-5:Init_LEM:LEM_Ang ang
1217   lx=lx+3-d*.1:ly=ly-.25:LEM 7,lx,ly:JET 2,6,lx,ly:JBeep
1218   FOR b=1 TO 5
1219     tm=tm+5:Alt=Alt-2.5*a*b:HVel=HVel-thr/5:VVel=-tm*1.5:Prt_SIM2:PAUSE 1
1220   END FOR b
1221 END FOR a
1222 END Define

```

1224 **DEFine PROCedure Run_SIM2**

```

1225 CURSOR#3,360,5:PRINT#3,'Computer Malfunction':CLS#3,4:yy=5E-2:xx=-.5
1226 REPEAT M_lp
1227   PAUSE Dem:tm=tm+1:k=KEYROW(1):get_keys
1228   IF eng=2 AND Fuel>=thr
1229     Fuel=Fuel-thr
1230     xx=xx+INT(COS(RAD(ang))*thr)/500:yy=yy-INT(SIN(RAD(ang))*thr)/500
1231   ELSE
1232     yy=yy-5E-2
1233   END IF
1234   IF lx+xx<-.20 OR lx+xx>380 OR ly+yy>230 OR KEYROW(1)=8:td=3:EXIT M_lp
1235   IF ly<38
1236     IF ABS(HVel)<20 AND ABS(VVel)<10 AND ang<-.85 AND ang>-.95
1237       td=1:EXIT M_lp : ELSE td=2:EXIT M_lp:END IF
1238   END IF
1239   LEM 0,lx,ly:JET 0,0,lx,ly:lx=lx-xx:ly=ly+yy
1240   VVel=yy*lym:HVel=xx*lxm:Prt_SIM2
1241   Init_LEM:LEM_Ang ang:LEM 7,lx,ly:IF eng=2 AND Fuel>=thr:JET 2,6,lx,ly
1242 END REPEAT M_lp
1243 JET 0,0,lx,ly:eng=4:BLOCK#4,11,3,244,17,eng:Alt=0:Prt_SIM2
1244 IF td=1:CURSOR#3,360,5:PRINT#3,'Soft Landing (Esc)':CLS#3,4
1245 IF td=2:CURSOR#3,360,5:PRINT#3,'Crash Landing (Esc)':CLS#3,4:LEM_Crash
1246 IF td=3:CURSOR#3,360,4:PRINT#3,'SIM ABORTED (Esc)':CLS#3,4
1247 REPEAT ans:IF KEYROW(1)=8:sim=2:Intro_SIM
1248 END Define

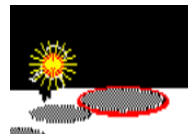
```

1250 **DEFine PROCedure LEM_Crash**

```

1251 INK 2:FILL 1:CIRCLE lx,ly,5:FILL 0:INK 6
1252 FOR i=1 TO 18:a=i*20:LINE lx,ly TO lx+COS(RAD(a))*12,ly+SIN(RAD(a))*12
1253 :
1254 END Define

```



1256 DEFine PROCEDURE MoonScape

1257 INK 7:FILL 1:LINE 0,0 TO 0,15:ARC TO 420,10,-PI/12:LINE TO 420,0 TO 0,0:FILL 0

1258 **RESTORE 1260**:INK 2:FILL 1:CIRCLE 180,21,22,.3,PI/2:FILL 0:INK 248

1259 FOR i=1 TO 12:**READ mx,my,mr**:FILL 1:CIRCLE mx,my,mr,.25,PI/2:FILL 0

1260 DATA 20,8,8, 35,12,5, 40,5,8, 90,12,26, 130,6,12, 150,15,15 ,180,22,20

1261 DATA 220,15,8, 240,10,9, 250,16,6, 290,8,24, 320,15,14

1262 **END DEFine**



Note: This is the simple version

1264 DEFine PROCEDURE get_keys

1265 k=KEYROW(1):tx=xx:ty=yy

1266 **SElect ON k**

1267 = 2:IF ang>-180:ang=ang-5

1268 = 16:IF ang< 0:ang=ang+5

1269 = 4:IF thr< 50:thr=thr+5

1270 =128:IF thr> 10:thr=thr-5

1271 = 64:IF eng=4 AND Fuel>0:st=tm:eng=2:ELSE eng=4

1272 **END SElect**

1273 IF xx>11 OR xx<-11 :xx=tx

1274 IF yy>8 OR yy<-8 :yy=ty

1275 **END DEFine**

:

1277 DEFine PROCEDURE Prt_SIM2

1278 IF ly<180:Alt=.25*ly*ly/2 :lym=50:lxm=500 :REMark 4500 - 1400

1279 IF ly<120:Alt=.2*ly*ly/2 :REMark 1400 - 300

1280 IF ly< 80:Alt=1E-2*ly*ly/2 :lym= 5:lxm= 50

1281 IF ly<=38:Alt=0 :REMark 0

1282 PrtNum 4,7,6,228,4,Alt :REMark Altitude

1283 PrtNum 4,7,5,198,13,HVel :REMark Horizontal velocity

1284 PrtNum 4,7,5,198,23,VVel :REMark vertical velocity

1285 QBold 4,7,0,243,22,thr :REMark 0 - 50 Throttle

1286 IF Fuel<=0:QBold 4,7,0,300,5,' OUT ':ELSE PrtNum 4,7,5,300, 4,Fuel

1287 TMin=tm DIV 60:CUSOR#3,300,13:PRINT#3,FILL\$('0',2-LEN(TMin))&TMin

1288 TSec=tm MOD 60:CUSOR#3,318,13:PRINT#3,FILL\$('0',2-LEN(TSec))&TSec

1289 **END DEFine**



1291 DEFine PROCEDURE PrtNum(ch,ni,ns,nx,ny,num)

1292 num\$=INT(num):INK#ch,ni:CUSOR#ch,nx,ny:PRINT#ch,FILL\$(' ',ns-LEN(num\$))&num\$

1293 **END DEFine**



```

1295 DEFine PROCEDURE Demo_SIM2
1296 REPEAT Dem_lp
1297 IF Dem=5
1298   PAUSE Dem:i=i+1:Fuel=Fuel-thr*5
1299   LEM 0,ix,ly:JET 0,0,ix,ly:BEEP:IF i<50:thr=20:
1300   ix=ix+1:ly=ly-.5:tm=tm+1:HVel=(ix-180):VVel=-ly+38::Prt_SIM2
1301   IF i=30:CURSOR#3,360,4:PRINT#3,'LEM Descent'
1302   IF i>30 AND i<48:ang=ang-5:ly=ly-1.5
1303   IF i=50:CURSOR#3,360,4:PRINT#3,'Final Approach':thr=15
1304   IF i>63 AND i<65:lz=lz+.1
1305   IF i>60:ly=ly-3
1306   IF i>70:lx=lx+.5:ly=ly-2:ang=-85
1307   IF i>76:lx=lx-2.4:ly=ly+4:ang=-90:thr=10
1308   Init_LEM:LEM_Ang ang:LEM 7,ix,ly:IF ly>44:JET 2,6,ix,ly:JBeep
1309   IF ly<38
1310     CURSOR#3,360,4:PRINT#3,'Safe Landing (Esc)':Dem=0
1311     BLOCK#4,11,3,244,17,4:BEEP:Alt=0:HVel=0:VVel=0:Prt_SIM2
1312   END IF
1313 END IF
1314 IF KEYROW(1)=8:BEEP:sim=2:Intro_SIM:END IF
1315 END REPEAT Dem_lp
1316 END DEFine

```