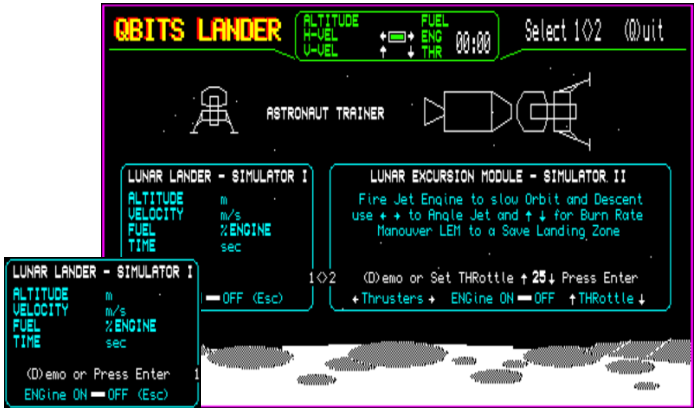
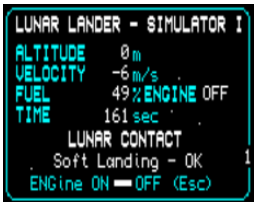


.

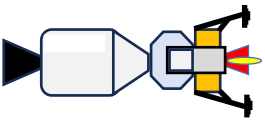
Introduction

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



QBITS Lander SIMULATOR 1 is a version of that Text based Game displaying Velocity of decent and Altitude as it reduces. The DEMO simulates the use of the Rocket Engine to slow the rate of decent and end with a soft landing.

QBITS Lander SIMULATOR II engages the Auto-Pilot, releases the Docking Clamps and Disengages the LEM from the Orbiter. A Jet blast slows the LEM's Orbit and drops it into a Decent path. Around this point the Onboard Computer Malfunctions and Manual Override is activated.



The DEMO displays a controlled flight path to enact a soft landing.

QBITS Lander - LEM Controls

Shown at top of screen **ALTITUDE**, **FUEL**, **H-VEL** Horizontal velocity and **V-VEL** Vertical velocity. Use Left ← → Right Cursors to Angle Direction and Up ↑ ↓ Down Cursors for **THR**ottle. Spacebar toggles **ENG**ine On/Off [bar shows Red/Green]. Digital clock **00:00** shows time taken**.** Press '**Esc**' to abort Play and '**Q**' to Quit Prog.

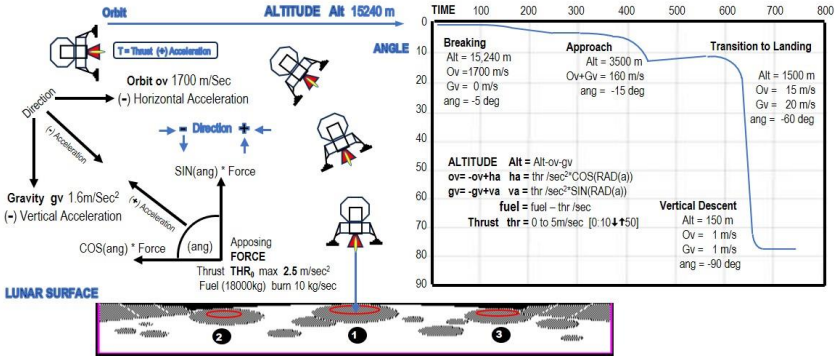
QBITS Lander - Game

Use Thrusters and Throttle to control angle, power and spacebar on/off for duration of each firing in guiding the LEM to a safe landing zone. As the LEM's Altitude and Forward Velocity reduces, movement across the Moonscape slows. There are three landing sites with varying difficulties. To make a successful landing the Horizontal and Vertical velocities and Angle of Descent must be within limits. The LEM must also be set down within the target areas or you are automatically graded as a crash landing.

QBITS Lander - LEM Simulation II

As the disengaged CSM fades into the distance Auto-Pilot action starts with the Jet angle set between 5^0 and 15^0 degrees to the horizontal to slow the LEM Orbital speed from 1700m/sec to 160m/sec. Then begins the main decent from 16,500m to the Moon surface. As Orbital (Horizontal velocity) slows Gravity (Vertical velocity) increases and further changes to Jet Angle are made to cut the rate of decent.

Approaching the moon surface changes in Jet Angle can manoeuvre the LEM to target the landing zone. The aim is for Horizontal and Vertical velocities to be less than 6m/Sec with a final Jet burst just prior to touch down leaving a drop of a less than 5m for a soft touchdown. A good Flight Path should take approximately twelve and a half minutes.



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

QBITS Lander - LEM Flight Calculations

The optimum Flight Path is a soft landing in shortest time and with minimal fuel used. Orbital Velocity **ov** and Gravity **gv** are viewed as Negative (**-**) velocities. The opposing Positive (**+**) velocities, the Thrust from the Jet Engine, Horizontal acceleration **ha** and Vertical acceleration **va** each dependent on Jet Angle **ang** 0 - 180^0 Throttle **thr** setting of 10 to 50 and length of time Jet **ENG**ine is **ON**.

ha = COS(RAD(ang))*(thr/20)*jt : ang [.999<>.001] : **thr** = 0.5 <> 2.5 in 0.25 steps
va = SIN(RAD(ang))*(thr/20)*jt : ang [.001<>.999] : **jt** = sec/sec count while ENG ON

Gravitational pull increases exponentially as Orbital velocity is reduced by the LEM's Jet [Horizontal Acceleration **ha]**. The higher the thrust the faster reduction in Orbital speed.

gv = 1.6 - 1.6*1700/**ov** and is proportional to Orbital velocity [**ov** = 1700 – **ha**]

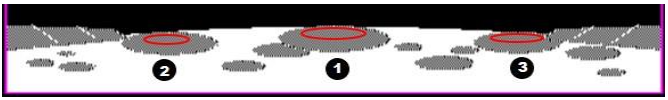
For simplicity **gv** = -1.6m/sec/sec*1700/**ov.** The Rate of Descent is controlled by varying the Jets Angle to give a more downward Thrust [Vertical acceleration **va]**.

ov orbital velocity : ha horizontal acceleration : gv gravitation velocity : va vertical acceleration

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, ha, gv, va, eng, thr, jt**

QBITS LEM - MoonScape

The Moon surface is represented as an arc across the screen scattered with craters and with hills to left and right. The three possible Landing areas are highlighted once the LEM starts its descent from Orbit. The site is selected randomly and being different sizes and positions adds to the complexity of making a soft touchdown.



QBITS LEM - Monitor

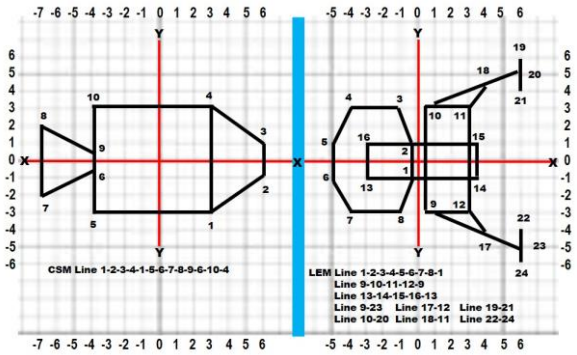
ALTITUDE: Alt=Alt-VVel	16500m <> 0	**FUEL**: 18000kg
Horizontal Velocity: **HVel**=HVel-ha	1700m/s <> 0	ha=INT(COS(RAD(ang))*thr)
Vertical Velocity: **VVel**=VVel-gv+va	1.6m/s/s <> 0	va=INT(SIN(RAD(ang))*thr)

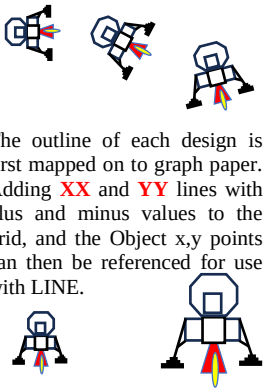
Check the Monitor display for **ALTITUDE, FUEL** remaining, **H-Vel** & **V-Vel**. If Jet is **On**, Bar shows RED and if sound active gives a rumbling background noise.

Check **THRottle** level and position of LEM with regards to the target landing area. Even if on target the Horizontal, Vertical speeds and Angle of Final decent must be within limits or LEM will crash land.

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.





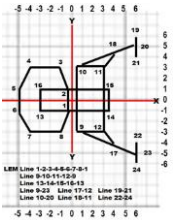
The outline of each design is first mapped on to graph paper. Adding **XX** and **YY** lines with plus and minus values to the grid, and the Object x,y points can then be referenced for use with LINE.

Note: LEM 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.

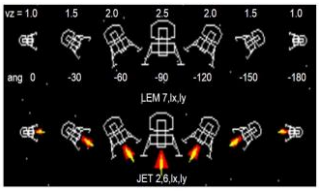
QBITS LANDER Code

```
1000 REMark QBITS_Lander_bas [QBITS Lander 2024 QL40th - QPC2] vM30

1002 dev$='win1_':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

1009 REMark Arrays
1010 DIM CM(13,2),LM(40,2),MS(40,2),vx(40),vy(40),x%(8),y%(8)

1018 Init_win:sim=1:Intro_SIM

1020 DEFine PROCedure Init_win
1021 OPEN#6,scr_:WINDOW#6,486,98,gx+12,gy+98:CLS#6:SCALE #6,100,0,0
1022 OPEN#5,scr_:WINDOW#5,508,20,2+gx,220+gy:SCALE#5,20,0,0
1023 OPEN#4,scr_:WINDOW#4,500,60,6+gx,2+gy
1024 OPEN#3,scr_:WINDOW#3,500,36,6+gx,2+gy:SCALE#3,20,0,0
1025 WINDOW#2,512,224,gx,gy          :BORDER#2,1,3:PAPER#2,0:CLS#2
1026 WINDOW#1,508,212,2+gx,gy+36 :INK 7:SCALE 200,-178,-40
1027 WINDOW#0,512,32,gx,224+gy     :CLS#0:BORDER#0,1,3:PAPER#0,7:INK#0,0:CLS#0
1028 OVER#3,1:CSIZE#3,2,1:CLS
1029 INK#3,2:FOR i=0 TO 1:CURSOR#3,2+i,4:PRINT#3,'QBITS LANDER'
1030 INK#3,6:FOR i=0 TO 1:CURSOR#3,4+i,5:PRINT#3,'QBITS LANDER'
1031 OVER#3,0:CURSOR#3,0,0:CSIZE#3,0,1
1032 REMark *** Display Panel ***
1033 INK#3,4:LINE#3,0,5 TO 60,5 TO 68,2:LINE#3,143,2 TO 151,5 TO 212,5
1034 RESTORE 1036:sx=67:LINE#3,sx+4,1
1035 FOR i=1 TO 4:READ x1,y1,x2,y2:LINE#3 TO sx+x1,y1:ARC#3 TO sx+x2,y2,PI/2
1036 DATA 72,1,75,4,75,16,72,19,4,19,1,16,1,4,4,1
1037 RESTORE 1038:FOR a=1 TO 8:READ bi,bx,by,str$:QBold 4,bi,1,bx,by,str$
1038 DATA 4,168,3,'ALTITUDE',4,168,12,'H-VEL',4,168,22,'V-VEL',4,270,3,'FUEL'
1039 DATA 4,270,13,'ENG',4,270,23,'THR',7,234,12,'←  →',7,234,22,'↑  ↓'
1040 BLOCK#4,15,7,242,15,7:BLOCK#4,13,5,243,16,0:BLOCK#4,11,3,244,17,4
1041 INK#3,7:CURSOR#3,300,13:PRINT#3,'00:00':tm=0
1042 END DEFine

1044 DEFine PROCedure Init_LEM
1045 RESTORE 1044:FOR a=1 TO 39:READ LM(a,1),LM(a,2)
1046 DATA -.5,-1,-.5,1,-1,3,-4,3,-5,1,-5,-1,-4,-3,-1,-3,-.5,-1
1047 DATA  .5,-3,.5,3,3,3,3,3,3.-3,.5,-3        :REMark Lander Module
1048 DATA  -3,-1,3.8,-1,3.8,1,-3,1,-3,-1        :REMark Hatch
1049 DATA 1, 3,6,  5,4, 4,3, 3, 6, 4,6, 6       :REMark Right Leg/Foot
1050 DATA 1,-3,6,-5,4,-4,3,-3, 6,-4,6,-6        :REMark Left  Leg/Foot
1051 DATA 5,.6,8,1,8,-1,5,-.6                   :REMark Red Jet Flame
1052 DATA 6,0,8,.4,10,0,8,-.4                   :REMark White Jet Flame
1053 END DEFine

1055 DEFine PROCedure QBold(ch,bi,bs,bx,by,b$)
1056 INK#ch,bi:FOR q=0 TO 1:OVER#ch,bs:CURSOR#ch,bx,by+q:PRINT#ch,b$:OVER#ch,0
1057 END DEFine
```

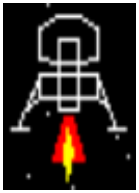


```
1059 DEFine PROCedure CSM(col,cx,cy)
1060 RESTORE 1062:INK col:LINE cx+3*cz,cy-3*cz
1061 FOR z=1 TO 12:READ cx1,cy1:LINE TO cx+cx1*cz,cy+cy1*cz
1062 DATA 6,-1,6,1,3,3,3,-3,-4,-3,-4,-.5,-7,-2,-7,2,-4,.5,-4,-2,-4,3,3,3
1060 RESTORE 1060:INK col:LINE cx+4*cz,cy-3*cz
1063 END DEFine
```

```
1065 DEFine PROCedure LEM_Ang(r)
1066 FOR a=1 TO 39
1067   vx(a)=LM(a,1)*COS(RAD(r))-LM(a,2)*SIN(RAD(r))
1068   vy(a)=LM(a,1)*SIN(RAD(r))+LM(a,2)*COS(RAD(r))
1069   LM(a,1)=vx(a):LM(a,2)=vy(a)
1070 END FOR a
1071 END DEFine
```




```
1073 DEFine PROCedure LEM(col,lx,ly)
1074 INK col:LINE lx+LM(1,1)*lz,ly+LM(1,2)*lz
1075 FOR z=2 TO 9:LINE TO lx+LM(z,1)*lz,ly+LM(z,2)*lz
1076 LINE lx+LM(10,1)*lz,ly+LM(10,2)*lz
1077 FOR z=11 TO 14:LINE TO lx+LM(z,1)*vz,ly+LM(z,2)*lz
1078 LINE lx+LM(15,1)*lz,ly+LM(15,2)*lz
1079 FOR z=16 TO 19:LINE TO lx+LM(z,1)*lz,ly+LM(z,2)*lz
1080 FOR z=20 TO 30 STEP 2
1081   LINE lx+LM(z,1)*vz,ly+LM(z,2)*lz TO lx+LM(z+1,1)*vz,ly+LM(z+1,2)*lz
1082 END FOR z
1083 END DEFine
```



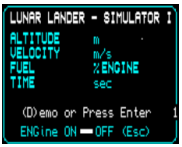
```
1085 DEFine PROCedure JET(col1,col2,lx,ly)
1086 FOR z=1 TO 8:x%(z)=lx+LM(31+z,1)*lz:y%(z)=ly+LM(31+z,2)*lz
1087 INK col1:FILL 1:LINE x%(1),y%(1) TO x%(2),y%(2) TO x%(3),y%(3)
1088 LINE TO x%(4),y%(4) TO x%(1),y%(1):FILL 0
1089 INK col2:FILL 1:LINE x%(5),y%(5) TO x%(6),y%(6) TO x%(7),y%(7)
1090 LINE TO x%(8),y%(8) TO x%(5),y%(5):FILL 0
1091 END DEFine
```

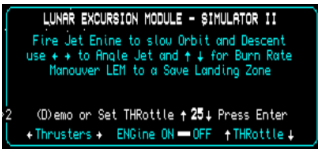
```
1093 DEFine PROCedure JBeep
1094 BEEP 20000,40,100,50,5,0,0,0              :REMark Jet Engine Sound
1095 END DEFine
```

```
1097 DEFine PROCedure LEM_Crash
1098 INK 2:FILL 1:CIRCLE lx,ly,5:FILL 0:INK 6
1099 FOR i=1 TO 18:a=i*20:LINE lx,ly TO lx+COS(RAD(a))*12,ly+SIN(RAD(a))*12
1100 END DEFine
```

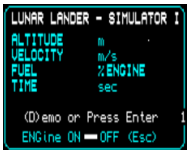
```
1102 DEFine PROCedure MoonScape
1103 BLOCK 508,29,0,183,7: BLOCK 200,2,154,181,7: BLOCK 120,2,194,180,7
1104 RESTORE 1106: INK 248
1105 FOR i=1 TO 13:READ mx,my,mr:FILL 1:CIRCLE mx,my,mr,.25,PI/2:FILL 0
1106 DATA -160,-30,8,-146,-24,5,-140,-32,10,-90,-18,26,-50,-32,12,-28,-22,16
1107 DATA 0,-16,30,40,-20,8,60,-30,16,70,-22,8,100,-18,24,140,-24,15,150,-35,8
1108 FILL 1 :LINE -178,-8 TO -120,-10 TO -112,-15 TO -178,-22 TO -178,-8:FILL 0
1109 FILL 1: LINE 178,-8 TO 120,-10 TO 112,-15 TO 178,-22 TO 178,-8:FILL 0
1110 INK 7: LINE 165,-7 TO 150,-18:LINE 140,-8 TO 130,-15
1111 LINE -165,-7 TO -150,-18:LINE -140,-8 TO -130,-15
1112 END DEFine
```

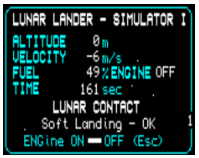
```
1150 REMark INTRO SIMULATOR I-II

1152 DEFine PROCedure Intro_SIM
1153 CSIZE 0,0:CLS:MoonScape:QBold 1,7,1,140,28,'ASTRONAUT TRAINER'
1154 cz=4:lz=4:ang=0:Init_LEM:CSM 7,44,130: LEM 7,92,130
1155 lz=2.5:LEM_Ang -90:lx=-112:ly=132:LEM 7,lx,ly
1156 ch=6:QBold ch,7,1,10,6,'LUNAR LANDER - SIMULATOR I'
1157 RESTORE 1159:INK#ch,5:LINE#ch,8,1
1158 FOR i=1 TO 4:READ x1,y1,x2,y2:LINE#ch TO x1,y1:ARC#ch TO x2,y2,PI/2
1159 DATA 124,1,129,6, 129,92,124,97, 8,97,3,92, 3,6,8,1
1160 QBold ch,7,1,220,6,'LUNAR EXCURSION MODULE - SIMULATOR II'
1161 RESTORE 1164:INK#ch,5:LINE#ch,146,1
1162 QBold 6,7,0,204,83,'←  →':QBold 6,7,0,392,83,' ↑  ↓'
1163 FOR i=1 TO 4:READ x1,y1,x2,y2:LINE#ch TO x1,y1:ARC#ch TO x2,y2,PI/2
1164 DATA 360,1,365,6, 365,92,360,97, 146,97,141,92, 141,6,146,1
1165 FOR i=1 TO 5:READ x,y,str$:QBold ch,5,1,x,y,str$
1166 DATA 10,20,'HEIGHT',10,30,'VELOCITY',10,40,'FUEL',98,40,'ENGINE',10,50,'TIME'
1167 FOR i=1 TO 11:READ x,y,str$:QBold ch,5,0,x,y,str$
1168 DATA 90,20,'m',90,30,'m/s',90,40,'%',90,50,'sec',20,83,'ENGine ON  OFF (Esc)'
1169 DATA 210,20,'Fire Jet Enine to slow Orbit and Descent'
1170 DATA 204,30,'use ← → to Angle Jet and↑ ↓ for Burn Rate'
1171 DATA 226,40,'Manouver LEM to a Save Landing Zone'
1172 DATA 212,83,'Thrusters',290,83,'ENGine ON  OFF',400,83,'THRottle'
1173 QBold 3,7,0,360,4,'Select 1<>2  (Q)uit ':QBold 6,7,0,166,70,'1<>2'
1174 BLOCK#ch,12,3,78,87,7:BLOCK#ch,12,3,348,87,7:k=0:thr=25
1175 REPeat Intro_lp
1176  IF sim=1
1177    IF k=10:Dem=10:Run_SIM1
1178    IF k=68 OR k=100:Dem=5:Run_SIM1     Note: (D) DEMO
1179    BLOCK#ch,250,10,210,70,0
1180    QBold ch,7,0,20,70,'(D)emo or Press Enter'
1181  END IF
1182  IF sim=2
1183    IF k= 10:ap=RND(20 to 26):Auto_SIM2:Dem=10:Landsite RND(1 TO 3):Run_SIM2
1184    IF k=68 OR k=100:Auto_SIM2:Dem=5:thr=50:Demo_SIM2
1185    IF k=208 AND thr<50:thr=thr+5
1186    IF k=216 AND thr>10:thr=thr-5
1187    QBold ch,7,0,212,70,'(D)emo or Set THRottle ↑  ↓ Press Enter'
1188    BLOCK#ch,150,24,6,60,0 :QBold ch,7,1,360,70,thr
1189  END IF
1190  k=CODE(INKEY$(-1))  :IF k=81 OR k=113:LRUN dn$:STOP
1191  IF k=49:sim=1:END IF :IF k=50:sim=2:END IF
1192 END REPeat Intro_lp
1193 END DEFine
```





```
1200 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

1202 DEFine PROCedure Run_SIM1
1203 BLOCK#ch,148,10,12,70,0 : TIMER=0 : t0=Timer : lastime=t0:chk=0
1204 eng=0:gm=-1.62:Alt=10000:Fuel=100:gv=0
1205 REPeat Sim_lp
1206   CURSOR#ch,58,20:PRINT#ch,FILL$(' ',5-LEN(INT(Alt)))&INT(Alt)
1207   CURSOR#ch,58,30:PRINT#ch,FILL$(' ',5-LEN(INT(gv)))&INT(gv)
1208   IF eng=  0:CURSOR#ch,138,40:PRINT#ch,'OFF':BEEP
1209   IF eng=  5:CURSOR#ch,138,40:PRINT#ch,'ON '
1210   IF Fuel>  0:CURSOR#ch, 64,40:PRINT#ch,FILL$(' ',4-LEN(Fuel))&Fuel
1211   IF Fuel<=0:CURSOR#ch, 64,40:PRINT#ch,' OUT'
1212   CURSOR#ch,64,51:PRINT#ch,FILL$(' ';4-LEN(INT(elt)))&INT(elt)
1213   PAUSE Dem:TIMER=TIMER+1
1214   IF KEYROW(1)=8 OR Alt>10000:chk=1:End_SIM1:EXIT Sim_lp
1215   IF Dem=10 AND KEYROW(1)=64:IF eng=0:eng=5:ELSE eng=0:JET 0,0,lx,ly
1216   IF Dem=5 AND Alt<3500 AND Alt>1500:eng=5
1217   IF Dem=5 AND gv>-5:eng=0:BEEP:JET 0,0,lx,ly
1218   IF Dem=5 AND Alt<150 AND gv<-5:eng=5
1219   IF Fuel<=0:eng=0:BEEP:JET 0,0,lx,ly
1220   nowtime=TIMER:elt=nowtime-t0:deltat=nowtime-lastime:lastime=nowtime
1221   accel=gm+eng                              :REMark Acceleration Gravity -1.6 + Thrust 10
1222   gv=gv+accel*deltat                        :REMark Velocity (-) or (+)
1223   Alt=Alt+gv*deltat+.5*accel*deltat*deltat  :REMark Decent New Altitude
1224   Fuel=Fuel-eng*deltat/5                    :REMark Fuel used
1225   IF Fuel>0 AND eng=5:JET 2,6,lx,ly:JBeep   :REMark Jet Flame & Sound
1226   IF Alt<0:End_SIM1:EXIT Sim_lp
1227 END REPeat Sim_lp
1228 REPeat Ans_lp
1229   IF KEYROW(4)=8 OR KEYROW(4)=8:sim=1:EXIT Ans_lp
1230   IF KEYROW(6)=2:sim=2:EXIT Ans_lp
1231 END REPeat Ans_lp
1232 BLOCK 30,10,40,36,0: BLOCK #ch,160,24,6,60,0:k=0
2323 END DEFine

1235 DEFine PROCedure End_SIM1
1236                       QBold ch,7,0,138,40,'OFF':JET 0,0,lx,ly:BEEP
1237 IF chk=1:             QBold ch,7,1,46,72,'SIM - ABORTED'    :RETurn
1238                       QBold ch,7,1,46,62,'LUNAR CONTACT'
1239 IF gv>=-5 AND Alt<10:QBold ch,7,0,36,72,'Soft Landing – OK ' :RETurn
1240                       QBold ch,7,0,24,72,'Crash Landing - FATAL'
1241 END DEFine
```





```
1250 REMark SIM2 - LUNAR EXCURSION MODULE

1252 DEFine PROCedure Auto_SIM2
1253 Alt=16500:HVel=1700:VVel=0:Fuel=16000:BLOCK 490,160,10,6,0
1254 FOR s=1 TO 60:POINT#1,RND(-160 TO 160),RND(5 TO 170)
1255 Init_LEM:LEM_Ang 0:cz=1.2:cx=-144:cy=150:CSM 7,cx,cy:
1256 lz=1.2:lx=-129:ly=150:LEM 7,lx,ly:eng=4:td=2:tm=0:RESTORE 1258
1257 FOR i=1 TO 3:READ Mes$:CURSOR#3,360,4:PRINT#3,Mes$:CLS#3,4:PAUSE 30
1258 DATA 'Release Docking Clamps','Disengage LEM','Engage LEM Auto-Pilot'
1259 FOR i=1 TO 4
1260   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
1261   CSM 7,cx,cy:LEM 7,lx,ly:PAUSE 5:tm=tm+5:Prt_SIM2
1262 END FOR i
1263 CSM 0,cx,cy:LEM 0,lx,ly:LEM_Ang -5:LEM 7,lx,ly:eng=2
1264 CURSOR#3,360,4:PRINT#3,'Reduce Orbital Speed  ':BLOCK#3,11,3,244,17,eng
1265 FOR a=1 TO ap
1266   LEM 0,lx,ly:JET 0,0,lx,ly:Fuel=Fuel-thr*5
1267   IF a>20:ang=ang-5:Init_LEM:LEM_Ang ang
1268   lx=lx+5-(1700/HVel)*1.2:ly=ly-.2:LEM 7,lx,ly:JET 2,6,lx,ly:JBeep
1269   FOR b=1 TO 5
1270     tm=tm+6:Alt=Alt-2.5*a*b:HVel=HVel-thr/5:VVel=-tm*1.5:Prt_SIM2:PAUSE 1
1271   END FOR b
1272 END FOR a
1273 END DEFine

1275 DEFine PROCedure Run_SIM2
1276 CURSOR#3,360,5:PRINT#3,'Computer Malfunction':CLS#3,4:yy=5E-2:xx=-.5:td=2
1277 REPeat M_lp
1278   PAUSE Dem:tm=tm+1:get_keys:BLOCK#4,11,3,244,17,eng
1279   IF eng=2 AND Fuel>=thr
1280     Fuel=Fuel-thr
1281     xx=xx+INT(COS(RAD(ang))*thr)/500 : yy=yy-INT(SIN(RAD(ang))*thr)/500
1282   ELSE
1283     yy=yy-5E-2
1284   END IF                                          Note: Screen Boundaries
1285   IF lx+xx<-190 OR lx+xx>190 OR ly+yy>185 OR KEYROW(1)=8:td=3:EXIT M_lp
1286   IF ly<-4
1287     IF lx>lsx-lw AND lsx<ls+lw
1288       IF ABS(HVel)<12 AND VVel>-8 AND ang>-95 AND ang<-85:td=1
1289     END IF
1290     EXIT M_lp
1291   END IF
1292   HVel=xx*lxm : VVel=yy*lym:LEM 0,lx,ly:IF ly>8:JET 0,0,lx,ly
1293   lx=lx-xx:ly=ly+yy:Prt_SIM2:Init_LEM:LEM_Ang ang:LEM 7,lx,ly
1294   IF ly>8 AND eng=2 AND Fuel>=thr:JET 2,6,lx,ly
1295 END REPeat M_lp
1296 eng=4:BLOCK#4,11,3,244,17,eng:Alt=0:Prt_SIM2
1297 IF td=1:CURSOR#3,360,5:PRINT#3,'Soft Landing  (Esc)':CLS#3,4
1298 IF td=2:CURSOR#3,360,5:PRINT#3,'Crash Landing (Esc)':CLS#3,4:LEM_Crash
1299 IF td=3:CURSOR#3,360,4:PRINT#3,'SIM ABORTED   (Esc)':CLS#3,4
1300 REPeat ans:IF KEYROW(1)=8:sim=2:CLS:Intro_SIM
1301 END DEFine
```

QBITS Lander - LEM Screen Calculations

The range of x,y coordinates are set by SCALE 200,-178,-40 which places 0,0 just above the central landing site. For the horizontal lx is between -178 to +178 and for vertical ly is 180 down to 0. Orbit reduction lies between 150 and 130, LEM Descent between 130 and 90 with Final Approach between 90 and 0.

After LEM separation lx,ly are -120, 150 and for Orbital reduction starting at 1700m/sec lx=lx+5 decreasing to 170m/sec lx=lx+0.5 or expressed as lx=lx+5*HVel/1700.

lx=lx+5*HVel/1700-xx [Jet Force xx=INT(COS(RAD(ang))*thr)/500]

For HVel at 1700m/sec Gravitational velocity gv=0 reducing HVel to 850m/sec and gv= -0.8, at 170m/sec gv= -0.5 and expressed as ly=ly –(1.6*(1700-HVel)/1700).

ly=ly-(1.6*(1700-HVel)/1700)+yy [Jet Force yy=INT(SIN(RAD(ang))*thr)/500]

The xx and yy being the opposing force from the LEM's Jet. As Gravitational velocity VVel increases the drop in LEM Altitude between 150 and 130 is approx. 13000m or 650m per axis point. For the LEM Descent Phase of 130 down to 90 is 2000/40 = 50m per axis point. In the Final Approach 1500m reduces to 0 and is 16.6m per axis point.

HVel==xx*lxm VVel=yy*lym

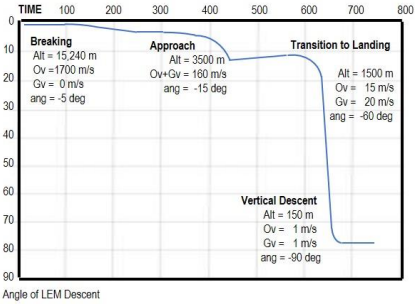
To calculate the HVel and VVel changes for Print out, lxm and lym are used as constants to represent the rate of decent with change in Altitude and for the Final approach with the lx offset to align with the selected landing area.

```
1303 DEFine PROCedure Prt_SIM2
1304 IF ly<140:Alt=.25*ly*ly/2  :lym=50:lxm=500 :REMark 4500 - 1400
1305 IF ly<  80:Alt=.2*ly*ly/2                  :REMark 1400 -  300
1306 IF ly<  40:Alt=1E-2*ly*ly/2 :lym= 5:lxm=50
1307 IF ly<= -4:Alt=0                           :REMark  0
1308 PrtNum 4,7,6,228,4,Alt                     :REMark Altitude
1309 PrtNum 4,7,5,198,13,HVel                   :REMark Horizontal velocity
1310 PrtNum 4,7,5,198,23,VVel                   :REMark Vertical velocity
1311 QBold 4,7,0,243,22,thr                     :REMark 0 - 50 Throttle
1312 IF Fuel<=0:QBold 4,7,0,300,5,' OUT ':ELSE PrtNum 4,7,5,300, 4,Fuel
1313 TMin=tm DIV    60:CURSOR#3,300,13:PRINT#3,FILL$('0',2-LEN(TMin))&TMin
1314 TSec=tm MOD 60:CURSOR#3,318,13:PRINT#3,FILL$('0',2-LEN(TSec))&TSec
1315 END DEFine

1317 DEFine PROCedure PrtNum(ch,ni,ns,nx,ny,num)
1318 num$=INT(num):INK#ch,ni:CURSOR#ch,nx,ny:PRINT#ch,FILL$(' ',ns-LEN(num$))&num$
1319 END DEFine

1321 DEFine PROCedure Landsite(ls)
1322 IF ls=3:lw=  9:lsx= -90:lsy= -6:lsr=15
1323 IF ls=2:lw=12:lsx=   0:lsy= -4:lsr=18
1324 IF ls=1:lw=14:lsx=100:lsy= -6:lsr=20
1325 INK 2:CIRCLE lsx,lsy,lsr,.2,PI/2
1326 END DEFine
```

Note: Landing Site Location and Boundaries

```
1328 DEFine PROCedure get_keys
1329 k=KEYROW(1):tx=xx:ty=yy
1330 SELect ON k
1331   =  2:IF ang>-180:ang=ang-5
1332   = 16:IF ang<   0:ang=ang+5
1333   =  4:IF thr<    50:thr=thr+5
1334   =128:IF thr>   10:thr=thr-5
1335   = 64:IF eng=4 AND Fuel>0:st=tm:eng=2:ELSE eng=4
1336 END SELect
1337 IF xx>11 OR xx<-11 :xx=tx
1338 IF yy>8 OR yy<-8   :yy=ty
1339 END DEFine
```

QBITS LEM - DEMO FlightPath




```
1350 DEFine PROCedure Demo_SIM2
1351 REPeat Dem_lp
1352  IF Dem=5
1353    xx=lx:yy=ly:PAUSE Dem:i=i+1:Fuel=Fuel-thr*5:IF i<50:thr=10
1354    lx=lx+1:ly=ly-.5:tm=tm+1:HVel = -lx+5:VVel =-ly+38:Prt_SIM2
1355    IF i=30:CURSOR#3,360,4:PRINT#3,'LEM Descent          '
1356    IF i>30 AND i<48:ang=ang-5:ly=ly-1.5
1357    IF i=50:CURSOR#3,360,4:PRINT#3,'Final Approach':thr=15
1358    IF i>62 AND i<66:lz=lz+.1
1359    IF i>66:ly=ly-2:INK 2:CIRCLE 0,-13,15,.25,PI/2
1360    IF i>72:lx=lx-.5:ly=ly+2:ang=-80
1361    IF i>78:lx=lx-1:ly=ly-1.5:ang=-90:thr=10
1362    LEM 0,xx,yy:If ly>4:JET 0,0,xx,yy:BEEP
1363    Init_LEM:LEM_Ang ang:LEM 7,lx,ly:IF ly>8:JET 2,6,lx,ly:JBeep
1364    IF ly<0
1365      CURSOR#3,360,4:PRINT#3,'Safe Landing   (Esc)':Dem=0
1366      BLOCK#4,11,3,244,17,4:BEEP:Alt=0:HVel=0:VVel=0:Prt_SIM2
1367    END IF
1368  END IF
1369  IF KEYROW(1)=8:BEEP:sim=2:LEM 0,lx,ly:Intro_SIM:END IF
1370 END REPeat Dem_lp
1371 END DEFine
```

Note: Central Landing Site