

Making of Jacks Bubble Island by Ross Simpson

I created this game from "a few bits and pieces" Source Code I had lying around which included some BASIC code, and a Sprite Driver I'd written in Assembly.

With the help of Peter Campbell, a small BASIC programme was video displayed on CPCWiki Forum showing a series of Randomly generated numbers between 1 and 58. The original appears to have been sourced from the WACCI PD Library, though during the coding of this game it helped me create my own, which would generate random positions for the final Fruit and Vegetables placement in various spots and not necessarily in sequence.

The Number Placement Problem

The Following Code is a BASIC representation of what I came up with:

```
100 RESTORE 150
110 CLEAR:DIM a(16),t(8)
120 FOR p=1 TO 15 STEP 2
130   READ a(p),a(p+1)
140 NEXT p
150 DATA 2,3,2,6,2,9,2,12
160 DATA 18,3,18,6,18,9,18,12
200 FOR s=1 TO 7 STEP 2
210   n=INT(RND*15)+1
220   IF n MOD 2=0 THEN GOTO 210
230   IF a(n)<>0 THEN t(s)=a(n):t(s+1)=a(a+1):a(n)=0:a(n+1)=0 ELSE GOTO 210
240 NEXT s
250 CLS:PRINT"My 4 Picks are:-":PRINT
260 FOR s=1 TO 7 STEP 2
270   PRINT "X:";t(s);"Y:";t(s+1)
280 NEXT s
290 END
```

Earlier Versions of the Game

A bit of BASIC was evident in the earlier version, with BASIC routines being used to setup the screen, once they carried out their operation, an assembly routine was used to draw the screen. The game was initially a character running around the screen with a simple jumping routine and checks to detect when positioned over a Ladder.

Ideas for game development

Perhaps no surprise for the name of this game has come from the popular Amstrad CPC games Bomb Jack, Bubble Bobble & Raindow Islands with a simple transfer of the 's' from Islands moved after Jack and I now have Jacks Bubble Island! I liked the idea of having a game like Bomb Jack where you collect the fruit in a series and getting extra bonus from it, the Bubbles have now become the hazard to avoid and Fruit and Veg left scatted about is perhaps the only resemblance the game has with Rainbow Islands.

Adding the Fruit and Veg to the game

This involved writing another routine, and having a man moving around the screen and checking for appropriate Scoring Increments, this was what I came up with:

```
100 MODE 0:BORDER 11:INK 0,11:INK 1,26
110 DEFINT a-z:DIM f1(4),f2(4),o(4)
120 x=10:y=10:LOCATE x,y:PRINT CHR$(248)
130 w=1:z=1:o(1)=1:LOCATE w,z:PRINT CHR$(228):f1(1)=w:f2(1)=z
140 w=1:z=25:o(2)=2:LOCATE w,z:PRINT CHR$(229):f1(2)=w:f2(2)=z
150 w=20:z=25:o(3)=3:LOCATE w,z:PRINT CHR$(230):f1(3)=w:f2(3)=z
160 w=20:z=1:o(4)=4:LOCATE w,z:PRINT CHR$(231):f1(4)=w:f2(4)=z
170 bn=25:sc=0:p=1
180 WHILE p<>5
190   IF INKEY(8)=0 THEN IF x>1 THEN LOCATE x,y:PRINT " ";:x=x-1:LOCATE x,y:PRINT
CHR$(248);
200   IF INKEY(1)=0 THEN IF x<20 THEN LOCATE x,y:PRINT " ";:x=x+1:LOCATE
x,y:PRINT CHR$(248);
210   IF INKEY(0)=0 THEN IF y>1 THEN LOCATE x,y:PRINT " ";:y=y-1:LOCATE x,y:PRINT
CHR$(248);
220   IF INKEY(2)=0 THEN IF y<25 THEN LOCATE x,y:PRINT " ";:y=y+1:LOCATE
x,y:PRINT CHR$(248);
230   IF x=f1(p) THEN IF y=f2(p) THEN IF o(p)<>0 THEN sc=sc+bn:p=p+1
240   IF x=f1(2) THEN IF y=f2(2) THEN IF p<>2 THEN sc=sc+o(2):o(2)=0
250   IF x=f1(3) THEN IF y=f2(3) THEN IF p<>3 THEN sc=sc+o(3):o(3)=0
260   IF x=f1(4) THEN IF y=f2(4) THEN IF p<>4 THEN sc=sc+o(4):o(4)=0
270   LOCATE 10,25:PRINT sc;
280   IF o(2)=0 THEN IF o(3)=0 THEN IF o(4)=0 THEN p=5
290 WEND
```

The process used in this is just like the game where if something is collected out of sequence minor points are scored and potentially limiting the bonus for whatever fruit and veg will still score the bonus points, the only difference here I found was to make this work in the final game the routine had to be coded in Assembly to speed up the checks.

Adding the Hazards and jumping them

It also involved writing some BASIC, which was eventually converted to Assembly to take advantage of the speed with the Jumping routine staying put in the main BASIC game. I started writing it in BASIC to deal with the problem of how to speed up the hazard and this was what I coded in BASIC:

```
100 MODE 0:BORDER 11:INK 0,11:INK 1,26
110 DEFINT a-z:d=0:w=10:s=0
120 x=10:y=10:GOSUB 1000
130 b=20:c=10:GOSUB 2000
140 WHILE d=0
150   IF INKEY(0)=0 THEN GOSUB 1030:y=y-1:GOSUB 1050:y=y+1:GOSUB 1000
160   GOSUB 2060
170   IF d=0 THEN GOSUB 2030:b=b-1:IF b<1 THEN b=20
180   GOSUB 2000
190   LOCATE 1,1:PEN 1:PRINT"Score:";s
200 WEND:IF d=2 THEN PRINT"Well Done!"
210 WHILE INKEY$<>"":WEND:PEN 1:END
1000 LOCATE x,y
1010 PEN 1:PRINT CHR$(248)
1020 RETURN
1030 LOCATE x,y:PRINT" "
1040 RETURN
1050 l=0
1060 WHILE l<3
1070   LOCATE x,y
1080   PEN 1:PRINT CHR$(248)
1090   l=l+1
1100   GOSUB 2030:b=b-1:IF b<1 THEN b=20
1110   GOSUB 2000
1120   GOSUB 2060
1130 WEND
1140 LOCATE x,y:PRINT" "
1150 RETURN
2000 FOR z=1 TO w
2010   LOCATE b,c:PEN 3:PRINT"O"
2020 NEXT z:RETURN
2030 LOCATE b,c
2040 PRINT" "
2050 RETURN
2060 IF y=10 THEN IF b=10 THEN d=1
2070 IF y=9 THEN IF b=10 THEN s=s+1
2080 IF s=3 THEN s=0:w=w-1
2090 IF w=0 THEN d=2
2100 RETURN
```

It works by placing a man in the path of a hazard, the man has to jump it in order to avoid death as it does the score is incremented, jump 3 of those and the score returns to zero with the speed of the hazard reduced to increase movement. For the final game the Bubble movement was altered to Assembly and a simple routine added to check when a certain number was reached and if it was move the Bubble. As the game progresses the value is lowered and likewise the speed of the Bubbles are increased for each level. There are 5 levels in all with a bit of a Surprise at the end of the game and a little bit extra if you manage to get a perfect score in the game.