

THREE SINCLAIR 1K WONDERS

Many games players find the limits imposed by 1K of memory as a frustration to be borne until they have saved up enough money to expand their computer.

Others accept the challenge to produce a worthwhile game within these restrictions and apply themselves via a little lat-

eral thinking to come up with some of the most imaginative of computer games.

Here we present three such games, kindly donated by our inspired readers for Sinclair ZX81 owners to try out: **Android Attack**, **Mole** and **Catch-a-Bag-of-Swag**.

The first of our 1K wonders is **Android Attack** by Michael Kincaid and David Chapman.

The aim of the game is to shoot down as many advancing androids as possible before three escape your fire.

One point for each ailing alien and once the score gets past 10 the going gets harder as the creatures start from closer-in. The game is over when three androids have escaped.

To move your gun, the commands are: U- to move up; N- to move down; F- to fire.

The variables used are:

K = Your score

C = Number of androids that have passed you.

D = Where android starts across the screen.

P = Line where your gun is.

S = Line where android is printed on.

V = Movement of your gun.

T = Count for moving android across screen.

Iain McWilliams' **Mole** game is an original test of foresight and dexterity.

The program simulates a mole diving deep into his burrow. Unfortunately he is blind and has to be guided down using the "N" and "M" keys on the Sinclair keyboard.

Projections which have to be negotiated, stick out from either side of the tunnel and a point is scored for each move until the poor mole stumbles into a wall.

For such a simple game it is surprisingly addictive.

List of variables:

A = FOR ... NEXT loops and position of passage

P = Position of mole.

R = Character at mole's next position.

S = Counter of score.

List of subroutines:

At line 20 = print out passage.

At line 24 = check for next position of mole. Print out mole. Increase score and create next position of mole.

Catch-a-Bag-of-Swag by Julian Downes invites you to catch the money as it falls in bags from the bank above.

You have control of a catching "bucket" which can move left or right, using the "5" and "8" keys.

The bags are of random value and you must catch as many as possible to make the "blag" worthwhile. Once 15 bags have been dropped the game starts again. A check is kept of the highest score so make sure you catch the valuable ones.

A rundown of the program follows:

10-30 — assign variables — H = High score X = catcher position S = Score.

40 — M counts the number of bags dropped

50-60 — A determines (randomly) which bag falls B determines (randomly) its value.

70 — displays bank

80-140 — Makes bag fall & moves catcher

160 — Has catcher caught bag?

170-190 — if "Yes" then adjust score

200-210 — otherwise continue.

220-250 — Once 15 bags have been dropped, check for new high-score and adjust if necessary. Then start again.



ANDROID ATTACK

BY MICHAEL KINCAID & DAVID CHAPMAN

```

1 LET K = 0
2 LET C = 0
3 LET D = 1
5 LET P = 4
6 LET S = 4
20 FOR T = D TO 28
30 PRINT AT S,T; " ■■■ "
35 PRINT AT P,25; "<"
45 LET V = 0
50 IF INKEY $ = "U" THEN LET V = -4
60 IF INKEY $ = "N" THEN LET V = 4
65 IF INKEY $ = "F" AND P = S THEN GOTO 200
70 IF P+V <= 3 THEN LET V = 0
85 PRINT AT P,25; " "
90 LET P = P+V
100 NEXT T
105 LET C = C+1
106 IF C = 3 THEN GOTO 230
110 LET S = INT (RND*4)*4+4
120 IF S >= 16 THEN LET S=4
130 GOTO 20
200 PRINT AT S,T; " "
210 LET K = K+1
215 IF K > 10 THEN LET D = D+2
220 GOTO 110
230 PRINT "SCORE = "; K
    
```