

Meet Smiley and beat the galloping Grumpies in
ROLAND WADDILOVE's version of an old favourite

Introducing Smiley...

In this colourful and competitive game you have to guide Smiley round a maze picking up coloured buttons that are scattered about. The buttons belong to the Grumpies, who take exception to you plucking them from under their nose and chase you round the maze, trying to catch you.

If you collect all the buttons in the maze it is redrawn and the game starts all over again, except that you and the Grumpies move a little faster.

Yes, it's an Amstrad version of the old favourite - no collection is complete without it! There is a bonus which decreases with time, three lives and a keyboard or joystick option.

The program is fully structured with no GOTOs to confuse you. It is controlled by calling the various subroutines from lines 40 to 310 to print the instructions, draw the maze and move the characters.

Each subroutine has been given a title in a REM statement at the start of it so you know where to look if it does not work first time. It's nearly impossible to type in a program without making at least one mistake!



SUBROUTINES

- Timer interrupt** I haven't worked out how to set the clock, yet so I have used Timer 1 to generate an interrupt and increment my own clock.
- Bonus interrupt** Timer 0 is the bonus counter, counting down to zero in 100 seconds. The bonus is printed in window 1 so as not to upset the print positions and colours of the other characters.
- Initialise** Draws the array, sets the colours and the high score. 4% is the "Y" key.
- Draw the maze** Reads the data statements and adds 82 to the Ascii code of each character before printing.

Start positions

Sets the start positions of the Grumpies and Smiley. Sets the bonus and dots left. Reduces delays. Sets score and delays to initial values. Prints score and three Smileys.

Set up

If it is not time then return. Sets the next time. Finds the new coordinates, and sets the flag (old) if they are the same as one of the Grumpies. Prints at the new position, erases the old one. Increments score if a button has been picked up.

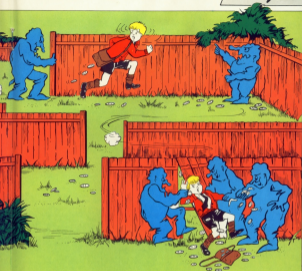
Move man

Move ghosts

Calls ghost to move the Grumpies if it is time.

Ghost

Looks where Smiley is and moves towards him if a wall is not in the way. Checks if he has been caught. Replaces button if necessary.



- Caught** Makes a rude noise and flashes several characters. Decreases lives.
- Instructions** The large title is drawn by printing it at the bottom in black, looking at the points and plotting it at the top of the screen. Prints the instructions, sets either keyboard (default setting) or joystick mode.
- Next screen** Prints screen completed in transparent mode. Adds the bonus to the score and increments the screen.
- Game over** Clears the screen by drawing black lines towards the centre. Prints scores. Asks if you want to play again.

VARIABLES

- time** Timer.
- bonus** Bonus.
- mass\$(00)** The mass.

ghost%(2,1)
ptime%(2)

galexy%(2)
%,%
h%
h%
ok

mass%,mass%
x%,y%

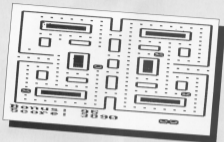
score%
dots%
lives
screen
a%,b%,c%,d%
waiters%
msdelay%

Grumpies' positions.
Time when the Grumpies next have to move.

Delay for the Grumpies.
Used in loops.
High score.
General variable.

A flag to show whether caught.
Smiley's coordinates.
Temporary coordinates of Grumpies or Smiley.
Score.

How many dots left.
Number of lives left.
Number of screen.
Values used by (NKEY)
The time Smiley can next move.
The delay for Smiley.



```

1298 :
1299 REM ** Caught **
1300 FOR i=0 TO 120 STEP -1
1301 FOR INT(ABS(i)/2) LOCATE x=1,y=0
1302 PRINT CHR$(i)
1303 CALL GETIN: SOUND 120,INT(ABS(i)/8)
1304 GOTO 1,1
1305 NEXT
1306 lives=lives-1
1307 IF lives=0 THEN LOCATE 4,13:FOR
1308 #PRINT CHR$(22);CHR$(11);"GAME OVER";
1309 CHR$(22);CHR$(8)
1310 RETURN
1311 :
1312 REM ** Instructions **
1313 GOSUB 1,18,1,188,18,-1,28
1314 BORDER B:INK 8,B:INK 1,B:INK 2,4
1315 INK 1,3,4
1316 PAPER B:CLS:FOR i LOCATE 1,25:FOR
1317 #T "hello = The Gruppies";
1318 SOUND 1,48,108,4,1: SOUND 2,88,1
1319 2,1: SOUND 4,98,118,8,1
1320 FOR v=0 TO 320 STEP 2
1321 FOR w=0 TO 34 STEP 2
1322 IF TEST(v,2,v) THEN PLOT v=258,
1323 280,w=250:PLUT v=128,280,w=242,2
1324 NEXT
1325 NEXT
1326 LOCATE 1,25:PRINT SPACE(125);
1327 GOSUB 1,1,1:FOR i LOCATE 1,1
1328 PRINT "Guide:hello around the w
1329 are collecting"PRINTPRINT "the color
1330 are buttons, but watch out for"
1331 PRINTPRINT "the gruppies who tr

```

```

v to catch you..."
1332 FOR #PRINTPRINTPRINT
1333 PRINT "Press r"
1334 FOR #PRINTPRINT " k for keypad
1335 and."
1336 PRINTPRINT " j for joystick."
1337 FOR #LOCATE 8,25:PRINT "% Press
1338 space to start ?"
1339 v=0:w=0:GOTO 1334:GOTO 1334
1340 WHILE INKEY()=0
1341 GOTO 1334:IF INKEY()=1 THEN LOCATE 1,17
1342 #PRINT CHR$(24);PRINTPRINT " *FOR
1343 LOCATE 1,25:PRINT "k...up 1...down
1344 k...left 2...right"FOR i=0:FOR
1345 #LOCATE 25:GOTO 1334
1346 IF INKEY()=0 THEN LOCATE 1,2
1347 #PRINT SPACE(48);LOCATE 1,17:PRINT
1348 " *PRINTPRINT CHR$(24);v=1:w=0:GOTO
1349 v=0:w=0:GOTO 1334
1350 NEXT
1351 PAPER 8
1352 RETURN
1353 :
1354 REM Next screen
1355 FOR #LOCATE 2,13:PRINT CHR$(22)
1356 ;CHR$(11)"Screen"screen"complete"
1357 CHR$(22);CHR$(8)
1358 screen=screen/2:IF screen=0 THEN
1359 screen=1
1360 score=score+bonus:FOR i LOCATE
1361 0,7,25:PRINT score;
1362 NEXT
1363 :
1364 REM ** Game over **

```

```

1365 FOR i=0 TO 5000:NEXT
1366 FOR v=0 TO 320 STEP 4
1367 FOR w=0 TO 340:GOTO v,188,w
1368 GOTO 148-v,0,1000:GOTO 148-v,188,4
1369 NEXT
1370 IF score>0:1 THEN #LOCATE
1371 179:FOR #LOCATE 1,13:PRINT STR$(ABS(
1372 28,"v")
1373 FOR #PRINT "Best score"v:12
1374 28:PRINTPRINTFOR i:PRINT "New s
1375 core"score:1
1376 PRINTFOR #PRINT STR$(ABS(28,"v")
1377 *1
1378 FOR #LOCATE 1,13:PRINT "Another
1379 game...?"v=1
1380 WHILE v=1
1381 GOTO 1378:IF INKEY()=1 THEN v=0
1382 GOTO 1378:IF INKEY()=2 THEN v=0
1383 NEXT
1384 RETURN

```



Give your fingers a rest...
All the listings from this month's issue are available on cassette.
See our special offer on Page 55.