

Simple Simon

Many people will already be familiar with this classic electronic game. The idea is simple - the game has four coloured pads which light up in a random sequence. Your aim is to repeat this sequence by pressing each of the pads in turn. It sounds easy but the more you get right the longer and faster the sequence gets, until you can no longer keep up.

The program presented here is a desktop implementation of the game which runs in a window and will multi-task with other applications. The pads are displayed as segments of a circle in red, blue, green and yellow (see figure 1), and the mouse is clicked over a segment in order to simulate pressing the pad.

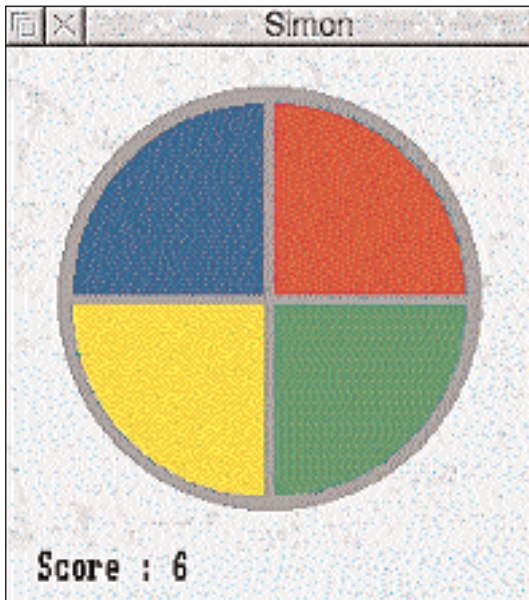


Figure 1. Simon on the desktop.

Creating the Application

First of all, create an application directory called !Simon. Inside this directory you should save a sprite file called !Sprites, containing a suitable 34 by 17 pixel mode 12 sprite called !simon. A suggested design is shown in figure 2. Next, use a text editor such as Edit

Colin Cassidy presents an addictive game which will tax the brain and keep the whole family entertained for hours



to create the following Obey file:

```
WimpSlot -min 32K -max 32K
```

```
Run <Obey$Dir>.!RunImage
```

This should be saved inside the application directory as !Run.

Finally, type in the program listed below and save it as !RunImage in the same directory.

Playing the Game

To play the game, simply click anywhere in the window, and you will immediately see one of the segments light up briefly. This is the first stage of the sequence, and you must replicate it by clicking over the same segment once the light is extinguished. The sequence now repeats with an additional segment added to it each time you replicate the complete sequence correctly. For example, if green lights up, you would click over the green segment. The next sequence will be green again followed by another colour, perhaps blue. You then click over green followed by blue. The third sequence will be green then blue then another colour, perhaps yellow or even green again. So it goes on until you either make a mistake, or take too much time to complete. When this happens, all four segments light up and your score is displayed briefly, followed by the current high score.

It sounds very simple, but once your score gets into double figures the game starts to become difficult. Happy clicking!

FEATURE

```

10 REM      >!RunImage
20 REM Program   Simon
30 REM Version   A1.00
40 REM Author    Colin Cassidy
50 REM RISC User October 1994
60 REM Program   Subject to Copyright
70 REM          Not Public Domain
80 :
90 DIM block% &200,col(3),mem(100)
100 quit%=FALSE:$block%="TASK"
110 SYS "Wimp_Initialise",200,!block%,
"Simon"
120 ON ERROR PROCerr:END
130 $block%=STRING$(100,CHR$0)
140 FOR n=0 TO 60 STEP 4
150   READ data%:block%!=data%
160 NEXT
170 $(block%+72)="Simon"
180 SYS "Wimp_CreateWindow",,block% TO
whan%
190 !block%=whan%
200 SYS "Wimp_GetWindowState",,block%
210 SYS "Wimp_OpenWindow",,block%
220 :
230 col( )=3:sc%=0:osc%=0:hi%=0
240 st%=0:st2%=0:st3%=0
250 PROCset_time:time%+=110
260 REPEAT
270   SYS "Wimp_Poll",,block% TO rsn%
280   CASE rsn% OF
290     WHEN 0:PROCact
300     WHEN 1:PROCredraw
310     WHEN 2:SYS "Wimp_OpenWindow",,bl
ock%
320     WHEN 3:quit%=TRUE
330     WHEN 6:PROCclick
340     WHEN 17,18
350     IF block%!16=0 quit%=TRUE
360   ENDCASE
370 UNTIL quit%
380 SYS "Wimp_CloseDown"
390 END
400 :
410 DEF PROCerr
420 !block%=ERR

430 $(block%+4)=REPORT$+" (internal er
ror code "+STR$ERL+") "+CHR$0
440 SYS "Wimp_ReportError",block%,1,"S
imon"
450 ENDPROC
460 :
470 DEF PROCredraw
480 SYS "Wimp_RedrawWindow",,block% TO
more%
490 x%=block%!4-block%!20
500 y%=block%!16-block%!24
510 cx%=x%+250:cy%=y%-240
520 WHILE more%
530   SYS "Wimp_SetColour",4
540   CIRCLE FILL cx%,cy%,200
550   FOR n=1 TO 4:PROCsect(n):NEXT
560   PROCscore
570   SYS "Wimp_GetRectangle",,block% T
O more%
580 ENDWHILE
590 ENDPROC
600 :
610 DEF PROCsect(n)
620   cx2%=cx%+COS((n+.5)*PI/2)*8
630   cy2%=cy%+SIN((n+.5)*PI/2)*8
640   SYS "Wimp_SetColour",col(n-1)
650   MOVE cx2%,cy2%
660   MOVE cx2%+COS(n*PI/2)*180+.1,cy2%+
SIN(n*PI/2)*180+.1
670   PLOT &B5,cx2%+COS((n+1)*PI/2)*180+
.1,cy2%+SIN((n+1)*PI/2)*180+.1
680   SYS "Wimp_SetColour",4
690 ENDPROC
700 :
710 DEF PROCscore
720   SYS "Wimp_SetColour",1
730   IF osc%<>sc% OR st%=0 RECTANGLE FI
LL x%,y%-530,500,50:osc%=sc%
740   IF st%=0 PROChiscore:ENDPROC
750   SYS "Wimp_SetColour",7
760   MOVE x%+30,y%-480
770   PRINT"Score : ";sc%
780 ENDPROC
790 :
800 DEF PROChiscore

```

```

810 SYS "Wimp_SetColour",7
820 MOVE x%+30,y%-480
830 PRINT "Hiscore : ",hi%
840 MOVE x%+255,y%-480
850 PRINT "Click to start"
860 osc%=-1
870 ENDPROC
880 :
890 DEF PROCclick
900 CASE st% OF
910 WHEN 0
920 st%=1:st2%=1
930 FOR n=0 TO 100:mem(n)=RND(4)-1:NE
XT
940 OTHERWISE
950 IF st2%<0 THEN
960 mx%=!block%:my%=block%!4
970 PROCwpos
980 IF mx%<cx% THEN
990 IF my%>cy% num%=0 ELSE num%=1
1000 ELSE
1010 IF my%>cy% num%=3 ELSE num%=2
1020 ENDIF
1030 IF num%=mem(-st2%) THEN
1040 col(num%)=num%+8:PROCupdate
1050 SOUND 1,-4,num%*30+80,5
1060 REPEAT:MOUSE tx,ty,tb
1070 UNTIL tb=0
1080 col()=3:PROCupdate
1090 st2%-=1:PROCset_time
1100 IF st2%<-st% sc%+=1:st%+=1:st2%
=1:time%+=100
1110 ELSE
1120 PROCend
1130 ENDIF
1140 ENDIF
1150 ENDCASE
1160 ENDPROC
1170 :
1180 DEF PROCact
1190 IF st%>0 AND st2%>0 THEN
1200 IF st3%=0 THEN
1210 IF FNread_time(10) THEN
1220 col(mem(st2%))=mem(st2%)+8
1230 SOUND 1,-4,mem(st2%)*30+80,5
1240 PROCupdate:PROCset_time
1250 st2%+=1:st3%=1
1260 ENDIF
1270 ELSE
1280 IF FNread_time(del%) THEN
1290 col()=3:PROCupdate:PROCset_time
1300 st3%=0:IF st2%>st% st2%=-1
1310 ENDIF
1320 ENDIF
1330 ENDIF
1340 IF st%>0 AND st2%=0 AND st3%=1 THE
N
1350 IF FNread_time(100) THEN
1360 st%=0:col()=3:PROCupdate
1370 ENDIF
1380 ENDIF
1390 IF st%>0 AND st2%<0 AND FNread_tim
e(300) PROCend
1400 ENDPROC
1410 :
1420 DEF PROCend
1430 FOR n=0 TO 3:col(n)=n+8:NEXT
1440 PROCupdate
1450 IF sc%>hi% hi%=sc%
1460 sc%=0:st2%=0:st3%=1
1470 PROCset_time
1480 ENDPROC
1490 :
1500 DEF PROCset_time
1510 SYS "OS_ReadMonotonicTime" TO time
%
1520 del%=100-st%*8:IF del%<15 del%=15
1530 ENDPROC
1540 :
1550 DEF FNread_time(d%)
1560 SYS "OS_ReadMonotonicTime" TO t%
1570 =(t%-time%)>d%
1580 :
1590 DEF PROCupdate
1600 PROCwpos
1610 block%!4=0:block%!8=-560
1620 block%!12=500:block%!16=0
1630 SYS "Wimp_UpdateWindow",,block% TO
more%
1640 WHILE more%
1650 FOR n=1 TO 4:PROCsect(n):NEXT
1660 PROCscore

```

