‘Pong’ Readme

The program ‘Pong’ runs under initial orders 2, and can be viewed on long tank 5. It should show a ball bouncing around the CRT screen. On fast computers it is best to run in real time.

‘Pong’ is comprised of three routines: a heading routine, an ‘Index-Write’ routine, and a master routine. The heading routine is not to be run, just loaded into the last 10 words of long tank 5 to show the word ‘Pong’ in lit spots. The ‘Index-Write’ routine copies the data stored in memory locations 4 and 5 to the address given in memory location 6. It does this by modifying its own 7th instruction to include the destination address. Watch locations 4-6 on long tank 0 to see the data being passed into this routine.

The ‘Master’ routine is a loop of 52 instructions, followed by 14 locations given over to its variables. The two most important variables are ‘ball’ and ‘addr’. All the bits of ‘ball’ are 0 except for a single 1 (in some position). ‘addr’ is a number between 160 and 180. The ball is drawn to the screen by writing ‘ball’ to the address in ‘addr’ (using the ‘Index-Write’ routine). The ranges of these variables allows a single spot of light to appear anywhere in the lower half of long tank 5.

The algorithm works by using two variable to indicate the direction of the ball ‘xdir’, and ‘ydir’. First a check is made that the ball is not about to go out of bounds, and if so the sign of one of these variables is reversed. Then the position of the ball is updated. The y component (‘addr’) is modified by adding ‘ydir’ to it (which is either +2 or –2, so represents either going up or down the screen). The x component (where the single ‘1’ bit in ‘ball’ is) is left or right shifted depending on the sign of ‘xdir’ (moving the ball left or right). The ball is written to the address ‘addr’, and then the algorithm repeats.

Actually, the program is a little more complicated than this, because it is necessary to blank out the old position of the ball. I used extra temporary variables ‘ball2’ and ‘addr2’ to hold the newly calculated position until all ‘0’s has been written to the old position. To see how the program behaves without this feature, remove the second G56D instruction (the 46th of the master routine).

NB The first instruction is ‘XF’ which means ‘do nothing’. This can be replaced with ‘ZF’ to stop the program before each pass around the loop (to slow it down, or to step through the program).

I have included on the next two pages a program listing with comments, in the standard notation used in WWG*.

Sam Garnham
January 2001

* Page 104 of *The Preparation of Programs for an Electronic Digital Computer* (Wilkes, Wheeler and Gill, 1951)

(a)Master Routine Listing

```
          G      K    Set θ parameter
          T   47 K    Set M parameter
          P   53 θ
          T   46 K    Set N parameter
          P   58 θ
          T      Z    Reset θ parameter
52-> θ 0  X      F
       1  T      F
       2  A      πN
       3  S    6 πN   IF ball is at right wall
       4  E    6 θ
       5  G    8 θ
 4->   6  S    8 πN   OR IF ball is at left wall
       7  G   11 θ
 5->   8  T      F
       9  S    4 πN   THEN xdir = -xdir
      10  T    4 πN   (bounce of a vertical wall)
 7->  11  T      F
      12  A      M    IF ball at bottom of long tank 5
      13  S    3 M
      14  E   16 θ
      15  G   18 θ    OR IF ball four rows from top of
14->  16  S    4 M    long tank 5
      17  G   21 θ
15->  18  T      F
      19  S    2 M    THEN ydir = -ydir
      20  T    2 M    (bounce of a horizontal wall)
17->  21  T      F
      22  A      M    Update y position
      23  A    2 M    (addr2 = addr+ydir)
      24  T    1 M
      25  A    4 πN   Update x position:
      26  E   32 θ    IF ball is going right (xdir=-1)
      27  T      D
      28  A      πN
      29  R      D    THEN shift ball right, store in ball2
      30  U    2 πN
      31  E   36 θ
      ----------
26->  32  T      D
      33  A      πN
      34  L      D    ELSE shift ball left, store in ball2
      35  U    2 πN
31->  36  T    4 D    Copy ball2 into address in addr2
      37  A    1 M    using the Index-Write routine
      38  T    6 F    (e.g. draw the ball in its new
      39  A   39 θ    position)
      40  G   56 D
      41  T      D
      42  T    4 D    Copy 0000... into address in addr
      43  A      M    using the Index-Write routine
      44  T    6 F    (e.g. blank out the old position)
      45  A   45 θ
      46  G   56 D
      ----------
      47  T      D
      48  A    2 πN   ball = ball2
      49  T      πN
      50  A    1 M    addr = addr2
      51  U      M
      52  E      θ    Jump back to start
      ----------
M      0  P   85 F    addr = 170
       1  P      F    addr2 (= 0)
       2  P    1 F    ydir = +2
       3  P   80 D    ymin + 1 = 161
       4  P    9 F    yrange - 1 = 18
πN     0  I      F    ball =
          P      F     00000000000000000010000000000000000
       2  P      F    ball2 (= 0)
          P      F
       4  P      D    xdir = +1
          P      F
       6  P    1 F    xmin + 1 =
          P      F     00000000000000000000000000000000010
       8  P      F    xrange - 1 =
          R      F     00100000000000000000000000000000000
```

(b) Table of routines

Routine	Location of First order	Number of storage locations occupied
Heading	182	10
Index-Write	56	10
Master	66	68

(c) Make-up of program tape

```
space P K

T 182 K


  Heading


space P Z

T 56 K


  Index-Write


space P Z


  Master


E Z P F
```

(d) Program tape

```
[Pong]
T182K
[Heading]
S799FS31FS1843FS49DZ823D*1585DK816FS817D
S799F*1567F
..PZ
T56K
[Index-Write]
GKA3FT8@A6FLDA9@T7@A4DTDEFTD
..PZ
[Master]
GKT47KP53@T46KP58@TZXFTFA#NS6#N
E6@G8@S8#NG11@TFS4#NT4#NTFAMS3M
E16@G18@S4MG21@TFS2MT2MTFAMA2M
T1MA4#NE32@TDA#NRDU2#NE36@TDA#N
LDU2#NT4DA1MT6FA39@G56DTDT4DAM
T6FA45@G56DTDA2#NT#NA1MUME@P85F
PFP1FP80DP9FIFPFPFPFPDPFP1FPFPFRF
EZPF
```

(e) Subroutines

On Screen Heading

```
S  799 F    01100000000111110001100011000111110
S   31 F
S 1843 F    01100000001100011001100111001100110
S   49 D
Z  823 D    01111110001100011001101011001101111
* 1585 D
K  816 F    01100011001100011001110011001100000
S  817 D
S  799 F    01111110000111110001100011000111110
* 1567 F
```

When you write this starting from address 182 and then view Long tank 5 on the EDSAC screen the words appear to spell PONG:

0 = . 1 = #

```
.######....#####...##...##...#####.
.##...##..##...##..###..##..##.....
.######...##...##..##.#.##..##.####
.##.......##...##..##..###..##..##.
.##........#####...##...##...#####.
```

Index-Write Routine

```
        G      K    Set θ parameter
θ     0 A    3 F    Plant Link
      1 T    8 θ
      2 A    6 F
      3 L      D    Address the 'write' instruction
      4 A    9 θ    with the location specified in 6F
      5 T    7 θ
      6 A    4 D    Copy data in 4D to that location
      7 (T     D)
      8 (E     F)   Link
        ----------
      9 T      D
```

This routine copies location 4-5 to the address specified by location 6.