

Let's Make a Retro Game

Episode 8 – Move and Shoot

In this episode, we are going to get some objects moving on the screen by:

- Testing the joystick direction and moving the player ship left or right
- Testing the joystick fire button and placing a bullet on screen
- Moving the current player bullet up the screen.

I have included the complete code for this section, so you can follow through the various steps without having to worry about typing in code.

The supplied code has two folders, Start, where we start in this episode and End the final code and a copy of the ROM file.

Moving the Player's Ship

Our first step is to add some code to look at the position of the joystick controller and move the players ship left or right in response.

Find the following section of code, that handles our main game logic loop and add a call to a new function called 'MOVE_PLAYER' as highlighted below:

```
    ; Main game logic loop
MLOOP:
    ; check that a base tick has occurred
    ; ensures consistent movement speed between 50 & 60Hz systems
    LD    A,(TickTimer)
    CALL  TEST_SIGNAL
    OR    A
    JR   Z,MLOOP2
    ; once per tick
    CALL MOVE_PLAYER

MLOOP2:
    LD A,(QtrSecTimer)
    CALL TEST_SIGNAL
    OR A
    JR Z,MLOOP

    JR MLOOP
```

This code will be called every time the screen has been drawn i.e. a virtual blank has occurred. We will not be drawing anything, just changing the stored location of items, so we don't need to be actually inside the interrupt routine.

Note: Trying to draw anything to the screen outside of the interrupt routine can cause on screen corruption.

Next, we need to add our new function that does the work of looking at the joystick and moving the ship sprite right or left by one pixel depending on the direction of the joystick. It also checks the current position of the ship so that our movement stops at either the left or right of the screen.

```
; Detect joystick direction and move the player accordingly
MOVE_PLAYER:
    CALL JOYDIR
    LD C,A
    BIT 1,C
    JR Z,NRIGHT
    ; move to the right
    LD A,(SPRTBL+1)
    CP 242
    JR NC,NLEFT
    INC A
    LD (SPRTBL+1),A
    LD (SPRTBL+5),A
    JR NLEFT
NRIGHT:
    BIT 3,C
    JR Z,NLEFT
    ; move to the left
    LD A,(SPRTBL+1)
    CP 0
    JR Z,NLEFT
    DEC A
    LD (SPRTBL+1),A
    LD (SPRTBL+5),A
NLEFT:
    RET
```

Firing a player bullet

Next, we need to look at the joystick trigger and if we haven't already fired a bullet, place a bullet sprite on the screen.

First, we add our function call just after our previous one as follows:

```
LD    A,(TickTimer)
CALL  TEST_SIGNAL
OR    A
JR   Z,MLOOP2
; once per tick
CALL  MOVE_PLAYER
CALL  FIRE_PLAYER_BULLET
```

And add the function code as follows:

```
; Check for player bullet firing
FIRE_PLAYER_BULLET:
; make sure there is not already a bullet
LD  A,(SPRTBL+8)
CP  209
RET NZ
; see if the fire button is pressed
CALL JOYTST
CP  0
RET Z
; fire bullet
; set Y based on player ship
LD  A,(SPRTBL)
SUB A,6
LD  (SPRTBL+8),A
; set X based on player ship
LD  A,(SPRTBL+1)
ADD A,6
LD  (SPRTBL+9),A
; set bullet sprite pattern
LD  A,24
LD  (SPRTBL+10),A
; set bullet colour
LD  A,11
LD  (SPRTBL+11),A
RET
```

This first checks to make sure we don't already have a bullet sprite on the screen, then if the joystick button is pressed a bullet sprite is placed on the screen relative to the current position of the player's ship.

Move the Player Bullet

Next, we need to add code to move the player bullet up the screen until it hits the top.

So we first add a call to a new function called 'MOVE_PLAYER_BULLET' just below our previous two calls as follows:

```
LD    A,(TickTimer)
CALL  TEST_SIGNAL
OR    A
JR    Z,MLOOP2
; once per tick
CALL  MOVE_PLAYER
CALL  FIRE_PLAYER_BULLET
CALL  MOVE_PLAYER_BULLET
```

Add our function code as follows:

```
; Move the players bullet
MOVE_PLAYER_BULLET:
; check that the bullet is visible
LD  A,(SPRTBL+8)
CP  209
RET  Z
; decrease bullets Y position
DEC  A
DEC  A
DEC  A
CP  4
JR  NC, MPB1
; bullet has reached the top of the screen, hide the bullet
LD  A,209
MPB1:
; save new position
LD  (SPRTBL+8),A
RET
```

This function first checks whether the player bullet sprite is on the screen, then if so decreases its Y position by 3 pixels. If it has reached the top of the screen it is removed, otherwise the sprite location is