Let’s Make A Retro Game

Episode 7 – Displaying our Background

In this episode we are going to start making our game “Mega Blast”, so to get things started we are going to go through:

- Initialising our graphics
- Displaying the graphics on both our title screen and then the main screen of the game

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.

Initialising our graphics

Using the latest version of my MSX Sprite & Tile Editor application, I have designed both the sprites and tiles we will need to get our application started.

From this tool, after opening the supplied file, you can save the file in assembler format.

The resultant file (Coleco-MegaBlast-Tileset.ASM) can be used directly in our project as an included file. That way if you want to change any of the graphics you can do so in the editor, resave the file, run the assembler and they will be updated in your game.

So the information that defines our sprites and tiles is now in our game, but they need to be transferred from our ROM into the video processors RAM (or vRAM as it's called).

To make this easier to read I have added a function (section of code) as follows:

```
PATSIZE:   EQU 71
; Load the character set, make all three sections the same
LOAD_CHR_SET:
    LD HL,0
SLOOP:
    LD DE,TILESET_1_PAT
    PUSH HL
    LD BC,PATSIZE*8
    CALL LDIRVM
    POP HL
    ; now load colour attributes
    PUSH HL
    LD BC,VRAM_COLOR
    ADD HL,BC
    LD DE,TILESET_1_COL
    LD BC,PATSIZE*8
    CALL LDIRVM
    POP HL
    LD BC,800h
    ADD HL,BC
    LD A,H
    CP 18h
    JR C,SLOOP
    RET
```

Note: I am keeping this really simple i.e. no compression, and we only use one set of tiles for each of the three regions on the screen.

This function basically grabs the tile data saved in the design tool and copies it to the VRAM so the video processor will use it to draw our background.

Displaying Our Title Screen

We then need to call this code to load our pattern (and colour) data into VRAM as follows:

So just after:

```
TITLESCREEN:
    ; display our title screen
    CALL DISABLE_NMI
    ; Clear the screen
    CALL CLEARPAT
```

We add the following code:

```
    ; Load the character set, make all three sections the same
    CALL LOAD_CHR_SET

    ; now setup the title screen layout
    LD HL,VRAM_NAME
    LD DE,TITLE_SCREEN_PAT
    LD BC,24*32
    CALL LDIRVM
```

We also add a section of data near the bottom of our code i.e. after:

```
; This is our routine called every VDP interrupt during the title screen
; - Do all VDP writes here to avoid corruption
OUTPUT_VDP_TITLE:
    RET
```

We add this section of tile layout data:

```
TITLE_SCREEN_PAT:
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,023,000,015,000,017,000,011,000,000,000,000
    DB 012,000,022,000,011,000,029,000,030,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
```

```
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 037,038,039,040,037,038,039,040,037,038,039,040,037,038,039,040
    DB 037,038,039,040,037,038,039,040,037,038,039,040,037,038,039,040
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    ; press fire to begin
    DB 000,000,000,000,000,000,026,028,015,029,029,000,016,019,028,015
    DB 000,030,025,000,012,015,017,019,024,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    ; EA Logo
    DB 061,061,061,061,061,061,061,061,061,061,061,057,058,059,060,061
    DB 062,063,064,065,066,061,061,061,061,061,061,061,061,061,061,061
```

And also include our sprite and tile data straight after:

```
;***************************************************************************
; Include external data files
;***************************************************************************

include "Coleco-MegaBlast-Tileset.ASM"
```

You should now be able to build and run the resultant MegaBlast.rom file using BlueMSX and see the title screen as follows:

Figure 1 - Our intial title screen as displayed on BlueMSX

Displaying Our Game Screen

Right next we can display the main screen of our game.

First let's add the tile layout data for the screen below the one we added previously as follows:

```
MAINLAYOUT:
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 037,038,039,040,037,038,039,040,037,038,039,040,037,038,039,040
    DB 037,038,039,040,037,038,039,040,037,038,039,040,037,038,039,040
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    ; planet surface
    DB 041,041,041,041,041,041,041,041,041,041,041,041,041,041,041,041
    DB 041,041,041,041,041,041,041,041,041,041,041,041,041,041,041,041
```

```
    ; score and life area
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    DB 000,000,000,001,001,001,001,001,001,001,000,000,000,000,000,000
    DB 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000
    ; EA Logo
    DB 061,061,061,061,061,061,061,061,061,061,061,057,058,059,060,061
    DB 062,063,064,065,066,061,061,061,061,061,061,061,061,061,061,061
```

And now just after:

```
NGAME:
    CALL DISABLE_NMI
    CALL INITRAM
```

We can add code to copy both our sprite and tile patterns to VRAM and then setup the layout for the main screen of the game as follows:

```
    ; Send the sprite definitions to the VDP
    LD HL,VRAM_SPRGEN
    LD DE,MAIN_SHIP
    LD BC,32*SPRITECOUNT
    CALL LDIRVM

    ; Load the character set, make all three sections the same
    CALL LOAD_CHR_SET

    ; now setup the title screen layout
    LD HL,VRAM_NAME
    LD DE,MAINLAYOUT
    LD BC,24*32
    CALL LDIRVM
```

After you build and run the rom in BlueMSX you should end up with a game screen (after you press fire on the title screen) as follows:

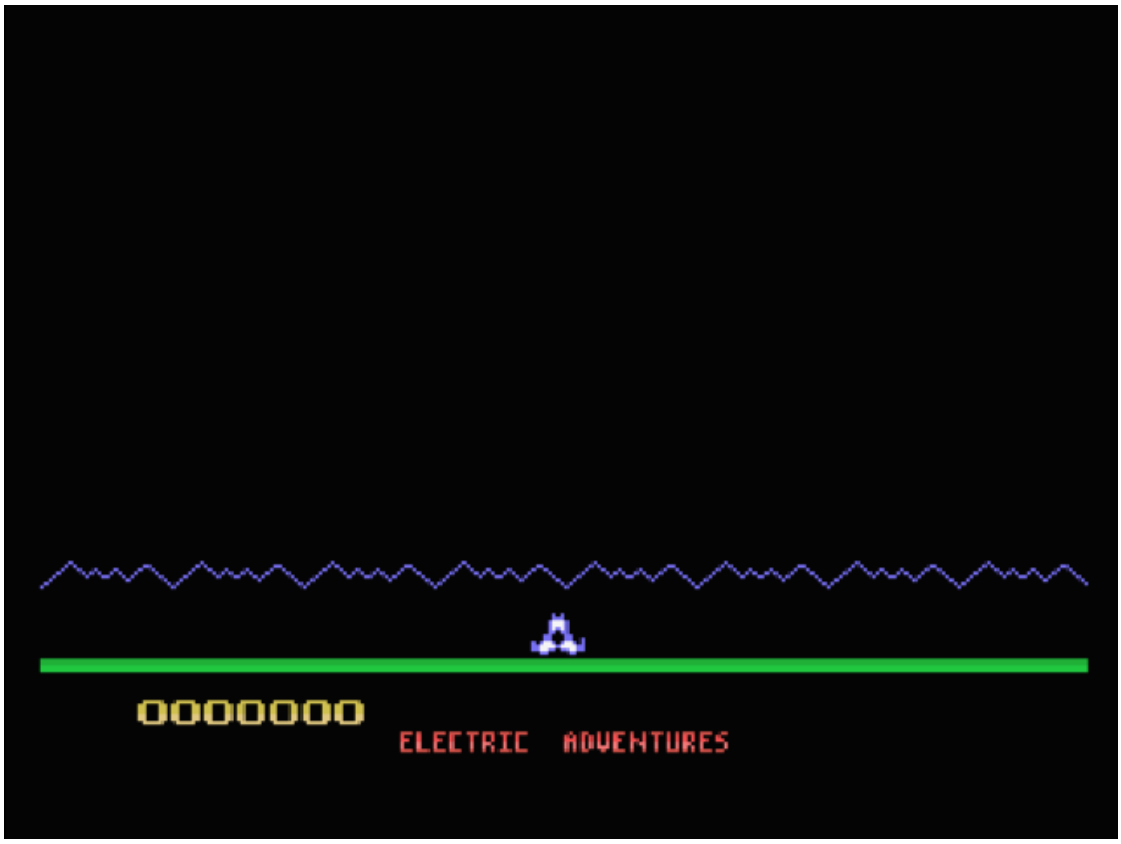

Figure 2 - Our main game screen