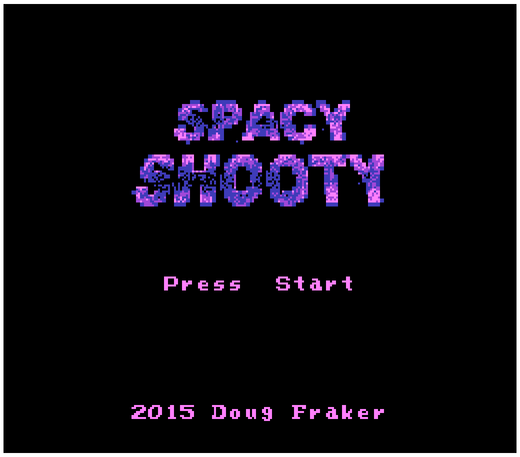
nesdoug

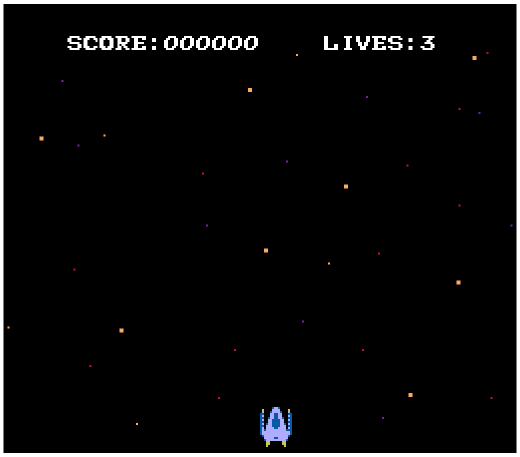
18. Game Coding

First thing is to get a skeleton of the game working. Title screen loads. Press 'start' moves to Game Mode. Press 'start' sets Pause Mode. 'Start' again goes back to Game Mode.

I opened Photoshop, and made a quick sketch of Title Sceen and Game Mode. Then I designed the Title by playing with fonts…Arial Black seems ok. I gave it some 'perspective', added a few filters, and resized it to 128 pixels wide. Converted to 4 color (color/mode/index/custom). Cut and paste it into YY-CHR.



The ship, I just sketched in YY-CHR. And some BG stars. Then I opened up NES Screen Tool, and randomly placed some stars, and saved nametable as RLE compressed C header. Similarly with the Title Screen, I arranged the text and Title, and saved that nametable as RLE in C.



Then I added some code to load the screens and switch between them with Start button presses. I also added 'Game Over Mode' and 'Victory Mode'. And, I'm testing them by tying them to the Select button (to be removed later). I'm also testing some Sound Fx that I made, tying them to some other button presses (Up/Down/B/A)…again, to be removed later. I just want to make sure they work.

For some reason I made all of my sound fx with the noise channel. Maybe this is a bad idea. Oh well.

At this point I got ahead of myself, and added the music. Usually, I wait till the very end to add music. I highly recommend waiting till the end, especially if it's a very long project, because you will get REALLY sick of hearing the same 1 minute song loop OVER and OVER and OVER.

I just happened to be in a music writing mood, and so I added it right away. Again, I used Famitracker and Famitone2. (See earlier post).

At this point, I added a HUD (the scoreboard), which is updated every frame with a new score. The scoreboard is actually written on the opposite nametable at startup. We will be using a sprite zero hit to switch nametables midframe.

Then, I put the ship in, and gave it some basic Left/Right movement physics. It moves a little slippery (you keep moving even if buttons aren't pressed), which I thought is appropriate for space. This can be changed later, if gameplay is odd.

Finally, I added the sprite zero hit and vertical scrolling. Switching nametables midframe is no trouble… just write it to $2000. If we were also changing H scroll, we would just need to write to $2005. But, I was doing the more complicated V scroll change. This is the rain dance you have to do to switch V scroll midframe…

```
Vert_scroll2 = ((Vert_scroll & 0xF8) << 2);
Sprite_Zero(); //wait for sprite zero hit
PPU_ADDRESS = 0;
SCROLL = Vert_scroll;
SCROLL = 0;
PPU_ADDRESS = Vert_scroll2;
```

I ran into some problems here, and had to do more debugging. It would work fine for a few seconds, then glitch, then it would screw up the sprite zero hit. I had to investigate. In FCEUX, debugger, I set breakpoints to writes to $2000, 2005, 2006, just to make sure.

They were writing the correct numbers. But, then I noticed that the first write to set the top of the screen (which should have happened during V-blank) was happening at Scanline 40! Oh, the music that I added was doing its update in the NMI routine…before anything else was happening. And, it was taking long enough (occasionally) to not get to all the other stuff that needs to happen during V-blank, until well into rendering the screen. (This was code from another game that I cut/pasted in…poorly). And, it was missing the Sprite Zero hit. It didn't actually crash the game, it was just setting it up WAY TOO late.

I moved the music update to go last on my list of things to do each frame, and it was working perfectly now.

After all that, I'm going to take a break.

Here's how the different modes work…

```
void main (void){


  while (1) { // infinite loop


    while (GameMode == TITLE_MODE){
    // Title Mode code


    }


    while (GameMode == RUN_GAME_MODE){
    // Game Mode code


    }


    while (GameMode == PAUSE_MODE){
    // Pause Mode code


    }


    while (GameMode == GAME_OVER_MODE){
    // Game Over code


    }
```

```
        while (GameMode == VICTORY_MODE){
        // Victory Mode code




        }
      }





    }
```

See you next time. Here’s the code I have so far.

http://dl.dropboxusercontent.com/s/vcnifnoooflgilq/spacy.zip
(http://dl.dropboxusercontent.com/s/vcnifnoooflgilq/spacy.zip)

Update: I’ve rewritten much of this. Now all sprite objects are going to be defined by structs…

struct ENEMY {
unsigned char anime; // for drawing object to screen
unsigned char dir; // direction, for flipping 0 = L
unsigned char Y; // top side
unsigned char X; // left side
unsigned char delay; // wait to start moves
unsigned char type; // type of object
unsigned char move; // which move to do
unsigned char count; // how far along on the move are we
};

(I changed type a little later to bullet_type, to define what bullet pattern to use, only in spacy4.c code)

And, I’m going to have a series of ‘waves’, where the ship movement pattern and type will be set at the start of the wave.

December 2, 2015April 15, 2017 dougfraker

3 thoughts on “18. Game Coding”

1.
   setheal says:

May 21, 2017 at 12:25 pm Edit
Hi, I have a problem.. When I enter in RUN_GAME_MODE, the scoreboard is show but my background no.. I've looked in FCEUX, the nametable of my background is here but nothing in screen..

Reply

dougfraker says:
May 22, 2017 at 3:15 am Edit
I would have to see the ROM to know what's wrong. Could be several things.

Reply

1. **setheal** says:
   May 23, 2017 at 2:37 pm Edit
   I find the solution, it was the sprite 0, thanks for all 🙂