12. Basic Platformer

In this example, I will set up a basic platform jumper that scrolls left to right. Again, I will have a level metatile map mirrored in the RAM. Since we have 2 screens worth of background, we will need 2 pages of RAM ($200).

The first thing you need for a platformer is…gravity. Every frame, sprites affected by gravity need to fall (++Y), unless they are standing on a platform. I will be calculating if the bottom of the metasprite is aligned to the background metatiles, and if so, check if the metatile below is a platform (for both bottom left and bottom right).

Here’s the platform / gravity code…

```
// first check the bottom left corner of character
 // which nametable am I in?
 NametableB = Nametable;
 Scroll_Adjusted_X = (X1 + Horiz_scroll + 3); // left
 high_byte = Scroll_Adjusted_X >> 8;
 if (high_byte != 0){ // if H scroll + Sprite X > 255, then we should use
  ++NametableB;   // the other nametable's collision map
  NametableB &= 1; // keep it 0 or 1
 }
 // we want to find which metatile in the collision map this point is in...is it s
 collision_Index = (((char)Scroll_Adjusted_X>>4) + ((Y1+16) & 0xf0));
 collision = 0;
 Collision_Down(); // if on platform, ++collision

// now check the bottom right corner of character
...same as above, but (X1 + Horiz_scroll + 12);


// oh, I might as well show you what collision_Down() looks like...
void Collision_Down(void){
 if (NametableB == 0){ // first collision map
  temp = C_MAP[collision_Index];
  collision += PLATFORM[temp];
 }
 else { // second collision map
  temp = C_MAP2[collision_Index];
  collision += PLATFORM[temp];
 }
}
// platform array has either 0 (no collision) or 1 (collision)
// based on what type of block you are standing on

// then gravity
if(collision == 0){
   Y_speed += 2;
}
else {
   Y_speed = 0;
   Y1 &= 0xf0; // align to metatile
}
```

The next thing I'm going to work on is smooth movement and smooth jumping. We need to use a lot of variables. 2 for X position, 2 for Y (the high byte = the screen position). 2 for X speed, 2 for Y speed. 1 for x acceleration. You could use constant acceration, but I want deceleration to be a bit faster. And, a max speed constant or two….

Actually, I punted on this one, and I just made 1 variable for each of those things, and I'm using the high nibble (upper 4 bits) of Xspeed as the speed. I had planned to do it like described above, but this seemed easier in my head. This is what I ended up doing…

```
Horiz_scroll += (X_speed >> 4);
```

We could also add a portion of the screen (in the middle), where L and R don't move the scroll, but rather the X position of the metasprite. I didn't do that here, because I want to keep it very simple.

I may come back and revise this code, to have 2 bytes for Xspeed and 2 for Xposition, etc. It would produce smoother movement.

Here's the link…

http://dl.dropboxusercontent.com/s/silvjdalw8vx5ib/lesson10.zip (http://dl.dropboxusercontent.com/s/silvjdalw8vx5ib/lesson10.zip)

November 29, 2015April 13, 2017 dougfraker