



GAMEBOY DEMO DEVELOPMENT



Nombre Asignatura: Tecnología del Videojuego
Módulo Asociado:
Profesor: Gustavo Aranda
Curso: 2019/2020
Autor/es: Victor Joaquin Ruiperez Oya



Index

1.- Project Description	Page 04
2.- GameBoy Specifications	Page 05
3.- GBDK	Page 06
4.- Pokémon	Page 07
5.- Summary	Page 10
6.- Bibliography	Page 11



1.- Project Description

1.1 Goal

The objective of this project is to investigate the technology of GameBoy and GameBoy Color consoles and achieve a small demo of what was learned by replicating the well-known Pokémon game, in its first generation.



2.- GameBoy Specifications

Since this system is old, we find much more limited resources than we can find nowadays, here are the specifications of the GameBoy console and the GameBoy Color.

1.1 GameBoy

RAM	8 KiB
Frame Rate	Around 60 fps
Color Palette	2-bit (4 shades of "gray")
Resolution	160 x 144 px

1.1 GameBoy Color

RAM	32 KiB
Frame Rate	Around 60 fps
Color Palette	15-bit (32,768)
Resolution	160 x 144 px



3.- GBDK

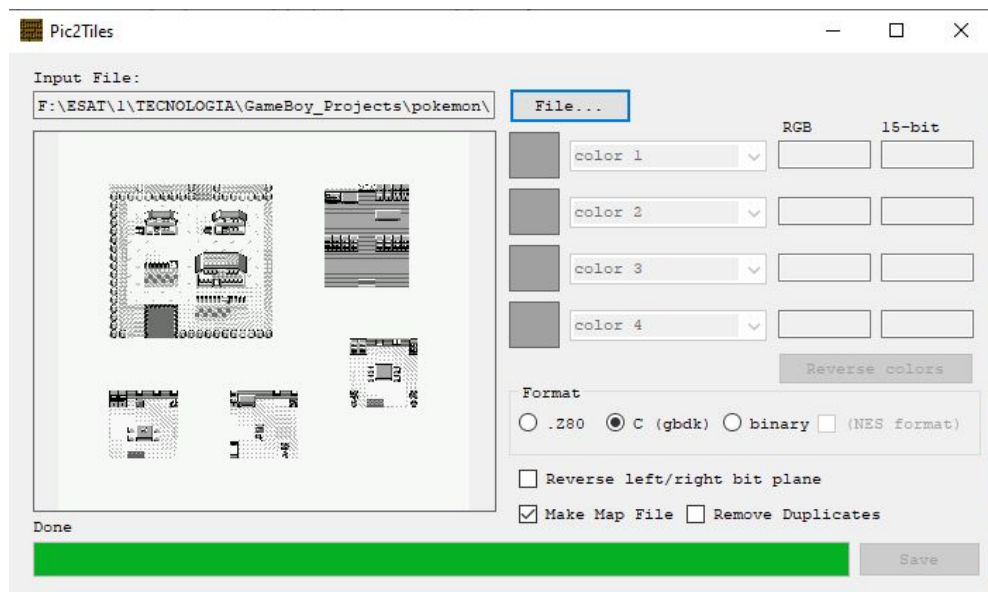
The GameBoy Developer's Kit (GBDK), is a set of tools that enable to develop programs for the Nintendo GameBoy system, either in C or in assembly. GBDK includes a set of libraries for the most common requirements and generates image files for use with a real GameBoy.

GBDK is freeware for non-commercial developments.

<http://gbdk.sourceforge.net/>

3.1.- Pic2Tiles

Pic2Tiles is a tool that allows you to convert 4-color images to a .c file with an array that represents this image.



3.2.- KiGB

KiGB is a GameBoy emulator that in addition to emulating games, can emulate the interconnection of two consoles.



4.- Pokémon

3.1 Character

The main character consists of eight 16 x 16 sprites, each one is a set of four 8 x 8 sprites, four for their idle state and another four to combine while moving.



Images of the four idle positions of the player in PNG format.

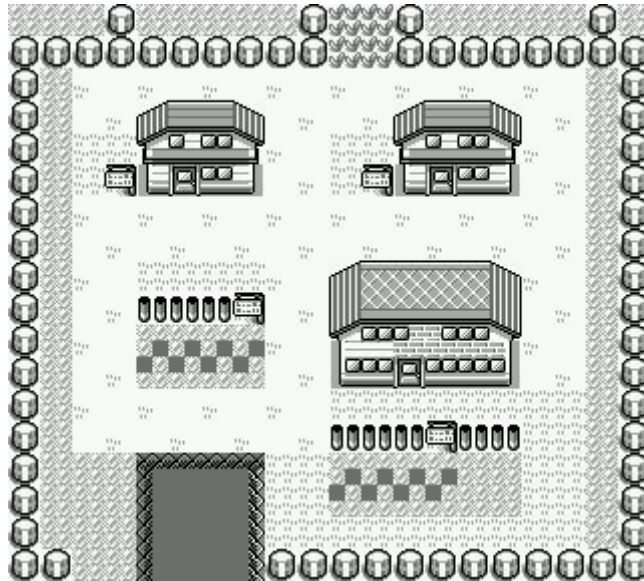
```
const unsigned char red_data[] = {  
    // Back Idle  
    0x07,0x07,0x08,0x0F,0x10,0x1F,0x10,0x1F,0x30,0x3F,0x38,0x3F,0x7F,0x5F,0x7F,0x4F,  
    0xE0,0xE0,0x10,0xF0,0x08,0xF8,0x08,0xF8,0x0C,0xFC,0x1C,0xFC,0xFE,0xFA,0xFE,0xF2,  
    0x3F,0x33,0x3C,0x3F,0x7B,0x5F,0x79,0x5E,0x3C,0x3F,0x17,0x1F,0x11,0x1F,0x0E,0x0E,  
    0xFC,0xCC,0x3C,0xFC,0xDE,0xFA,0x9E,0x7A,0x3C,0xFC,0xE8,0xF8,0x88,0xF8,0x70,0x70,  
    // Front Idle  
    0x07,0x07,0x08,0x0F,0x10,0x1F,0x10,0x1F,0x3B,0x3C,0x3F,0x37,0x7F,0x50,0x7F,0x42,  
    0xE0,0xE0,0x10,0xF0,0x08,0xF8,0x08,0xF8,0xDC,0x3C,0xFC,0xEC,0xFE,0x0A,0xFE,0x42,  
    0x3F,0x32,0x3E,0x39,0x7F,0x4F,0x7F,0x4F,0x39,0x3F,0x16,0x1F,0x11,0x1F,0x0E,0x0E,  
    0xFC,0x4C,0x7C,0x9C,0xFE,0xF2,0xFE,0xF2,0x9C,0xFC,0x68,0xF8,0x88,0xF8,0x70,0x70,  
    // Left Idle  
    0x07,0x07,0x08,0x0F,0x10,0x1F,0x38,0x37,0x7C,0x43,0x31,0x3F,0x1F,0x14,0x1F,0x14,  
    0xE0,0xE0,0x10,0xF0,0x08,0xF8,0x08,0xF8,0x1C,0xFC,0xFC,0xFC,0xFC,0xF8,0x98,  
    0x1F,0x10,0x0B,0x0C,0x07,0x07,0x03,0x03,0x03,0x03,0x04,0x07,0x04,0x07,0x03,0x03,  
    0xF0,0x10,0xE8,0x78,0xC8,0xF8,0xE8,0x38,0xE8,0x38,0xF0,0xF0,0x20,0xE0,0xC0,0xC0,  
    // Right Idle  
    0x07,0x07,0x08,0x0F,0x10,0x1F,0x10,0x1F,0x38,0x3F,0x3F,0x3F,0x3F,0x3F,0x1F,0x19,  
    0xE0,0xE0,0x10,0xF0,0x08,0xF8,0x1C,0xEC,0x3E,0xC2,0x8C,0xFC,0xF8,0x28,0xF8,0x28,  
    0x0F,0x08,0x17,0x1E,0x13,0x1F,0x17,0x1C,0x17,0x1C,0x0F,0x0F,0x04,0x07,0x03,0x03,  
    0xF8,0x08,0xD0,0x30,0xE0,0xE0,0xC0,0xC0,0xC0,0xC0,0x20,0xE0,0x20,0xE0,0xC0,0xC0,  
};
```

Images of the four idle positions of the player in array C format.

For the movement of the character sprite changes have been used with a delay to recreate the original effect.

3.2 Background

To convert the background, two files are created, the data file and the map file.



Palette town image in PNG format.

```
const unsigned char pallet_town_data[] = {
0xAB,0x00,0x45,0x00,0x8E,0x00,0x5D,0x00,0xBA,0x00,0x53,0x00,0x26,0x00,0x7D,0x00,
0x03,0x03,0x0D,0x0E,0x14,0x18,0x20,0x30,0x30,0x20,0x2C,0x30,0x2B,0x34,0x68,0x36,
0xE0,0xE0,0xD8,0x38,0x14,0x0C,0x02,0x06,0x06,0x02,0x1A,0x06,0xEA,0x16,0x0A,0x36,
0x01,0x02,0x00,0x05,0x06,0xC9,0x05,0xAA,0x20,0x9E,0x12,0x4C,0xD0,0x2C,0x64,0x18,
0xF8,0x26,0xBE,0x60,0xA8,0x70,0xA8,0x76,0x18,0x76,0x4C,0x3E,0x23,0x1F,0x1C,0x03,
0x0E,0x32,0x3E,0x02,0x3A,0x06,0x0A,0x36,0x0C,0x36,0x18,0x3C,0xE0,0xF8,0x00,0xE0,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x50,0x00,0x55,0x00,0x05,0x00,0x00,0x00,0x28,0x00,0x28,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x06,0x06,0x18,0x1E,0x60,0x76,
0x01,0x01,0x06,0x07,0x18,0x1B,0x60,0x7B,0x80,0xDB,0x00,0xDB,0x00,0xDB,0x00,0xDB,
0xFF,0xFF,0x00,0x00,0xFF,0x00,0xFF,0x00,0x00,0xFF,0xFF,0x00,0xFF,0x00,0xFF,0x00,
0x80,0x80,0x60,0xE0,0x18,0xD8,0x06,0xDE,0x01,0xDB,0x00,0xDB,0x00,0xDB,0x00,0xDB,
};

const unsigned char pallet_town_map[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x02,0x03,0x03,0x03,0x03,0,
0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x05,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x05,0x03,0x03,0x03,0x03,0,
0x01,0x02,0x01,0x02,0x01,0x02,0x01,0x02,0x01,0x02,0x01,0x02,0x01,0x02,0x01,0x02,0x01,0x02,0x03,0x03,0x03,0x03,0,
0x04,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x04,0x05,0x03,0x03,0x03,0x03,0,
0x01,0x02,0x00,0x00,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0,
0x04,0x05,0x00,0x00,0x07,0x06,0x06,0x06,0x07,0x06,0x06,0x06,0x07,0x06,0x06,0x07,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0,
0x01,0x02,0x00,0x00,0x06,0x06,0x06,0x06,0x06,0x08,0x09,0x0A,0x0A,0x0A,0x0B,0x0C,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0,
0x04,0x05,0x00,0x00,0x06,0x06,0x07,0x06,0x0D,0x0F,0x0F,0x0F,0x0F,0x0F,0x11,0x06,0x06,0x07,0x06,0x06,0x06,0x07,0x06,0x06,0,
0x01,0x02,0x00,0x00,0x07,0x07,0x07,0x07,0x12,0x13,0x14,0x15,0x14,0x14,0x16,0x17,0x06,0x06,0x06,0x06,0x07,0x07,0x07,0x07,0,
0x04,0x05,0x00,0x00,0x07,0x07,0x07,0x07,0x18,0x0F,0x0F,0x0F,0x0F,0x0F,0x19,0x07,0x06,0x06,0x06,0x07,0x07,0x07,0x07,0x07,0,
0x01,0x02,0x00,0x00,0x07,0x07,0x1A,0x1B,0x1C,0x15,0x1D,0x1E,0x14,0x14,0x15,0x1F,0x06,0x06,0x06,0x06,0x07,0x07,0x1A,0x1B,0x1B,0,
};
```

Palette town image in C format.

For the character movement, the background slides so that it seems to move and is always centered.



3.3 Collision Map

For collisions with the background elements, a collision matrix has been used, it is a conversion of the background to a board, each square is equivalent to a 16 x 16 position in the background image, in this way, maintaining an index, We can know where we are, where we can and cannot go and even what we have around.

```
unsigned char pallet_town_collisions[] = {
    0,0,0,1,0,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,
    1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,
    0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,1,
    0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,1,
    0,0,1,1,5,1,1,0,0,0,1,1,7,1,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,
    0,0,0,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,1,1,6,1,1,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,
    0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,
    0,0,0,2,2,2,2,0,0,0,0,0,0,0,0,1,1,1,1,1,
    0,0,0,2,2,2,2,0,0,0,0,0,0,0,0,1,1,1,1,1,
    1,1,1,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
};
```

Image of the palette town collisions map, the values set to 1 are walls, you cannot pass over them, the values set to 0 means that you can pass without problem, those set to 2 mean that there is water in that position and values greater than 5 indicate doors.



5.- Summary

It has been achieved that the character is centered, it has been possible to establish a map of functional collisions and to perform a fluid movement and as close to the original as possible.



I think that the most complicated thing to manage has been the conversion of the images to C and the implementation in C of it, since it does not use a very intuitive way to manage the colors.

On a personal level I think there should be a debug console since sometimes if you want to print a value, you cannot see something of the game that you may need to see.



6. BIBLIOGRAPHY

https://en.wikipedia.org/wiki/Game_Boy#Technical_specifications

https://en.wikipedia.org/wiki/Game_Boy_Color#Specifications

<http://gbdk.sourceforge.net/>

https://www.youtube.com/watch?v=HlsWR_jLdwo&list=PLeEj4c2zF7PaFv5MPYhNAkBGkx4iPGJo&index=1

<https://www.sprisers-resource.com/>