Treball Final de Grau PAC3

David Soler Bartomeu

UOC

Index

Universitat Oberta
de Catalunya

Introduction

In this PAC the executable file allows us to run the emulator. The emulator is fully functional except for the parts discussed in the first PAC document such as audio and savestates.

Repository: https://github.com/solbarda/draconic-gb
Explanation Video: https://www.youtube.com/watch?v=A9ITme5FrAI

Execution

The deliverable is located in the "Entregables" folder as discussed on PAC2. The emulator can be run by running the 'DraconicGB.exe' program found in the folder 'Executable'.

When the emulator is started it will open an SDL window and wait for the user input. In order to open a ROM the "File" menu can be used to search for a file in the user folder or a preset ROM can be executed selecting one of the "Open Preset File".

Once a ROM has been loaded the output of the screen can be seen on the window.
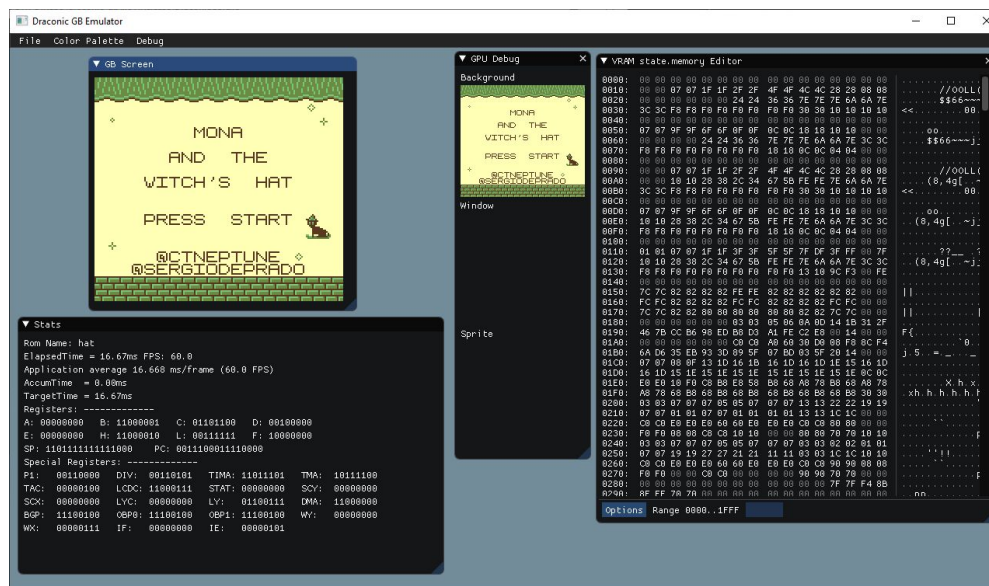
Panels

The emulator window contains different panels in order to display information.

The stats window display stats such as the ROM name, the average FPS (Frames Per Second) and the current state of the registers.

```
▼ Stats

Rom Name: hat
ElapsedTime = 16.67ms FPS: 60.0
Application average 16.665 ms/frame (60.0 FPS)
AccumTime  = 0.00ms
TargetTime = 16.67ms
Registers: -------------
A: 00000000   B: 11000001   C: 01101100   D: 00100000
E: 00000000   H: 11000010   L: 00111111   F: 10000000
SP: 1101111111111000    PC: 0011100011110000
Special Registers: -------------
P1:   00110000   DIV:  01001101   TIMA: 11001111   TMA:  10111100
TAC:  00000100   LCDC: 11000111   STAT: 00000011   SCY:  00000000
SCX:  00000000   LYC:  00000000   LY:   00101000   DMA:  11000000
BGP:  11100100   OBP0: 11100100   OBP1: 11100100   WY:   00000000
WX:   00000111   IF:   00000000   IE:   00000101
```

The RAM panel allows us to see what is currently stored on the Gameboy RAM.

```
▼ VRAM state.memory Editor                                          ×

0000:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0010:  00 00 07 07 1F 1F 2F 2F   4F 4F 4C 4C 28 28 08 08   ....../OOLL(
0020:  00 00 00 00 00 00 24 24   36 36 7E 7E 7E 6A 6A 7E   ......$$66~~~
0030:  3C 3C F8 F8 F0 F0 F0 F0   F0 F0 30 30 10 10 10 10   <<........00.
0040:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0050:  07 07 9F 9F 6F 6F 0F 0F   0C 0C 18 18 10 10 00 00   ....oo........
0060:  00 00 00 00 24 24 36 36   7E 7E 7E 6A 6A 7E 3C 3C   ....$$66~~~j.
0070:  F8 F8 F0 F0 F0 F0 F0 F0   18 18 0C 0C 04 04 00 00   ................
0080:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0090:  00 00 07 07 1F 1F 2F 2F   4F 4F 4C 4C 28 28 08 08   ....../OOLL(
00A0:  00 00 10 10 28 38 2C 34   67 5B FE FE 7E 6A 6A 7E   ....(8,4g[..~
00B0:  3C 3C F8 F8 F0 F0 F0 F0   F0 F0 30 30 10 10 10 10   <<........00.
00C0:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00D0:  07 07 9F 9F 6F 6F 0F 0F   0C 0C 18 18 10 10 00 00   ....oo........
00E0:  10 10 28 38 2C 34 67 5B   FE FE 7E 6A 6A 7E 3C 3C   ..(8,4g[..~j.
00F0:  F8 F8 F0 F0 F0 F0 F0 F0   18 18 0C 0C 04 04 00 00   ................
0100:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0110:  01 01 07 07 1F 1F 3F 3F   5F 5F 7F DF 3F FF 00 7F   ......??__ ;
0120:  10 10 28 38 2C 34 67 5B   FE FE 7E 6A 6A 7E 3C 3C   ..(8,4g[..~j.
0130:  F8 F8 F0 F0 F0 F0 F0 F0   F0 F0 13 10 9C F3 00 FE   ................
0140:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0150:  7C 7C 82 82 82 82 FE FE   82 82 82 82 82 82 00 00   ||...........
0160:  FC FC 82 82 82 82 FC FC   82 82 82 82 FC FC 00 00   ...........
0170:  7C 7C 82 82 80 80 80 80   80 80 82 82 7C 7C 00 00   ||.........|
0180:  00 00 00 00 00 00 03 03   05 06 0A 0D 14 1B 31 2F   ................
0190:  46 7B CC B6 98 ED B8 D3   A1 FE C2 E8 00 14 00 00   F{...........
01A0:  00 00 00 00 00 00 C0 C0   A0 60 30 D0 08 F8 8C F4   ...........`.0.
01B0:  6A D6 35 EB 93 3D 89 5F   07 BD 03 5F 20 14 00 00   j.5..=._...._
```

The GPU Window allows us to see the output of the VRAM before it will be displayed on the LCD screen. We can see the static background, the foreground windows and the sprites window.

Finally, the palette menu allows the user to change how the colors are displayed on the emulator.

Controls

The emulator controls are the following:

- **A:** Z key
- **B:** X key
- **Start:** A Key
- **Select:** S Key
- **D-Pad:** Arrow Keys

Development

The development process has been harder than initially expected. One of the major problems has been understanding the Gameboy hardware, such as the memory, cpu or gpu and how they communicate to one another using RAM and registers.

In order to understand it the resources explained on the PAC2 have been consulted a lot as well as other already existing open source emulators so understanding how emulators work can be easier. The current emulator architecture is based on a combination of some of the existing emulators as well as personal preference.

Implementing all the Opcodes has also been a very difficult task as well as testing the emulator as homebrew ROMs have had to be used. Initially the plan was to dump an existing Gameboy game and use that as reference, however, due to the problems with the pandemic it was impossible to access to the hardware needed to dump those games and, as such, alternative ROMs have been used as well as generic testing ROMs created by the emulator development community.

The emulator lacks sound implementation, it is correctly being emulated on the RAM, however, the implementation of sound output has not been created.
The emulator also lacks a feature to save and load states that was initially planned, some future work could be done in order to implement such a feature, as it is common for an emulator to have it.