



GameBoy Game in C

Embedded devices ITFactory

Mathieu Du Jardin

Inhoudstafel

1. Intro
2. Tools
 - 2.1 Visual Studio Code
 - 2.2 Game Web Developer Kit
 - 2.3 BGB-emulator
 - 2.4 GameBoy Tile Designer (GBTD)
 - 2.5 GameBoy Map Builder (GBMB)
 - 2.6 Paint.net
 - 2.7 GNU Image Manipulation Program en GameBoyPNGConverter
3. De Game
 - 3.1 Sprites
 - 3.2 Joypad
 - 3.3 Background
 - 3.4 Meta-sprites
 - 3.5 Sprite-collision detection
 - 3.6 Splashscreen
 - 3.7 Sound
 - 3.8 Extra toevoegingen
 - 3.8.1 Background collision
 - 3.8.2 Game over
4. Code
 - 4.1 main.c
 - 4.2 GameCharacter.c
 - 4.3 GameSprites.c
 - 4.4 GameSprites.h
 - 4.5 mario_data.c
 - 4.6 mario_map.c
 - 4.7 simplebackground.c
 - 4.8 simplebackground.h
 - 4.9 simplebackgroundmap.c
 - 4.10 simplebackgroundmap.h
5. Besluit
6. Bronnenlijst

1. Intro

Voor Embedded Devices Advanced heb ik ervoor gekozen om als project de simpele gameboy game te maken in C.

Voor ik aan dit project begon kende ik nog niet veel over C en daarom wou ik het juist maken. Één van mijn doelen van dit project is om beter C te leren.

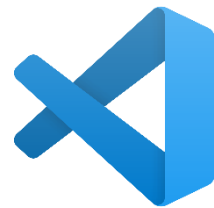
In dit verslag ga ik meer praten over mijn bevindingen, ontwikkelproces, moeilijkheden, tools en mijn game die ik heb gemaakt.

2. Tools

Ik ben dit project begonnen met veel research te doen. Ik ben gaan opzoeken wat er al allemaal bestaat, welke tools er allemaal zijn en wat ik allemaal nodig heb.

2.1 Visual Studio Code

Om mijn game te kunnen programmeren heb ik gekozen voor Visual Studio Code. Ik heb hier al redelijk wat met gewerkt en vond dit dan de meest logische keuze.



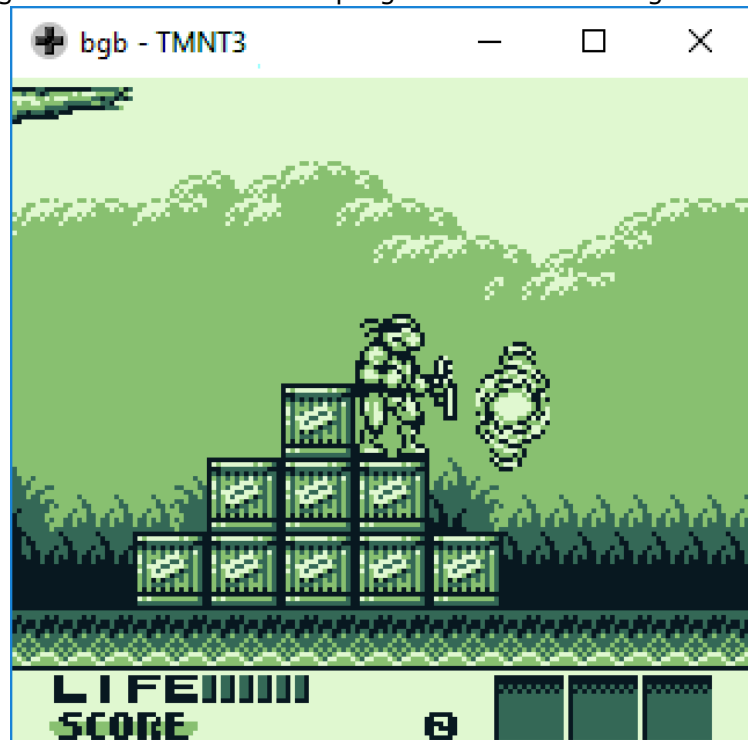
2.2 Game Web Developer Kit

Deze raden de meeste mensen aan. Je hebt het ook nodig om je game te ontwikkelen

2.3 BGB-emulator

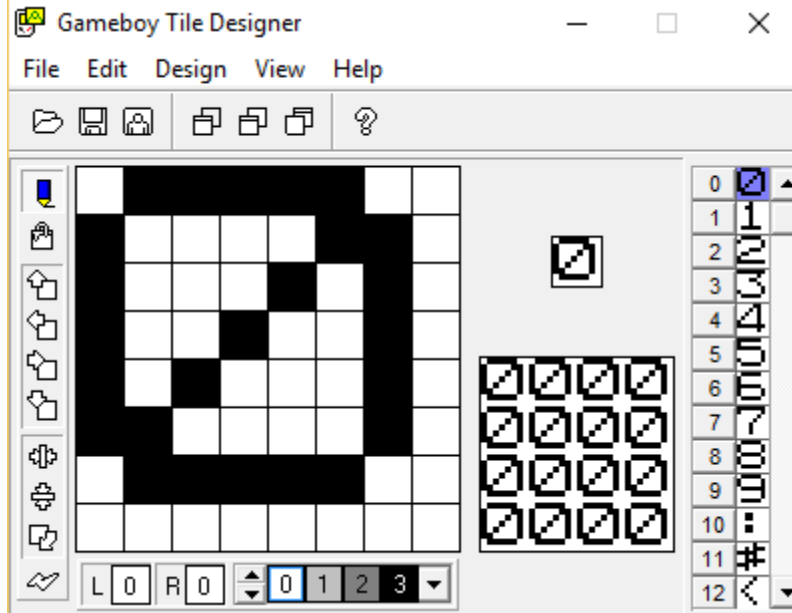
Om mijn game te kunnen simuleren heb ik gebruik gemaakt van de BGB-emulator. De emulator is een computerprogramma dat het mogelijk maakt om programma's die voor een andere computer ontwikkeld zijn te gebruiken.

In mijn geval ontwikkelde ik dus programma's voor een gameboy.



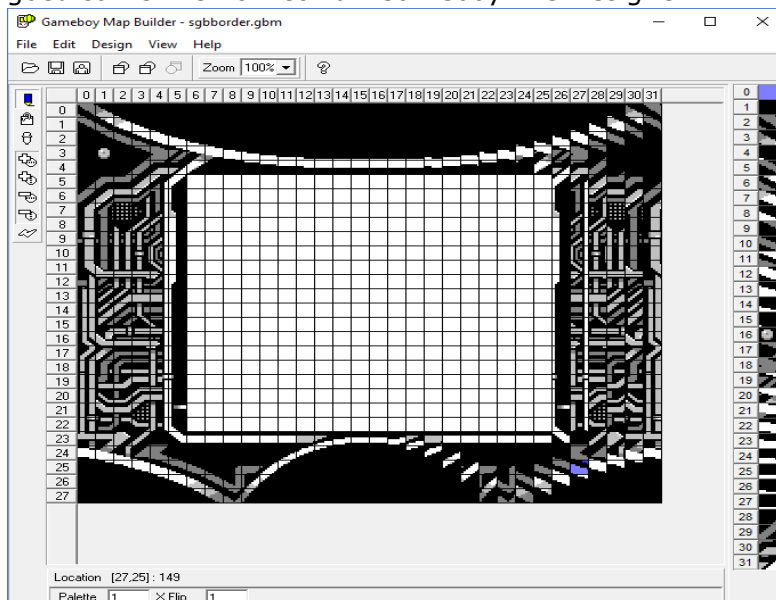
2.4 Gameboy Tile Designer (GBTD)

Om mijn "Tiles (tegels)" te designen heb ik de Gameboy Tile Designer van devrs gebruikt. Deze software zorgt ervoor dat ik in 8x8, 8x16, 16x16 en 32x32 mijn tegels kan maken.



2.5 Gameboy Map Builder (GBMB)

Om mijn maps te maken en bijvoorbeeld mijn achtergrond heb ik de Gameboy Map Builder van devrs gebruikt. Dit was de logische keuze aangezien deze goed samenwerkt met hun Gameboy Tile Designer.



2.6 Paint.net

Paint.net heb ik programma gebruikt om mijn Meta-Sprites in te tekenen.

2.7 GNU Image Manipulation Program en GameBoyPNGConverter

GIMP en de converter programmas waren nodig om mijn splashscreen te maken.

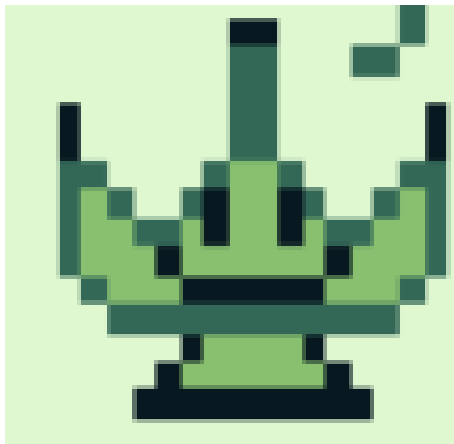
3. De Game

In de game die ik heb gemaakt ben je een "spaceship" dat de "monsters" moet ontwijken. De monsters komen van boven op je afgevlogen en jij moet door naar links en rechts te gaan deze ontwijken. Het is geen advanced game maar er kwam toch heel wat bij kijken om dit te kunnen maken.

Ik bespreek in chronologische volgorde hoe ik mijn game heb gemaakt.

3.1 Sprites

Mijn sprites heb ik gemaakt met de Gameboy Tile Designer. Een sprite bestaat uit tegels. Zo'n tegel bestaat uit 8x8 pixels. Je kan dus sprites maken die uit 8x8 pixels bestaan maar ook uit bijvoorbeeld 8x16 pixels. Je kan dus eigenlijk grote sprites (Meta-Sprites) maken door meerdere tegels (tiles) te combineren.



Als je je Sprite export dan krijg je een .c en een .h file. Ik heb niet in detail uitgezocht hoe en waarom je deze 2 files krijgt. Ik weet alleen dat je ze moet importeren in je "main" file en dan kan je je sprite gebruiken.

3.2 Joypad

Een gameboy heeft een joypad die je kan gebruiken om met te bewegen. De Gameboy Developer Kit heeft een functie "joypad()". Deze functie heb ik in mijn code gebruikt (zie later) om mijn sprite te laten bewegen.



3.3 Background

Om mijn achtergrond te maken heb ik de Gameboy Tile Designer en de Gameboy Map Builder gebruikt.

Als eerste heb ik in de Gameboy Tile Designer een paar tegels gemaakt met wat "sterren" die later de ruimte moeten voorstellen. Ik heb ook een "border" gemaakt die de randen aangeeft.

In de GBTD is het handig als je de eerste tegel (0) wit laat. Dit zorgt ervoor dat je altijd een witte achtergrond hebt. Voor sprites hoeft dit dus niet.

Als je je tegels hebt gemaakt kun je deze opslaan. In de Gameboy Map Builder kan je deze opgeslagen file dan openen waardoor je deze tegels kunt gebruiken om je map te tekenen.

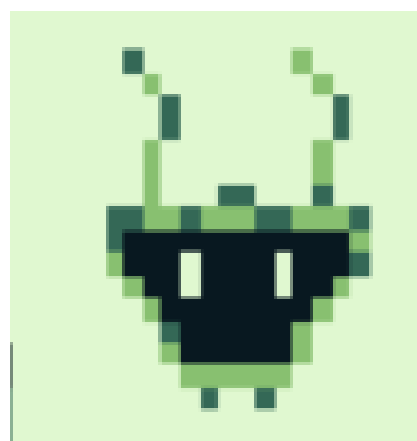
Als je dit allemaal hebt gedaan moet je zowel de GBTD file en de GBMB file exporten naar de map waar je je game aan het maken bent.



3.4 Meta-Sprites

Meta-Sprites zijn eigenlijk sprites die uit meerdere tegels bestaan. Een sprite bestaat uit 1 tegel (8x8 pixels) en vanaf dat je meerdere tegels gebruik spreken we over meta-sprites.

Ik heb meta-sprites gebruikt om mijn spaceship te maken en mijn monsters.



3.5 Sprite-Collision Detection

Nadat ik mijn spaceship, monsters en achtergrond had gemaakt was het tijd om ervoor te zorgen dat als de monsters mijn spaceship raken het spel stopt. Dit heb ik gedaan met collision detection.

Ik heb op het YouTube kanaal van GamingMonsters een goede video gevonden die uitlegt hoe dit werkt.

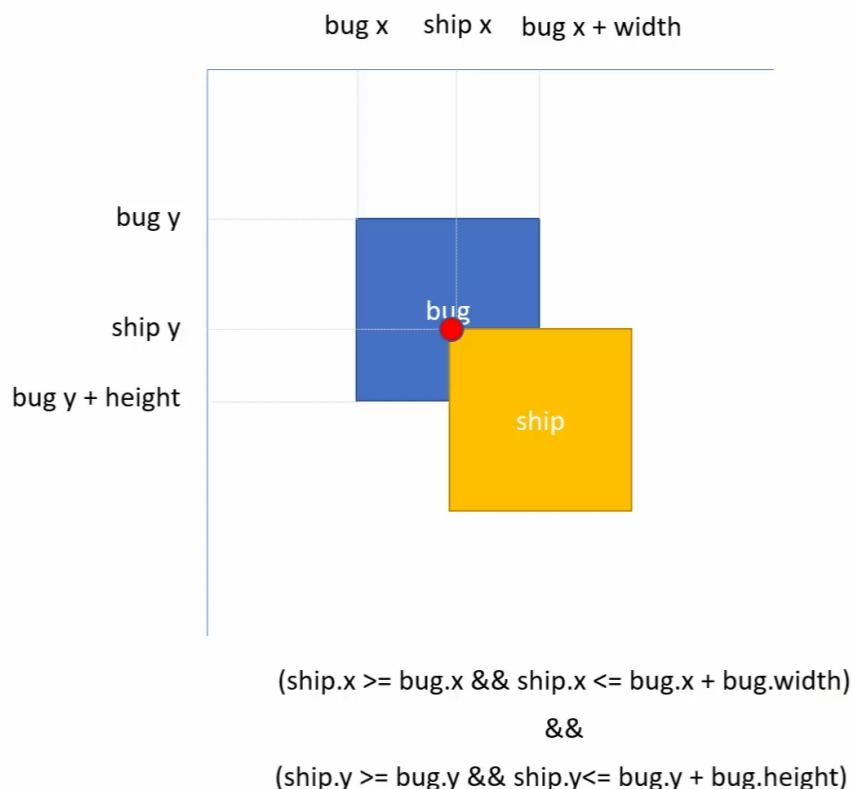
Stel dat we elke meta-sprite zien als een vierkant. Hieronder zie je dan het vierkant: bug (wat bij mij monsters zou zijn) en ship (spaceship).

De bug heeft een x en y coördinaat in de top linkse hoek. Het ship (die hieronder botst met de bug) heeft ook een x en y in de top linkse hoek. De bug heeft ook nog een x + width en een y + height.

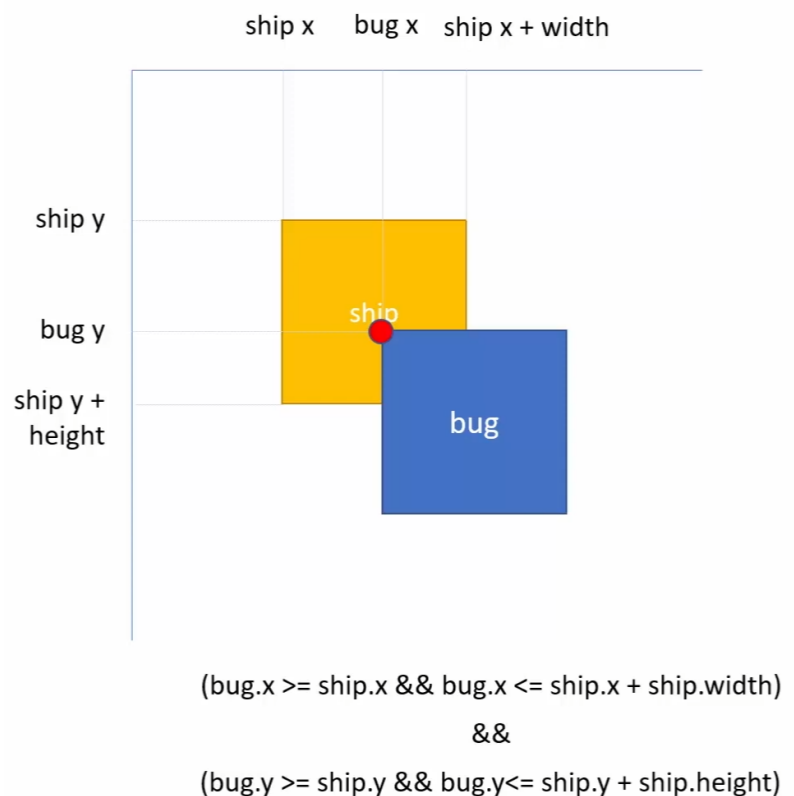
Het rode punt (x en y van het ship) is eigenlijk het punt dat wij willen controleren. Is het x en y coördinaat van het ship tussen het x en y coördinaat van bug + breedte en hoogte?

Je kan dit ook simpeler formuleren: Is het rode punt ergens in het blauwe vierkant?

De code die je onder de figuur ziet staan is de code om dit te controleren.



Dit was de eerst helft. We kijken of de top linkse hoek van ship in bug is maar als we alleen hiernaar kijken dan zal het niet altijd botsten. Wij moeten ook kijken of de top linkse hoek van bug in ship is. Als wij deze 2 controleren hebben we altijd een "collision".



Enkele voorbeelden in mijn game:



3.6 Splashscreen

Een splashscreen is het scherm dat getoond wordt tijdens het laden van het spel. Mij leek het een leuke toevoeging om vooraleer je aan de game begint een splashscreen te hebben. Ik heb gewoon gekozen voor een Mario mannetje omdat dit mij aan de gameboy doet denken. We kunnen er ook van uit gaan dat Mario in mijn spaceship zit.

Om dit te realiseren heb ik een Mario op het internet opgezocht en dan bewerkt met GIMP (GNU Image Manipulation Program). Ik heb de pixels verandert zodat het op het scherm past en heb er ook voor gezorgd dat de afbeelding maar uit 4 kleuren bestond.



3.7 Sound

Als laatste heb ik ook nog wat geluid toegevoegd aan mijn gameboy. Dit zal ik meer uitgebreid uitleggen in mijn code.

3.8 Extra toevoegingen

Ik heb nog wat features proberen toe te voegen maar deze heb ik niet werkende gekregen. Ik heb besloten om verder te gaan met andere extra's omdat ik na veel tijd nog altijd niet verder stond.

3.8.1 Background collision

Één van mijn problemen in mijn game is dat je buiten de map kan gaan. Als je te veel naar links of rechts gaat dan ga je buiten het zichtbare scherm.

Met background collision kan je dit oplossen. Daarvoor had ik ook de borders getekend. Spijtig genoeg heb ik dit niet werkende gekregen.

3.8.2 Game over

Ik wil als je doodgaat een "Game Over" op het scherm laten printen. Opzich is dit niet moeilijk maar omdat mijn achtergrond beweegt op de Y-as is de "Game Over" soms buiten en scherm en wordt deze niet altijd geprint.

4. Code

4.1 main.c

```
#include <gb/gb.h>
#include <stdio.h>
#include "simplebackground.c"
#include "simplebackgroundmap.c"
#include "GameCharacter.c"
#include "GameSprites.c"
#include "mario_data.c"
#include "mario_map.c"
```

Hier importer ik alle belangrijke files en libraries

```
GameCharacter spaceship;
GameCharacter monster;
UBYTE spritesize = 8;
```

De regels `GameCharacter spaceship;`, `GameCharacter monster;`, en `UBYTE spritesize = 8;` definiëren variabelen voor twee gamepersonages (ruimteschip en monster) en bepalen de grootte van de sprites in het spel (8 pixels).

```
void performantdelay(UINT8 numloops){
    UINT8 i;
    for(i = 0; i < numloops; i++){
        wait_vbl_done();
    }
}
```

De `performantdelay`-functie zorgt voor een vertraging door te wachten op het einde van VBLANK voor het opgegeven aantal iteraties, wat nuttig kan zijn om de uitvoersnelheid van de game te regelen.

```

UBYTE checkcollisions(GameCharacter* one, GameCharacter* two){
    return (one->x >= two->x && one->x <= two->x + two->width) && (one->y >=
two->y && one->y <= two->y + two->height) || (two->x >= one->x && two->x <=
one->x + one->width) && (two->y >= one->y && two->y <= one->y + one->height);
}

```

De functie checkcollisions controleert of er een botsing is tussen twee gamepersonages.

```

void movegamecharacter(GameCharacter* character, UINT8 x, UINT8 y){
    move_sprite(character->spritids[0], x, y);
    move_sprite(character->spritids[1], x + spritesize, y);
    move_sprite(character->spritids[2], x, y + spritesize);
    move_sprite(character->spritids[3], x + spritesize, y + spritesize);
}

```

De functie movegamecharacter wordt gebruikt om de sprites van een gamepersonage naar een nieuwe positie te verplaatsen.

```

void setupship(){
    spaceship.x = 80;
    spaceship.y = 130;
    spaceship.width = 16;
    spaceship.height = 16;

    // load sprites for spaceship
    set_sprite_tile(0, 0);
    spaceship.spritids[0] = 0;
    set_sprite_tile(1, 1);
    spaceship.spritids[1] = 1;
    set_sprite_tile(2, 2);
    spaceship.spritids[2] = 2;
    set_sprite_tile(3, 3);
    spaceship.spritids[3] = 3;

    movegamecharacter(&spaceship, spaceship.x, spaceship.y);
}

```

De functie setupship wordt gebruikt om de eigenschappen van het ruimteschip-personage in te stellen en de bijbehorende sprites te laden.

```

void setupmonster(){
    monster.x = 30;
    monster.y = 0;
    monster.width = 16;
    monster.height = 16;

    // load sprites for monster
    set_sprite_tile(4, 4);
    monster.spritids[0] = 4;
    set_sprite_tile(5, 5);
    monster.spritids[1] = 5;
    set_sprite_tile(6, 6);
    monster.spritids[2] = 6;
    set_sprite_tile(7, 7);
    monster.spritids[3] = 7;

    movegamecharacter(&monster, monster.x, monster.y);
}

```

De functie setupmonster wordt gebruikt om de eigenschappen van het monster-personage in te stellen en de bijbehorende sprites te laden.

```

void main(){
    // these registers must be in this specific order!
    NR52_REG = 0x80; // is 1000 0000 in binary and turns on sound
    NR50_REG = 0x77; // sets the volume for both left and right channel just
set to max 0x77
    NR51_REG = 0xFF; // is 1111 1111 in binary, select which channels we want
to use in this case all of them. One bit for the L one bit for the R of all
four channels

    set_bkg_data(0, 114, mario_data);
    set_bkg_tiles(0, 0, 20, 18, mario_map);
    SHOW_BKG;
    DISPLAY_ON;
    waitpad(J_START);

    set_bkg_data(0, 7, backgroundtiles);
    set_bkg_tiles(0, 0, 20, 36, backgroundmap);
    set_sprite_data(0, 8, GameSprites);
    setupship();
    setupmonster();

    SHOW_BKG;
    DISPLAY_ON;
    SHOW_SPRITES;
    DISPLAY_ON;
}

```

Hier initialiseer ik het geluid, zet de flashscreen klaar, de background en de sprites worden hier klaargezet.

```
while(!checkcollisions(&spaceship, &monster)){

    UBYTE joypad_state = joypad();

    if(joypad_state) {

        NR10_REG = 0x16;

        NR11_REG = 0xE2;

        NR12_REG = 0x82;

        NR13_REG = 0x00;

        NR14_REG = 0x87;

    }
    scroll_bkg(0,1);

    if(joypad() & J_LEFT){
        spaceship.x -= 2;
        movegamecharacter(&spaceship, spaceship.x, spaceship.y);
    }
    if(joypad() & J_RIGHT){
        spaceship.x += 2;
        movegamecharacter(&spaceship, spaceship.x, spaceship.y);
    }

    monster.y += 5;

    if(monster.y >= 144){
        monster.y=0;
        monster.x = spaceship.x;
    }

    movegamecharacter(&monster,monster.x,monster.y);

    performantdelay(2);
}

}
```

In de loop wordt er gekeken of er al is gebotst. Als er een joypad wordt gebruikt dan komt er een geluid. Ik heb ook gezegd dat je naar links en rechts kunt bewegen. De scroll_bkg is voor de achtergrond de laten "scrollen" naar beneden waardoor het lijkt alsof die beweegt.

4.2 GameCharacter.c

```
#include <gb/gb.h>

//generic character structure: id, position, graphics
typedef struct GameCharacter {
    UBYTE spritids[4]; // all characters use 4 sprites
    UINT8 x;
    UINT8 y;
    UINT8 width;
    UINT8 height;
} GameCharacter;
```

Ik heb structs gebruikt. Hier wordt de game character structuur bepaald.

4.3 GameSprites.c

```
/*

GAMESPRITES.C

Tile Source File.

Info:
Form          : All tiles as one unit.
Format        : Gameboy 4 color.
Compression   : None.
Counter       : None.
Tile size     : 8 x 8
Tiles         : 0 to 7

Palette colors : None.
SGB Palette    : None.
CGB Palette    : None.

Convert to metatiles : No.

This file was generated by GBTD v2.2

*/

/* Start of tile array. */
unsigned char GameSprites[] =
{
    0x01,0x01,0x00,0x01,0x00,0x01,0x80,0x81,
```

```

0x80,0x81,0x01,0xC2,0x43,0xA6,0x67,0x9A,
0x80,0x80,0x00,0x80,0x00,0x80,0x01,0x81,
0x01,0x81,0x80,0x43,0xC2,0x65,0xE6,0x59,
0x7F,0x88,0x3F,0x47,0x00,0x3F,0x07,0x04,
0x0F,0x08,0x1F,0x1F,0x00,0x00,0x00,0x00,
0xFE,0x11,0xFC,0xE2,0x00,0xFC,0xE0,0x20,
0xF0,0x10,0xF8,0xF8,0x00,0x00,0x00,0x00,
0x00,0x20,0x10,0x00,0x00,0x08,0x00,0x08,
0x10,0x00,0x10,0x00,0x10,0x01,0x1B,0x64,
0x10,0x00,0x08,0x00,0x00,0x04,0x00,0x04,
0x08,0x00,0x08,0x00,0x00,0x88,0x9C,0x62,
0x3F,0x7F,0x7B,0x3B,0x3B,0x1B,0x1F,0x0F,
0x07,0x0F,0x0F,0x07,0x07,0x00,0x00,0x02,
0xFE,0xFC,0xDC,0xDE,0xDC,0xD8,0xF8,0xF0,
0xF0,0xE0,0xF0,0xE0,0xE0,0x00,0x00,0x40
};

/* End of GAMESPRITES.C */

```

4.4 GameSprites.h

```

/*

GAMESPRITES.H

Include File.

Info:
Form           : All tiles as one unit.
Format         : Gameboy 4 color.
Compression    : None.
Counter        : None.
Tile size      : 8 x 8
Tiles          : 0 to 7

Palette colors : None.
SGB Palette    : None.
CGB Palette    : None.

Convert to metatiles : No.

This file was generated by GBTD v2.2

*/

/* Bank of tiles. */
#define GameSpritesBank 0

```

```

/* Start of tile array. */
extern unsigned char GameSprites[];

/* End of GAMESPRITES.H */

```

4.5 mario_data.c

```

/*
Generated By GameBoyPngConverter

Tiles data

Number of Tiles   : 114
*/

const unsigned char mario_data[] = {
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x00,0x1F,0x00,0x7B,0x00,0xC2,
    0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0xFF,0x00,0xFF,0x00,0xFF,
    0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x00,0xC0,0x00,0xF0,
    0x00,
    0x01,0x00,0x01,0x00,0x02,0x00,0x06,0x00,0x06,0x00,0x0F,0x00,0x0F,0x00,0x3F,
    0x00,
    0xA6,0x00,0x7E,0x00,0x7E,0x00,0xD7,0x00,0x83,0x00,0x83,0x00,0x82,0x00,0xF1,
    0x00,
    0x7F,0x00,0x7F,0x00,0x7F,0x00,0x7F,0x00,0x7F,0x00,0xFF,0x00,0xFF,0x00,0xFF,
    0x00,
    0xE8,0x10,0xC4,0x38,0xC2,0x3C,0xC1,0x3E,0xC0,0x3F,0xE0,0x1F,0xE0,0x1F,0xE0,
    0x1F,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x00,0x40,0x80,0x20,0xC0,0x00,
    0xE0,
    0x01,0x00,0x02,0x01,0x04,0x03,0x00,0x07,0x08,0x07,0x08,0x07,0x00,0x07,0x04,
    0x03,
    0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x02,0xFF,0x0E,0xFF,0x1E,
    0xFF,
    0x0F,0xF0,0x01,0xFE,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0xFE,0x01,0xFF,
    0x01,
    0xFF,0x00,0xE7,0x18,0x20,0xDF,0x00,0xFF,0x00,0xFF,0x78,0xFF,0xFC,0xFF,0xFE,
    0xFF,
    0xE0,0x1F,0x80,0x7F,0x80,0x7F,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,
    0xFF,
    0x10,0xE0,0x08,0xF0,0x00,0xF8,0x04,0xF8,0x00,0xFC,0x02,0xFC,0x00,0xFE,0x01,
    0xFE,
    0x00,0x03,0x02,0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,

```


0x1E,0xFF,0x18,0xFF,0x1F,0xF0,0x1E,0x70,0x10,0x00,0x02,0x01,0x02,0x01,0x01
,0x00,
0xFF,0x03,0xFD,0x02,0x07,0x00,0x07,0x00,0x04,0x00,0x04,0x00,0x01,0x00,0x00
,0x01,
0x8E,0xFF,0xFB,0x06,0xFD,0x02,0xFF,0x00,0x03,0x00,0x91,0x60,0x00,0xF0,0x10
,0xA0,
0x00,0xFF,0x00,0xFF,0xC0,0x3F,0xF0,0x0F,0xFA,0x07,0xFF,0x03,0x7F,0x03,0x3B
,0x07,
0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x80,0xFF,0xC0
,0xFF,
0x00,0x00,0x00,0x00,0x80,0x00,0x20,0xC0,0x10,0xE0,0x08,0xF0,0x00,0xF8,0x00
,0xF8,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01
,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x03,0x00,0x07,0x00,0x1F,0x00,0xFF
,0x00,
0x0A,0x01,0xCA,0x01,0xEA,0x01,0xF8,0x01,0xFF,0x00,0xFC,0x00,0xFF,0x00,0xEF
,0x1F,
0x70,0xE0,0x10,0xE0,0x30,0xC0,0xE0,0x00,0xCC,0x00,0x6E,0x1C,0x7F,0xFC,0xFF
,0xFC,
0x3F,0x07,0x31,0x0F,0x31,0x0F,0x23,0x1F,0x27,0x1F,0x3E,0x0F,0x39,0x07,0xF9
,0x07,
0xE0,0xFF,0xF0,0xFF,0xF9,0xE0,0xC7,0xE0,0xFF,0xC0,0xBF,0xC0,0xFF,0x80,0x7C
,0x80,
0x00,0xF8,0x00,0xF8,0x80,0x78,0xC8,0x30,0xE0,0x10,0xE0,0x00,0xE0,0x00,0x60
,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x01,0x00,0x01
,0x00,
0x01,0x00,0x01,0x00,0x00,0x01,0x01,0x00,0x20,0x00,0x20,0x00,0xB0,0x00,0xF0
,0x00,
0xFF,0x00,0xFF,0x00,0xFE,0x01,0xC3,0xFF,0x7F,0xFF,0xF9,0x1F,0x7F,0x00,0x4F
,0x00,
0xDF,0x3F,0xBF,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFB,0xC4,0xFF,0x00,0xFF
,0x00,
0xFB,0xFC,0xF7,0xF8,0xDF,0xE0,0xFF,0x80,0xFF,0x00,0xEF,0x00,0xFF,0x00,0xFF
,0x00,
0xFF,0x07,0xFA,0x07,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF
,0x01,
0xFE,0x00,0xFA,0x00,0xF8,0x00,0xF8,0x00,0xF1,0x00,0xF7,0x00,0xFE,0x00,0xF8
,0x00,
0x60,0x00,0x40,0x00,0x40,0x00,0x80,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x09,0x00,0x0C,0x00,0x9E,0x00,0xDE,0x00,0xFE,0x00,0xF8,0x06,0xF9,0x06,0xD3
,0x2C,
0xC0,0x00,0x0C,0x00,0x1E,0x00,0x3E,0x00,0x3E,0x00,0x7C,0x00,0xF0,0x08,0xF8
,0x00,
0x60,0x00,0x37,0x00,0x10,0x00,0x00,0x00,0x03,0x00,0x03,0x00,0x01,0x00,0x07
,0x00,

0xF8,0x00,0xE3,0x00,0x07,0x00,0x1F,0x00,0xFE,0x01,0xF8,0x07,0xC0,0x3F,0xC0
,0x3F,
0xFF,0x00,0xFF,0x00,0xFF,0x00,0xE3,0x1C,0x0F,0xF0,0x1F,0xE0,0x3F,0xC0,0x7F
,0x80,
0xFD,0x03,0xFF,0x03,0x7F,0x87,0xF7,0x0F,0xFD,0x03,0xFF,0x00,0xFF,0x00,0xFF
,0x00,
0x90,0xE0,0xE0,0xF0,0xE0,0xF0,0xE0,0xE0,0x40,0x80,0x80,0x00,0xC0,0x00,0xF0
,0x00,
0x07,0x00,0x03,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x07,0x00,0x07
,0x00,
0xF7,0x08,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFE,0x01,0xFC,0x03,0xFF,0x00,0xFE
,0x01,
0xE0,0x10,0xC0,0x20,0xA0,0x40,0x00,0xC1,0x04,0xC3,0x08,0xC7,0x10,0x8F,0x80
,0x1F,
0x17,0x08,0x5F,0x20,0xBE,0x41,0x61,0x9E,0x04,0xF8,0x00,0xF8,0x08,0xF0,0x08
,0xF0,
0x80,0x7F,0x00,0xFF,0x00,0xFF,0xC2,0x3F,0x13,0x0E,0x10,0x0F,0x10,0x0F,0x30
,0x0F,
0x7F,0x80,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFC,0x03,0xF8,0x07,0x78
,0x87,
0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0x1F,0xE0,0x0F,0xF0,0x07
,0xF8,
0xF8,0x00,0xFE,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF
,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x00,0xE0,0x00,0xF0,0x00,0xF8,0x00,0xFC
,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x00,0x01,0x01,0x00,0x00,0x00,0x00
,0x00,
0xE6,0x01,0x1D,0x03,0x7B,0x07,0x8F,0x7F,0x03,0xFF,0x02,0xFF,0x02,0xFF,0x80
,0x7F,
0x80,0x3F,0x00,0xFF,0x00,0xFF,0x80,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0x04,0xF8,0x00,0xFF,0x00,0xFF,0x01,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0x40,0x3F,0x08,0xFF,0x38,0xFF,0x80,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0x18,0xE7,0x00,0xFF,0x00,0xFF,0x0E,0xFF,0x03,0xFF,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0x03,0xFC,0x31,0xFE,0x38,0xFF,0x7C,0xFB,0xF8,0xF1,0x38,0xF0,0x70,0xF0,0x60
,0xF0,
0xFF,0x00,0xFF,0x00,0xFE,0x00,0xFC,0x00,0x78,0x80,0x30,0xC0,0xB0,0x40,0x50
,0x20,
0xFC,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x00,0x08,0x00,0x00,0x00,0x00
,0x00,
0x40,0x3F,0x20,0x1F,0x10,0x0F,0x08,0x07,0x02,0x01,0x02,0x01,0x02,0x01,0x02
,0x01,

0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,
,0xFF,
0x20,0xF0,0x30,0xF0,0x18,0xF0,0x10,0xF8,0x10,0xF8,0x14,0xF8,0x08,0xFC,0x08
,0xFC,
0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x00,0x01,0x00,0x01,0x01,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x80,0x7F,0x40,0x3F,0x00
,0x3F,
0x08,0xFC,0x28,0xFC,0x60,0xFC,0x00,0xFC,0x02,0xFC,0x00,0xFE,0x01,0xFE,0x00
,0xFF,
0x40,0x00,0x60,0x00,0x90,0x00,0x10,0x00,0x10,0x00,0x80,0x00,0x81,0x00,0x83
,0x00,
0x00,0x00,0x00,0x00,0x18,0x00,0x39,0x00,0x7B,0x00,0xFB,0x00,0xFF,0x00,0xFF
,0x00,
0x00,0x00,0x00,0x00,0x88,0x00,0xD8,0x00,0xF8,0x00,0xF8,0x00,0xF0,0x00,0xF0
,0x00,
0x20,0x1F,0x00,0x1F,0x10,0x0F,0x11,0x0F,0x00,0x0F,0x00,0x0F,0x00,0x0F,0x08
,0x07,
0x00,0xFF,0xB7,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x3F,0xFF,0x1F,0xFF,0x07
,0xFF,
0x1F,0xFF,0xFE,0xFF,0xFE,0xFF,0xFC,0xFF,0xF8,0xFF,0xF8,0xFF,0xF8,0xFF,0xFC
,0xFF,
0x83,0x00,0x01,0x80,0x40,0x80,0x00,0xC0,0x20,0xC0,0x00,0xE0,0x10,0xE0,0x00
,0xF0,
0xFF,0x00,0xFF,0x00,0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0xFF,0x00,0xFF,0x00,0xFF,0x00,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0xE0,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x08,0x07,0x08,0x07,0x00,0x07,0x00,0x07,0x00,0x07,0x04,0x03,0x04,0x03,0x00
,0x03,
0x03,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0xFC,0xFF,0xFC,0xFF,0x7F,0xFE,0x3F,0xFE,0x3F,0xFE,0x3F,0xFE,0x3E,0xFF,0x3F
,0xFF,
0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x80,0x7F,0x00,0x7F,0x40,0x3F,0x00
,0x3F,
0x08,0xF0,0x00,0xF8,0x00,0xFC,0x00,0xFC,0x00,0xFE,0x00,0xFE,0x01,0xFE,0x00
,0xFF,
0x00,0x03,0x00,0x03,0x00,0x03,0x00,0x03,0x02,0x01,0x02,0x01,0x02,0x01,0x00
,0x03,
0x1F,0xFF,0x1F,0xFF,0x1F,0xFF,0x1F,0xFF,0x1F,0xFF,0x1F,0xFF,0x1F,0xFF,0x1F
,0xFF,
0x20,0x1F,0x80,0x1F,0x80,0x1F,0x10,0x8F,0x80,0x8F,0x88,0x87,0x80,0x87,0xC0
,0x87,

0x80,0x00,0x00,0x80,0x40,0x80,0x00,0xC0,0x00,0xC0,0x20,0xC0,0x00,0xE0,0x10
,0xE0,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x07,0x00,0x0F,0x00,0x1F
,0x00,
0x00,0x03,0x04,0x03,0x08,0x07,0xF0,0x0F,0xE0,0x1F,0xE0,0x1F,0xF0,0x0F,0xF0
,0x0F,
0x0F,0xFF,0x0F,0xFF,0x0F,0xFF,0x0F,0xFF,0x0F,0xFF,0x0F,0xFF,0x07,0xFF,0x01
,0xFF,
0xC4,0x83,0x80,0xC3,0x82,0xC1,0xC0,0xC1,0xE0,0xC1,0xE1,0xE0,0xE0,0xE0,0xD0
,0xE0,
0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x80
,0x7F,
0x10,0xE0,0x00,0xF0,0x08,0xF0,0x00,0xF8,0x00,0xF8,0x04,0xF8,0x00,0xFC,0x02
,0xFC,
0x1F,0x00,0x3F,0x00,0x3F,0x00,0x3F,0x00,0x3F,0x00,0x3F,0x00,0x00,0x1F,0x10
,0x0F,
0xF8,0x07,0xF8,0x07,0xF8,0x07,0xF0,0x0F,0xF0,0x0F,0xE0,0x1F,0x00,0xFF,0x00
,0xFF,
0x8F,0xFF,0x3F,0xFF,0x01,0xFF,0x00,0xFF,0x01,0xFF,0x03,0xFF,0x1F,0xFF,0x7C
,0xFF,
0xF0,0xE0,0xF0,0xE0,0xF0,0xE0,0xF1,0xE0,0xC0,0xE1,0xC3,0xE0,0x83,0xC0,0x03
,0xC0,
0x8F,0x70,0x3F,0xC0,0x7F,0x80,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF
,0x00,
0xF0,0x0F,0xFC,0x03,0xFE,0x01,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF
,0x00,
0x02,0xFC,0x00,0xFE,0x01,0xFE,0x01,0xFE,0x03,0xFC,0x01,0xFE,0x01,0xFE,0x01
,0xFE,
0x00,0x0F,0x00,0x07,0x06,0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x00,0xFF,0x00,0xFF,0x07,0xFF,0xFF,0x00,0x1E,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x07,0xFF,0x70,0xFF,0x02,0xFC,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x00,0xFF,0xCC,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x87,0x00,0x07,0x00,0x05,0x02,0x04,0x03,0x00,0x03,0x00,0x03,0x02,0x01,0x00
,0x01,
0xFF,0x00,0xFF,0x00,0xFF,0x00,0xFF,0x00,0x1F,0xE0,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0xFE,0x01,0xFC,0x03,0xF8,0x07,0xF0,0x0F,0xC0,0x3F,0x00,0xFF,0x00,0xFF,0x00
,0xFF,
0x01,0xFE,0x01,0xFE,0x01,0xFE,0x01,0xFE,0x01,0xFE,0x00,0xFE,0x02,0xFC,0x00
,0xFC,
0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
0x00,0xFF,0x80,0x7F,0xFF,0x00,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,

```

    0x00,0xFF,0x00,0xFF,0xFF,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,
    0xC4,0xF8,0x0C,0xF0,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00
};

```

4.6 mario_map.c

```

/*
Generated By GameBoyPngConverter

Tiles map

TileMap Size      : 20 x 18
*/

const unsigned char mario_map[] = {
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x02,0x03,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x05,0x
06,0x07,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x09,0x0A,0x0B,0x0C,0x0D,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x0F,0x10,0x11,0x12,0x13,0x14,0x15,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x16,0x17,0x18,0x19,0x1A,0x1B
,0x1C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1D,0x1E,0x
1F,0x20,0x21,0x22,0x23,0x24,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x16,0x25,0x26,0x27,0x28,0x29,0x2A,0x2B,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x2C,0x2D,0x2E,0x2F,0x30,0x31,0x32,0x33,0x34,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x3F,0x40,0x40,0x40,0x40,0x41,0x42,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x43,0x44,0x40,0x40,0x40,0x40,0x45,0x00,0x46,0x47,0x48,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x49,0x4A,0x4B,0x40,0x40,0x4C,0
x4D,0x4E,0x4F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x50,0x51
,0x52,0x53,0x40,0x54,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x55,0x40,0x56,0x57,0x40,0x40,0x58,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x59,0x5A,0x40,0x5B,0x5C,0x5D,0x40,0x5E,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x5F,0x60,0x40,0x61,0x62,0
x63,0x64,0x65,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x66
,0x67,0x68,0x69,0x6A,0x6B,0x6C,0x6D,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x6E,0x6F,0x70,0x71,0x00,0x00,0x00,0x00,
0x00,0x00
};

```

4.7 simplebackground.c

```
/*  
  
SIMPLEBACKGROUND.C  
  
Tile Source File.  
  
Info:  
  Form          : All tiles as one unit.  
  Format         : Gameboy 4 color.  
  Compression    : None.  
  Counter       : None.  
  Tile size     : 8 x 8  
  Tiles         : 0 to 6  
  
  Palette colors : None.  
  SGB Palette    : None.  
  CGB Palette    : None.  
  
  Convert to metatiles : No.  
  
This file was generated by GBTD v2.2  
  
*/  
  
/* Start of tile array. */  
unsigned char backgroundtiles[] =  
{  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x80,0x80,0x80,0x00,0x80,0x80,0x80,0x00,  
    0x80,0x80,0x80,0x00,0x80,0x80,0x80,0x00,  
    0x01,0x01,0x01,0x00,0x01,0x01,0x01,0x00,  
    0x01,0x01,0x01,0x00,0x01,0x01,0x01,0x00,  
    0x00,0x00,0x42,0x00,0x24,0x00,0x18,0x00,  
    0x18,0x00,0x24,0x00,0x42,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x18,0x00,0x18,0x00,0x7E,  
    0x00,0x7E,0x00,0x18,0x00,0x18,0x00,0x00,  
    0x00,0x00,0x00,0x0C,0x00,0x02,0x00,0x02,  
    0x00,0x02,0x00,0x02,0x00,0x0C,0x00,0x00,  
    0x00,0x00,0x60,0x00,0x80,0x00,0x80,0x00,  
    0x80,0x00,0x80,0x00,0x60,0x00,0x00,0x00  
};  
  
/* End of SIMPLEBACKGROUND.C */
```

4.8 simplebackground.h

```
/*  
  
SIMPLEBACKGROUND.H  
  
Include File.  
  
Info:  
  Form           : All tiles as one unit.  
  Format          : Gameboy 4 color.  
  Compression     : None.  
  Counter        : None.  
  Tile size      : 8 x 8  
  Tiles          : 0 to 6  
  
  Palette colors  : None.  
  SGB Palette     : None.  
  CGB Palette     : None.  
  
  Convert to metatiles : No.  
  
This file was generated by GBTD v2.2  
  
*/  
  
/* Bank of tiles. */  
#define backgroundtilesBank 0  
/* Start of tile array. */  
extern unsigned char backgroundtiles[];  
  
/* End of SIMPLEBACKGROUND.H */
```

4.9 simplebackgroundmap.c

```
/*  
  
SIMPLEBACKGROUNDMAP.C  
  
Map Source File.  
  
Info:  
  Section        :  
  Bank           : 0  
  Map size       : 20 x 36  
  Tile set       :  
C:\Users\mathi\OneDrive\Bureaublad\gbtd22\simple_background.gbr
```

Plane count : 1 plane (8 bits)
Plane order : Tiles are continues
Tile offset : 0
Split data : No

This file was generated by GBMB v1.8

*/

```
#define backgroundmapWidth 20  
#define backgroundmapHeight 36  
#define backgroundmapBank 0
```

```
unsigned char backgroundmap[] =  
{  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x06,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x05,0x00,0x00,0x00,0x03,0x00,0x04,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x06,0x00,0x00,0x00,0x00,0x00,0x04,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
    0x00,0x00,0x00,0x00,0x00,0x00,0x06,0x00,0x00,0x02,  
    0x01,0x00,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x00,  
    0x00,0x05,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,  
}
```



```
/* End of SIMPLEBACKGROUNDMAP.C */
```

4.10 simplebackgroundmap.h

```
/*  
  
SIMPLEBACKGROUNDMAP.H  
  
Map Include File.  
  
Info:  
    Section      :  
    Bank         : 0  
    Map size     : 20 x 36  
    Tile set     :  
C:\Users\mathi\OneDrive\Bureaublad\gbtd22\simple_background.gbr  
    Plane count  : 1 plane (8 bits)  
    Plane order  : Tiles are continues  
    Tile offset  : 0  
    Split data   : No  
  
This file was generated by GBMB v1.8  
  
*/  
  
#define backgroundmapWidth 20  
#define backgroundmapHeight 36  
#define backgroundmapBank 0  
  
extern unsigned char backgroundmap[];  
  
/* End of SIMPLEBACKGROUNDMAP.H */
```

5. Besluit

Dit was zeker geen makkelijk project en ik heb hier veel tijd ingestoken. Ik heb wel geluk gehad dat ik een goede tutorial en github heb gevonden die mij wat heeft kunnen helpen.

Je kan hier zeer uitgebreid in gaan en ik zie het zeker zitten om eens een meer advanced game te maken.

Het is wel jammer dat ik in mijn spel de background collision niet werkende heb gekregen. Ook had ik graag nog misschien een puntensysteem erin verwerkt.

Ik had ook nog een gameboy kunnen kopen en zo met een sd kaartje het spel op een echte gameboy kunnen zetten. Helaas was de levertermijn te lang en heb ik deze niet op tijd kunnen bestellen.

6. Bronnenlijst

DEVRS. (z.d.). Gameboy Tile Data Guide. Geraadpleegd van <https://www.devrs.com/gb/hmgd/gbtd.html>

DEVRS. (z.d.). Gameboy Memory Bank Guide. Geraadpleegd van <https://www.devrs.com/gb/hmgd/gbmb.html>

Wikipedia. (z.d.). Emulator. Geraadpleegd van <https://en.wikipedia.org/wiki/Emulator>

YouTube. (2018, 24/11). Geraadpleegd van https://www.youtube.com/watch?v=HIsWR_jLdwo&list=PLeEj4c2zF7PaFv5MPYhNAkBGrkx4iPGJo

BGB Emulator. (z.d.). Geraadpleegd van <https://bgb.bircd.org/>

SourceForge. (z.d.). GBDK - Game Boy Development Kit. Geraadpleegd van <https://sourceforge.net/projects/gbdk/files/gbdk-win32/2.95-3/>

GitHub. (z.d.). GamingMonstersGameBoySampleCode. Geraadpleegd van <https://github.com/gingemonster/GamingMonstersGameBoySampleCode>

GBDev Wiki. (z.d.). GBDK - Game Boy Development Kit. Geraadpleegd van <https://gbdev.gg8.se/wiki/articles/GBDK>

GBDev. (z.d.). Tools - Game Boy Development. Geraadpleegd van <https://gbdev.io/guides/tools.html>

GBDK-2020. (z.d.). GBDK-2020 API Documentatie. Geraadpleegd van <https://gbdk-2020.github.io/gbdk-2020/docs/api/index.html>