2024/11/02 @ バチャリアLT

ゲームボーイに画像表示させよう！

kimkim0106

自己紹介

kimkim0106

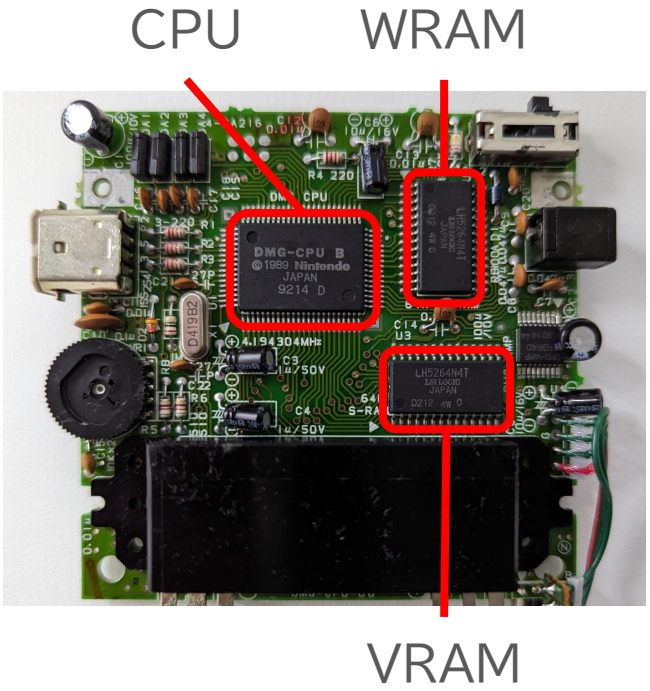
VRChat　: kimkim0106
Website : kimkim0106.net
X　　　: @kimkim0106_3218
その他　: ゲームボーイが好きです
　　VRChatのITインフラ集会スタッフ
　　本業はサーバエンジニア

ゲームボーイとは？

- 任天堂の携帯型ゲーム機
- 1989年 発売
  - モノクロ4階調液晶
- 1998年 カラー発売
  - 32000色カラー液晶
  - 性能もアップ

ゲームボーイのハードウェア

	ゲームボーイ(DMG-01)	ゲームボーイカラー
CPU	SHARP 8-bit CPU (Intel 8080 like) @ 4.19 MHz	SHARP 8-bit CPU (Intel 8080 like) @ 4.19 MHz, 8.39 MHz
WRAM	8 KiB	32 KiB
VRAM	8 KiB	16 KiB
Screen	モノクロ4階調 液晶 160 x 144	32768色カラー TFT液晶 160 x 144
Sound	4チャンネル ステレオ出力	4チャンネル ステレオ出力
Power	DC 6V 0.7W (単3電池 x 4)	DC 3V 0.6W (単3電池 x 2)

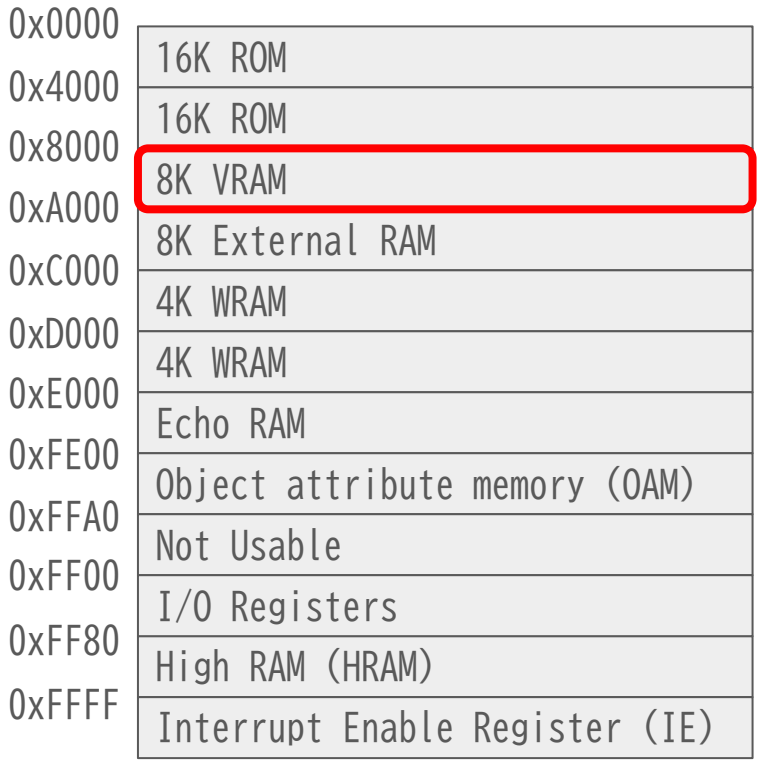


今回の主人公

- VRAM
  - 背景画像の表示に使う

今回は割愛しますが…

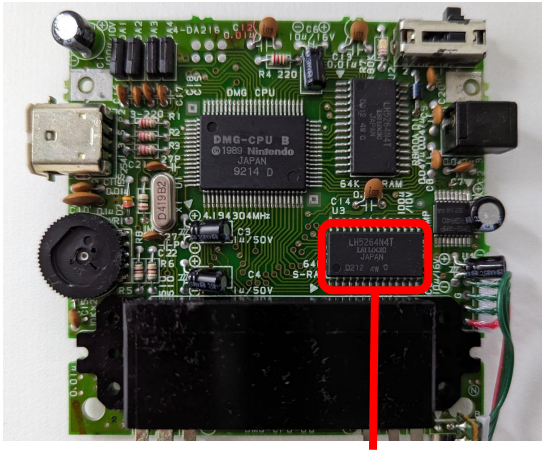
- OAM
  - スプライトの表示に使う



VRAM

- 背景画像の表示に使われる
- 容量は 8 KiB (8192 Byte)
  - ゲームボーイカラーの場合2バンク切り替え可能(16 KiB)

0x8000 - 0x97FF (6 KiB)	Tile Data
0x9800 - 0x9BFF (1 KiB)	Tile Maps
0x9C00 - 0x9FFF (1 KiB)	Tile Maps



VRAM

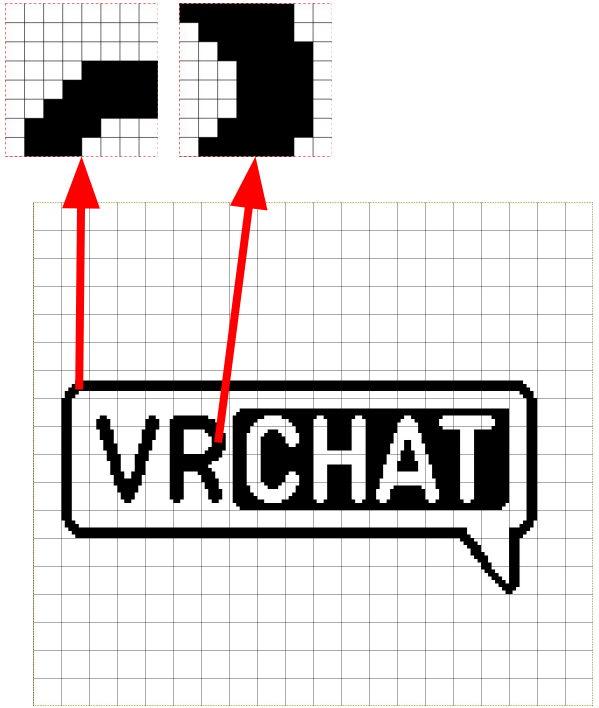
早速表示させてみよう！

用意するもの

- タイルデータ・タイルマップ
  - 画像もしくは文字
- コンパイラ
  - 今回はRGBDS
- エミュレータ
  - もしくはFlashROM + 実機

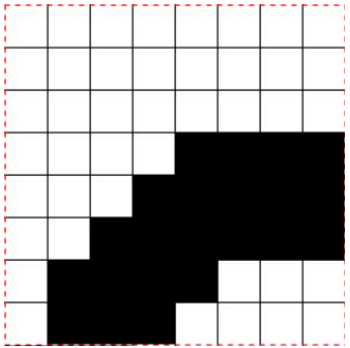
タイルデータとは

- 8×8の画像データ
  - 1ピクセルあたり2bitなので4色
  - （カラーパレットは別な場所）
- タイルを組み合わせて画像を表示
- 再利用できるので容量を削減可能
- VRAMには一度に384枚入る
  - 16 Byte/Tile
    - (8×8)×2 bit = 16 Byte
  - 16×384 = 6144 Byte



タイルデータのデータ構造

データ構造はシンプルなので、手作業でも作れる
…が、大変なので色々とツールがある



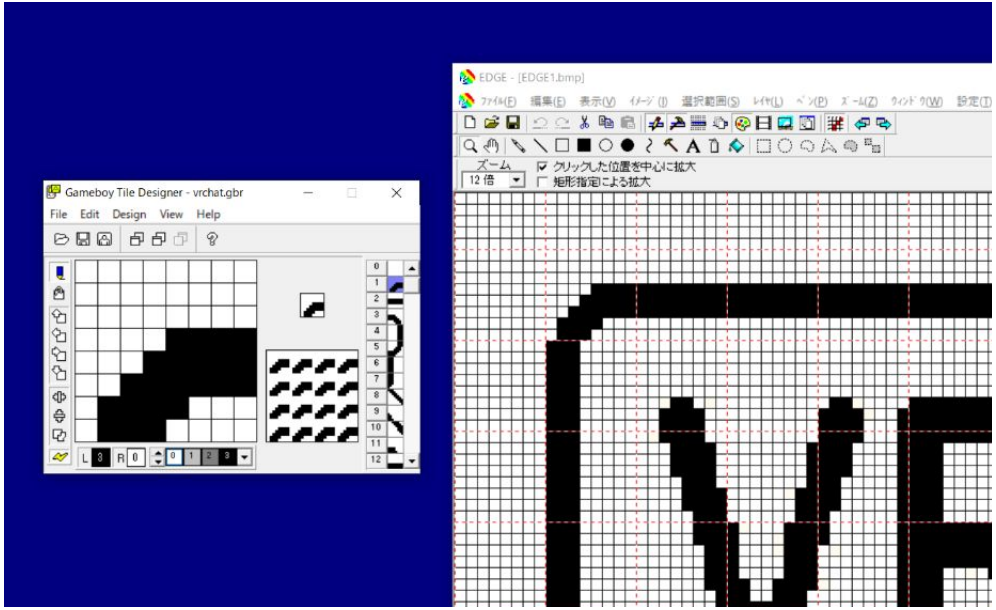
1pxあたり2bit
→ 00 00 00 00 11 11 11 11

1バイト目はそれぞれの上位1ビット
2バイト目はそれぞれの下位1ビット
→ 0b00001111 0b00001111
→ 0x0F 0x0F

タイルデータの作り方

今回はGameboy Tile Designer（GBTD）で作成

アセンブラで書き出せて便利



今回のタイルデータ

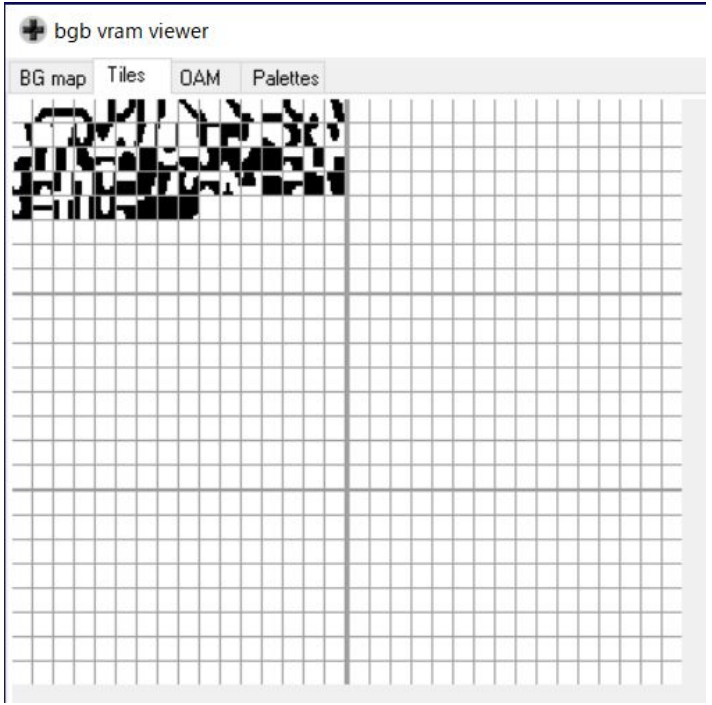
といって、バイナリ見せられてもわからないよね…

```
107  CharTiles:
108      db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
109      db $00,$00,$00,$00,$00,$00,$0F,$0F,$1F,$1F,$3F,$3F,$78,$78,$70,$70
110      db $00,$00,$00,$00,$00,$00,$FF,$FF,$FF,$FF,$FF,$FF,$00,$00,$00,$00
111      db $00,$00,$00,$00,$00,$00,$F0,$F0,$F8,$F8,$FC,$FC,$1E,$1E,$0E,$0E
112      db $07,$07,$07,$07,$07,$07,$07,$07,$07,$07,$07,$07,$07,$07,$07,$07
113      db $07,$07,$07,$07,$07,$07,$0E,$0E,$1E,$1E,$FC,$FC,$F8,$F8,$F0,$F0
114      db $E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0
115      db $E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$C0,$C0,$80,$80
116      db $E0,$E0,$70,$70,$38,$38,$1C,$1C,$0E,$0E,$07,$07,$03,$03,$01,$01
117      db $00,$00,$00,$00,$00,$00,$00,$00,$80,$80,$C0,$C0,$C0,$C0,$E0,$E0
118      db $3C,$3C,$1E,$1E,$0F,$0F,$07,$07,$07,$07,$03,$03,$01,$01,$00,$00
119      db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$F0,$F0,$F0,$F0,$F8,$F8
120      db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$FF,$FF,$FF,$FF,$FF,$FF
121      db $E0,$E0,$E0,$E0,$E0,$E0,$70,$70,$78,$78,$3F,$3F,$1F,$1F,$0F,$0F
122      db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$18,$18,$3C,$3C,$3C,$3C
123      db $3C,$3C,$1E,$1E,$1E,$1E,$1E,$1E,$0E,$0E,$0F,$0F,$0F,$0F,$07,$07
124      db $07,$07,$07,$07,$03,$03,$03,$03,$03,$03,$03,$03,$01,$01,$01,$01
125      db $01,$01,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
126      db $00,$00,$01,$01,$01,$01,$01,$01,$01,$01,$03,$03,$03,$03,$03,$03
127      db $87,$87,$87,$87,$87,$87,$87,$87,$87,$87,$CF,$CF,$CE,$CE,$FE,$FE
128      db $FE,$FE,$FC,$FC,$FC,$FC,$78,$78,$78,$78,$30,$30,$00,$00,$00,$00
129      db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$60,$60,$F1,$F1,$F1,$F1
130      db $F1,$F1,$E1,$E1,$E1,$E1,$E1,$E1,$C1,$C1,$C1,$C1,$C1,$C1,$81,$81
131      db $81,$81,$81,$81,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01
132      db $01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$00,$00,$00,$00,$00,$00
133      db $FF,$FF,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0
134      db $FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF,$E1,$E1,$E0,$E0,$E0,$E0
135      db $E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$E0,$C0,$C0,$00,$00,$00,$00
136      db $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$C0,$C0,$F0,$F0,$F8,$F8
137      db $FC,$FC,$7C,$7C,$3E,$3E,$1E,$1E,$1E,$1E,$1E,$1E,$3E,$3E,$7C,$7C
```

今回のタイルデータ

空白1枚 + 72枚 = 73枚

画像自体のサイズは 17×8=136枚
→ 半分近く使いまわしできた！



タイルデータについて補足

- 大きいゲームだと384枚では足りない
  - ひらがな・カタカナだけで3分の1使う
- なのでシーンに応じて切り替えている
  - RPGであれば、フィールド画面、戦闘画面 etc...
- 一気に書き換える場合は画面を暗転させたりする
- 実は書き換えには色々とテクニックはあり…
  - 描画途中にタイルを書き換えることができる
  - タイミングが限られるので割り込みとかを使ってやっているっぽい

タイルマップ

- タイルデータを並べるためのデータ
- 32×32(256×256px)のマップを2つ持てる
  - ゲームボーイの画面は 160×144 なのでスクロールできる

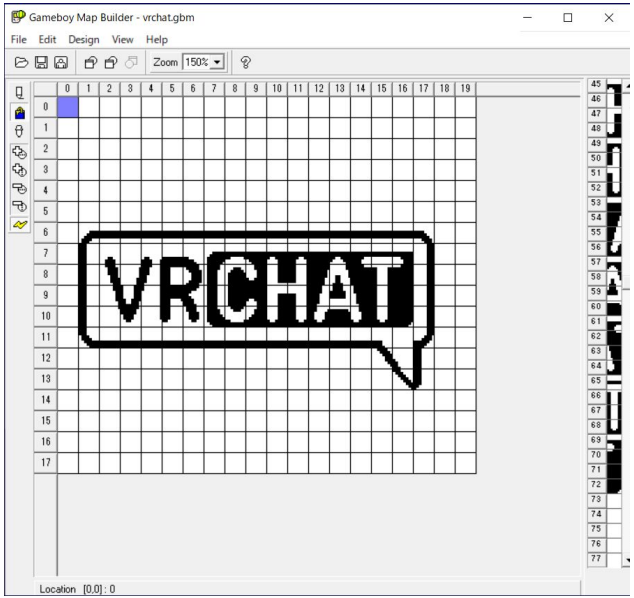
(イメージ図)
VRAMにタイル番号が順番に並んでいるだけ

00	00	00	00	00	00	00	...
00	01	02	02	02	02	02	...
00	06	0E	00	15	0C	1C	...
00	06	0F	12	16	16	1D	...

タイルマップ

今回は Gameboy Map Builder で作成

こちらもアセンブラで書き出せる

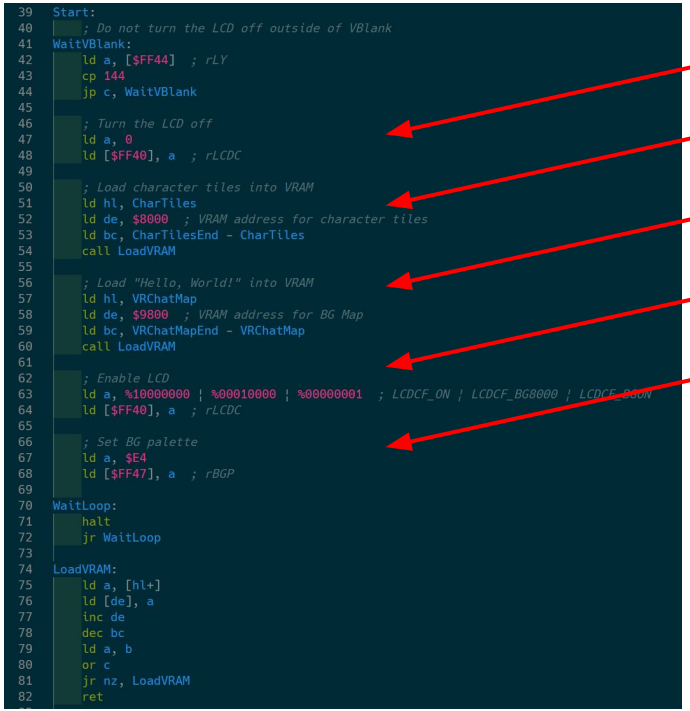


タイルマップ

32×18だけとりあえずセットした

```
86  VRChatMap:
87      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
88      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
89      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
90      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
91      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
92      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
93      db $00, $01, $02, $02, $02, $02, $02, $02, $02, $02, $02, $02, $02, $02, $02, $02, $02, $03, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
94      db $00, $06, $0E, $00, $15, $0C, $1C, $20, $24, $28, $2D, $31, $35, $39, $3D, $41, $45, $04, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
95      db $00, $06, $0F, $12, $16, $19, $1D, $21, $25, $29, $2E, $32, $36, $3A, $3E, $42, $46, $04, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
96      db $00, $06, $10, $13, $17, $1A, $1E, $22, $26, $2A, $2F, $33, $37, $3B, $3F, $43, $47, $04, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
97      db $00, $06, $11, $14, $18, $1B, $1F, $23, $27, $2B, $30, $34, $38, $3C, $40, $44, $48, $04, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
98      db $00, $0D, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0C, $0B, $00, $05, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
99      db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $0A, $09, $06, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
100     db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $08, $07, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
101     db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
102     db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
103     db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
104     db $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, 0,0,0,0,0,0,0,0,0,0,0,0,
105 VRChatMapEnd:
```

メモリにロードする

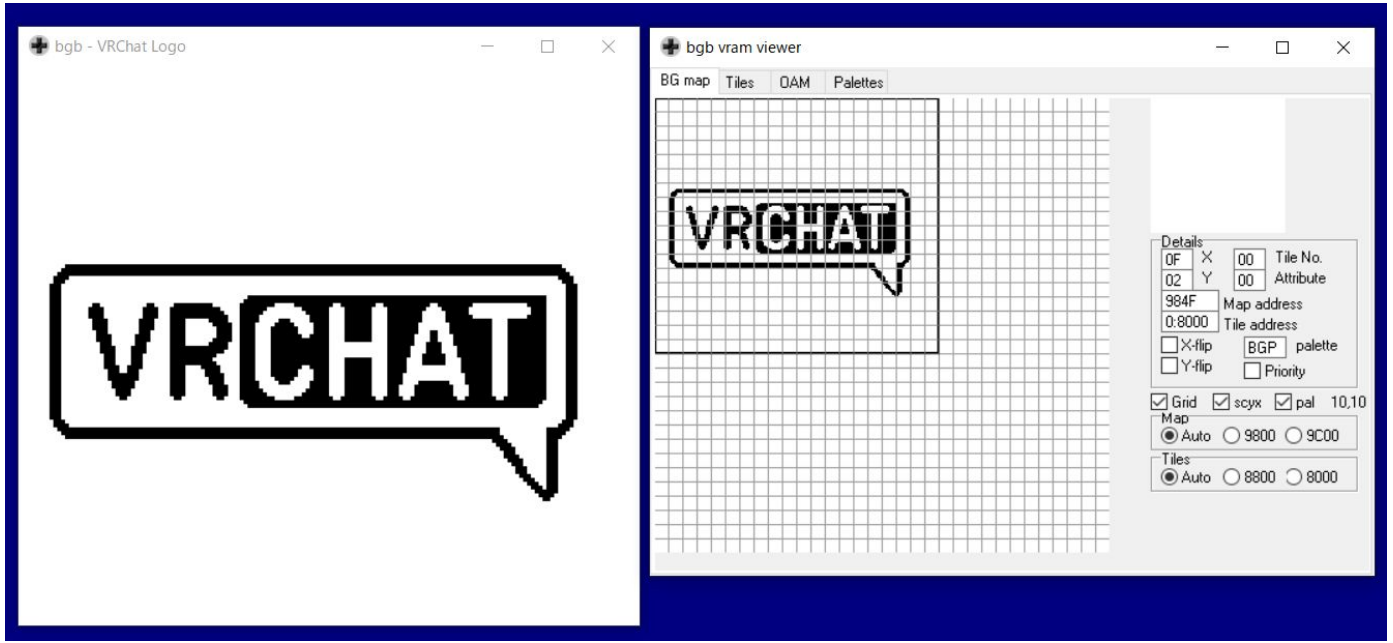


```
39  Start:
40      ; Do not turn the LCD off outside of VBlank
41  WaitVBlank:
42      ld a, [$FF44]  ; rLY
43      cp 144
44      jp c, WaitVBlank
45
46      ; Turn the LCD off
47      ld a, 0
48      ld [$FF40], a  ; rLCDC
49
50      ; Load character tiles into VRAM
51      ld hl, CharTiles
52      ld de, $8000  ; VRAM address for character tiles
53      ld bc, CharTilesEnd - CharTiles
54      call LoadVRAM
55
56      ; Load "Hello, World!" into VRAM
57      ld hl, VRChatMap
58      ld de, $9800  ; VRAM address for BG Map
59      ld bc, VRChatMapEnd - VRChatMap
60      call LoadVRAM
61
62      ; Enable LCD
63      ld a, %10000000 | %00010000 | %00000001  ; LCDCF_ON | LCDCF_BG8000 | LCDCF_BGON
64      ld [$FF40], a  ; rLCDC
65
66      ; Set BG palette
67      ld a, $E4
68      ld [$FF47], a  ; rBGP
69
70  WaitLoop:
71      halt
72      jr WaitLoop
73
74  LoadVRAM:
75      ld a, [hl+]
76      ld [de], a
77      inc de
78      dec bc
79      ld a, b
80      or c
81      jr nz, LoadVRAM
82      ret
```

- LCDをオフにする
- タイルをROMからVRAMに転送
- マップをROMからVRAMに転送
- LCDを設定・オンにする
- パレットを設定

動作確認

表示できた！


bgb - VRChat Logo
VRCHAT
bgb vram viewer
BG map
Tiles
OAM
Palettes
VRCHAT
Details
0F X
02 Y
00 Tile No.
00 Attribute
984F Map address
0:8000 Tile address
X-flip
Y-flip
BGP palette
Priority
Grid
scyx
pal 10,10
Map
Auto 9800 9C00
Tiles
Auto 8800 8000

実機で動かしてみた

ゲームボーイ版VRChat！（ではない）

おわり

- スクロールの話とかもしたかったなぁ…
  - ウインドウとか、パレットとかも
  - あとスプライトも

- リアル会場にいる方へ
  - 今回は一式持ってきてるのでお見せできます！
  - ソースコードとかその他もろもろもあります