

# asmboy4gb CHEATSHEET

## Graphics

All the values below can only be changed while in a VBlank.

### VRAM tiles

Every tile (8x8 pixels) consists of 16 bytes. 2 bit per pixel give the color index. Two bytes represent one line. For every pixel the most significant bit is taken from the second byte and the least significant bit from first byte for the line.

Address	Object	Background
\$8000-\$87FF	0-128	-
\$8800-\$8FFF	128-255	128-255
\$9000-\$97FF	-	0-127
The background is 32x32 tiles big (256x256 pixels). Every byte of the tilemap is one id of background tile from the table above.		
\$9800-\$981F	first line	
\$9820-\$983F	second line	
...		
\$9BE0-\$9BFF	last line	

### Object attribute memory (OAM)

Every of the 40 objects has 4 bytes in the memory area \$FE00-\$FE9F

Byte 0	Y position. Top = 16
Byte 1	X position. Left = 8
Byte 2	Object VRAM tile number
Byte 3	Is a bitmap, see table below
Bit 4	Palette number
Bit 5	X flip if set
Bit 6	Y flip if set
Bit 7	Background before objects if set

### Moving Background view

With \$FF42 the Y position and with \$FF43 the X position of the visible view is controlled. The visible view is 160x144 pixels out of the 256x256 pixels in the background tilemap.

### Color palettes

\$FF47 Is the color palette for the background tiles. The first two bits give the color for index 0, third and forth bit for color of index 1 etc.

0	White
1	Light gray
2	Darker gray
3	Black

\$FF48 and \$FF49 give the color for the object tiles. Which of this two palettes is taken is choosen by bit 4 in byte 3 in OAM. They work as the background palette but 0 is transparent instead of white.

~16kHz timer: \$FF04

### Functions to call

All the functions below can be called with their address. They dont modify any other registers besides the argument and the return.

#### \$0150 Wait for next VBlank

No arguments, no return. After this returns the VBlank just started and can be fully used to access graphic memory.

#### \$0163 Clear Background

No arguments, no return. Sets the background to the empty tile 0 which is set to all white. After this returns the VBlank just started and can be fully used to access graphic memory.

#### \$018C Get DPad

Returns all keys pressed in register a. If the bit is 0 it means the corresponding key is pressed

Bit 0	Right
Bit 1	Left
Bit 2	Up
Bit 3	Down

#### \$019E Get Buttons

Same as DPad. Start button will exit.	
Bit 0	A Button
Bit 1	B Button
Bit 2	Select Button
Bit 3	Start Button

#### \$01B0 Print Number

Number to print in a. Destination in tilemap \$9800-\$9BFF in hl

#### \$01D9 Print Game Over Screen

Destination in tilemap \$9800-\$9BFF in hl. Returns only after A was pressed.