

[console](#), [gameboy](#), [jeu-video](#), [homebrew](#), [em](#)

Game Boy

Différentes ressources concernant la console Game Boy

Histoire

DMG: Game Boy (monochrome), introduced on April 21, 1989
MGB: Game Boy (monochrome), introduced on April 21, 1989
MGL: Game Boy Light (monochrome), introduced on April 14, 1998
CGB: Game Boy Color (color), introduced on October 21, 1998
SGB: Super Game Boy, introduced on June 14, 1994
SGB2: Super Game Boy 2, introduced on January 30, 1998

Réaliser ses propres jeux et cartouches

Les jeux d'origine sont stockés sur cartouche, on appelle «ROM» le fichier numérique correspondant à un jeu.

[GB Studio](#) permet de créer un jeu, et de l'exporter pour qu'il soit jouable par un émulateur logiciel, par un émulateur dans le navigateur ou inscrit sur une cartouche. On peut aussi utiliser des outils de plus bas niveau en C ou assembleur...

Une page de ressources sur le sujet : <https://gbdev.io/resources.html> et une autre : <https://github.com/gbdev/awesome-gbdev>

Le Homebrew Hub recense des jeux créées par des passionné·e·s : <https://hh.gbdev.io/>

Pour mettre des jeux *homebrew* sur cartouche, il existe deux solutions : utiliser une cartouche avec carte microSD, ou un programmeur de cartouches vierges. La carte avec microSD coûte relativement cher mais ne nécessite pas de «consommables» alors que la deuxième solution crée une cartouche (réinscriptible) par jeu. Pour ces deux solutions, il existe différents fabricants, le site de GBstudio a publié en 2020 une veille approfondie sur ces sujets : <https://gbstudiocentral.com/tips/getting-your-gb-studio-games-on-real-hardware/>

Les portes logiques sont équipées des deux

- cartouche avec carte microSD : [Krikz EverDrive-GB X5](#)
- programmeur de cartouche GBxCart RW : [GBxCart RW](#)

Documents techniques

- <https://gbwiki.org/>
- <https://gbdev.io/pandocs/>

Réparation

Cache pile manquant

On peut trouver des modèles de cache-pile à imprimer sur thingiverse (chercher «game boy battery cover»)

- <https://www.thingiverse.com/thing:104919> par 666Aenur
- <https://www.thingiverse.com/thing:5742279>

Test avec le premier modèle de 666Aenur, c'est un peu grand mais ça rentre dans le boîtier avec des petits ajustements manuels (cutter, ponçage)



Émulation

binjgb <https://github.com/binji/binjgb>

émulateur et debugger de game boy / game boy color / super game boy en C utilisable aussi dans le navigateur (WebAssembly)

Cette version permet d'uploader un fichier ROM : <https://binji.github.io/binjgb/>

emulicious <https://emulicious.net/>

émulateur / debugger en java (Game Boy, Game Boy Color, Sega Master System, Sega Game Gear, MSX)

```
chmod +x Emulicious.jar
java -jar ./Emulicious.jar
```

pizza boy <https://play.google.com/store/apps/details?id=it.dbtecno.pizzaboy>

émulateur pour téléphones android

Bibliographie

Dans la bibliothèque des portes logiques



JV Hors série #15

Été 2024

<https://www.jvlemag.com/product/hors-serie-game-boy>

Game Boy Modding

Greg Farrell, 2020, No Starch Press

<https://nostarch.com/gameboymod>



Et aussi :

[La bible Game Boy](#), éditions Pix'n'Love

[L'histoire de Nintendo, vol.4 : la Game Boy](#), éditions Pix'n'Love

Ressources diverses

Reconstruire une ROM d'après une photo du circuit intégré : <https://github.com/travisgoodspeed/gbrom-tutorial>

Extraction de *tiles* en python : <https://github.com/stephaneseng/gameboy-tiles-extractor>

Reverse engineering a tilemap : <https://www.bradyanderson.tech/posts/gb-graphics-reverse-engineering-a-tilemap-part-1/>

Extracting a Gameboy cartridge ROM :

<https://cronop-io.github.io/posts/retrocomputing,%20binary%20analysis,%20hardware/2020-11-25-GameBoyPart1/>

Article extrait de : <https://lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse : <https://lesporteslogiques.net/wiki/ressource/gameboy?rev=1744215122>

Article mis à jour: **2025/04/09 18:12**