



# Documentation – Plateforme *Arcade*



## Objectif

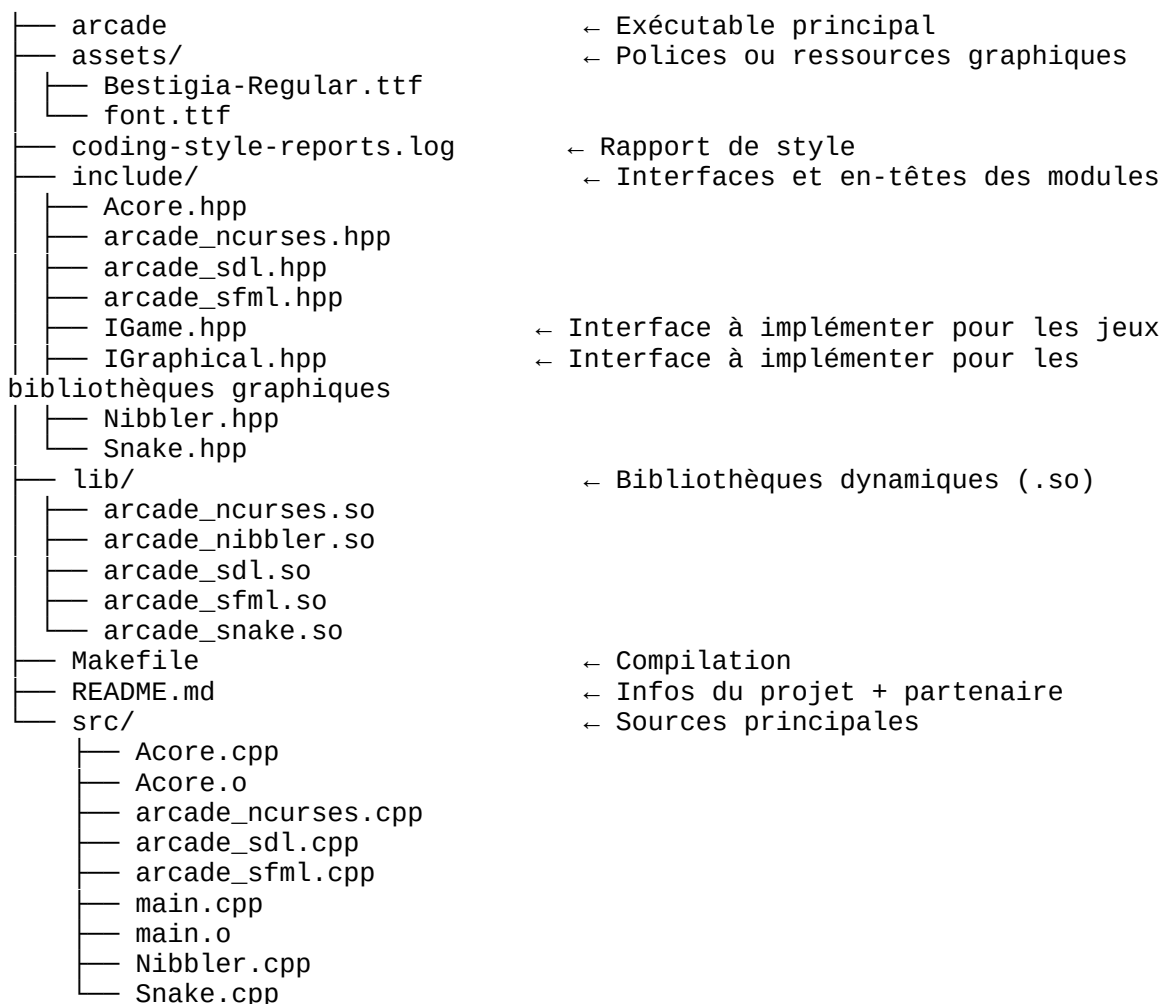
Le projet Arcade consiste à développer une plateforme de jeux rétro, en C++, modulaire et dynamique. Cette plateforme permet de charger des bibliothèques graphiques et des jeux à la volée grâce au système de chargement dynamique (dlopen, dlsym, etc.).

---



## Arborescence du projet

Voici l'organisation actuelle du projet :



## Fonctionnement général

Le programme ./arcade est un noyau qui charge dynamiquement :

- des **jeux** (comme Snake ou Nibbler)

- des **bibliothèques graphiques** (comme ncurses, SDL ou SFML)

Tout est fait **sans dépendance directe** : la plateforme utilise des **interfaces génériques** qui permettent de communiquer avec les modules chargés dynamiquement, sans savoir ce qu'ils contiennent exactement.

---



## Interfaces à respecter

Pour nous garantir la compatibilité entre les jeux et les bibliothèques graphiques, deux interfaces principales sont définies dans le dossier `include/` :



### **IGraphical.hpp**

Tout module graphique (ex : SDL, SFML, ncurses) doit **hériter de cette interface**. Elle définit les fonctions obligatoires à implémenter pour :

- Initialiser et fermer la fenêtre
- Gérer l'affichage du menu
- Dessiner la carte du jeu
- Récupérer les entrées clavier
- Afficher le nom du joueur



### **IGame.hpp**

Tout jeu (ex : Snake, Nibbler) doit **hériter de cette interface**. Elle définit les fonctions nécessaires pour :

- Initialiser le jeu
- Réinitialiser une partie
- Mettre à jour l'état du jeu selon les entrées
- Fournir la carte à dessiner
- Transmettre le score
- Indiquer si le jeu est terminé



Pour créer un nouveau jeu ou une nouvelle bibliothèque, il suffit **d'hériter de la bonne interface** (respectivement `IGame` ou `IGraphical`) et de respecter les fonctions demandées.

---



## Utilisation de la plateforme

Lancement classique :

```
./arcade ./lib/arcade_sfml.so
```

Cela démarre le menu avec la bibliothèque SFML. Depuis ce menu, on peut :

- Sélectionner un jeu
  - Changer de moteur graphique
  - Saisir un nom de joueur
  - Jouer, recommencer, quitter
- 

## Création de jeux et de bibliothèques graphiques

### Ajouter un jeu (ex : sokoban)

1. Créer un fichier `sokoban.hpp` dans `include/` qui hérite de `IGame`.
2. Créer un fichier `sokoban.cpp` dans `src/` ou `games/` pour l'implémentation.
3. Compiler en `.so` et le placer dans `lib/`.

### Ajouter une librairie graphique (ex : OpenGL)

1. Créer un fichier `arcade_opengl.hpp` dans `include/` qui hérite de `IGraphical`.
  2. Implémenter dans `src/` ou `graphics/`.
  3. Compiler en `.so` et placer le fichier généré dans `lib/`.
- 

## Compilation

### Avec Makefile :

- `make` : compile tout
  - `make core` : uniquement le noyau (arcade)
  - `make games` : compile les jeux (\* .so)
  - `make graphics` : compile les bibliothèques graphiques
  - `make clean / fclean / re` : nettoyage et recompilation
- 

## Système de scores

Les jeux doivent retourner un score. Celui-ci peut être stocké dans un fichier ou affiché directement dans le menu pour chaque joueur.

---