

DOCUMENTATION

1. Implémentation de nouvelles bibliothèques (graphique ou jeu)

1.1. Structure générale

Toutes les bibliothèques (graphique ou jeu) doivent respecter une interface commune afin de garantir l'interopérabilité avec le moteur principal. Ces interfaces sont fournies sous forme d'entêtes (IGraphical.hpp et IGame.hpp) et doivent être implémentées dans chaque bibliothèque.

1.2. Interface graphique (IGraphical)

Chaque bibliothèque graphique doit :

- Implémenter la classe IGraphical
- Être compilée sous forme de bibliothèque dynamique (.so)
- Respecter les méthodes suivantes :

1.3. Interface jeu (IGame)

Chaque bibliothèque de jeu doit :

- Implémenter la classe IGame
- Être compilée sous forme de bibliothèque dynamique (.so)
- Respecter les méthodes suivantes :

1.4. Nom de la fonction d'entrée (extern "C")

Chaque bibliothèque doit contenir une fonction C pour permettre le chargement dynamique :

2. Diagramme de classes

(Inclure ici un schéma UML ou un fichier .png avec les classes suivantes par exemple :)

- Core (le gestionnaire principal)
- DynamicLoader (chargement des .so)
- IGraphical (interface graphique)
- IGame (interface de jeu)
- ConcreteGraphical (ex : SDL, SFML, Ncurses)
- ConcreteGame (ex : Snake, Pacman, etc.)

3. Manuel explicatif

3.1. Lancement

Le main.cpp instancie le Core, qui charge dynamiquement une bibliothèque graphique et une bibliothèque de jeu.

3.2. Boucle principale

1. Le Core appelle getInput() sur la bibliothèque graphique.
2. Cet input est transmis à la bibliothèque de jeu via update().
3. Le GameState mis à jour est renvoyé.
4. Ce GameState est affiché via render() de la bibliothèque graphique.
5. Si l'utilisateur change de bibliothèque (graphique ou jeu), le Core relance le cycle avec la nouvelle lib