

DOCUMENTATION

ARCADE

Chef de l'équipe avec laquelle on a partagé nos interfaces : frederic.ye@epitech.eu

Description :

Le but du projet "ARCADE" est de créer 3 librairies graphiques et 2 jeux.

Lors du lancement du jeu, nous devons avoir la possibilité de changer de librairie graphique et de jeu.

Information :

Langage de programmation : C++

Equipe :

Joris Bertaigne : joris.bertaigne@epitech.eu

Elias El-Fallah : ellias.el-fallah@epitech.eu

Eric Tran : eric1.tran@epitech.eu

Temps du projet : 35 jours

Comment jouer :

1ère étape :

Tapper la commande “make” dans le dossier OOP_arcade_2019 afin de créer le binaire “arcade”

2ème étape :

Utilisez le binaire “arcade” en ajoutant en argument une librairie graphique qui vous intéresse.

Par exemple : ./arcade ./lib/lib_arcade_sfml.so

3ème étape :

Suivez les instructions dans le terminal

4ème étape :

Jouer !

Toutes les commandes possibles :

“make” : Compile le core, les librairies de jeux et les librairies graphiques

“make core” : Compile le core

“make games” : Compile et crée les librairies de jeux

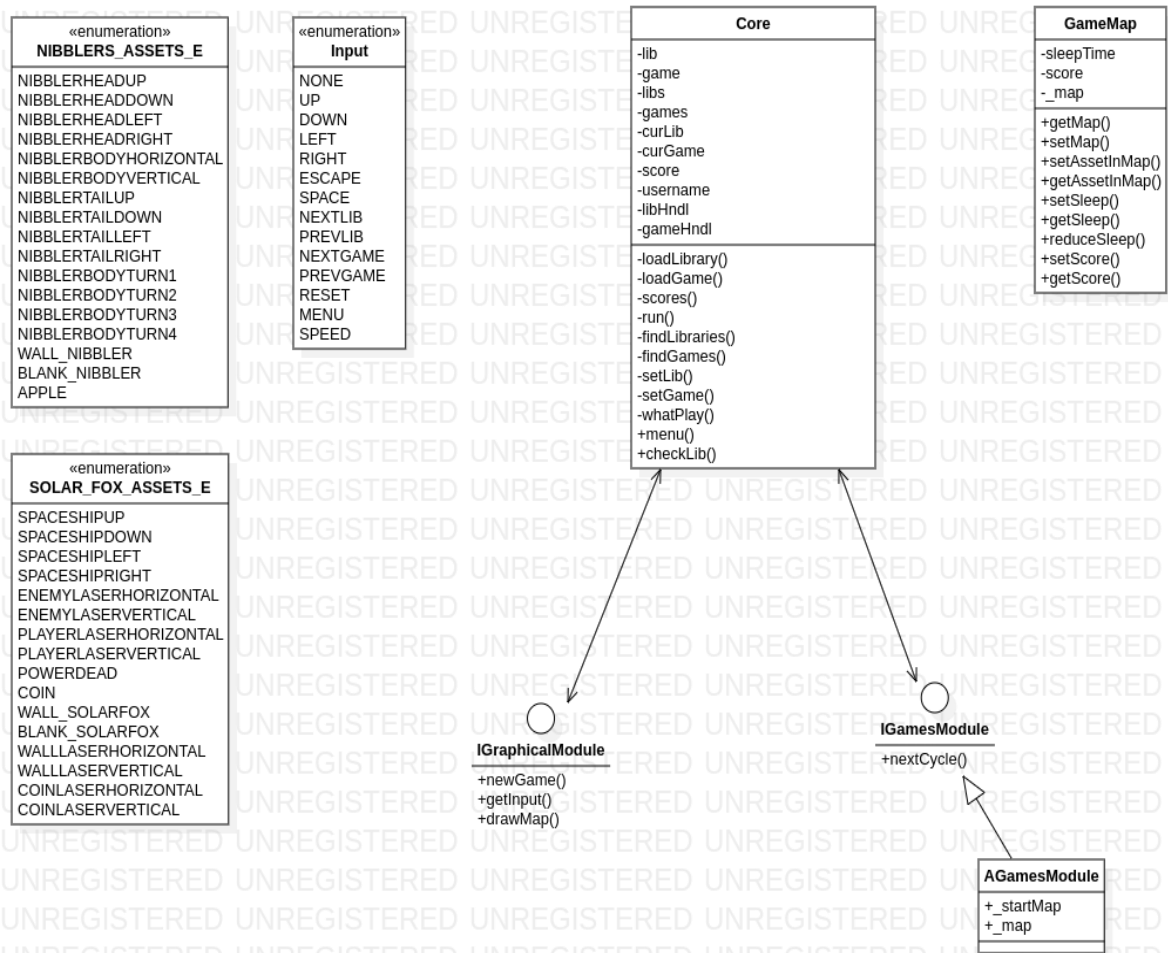
“make graphics” : Compile et crée les librairies graphiques

“make clean” : Supprime les .o

“make fclean” : Supprime les librairies et supprime les .o

“make re” : Recompile le projet en faisant en “make fclean” puis un “make”

UML :



Comment implémenter une librairie graphique :

Pour implémenter une librairie graphique, il faut tout d'abord que votre classe hérite de l'interface "IGraphicalModule".

Il faut que votre classe implémente les fonctions virtuelles de l'interface "IGraphicalModule".

virtual void newGame(std::string newGame)

: NewGame permet de créer les textures nécessaires aux jeux.

Le nom du jeu est indiqué grâce à "std::string newGame".

virtual Input getInput()

: Renvoie une valeur de l'énumération "graphics::Input" dans "Assets.hpp" selon la touche pressé.

virtual void drawMap(const utilities::GameMap &map)

: Affiche les textures selon une vector<vector<int>> appartenant à "const utilities::GameMap &map".

Pour le récupérer, il suffit d'utiliser le getter de GameMap qui est "getMap()".

Comment implémenter un jeu :

Pour implémenter une librairie graphique, il faut tout d'abord que votre classe hérite de la classe abstraite "AGamesModule" qui lui hérite de l'interface "IGamesModule".

Il faut que votre classe implémente les fonctions virtuelles de l'interface "IGamesModule".

```
virtual const utilities::GameMap &nextCycle(graphicals::Input input  
= graphicals::Input::NONE);
```

: nextCycle est une fonction qui effectue une itération de la logique du jeu. Elle prend en argument un graphicals::Input pour indiquer l'action du joueur. La fonction renvoie une copie de gameMap _map qui est présente dans la classe abstraite.