

Arcade

- Antoine-Zachary Khalidy (Gestionnaire de projet, Core et des librairies graphiques)
- Yoan Gerrard (Créateur de Pacman)
- Louis Lejaille (Créateur de Snake)

Sommaire

- **Les Interfaces**
- **Le Makefile**
- **Implémentations Jeux, Graphiques et Audio**

Les Interfaces

Les interfaces sont multiples :

- IAudio
- IGraphical
- IGame
- IData
- IEntity

Chacune est spécifique à une partie du projet et est indépendante pour la majeure partie.

IAudio : IAudio, comme son nom l'indique, est la partie audio des jeux. Elle permet d'ajouter de la musique pour rendre le jeu plus vivant.

IGraphical: IGraphical s'occupe de la partie graphique des jeux et permet d'obtenir un rendu visuel, que ce soit en mode Terminal, 2D ou même 3D.

IGame: IGame s'occupe de la partie principale du jeu, tout ce qui est gestionnaire des entrées de l'utilisateur doit être réalisé dedans et doit remplir IData pour que IGraphical affiche ce qui a été réalisé dans IGame.

IData: IData contient les données principales du jeu, que ce soit le score, le nombre de vies, le niveau ou la carte. Toutes ces données doivent être fournies à IData pour que le core puisse les utiliser ensuite.

IEntity: IEntity est chaque partie de la carte de IData. Par exemple, un mur est une entité, c'est-à-dire qu'elle peut avoir une couleur, un sprite, une position, etc. IEntity permet d'avoir une carte flexible, que ce soit en 2D, 3D ou en mode Terminal.

Makefile

Le Makefile est assez simple à utiliser malgré ce que l'on pourrait croire :

- Tout en haut, il y a les noms des bibliothèques partagées dans notre cas `arcade_[lib].so`.
- En dessous, il y a l'emplacement des fichiers sources `.cpp` et `.hpp` nécessaires à l'utilisation de la bibliothèque.
- Ensuite, il y a les `OBJ`, qui sont le moment où ils sont transformés pour que notre PC puisse les lire.
- Maintenant, les flags. La plupart des bibliothèques ont besoin de flags pour être utilisées, c'est ici que vous devez mettre vos flags.
- Enfin, toutes les règles sont en dessous.

Implémentations

Les implémentations dans le projet Arcade se décomposent en trois parties principales: les jeux, les bibliothèques graphiques et les bibliothèques audio.

1. Pour implémenter un jeu, il faut créer un nouveau dossier dans le dossier "Game" et nommer celui-ci comme on le souhaite. Ensuite, il faut créer un fichier hpp et un fichier cpp. Dans le fichier hpp, il faut créer une classe qui hérite de l'interface IGame et implémenter les fonctions associées. Dans le fichier cpp, on réalise la partie logique du jeu en utilisant le vecteur d'entiers pour les inputs de l'utilisateur, l IData pour les données à afficher et le temps pour gérer les mouvements ou les animations. Enfin, dans le Makefile, on ajoute les informations nécessaires pour que cela fonctionne, notamment le nom, le chemin et les flags. On copie-colle ensuite les lignes dans les règles "games" et on les modifie en conséquence.
2. Pour les bibliothèques graphiques, il faut créer un nouveau dossier dans le dossier "Graphical" et nommer celui-ci comme on le souhaite. Ensuite, il faut créer un fichier hpp et un fichier cpp. Dans le fichier hpp, il faut créer une classe qui hérite de l'interface IGraphical et implémenter les

fonctions associées. Ensuite, il faut remplir la fonction `getInputs` pour que le core reçoive une valeur à chaque fois que l'utilisateur appuie sur une touche. Enfin, on réalise une double boucle en utilisant `IData` comme source pour afficher chaque entité selon nos besoins (caractères pour une librairie graphique en mode terminal, sprites pour une librairie graphique en mode 2D ou 3D). Dans le Makefile, on ajoute les informations nécessaires pour que cela fonctionne, notamment le nom, le chemin et les flags. On copie-colle ensuite les lignes dans les règles "graphics" et on les modifie en conséquence.

3. Pour les librairies audio, on suit le même processus que pour les librairies graphiques. On crée un nouveau dossier dans le dossier "Audio" et on nomme celui-ci comme on le souhaite. Ensuite, on crée un fichier `hpp` et un fichier `cpp`. Dans le fichier `cpp`, on génère notre musique à partir du chemin et on la joue. Dans le Makefile, on ajoute les informations nécessaires pour que cela fonctionne, notamment le nom, le chemin et les flags. On copie-colle ensuite les lignes dans les règles "audios" et on les modifie en conséquence.

Voici comment implémenter dans mon projet arcades les librairies.