

Documentation projet :

Arcade

A Retro Platform

La documentation suivante est un référencement de l'ensemble des choses à savoir lorsque vous voulez utiliser le programme « Arcade ».

Le projet a été réalisé dans le cadre de la deuxième année du cursus d'Epitech par Jean-Baptiste ROESCH, Florian GOLLING et Dmitry YAKOVLEV.

I. Utilisation du programme

1. Installation

Pour utiliser l' « Arcade », il vous faudra d'abord avoir le programme sur votre poste. Il faudra également procéder au téléchargement des librairies suivantes :

- NCURSES
- SDL
- SFML

Une fois ces librairies mise en place la partie installation est terminée.

2. Compilation

Pour la compilation du projet, il vous suffira d'entrer les commandes suivantes dans le terminal en vous situant là où se trouvent les fichiers du programme.

- « make » : pour compiler l'ensemble de l'Arcade.
- « make graphics » : pour compiler les librairies graphiques de l'Arcade.
- « make games » : pour compiler les librairies de jeux de l'Arcade.
- « make core » : pour compiler le « Core » de l'Arcade.

3. Une fois le programme lancé

Une fois les étapes du dessus réalisées, rentrez cette commande au même endroit où vous avez lancé la compilation :

```
./arcade ./lib/arcade_XXXX.so
```

Veillez à remplacer les « XXXX » par le nom de la librairie graphique souhaitée (par défaut : ncurses, sfml et sdl).

a. Le menu

Le menu est utilisé afin de sélectionner un jeu, de changer de lib graphique et à renseigner un nom de joueur.

Voici quelques touches qui pourront vous être utiles :

- Echap : permet de quitter le programme.
- F1, F2, F3 : permettent de changer de librairie graphique.
- Les touches directionnelles du bas et du haut : permettent de naviguer.
- Entrée : permet de sélectionner le jeu souhaité.
- Les caractères alpha-numériques : permettent de rentrer un pseudonyme.

b. Le pacman

Le jeu consiste à déplacer Pac-Man, un personnage qui, vu de profil, ressemble à un diagramme circulaire à l'intérieur d'un labyrinthe, afin de lui faire manger toutes les pac-gommes qui s'y trouvent en évitant d'être touché par des fantômes.

Voici quelques touches qui pourront vous être utiles :

- Echap : permet de revenir au menu.
- Z, S, Q, D et les flèches directionnelles : permettent de se déplacer.
- Espace : permet de mettre le jeu en pause.

c. Le nibbler

Le jeu nibbler est très similaire au jeu plus connu Snake. Le joueur contrôle un serpent où le but est de manger des pommes pour agrandir le serpent sans se prendre de mur, sans sortir de la zone de la map et sans se mordre la queue.

Voici quelques touches qui pourront vous être utiles :

- Echap : permet de revenir au menu.
- Z, S, Q, D et les flèches directionnelles : permettent de se déplacer.
- Espace : permet de mettre le jeu en pause.

II. Documentation technique du projet

1. Rajouter une librairie de jeu

Si vous souhaitez rajouter une librairie de jeu en plus de celles existantes, il vous faudra implémenter les fonctions nécessaires à la compatibilité avec le reste de l'Arcade et compiler votre nouvelle librairie de jeu de façon à obtenir un fichier .so.

Les fonctions à implémenter sont les suivantes :

- `void draw(IGraphic&)` : permet d'afficher des éléments graphique du jeu.
- `CommandType getEvent(IGraphic&)` : permet de gérer les entrées utilisateur.
- `void update(double timeElapsed)` : permet d'actualiser le jeu.
- `void setPlayerName(std::string& player)` : permet de mettre à jour le pseudonyme du joueur.

Nous mettons également à votre disposition un système de highscore, pour l'utiliser il suffira de faire appel à la fonction « `save_score` » présente dans le fichier `SaveHighScore.cpp`, et lui envoyer le pseudonyme du joueur, le score ainsi que le nom de votre jeu. N'oubliez pas de l'intégrer dans vos Makefile !

2. Rajouter une librairie graphique

Si vous souhaitez rajouter une librairie de jeu en plus de celles existantes, il vous faudra implémenter les fonctions nécessaires à la compatibilité avec le reste de l'Arcade et compiler votre nouvelle librairie de jeu de façon à obtenir un fichier .so.

Les fonctions à implémenter sont les suivantes :

- `void drawPixel(Pixel *pixel)` : permet de dessiner des pixels sur la fenêtre de jeu.
- `void drawText(Text *text)` : permet d'afficher du texte sur la fenêtre de jeu.
- `void myClear()` : permet de réinitialiser la fenêtre.
- `void myRefresh()` : permet de rafraîchir la fenêtre de jeu.
- `CommandType getInput()` : permet de récupérer l'action de l'utilisateur.
- `std::string getName()` : permet de récupérer le nom de la librairie.

3. Pannel d'outils

Si vous le souhaitez, nous vous offrons la possibilité d'utiliser les classes `Pixel` et `Texte` présents dans le dossier « `utils` » qui faciliteront la gestion de l'affichage. Pour générer un pixel, il faut renseigner sa position sur l'axe x, sa position sur l'axe y, sa couleur et sa taille. Pour le texte veuillez à renseigner sa position sur l'axe x, sur l'axe y, la phrase à afficher, sa couleur et sa taille.