Arcade

Documentation

Utilisation de GitBook

Vous pouvez ouvrir le fichier `DocumentationArcadeInterfaces.pdf` dans le dossier `/doc` pour consulter la documentation des interfaces `IDisplayMenu`, `IDisplayModule`, `IGameModule` et `IType`.

Explication des interfaces

Interface IDisplayMenu

- getSelectedGraphicLibrary() : retourne le nom de la bibliothèque graphique sélectionnée.
- getSelectedGameLibrary() : retourne le nom de la bibliothèque de jeu sélectionnée.
- setPlayerName(name) : défini le nom du joueur.
- getPlayerName() : retourne le nom du joueur.
- initializeAvailableLibraries(gameLibraries, GraphiLibraries) : initialise les bibliothèques de jeux et graphiques disponibles.
- initializeBackground(displayModule) : initialise l'arrière-plan pour le module d'affichage.
- displayTextOnScreen(displayModule) : affiche du texte à l'écran en utilisant le module d'affichage fourni.
- displayBackgroundImage(displayModule) : affiche l'image d'arrière-plan en utilisant le module d'affichage fourni.
- displayPlayerNameOnScreen(displayModule) : affiche le nom du joueur à l'écran en utilisant le module d'affichage fourni.
- handleEvent(event, displayModule) : gère un événement en utilisant le module d'affichage fourni.

Interface IDisplayModule

- drawPixel(pixelPtr) : prend en entrée un pointeur Pixel et dessine un carré de pixels avec la taille, la position et la couleur données.
- drawText(textPtr) : prend en entrée un pointeur Text et affiche du texte avec la position et la taille données.
- clearScreen() : efface l'écran entre chaque boucle de jeu.
- refreshScreen() : rafraîchit les données et met à jour l'écran.
- getInput() : récupère l'entrée du clavier de l'utilisateur et retourne un Arcade::CommandType.

Interface IGameModule

- initializePlayerName(playerName) : initialise le nom du joueur.
- drawGraphics(displayModule) : dessine les changements effectués pendant la boucle de jeu en utilisant le module d'affichage fourni.
- handleEvent(cmd, displayModule)
- handleEvent(cmd, displayModule) : gère les événements en fonction de l'entrée du clavier en utilisant le module d'affichage fourni.
- updateGameState(timeElapsed) : met à jour l'état du jeu en fonction du temps écoulé depuis le dernier tour de boucle.

Enums (IType)

Arcade::CommandType

Enumération des types de commandes correspondant aux touches du clavier.

Arcade::LibraryType

Enumération des types de bibliothèques (GRAPHIC, GAME, MENU).

Arcade::Color

Enumération des couleurs disponibles pour les pixels et le texte.

Arcade::MaxWindow

Enumération des dimensions maximales de la fenêtre de jeu (MYCOLS, MYLINES).

Description

Le projet Arcade consiste à créer une borne d'arcade avec au moins deux jeux et un système de chargement dynamique de trois bibliothèques graphiques différentes.

Usage

```
>$ make -j
>$ ./arcade lib/dynamic_library.so
```

or

```
>$ ./arcade ./lib/dynamic_library.so
```

Problèmes rencontrés

Seuls les .so de ncurses (arcade_ncurses.so) et sfml (arcade_sfml.so) peuvent être lancés pour le moment.

Choix de dynamic_library :

arcade_ncurses.so || arcade_sfml.so || arcade_sdl2.so || arcade_menu.so || arcade_snake.so || arcade_nibbler.so

---

Libraries utilisées

- Sdl2
- Sfml
- Ncurses

Jeux

- Snake
- Nibbler

Groupe

- Marc Allari
- Pierre Maurer
- Romain Reynaert

Contact:
