

Bibliothèques Graphiques

Implémentation et utilisation des bibliothèques d'affichage dans ARCADE

Introduction aux bibliothèques graphiques

Les bibliothèques graphiques sont des composants essentiels du projet ARCADE. Elles permettent d'afficher les jeux avec différents rendus graphiques tout en maintenant une logique de jeu cohérente. L'architecture du projet permet de changer de bibliothèque graphique en temps réel, sans redémarrer l'application.

Chargement dynamique

Les bibliothèques graphiques sont chargées dynamiquement à l'exécution sous forme de bibliothèques partagées (.so).

Interface IDisplayModule

Toutes les bibliothèques graphiques doivent implémenter l'interface `IDisplayModule`. Cette interface définit les méthodes nécessaires pour initialiser, afficher et gérer les entrées utilisateur.

```
namespace arcade {
    class IDisplayModule {
    public:
        virtual ~IDisplayModule() = default;
        virtual void init() = 0;           // Initialise la bibliothèque graphique
        virtual void stop() = 0;           // Arrête la bibliothèque graphique
        virtual void clear() = 0;          // Efface l'écran
        virtual void render() = 0;         // Effectue le rendu
        virtual void present() = 0;        // Présente le rendu à l'écran
        virtual bool isKeyPressed(const std::string &key) = 0; // Vérifie si une touche est pressée
        virtual std::string getName() const = 0; // Retourne le nom de la bibliothèque
        virtual void handleInput() = 0;    // Gère les entrées utilisateur
        virtual bool shouldQuit() const = 0; // Vérifie si l'utilisateur veut quitter
        virtual void drawRect(int x, int y, int width, int height, int r, int g, int b, int a) = 0; // Dessine un rectangle
        virtual void drawText(const std::string& text, int x, int y, int r, int g, int b, int a) = 0; // Affiche du texte
        virtual void drawCircle(int x, int y, int radius, int r, int g, int b, int a) = 0; // Dessine un cercle
    };
}
```

Copy

Description des méthodes

init()

Initialise la bibliothèque graphique (création de fenêtre, chargement de ressources, etc.).

stop()

Arrête proprement la bibliothèque graphique (libération des ressources, fermeture de fenêtre).

clear()

Efface le contenu de l'écran pour préparer un nouveau rendu.

render()

Effectue le rendu des éléments graphiques.

present()

Présente le rendu à l'écran (équivalent à un swap buffer).

isKeyPressed(const std::string &key)

Vérifie si une touche spécifique est pressée. Les touches sont identifiées par des chaînes standardisées comme "UP", "DOWN", "ESCAPE", etc.

getName() const

Retourne le nom de la bibliothèque graphique pour l'affichage dans l'interface.

handleInput()

Gère les événements d'entrée utilisateur (clavier, souris).

shouldQuit() const

Indique si l'utilisateur a demandé à quitter l'application (par exemple en fermant la fenêtre).

drawRect(int x, int y, int width, int height, int r, int g, int b, int a)

Dessine un rectangle aux coordonnées (x,y) avec la largeur, hauteur et couleur RGBA spécifiées.

drawText(const std::string& text, int x, int y, int r, int g, int b, int a)

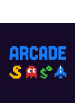
Affiche du texte aux coordonnées (x,y) avec la couleur RGBA spécifiée.

drawCircle(int x, int y, int radius, int r, int g, int b, int a)

Dessine un cercle aux coordonnées (x,y) avec le rayon et la couleur RGBA spécifiés.

virtual std::string getTextInput(const std::string& prompt, const std::string& initialText = "") = 0;

Permet de rentrer des caractères dans une chaîne de caractères.



 [Accueil](#)

 [Démarrage](#)

 [Architecture](#)

 [Biblio. Graphiques](#)

 [Créer un jeu](#)

 [API](#)

☐ Mode sombre