

Arcade

Généré par Doxygen 1.9.1

Chapitre 1

Page principale

1.1 Introduction

Le projet Arcade est un projet de 2ème année de l'école Epitech. Il consiste à créer une bibliothèque dynamique qui permet de gérer des jeux et des librairies graphiques. Le but est de pouvoir changer de jeu et de librairie graphique à la volée. Dans cette documentation, vous trouverez les informations nécessaires pour développer une librairie graphique ou un jeu compatible avec l'Arcade.

1.2 Description

L'Arcade est composée de 3 parties toutes développées en C++:

- La librairie graphique, elle permet d'afficher les jeux.
- Les jeux, ils sont affichés par la librairie graphique.
- Le programme principal, il permet de charger les librairies graphiques et les jeux et de faire la correspondance entre les deux librairies.

Vous allez pouvoir développer vos propres librairies graphiques et vos propres jeux pour l'Arcade. Pour cela, vous devrez respecter les interfaces IGraphic et IGame ([Librairies graphiques](#) [Librairies de jeu](#)).

En dehors de ces interfaces, vous êtes libres de développer vos librairies graphiques et vos jeux comme vous le souhaitez. Bon développement !

1.3 Conclusion

Vous avez maintenant toutes les informations nécessaires pour développer une librairie graphique ou un jeu compatible avec l'Arcade.

Ce projet à été réalisé par:

- [Kylian Tranchet](#)
- [Simon](#)
- [Quentin Tennerel](#)

Nous remercions également les groupes SOURCILIÈRE et Arrrrrcade pour avoir créé avec nous la structure de ce projet. Ainsi que de nous avoir partager leurs libraries afin de tester notre projet.

SOURCILIÈRE:

- [Alban](#)
- [Landry](#)
- [Aubane](#)

Arrrrrcade:

- [Mathieu](#) (mathieul.robert@epitech.eu)
- [Thomas](#)
- [Théophile](#)

Chapitre 2

Interface IColor

2.1 Introduction

L'interface IColor permet de définir les couleurs du jeu afin que le jeu puisse communiquer avec la lib graphiques facilement. Toutes les couleurs doivent hériter de IColor.

2.2 Configurer une couleur

L'interface couleur est composée d'un constructeur et de 4 getters:

- Le constructeur doit prendre 4 paramètres r (pour le niveau de rouge), g (pour le niveau de vert), b (pour le niveau de bleu), a (pour la transparence).
- **getR** doit retourner la valeur de r.
- **getG** doit retourner la valeur de g.
- **getB** doit retourner la valeur de b.
- **getA** doit retourner la valeur de a.

Chapitre 3

Interface IEntity

3.1 Introduction

L'Interface IEntity permet de définir les entités du jeu afin que le jeu puisse communiquer avec la lib graphiques facilement. Toutes les entités doivent hériter de IEntity.

3.2 Configurer une entité

Pour configurer une entité, vous devez implémenter les méthodes suivantes:

- **setPos**, cette méthode doit définir la position de l'entité (le système de position correspond à un système de cadrillage chaque case fait 29x29 pixels, vous devez renseigner la position en case, par exemple si on set la position a 2, 10 l'entité s'affichera a 58x290 pixels).
- **setSize**, cette méthode doit définir la taille de l'entité en pixels.
- **setChar**, cette méthode doit définir le caractère de l'entité (utiliser en cas de problèmes avec l'asset ou pour les bibliothèques graphiques ne gérant pas les images).
- **setColor**, cette méthode doit définir la couleur de l'entité (on utilise l'interface IColor [Interface IColor](#)).
- **setPath**, cette méthode doit définir le chemin de l'asset de l'entité.
- **setRotation**, cette méthode doit définir la rotation de l'entité.
- **getPos**, cette méthode doit retourner la position de l'entité.
- **getSize**, cette méthode doit retourner la taille de l'entité.
- **getChar**, cette méthode doit retourner le caractère de l'entité.
- **getColor**, cette méthode doit retourner la couleur de l'entité.
- **getPath**, cette méthode doit retourner le chemin de l'asset de l'entité.
- **getRotation**, cette méthode doit retourner la rotation de l'entité.

Chapitre 4

Librairies de jeu

4.1 IGame

Pour utiliser vos librairies de jeu dans l'Arcade, TOUTES vos librairies devront hériter de l'interface IGame. Vous pouvez récupérer l'interface dans le [repository de l'Arcade](#) dans le dossier Interfaces.

4.2 Développer une librairie de jeu

Vos librairies devront respecter l'interface IGraphic. Vous devrez donc implémenter les méthodes suivantes:

- **startGame**, cette méthode doit démarrer le jeu.
- **stopGame**, cette méthode doit arrêter le jeu.
- **getScore**, cette méthode doit retourner le score du joueur.
- **simulate**, cette méthode doit simuler un frame du jeu.
- **catchKeyEvent**, cette méthode doit gérer les évènements clavier.
- **catchMousePosition**, cette méthode doit gérer les évènements souris.
- **setUserName**, cette méthode doit définir le nom du joueur.
- **getUserName**, cette méthode doit retourner le nom du joueur.
- **getEntities**, cette méthode doit retourner les entités du jeu à afficher, voir [Interface IEntity](#) pour plus d'information sur les entités.
- **getTexts**, cette méthode doit retourner les textes du jeu à afficher, voir [Interface IText](#) pour plus d'information sur les textes.
- **getSounds**, cette méthode doit retourner les sons du jeu à afficher, voir [Interface ISound](#) pour plus d'information sur les sons.

Chapitre 5

Librairies graphiques

5.1 IGraphic

Pour utiliser vos librairies graphiques dans l'Arcade, TOUTES vos librairies devront hériter de l'interface IGraphic. Vous pouvez récupérer l'interface dans le [repository de l'Arcade](#).

5.2 Développer une librairie graphique

Vos librairies devront respecter l'interface IGraphic. Vous devrez donc implémenter les méthodes suivantes:

- **isWindowOpen**, cette méthode doit retourner un booléen indiquant si la fenêtre est ouverte ou non.
- **closeWindow**, cette méthode doit fermer la fenêtre.
- **clearWindow**, cette méthode doit effacer le contenu de la fenêtre.
- **getKeyEvent**, cette méthode doit retourner le code de la touche appuyée conformément à la documentation.
- **getMousePosition**, cette méthode doit retourner la position de la souris, via une pair d'int (x, y).
- **displayWindow**, cette méthode doit afficher la fenêtre.
- **displayEntities**, cette méthode doit afficher les entités passées en paramètre, voir [Interface IEntity](#) pour plus d'informations sur les entités.
- **displayText**, cette méthode doit afficher le texte passé en paramètre, [Interface IText](#) pour plus d'informations sur les textes.
- **playSound**, cette méthode doit jouer le son passé en paramètre, voir [Interface ISound](#) pour plus d'information sur les sons.

Chapitre 6

Interface ISound

6.1 Introduction

L'interface ISound permet de définir les sons du jeu afin que le jeu puisse communiquer avec la lib graphiques facilement. Tous les sons doivent hériter de ISound.

6.2 Configurer un son

Pour configurer un son, vous devez implémenter les méthodes suivantes:

- **setPathSound**, cette méthode doit définir le chemin du son.
- **setVolume**, cette méthode doit définir le volume du son.
- **setStatus**, cette méthode doit définir le status du son via une enum (start, loop, stop, done).
- **getPathSound**, cette méthode doit retourner le chemin du son.
- **getVolume**, cette méthode doit retourner le volume du son.
- **getStatus**, cette méthode doit retourner le status du son.

Chapitre 7

Interface IText

7.1 Introduction

L'interface IText permet de définir les textes du jeu afin que le jeu puisse communiquer avec la lib graphique facilement. Tous les textes doivent hériter de IText.

7.2 Configurer un texte

Pour configurer un texte, vous devez implémenter les méthodes suivantes:

- **setFontPath**, cette méthode doit définir le chemin de la police du texte.
- **setText**, cette méthode doit définir le texte à écrire.
- **setColor**, cette méthode doit définir la couleur du texte, conformément à l'interface IColor [Interface IColor](#).
- **setPos**, cette méthode doit définir la position du texte.
- **setSize**, cette méthode doit définir la taille du texte.
- **setRotation**, cette méthode doit définir la rotation du texte.
- **getFontPath**, cette méthode doit retourner le chemin de la police du texte.
- **getText**, cette méthode doit retourner le texte à écrire.
- **getColor**, cette méthode doit retourner la couleur du texte.
- **getPos**, cette méthode doit retourner la position du texte.
- **getSize**, cette méthode doit retourner la taille du texte.
- **getRotation**, cette méthode doit retourner la rotation du texte.

