



Arcade Documentation

Roman Lopez, Stephane Fievez, Alexandre De Freitas Martins

Mars / Avril 2023

0.1 Introduction

Bienvenue dans la documentation de notre projet Arcade, ici vous pourrez retrouver tous les éléments afin de créer votre propre librairie graphique ou jeu afin de l'intégrer dans notre projet.

0.1.1 L'Arcade c'est quoi ?

L'Arcade est une plate-forme de jeu, c'est un programme qui permet à l'utilisateur de choisir un jeu et qui tient un registre des scores des joueurs.

Afin de pouvoir traiter les éléments via notre plate-forme de jeu au moment de l'exécution, nos bibliothèques graphiques et nos jeux doivent être implémentées en tant que bibliothèques dynamiques, qui sont alors chargées au moment de l'exécution.

0.2 Classe Core

Le "Core" du programme est initialisé dans le main. Il permet le chargement de la librairie graphique indiqué par l'utilisateur ainsi que celui du menu.

```
namespace Arcade {
    class Core {
    public:
        ////////////////////////////////// Constructor //////////////////////////////////
        Core();
        Core(std::string lib);
        Core(std::string lib, std::string game);
        ~Core();
        ////////////////////////////////// Setter //////////////////////////////////
        void setGraphic(IDisplay *graphic);
        void setGame(IGame *game);
        void setMenu();
        void setChangeLib(EventType event);
        void setChangeGame(EventType event);
        ////////////////////////////////// Functions //////////////////////////////////
        void loop();
        void MenuGetChange();
        void changeLib(std::string lib);
        void changeGame(std::string game);
    private:
        size_t _indexLib;
        size_t _indexGame;
        MenuInfo _menuInfo;
        IGame *game;
        IDisplay *graphic;
        DLloader *gameDll;
        DLloader *graphicDll;
        std::vector<std::string> _libs;
        std::vector<std::string> _games;
    };
}
```

0.3 Implémentation de bibliothèques graphiques et de jeux

0.3.1 Interface de jeux

Pour que les jeux soient compatibles, ils doivent avoir l'interface suivante :

```
namespace Arcade {
    class IGame {
    public:
        virtual ~IGame() = default;
        //////////////// Functions ////////////////
        virtual void init() = 0;
        virtual void update(Arcade::EventType event) = 0;
        virtual void close() = 0;
        //////////////// Getters ////////////////
        virtual bool isRunning() const = 0;
        virtual const std::map<char, std::string> &getAssets() const = 0;
        virtual const std::vector<Drawable> &getDrawable() const = 0;
        virtual const std::vector<DrawableText> &getDrawableText() const = 0;
        virtual MenuInfo getMenuInfo(void) const = 0;
        //////////////// Setters ////////////////
        virtual void setMenuInfo(MenuInfo menuInfo) = 0;
        virtual void setIsRunning(bool isRunning) = 0;
    };
}
```

0.3.2 Implémenter un jeu

Pour pouvoir implémenter un jeu il a besoin d'un "entrypoint" ainsi qu'une fonction "getType" :

```
extern "C" {
    IGame *entryPoint()
    {
        return new Menu();
    }
    char *getType()
    {
        return (char *) "gameMenu";
    }
}
```

0.3.3 Lancement de jeux

Afin de lancer un jeu il faut :

- D'abord que la bibliothèque du jeu soit placée dans le répertoire `./lib/`.
- Puis lancer l'Arcade, le jeu sera alors disponible dans le menu.

0.3.4 Interface de librairie graphique

Pour que les librairies graphiques soient compatibles, elles doivent avoir l'interface suivante :

```
namespace Arcade {
    class IDisplay {
    public:
        virtual ~IDisplay() = default;
        //////////////// Functions ////////////////
        virtual void init(const std::map<char, std::string>& gameAssets) = 0;
        virtual void update() = 0;
        virtual void close() = 0;
        virtual void display(std::vector<Drawable> drawables) = 0;
        virtual void display(std::vector<DrawableText> drawables) = 0;
        virtual void clear() = 0;
        //////////////// Getters ////////////////
        virtual Arcade::EventType getEvent() = 0;
    };
}
```

0.3.5 Implémenter une librairie graphique

Pour pouvoir implémenter une librairie graphique elle a besoin d'un "entrypoint" ainsi qu'une fonction "getType" :

```
extern "C" {
    IGame *entryPoint()
    {
        return new SDL2();
    }
    char *getType()
    {
        return (char *) "libSDL2";
    }
}
```

0.3.6 Lancement d'une librairie graphique

Afin de lancer une librairie graphique il faut :

- D'abord que la librairie du jeu soit placée dans le répertoire "./lib/".
- Puis lancer l'Arcade, la librairie graphique sera disponible dans le menu.

Ou bien il est possible de la choisir au lancement en la sélectionnant à l'exécution :

```
./arcade ./lib/NomDeLaLib.so
```

0.4 Interface Partagées

Les Interfaces de ce projet ont été partagé avec les groupes suivants :

- gwenael.hubler@epitech.eu, ewan.bigotte@epitech.eu, nathan.donadey@epitech.eu
- hugo.houbert@epitech.eu, victor.delamonica@epitech.eu
- maxime.mallet@epitech.eu, allan.charlier@epitech.eu