
ARCADE Documentation



June, 2020

Project Description

Arcade is a gaming platform that lets user choose a game to play.

Games are implemented as dynamic libraries, loaded at runtime.

Project Implementation

C++ programming language.

Core

The purpose of the Core is to handle the main loop of the program and load the library that user choses. Core gets the events and uses them to the game.

We implemented Menu class to load libraries and change it if necessary.

Below you can see the Menu.hpp and the functions we implemented.

```

/*
** EPITECH PROJECT, 2020
** Arcade
** File description:
** Menu
*/

#ifndef MENU_HPP_
# define MENU_HPP_

# include <memory>
# include <array>
# include "../include/IGameLib.hpp"
# include "../include/IGraphicLib.hpp"

class Menu {
public:
    Menu(std::string &lib);
    ~Menu();

    void launch();
    void changeLib(IGraphicLib::Ikey key);
    void changeGame(IGraphicLib::Ikey key);
    void writeMenu();
    void openLib(std::string &lib);
    void openGame();
    void openDL();
    void Frame();
    void play_game();
    int react(IGraphicLib::Ikey key);

private:
    IGameLib::e_end _end;

    // Graphic libs
    std::shared_ptr<IGraphicLib> _lib;
    int _actualLib;
    void* _dLib;
    std::array<std::string, 3> _libName;

    // Game libs
    std::shared_ptr<IGameLib> _game;
    int _actualGame;
    void* _dGame;
    std::array<std::string, 3> _gameName;
};

#endif

```

Graphics/Lib

Lib directory contains graphic libraries. The interface we used to implement graphic libraries is IGraphicLib. Below you can see the IGraphicLib.hpp and the functions we implemented.

[illegible]

Games

Games directory contains games libraries. The interface we used to implement games libraries is IGameLib. Below you can see the IGameLib.hpp and the functions we implemented.

```
/*
** EPITECH PROJECT, 2020
** Arcade
** File description:
** IGameLib
**/

#ifndef IGAMELIB_HPP_
# define IGAMELIB_HPP_

#include "IGraphicLib.hpp"

class IGameLib
{
public:
    typedef enum {
        E_NONE,
        E_WIN,
        E_LOSE
    } e_end;

    virtual ~IGameLib() {};
    virtual void WindowOpen() = 0;
    virtual void CreateAssets(void) = 0;
    virtual void Movement(void) = 0;
    virtual void GameAction(void) = 0;
    virtual void DisplayGame() = 0;
    virtual void start_Game(void) = 0;
};

#endif /* !IGAMELIB_HPP_ */
```

Compile:

run one of commands shown below:

-make

-make re

Run program:

`./arcade [graphic_library]`

graphic_library:

1. `./lib/lib_arcade_sfml.so`
2. `./lib/lib_arcade_ncurses.so`

Below you can find an image of how the window will look like if you run:

`./arcade ./lib/lib_arcade_sfml.so`



- Press '2' or '3' key to change graphic library
- Press '4' or '5' key to change game
- Press 'ESC' key to exit

Group members

Besart Cerri (besart.cerri@epitech.eu)

Donjeta Vojvoda (donjeta.vojvoda@epitech.eu)

Florentina Vojvoda (florentina.vojvoda@epitech.eu)

June,
2020