# **ARCADE Documentation**



## **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.

```
#ifndef MENU HPP
# define MENU HPP
# include <memor
# include "../include/IGraphicLib.hpp
class Menu {
   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);
orivate:
    IGameLib::e end end;
    std::shared_ptr<IGraphicLib> _lib;
   int _actualLib;
   void* dLib;
    std::array<std::string, 3> _libName;
    std::shared ptr<IGameLib> game;
    int actualGame;
   void* dGame;
    std::array<std::string, 3> _gameName;
};
#endif
```

# Graphicals/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.

```
tifndef IGRAPHICLIB_HPP_
     typedef enum {
    I_RED,
    I_WHITE,
    I_YELLOW,
    I_BLUE,
    I_GREEN
     } Icolor:
     typedef enum {
   I_NONE = -1,
   I_LEFT = 0,
   I_ESC,
   None 1
     } Ikey;
     virtual ~IGraphicLib() {};
                                     open(const int &x, const int &y) = 0;
                                    refresh() = 0;
clear() = 0;
                                    close() = 0;
getKey() const = 0;
                                    writeSmth(const int &x,
                                      const int &y,
const std::string& text) = 0;
                                  build(const int &x,
                                 const int &y,
const Icolor &col) = 0;
};
```

#### Games

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

```
#ifndef IGAMELIB_HPP_
# define IGAMELIB HPP
#include "IGraphicLib.
public:
   typedef enum {
    } 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
```

#### **Compile:**

run one of commands shown below:

- -make
- -make re

#### Run program:

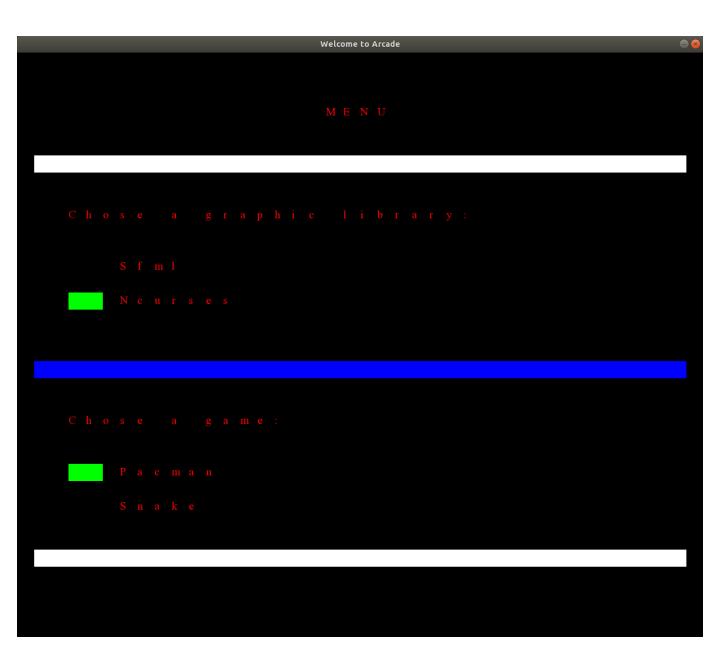
```
./arcade [graphic_library]
```

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)