

# Arcade

02/03/2020 - 05/04/2020

[alexandre.ohayon@epitech.eu](mailto:alexandre.ohayon@epitech.eu)

[julien.rondin@epitech.eu](mailto:julien.rondin@epitech.eu)

[erwan.sinou@epitech.eu](mailto:erwan.sinou@epitech.eu)

# Sommaire

## Interfaces

- [IGames](#)
- [IGraphics](#)

## Touches

- [Liste des touches globales](#)
- [Liste des touches en jeu](#)

## Lancer un jeu

## Ajouter une nouvelle librairie graphique

## Diagramme de classes du programme

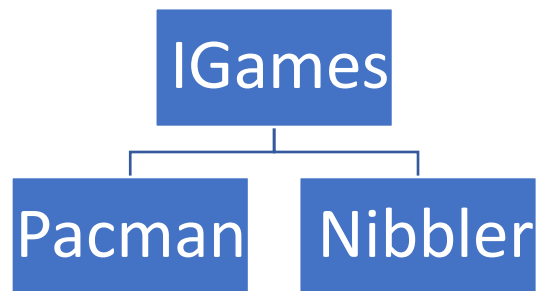
## Interfaces

IGames :

Nos deux jeux implémentent cette interface. Elle nous permet de rendre le projet générique, et ainsi d'avoir les mêmes fonctions pour chaque jeu.

```
class IGames {  
public:  
    virtual void initGame(void) = 0;  
    virtual bool processInput(int key) = 0;  
    virtual bool checkEnd(void) = 0;  
    virtual std::string const getName() = 0;  
    virtual std::vector<std::string> const &getMap(void) = 0;  
    virtual int getScore() = 0;  
};
```

Fonctions de l'interface IGames



## IGraphics :

Nos trois bibliothèques graphiques implémentent cette interface. Comme pour IGames, elle nous permet de rendre le projet générique, et ainsi d'avoir les mêmes fonctions pour chaque bibliothèque graphique.

```
class IGraphics {
public:
    virtual void initGraphic(Player &) = 0;

    virtual void showGame(std::vector<std::string> const &, int) = 0;
    // RETURN VALUE:
    //
    // -1 if player left the game
    //  0 if player goes back to menu
    // > 0 == player score

    virtual int showPause(void) = 0;
    // RETURN VALUE:
    //
    // RESUME if player continues the game
    // MENU   if player goes back to menu
    // EXIT   if player leaves the game

    virtual int showMenu(Player &) = 0;
    // RETURN VALUE:
    //
    // NIBBLER if player selects nibbler
    // PACMAN  if player selects pacman
    // STATS   if player selects stats
    // EXIT     if player leaves the game

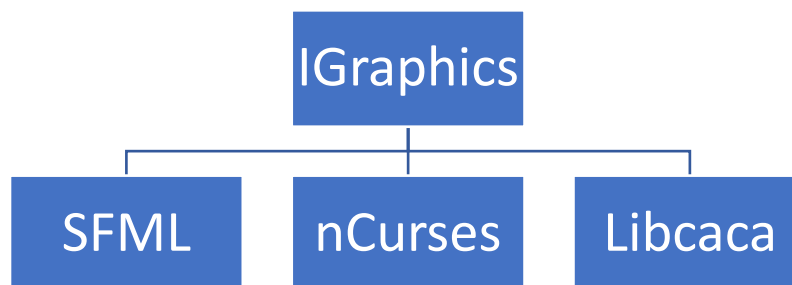
    virtual void showWin(int) = 0;
    virtual void showLose(int) = 0;

    virtual int getInput() = 0;

    virtual std::string getLibName(void) = 0;

    virtual void exitGraphics(void) = 0;
};
```

Fonctions de l'interface IGraphics



## Touches

Liste des touches globales :

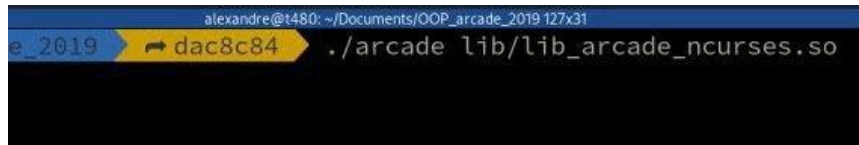
- Librairie précédente : **f**
- Librairie suivante : **g**
- Jeu précédent : **F3**
- Jeu suivant : **F4**
- Relancer jeu : **r**
- Retourner au menu : **m**
- Quitter : **e**

Liste des touches en jeu :

- Haut : **z**
- Bas : **s**
- Gauche : **q**
- Droite : **d**
- Pause : **Echap**

## Lancer un jeu

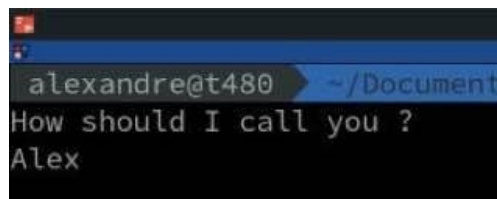
Pour lancer un jeu d'exécuter le binaire arcade avec la librairie graphique que l'on souhaite qui est située dans le dossier lib.



```
alexandre@t480: ~/Documents/OOP_arcade_2019 127x31
e_2019 ➡ dac8c84 ➡ ./arcade lib/lib_arcade_ncurses.so
```

Capture d'écran présentant l'exécution du programme

Le programme se lance et demande à l'utilisateur quel nom utiliser pour le joueur.



```
alexandre@t480 ➡ ~/Document
How should I call you ?
Alex
```

Capture d'écran présentant l'attribution du pseudonyme du joueur

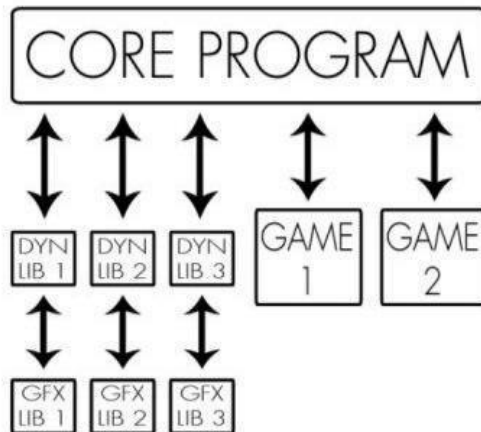
Le menu est lancé et nous avons le choix du jeu auquel jouer ou bien analyser nos scores précédents.



Capture d'écran présentant le menu d'Arcade en SFML

## Ajouter une nouvelle librairie graphique

La nouvelle librairie graphique doit forcément implémenter IGraphics. Une fois les sources mises dans le Makefile, il suffit de lancer le jeu avec cette nouvelle librairie. Si toutes les fonctions sont correctement implémentées, arcade marchera parfaitement.



Capture d'écran présentant la modularité du programme Arcade

## Diagramme de classes du programme

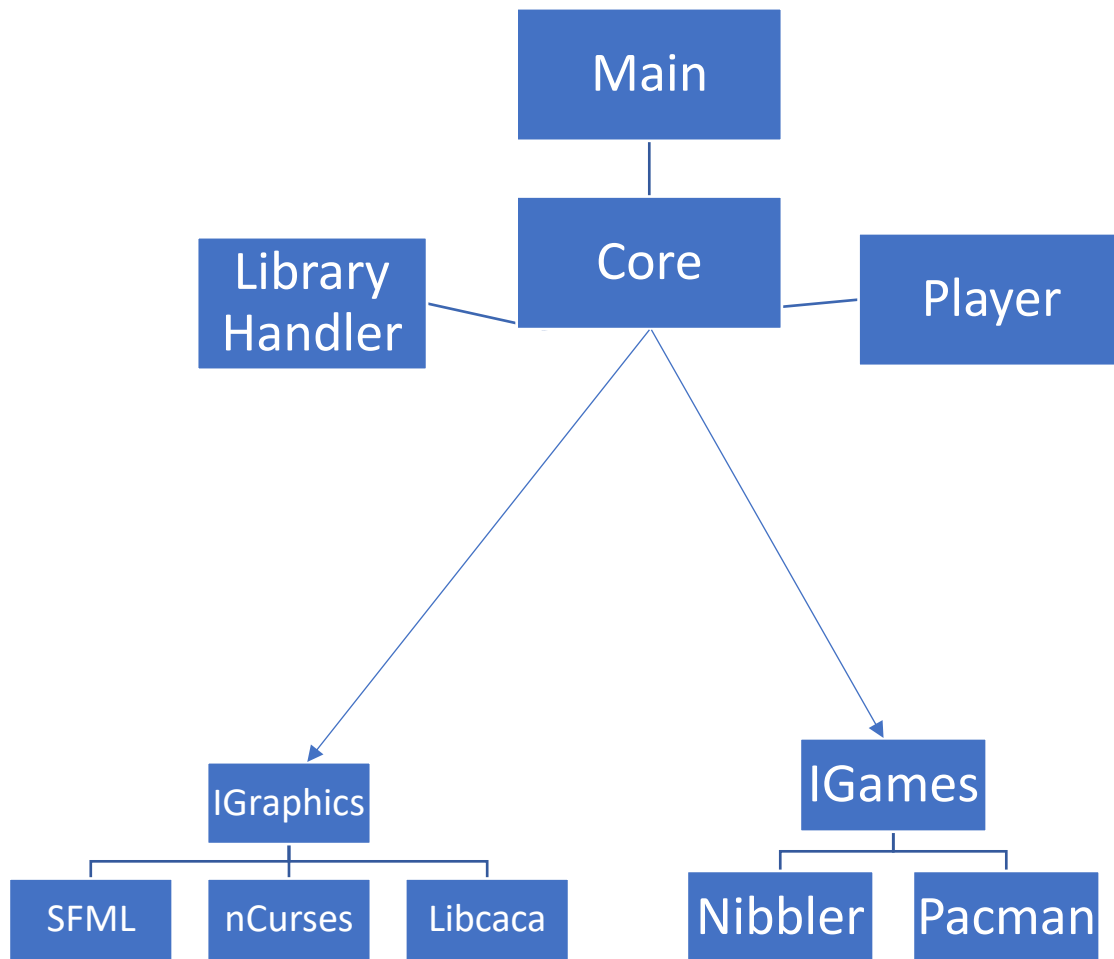


Diagramme présentant le fonctionnement du programme