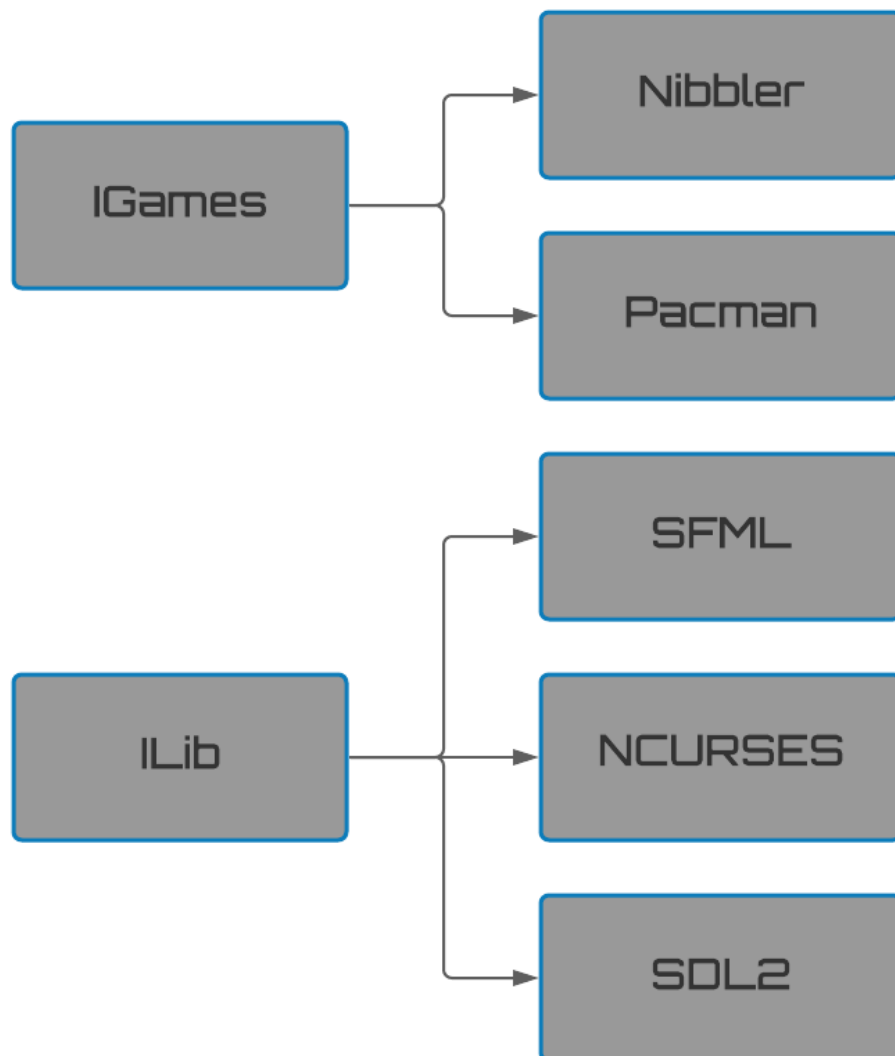




Documentation Arcade

Sébastien Perpina, Alexis Lalande, Mattis Litot





Les Interfaces:

IGame (Librairie de Jeu)

```
virtual ~IGame() = default;
virtual void init() = 0;
virtual void stop() = 0;

virtual std::vector<std::vector<char>> play() = 0;
virtual std::vector<std::vector<char>> getField() = 0;
virtual std::deque<std::pair<std::size_t, std::size_t>>
getSnakeCoord() = 0;
virtual std::size_t getSizeSnake() = 0;
virtual int make_move() = 0;

virtual void set_direction(Directions dir) = 0;
virtual Directions getDir() = 0;
virtual void add_random_fruit() = 0;
virtual void setPathMap(std::string path) = 0;
virtual std::string getPathMap() = 0;
virtual const std::string &getName() const = 0;
```

ILib (Librairie Graphique Sfml, Sdl, Ncurses)

```
virtual ~ILib() = default;
virtual void init() = 0;
virtual void stop() = 0;

virtual void createWinMenu() = 0;
virtual int menuLoop() = 0;
virtual void setAsciiGameName(const std::string nameGame) = 0;

virtual const std::string &getName() const = 0;
virtual int getLibChoice() const = 0;
virtual int getGameChoice() const = 0;

virtual void initPac(std::vector<std::vector<char>> field) = 0;
virtual void initGame(std::vector<std::vector<char>> field) = 0;
```



```
virtual void show(std::vector<std::vector<char>> field) = 0;
virtual void showSnake(std::deque<std::pair<std::size_t, std::size_t>>
coordinate, Direction dir, std::size_t size,
std::vector<std::vector<char>> updateMap) = 0;
virtual void showPacman(std::deque<std::pair<std::size_t, std::size_t>>
coordinates, Direction dir, std::size_t size,
std::vector<std::vector<char>> updateMap) = 0;
virtual void showMapPac(std::vector<std::vector<char>> field) = 0;

virtual Direction getKey() = 0;

virtual void closeGame() = 0;
virtual void endgame(std::size_t score) = 0;
```

Nibbler (Jeu)

```
~Snake() = default;

Field* field;
std::size_t size;
std::deque<std::pair<std::size_t, std::size_t>> coordinates;
Directions next_dir;
std::vector<std::vector<char>> updateMap;

std::vector<std::vector<char>> getField();
std::size_t getSizeSnake();
std::deque<std::pair<std::size_t, std::size_t>> getSnakeCoord();

std::vector<std::vector<char>> play();
void eat_fruit(std::size_t y, std::size_t x);

void increase_size();
void set_direction(Directions dir);
Directions getDir();
```



PacMan (Jeu)

```
~PacMan() = default;

Field* field;
std::size_t size;
std::deque<std::pair<std::size_t, std::size_t>> coordinates;
Directions next_dir;
std::vector<std::vector<char>> updateMap;

std::vector<std::vector<char>> getField();
std::deque<std::pair<std::size_t, std::size_t>> getPacManCoord();

std::vector<std::vector<char>> play();
void eat_fruit(std::size_t y, std::size_t x);

void increase_size();
void set_direction(Directions dir);
Directions getDir();
```

Les interfaces IGame et ILib sont respectivement utilisées comme interface pour les librairies de jeux et pour les librairies graphiques.

Les 2 jeux sont basés sur le même fonctionnement et partagent donc les mêmes fonctions. De plus les 3 librairies graphiques utilisent toutes les mêmes fonction de l'interface ILib, permettant de lancer en temps réel n'importe quel jeu à partir d'une de nos 3 librairies.



Ajout d'une librairie graphique

- Installer les dépendances de la librairie voulu sur son système.
- Créer un dossier, qui contiendra les fichiers de votre librairie dans le dossier .lib.
- Créer un fichier .cpp contenant toutes les fonctions incluses dans l'interface ILib ainsi qu'un EntryPoint afin que le Core puisse accéder à vos fonctions.
- Ajouter au Makefile, une ligne de compilation incluant votre librairie ainsi que les flags “-shared -fPIC” en plus des flags nécessaires à votre librairie.

Ajout d'un Jeu

- Créer un dossier, qui contiendra les fichiers de votre Jeu dans le dossier .lib/Game.
- Créer un fichier .cpp, votre jeu doit partager toutes les fonctions contenu dans l'interface IGame, lui permettant d'être compatible avec les librairies graphique.
- Ajouter au Makefile, une ligne de compilation incluant votre jeu ainsi que les flags “-shared -fPIC” en plus des flags nécessaires à votre jeu.