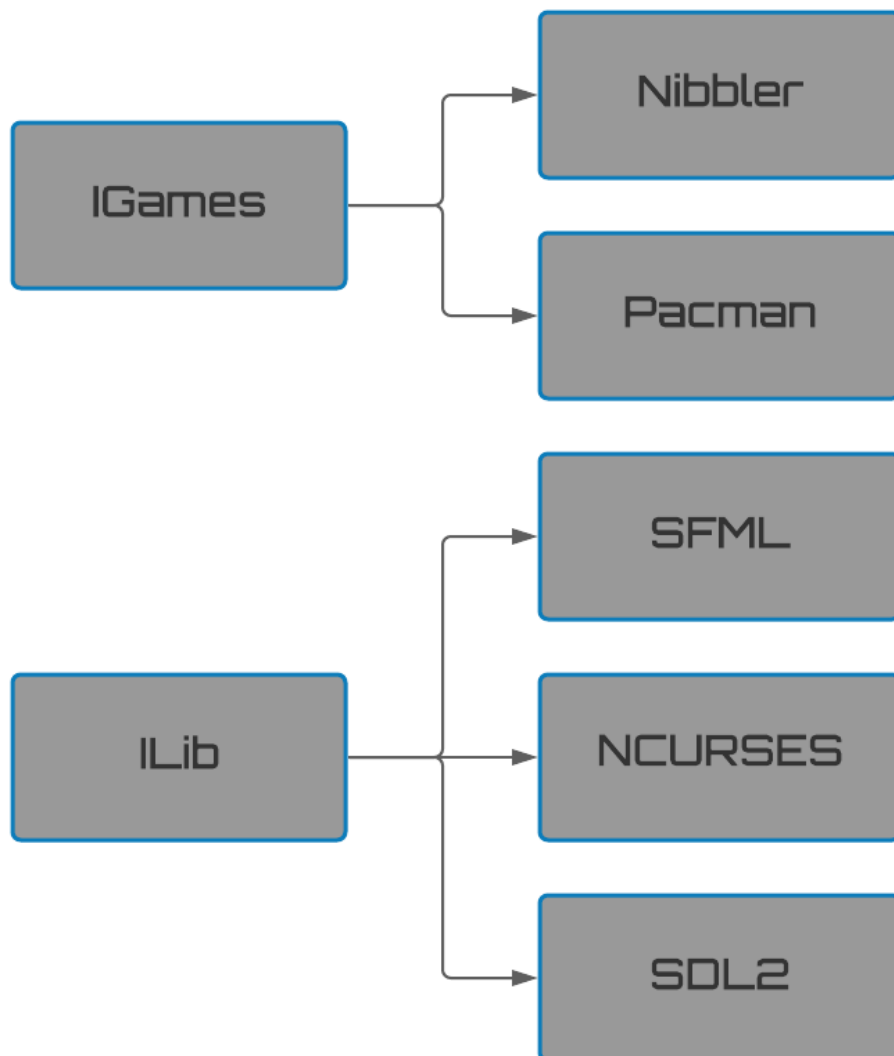




Arcade Documentation

Sébastien Perpina, Alexis Lalande, Mattis Litot





Interfaces:

IGame (Game Library)

```
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 (Graphic Library 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 (Game)

```
~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 I

```
~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();
```

The IGame and ILib interfaces are respectively used as an interface for the game libraries and for the graphics libraries.

The 2 games are based on the same operation and thus share the same functions. Moreover the 3 graphic libraries use all the same functions of the ILib interface, allowing to launch in real time any game from one of our 3 libraries.



Adding a graphic library

- Install the desired library dependencies on your system.
- Create a folder, which will contain the files of your library in the .lib folder.
- Create a .cpp file containing all the functions included in the interface. ILib and an EntryPoint so that the Core can access your functions.
- Add to the Makefile, a compilation line including your library and the "-shared -fPIC" flags in addition to the flags necessary for your library.

Adding a game

- Create a folder, which will contain your game files in the .lib/Game folder.
- Create a .cpp file, your game must share all the functions contained in the IGame interface, allowing it to be compatible with the graphic libraries.
- Add to the Makefile, a compilation line including your game and the "-shared -fPIC" flags in addition to the flags necessary for your game.