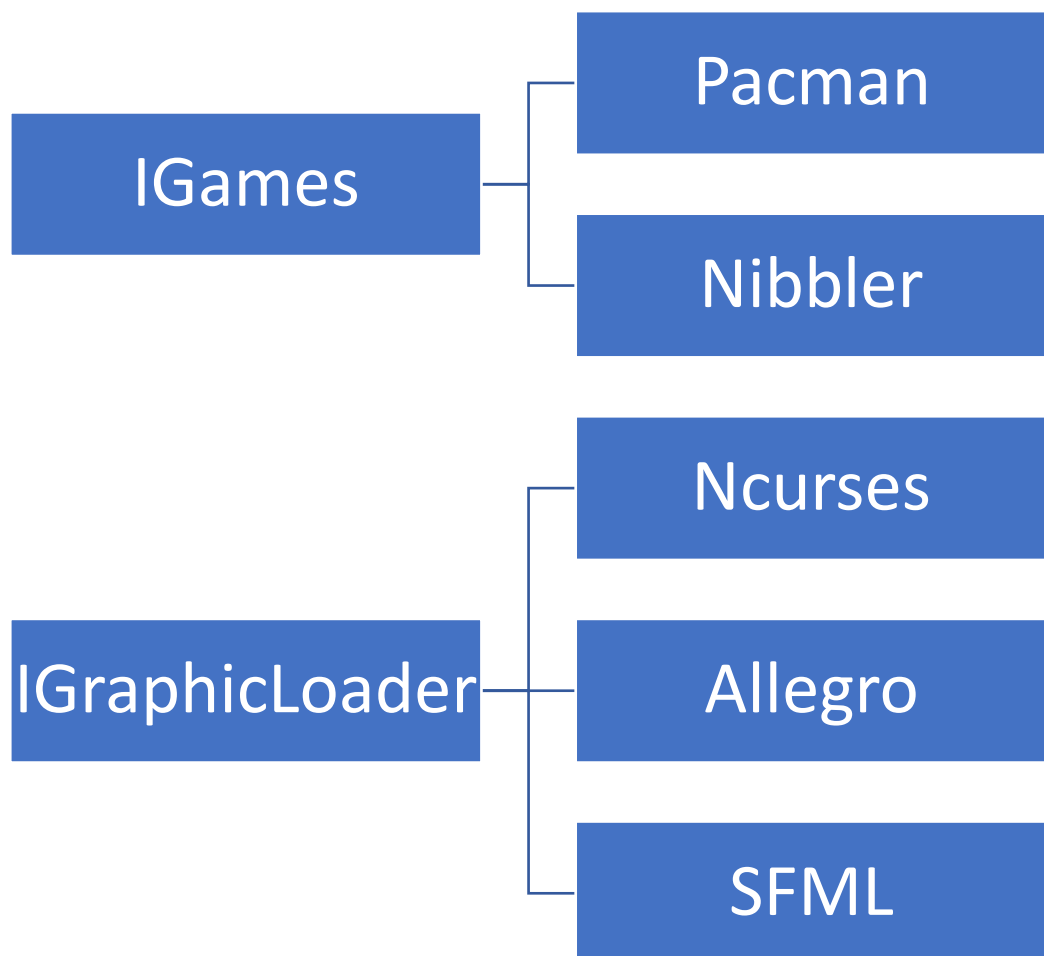


Documentation de l'Arcade

Epitech 2019

Hugo sohm, Enzo lenzi, Ryan chamayrac



LES DIFFERENTES FONCTIONS & VARIABLES

IGRAPHICLOADER (LIBRAIRE GRAPHIQUE)

```
virtual ~IGraphicLoader()
virtual void drawASquare(float, float, float, float, unsigned int)
virtual void drawASquare(float, float, float, float, std::string)
virtual void drawASquare(float, float, float, float, std::string, unsigned int, bool = true, bool = false)
virtual int getKeyPress()
virtual void clearThisSquare(float, float, char)
virtual void clear()
virtual void display()
virtual void clearAll()
virtual void setATextField(float x, float y, std::string fontPath = "font/BebasNeue-Bold.ttf")
virtual void drawATextField(std::string &textToDraw)
virtual void drawAText(float, float, std::string)
virtual void initSprite(std::string, int, int, int)
virtual void execThisSprite(float sizeX, float sizeY, float x, float y, std::string name, unsigned int color,
bool dinamic, bool bold)
virtual void drawThisSprite(float sizeX, float sizeY, float x, float y, std::string name, unsigned int color,
bool dinamic, bool bold)
```

IGAMES (LIBRAIRIE DE JEU (SFML | ALLEGRO | NCURSES))

```
virtual ~IGames()
virtual int run(IGraphicLoader *, bool)
virtual void DrawMap(IGraphicLoader*, bool)
virtual int checkChar(IGraphicLoader *)
virtual void setCh(int)
virtual void initGame(IGraphicLoader *)
```

NIBBLER (JEU)

```
Nibbler()
~Nibbler()
int run(IGraphicLoader *lib, bool)
void DrawMap(IGraphicLoader *lib, bool)
int checkChar(IGraphicLoader *lib)
void setCh(int i)
void initGame(IGraphicLoader *lib)
std::map<int, std::string> pacmap
std::map<int, std::string>::iterator it = pacmap.begin()
std::string line
```

PACMAN (JEU)

Pacman()

~Pacman()

*int run(IGraphicLoader *lib, bool)*

*void DrawMap(IGraphicLoader *lib, bool)*

*int checkChar(IGraphicLoader *lib)*

void setCh(int i)

*void initGame(IGraphicLoader *lib)*

std::map<int, std::string> pacmap

std::map<int, std::string>::iterator it = pacmap.begin()

std::string line

std::string _content

LES INTERFACES

IGraphicLoader est une interface utilisée pour les librairies graphiques.

IGames est une interface utilisée pour les librairies de jeux.

Les 3 librairies graphiques utilisent exactement les mêmes fonctions permettant ainsi à n'importe quel jeu de se lancer dans n'importe quelle librairie et ceux même durant l'exécution du jeu.

AJOUTER UNE LIBRAIRIE GRAPHIQUE

Afin d'ajouter une librairie graphique, il faut installer les dépendances sur son système et vérifier toute comptabilité afin de par la suite, ajouter ses fonctions.

Pour cela, il faut créer un nouveau dossier avec le nom de sa librairie dans GfxSrc comportant un fichier .hpp avec toutes les fonctions de l'interface graphique ainsi que ses variables.

De plus, un fichier .cpp est requis avec toutes les fonctions de l'interface graphique permettant d'exécuter les jeux avec chacune des fonctions présente dans la nouvelle interface.

Pour finir, un Makefile est nécessaire afin de créer une librairie dynamique en .so qui se retrouvera dans le dossier lib à la racine du projet.

AJOUTER UN JEU

Ajouter d'ajouter un jeu, il suffit de conceptualiser son jeu et de la réaliser à l'aide de l'interface graphique pour promettre une compatibilité avec n'importe quelle librairie graphique.

Pour cela, il faut créer un nouveau dossier avec le nom de son jeu dans GameSrc comportant un fichier .hpp avec toutes les fonctions de l'interface de jeu ainsi que ses variables.

De plus, un fichier .cpp est requis avec toutes les fonctions de l'interface de jeu permettant d'exécuter le jeu avec n'importe quelle librairie graphique.

Pour finir, un Makefile est nécessaire afin de créer une librairie dynamique en .so qui se retrouvera dans le dossier games à la racine du projet.