DOCUMENTATION ARCADE

**IGameModel :**

virtual void setMap() = 0;  
virtual void drawMap(ILibraryViewController \*libraryInstance) = 0;  
virtual ChangeCommandType play(ILibraryViewController \*library, size\_t &currentGame, size\_t &currentLibrary, bool &) = 0;  
virtual int getScore() = 0;  
virtual void setScore(int) = 0;  
virtual std::string getGameName() = 0;  
virtual void playProtocol(void) = 0;

**Snake :**

Snake(std::string const &libname);  
Snake(Snake const &);  
Snake &operator=(Snake const &);  
virtual ~Snake();  
virtual void setMap();  
virtual void drawMap(ILibraryViewController \*libraryInstance);  
virtual ChangeCommandType play(ILibraryViewController \*library, size\_t &currentGame, size\_t &currentLibrary, bool &exit, std::string const &playerName);  
virtual int getScore();  
virtual void setScore(int);  
virtual std::string getGameName();  
virtual void playProtocol(void);

**SolarFox :**

SolarFox(std::string const &libname);  
SolarFox(SolarFox const &);  
SolarFox &operator=(SolarFox const &);  
virtual ~SolarFox();  
virtual void setMap();  
virtual void drawMap(ILibraryViewController \*libraryInstance);  
virtual ChangeCommandType play(ILibraryViewController \*library, size\_t &currentGame, size\_t &currentLibrary, bool &exit, std::string const &playerName);  
virtual int getScore();  
virtual void setScore(int);  
virtual std::string getGameName();  
virtual void playProtocol(void);

**SFML / Ncurses / OpenGL :**

virtual void drawSquare(int width, int x, int y, Color const &);  
virtual void drawMenu(size\_t &currentGame, std::vector<std::string> const &games, bool &exit, size\_t &currentLibrary, ChangeCommandType &action, std::string &playerName);  
virtual bool getKey(arcade::CommandType \*commandType, ChangeCommandType &action, bool &exit);  
virtual void initScreen(std::string const &name, std::string const &playerName);  
virtual void displayScore(int width, std::string const &, std::string const &, int);  
virtual void endScreen();  
virtual void refresh();  
virtual void clear();  
virtual void gameOver(int);  
virtual void win(int);  
virtual std::string getLibraryName() ;

virtual void displayPlayerName() ;

Le ILibraryViewController est une interface utilisée pour les librairies graphiques.  
Le IGameModel est une interface utilisée pour les librairies de jeux.

Chaque libraries graphiques utilisent les mêmes fonctions permettant ainsi à n’importe quel jeu de se lancer dans n’importe quel librarie.  
Exemple : la fonction DrawSquare permet de d’afficher un carré à l’écran. En appelant cette fonction dans le jeu, le carré s’affichera en SFML, Ncurses et en OpenGL.

Il vous faudras pour créer un nouvelle librairie graphique ou librairie de jeu la compiler en .so (Librairie dynamique) et la mettre dans le dossier lib (pour une librairie graphique) ou games (pour une librairie de jeu) a la racine du projet, ainsi le GameCore se chargeras de récupérer la librairie et la lancera.