DOCUMENTATION ARCADE

- Samuel Palmer
- Mathieu Blais
- Aimé Bertrand

Epitech Nancy 2021

Helper :

./arcade [lib]

example:

- ./arcade lib/arcade_ncurses.so
- ./arcade lib/arcade_sdl2.so
- ./arcade lib/arcade_sfml.so

- A : previous graphics library
- Z : next graphics library
- Q : previous game
- S : next game
- R : restart the game
- M : back to the menu
- R : restart
- SPACE : change letters
- ESC : quit

Interface Classes:

- IDisplayModule :I IDisplayModule :IDisplayModule is a class used by all the graphical libraries. It's only purpose is to display the information organised in a grid like fashion.

Methods :
virtual void draw_sprite() : draw the grid

virtual void initMap(std::map<int, std::vector<int>> map) : initialize the graphical grid with the game's gird

virtual void refreshMap(std::map<int, std::vector<int>> map) : refresh the grid with updated values

virtual std::string getKey() : retrieves user input from the graphical interface

- IGameModule :IgameModule is the class used by the games. It manages all game functions and updates the game's grid.

Methods :
virtual int endGame() : checks if the game has ended

virtual void initMap() : initialize the grid with the text file

virtual void getEntityEvent() : refreshes the games main loop

virtual void manageEvents(const std::string &key) : sends user input events to the game

virtual std::map<int, std::vector<int>> getMap() const : : getter for game's grid

protected:
private:

how to implement new graphics libraries :

To do so, you will need to create a class that inherits from the IDisplayModule class. Once you've done this step, you will have initialize the object of your own class by using extern "C" method.
Once you've done these previous step, all you will need to do is compile the source code to make the dynamic library and move the .so file into the lib directory.

how to implement new grames libraries :

you can follow the exact same step using IGameModule