ARCADE DOC

How to compile ?

Our Makefile has the following rules (including *all, clean, fclean, re*):

- core: it build the core of the program (not the games nor the graphical librairies)
- *games*: it build games librairies
- *graphicals*: it build graphical librairies

All rules build core, games and graphicals at the same time.

The core build an executable that is found in: *./arcade*

The games build libraries that are found in: *./games/*

The graphicals build libraries that are found in: *./lib/*

How to launch Arcade :
Compile then : *./arcade ./lib/lib_arcade_{name of the graphical lib you want to start}*.so It launch the menu.

Key Config:

the key are :

Quit: esc
Restart: r

Nextlib: n

Nextgame: z

Prevlib: p

Prevgame: a

Left:KEY_LEFT

Right:KEY_RIGHT

Up:KEY_UP

Down:KEY_DOWN

Describe of the action:

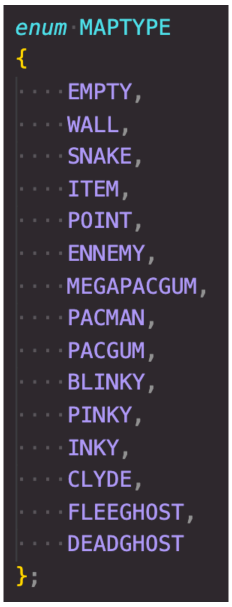
- - Quit: It quit the game.
- - Restart: It restart your game.
- - Nextlib: It switch with the next graphical library.
- - Prevlib: It switch with the previous graphical library.
- - Nextgame: It switch with the next game library.
- - Prevgame: It switch with the previous game library.
- - Left: Move the player to the left cell.
- - Right: Move the player to the right cell.
- - Up: Move the player to the up cell.
- - Down: Move the player to the down cell.

Score gestion:

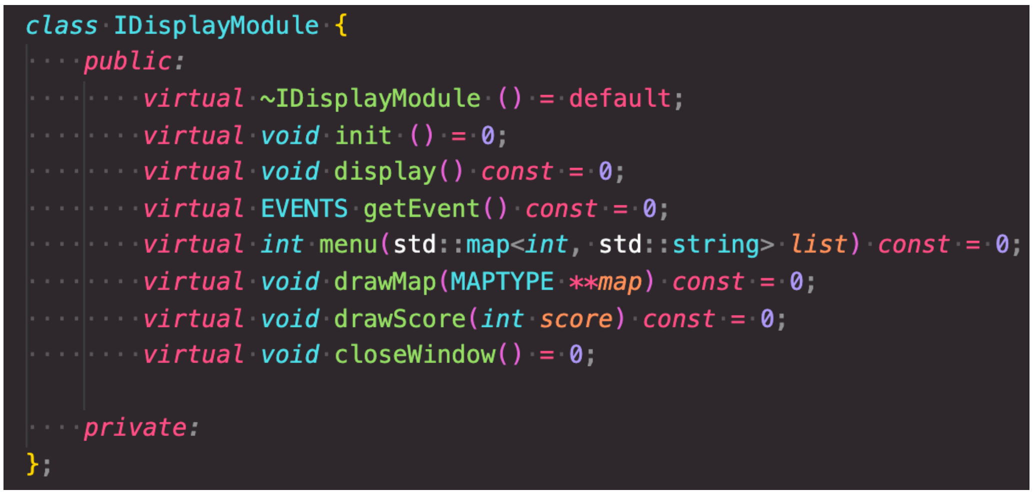
Each game has a file named ./highscore_{name of the game}.txt
The file is composed like that: {scorePoint}\n{pseudo}.

How to integrate a graphical library:

You have to heritate from IDisplayModule class.



```
enum MAPTYPE
{
    EMPTY,
    WALL,
    SNAKE,
    ITEM,
    POINT,
    ENNEMY,
    MEGAPACGUM,
    PACMAN,
    PACGUM,
    BLINKY,
    PINKY,
    INKY,
    CLYDE,
    FLEEGHOST,
    DEADGHOST
};
```



```
class IDisplayModule {
    public:
        virtual ~IDisplayModule () = default;
        virtual void init () = 0;
        virtual void display() const = 0;
        virtual EVENTS getEvent() const = 0;
        virtual int menu(std::map<int, std::string> list) const = 0;
        virtual void drawMap(MAPTYPE **map) const = 0;
        virtual void drawScore(int score) const = 0;
        virtual void closeWindow() = 0;

    private:
};
```

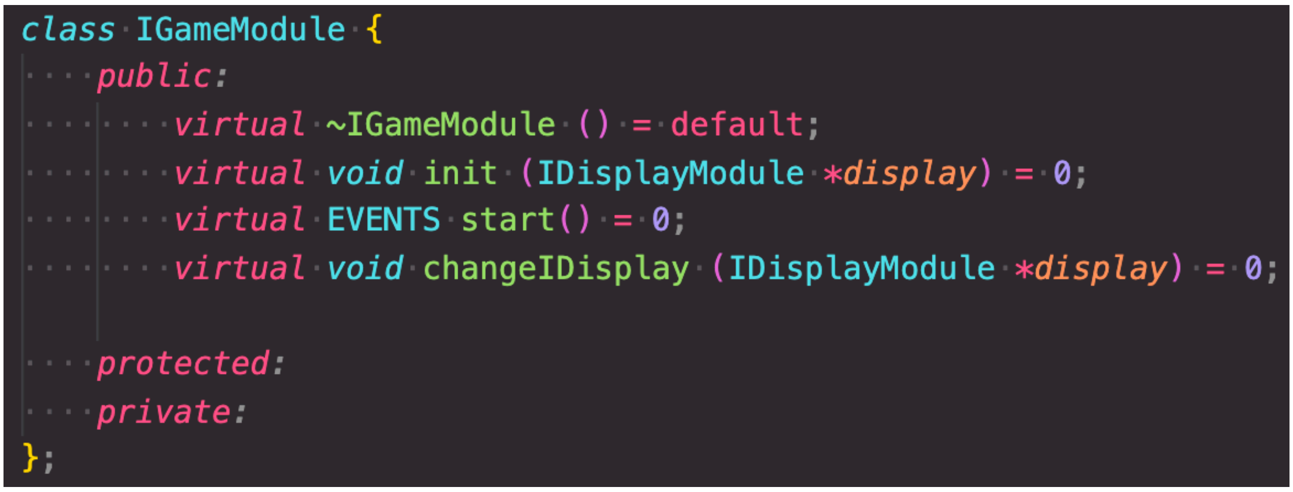
Following these methods:

- - Init() : Initialize the library (create a window for example).
- - display() : refresh the window's screen to get the changes made during the game.
- - getEvent() : When the library get an event, it catch it and return the value of enum Key.
- - menu() : Display the Menu of the lib.
- - drawMap(MAPTYPE **map) : display the map. The map is a 2 dimensional Enum tab (Enum of Enum) where each cell has a Enum like Wall or ENNEMY.
- - drawScore() : Displays the score of the game launched.

- - closeWindow() : stop the library (close the window for example).

How to integrate a game :

You have to heritate from IGameModule class.



```
class IGameModule {
    public:
        virtual ~IGameModule () = default;
        virtual void init (IDisplayModule *display) = 0;
        virtual EVENTS start() = 0;
        virtual void changeIDisplay (IDisplayModule *display) = 0;

    protected:
    private:
};
```

Following these methods :

- - Init(IDisplayModule*display) :initialize the game environment.
- - start() : the start function called by the play loop is used to start the play loop. If an event is detected from the graphics library, our function returns the detected event.
- - changeIDisplay(IDisplayModule*display): This function allows you to change the graphics library.