The IGames class :

The Igames class is the basis of any game that you want to create. It utilizes 3 main functions and 1 enumeration named “Event”.

- Key Event *[void key_event (int key)]* :

This void function is designed to take into account inputs from the user and send them to the game using the enumeration “Event”. It takes as a parameter from the graphics library of your choice the inputs and lets you interact with your game depending on all the available events (see “Event” enumeration).

- Main Loop *[int loop(int deltatime)]* :

This function is for the main loop of your game. It should return 0 as long as you wish the game to continue and 1 in case of game over or error. It takes as input the time as an int starting from 0 at for the first call of the loop function.

- Game Getter *[Games GetGame()]* :

This function is to send all the required data to the graphics library under the form of a Game object. It returns an object from the Game class containing player data, a map of your game, etc... (see “Game class”)

- Event enumeration :

This enumeration allows your game to receive inputs through the Key Event function (see above). It has 13 values (int) ranging from 0 to 12. They correspond accordingly to the following :

0 - LEFT
1 - RIGHT
2 - UP
3 - DOWN
4 - ENTER
5 - SHOOT
6 - PAUSE
7 - RESTART
8 - QUIT
9 - PREV_GRAPH
10 - NEXT_GRAPH
11 - PREV_GAME
12 - NEXT_GAME

The Igraphiclib class :

It contains 7 functions usually used for making a graphic library :

- Void init(int x, int y)

  The init function initializes the screen with its proportions in arguments. It initializes the window as well when needed.

- Event Keypressed()

  The Keypressed takes the input key of the player and returns one of the indexes of the "Event" enumeration. Basically it provides us with a universally translated way to get a input.

- Void refresh(Games)

  Refresh does exactly what it says. It clears the window and displays it again with new values. It demands to have access to the Games class because it contains all the information.

- Void clear()

  Clear empties the window. Nothing else.

- Void destroy()

  Destroy does the opposite of init. It destroys the window. It's the last function to do before ending the program.

- Void endgame()

  Endgame displays the end game screen with the score on display.

- Void assign_game(Games)

  Assign_game changes the Games member of the graphic library. Since its not making any modifications to it, this function must be called in pair with the Igames Games member in order to actualise it.

The Game class :

Is a class that contains all the information needed by the game. It contains the Event enumeration we have talked about earlier.

- Std::string name

  The name of the game.

- Int posx, int posy and int playerdirx, int playerdiry

  The position on the x axis and position on the y axis of the player as well as their direction on the x axis and y axis (playerdirx and playerdir y).

- Int Mat[height][width]

  The matrix contains the area of the game which is really important for the display. The Igames fonctions modify it quite a lot.

The Game class also contains the size of the area in which the game is going to be played in, its width and height.

- Int Score

  And finally, it contains the score which is incremented during the game.