UI_library.c

In order to update the screen with the correct images, a set of auxiliary functions are available in the files `UI_library.c` and `UI_library.h` .

The list of functions available for the development of the project is the following:

- **`int create_board_window(int dim_x, int dim_y);`**
  - This function initializes the graphical environment and creates a window representin a board with dim_x by dim_y places.

- **`void get_board_place(int mouse_x, int mouse_y, int * board_x, int *board_y);`**
  - When a user clicks on the windows or moves the mouse the SDL functions return the cursor position in the window. This function transforms that pixel position in the window (mouse_x and mouse_y) into the coordinate of a place in the board (the arguments board_x and board_y are passed to the function as references).

- **`void close_board_windows();`**
  - This function closes the board windows before the games terminates

- **`void paint_pacman(int board_x, int board_y , int r, int g, int b);`**
  - This function prints a pacman on the defined coordinates (board_x, board_y) with a certain color (r, g and b).

- **`void paint_powerpacman(int board_x, int board_y , int r, int g, int b);`**
  - This function prints a superpowered pacman on the defined coordinates (board_x, board_y) with a certain color (r, g and b).

- **`void paint_monster(int board_x, int board_y , int r, int g, int b);`**
  - This function prints a monster on the defined coordinates (board_x, board_y) with a certain color (r, g and b).

- **`void paint_lemon(int board_x, int board_y );`**
  - This function prints a lemon on the defined coordinates (board_x, board_y).

- **`void paint_cherry(int board_x, int board_y);`**
  - This function prints a cherry on the defined coordinates (board_x, board_y).

- **`void paint_brick(int board_x, int board_y );`**
  - This function prints a brick on the defined coordinates (board_x, board_y).

- **`void clear_place(int board_x, int board_y);`**
  - This function clears a place (paints it white).

These functions can only be called from the main thread!!!!.

Libraries installation

In the project, student will use the SDL2 library to program the graphical part.
It is necessary to install the SDL2 and SDL2_image libraries

- https://wiki.libsdl.org/Installation

Linux
In Linux (or WSL) it will be necessary to use the correct package manager:

- Ubuntu: `sudo apt-get install libsdl2-dev`
- Ubuntu: `apt-get install libsdl2-image-dev`
- OpenSUSE: `sudo zypper install SDL2-devel`
- OpenSUSE: `sudo zypper install SDL2_image-devel`

More information can be retrieved in the following pages:

- https://lazyfoo.net/tutorials/SDL/01_hello_SDL/linux/index.php
- https://lazyfoo.net/tutorials/SDL/06_extension_libraries_and_loading_other_image_formats/linux/index.php

MacOS X with XCode
In MacOS X you should follow the next tutorial:

- http://lazyfoo.net/tutorials/SDL/01_hello_SDL/mac/index.php
- http://lazyfoo.net/tutorials/SDL/06_extension_libraries_and_loading_other_image_formats/mac/index.php
-

MacOS X with gcc in terminal
In MacOS X you should follow the next tutorial:

- http://lazyfoo.net/tutorials/SDL/01_hello_SDL/mac/index.php
- http://lazyfoo.net/tutorials/SDL/06_extension_libraries_and_loading_other_image_formats/mac/index.php
- https://www.youtube.com/watch?v=BCcvwoIOHVI

Program compilation

The project Serve and client should be compiled with the UI_library.c and the `SDL2`, `SDL2_image` and `pthread` libraries.

Compile each one of the applications using the gcc:

- `gcc server.c -o app1 UI_library.c -lSDL2 -lSDL2_image -lpthread`
- `gcc client.c -o app2 UI_library.c -lSDL2 -lSDL2_image -lpthread`

if the compiler gives the following error:

- `…bin/ld: cannot find -lSDL2`
- `…bin/ld: cannot find -lSDL2_image`

It is necessary to install the SDL2 / SDL2_image libraries.