Arcade

Implementing a new graphics library

A graphics library class inherits from the IDisplayModule interface. You can find this interface in the *DisplayModule.hpp* header located in *src/interface*.
To implement your graphics library, in your class, you need to override the following functions :

void initScreen()
This function is used to initiate your graphics library window. It does not return anything, and does not take any parameter.

char getInput()
This function is used to retrieve the input of the user. It returns a char, which corresponds to a character from the ASCII table, and does not take any parameter.

void clearScreen()
This function is used to clear everything from the screen. It does not return anything and does not take any parameter.

void refreshScreen()
This function is used to display the elements you have previously draw in your game using the drawSquare and drawText functions. It does not return anything and does not take any parameter.

void endScreen()
This function is used to destroy your graphics library. It does not return anything, and does not take any parameter. This function is intended to be called only by the core of Arcade each time a graphics library needs to be unloaded.

std::tuple<int, int> getWindowSize()
This function is used to retrieve the size of the current window. It returns a *std::tuple<int, int>* and does not take any parameter. The first value of the tuple correspond to the width of the window and the second value to the height.

void drawSquare(std::tuple<float, float> position, std::tuple<int, int> rectSize, int color)
This function is used to draw a square or a rectangle inside the current window. It does not return anything and it takes multiple parameters :

- a *std::tuple<float, float> position*, used to set the position of the shape. The first value of the tuple correspond to the x position and the second to the y position.
- a *std::tuple<int, int> rectSize*, used to set the size of the shape. The first value of the tuple correspond to the width and the second to height.
- a *int color*, used to set the color of the shape. You can use a macro from the *src/interface/constants.hpp* header and have a function inside your class that translate that macro to a color of your graphical library.

void drawtext(std::tuple<int, int> position, float scale, std::string &text)

This function is used to display text inside the current window. It does not return anything and it takes multiple parameters :

- a std::tuple<float, float> position, used to set the position of the text. The first value of the tuple correspond to the x position and the second to the y position.
- a float scale, used to set the scale of the text.
- a std::string &text, which correspond to the text you want to display.

Next, in your *class.cpp* file you will need to create the **IDisplayModule *create_object()** function, using the extern “C” keyword. This function is used by the Arcade core at library opening to retrieve a new object of your class. This function should look like this :

```
extern "C" IDisplayModule *create_object() {
   return new yourClassName;
}
```

Finally, you need to compile your graphical library using the *-shared -fpic* flags and put the .so file inside the *lib/* folder.