

# # Soft356 coursework 2 – markdown

## # How does the user interact with your executable? How do you open and control the software you wrote (exe file)?

The executable file is called 'Assignment2' and can be found in the 'SOFT356Coursework2Executable' folder. When you open the file, you will be greeted with a yellow background. Using the mouse, you can adjust the camera direction till you find the boat, with the creeper light orbiting it. You can use 'W' 'A' 'D' and 'S' to adjust the camera position and the mouse to adjust the camera direct, so you can move around the screen and look at the objects from all angles. The position of the creeper is representative of the position of the light.

The programmes default settings are 'Automatic mode', which means the orbiting light cube will automatically change colours as it goes around the boat. By pressing the 'M' button on the keyboard, you can change this to manual.

In manual mode, the light source will stay the same colour, unless told otherwise by the user. This is done by the user pressing 'R' and 'T' to adjust the amount of red, 'G' and 'H' to adjust the amount of green and 'B' and 'N' to adjust the amount of blue.

You can the press the 'M' key to change it back to automatic. Pressing the 'C' key will clear the objects from the screen.

When the user wants to exit the programme, they will have to press the 'Start' button on the keyboard(so that the mouse is not locked onto the middle of the programme) and click the exit button on the top right corner of the window.

## # How does the program code work? How do the classes and functions fit together and who does what?

The program is extended from coursework 1, which reads the creeper and boat files then displays them on the screen.

The Display function works out the x and z position for a 2D circle so the light source can orbit the boat, using the a variable updated every loop to determine its position in the circle, this can be adjusted to speed up/slow down how fast the light source orbits. it then places this relative to the boat. If its on 'automatic' mode, the light colour will change, depending on where it is in the circle. This is because I used the normalised light position to update the colour.

The display function is also where I add in the mouse control. I ensure the cursor stays in the middle of the widow and use how much the user adjusts its x and y values, to work out what direction it should be facing.

This is also where the data is cleared, ready for more data to be drawn and where the background colour is set. I also have the default camera position here and the ability for it to be changed, depending on what keys the users presses.

In the `key_callback` function, I assign the different keyboard keys to call different functions. Such as the `clearobjs` function, which clears the objects from screen and the `manual` function, which toggles between manual and automatic mode. User input can also adjust the colour of the light source, ensuring that the input stays between 0 and 1.

I've used object orientated programming which promotes encapsulation, meaning all related code is kept together in its own class/methods such as with the data loader. As I've changed things which are part of the main display function, eg shaders and lighting, I have not needed to encapsulate it. C++ is also very focused on memory management, meaning using objects for everything can be inefficient and slow the program down, as well as add unnecessary complexity to the code. With this in mind, the main method creates the window, saves memory location of shaders to variables and calls the `loadobjs` function.

The `triangles.vert` file uses position, light position, view matrices and model matrices to work out the direction of the light, normals and camera. And the `triangles.frag` file sets the colour, defuse, reflection and transparency of the light using these directions.

# What makes your program special and how does it compare to similar things?  
(Where did you get the idea from? What did you start with? How did you make yours unique? Did you start with a given project?)

I started this project by continuing from coursework one. In coursework one, I created a cube and boat object that had a creeper skin, with no user interaction. I added the following functionality to this project to make it more complex and to have user interaction:

- A light source which revolved around the object, changing colours as it goes.
- The ability for the user to change the direction the camera is looking, using the mouse.
- The ability for the user to change the position of the camera, using the keyboard ('W' for up, 'S' for down, 'A' for left and 'D' for right).
- The ability to put the light on a 'manual' mode, in which the user can change the colour of the light, by adjusting the amount of green, Red and Blue used. ('G' for increasing Green 'H' for reducing, 'B' for increasing Blue 'N' for reducing and 'R' for increasing Red and 'T' for reducing).

Other than these changes, the rest of the code was from assignment one. I originally had a different, more complex idea in mind but due to the time constraints, focused my project mainly on shading.

# Include a link to a video report that details the above

Here's a link to the video:

<https://youtu.be/Qw7O5W6IOok>