

CS6610 PROJECT 1 – Hello World

Varsha Alangar

Implemented features:

All requirements listed in the Project 1 including the additional requirements for CS 6610 students has been implemented.

1. This project uses C++ coding practices.
2. The FreeGLUT library is used to create the OpenGL window.
3. The window size is explicitly mentioned using the `glutInitWindowSize` call. A size of 600 x 600 is created.
The window is created using the `glutCreateWindow` call.
4. The “Esc” key can be used to close the window. The `glutKeyboardFunc` and `glutLeaveMainLoop` calls help achieve this.
5. The background is animated automatically. Interpolation is done between white and steelblue colors. Glut calls `glutIdleFunc`, `glutPostRedisplay` are used along with the `glutSwapBuffers` call.
6. Other calls like `glutInit`, `glutInitDisplayMode`, `glutMainLoop`, `glutDisplayFunc` have also been used.

Additional functionalities:

The project did not require the OpenGL window to be explicitly positioned. However, I have used the `glutInitWindowPosition` call to place the window at 100 x 100 location on the screen.

Using the above implementations:

After running the program, a window will open with a 600 x 600 dimension at a 100 x 100 distance from the top left corner of the screen. The background color changes automatically. To close the window, hit the Esc key.

OS Used: Windows 10

IDE: Visual Studio 2013

Libraries and dependencies:

All the libraries used in the project are included under the `lib` folder within the zip file. They include: `opengl32.lib`, `glu32.lib` and `freeglut.lib`

All the header files are included within the `GL` folder contained in an include folder within the zip file. They have been included as `#include <GL/gl.h>` and `#include <GL/freeglut.h>` in the code.

All the DLLs required are placed in the `Debug` folder of the zip file.

The source code itself was created and compiled in Visual Studio and is available as `main.cpp` in `HelloWorld` folder along with the solution.

The executable is available in the `Debug` folder.

Requirements to compile the project:

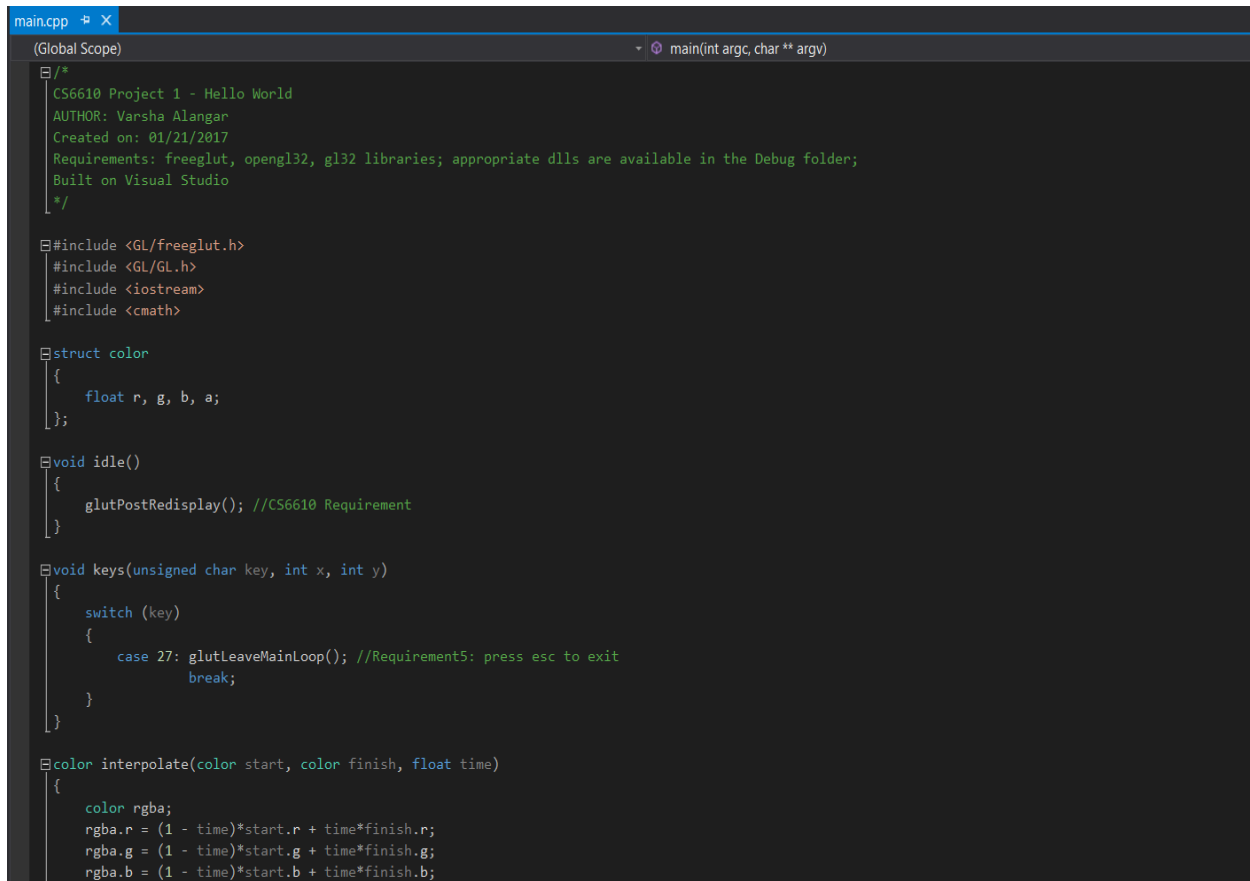
Unzip the project zip file and open the solution in Visual Studio.

In the properties of the project, link to the libs, dlls and header files. Make sure to choose “All configurations” in the properties window before adding the dependencies.

As mentioned earlier, all the required libraries, dlls and header files are available in the lib, Debug and include folders respectively of the zip file.

Please let me know if there is any issue in running the code.

Screenshot of code and output:



```
main.cpp  X
(Global Scope)  main(int argc, char ** argv)

/*
CS6610 Project 1 - Hello World
AUTHOR: Varsha Alangar
Created on: 01/21/2017
Requirements: freeglut, opengl32, gl32 libraries; appropriate dlls are available in the Debug folder;
Built on Visual Studio
*/

#include <GL/freeglut.h>
#include <GL/GL.h>
#include <iostream>
#include <cmath>

struct color
{
    float r, g, b, a;
};

void idle()
{
    glutPostRedisplay(); //CS6610 Requirement
}

void keys(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 27: glutLeaveMainLoop(); //Requirement5: press esc to exit
                break;
    }
}

color interpolate(color start, color finish, float time)
{
    color rgba;
    rgba.r = (1 - time)*start.r + time*finish.r;
    rgba.g = (1 - time)*start.g + time*finish.g;
    rgba.b = (1 - time)*start.b + time*finish.b;
```

```
main.cpp [X]
(Global Scope) main(int argc, char** argv)

    rgba.a = start.a * finish.a;
    return rgba;
}

void render()
{
    //interpolating between white and steelblue colors.
    color begin = { 1.0, 1.0, 1.0, 1.0 }; //White
    color end = { 0.137255, 0.419608, 0.556863, 1.0 }; //Steelblue

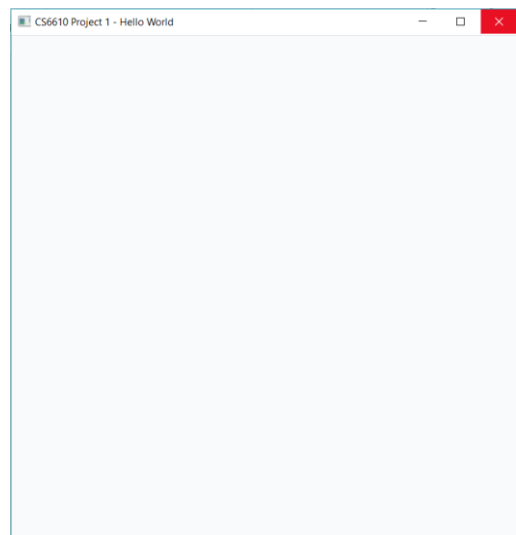
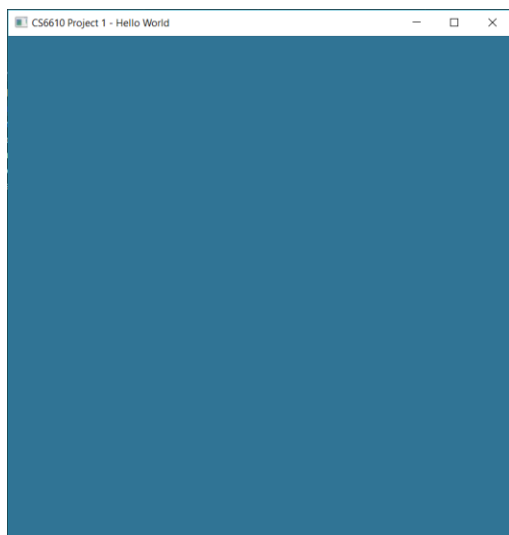
    //using trigonometric functions to determine the new color using interpolation
    const int current_t = glutGet(GLUT_ELAPSED_TIME);
    const float t = std::cos(float(current_t)*0.001)*0.5 + 0.5; //formula: float t = cos(x)*0.5+0.5 to scale it between 0 and 1.
    color current_c = interpolate(begin, end, t);

    glClearColor(current_c.r, current_c.g, current_c.b, current_c.a); //Setting the new color
    glClear(GL_COLOR_BUFFER_BIT); // Clearing the buffer with the preset color
    glutSwapBuffers();
}

void initialize()
{
    glClearColor(0.209f, 0.224f, 0.224f, 1.0f); //Requirement4: Setting background color of the window initially
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv); //to initialize GLUT
    glutInitDisplayMode(GLUT_RGBA);
    glutInitWindowSize(600, 600); //Requirement3: Specifying the size of the window
    glutInitWindowPosition(100, 100);
    glutCreateWindow("CS6610 Project 1 - Hello World");
    glutDisplayFunc(render);
    glutKeyboardFunc(keys);
    glutIdleFunc(idle);
    //initialize();
    glutMainLoop();
    return 0;
}
```

Background interpolated between white and steelblue colors:



A mid color achieved through interpolation:

