**Session 1: Introduce to OpenGL**

Intended Learning Outcome:

1. Students will be introduced to OpenGL.
2. Students will be able to use OpenGL with CodeBlocks.
3. Students will be able to implement computer graphics course using OpenGL.

**Expected Skills:**

1. Gathering knowledge about OpenGL
2. Clear idea about OpenGL for answering OpenGL related questions.

**Tools Required:**

1. CodeBlocks
2. OpenGL and GLUT using CodeBlocks.

**Session Detail:**

In this session, teachers will be given some theoretical resource and will be taught how to use OpenGL in CodeBlocks. After this, a sample program will be run in CodeBlocks using OpenGL. Mainly, this session is focused on to provide knowledge about OpenGL.



**Step 1**

Download CodeBlocks and MinGW from the CodeBlocks website. Be sure to select the binary installer (.exe file) which comes with the MinGW compiler. As of writing this tutorial the file you need is: 'codeblocks-10.05mingw-setup.exe' (found by clicking on 'Downloads' > 'Download the binary release').

**Step 2**

Download the GLUT files and downloading ‘glut.zip’

**Step 3**

Install CodeBlocks and MinGW by running the '.exe' file you downloaded in Step 1. The default install options are sufficient.

**Step 4**

Extract the 'Glut.zip' file you downloaded in Step 2 and do the following:-

Copy glut32.dll into your 'C:\Windows\System32' folder (If you're using Windows 7 64-bit, you'll need to copy this file into 'C:\Windows\sysWOW64').

Copy glut.h into the ‘include\GL’ folder in the MinGW compiler folder. If you installed CodeBlocks and MinGW to the default directory in Step 3, this folder will be 'C:\Program Files\CodeBlocks\MinGW\include\GL'.

3

Copy libglut32.a into the 'MinGW\lib' folder. If you installed CodeBlocks and MinGW to the default directory in Step 3, this folder will be 'C: \Program Files\CodeBlocks\MinGW\lib'.

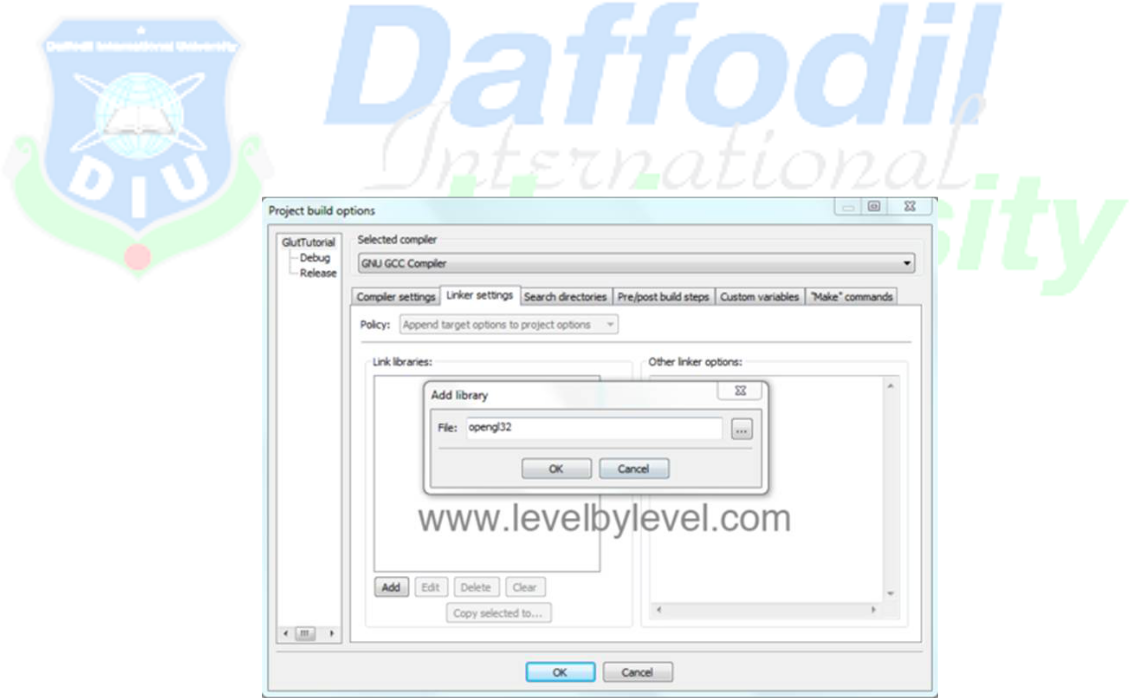
**Step 5**

Now you have all the required software installed you can create a new OpenGL and GLUT project. CodeBlocks has an option for creating a GLUT project when creating a new project, we won't be using this option today though as I want to show you how to link your files:

Startup CodeBlocks and select 'File > New > Project…', in the new project window select 'Empty project' and click the 'Go' button.

In the wizard that follows press 'Next' to the first page (the one welcoming you to the wizard) and in the second page enter a project title name into the textbox labelled 'Project title:' and leave the other 3 textboxes with their default properties (you can change these if you like, it’s up to you) and click 'Next'. In this final wizard page, ensure that 'GNU GCC Compiler' is selected in the drop-down menu box labeled 'Compiler:' and leave the other options with their default setting and click 'Finish'.

Once your project has been created, right click on the project in the Workspace explorer frame on the left of the screen and select 'Build options…'. In this options window select your project name in the tree to the left of the window (do not select either Debug or Release) and open the 'Linker settings' tab.

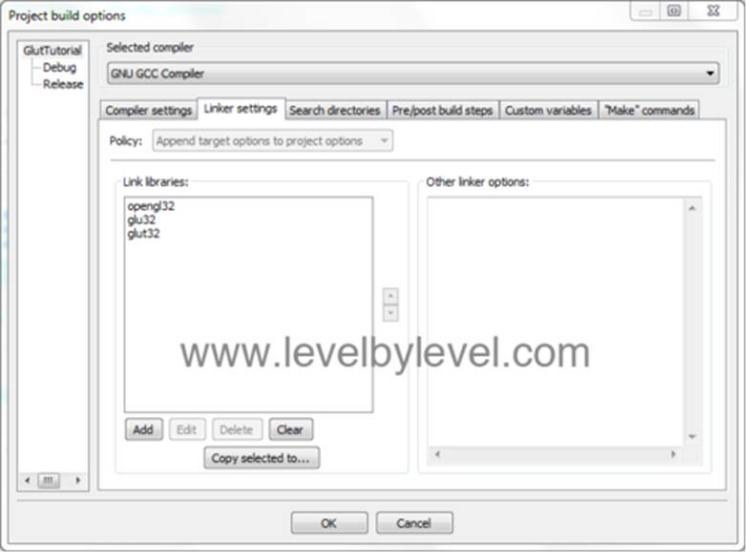


In this tab, click the 'Add' button below the 'Link libraries:' text box and in the textbox next to 'File:' enter 'opengl32' (without the quotes) and click 'OK'. (See Picture 1.1 Below)

Picture 1.1 - add 'opengl32' to the linker

Do the same again, this time entering 'glu32' and once again entering 'glut32'. (See Picture 1.2 Below)

4

****

Picture 1.2 - your linker settings should look like this

**A Sample Program:**

****

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

/\* clear all pixels \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

/\* draw white polygon (rectangle) with corners at

* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0) \*/

glColor3f (1.0, 0.25, 1.0); glBegin(GL\_POLYGON); glVertex3f (0.25, 0.25, 0.0); glVertex3f (0.75, 0.25, 0.0); glVertex3f (0.75, 0.75, 0.0); glVertex3f (0.25, 0.75, 0.0);

glEnd();

/\* don't wait!

* start processing buffered OpenGL routines \*/

glutSwapBuffers ();

}

void init (void)

{

/\* select clearing (background) color \*/ glClearColor (0.0, 0.0, 0.0, 0.0);

/\* initialize viewing values \*/ glMatrixMode(GL\_PROJECTION); glLoadIdentity();

glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);

}

/\*

* Declare initial window size, position, and display mode
* (single buffer & RGBA). Open window with "hello"
* In its title bar. Call initialization routines.
* Register callback function to display graphics.
* Enter main loop and process events.



\*/

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

6

glutInitWindowPosition (100, 100);

glutCreateWindow ("hello");

init ();

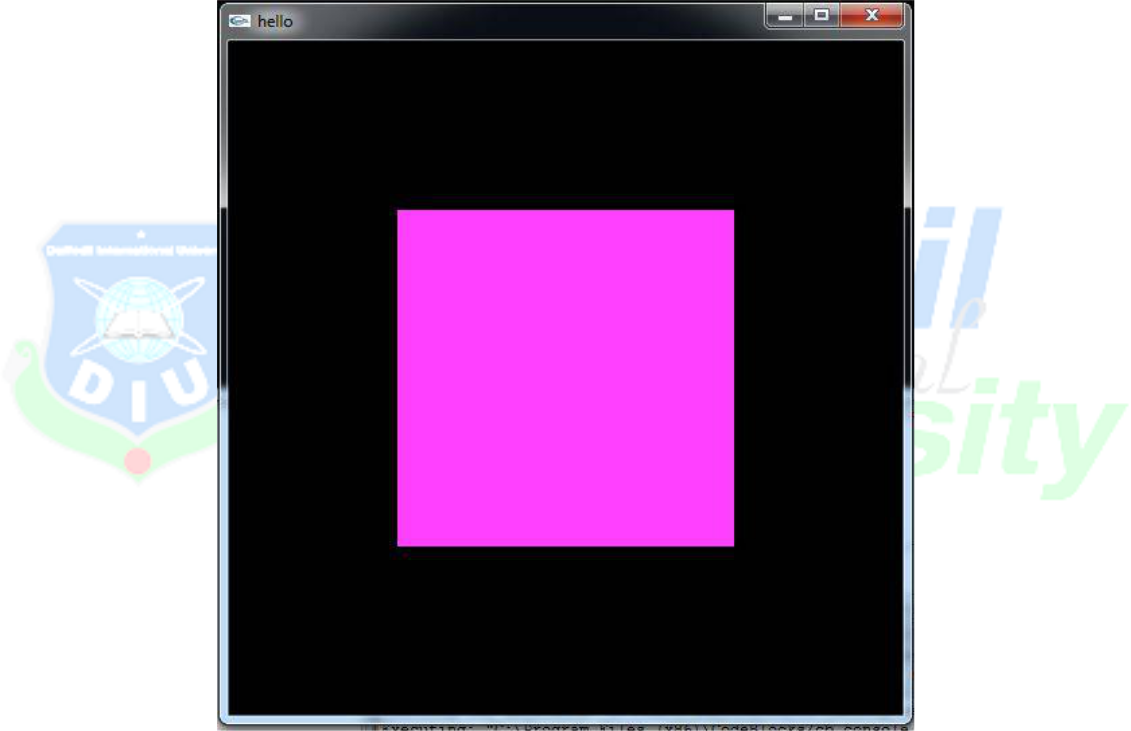
glutDisplayFunc(display);

glutMainLoop();

return 0; /\* ISO C requires main to return int. \*/

}

**Output:**

****