

Laboratory No: 01

Title : Write and explain a Program for drawing a simple 2-D object House

Course Title : Computer Graphics

Course Code : CSE403

3rd Year 2nd Semester Examination 2022

Date of Submission : 01-05-2022

Submitted to-

Dr. Md. Zulfiker Mahmud

Associate Professor

Department of Computer Science and Engineering

Jagannath University

Dhaka, Bangladesh

Submitted By-

SL	ID	Name	Session	Signature
01	1951CSE00817	Md Abid Hasan Dihan	2019	Dihan
02	1950CSE00801	Md Ahmad Habib	2019	Habib
03	1950CSE00807	Ayetullah Al Khomane	2019	Ayettullah

Department of Computer Science and Engineering

Manarat International University

Ashuliya ,Dhaka, Bangladesh

Experiment No: 01

Experiment title : Write and explain a Program for drawing a simple 2-D object House

Introduction : The task is to write C++ Program to create a house using graphic. To run the program we have to include the below header files:

```
#include<windows.h>

#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>
```

Apertures :

- Computer
- Code Blocks software(version 17.12)

Approach:

- Set display window color to as `glClearColor(R,G,B,Alpha)`
- Set projection parameters.
- Set 2D Transformation with 4 parameter as `gluOrtho2D(Min Width, Max Width, Min Height, Max Height)`
- `glClear` sets the bitplane area of the window to values previously selected by `glClearColor`. `GL_COLOR_BUFFER_BIT` Indicates the buffers currently enabled for color writing.
- We use `glColor` function to set the *foreground color*, and `glClearColor` function to set the *background* (or *clearing*) color.
- A geometric primitive is defined by specifying its vertices via `glVertex` function, enclosed within a pair `glBegin` and `glEnd`.
- By using the function “`glBegin(GL_POLYGON);` ” Draws a single, convex polygon. Vertices 1 through N define this polygon. The space between 2/3/4 points we have taken by using the function for making the structure of the house will be completely filled.
- To make the body of house we will call “`glBegin(GL_POLYGON);`” function. By calling this function body of the house will be visible with appropriate color and width, height.
- Now for making the roof of the house. We will call “`glBegin(GL_TRIANGLES);` ” function. Treats each triplet of vertices as an independent triangle. Vertices $3n-2$, $3n-1$, and $3n$ define triangle n . $N/3$ triangles are drawn. By using this function the roof of 2D house will be visible by taking 3 points to make a triangle and the space between the triangle will be filled with appropriate color and accurate width, height.
- For making the door and window of that house we will call “`glBegin(GL_POLYGON);` ” function. This function will help to draw the door

and windows by taking points and filling the space between points with proper size and color.

- In the main function-
 - i. **glutInit:** initializes GLUT, must be called before other GL/GLUT functions. It takes the same arguments as the main().
 - ii. **glutInitWindowSize:** specifies the initial window width and height, in pixels.
 - iii. **glutInitWindowPosition:** positions the top-left corner of the initial window at (x, y) . The coordinates (x, y) , in term of pixels, is measured in window coordinates, i.e., origin $(0, 0)$ is at the top-left corner of the screen; x-axis pointing right and y-axis pointing down.
 - iv. **glutDisplayFunc:** registers the callback function (or event handler) for handling window-paint event. The OpenGL graphic system calls back this handler when it receives a window re-paint request. In the example, we register the function `display()` as the handler.
 - v. **glutMainLoop:** We then put the program into the event-handling loop, awaiting for events (such as window-paint request) to trigger off the respective event handlers (such as `display()`)

Experiment :

We are writing code using C++ language. Here is the Code Segment –

```
#include<windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.h>

void init()

{

    // Set display window color to as glClearColor(R,G,B,Alpha)

    glClearColor(0.52, 0.808, 0.922, 0.0);

    // Set projection parameters.

    glMatrixMode(GL_PROJECTION);

    // Set 2D Transformation as gluOrtho2D(Min Width, Max Width, Min Height, Max Height)

    gluOrtho2D(0.0, 2400, 0.0, 1800);

}

void home()

{

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(0.297, 0.519, 0.152);

    glBegin(GL_POLYGON);

    glVertex2i(0, 300);

    glVertex2i(2400, 300);

    glVertex2i(2400, 0);

    glVertex2i(0, 0);

}
```

```
glEnd();

glColor3f(0.1875, 0.3672, 0.379); //Body Color

glBegin(GL_POLYGON);

glVertex2i(600, 1200);

glVertex2i(1800, 1200);

glVertex2i(1800, 300);

glVertex2i(600, 300);

glEnd();

glBegin(GL_TRIANGLES); //roof color

glColor3f(0.524, 0.528, 0.532);

glVertex2i(1200, 1700);

glVertex2i(500, 1200);

glVertex2i(1900, 1200);

glEnd();

glColor3f(0.400, 0.404, 0.408); //Door color

glBegin(GL_POLYGON);

glVertex2i(1100, 950);

glVertex2i(1300, 950);

glVertex2i(1300, 300);

glVertex2i(1100, 300);

glEnd();

    glColor3f(0.18, 0.32, 0.35); //window color

glBegin(GL_POLYGON);

glVertex2i(1450, 1000);
```

```
    glVertex2i(1700, 1000);

    glVertex2i(1700, 700);

    glVertex2i(1450, 700);

    glEnd();

    glFlush();

}

int main(int argc, char ** argv)

{

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutInitWindowPosition(100, 100);

    glutInitWindowSize(1920, 1080);

    glutCreateWindow("ID:817 ,801 ,807 -> Project House 2D ");

    init();

    glutDisplayFunc(home);

    glutMainLoop();

}
```

Result Data (Output) :

Below is the output of the above program:

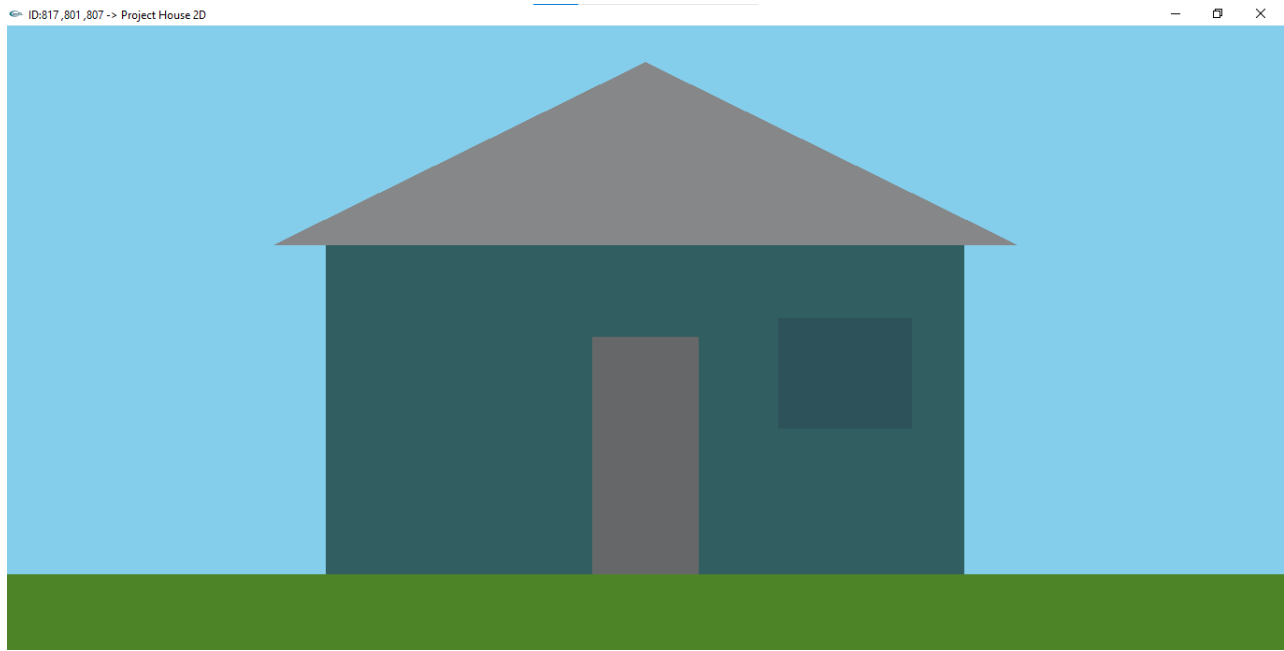


Fig : 2-D House

Conclusions :

The project on the whole presented a challenge, the course is different from other programming courses offered since there is a subjective parameter that's hard to measure. The team took on many challenges to bring the deliverable to this stage and all are pleased with the effort and the end result.