Department: Information Technology Engineering

Course: Computer Graphic

Name: Beun Bunleap

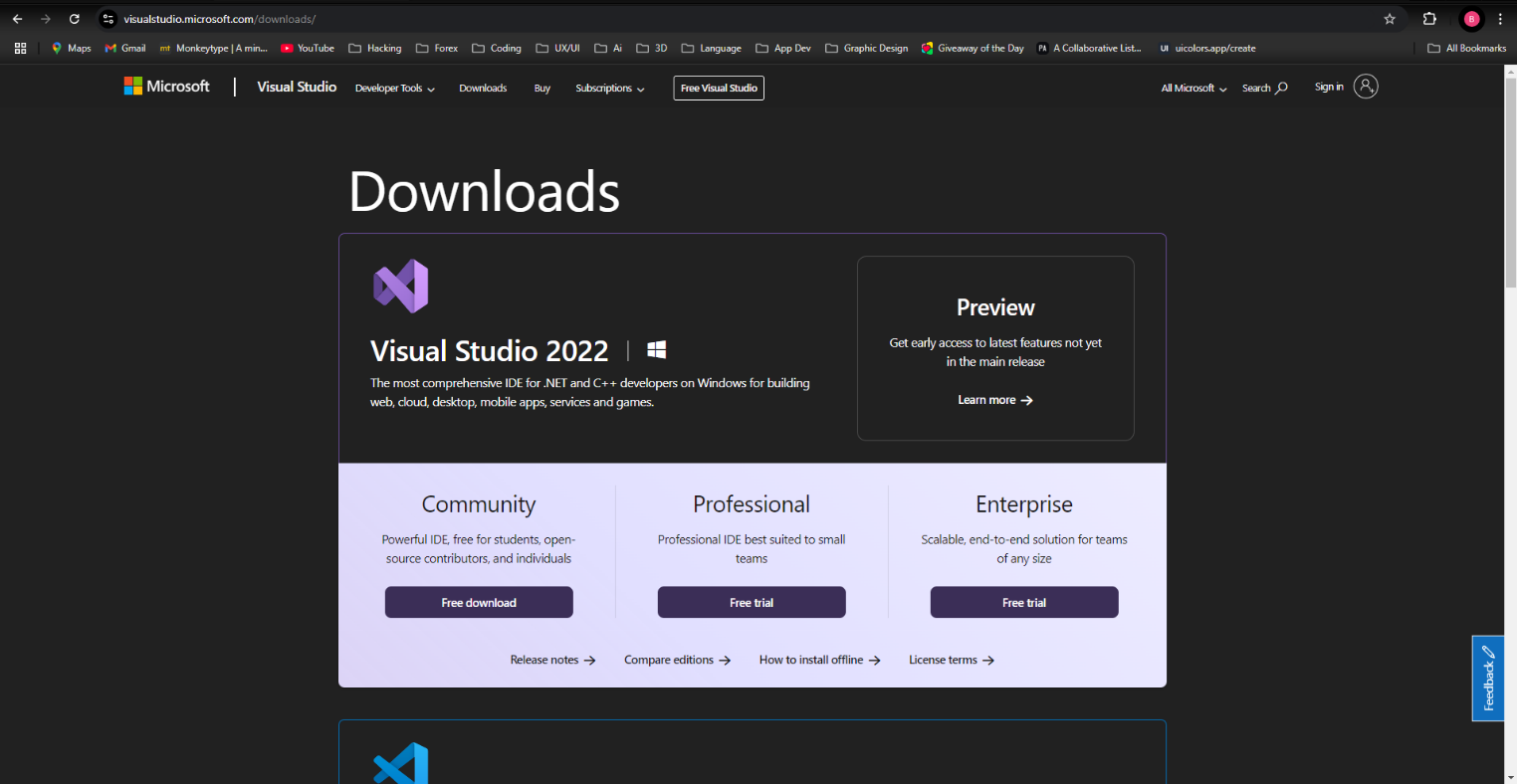
**Assignment I**

**Practical Homework Assignment 1: Development Environment Setup**

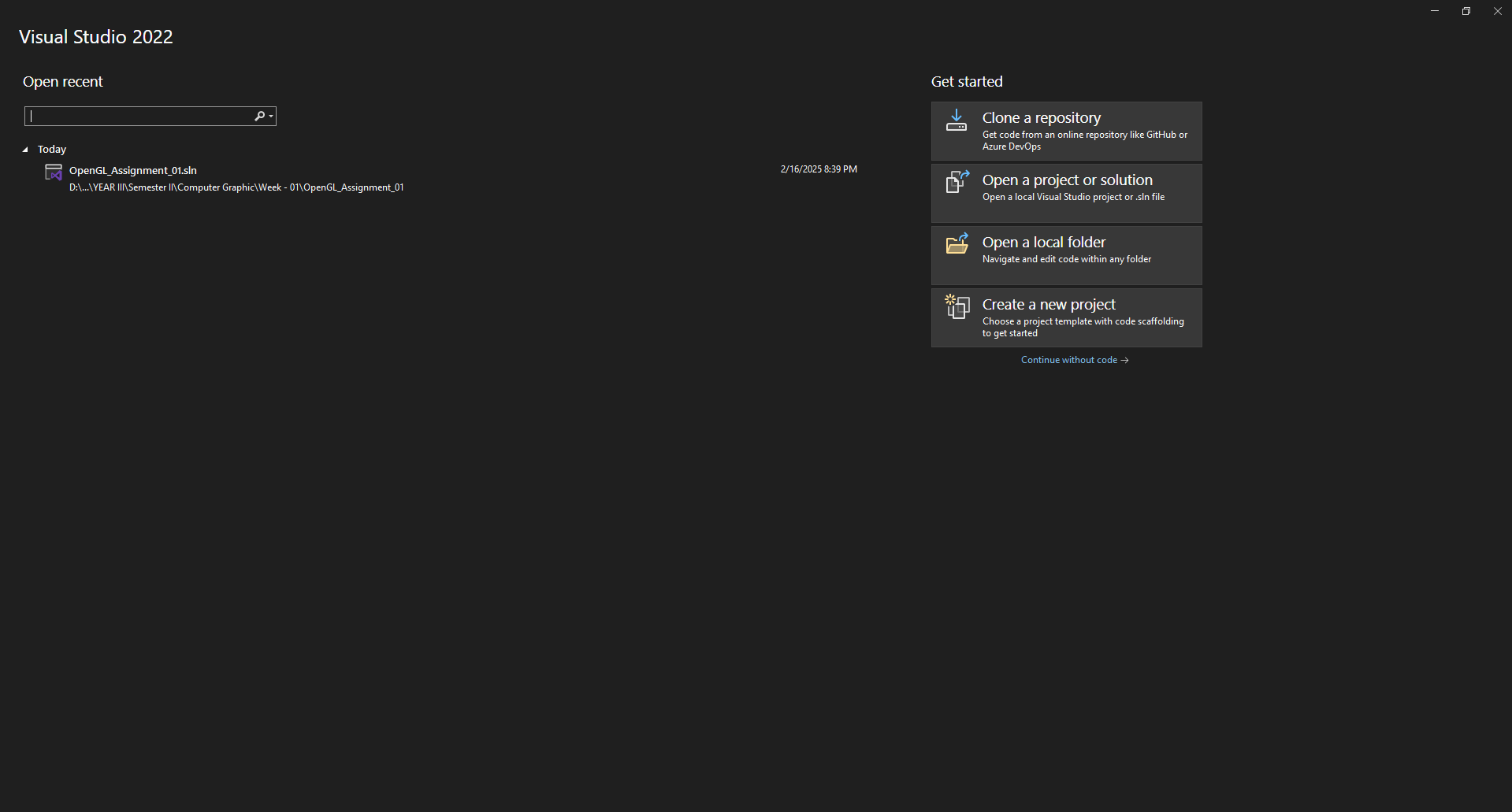
**Tasks and Solutions**

**1. Software Installation**

1. **Install Visual Studio Community Edition**:
   * Download and install Visual Studio Community Edition from the official website: [Visual Studio](https://visualstudio.microsoft.com/).
   * During installation, select the **Desktop development with C++** workload.



1. **Set up OpenGL and GLUT libraries**:
   * Download the GLUT library files:
     + glut.h
     + glut32.lib
     + glut32.dll
   * Place glut.h in the include folder of your Visual Studio installation (e.g., C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.XX.XXXXX\include\GL).
   * Place glut32.lib in the lib folder (e.g., C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.XX.XXXXX\lib\x86).
   * Place glut32.dll in the System32 folder (e.g., C:\Windows\System32).
2. **Configure Project Settings**:
   * Open Visual Studio and create a new C++ project.
   * Go to **Project Properties** > **Linker** > **Input**.
   * Add opengl32.lib, glu32.lib, and glut32.lib to the **Additional Dependencies** field.

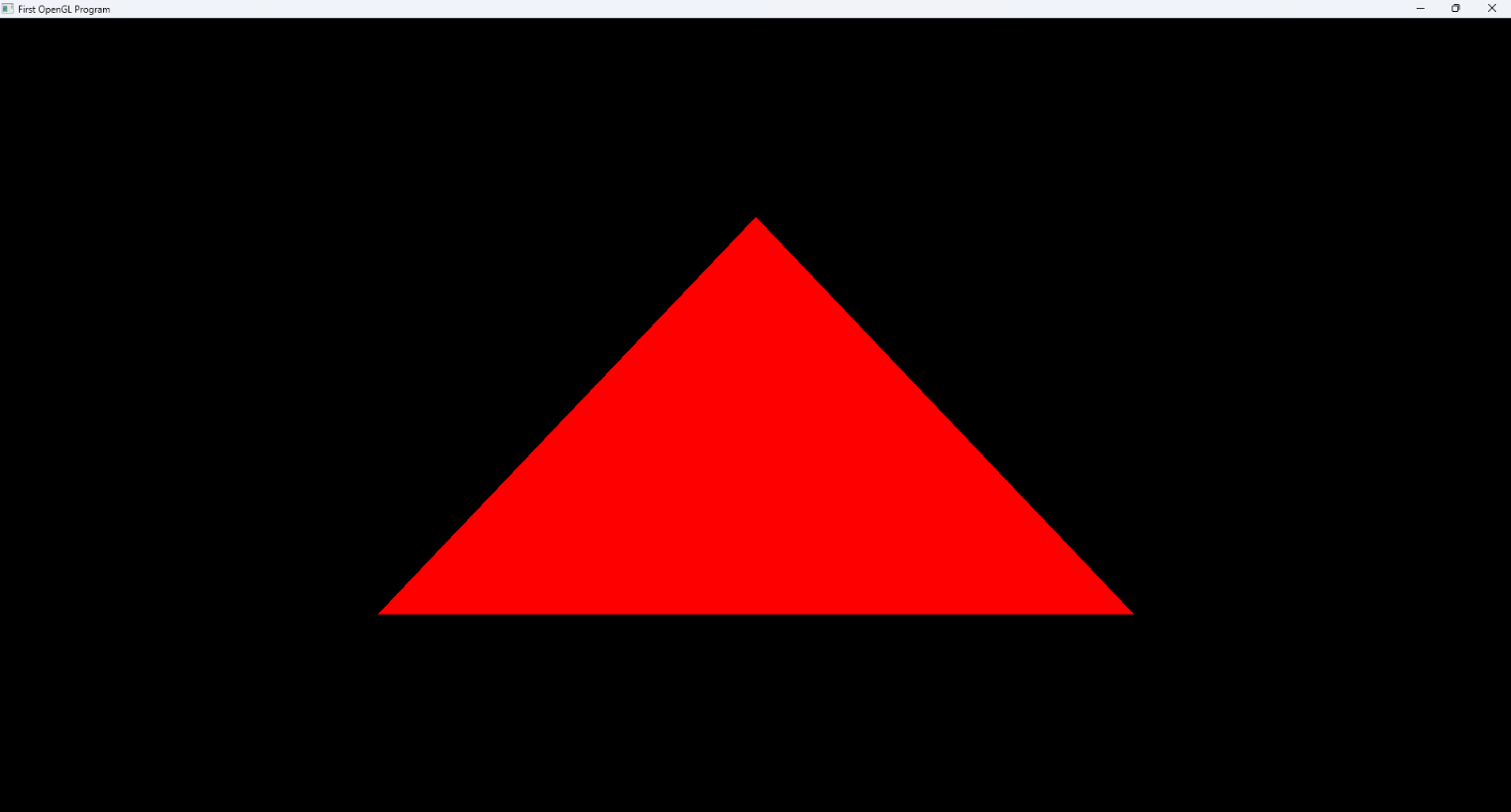


**2. Test Program Creation**

Create a test program to verify the setup:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #include <GL/glut.h>  **void** **display**() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3f(**1.0**, **0.0**, **0.0**); // Red color  // Draw a simple triangle  glBegin(GL\_TRIANGLES);  glVertex2f(-**0.5**, -**0.5**);  glVertex2f(**0.5**, -**0.5**);  glVertex2f(**0.0**, **0.5**);  glEnd();  glFlush();  }  **int** **main**(**int** argc, **char**\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("First OpenGL Program");  glutDisplayFunc(display);  glutMainLoop();  **return** **0**;  } | |

2. **Screenshot of Test Program Running**:



3. **List of Errors and Solutions**:

* **Error**: glut.h not found.
  + **Solution**: Ensure glut.h is placed in the correct include folder.
* **Error**: glut32.lib not found.
  + **Solution**: Ensure glut32.lib is placed in the correct lib folder.

**Practical Homework Assignment 2: Basic Graphics Programming**

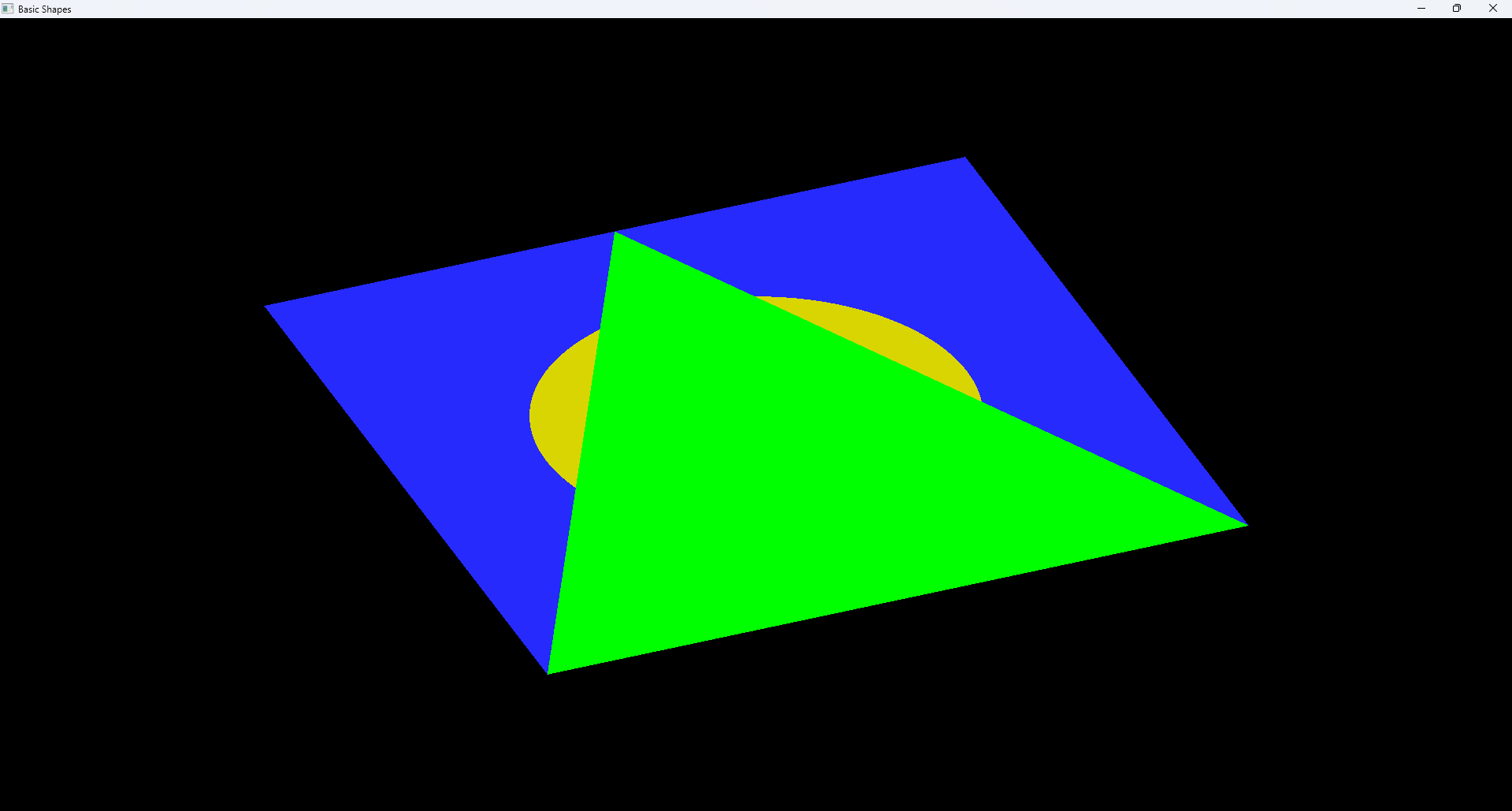
**Tasks and Solutions**

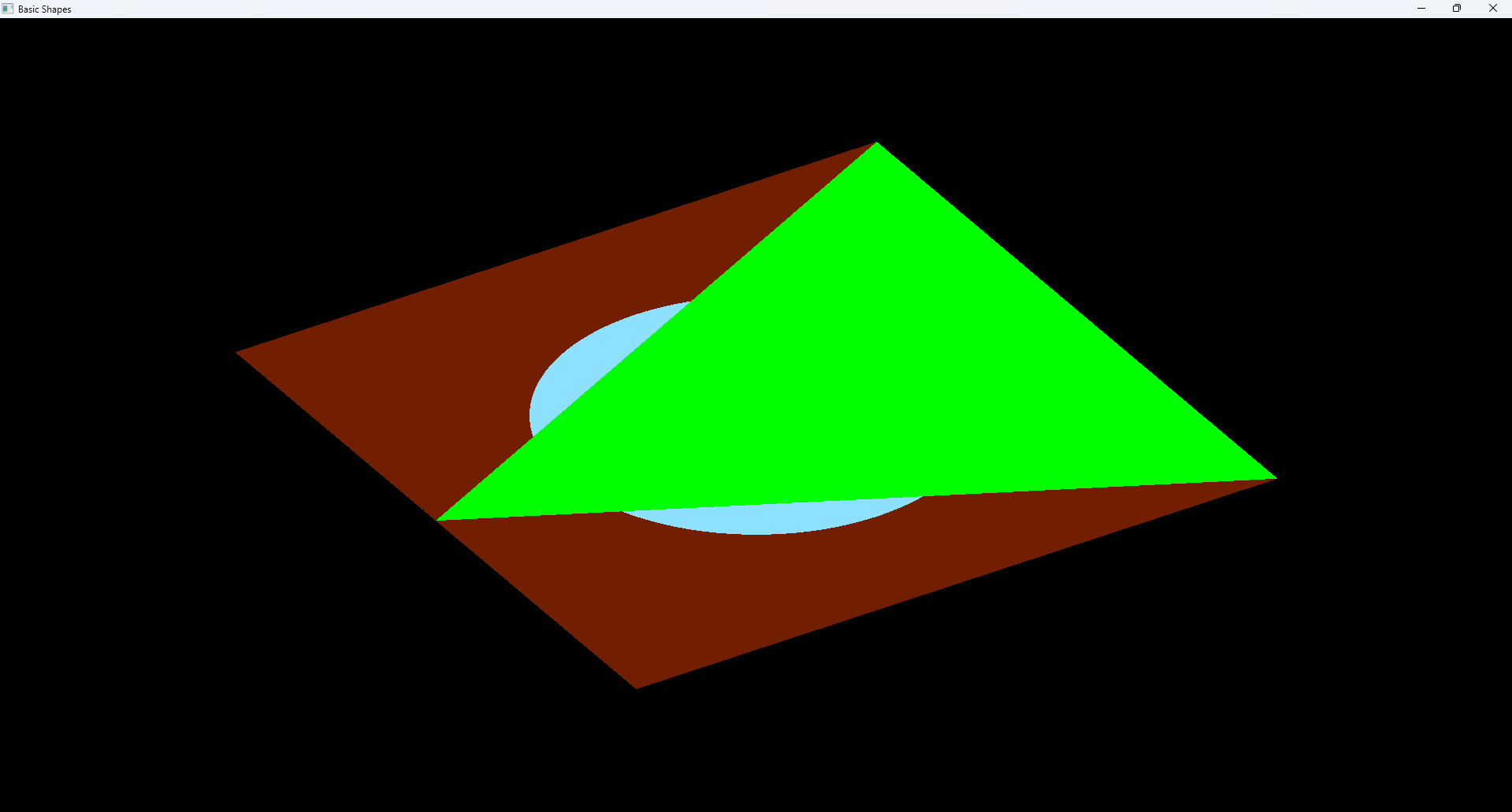
**1. Program Implementation**

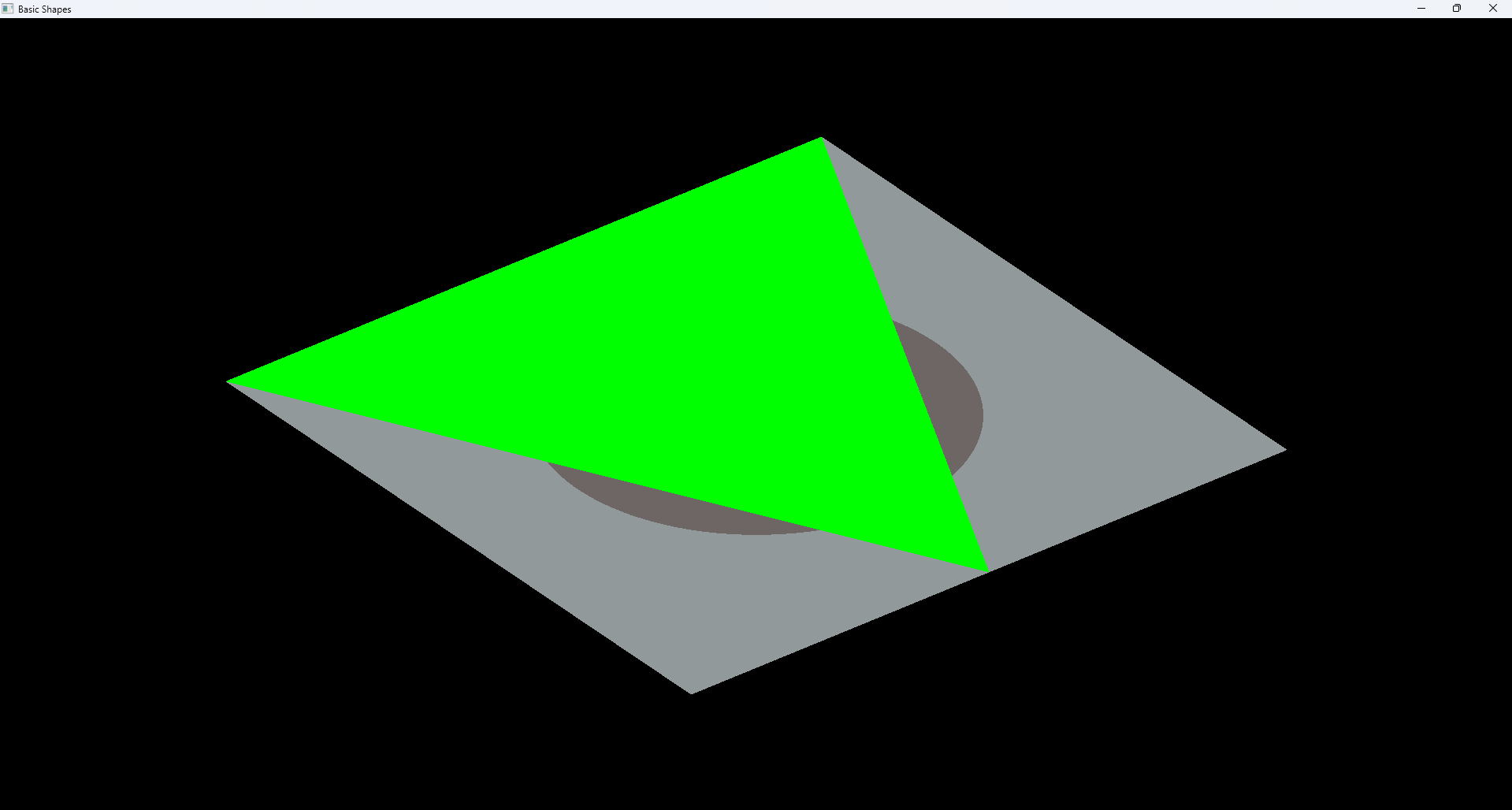
The program draws basic shapes, implements mouse interaction, and changes colors.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92 | #include <GL/glut.h>  #include <cstdlib> // For rand()  // Window dimensions  **const** **int** WINDOW\_WIDTH = **800**;  **const** **int** WINDOW\_HEIGHT = **600**;  // Global variables for color and animation  **float** red = **1.0**, green = **0.0**, blue = **0.0**;  **float** angle = **0.0**;  // Draw a rectangle  **void** **drawRectangle**() {  glBegin(GL\_POLYGON);  glVertex2f(-**0.5**, -**0.5**);  glVertex2f(**0.5**, -**0.5**);  glVertex2f(**0.5**, **0.5**);  glVertex2f(-**0.5**, **0.5**);  glEnd();  }  // Draw a circle  **void** **drawCircle**(**float** radius) {  glBegin(GL\_POLYGON);  **for** (**int** i = **0**; i < **360**; i++) {  **float** angle = i \* **3.14159** / **180**;  glVertex2f(cos(angle) \* radius, sin(angle) \* radius);  }  glEnd();  }  // Draw a triangle  **void** **drawTriangle**() {  glBegin(GL\_TRIANGLES);  glVertex2f(-**0.5**, -**0.5**);  glVertex2f(**0.5**, -**0.5**);  glVertex2f(**0.0**, **0.5**);  glEnd();  }  // Display function  **void** **display**() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glLoadIdentity();  // Rotate shapes  glRotatef(angle, **0.0**, **0.0**, **1.0**);  // Draw shapes with different colors  glColor3f(red, green, blue); // Set color  drawRectangle();  glColor3f(**1.0** – red, **1.0** – green, **1.0** – blue); // Complementary color  drawCircle(**0.3**);  glColor3f(**0.0**, **1.0**, **0.0**); // Green  drawTriangle();  glFlush();  glutSwapBuffers();  }  // Mouse interaction  **void** **mouse**(**int** button, **int** state, **int** x, **int** y) {  **if** (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN) {  // Change color on left click  red = (**float**)rand() / RAND\_MAX;  green = (**float**)rand() / RAND\_MAX;  blue = (**float**)rand() / RAND\_MAX;  }  glutPostRedisplay();  }  // Timer function for animation  **void** **update**(**int** value) {  angle += **2.0**; // Rotate shapes  **if** (angle > **360**) angle -= **360**;  glutPostRedisplay();  glutTimerFunc(**16**, update, **0**); // ~60 FPS  }  **int** **main**(**int** argc, **char**\*\* argv) {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);  glutCreateWindow(“Basic Shapes”);  glutDisplayFunc(display);  glutMouseFunc(mouse);  glutTimerFunc(**0**, update, **0**);  glutMainLoop();  **return** **0**;  } | |

* When click on it, it will change color







**Troubleshooting Notes**

* **Error**: glut.h not found.
  + **Solution**: Ensure glut.h is placed in the correct include folder.
* **Error**: glut32.lib not found.
  + **Solution**: Ensure glut32.lib is placed in the correct lib folder.
* **Error**: Program crashes on mouse click.
  + **Solution**: Ensure the mouse() function is correctly implemented and linked.