OpenGL - Adarsh Kumar Singh 111915005.md

Adarsh Kumar Singh

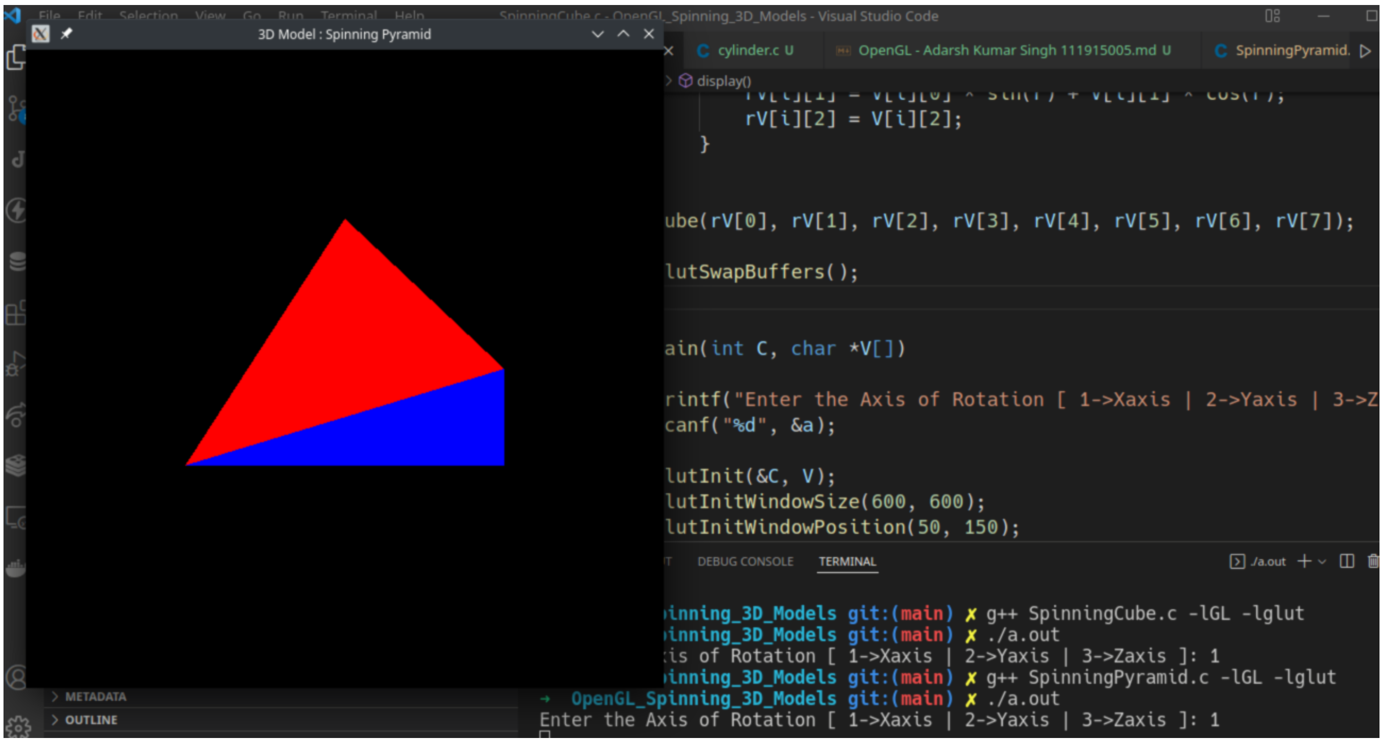
111915005

Table of Contents

3D Model : Spinning Pyramid

```
#include <GL/gl.h>
#include <GL/glut.h>
#include <stdio.h>
#include <math.h>

GLfloat d = 0;
int a = 0;

void MyInit()
{
    glClearColor(0, 0, 0, 1);
    glEnable(GL_DEPTH_TEST);
}

void Spin()
{
    d = d + 0.5;
    if (d > 360)
        d = 0;
    glutPostRedisplay();
}

void Face(GLfloat A[], GLfloat B[], GLfloat C[])
{
    glBegin(GL_POLYGON);
```

```c
    glVertex3fv(A);
    glVertex3fv(B);
    glVertex3fv(C);
    glEnd();
}

void Tri(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[])
{
    glColor3f(1, 0, 0);
    Face(V0, V1, V2);
    glColor3f(0, 1, 0);
    Face(V0, V1, V3);
    glColor3f(0, 0, 1);
    Face(V0, V2, V3);
    glColor3f(1, 1, 0);
    Face(V1, V2, V3);
}

void disp()
{
    GLfloat V[4][3] = {
        {0.5, 0, 0},
        {0, 0.5, 0},
        {-0.5, -0.5, 0.5},
        {0.5, -0.5, 0.5},
    };

    GLfloat rV[4][3], r;
    int i;
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    r = d * 3.14 / 180;

    if (a == 1)
    {
        for (i = 0; i < 4; i++)
        {
            rV[i][0] = V[i][0];
            rV[i][1] = V[i][1] * cos(r) - V[i][2] * sin(r);
            rV[i][2] = V[i][1] * sin(r) + V[i][2] * cos(r);
        }
    }
    if (a == 2)
    {
        for (i = 0; i < 4; i++)
        {
            rV[i][0] = V[i][2] * sin(r) + V[i][0] * cos(r);
            rV[i][1] = V[i][1];
            rV[i][2] = V[i][2] * cos(r) - V[i][0] * sin(r);
        }
    }

    if (a == 3)
    {
        for (i = 0; i < 4; i++)
        {
            rV[i][0] = V[i][0] * cos(r) - V[i][1] * sin(r);
            rV[i][1] = V[i][0] * sin(r) + V[i][1] * cos(r);
            rV[i][2] = V[i][2];
        }
    }

    Tri(rV[0], rV[1], rV[2], rV[3]);
    glutSwapBuffers();
}

int main(int C, char *V[])
{
    printf("Enter the Axis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis ]:
");
    scanf("%d", &a);

    glutInit(&C, V);
    glutInitWindowSize(600, 600);
    glutInitWindowPosition(50, 150);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);

    glutCreateWindow("3D Model : Spinning Pyramid");
    MyInit();
    glutDisplayFunc(disp);
    glutIdleFunc(Spin);
    glutMainLoop();
    return 0;
}
```

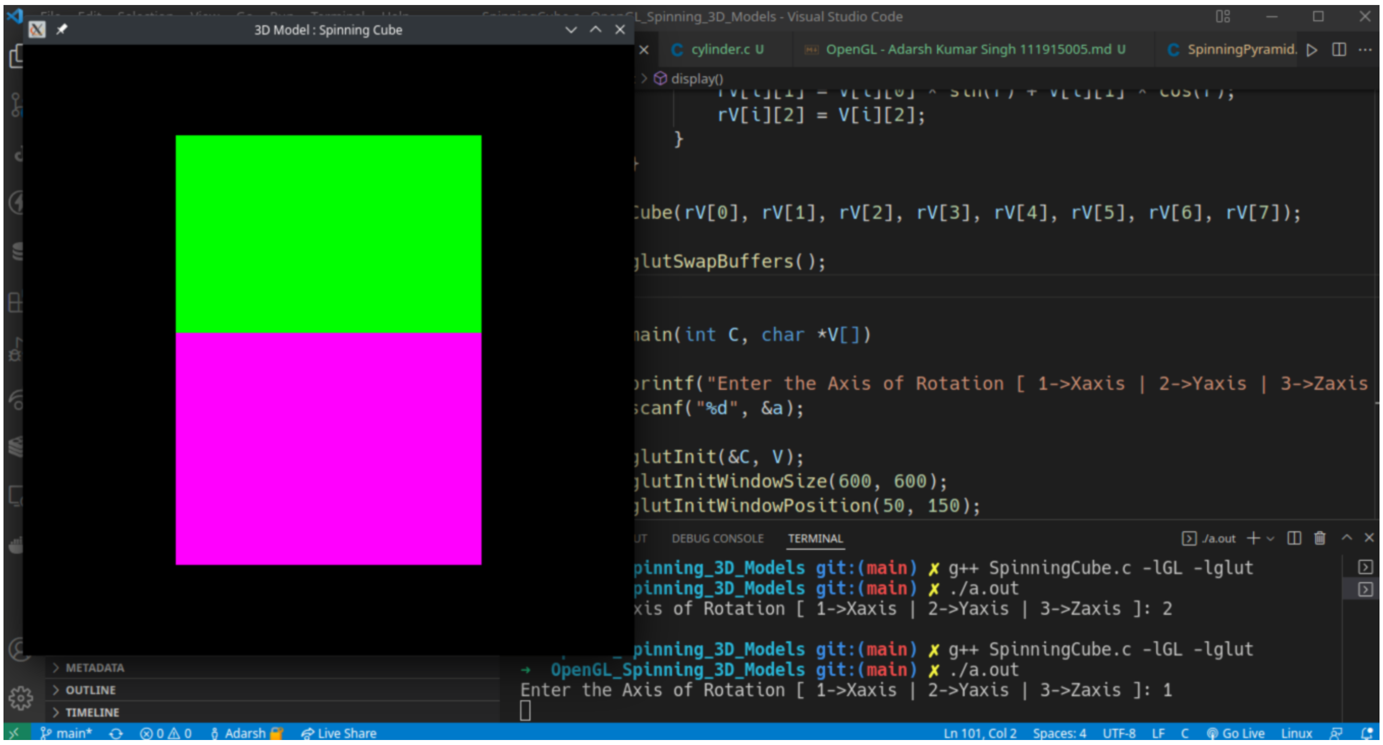

File Edit Selection View Go Run Terminal Help
SpinningCube.c - OpenGL_Spinning_3D_Models - Visual Studio Code
3D Model : Spinning Pyramid
cylinder.c U
OpenGL - Adarsh Kumar Singh 111915005.md U
SpinningPyramid.
display()
DEBUG CONSOLE
TERMINAL
./a.out
METADATA
OUTLINE

```
        rV[i][2] = V[i][2];
    }

ube(rV[0], rV[1], rV[2], rV[3], rV[4], rV[5], rV[6], rV[7]);

lutSwapBuffers();


ain(int C, char *V[])

rintf("Enter the Axis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Z
canf("%d", &a);

lutInit(&C, V);
lutInitWindowSize(600, 600);
lutInitWindowPosition(50, 150);
```

```
inning_3D_Models git:(main) ✗ g++ SpinningCube.c -lGL -lglut
inning_3D_Models git:(main) ✗ ./a.out
is of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis ]: 1
inning_3D_Models git:(main) ✗ g++ SpinningPyramid.c -lGL -lglut
→  OpenGL_Spinning_3D_Models git:(main) ✗ ./a.out
Enter the Axis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis ]: 1
```

3D Model : Spinning Cube

```
#include <GL/glu.h>
#include <GL/glut.h>
#include <stdio.h>
#include <math.h>

GLfloat d = 0;
int a = 0;

void MyInit()
{
    glClearColor(0, 0, 0, 1);
    glEnable(GL_DEPTH_TEST);
}

void Spin()
{
    d = d + 0.25;
    if (d > 360)
        d = 0;
    glutPostRedisplay();
}

void Face(GLfloat A[], GLfloat B[], GLfloat C[], GLfloat D[])
{
    glBegin(GL_POLYGON);
    glVertex3fv(A);
    glVertex3fv(B);
    glVertex3fv(C);
    glVertex3fv(D);
    glEnd();
}

void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat
V4[], GLfloat V5[], GLfloat V6[], GLfloat V7[])
{
    glColor3f(1, 0, 0);
    Face(V0, V1, V2, V3);
    glColor3f(0, 1, 0);
    Face(V4, V5, V6, V7);
    glColor3f(0, 0, 1);
    Face(V0, V4, V7, V3);
    glColor3f(1, 1, 0);
    Face(V1, V5, V6, V2);
    glColor3f(1, 0, 1);
    Face(V2, V3, V7, V6);
    glColor3f(0, 1, 1);
    Face(V0, V1, V5, V4);
}

void display()
{
    GLfloat V[8][3] = {
        {-0.5, 0.5, 0.5},
        {0.5, 0.5, 0.5},
        {0.5, -0.5, 0.5},
        {-0.5, -0.5, 0.5},
        {-0.5, 0.5, -0.5},
        {0.5, 0.5, -0.5},
        {0.5, -0.5, -0.5},
        {-0.5, -0.5, -0.5},
    };

    GLfloat rV[8][3], r;
    int i;
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    r = d * 3.14 / 180;

    if (a == 1)
    {
        for (i = 0; i < 8; i++)
        {
            rV[i][0] = V[i][0];
            rV[i][1] = V[i][1] * cos(r) - V[i][2] * sin(r);
            rV[i][2] = V[i][1] * sin(r) + V[i][2] * cos(r);
        }
    }

    if (a == 2)
    {
        for (i = 0; i < 8; i++)
        {
            rV[i][0] = V[i][2] * sin(r) + V[i][0] * cos(r);
            rV[i][1] = V[i][1];
            rV[i][2] = V[i][2] * cos(r) - V[i][0] * sin(r);
        }
    }

    if (a == 3)
    {
        for (i = 0; i < 8; i++)
        {
            rV[i][0] = V[i][0] * cos(r) - V[i][1] * sin(r);
            rV[i][1] = V[i][0] * sin(r) + V[i][1] * cos(r);
            rV[i][2] = V[i][2];
        }
    }

    Cube(rV[0], rV[1], rV[2], rV[3], rV[4], rV[5], rV[6], rV[7]);

    glutSwapBuffers();
}

int main(int C, char *V[])
{
    printf("Enter the Axis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis ]: 
");
    scanf("%d", &a);

    glutInit(&C, V);
    glutInitWindowSize(600, 600);
    glutInitWindowPosition(50, 150);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);

    glutCreateWindow("3D Model : Spinning Cube");
    MyInit();
    glutDisplayFunc(display);
    glutIdleFunc(Spin);
    glutMainLoop();
    return 0;
}
```

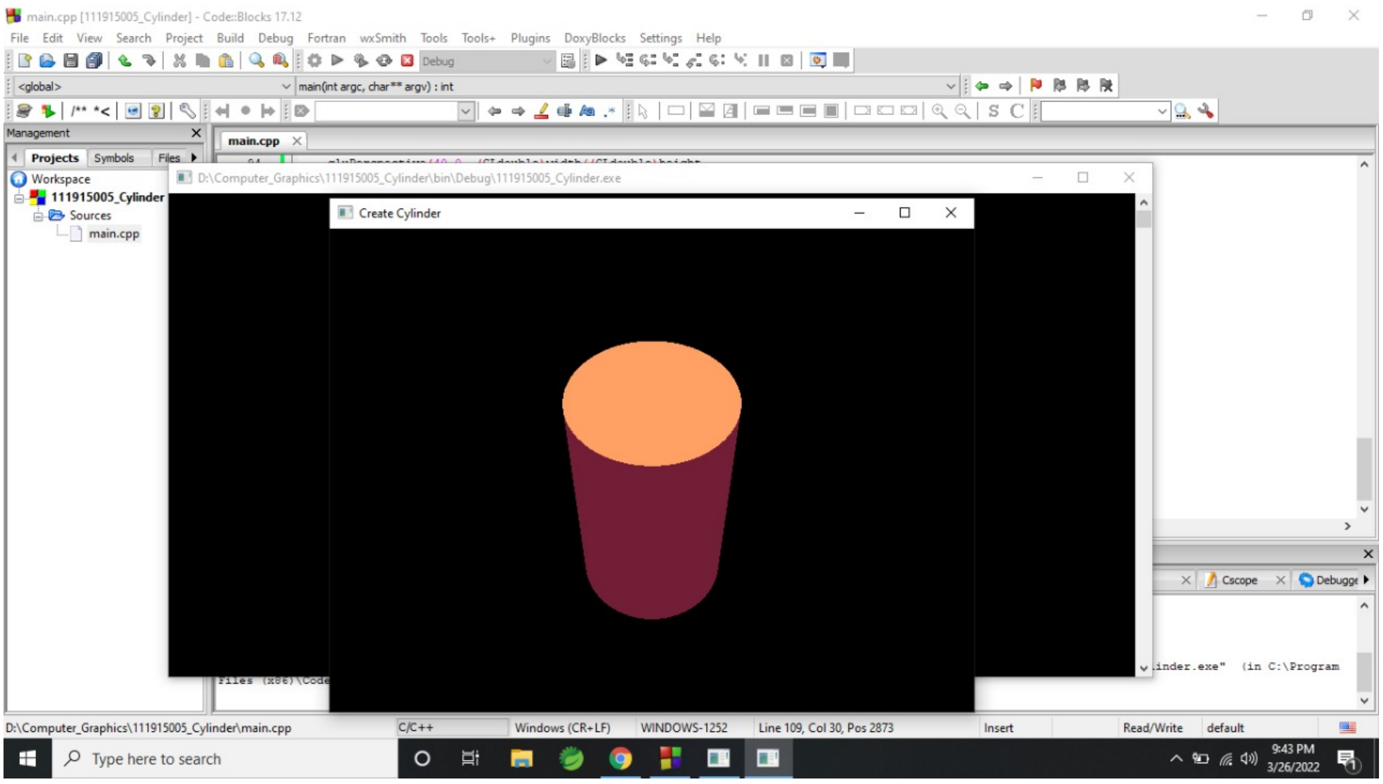


```
            rV[i][2] = V[i][2];
        }

ube(rV[0], rV[1], rV[2], rV[3], rV[4], rV[5], rV[6], rV[7]);

lutSwapBuffers();

ain(int C, char *V[])

rintf("Enter the Axis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis
canf("%d", &a);

lutInit(&C, V);
lutInitWindowSize(600, 600);
lutInitWindowPosition(50, 150);
```

```
pinning_3D_Models git:(main) ✗ g++ SpinningCube.c -lGL -lglut
pinning_3D_Models git:(main) ✗ ./a.out
xis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis ]: 2

pinning_3D_Models git:(main) ✗ g++ SpinningCube.c -lGL -lglut
→  OpenGL_Spinning_3D_Models git:(main) ✗ ./a.out
Enter the Axis of Rotation [ 1->Xaxis | 2->Yaxis | 3->Zaxis ]: 1
```

3D Model : Cylinder

```c
#include <stdio.h>
#include <stdlib.h>
#include <GL/glut.h>
#include <math.h>

#define PI 3.1415927

void draw_cylinder(GLfloat radius,
                   GLfloat height,
                   GLubyte R,
                   GLubyte G,
                   GLubyte B)
{
    GLfloat x = 0.0;
    GLfloat y = 0.0;
    GLfloat angle = 0.0;
    GLfloat angle_stepsize = 0.1;

    glColor3ub(R - 140, G - 130, B - 45);
    glBegin(GL_QUAD_STRIP);
    angle = 0.0;
    while (angle < 2 * PI)
    {
        x = radius * cos(angle);
        y = radius * sin(angle);
        glVertex3f(x, y, height);
        glVertex3f(x, y, 0.0);
        angle = angle + angle_stepsize;
    }
    glVertex3f(radius, 0.0, height);
    glVertex3f(radius, 0.0, 0.0);
    glEnd();

    glColor3ub(R, G, B);
    glBegin(GL_POLYGON);
    angle = 0.0;
    while (angle < 2 * PI)
    {
        x = radius * cos(angle);
        y = radius * sin(angle);
        glVertex3f(x, y, height);
        angle = angle + angle_stepsize;
    }
    glVertex3f(radius, 0.0, height);
    glEnd();
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();

    glTranslatef(0.0, -0.4, -3.0);
    glRotatef(-40, 1.0, 0.0, 0.0);

    draw_cylinder(0.3, 1.0, 255, 160, 100);

    glFlush();
}

void reshape(int width, int height)
{
    if (width == 0 || height == 0)
        return;

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(40.0, (GLdouble)width / (GLdouble)height,
                   0.5, 20.0);

    glMatrixMode(GL_MODELVIEW);
    glViewport(0, 0, width, height);
}

int main(int C, char *V[])
{
    glutInit(&C, V);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutCreateWindow("Create Cylinder");
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();

    return 0;
}
```


main.cpp [111915005_Cylinder] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
<global>
main(int argc, char** argv) : int
Management
Projects Symbols Files
Workspace
111915005_Cylinder
Sources
main.cpp
D:\Computer_Graphics\111915005_Cylinder\bin\Debug\111915005_Cylinder.exe
Create Cylinder
Cscope
Debugger
Type here to search
9:43 PM
3/26/2022

D:\Computer_Graphics\111915005_Cylinder\main.cpp   C/C++   Windows (CR+LF)   WINDOWS-1252   Line 109, Col 30, Pos 2873   Insert   Read/Write   default