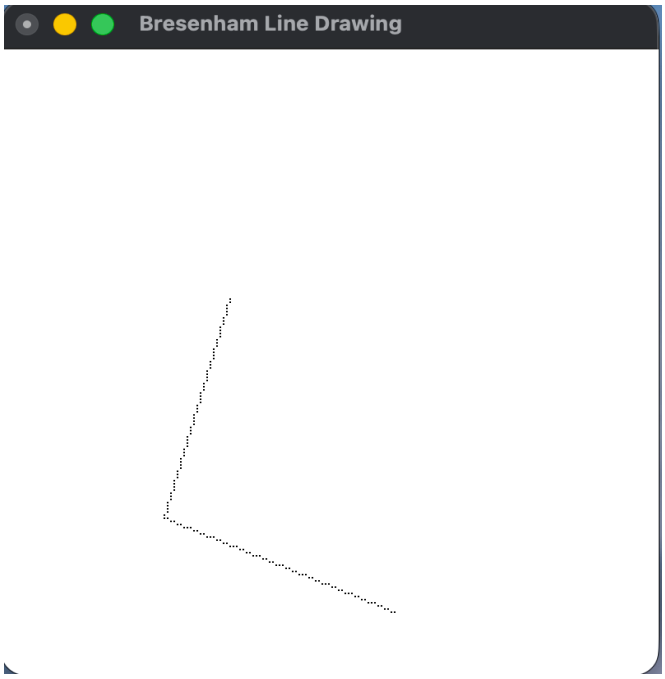
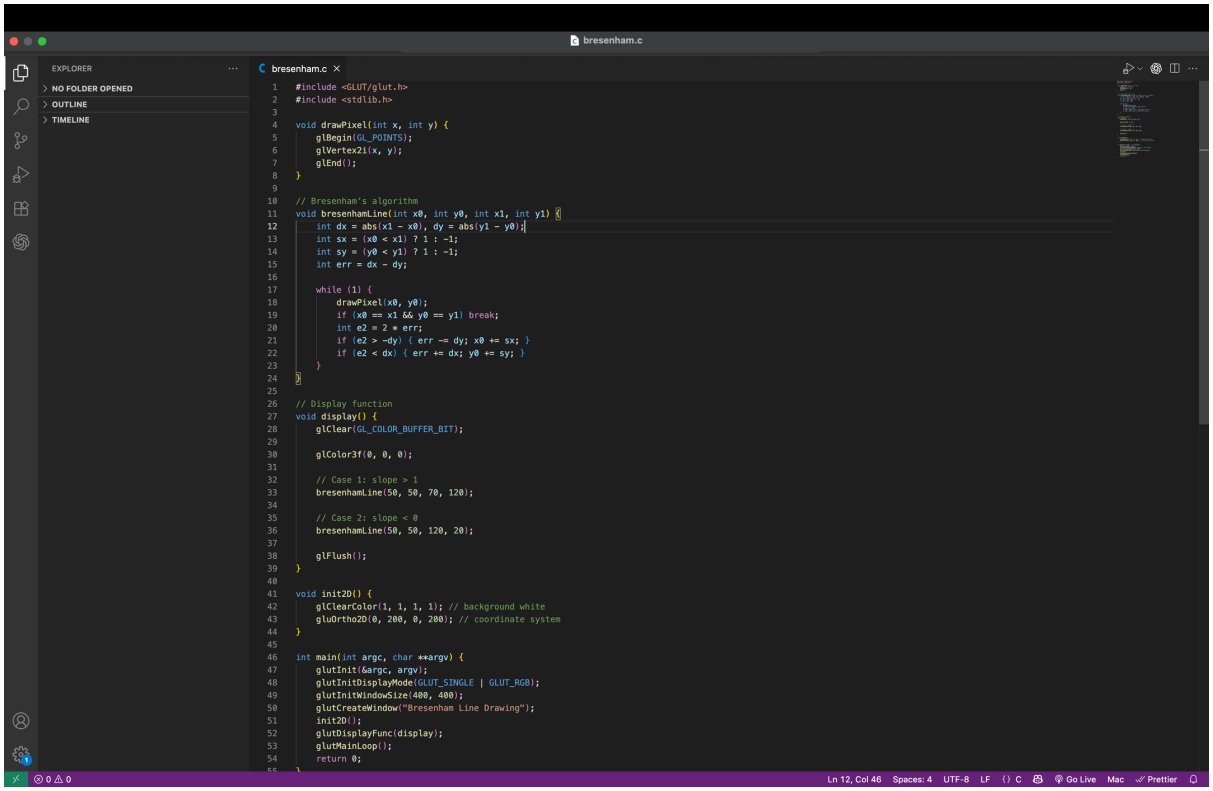
Task 1

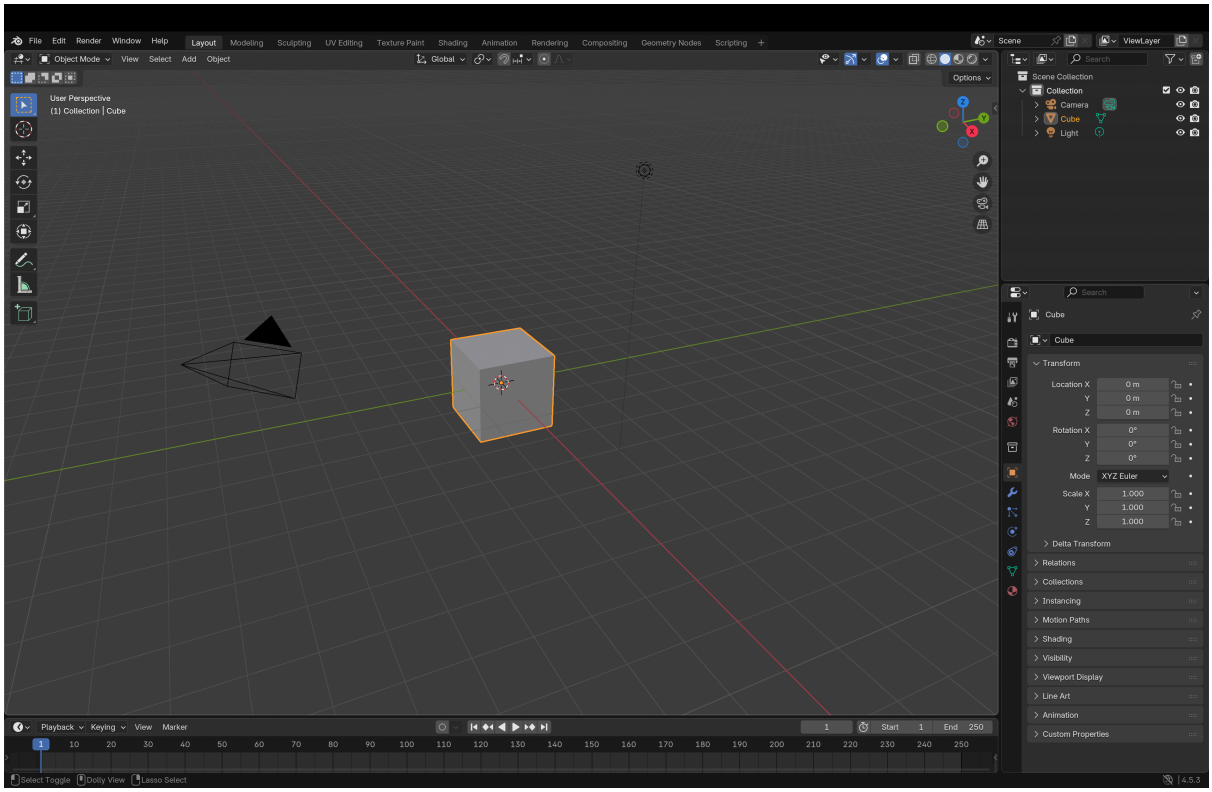

Bresenham Line Drawing


bresenham.c
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
bresenham.c
#include <GLUT/glut.h>
#include <stdlib.h>
void drawPixel(int x, int y) {
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
// Bresenham's algorithm
void bresenhamLine(int x0, int y0, int x1, int y1) {
int dx = abs(x1 - x0), dy = abs(y1 - y0);
int sx = (x0 < x1) ? 1 : -1;
int sy = (y0 < y1) ? 1 : -1;
int err = dx - dy;
while (1) {
drawPixel(x0, y0);
if (x0 == x1 && y0 == y1) break;
int e2 = 2 * err;
if (e2 > -dy) { err -= dy; x0 += sx; }
if (e2 < dx) { err += dx; y0 += sy; }
}
}
// Display function
void display() {
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(0, 0, 0);
// Case 1: slope > 1
bresenhamLine(50, 50, 70, 120);
// Case 2: slope < 0
bresenhamLine(50, 50, 120, 20);
glFlush();
}
void init2D() {
glClearColor(1, 1, 1, 1); // background white
gluOrtho2D(0, 200, 0, 200); // coordinate system
}
int main(int argc, char **argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(400, 400);
glutCreateWindow("Bresenham Line Drawing");
init2D();
glutDisplayFunc(display);
glutMainLoop();
return 0;
}
Ln 12, Col 46
Spaces: 4
UTF-8
LF
C
Go Live
Mac
Prettier

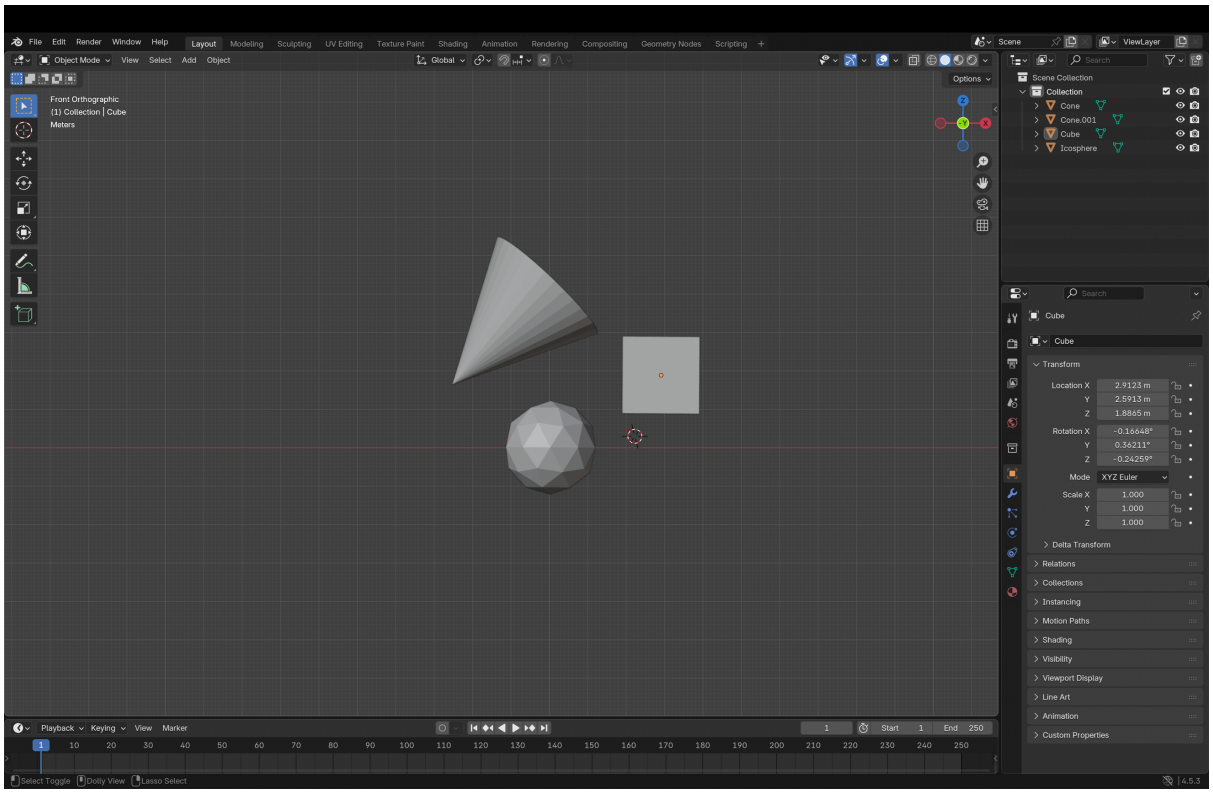
Observations:

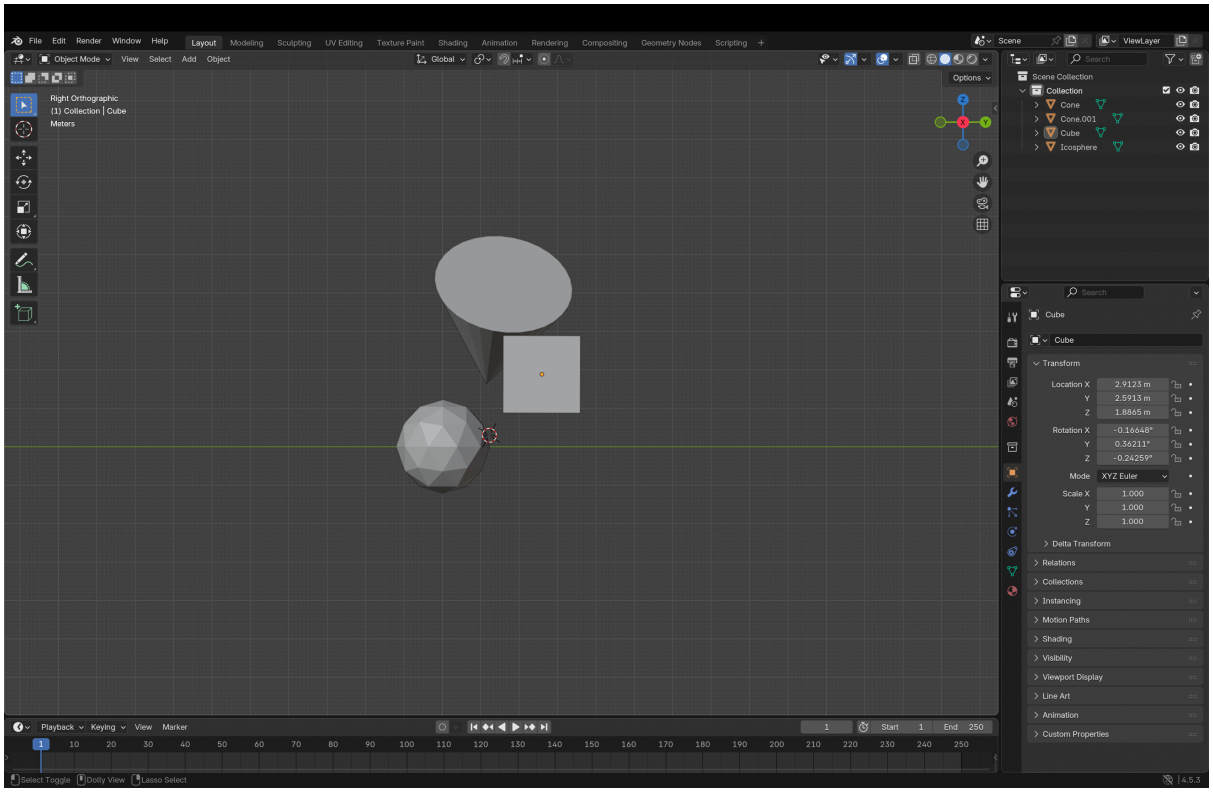
- *Slope > 1*: y changes faster than x; decision term triggers more y-steps; visually steeper; pixels look denser vertically.
- *Slope < 0*: y decrements while x increments; symmetric logic with sy = -1; line goes down-right; no gaps.
- Integer arithmetic gives crisp, aliased line; no floating-point used.

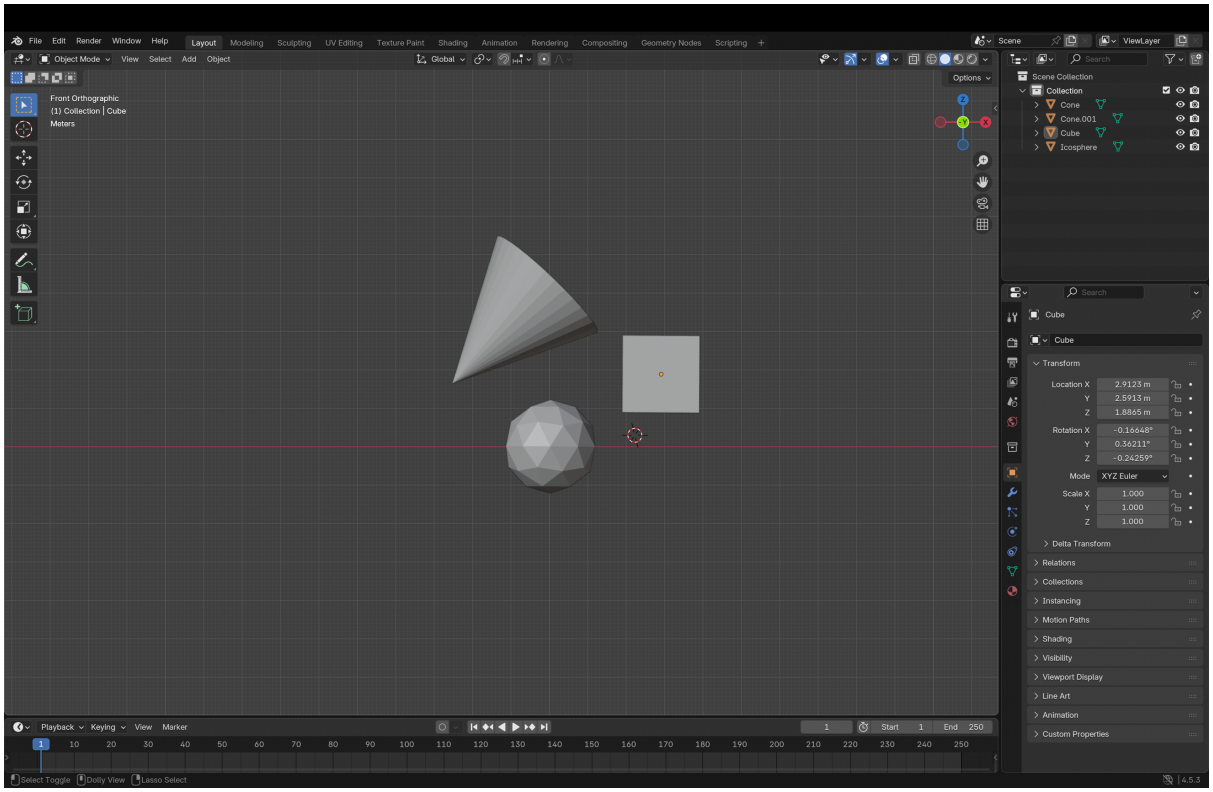
Task 2



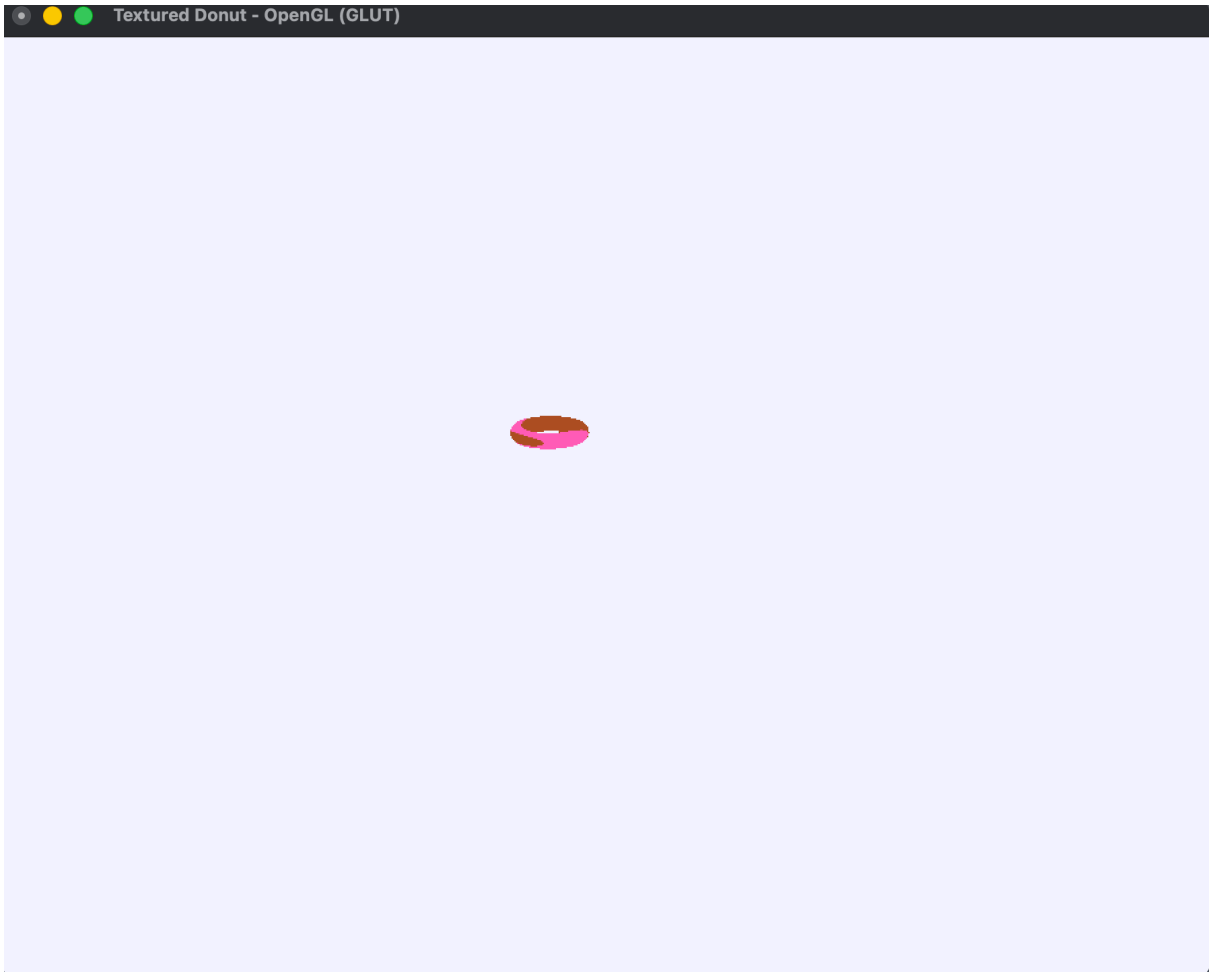
Task 3

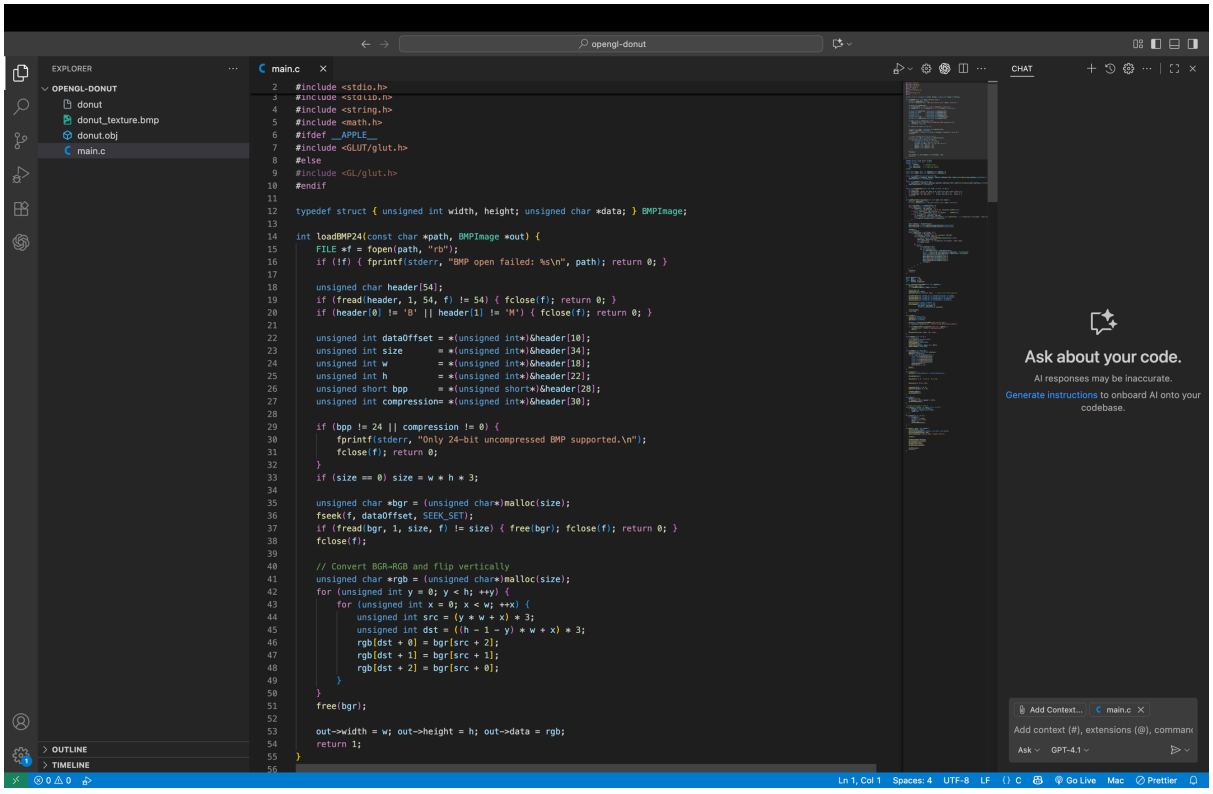

File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Options
Front Orthographic
(1) Collection | Cube
Meters
Scene
ViewLayer
Scene Collection
Collection
Cone
Cone.001
Cube
Icosphere
Search
Cube
Transform
Location X 2.9123 m
Y 2.5913 m
Z 1.8865 m
Rotation X -0.16648°
Y 0.36211°
Z -0.24259°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Shading
Visibility
Viewport Display
Line Art
Animation
Custom Properties
Playback Keying View Marker
Start 1 End 250
Select Toggle Dolly View Lasso Select
4.5.3

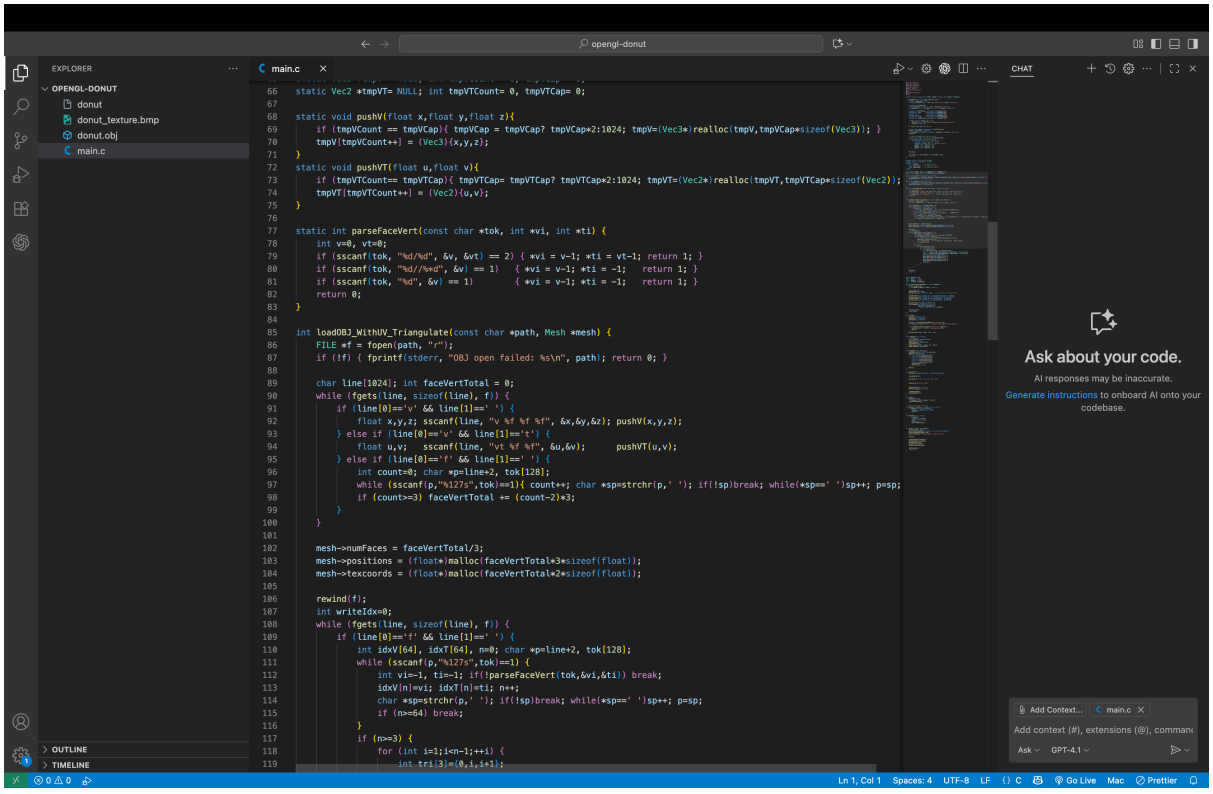

File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Options
Right Orthographic
(1) Collection | Cube
Meters
Scene
ViewLayer
Scene Collection
Collection
Cone
Cone.001
Cube
Icosphere
Search
Cube
Transform
Location X 2.9123 m
Y 2.5913 m
Z 1.8865 m
Rotation X -0.16648°
Y 0.36211°
Z -0.24259°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Shading
Visibility
Viewport Display
Line Art
Animation
Custom Properties
Playback Keying View Marker
Start 1 End 250
Select Toggle Dolly View Lasso Select
4.5.3

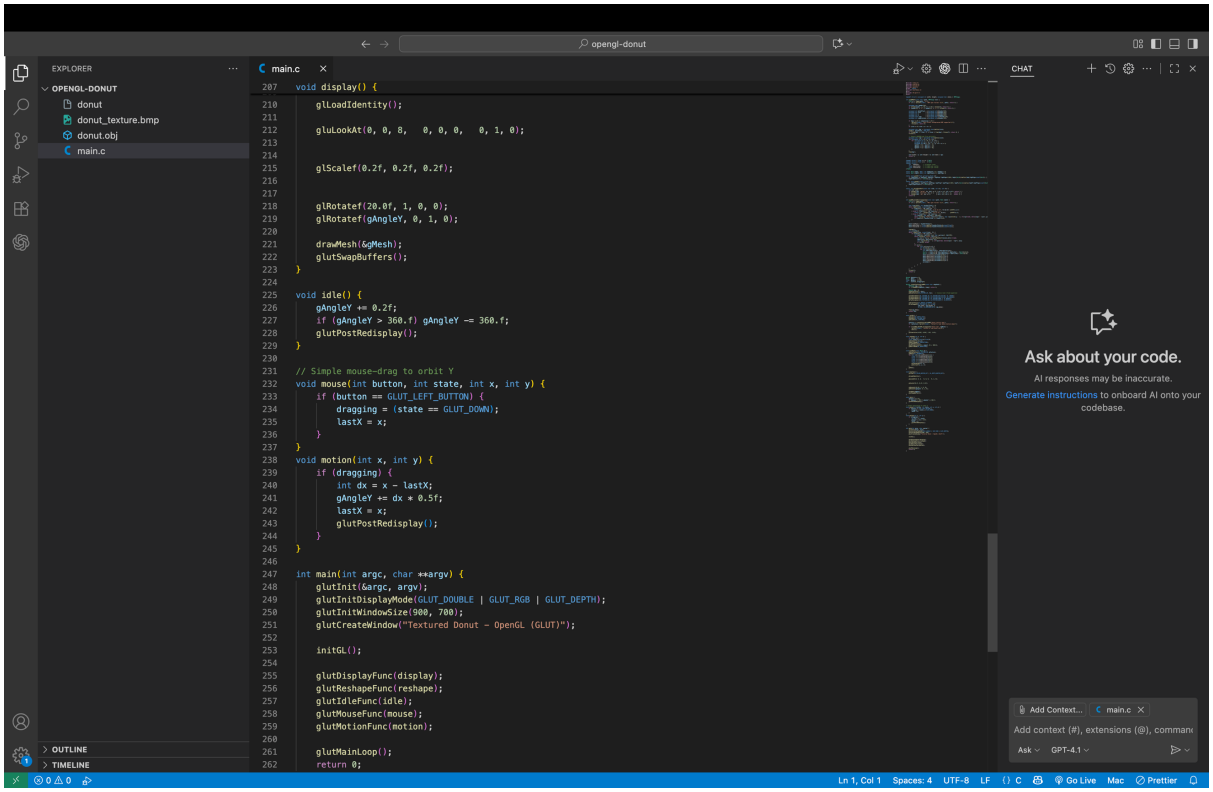

File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Options
Front Orthographic
(1) Collection | Cube
Meters
Scene
ViewLayer
Scene Collection
Collection
Cone
Cone.001
Cube
Icosphere
Cube
Transform
Location X 2.9123 m
Y 2.5913 m
Z 1.8865 m
Rotation X -0.16648°
Y 0.36211°
Z -0.24259°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Shading
Visibility
Viewport Display
Line Art
Animation
Custom Properties
Playback Keying View Marker
Start 1 End 250
Select Toggle Dolly View Lasso Select
4.5.3

Task 4

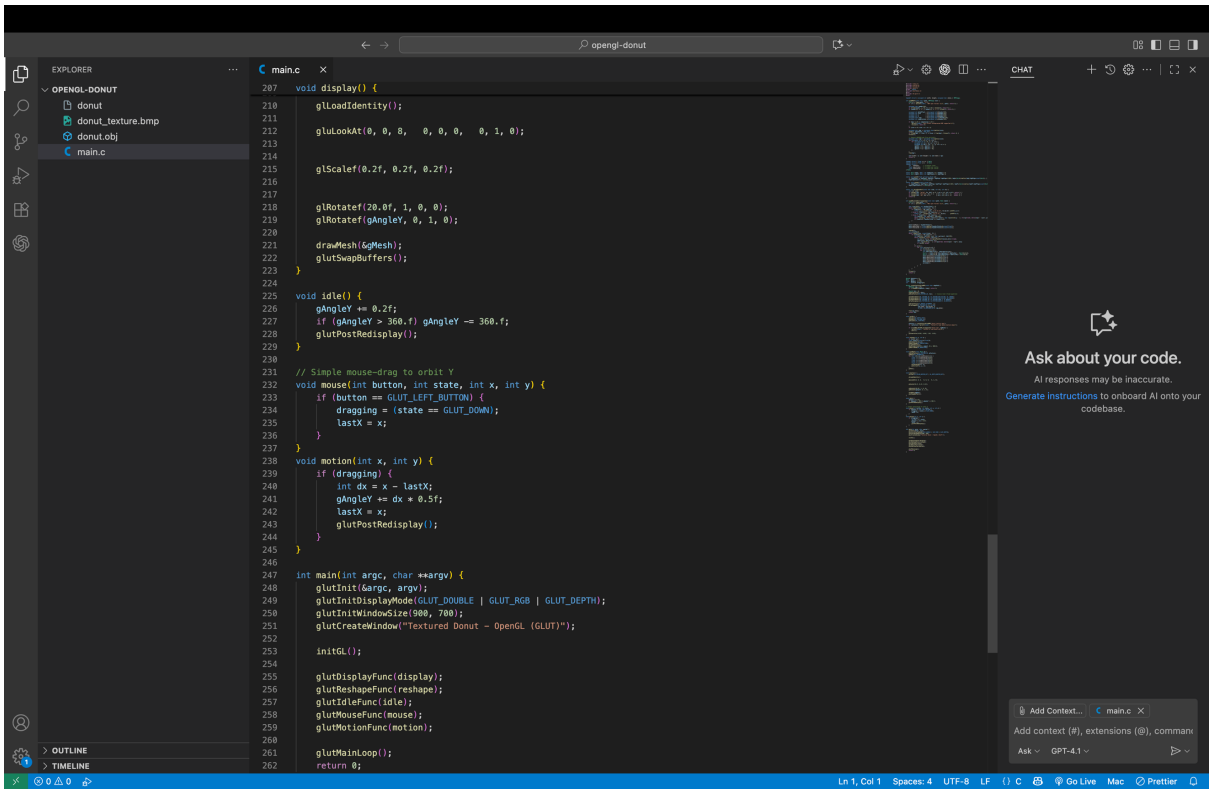



opengl-donut
EXPLORER
OPENGL-DONUT
donut
donut_texture.bmp
donut.obj
main.c
OUTLINE
TIMELINE
main.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
typedef struct { unsigned int width, height; unsigned char *data; } BMPImage;
int loadBMP24(const char *path, BMPImage *out) {
FILE *f = fopen(path, "rb");
if (!f) { fprintf(stderr, "BMP open failed: %s\n", path); return 0; }
unsigned char header[54];
if (fread(header, 1, 54, f) != 54) { fclose(f); return 0; }
if (header[0] != 'B' || header[1] != 'M') { fclose(f); return 0; }
unsigned int dataOffset = *(unsigned int*)&header[10];
unsigned int size = *(unsigned int*)&header[34];
unsigned int w = *(unsigned int*)&header[18];
unsigned int h = *(unsigned int*)&header[22];
unsigned short bpp = *(unsigned short*)&header[28];
unsigned int compression= *(unsigned int*)&header[30];
if (bpp != 24 || compression != 0) {
fprintf(stderr, "Only 24-bit uncompressed BMP supported.\n");
fclose(f); return 0;
}
if (size == 0) size = w * h * 3;
unsigned char *bgr = (unsigned char*)malloc(size);
fseek(f, dataOffset, SEEK_SET);
if (fread(bgr, 1, size, f) != size) { free(bgr); fclose(f); return 0; }
fclose(f);
// Convert BGR→RGB and flip vertically
unsigned char *rgb = (unsigned char*)malloc(size);
for (unsigned int y = 0; y < h; ++y) {
for (unsigned int x = 0; x < w; ++x) {
unsigned int src = (y * w + x) * 3;
unsigned int dst = ((h - 1 - y) * w + x) * 3;
rgb[dst + 0] = bgr[src + 2];
rgb[dst + 1] = bgr[src + 1];
rgb[dst + 2] = bgr[src + 0];
}
}
free(bgr);
out->width = w; out->height = h; out->data = rgb;
return 1;
}
CHAT
Ask about your code.
AI responses may be inaccurate.
Generate instructions to onboard AI onto your codebase.
Add Context...
main.c
Add context (#), extensions (@), comman
Ask
GPT-4.1
Ln 1, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Mac
Prettier


opengl-donut
EXPLORER
OPENGL-DONUT
donut
donut_texture.bmp
donut.obj
main.c
OUTLINE
TIMELINE
main.c
static Vec2 *tmpVT= NULL; int tmpVTCount= 0, tmpVTCap= 0;
static void pushV(float x,float y,float z){
if (tmpVCount == tmpVCap){ tmpVCap = tmpVCap? tmpVCap*2:1024; tmpV=(Vec3*)realloc(tmpV,tmpVCap*sizeof(Vec3)); }
tmpV[tmpVCount++] = (Vec3){x,y,z};
}
static void pushVT(float u,float v){
if (tmpVTCount== tmpVTCap){ tmpVTCap= tmpVTCap? tmpVTCap*2:1024; tmpVT=(Vec2*)realloc(tmpVT,tmpVTCap*sizeof(Vec2));
tmpVT[tmpVTCount++] = (Vec2){u,v};
}
static int parseFaceVert(const char *tok, int *vi, int *ti) {
int v=0, vt=0;
if (sscanf(tok, "%d/%d", &v, &vt) == 2) { *vi = v-1; *ti = vt-1; return 1; }
if (sscanf(tok, "%d//%*d", &v) == 1) { *vi = v-1; *ti = -1; return 1; }
if (sscanf(tok, "%d", &v) == 1) { *vi = v-1; *ti = -1; return 1; }
return 0;
}
int loadOBJ_WithUV_Triangulate(const char *path, Mesh *mesh) {
FILE *f = fopen(path, "r");
if (!f) { fprintf(stderr, "OBJ open failed: %s\n", path); return 0; }
char line[1024]; int faceVertTotal = 0;
while (fgets(line, sizeof(line), f)) {
if (line[0]=='v' && line[1]==' ') {
float x,y,z; sscanf(line, "v %f %f %f", &x,&y,&z); pushV(x,y,z);
} else if (line[0]=='v' && line[1]=='t') {
float u,v; sscanf(line, "vt %f %f", &u,&v); pushVT(u,v);
} else if (line[0]=='f' && line[1]==' ') {
int count=0; char *p=line+2, tok[128];
while (sscanf(p,"%127s",tok)==1){ count++; char *sp=strchr(p,' '); if(!sp)break; while(*sp==' ')sp++; p=sp;
if (count>=3) faceVertTotal += (count-2)*3;
}
}
mesh->numFaces = faceVertTotal/3;
mesh->positions = (float*)malloc(faceVertTotal*3*sizeof(float));
mesh->texcoords = (float*)malloc(faceVertTotal*2*sizeof(float));
rewind(f);
int writeIdx=0;
while (fgets(line, sizeof(line), f)) {
if (line[0]=='f' && line[1]==' ') {
int idxV[64], idxT[64], n=0; char *p=line+2, tok[128];
while (sscanf(p,"%127s",tok)==1) {
int vi=-1, ti=-1; if(!parseFaceVert(tok,&vi,&ti)) break;
idxV[n]=vi; idxT[n]=ti; n++;
char *sp=strchr(p,' '); if(!sp)break; while(*sp==' ')sp++; p=sp;
if (n>=64) break;
}
if (n>=3) {
for (int i=1;i<n-1;++i) {
int tri[3]={0,i,i+1};
CHAT
Ask about your code.
AI responses may be inaccurate.
Generate instructions to onboard AI onto your codebase.
Add Context...
main.c
Add context (#), extensions (@), comman
Ask
GPT-4.1
Ln 1, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Mac
Prettier




```c
void display() {
    glLoadIdentity();

    gluLookAt(0, 0, 8,   0, 0, 0,   0, 1, 0);


    glScalef(0.2f, 0.2f, 0.2f);


    glRotatef(20.0f, 1, 0, 0);
    glRotatef(gAngleY, 0, 1, 0);

    drawMesh(&gMesh);
    glutSwapBuffers();
}

void idle() {
    gAngleY += 0.2f;
    if (gAngleY > 360.f) gAngleY -= 360.f;
    glutPostRedisplay();
}

// Simple mouse-drag to orbit Y
void mouse(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON) {
        dragging = (state == GLUT_DOWN);
        lastX = x;
    }
}
void motion(int x, int y) {
    if (dragging) {
        int dx = x - lastX;
        gAngleY += dx * 0.5f;
        lastX = x;
        glutPostRedisplay();
    }
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(900, 700);
    glutCreateWindow("Textured Donut — OpenGL (GLUT)");

    initGL();

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutIdleFunc(idle);
    glutMouseFunc(mouse);
    glutMotionFunc(motion);

    glutMainLoop();
    return 0;
```




```c
void display() {
    glLoadIdentity();

    gluLookAt(0, 0, 8,   0, 0, 0,   0, 1, 0);


    glScalef(0.2f, 0.2f, 0.2f);


    glRotatef(20.0f, 1, 0, 0);
    glRotatef(gAngleY, 0, 1, 0);

    drawMesh(&gMesh);
    glutSwapBuffers();
}

void idle() {
    gAngleY += 0.2f;
    if (gAngleY > 360.f) gAngleY -= 360.f;
    glutPostRedisplay();
}

// Simple mouse-drag to orbit Y
void mouse(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON) {
        dragging = (state == GLUT_DOWN);
        lastX = x;
    }
}
void motion(int x, int y) {
    if (dragging) {
        int dx = x - lastX;
        gAngleY += dx * 0.5f;
        lastX = x;
        glutPostRedisplay();
    }
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(900, 700);
    glutCreateWindow("Textured Donut — OpenGL (GLUT)");

    initGL();

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutIdleFunc(idle);
    glutMouseFunc(mouse);
    glutMotionFunc(motion);

    glutMainLoop();
    return 0;
```