Chavangkon Kanjanakuldit (Ken) 6681371

<https://github.com/Ken19149/Graphics/tree/main/week_7>

#include <GL/glut.h>

#include <stdio.h>

#include <stdlib.h>

#include <cmath>

GLuint textureID;

// Create a simple checkerboard texture programmatically

void createCheckerboardTexture(int x)

{

const int texSize = (int) pow(2, x);

GLubyte texture[texSize][texSize][3];

for (int i = 0; i < texSize; i++)

{

for (int j = 0; j < texSize; j++)

{

// Create checkerboard pattern

int checkSize = 8;

int pattern = ((i / checkSize) + (j / checkSize)) % 2;

GLubyte color = pattern ? 255 : 128; // White and gray

texture[i][j][0] = color; // Red

texture[i][j][1] = color; // Green

texture[i][j][2] = pattern ? 255 : 0; // Blue (adds some color variation)

}

}

glGenTextures(1, &textureID);

glBindTexture(GL\_TEXTURE\_2D, textureID);

glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGB, texSize, texSize, 0, GL\_RGB, GL\_UNSIGNED\_BYTE, texture);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

}

// Function to draw a textured cube manually

void drawTexturedCube()

{

glBindTexture(GL\_TEXTURE\_2D, textureID);

glBegin(GL\_QUADS);

// Front face

glTexCoord2f(0.0f, 0.0f);

glVertex3f(-0.5f, -0.5f, 0.5f);

glTexCoord2f(2.0f, 0.0f);

glVertex3f(0.5f, -0.5f, 0.5f);

glTexCoord2f(2.0f, 2.0f);

glVertex3f(0.5f, 0.5f, 0.5f);

glTexCoord2f(0.0f, 2.0f);

glVertex3f(-0.5f, 0.5f, 0.5f);

// Back face

glTexCoord2f(2.0f, 0.0f);

glVertex3f(-0.5f, -0.5f, -0.5f);

glTexCoord2f(2.0f, 2.0f);

glVertex3f(-0.5f, 0.5f, -0.5f);

glTexCoord2f(0.0f, 2.0f);

glVertex3f(0.5f, 0.5f, -0.5f);

glTexCoord2f(0.0f, 0.0f);

glVertex3f(0.5f, -0.5f, -0.5f);

// Top face

glTexCoord2f(0.0f, 2.0f);

glVertex3f(-0.5f, 0.5f, -0.5f);

glTexCoord2f(0.0f, 0.0f);

glVertex3f(-0.5f, 0.5f, 0.5f);

glTexCoord2f(2.0f, 0.0f);

glVertex3f(0.5f, 0.5f, 0.5f);

glTexCoord2f(2.0f, 2.0f);

glVertex3f(0.5f, 0.5f, -0.5f);

// Bottom face

glTexCoord2f(2.0f, 2.0f);

glVertex3f(-0.5f, -0.5f, -0.5f);

glTexCoord2f(0.0f, 2.0f);

glVertex3f(0.5f, -0.5f, -0.5f);

glTexCoord2f(0.0f, 0.0f);

glVertex3f(0.5f, -0.5f, 0.5f);

glTexCoord2f(2.0f, 0.0f);

glVertex3f(-0.5f, -0.5f, 0.5f);

// Right face

glTexCoord2f(2.0f, 0.0f);

glVertex3f(0.5f, -0.5f, -0.5f);

glTexCoord2f(2.0f, 2.0f);

glVertex3f(0.5f, 0.5f, -0.5f);

glTexCoord2f(0.0f, 2.0f);

glVertex3f(0.5f, 0.5f, 0.5f);

glTexCoord2f(0.0f, 0.0f);

glVertex3f(0.5f, -0.5f, 0.5f);

// Left face

glTexCoord2f(0.0f, 0.0f);

glVertex3f(-0.5f, -0.5f, -0.5f);

glTexCoord2f(2.0f, 0.0f);

glVertex3f(-0.5f, -0.5f, 0.5f);

glTexCoord2f(2.0f, 2.0f);

glVertex3f(-0.5f, 0.5f, 0.5f);

glTexCoord2f(0.0f, 2.0f);

glVertex3f(-0.5f, 0.5f, -0.5f);

glEnd();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity();

// Set camera position

gluLookAt(3.0, 3.0, 3.0, // Eye position

0.0, 0.0, 0.0, // Look at point

0.0, 1.0, 0.0); // Up vector

// Draw textured cube in the center

glEnable(GL\_TEXTURE\_2D);

glColor3f(1.0f, 1.0f, 1.0f);

drawTexturedCube();

glDisable(GL\_TEXTURE\_2D);

// Draw additional colored cubes around the center

glEnable(GL\_TEXTURE\_2D);

glPushMatrix();

glTranslatef(2.0f, 0.0f, 0.0f);

glColor3f(0.0f, 1.0f, 0.0f); // Green color

glutSolidCube(0.5);

drawTexturedCube();

glPopMatrix();

glDisable(GL\_TEXTURE\_2D);

glPushMatrix();

glTranslatef(-2.0f, 0.0f, 0.0f);

glColor3f(0.0f, 0.0f, 1.0f); // Blue color

glutSolidCube(1.0);

drawTexturedCube();

glPopMatrix();

glPushMatrix();

glTranslatef(0.0f, 2.0f, 0.0f);

glColor3f(1.0f, 1.0f, 0.0f); // Yellow color

glutSolidCube(0.5);

drawTexturedCube();

glPopMatrix();

glPushMatrix();

glTranslatef(0.0f, -2.0f, 0.0f);

glColor3f(1.0f, 0.0f, 1.0f); // Magenta color

drawTexturedCube();

glutSolidCube(0.5);

glPopMatrix();

glutSwapBuffers();

}

void reshape(int w, int h)

{

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0, (double)w / (double)h, 1.0, 20.0);

glMatrixMode(GL\_MODELVIEW);

}

void keyboard(unsigned char key, int x, int y)

{

switch (key)

{

case 27: // ESC key

exit(0);

break;

}

}

void init()

{

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glEnable(GL\_DEPTH\_TEST);

// Enable lighting

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

// Set light properties

GLfloat lightPos[] = {5.0f, 5.0f, 5.0f, 1.0f};

GLfloat lightAmbient[] = {0.3f, 0.3f, 0.3f, 1.0f};

GLfloat lightDiffuse[] = {0.8f, 0.8f, 0.8f, 1.0f};

glLightfv(GL\_LIGHT0, GL\_POSITION, lightPos);

glLightfv(GL\_LIGHT0, GL\_AMBIENT, lightAmbient);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, lightDiffuse);

// Enable color material

glEnable(GL\_COLOR\_MATERIAL);

glColorMaterial(GL\_FRONT, GL\_AMBIENT\_AND\_DIFFUSE);

// Create checkerboard texture

createCheckerboardTexture(8);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize(600, 600);

glutCreateWindow("GLUT Textured and Colored Cubes - Checkerboard Texture");

init();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

printf("Textured Cube with Checkerboard Pattern\n");

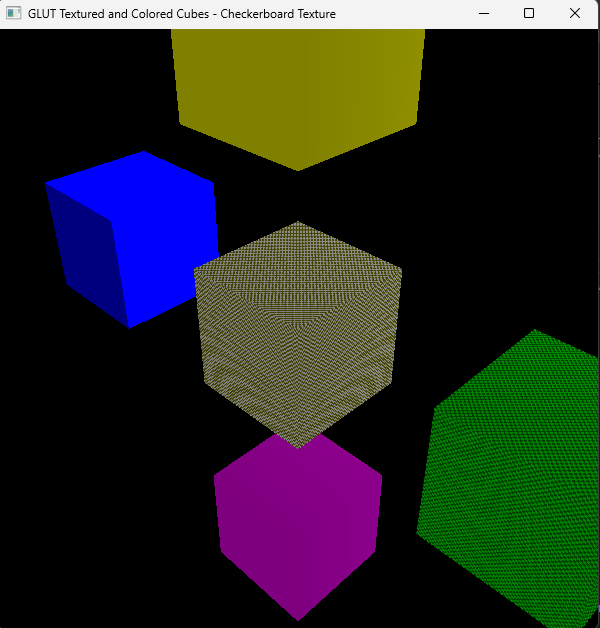
printf("Press ESC to exit\n");

glutMainLoop();

return 0;

}

Output:



I wanted to try to create a different grid size for each cube but it wasn’t a successful attempt. At first I experimented with the texSize variable and I noticed that the value is in power of 2. So I tried changing the value and noticed that it cannot be lower than 16 because I tried 2^3 and it didn’t work. Another value I tried is 2^10 which is 1024. I’m guessing that it’s bigger than the value that the variable can handle, but 2^8 works well. Anyway, I’ll try experimenting with it more and hopefully make it functional.