

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Building()

{

glBegin(GL\_POLYGON);

glColor3ub(182, 117, 19);

glVertex2f(-9,-6);

glVertex2f(-9,7);

glVertex2f(-4,7);

glVertex2f(-4,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(208, 132, 35);

glVertex2f(-8.7,-6);

glVertex2f(-8.7,6.7);

glVertex2f(-4.3,6.7);

glVertex2f(-4.3,-6);

glEnd();

//5

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,5);

glVertex2f(-8,6);

glVertex2f(-7,6);

glVertex2f(-7,5);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,5);

glVertex2f(-6,6);

glVertex2f(-5,6);

glVertex2f(-5,5);

glEnd();

//4

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,3);

glVertex2f(-8,4);

glVertex2f(-7,4);

glVertex2f(-7,3);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,3);

glVertex2f(-6,4);

glVertex2f(-5,4);

glVertex2f(-5,3);

glEnd();

//3

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,2);

glVertex2f(-8,1);

glVertex2f(-7,1);

glVertex2f(-7,2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,1);

glVertex2f(-6,2);

glVertex2f(-5,2);

glVertex2f(-5,1);

glEnd();

//2

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,-1);

glVertex2f(-8,0);

glVertex2f(-7,0);

glVertex2f(-7,-1);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,-1);

glVertex2f(-6,0);

glVertex2f(-5,0);

glVertex2f(-5,-1);

glEnd();

//1

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,-3);

glVertex2f(-8,-2);

glVertex2f(-7,-2);

glVertex2f(-7,-3);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,-3);

glVertex2f(-6,-2);

glVertex2f(-5,-2);

glVertex2f(-5,-3);

glEnd();

//door

glBegin(GL\_POLYGON);

glColor3ub(57, 34, 6);

glVertex2f(-8,-6);

glVertex2f(-8,-4);

glVertex2f(-5,-4);

glVertex2f(-5,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(139, 82, 7);

glVertex2f(-7.7,-6);

glVertex2f(-7.7,-4.3);

glVertex2f(-5.3,-4.3);

glVertex2f(-5.3,-6);

glEnd();

}

void display()

{

glClearColor(0,0,0,0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Building();

glFlush();

}

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

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

gluOrtho2D(-20,20,-20,20);

glutMainLoop();

return 0;

}

A computer screen shot of a building

Description automatically generated

A graphing diagram with circles and dots

Description automatically generated

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Building\_Tree()

{

glBegin(GL\_POLYGON);

glColor3ub(128, 75, 7);

glVertex2f(0,-5);

glVertex2f(0,2);

glVertex2f(2,2);

glVertex2f(2,-5);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(128, 75, 7);

glVertex2f(-1,-6);

glVertex2f(0,-5);

glVertex2f(2,-5);

glVertex2f(3,-6);

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=6.73-3.79;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+1.18,y+3.79);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=8.54-5.87;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+1.35,y+5.87);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=6.77-4;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x-0.95,y+4);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=6.84-4;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+4,y+4);

}

glEnd();

}

void display()

{

glClearColor(0,0,0,0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Building\_Tree();

glFlush();

}

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

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

gluOrtho2D(-20,20,-20,20);

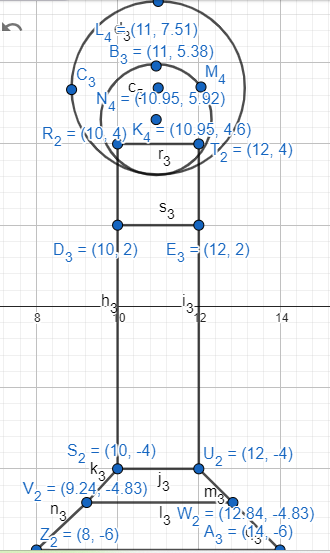
glutMainLoop();

return 0;

}

A screenshot of a computer screen

Description automatically generated



#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Building\_Tree\_Lamp()

{

glBegin(GL\_POLYGON);

glColor3ub(55, 54, 54);

glVertex2f(10,-4);

glVertex2f(10,2);

glVertex2f(12,2);

glVertex2f(12,-4);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(66, 66, 65);

glVertex2f(10,2);

glVertex2f(10,4);

glVertex2f(12,4);

glVertex2f(12,2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(60, 59, 59);

glVertex2f(9.24,-4.83);

glVertex2f(10,-4);

glVertex2f(12,-4);

glVertex2f(12.84,-4.83);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(60, 59, 59);

glVertex2f(8,-6);

glVertex2f(9.24,-4.83);

glVertex2f(12.84,-4.83);

glVertex2f(14,-6);

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(246, 86, 122);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=7.51-5.38;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+11,y+5.38);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(252, 219, 226);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=5.92-4.6;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+10.95,y+4.6);

}

glEnd();

}

void display()

{

glClearColor(0,0,0,0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Building\_Tree\_Lamp();

glFlush();

}

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

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

gluOrtho2D(-20,20,-20,20);

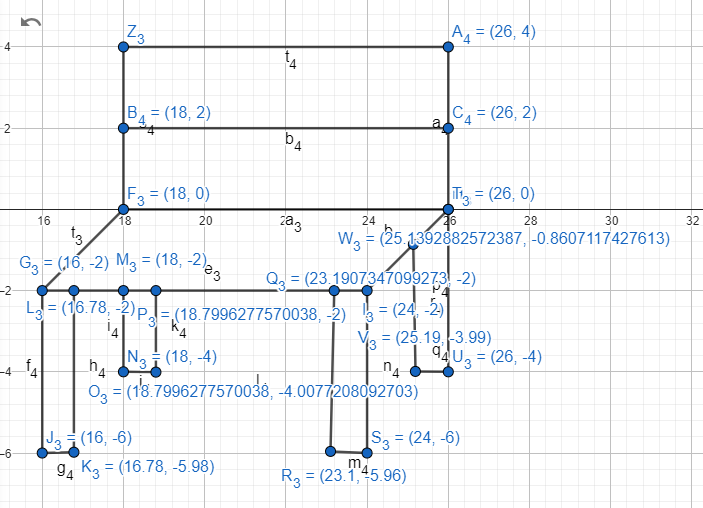
glutMainLoop();

return 0;

}

A screenshot of a video game

Description automatically generated



#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Building\_Tree\_Lamp\_Bench()

{

glBegin(GL\_POLYGON);

glColor3ub(166, 125, 66);

glVertex2f(16,-2);

glVertex2f(18,0);

glVertex2f(26,0);

glVertex2f(24,-2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(182, 137, 71);

glVertex2f(18,2);

glVertex2f(18,4);

glVertex2f(26,4);

glVertex2f(26,2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(184, 141, 78);

glVertex2f(18,0);

glVertex2f(18,2);

glVertex2f(26,2);

glVertex2f(26,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(16,-6);

glVertex2f(16,-2);

glVertex2f(16.78,-2);

glVertex2f(16.78,-5.98);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(18,-4);

glVertex2f(18,-2);

glVertex2f(18.79,-2);

glVertex2f(18.79,-4);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(23.1,-5.96);

glVertex2f(23.1,-2);

glVertex2f(24,-2);

glVertex2f(24,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(25.19,-3.99);

glVertex2f(25.19,-0.86);

glVertex2f(26,0);

glVertex2f(26,-4);

glEnd();

}

void display()

{

glClearColor(0,0,0,0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Building\_Tree\_Lamp\_Bench();

glFlush();

}

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

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL");

glutInitWindowSize(320,320);

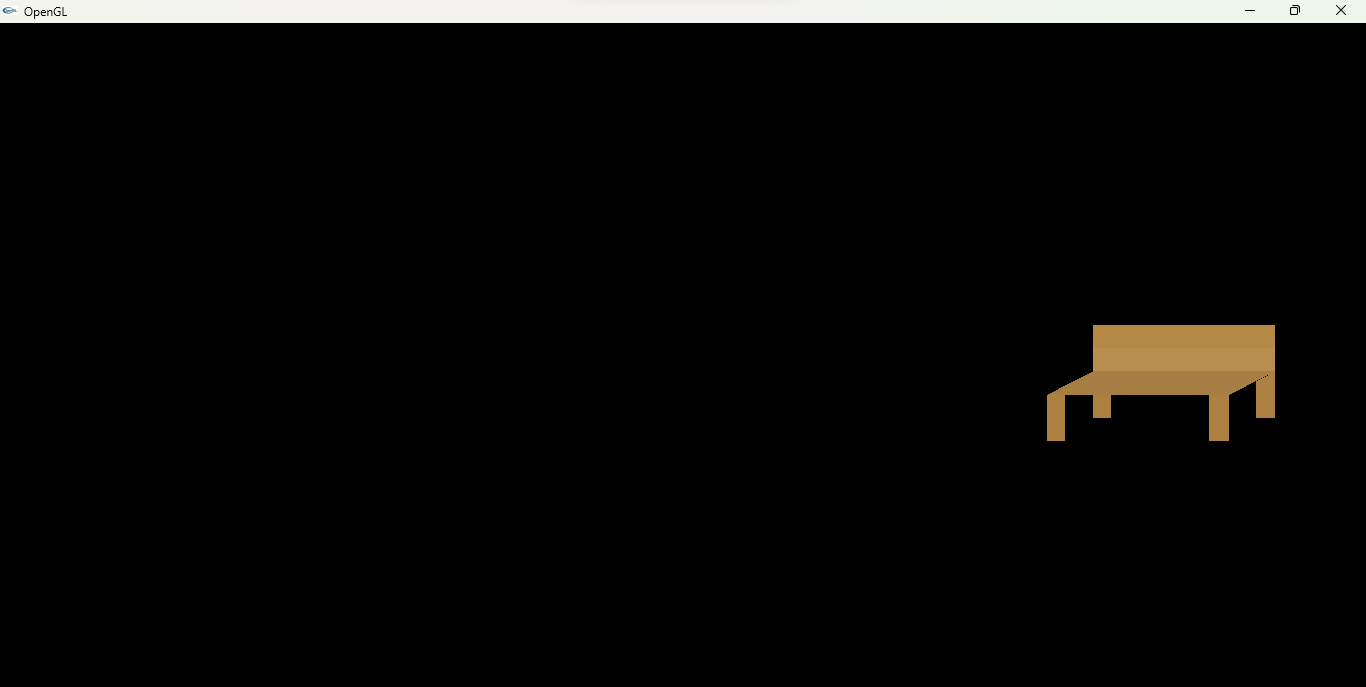
glutDisplayFunc(display);

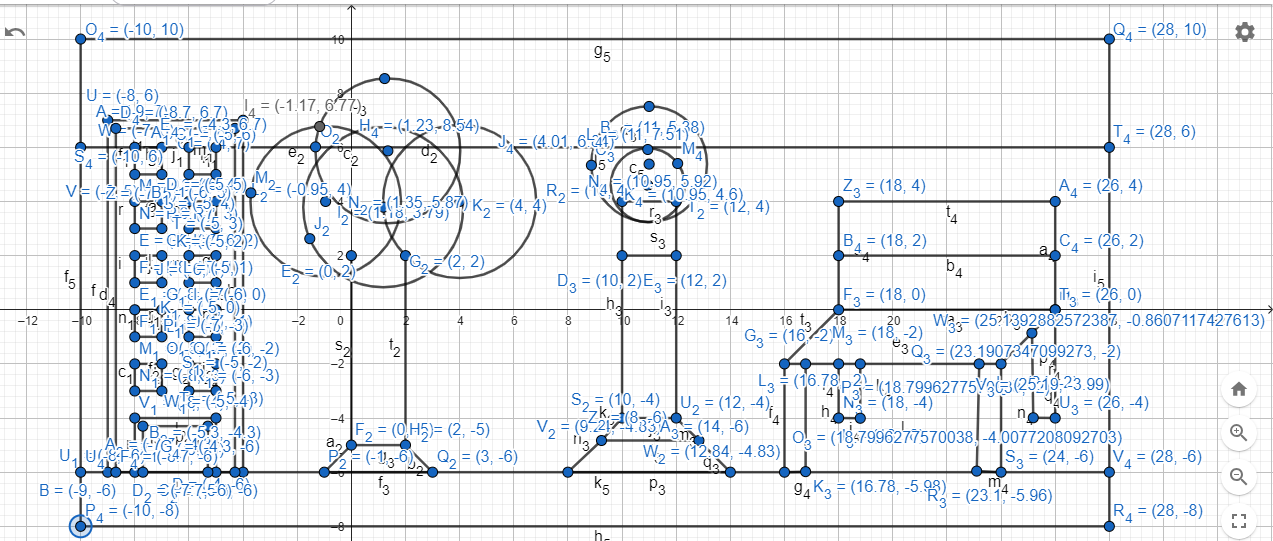
gluOrtho2D(-30,30,-30,30);

glutMainLoop();

return 0;

}





#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Building\_Tree\_Lamp\_Bench()

{

glBegin(GL\_POLYGON);

glColor3ub(88, 84, 80);

glVertex2f(-10,6);

glVertex2f(-10,10);

glVertex2f(28,10);

glVertex2f(28,6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(67, 64, 61);

glVertex2f(-10,-8);

glVertex2f(-10,-6);

glVertex2f(28,-6);

glVertex2f(28,-8);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(182, 117, 19);

glVertex2f(-9,-6);

glVertex2f(-9,7);

glVertex2f(-4,7);

glVertex2f(-4,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(208, 132, 35);

glVertex2f(-8.7,-6);

glVertex2f(-8.7,6.7);

glVertex2f(-4.3,6.7);

glVertex2f(-4.3,-6);

glEnd();

//5

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,5);

glVertex2f(-8,6);

glVertex2f(-7,6);

glVertex2f(-7,5);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,5);

glVertex2f(-6,6);

glVertex2f(-5,6);

glVertex2f(-5,5);

glEnd();

//4

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,3);

glVertex2f(-8,4);

glVertex2f(-7,4);

glVertex2f(-7,3);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,3);

glVertex2f(-6,4);

glVertex2f(-5,4);

glVertex2f(-5,3);

glEnd();

//3

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,2);

glVertex2f(-8,1);

glVertex2f(-7,1);

glVertex2f(-7,2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,1);

glVertex2f(-6,2);

glVertex2f(-5,2);

glVertex2f(-5,1);

glEnd();

//2

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,-1);

glVertex2f(-8,0);

glVertex2f(-7,0);

glVertex2f(-7,-1);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,-1);

glVertex2f(-6,0);

glVertex2f(-5,0);

glVertex2f(-5,-1);

glEnd();

//1

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-8,-3);

glVertex2f(-8,-2);

glVertex2f(-7,-2);

glVertex2f(-7,-3);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(38, 223, 143);

glVertex2f(-6,-3);

glVertex2f(-6,-2);

glVertex2f(-5,-2);

glVertex2f(-5,-3);

glEnd();

//door

glBegin(GL\_POLYGON);

glColor3ub(57, 34, 6);

glVertex2f(-8,-6);

glVertex2f(-8,-4);

glVertex2f(-5,-4);

glVertex2f(-5,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(139, 82, 7);

glVertex2f(-7.7,-6);

glVertex2f(-7.7,-4.3);

glVertex2f(-5.3,-4.3);

glVertex2f(-5.3,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(128, 75, 7);

glVertex2f(0,-5);

glVertex2f(0,2);

glVertex2f(2,2);

glVertex2f(2,-5);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(128, 75, 7);

glVertex2f(-1,-6);

glVertex2f(0,-5);

glVertex2f(2,-5);

glVertex2f(3,-6);

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=6.73-3.79;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+1.18,y+3.79);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=8.54-5.87;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+1.35,y+5.87);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=6.77-4;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x-0.95,y+4);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(40, 161, 32);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=6.84-4;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+4,y+4);

}

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(55, 54, 54);

glVertex2f(10,-4);

glVertex2f(10,2);

glVertex2f(12,2);

glVertex2f(12,-4);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(66, 66, 65);

glVertex2f(10,2);

glVertex2f(10,4);

glVertex2f(12,4);

glVertex2f(12,2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(60, 59, 59);

glVertex2f(9.24,-4.83);

glVertex2f(10,-4);

glVertex2f(12,-4);

glVertex2f(12.84,-4.83);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(60, 59, 59);

glVertex2f(8,-6);

glVertex2f(9.24,-4.83);

glVertex2f(12.84,-4.83);

glVertex2f(14,-6);

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(246, 86, 122);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=7.51-5.38;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+11,y+5.38);

}

glEnd();

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(252, 219, 226);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=5.92-4.6;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+10.95,y+4.6);

}

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(166, 125, 66);

glVertex2f(16,-2);

glVertex2f(18,0);

glVertex2f(26,0);

glVertex2f(24,-2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(182, 137, 71);

glVertex2f(18,2);

glVertex2f(18,4);

glVertex2f(26,4);

glVertex2f(26,2);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(184, 141, 78);

glVertex2f(18,0);

glVertex2f(18,2);

glVertex2f(26,2);

glVertex2f(26,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(16,-6);

glVertex2f(16,-2);

glVertex2f(16.78,-2);

glVertex2f(16.78,-5.98);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(18,-4);

glVertex2f(18,-2);

glVertex2f(18.79,-2);

glVertex2f(18.79,-4);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(23.1,-5.96);

glVertex2f(23.1,-2);

glVertex2f(24,-2);

glVertex2f(24,-6);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(171, 128, 64);

glVertex2f(25.19,-3.99);

glVertex2f(25.19,-0.86);

glVertex2f(26,0);

glVertex2f(26,-4);

glEnd();

}

void display()

{

glClearColor(0,0,0,0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Building\_Tree\_Lamp\_Bench();

glFlush();

}

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

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

gluOrtho2D(-10,28,-8,10);

glutMainLoop();

return 0;

}

A group of trees and chairs

Description automatically generated