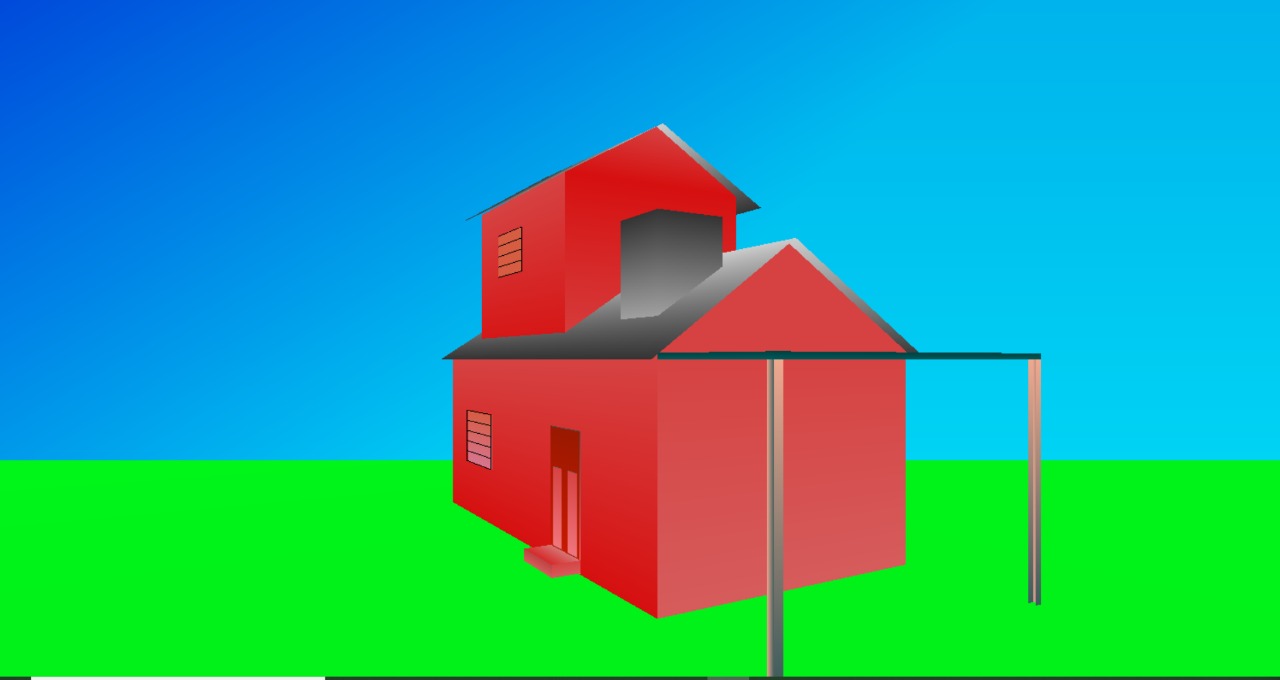
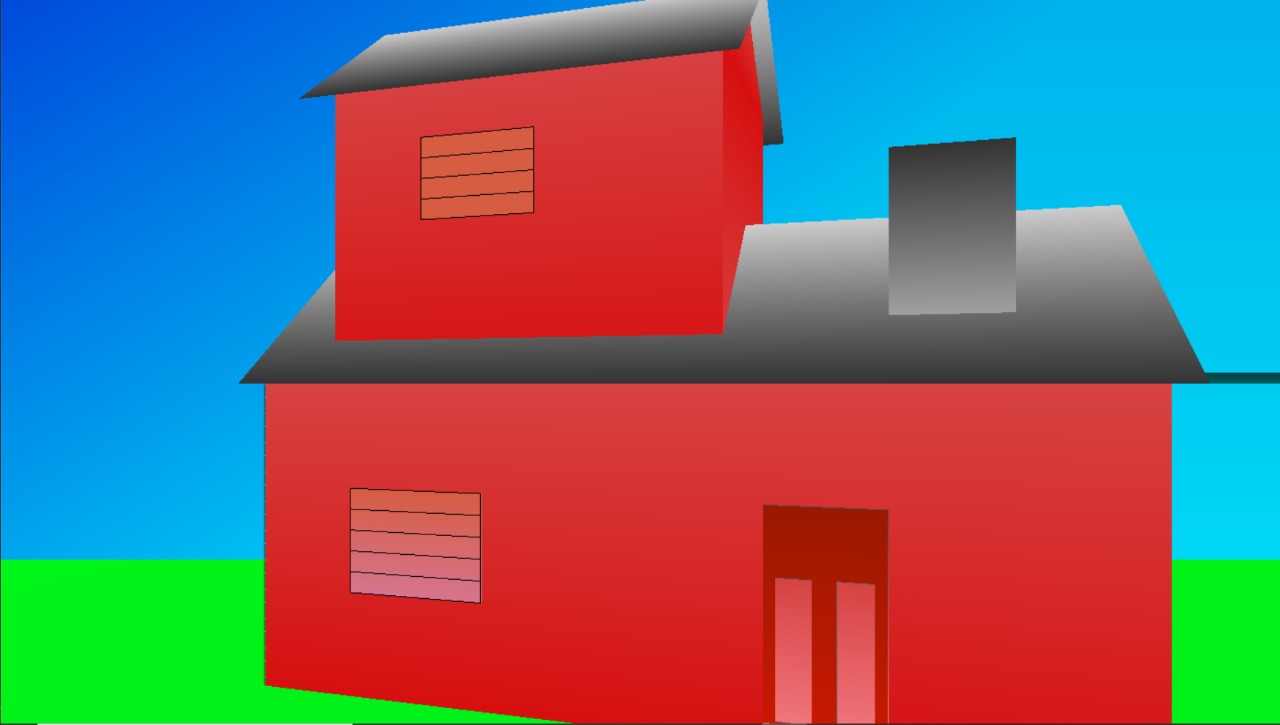
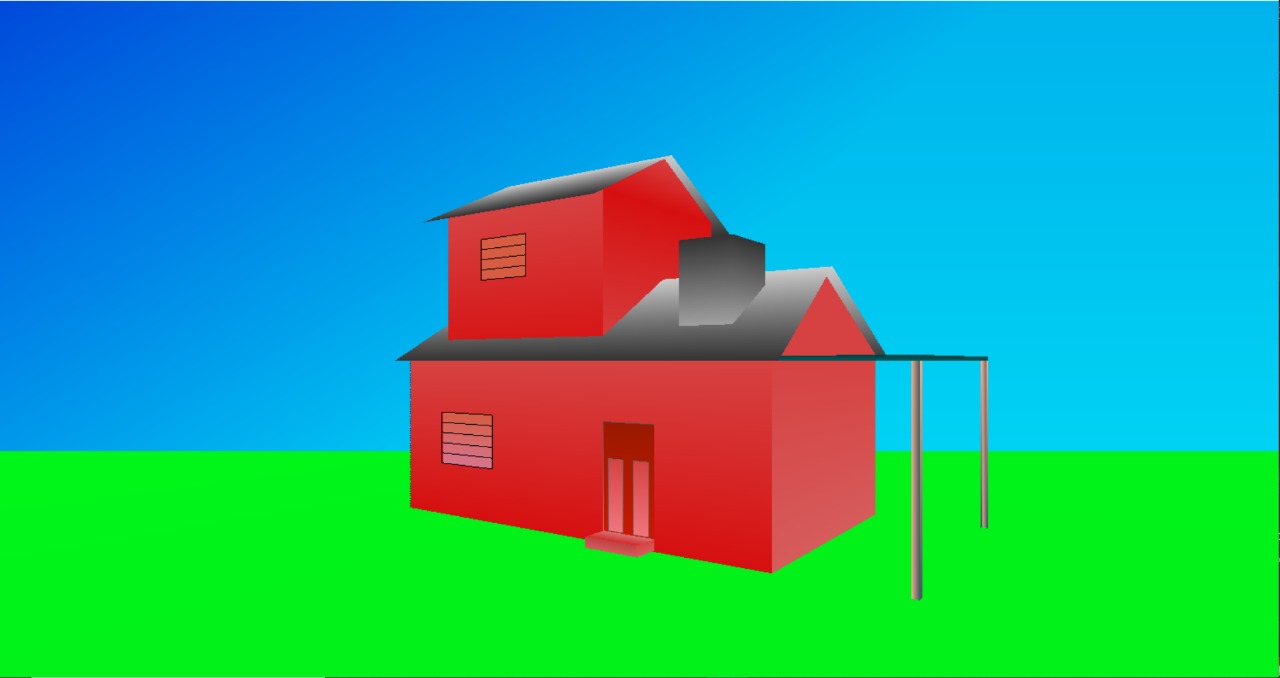
**Computer graphics project**



**How to use :**

Up arrow : zoom in

Down arrow: zoom out

Right arrow: rotate right

Left arrow : rotate left

Page up: tilt right

***Code:***

#include<stdio.h>

#include<GL/glut.h>

float angle = -45;

float xScale = 1, yScale = 1, zangle = 0;

//door with texture code but not working

\*/ void door()

{

GLfloat lightIntensity[] = {2.7f, 2.7f, 2.7f, 1.0f};

GLfloat lightPosition[] = {2.0f, 6.0f, 3.0f, 0.0f};

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

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

glPushMatrix();

glBindTexture(GL\_TEXTURE\_2D,8);

glEnable(GL\_TEXTURE\_2D);

roadFunc();

glDisable(GL\_TEXTURE\_2D);

glPopMatrix();

}

void door1()

{

glPushMatrix();

glRotated(90, 1, 0, 0);

glScalef(0.5, 1, 1);

glTranslated(0,.5,-0.5);

door();

glPopMatrix();

}

void doorFinal()

{

char \*data –stbi load(“wood.jpg”, 5, 10 ,0,0);

glPushMatrix();

glTranslated(-0.22,-0.18,-0.9);

glScalef(1,0.8,1);

door1();

glPopMatrix();

}

glutPostRedisplay();

}

\*/

// lights code from section

void lights()

{

GLfloat lightpos[] = { 12,15,10 }; //light direction.

GLfloat lightcolor[] = { 1,0,0 };

GLfloat ambColor[] = { 1,1,0 };//

glEnable(GL\_LIGHTING);

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, ambColor); //sets lighting model parameterto give shinny look

glEnable(GL\_LIGHT0);

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

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

}

static void resize(int width, int height)

{

const float ar = (float)width / (float)height;

glViewport(0, 0, width, height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glFrustum(-ar, ar, -1.0, 1.0, 2.0, 100.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void rotateFunc(int key, int x, int y) {

switch (key) {

case GLUT\_KEY\_RIGHT:

angle += 1;

if (angle > 360) angle = 0.0;

break;

case GLUT\_KEY\_LEFT:

angle -= 1;

if (angle > 360) angle = 0.0;

break;

case GLUT\_KEY\_DOWN:

if (xScale > 0.7)

{

xScale -= 0.01;

yScale -= 0.01;

}

break;

case GLUT\_KEY\_UP:

if (xScale < 1.5) {

xScale += 0.01;

yScale += 0.01;

}

break;

case GLUT\_KEY\_PAGE\_UP:

zangle = 1;

break;

case GLUT\_KEY\_PAGE\_DOWN:

zangle = 0;

break;

}

////////////////////////////

void scaleFunc(int key, int x, int y) {

switch (key) {

}

glutPostRedisplay();

}

////////////////////////////

void renderScene(void) {

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

glPushMatrix();

glTranslatef(0, 0, -10);

glColor3f(0, 0, 1);

glBegin(GL\_QUADS);

glColor3f(0, 0, .8);

glVertex3f(-10, 10, 0);

glColor3f(0, .6, .9);

glVertex3f(10, 10, 0);

glColor3f(0, 1, 1);

glVertex3f(10, -10, 0);

glVertex3f(-10, -10, 0);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glColor3f(0, 1, .1);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

//grass

glBegin(GL\_QUADS);

glVertex3f(-30, -1.5, 100);

glVertex3f(-30, -1.5, -100);

glVertex3f(30, -1.5, -100);

glColor3f(0, .9, .1);

glVertex3f(30, -1.5, 100);

glEnd();

glPopMatrix();

///////////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(-2, 0, 1);

glVertex3f(2, 0, 1);

glColor3f(0.84, .06, 0.06);

glVertex3f(2, -1.5, 1);

glVertex3f(-2, -1.5, 1);

glEnd();

glPopMatrix();

//////////////////////////Second Floor/////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS);

glColor3f(0.84, .06, 0.06);// Wall

glVertex3f(-1.7, 0, .8);

glVertex3f(.3, 0, .8);

glColor3f(0.84, .26, 0.26);

glVertex3f(.3, 1.5, .8);

glVertex3f(-1.7, 1.5, .8);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS);

glColor3f(0.84, .06, 0.06); // Wall

glVertex3f(-1.7, 0, -.8);

glVertex3f(.3, 0, -.8);

glColor3f(0.84, .26, 0.26);

glVertex3f(.3, 1.5, -.8);

glVertex3f(-1.7, 1.5, -.8);

glEnd();

glPopMatrix();

///////////////////////////////Second floor top///////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_TRIANGLES);

glColor3f(0.84, .06, 0.06); // Wall

glVertex3f(-1.7, 1.5, .8);

glVertex3f(-1.7, 1.5, -.8);

glColor3f(0.84, .26, 0.26);

glVertex3f(-1.7, 2, 0);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_TRIANGLES);

glColor3f(0.84, .06, 0.06); // Wall

glVertex3f(.3, 1.5, .8);

glVertex3f(.3, 1.5, -.8);

glColor3f(0.84, .26, 0.26);

glVertex3f(.3, 2, 0);

glEnd();

glPopMatrix();

/////////////////////////////////////////////////////////////WINDOW/////////////////////////////

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(.84, .36, 0.26);

glVertex3f(-1.5, -.5, 1.0005);

glVertex3f(-.8, -.5, 1.0005);

glColor3f(.84, .46, 0.56);

glVertex3f(-.8, -1, 1.0005);

glVertex3f(-1.5, -1, 1.0005);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINE\_LOOP); // Wall

glColor3f(0, 0, 0);

glVertex3f(-1.5, -.5, 1.01);

glVertex3f(-.8, -.5, 1.01);

glVertex3f(-.8, -1, 1.01);

glVertex3f(-1.5, -1, 1.01);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINE\_LOOP); // Wall

glColor3f(0, 0, 0);

glVertex3f(-1.2, .8, .81);

glVertex3f(-.6, .8, .81);

glVertex3f(-.6, 1.2, .81);

glVertex3f(-1.2, 1.2, .81);

glEnd();

glPopMatrix();///////////////////////////Window Top Fill/////////////////////

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(.84, .36, 0.26);

glVertex3f(-1.2, .8, .809);

glVertex3f(-.6, .8, .809);

glVertex3f(-.6, 1.2, .809);

glVertex3f(-1.2, 1.2, .809);

glEnd();

glPopMatrix();

////////////////////////////////////////////////////Top Window Strips/////////

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINES); // Wall

glColor3f(0, 0, 0);

glVertex3f(-1.2, .9, .81);

glVertex3f(-.6, .9, .81);

glVertex3f(-1.2, 1.0, .81);

glVertex3f(-.6, 1.0, .81);

glVertex3f(-1.2, 1.1, .81);

glVertex3f(-.6, 1.1, .81);

glVertex3f(-1.2, 1.2, .81);

glVertex3f(-.6, 1.2, .81);

glEnd();

glPopMatrix();

////////////////////////////////////STRIPS/////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINES); // Wall

glColor3f(0, 0, 0);

glVertex3f(-1.5, -.6, 1.009);

glVertex3f(-.8, -.6, 1.009);

glVertex3f(-1.5, -.7, 1.009);

glVertex3f(-.8, -.7, 1.009);

glVertex3f(-1.5, -.8, 1.009);

glVertex3f(-.8, -.8, 1.009);

glVertex3f(-1.5, -.9, 1.009);

glVertex3f(-.8, -.9, 1.009);

glEnd();

glPopMatrix();

/////////////////////////////////////////cyliner//////////////////////////////

glPushMatrix();

glColor3f(0.84, .16, 0.16);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINES); // Wall

glColor3f(0, 0, 0);

glVertex3f(-1.5, -.6, 1.009);

glVertex3f(-.8, -.6, 1.009);

glVertex3f(-1.5, -.7, 1.009);

glVertex3f(-.8, -.7, 1.009);

glVertex3f(-1.5, -.8, 1.009);

glVertex3f(-.8, -.8, 1.009);

glVertex3f(-1.5, -.9, 1.009);

glVertex3f(-.8, -.9, 1.009);

glEnd();

glPopMatrix();

///////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(-2, 0, -1);

glVertex3f(2, 0, -1);

glColor3f(0.84, .06, 0.06);

glVertex3f(2, -1.5, -1);

glVertex3f(-2, -1.5, -1);

glEnd();

glPopMatrix();

//////////////////////////\*\*\*\*\*POURCH\*\*\*\*///////////////////////////////////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.94, .66, 0.56);

glVertex3f(3, 0, .95);

glVertex3f(3, 0, 1);

glColor3f(0.24, .36, 0.36);

glVertex3f(3, -1.5, 1);

glVertex3f(3, -1.5, .95);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.94, .66, 0.56);

glVertex3f(2.95, 0, .95);

glVertex3f(2.95, 0, 1);

glColor3f(0.24, .36, 0.36);

glVertex3f(2.95, -1.5, 1);

glVertex3f(2.95, -1.5, .95);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.94, .66, 0.56);

glVertex3f(2.95, 0, 1);

glVertex3f(2.95, -1.5, 1);

glColor3f(0.24, .36, 0.36);

glVertex3f(3, -1.5, 1);

glVertex3f(3, 0, 1);

glEnd();

glPopMatrix();

//////////////////////////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.94, .66, 0.56);

glVertex3f(3, 0, -.95);

glVertex3f(3, 0, -1);

glColor3f(0.24, .36, 0.36);

glVertex3f(3, -1.5, -1);

glVertex3f(3, -1.5, -.95);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.94, .66, 0.56);

glVertex3f(2.95, 0, -.95);

glVertex3f(2.95, 0, -1);

glColor3f(0.24, .36, 0.36);

glVertex3f(2.95, -1.5, -1);

glVertex3f(2.95, -1.5, -.95);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.94, .66, 0.56);

glVertex3f(2.95, 0, -1);

glVertex3f(2.95, -1.5, -1);

glColor3f(0.24, .36, 0.36);

glVertex3f(3, -1.5, -1);

glVertex3f(3, 0, -1);

glEnd();

glPopMatrix();

/////////////////////////////pourch roof/////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_POLYGON); // right Wall

glColor3f(0, .26, 0.26);

glVertex3f(2, 0.04, 1);

glVertex3f(3, 0.04, 1);

glColor3f(0, .36, 0.36);

glVertex3f(3, 0.04, -1);

glVertex3f(2, 0.04, -1);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_POLYGON); // right Wall

glColor3f(0, .26, 0.26);

glVertex3f(2, 0.04, 1);

glVertex3f(3, 0.04, 1);

glColor3f(0, .36, 0.36);

glVertex3f(3, .0, 1);

glVertex3f(2, .0, 1);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_POLYGON); // right Wall

glColor3f(0, .26, 0.26);

glVertex3f(2, 0.04, -1);

glVertex3f(3, 0.04, -1);

glColor3f(0, .36, 0.36);

glVertex3f(3, .0, -1);

glVertex3f(2, .0, -1);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_POLYGON); // right Wall

glColor3f(0, .26, 0.26);

glVertex3f(3, 0.04, 1);

glVertex3f(3, 0.04, -1);

glColor3f(0, .36, 0.36);

glVertex3f(2, .0, -1);

glVertex3f(2, .0, 1);

glEnd();

glPopMatrix();

//Wall///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // right Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(2, 0, -1);

glVertex3f(2, 0, 1);

glColor3f(0.84, .36, 0.36);

glVertex3f(2, -1.5, 1);

glVertex3f(2, -1.5, -1);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); //left Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(-2, 0, -1);

glVertex3f(-2, 0, 1);

glColor3f(0.84, .36, 0.36);

glVertex3f(-2, -1.5, 1);

glVertex3f(-2, -1.5, -1);

glEnd();

glPopMatrix();

/////////////////////////////////////////////////////////////////////////////////////////

GLubyte halftone[] = {

//0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0x0A, 0x0A, 0x0A, 0x0A, 0x05, 0x05, 0x05, 0x05,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0x0A, 0x0A, 0x0A, 0x0A, 0x05, 0x05, 0x05, 0x05,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,

0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55 };

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glEnable(GL\_POLYGON\_STIPPLE); // Enable POLYGON STIPPLE

glPolygonStipple(halftone);

glBegin(GL\_QUADS); //left Wall

glColor3f(0.1, 0.2, 0.1);

glVertex3f(-2.01, 0, -1.0);

glVertex3f(-2.01, 0, 1.0);

glColor3f(0.2, 0.2, 0.1);

glVertex3f(-2.01, -1.5, 1.0);

glVertex3f(-2.01, -1.5, -1.0);

glEnd();

glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

//Roof//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); //Roof

glColor3f(1, 1, 0);

glColor3f(0.2, 0.2, 0.2);

glVertex3f(-2.1, 0, 1.1);

glVertex3f(2.1, 0, 1.1);

glColor3f(.8, .8, .8);

glVertex3f(2, .8, 0);

glVertex3f(-2, .8, 0);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); //Roof

glColor3f(1, 1, 0);

glColor3f(0.2, 0.2, 0.2);

glVertex3f(-1.8, 1.4, 1);

glVertex3f(.4, 1.4, 1);

glColor3f(.8, .8, .8);

glVertex3f(.4, 2, 0);

glVertex3f(-1.8, 2, 0);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); //Roof

glColor3f(1, 1, 0);

glColor3f(0.2, 0.2, 0.2);

glVertex3f(-1.8, 1.4, -1);

glVertex3f(.4, 1.4, -1);

glColor3f(.8, .8, .8);

glVertex3f(.4, 2, 0);

glVertex3f(-1.8, 2, 0);

glEnd();

glPopMatrix();

//roof back/////////////////////////////////////////////////////////////CHIMINY////////////////////////////////////

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

// glEnable(GL\_POLYGON\_STIPPLE);

// glPolygonStipple(myInitial);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); //Roof

glColor3f(0.2, 0.2, 0.2);

glVertex3f(1, 1, .7);

glVertex3f(1.5, 1, .7);

glColor3f(.8, .8, .8);

glVertex3f(1.5, 0, .7);

glVertex3f(1, 0, 0.7);

glEnd();

//glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

// glEnable(GL\_POLYGON\_STIPPLE);

// glPolygonStipple(myInitial);

glBegin(GL\_QUADS); //Roof

glColor3f(0.2, 0.2, 0.2);

glVertex3f(1, 1, .2);

glVertex3f(1.5, 1, .2);

glColor3f(.8, .8, .8);

glVertex3f(1.5, 0, .2);

glVertex3f(1, 0, .2);

glEnd();

//glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

// glEnable(GL\_POLYGON\_STIPPLE);

// glPolygonStipple(myInitial);

glBegin(GL\_QUADS); //Roof

glColor3f(0.2, 0.2, 0.2);

glVertex3f(1, 1, .2);

glVertex3f(1, 1, .7);

glColor3f(.8, .8, .8);

glVertex3f(1, 0, .7);

glVertex3f(1, 0, .2);

glEnd();

//glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

// glEnable(GL\_POLYGON\_STIPPLE);

// glPolygonStipple(myInitial);

glBegin(GL\_QUADS); //Roof

glColor3f(0.2, 0.2, 0.2);

glVertex3f(1.5, 1, .2);

glVertex3f(1.5, 1, .7);

glColor3f(.8, .8, .8);

glVertex3f(1.5, 0, .7);

glVertex3f(1.5, 0, .2);

glEnd();

//glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

////////////////////////////Second Floor//////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(-1.7, 0, .8);

glVertex3f(-1.7, 0, -.8);

glColor3f(0.84, .06, 0.06);

glVertex3f(-1.7, 1.5, -.8);

glVertex3f(-1.7, 1.5, .8);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(.3, 0, .8);

glVertex3f(.3, 0, -.8);

glColor3f(0.84, .06, 0.06);

glVertex3f(.3, 1.5, -.8);

glVertex3f(.3, 1.5, .8);

glEnd();

glPopMatrix();

/////////////////////////////////////////////////DOOR/////////////////////////////////////////////

/\* void doorFinal()\*/

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS);

glColor3f(0.6f, 0.1f, 0.0f);

glVertex3f(.5, -.5, 1.0005);

glVertex3f(1, -.5, 1.0005);

glColor3f(0.8f, 0.1f, 0.0f);

glVertex3f(1, -1.49, 1.0005);

glVertex3f(.5, -1.49, 1.0005);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINE\_LOOP);

glColor3f(.3, .3, .3);

glVertex3f(.5, -.5, 1.0007);

glVertex3f(1, -.5, 1.0007);

glColor3f(0.94, .56, 0.59);

glVertex3f(1, -1.49, 1.0007);

glVertex3f(.5, -1.49, 1.0007);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINE\_LOOP);

glColor3f(.4, .4, .4);

glVertex3f(.55, -.8, 1.00069);

glVertex3f(.7, -.8, 1.00069);

glVertex3f(.7, -1.39, 1.00069);

glVertex3f(.55, -1.39, 1.00069);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_LINE\_LOOP);

glColor3f(.4, .4, .4);

glVertex3f(.8, -.8, 1.00069);

glVertex3f(.95, -.8, 1.00069);

glVertex3f(.95, -1.39, 1.00069);

glVertex3f(.8, -1.39, 1.00069);

glEnd();

glPopMatrix();

////////////////////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_POLYGON);

glColor3f(0.84, .26, 0.26);

glVertex3f(.55, -.8, 1.00069);

glVertex3f(.7, -.8, 1.00069);

glColor3f(0.94, .46, 0.49);

glVertex3f(.7, -1.39, 1.00069);

glVertex3f(.55, -1.39, 1.00069);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_POLYGON);

glColor3f(0.84, .26, 0.26);

glVertex3f(.8, -.8, 1.00069);

glVertex3f(.95, -.8, 1.00069);

glColor3f(0.94, .46, 0.49);

glVertex3f(.95, -1.39, 1.00069);

glVertex3f(.8, -1.39, 1.00069);

glEnd();

glPopMatrix();

//////////////////////////////////////////Steps////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.74, .26, 0.26);

glVertex3f(.5, -1.4, 1.2);

glVertex3f(1, -1.4, 1.2);

glColor3f(0.84, .36, 0.36);

glVertex3f(1, -1.49, 1.2);

glVertex3f(.5, -1.49, 1.2);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .16, 0.16);

glVertex3f(.5, -1.4, 1.2);

glVertex3f(.5, -1.4, 1.00);

glColor3f(0.84, .36, 0.36);

glVertex3f(.5, -1.49, 1.00);

glVertex3f(.5, -1.49, 1.2);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .16, 0.16);

glVertex3f(.5, -1.4, 1.2);

glVertex3f(1, -1.4, 1.2);

glColor3f(0.84, .46, 0.46);

glVertex3f(1, -1.4, 1.00);

glVertex3f(.5, -1.4, 1.00);

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

glColor3f(0.84, .26, 0.26);

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); // Wall

glColor3f(0.84, .26, 0.26);

glVertex3f(1, -1.4, 1.2);

glVertex3f(1, -1.4, 1.00);

glColor3f(0.84, .36, 0.36);

glVertex3f(1, -1.49, 1.00);

glVertex3f(1, -1.49, 1.2);

glEnd();

glPopMatrix();

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

// glEnable(GL\_POLYGON\_STIPPLE);

// glPolygonStipple(myInitial);

glScalef(xScale, yScale, 1);

glBegin(GL\_QUADS); //Roof

glColor3f(0.2, 0.2, 0.2);

glVertex3f(-2.1, 0, -1.1);

glVertex3f(2.1, 0, -1.1);

glColor3f(.8, .8, .8);

glVertex3f(2, .8, 0);

glVertex3f(-2, .8, 0);

glEnd();

//glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

//Wall Upper////////////////////////////////////////////////////////////////////////////////

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_TRIANGLES); //Roof

glColor3f(0.84, .26, 0.26);

glVertex3f(-2, 0, 1.1);

glVertex3f(-1.9, .8, 0);

glVertex3f(-2, 0, -1.1);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glEnable(GL\_POLYGON\_STIPPLE);

glPolygonStipple(halftone);

glBegin(GL\_TRIANGLES); //Roof

glColor3f(0, .26, 0.26);

glVertex3f(-2.01, 0, 1.1);

glVertex3f(-1.9, .8, 0);

glVertex3f(-2.01, 0, -1.1);

glEnd();

glDisable(GL\_POLYGON\_STIPPLE);

glPopMatrix();

glPushMatrix();

glTranslatef(0, 0, -6);

glRotatef(angle, 0.0, 1, zangle);

glScalef(xScale, yScale, 1);

glBegin(GL\_TRIANGLES); //Roof

glColor3f(0.84, .26, 0.26);

glVertex3f(2, 0, 1.1);

glVertex3f(1.9, .8, 0);

glVertex3f(2, 0, -1.1);

glEnd();

glPopMatrix();

////////////////////////////////////////////////////////////////////////////////////

glutSwapBuffers();

}

void Initialize() {

glClearColor(0.0, 0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-10.0, 10.0, -10.0, 10.0, -10.0, 10.0);

}

void lights()

{

GLfloat lightpos[] = { 12,15,10 };

GLfloat lightcolor[] = { 1,0,0 };

GLfloat ambColor[] = { 1,1,0 };

glEnable(GL\_LIGHTING);

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, ambColor);

glEnable(GL\_LIGHT0);

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

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

}

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

glutInit(&argc, argv);

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

glutInitWindowPosition(0, 0);

glutInitWindowSize(1500, 1000);

glutCreateWindow("Assignment ");

glEnable(GL\_DEPTH\_TEST);

glutReshapeFunc(resize);

glutDisplayFunc(renderScene);

glutSpecialFunc(rotateFunc);

Initialize();

lights();

glutMainLoop();

return 0;

}