Table of Contents

1. Introduction to Project

This project is a 3D house architecture visualization implemented using OpenGL. It allows users to explore and interact with a virtual house environment. The project includes various elements such as the outer cover of the house, main door design, rooms, and a kitchen. Users can navigate and view the house from different perspectives using keyboard controls.

Controls through Keys:

1. Navigation Controls:

- Press 'x' to move the viewpoint left.
- Press 'X' to move the viewpoint right.
- Press 'y' to move the viewpoint down.
- Press 'Y' to move the viewpoint up.
- Press 'z' to move the viewpoint backward.
- Press 'Z' to move the viewpoint forward.

2. Interaction Controls:

- Press 'o' to interact with the door, triggering a function called **doort(o).**

3. Special Key Controls:

- Use the arrow keys to rotate the viewpoint left, right, up, or down.
- Use Page Up ('PAGE_UP') to move the viewpoint upward.
- Use Page Down ('PAGE_DOWN') to move the viewpoint downward.

These controls enable users to explore the 3D house environment dynamically, change viewing angles, and interact with specific elements such as doors, providing an immersive experience of the architectural design.

2. Computer Graphics concepts used

Sr. No	Computer Graphics Concept	Description	Usage
1.	Coordinate Systems	A coordinate system is a reference system used to represent the positions of objects in a two-dimensional or three-dimensional space.	**glMatrixMode(GL_MODELVIEW)** and **glLoadIdentity()** are functions used to set and manipulate the current coordinate system for modeling and viewing transformations.
2.	Color Rendering	Color rendering involves specifying colors for objects in the scene. OpenGL uses RGB color model where colors are defined using red, green, and blue components.	Functions like glColor3f(r, g, b) to set the current color for rendering objects with specific RGB values.
3.	Primitive Rendering	Primitive rendering involves drawing basic geometric shapes like points, lines, and polygons in the 3D space.	Functions like **glBegin(GL_POLYGON)** and **glVertex3f(x, y, z)** are used to define and render polygons by specifying their vertices.
4.	Projection and View Transformation	Projection and view transformation involve setting up the perspective projection and defining the viewer's position and orientation.	Functions **gluPerspective()** and **gluLookAt()** are used to set up the perspective projection and define the viewer's position and orientation, respectively.
5.	Depth Testing and Blending	Depth testing ensures proper rendering of objects based on their depth in the scene, and blending controls transparency and overlapping of objects.	Enabled depth testing using **glDepthFunc(GL_LEQUAL)** and sets up blending using **glEnable(GL_BLEND)** and **glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)**.
6.	Event Handling	Event handling involves capturing user input events such as keyboard key presses for interaction and navigation.	Functions **glutKeyboardFunc(keys)** and **glutSpecialFunc(specialKey)** are used to handle keyboard events for controlling navigation, interaction, and viewpoint changes.

3. User Defined Functions

User Defined Functions	Descriptions
`void outercover()`	The function outercover()draws the outer cover or boundary of the 3D scene. It defines a dark-colored polygon that serves as the backdrop against which other objects are rendered.
`void maindoor()`	The function maindoor() is responsible for rendering the main door of the house. It draws polygons representing the design elements of the door.
`void room1(), void room2()`	These functions `void room1(),void room2()` handle the rendering of specific rooms within the house. Each function draws polygons to represent walls, floors, and other elements of the respective rooms. Called within the house() function to visualize the different rooms of the house in the 3D scene.
`void kitchen()`	This function `void kitchen()` handle the rendering of kitchen within the house. Function draws polygons to represent walls, floors, and other elements of the kitchen.
`void mid()`	Function `void mid()` represents the hall or central area of the architectural visualization. It draws polygons to depict the layout or structure of the hall.

`void keys(unsigned char key, int x, int y)`	Function `keys(unsigned char key, int x, int y)` handles keyboard input events, allowing users to interact with the 3D scene by controlling the viewpoint or triggering specific actions based on key presses.
`void specialKey(int key, int x, int y)`	Similar to keys() function, function `specialKey(int key, int x, int y)` handles special keyboard input events such as arrow keys or page up/down keys, providing additional control options for the user.

4. Code

```
#include<stdio.h>
#include<GL/glut.h>
#include<math.h>
#include<string.h>
#pragma GCC diagnostic ignored "-Wwrite-strings"
//#include <stdlib.h>
int turn;
void doort(int v);
GLfloat xx = 0.0, zz = 0.0, xx1 = 1;
float lx = 0.0f, lz = -1.0f, angle = 0.0;
static GLdouble viewer[] = { 0.0, 0.2, 4.2 };
static GLdouble lat[] = { 0.0, 0.2, 0.0 };
GLfloat oo = 0, cr = 0, cb = 0, cg = 0;
void* font = GLUT_BITMAP_HELVETICA_18;
void output(float x, float y, char* s)
{
	glRasterPos2f(x, y);
	for (unsigned int i = 0; i < strlen(s); i++)
	{
		glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, s[i]);
	}
}


void tfs(int v)
{
	cr += 0.01;
	cb += 0.001;
	cg += 0.0001;
	if (cr > 1.0)
	{
		cr = 0;

	}
	if (cb > 1.0)
	{
		cb = 0;

	}
	if (cg > 1.0)
	{
		cg = 0;

	}

	glutPostRedisplay();
	glutTimerFunc(100, tfs, 0);

}
void doort(int v)    // door timing
```

```
{
	if (xx < 0.2 && zz < 0.2) {
		xx += 0.01;
		zz += 0.01;
	}


	xx1 = 0;



	glutPostRedisplay();
	glutTimerFunc(100, doort, 0);

}
void lines()
{
	glColor3f(0.0, 0.0, 0.0);   //2x2 ka room   every tile spacing is 0.2
	glLineWidth(0.2);
	glBegin(GL_LINES);
	glVertex3f(-0.8, 0.00, 1);
	glVertex3f(-0.8, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(-0.6, 0.00, 1);
	glVertex3f(-0.6, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(-0.4, 0.00, 1);
	glVertex3f(-0.4, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(-0.2, 0.00, 1);
	glVertex3f(-0.2, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(0.0, 0.00, 1);
	glVertex3f(0.0, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(0.2, 0.00, 1);
	glVertex3f(0.2, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(0.4, 0.00, 1);
	glVertex3f(0.4, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(0.6, 0.00, 1);
	glVertex3f(0.6, 0.00, -1);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(0.8, 0.00, 1);
```

```
        glVertex3f(0.8, 0.00, -1);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1.0, 0.00, 1);
        glVertex3f(1.0, 0.00, -1);
        glEnd();
        //vertical

        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, -0.8);
        glVertex3f(1, 0.00, -0.8);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, -0.6);
        glVertex3f(1, 0.00, -0.6);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, -0.4);
        glVertex3f(1, 0.00, -0.4);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, -0.2);
        glVertex3f(1, 0.00, -0.2);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, 0.0);
        glVertex3f(1, 0.00, 0.0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, 0.2);
        glVertex3f(1, 0.00, 0.2);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, 0.4);
        glVertex3f(1, 0.00, 0.4);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, 0.6);
        glVertex3f(1, 0.00, 0.6);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, 0.8);
        glVertex3f(1, 0.00, 0.8);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1, 0.00, 1.0);
        glVertex3f(1, 0.00, 1.0);
        glEnd();
}
void redwall()
{
```

```
//--------------dark red top---------------------
glColor3f(0.53, 0.12, 0.12);
glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.0, 1);
glVertex3f(-0.98, 0.5, 1);
glVertex3f(-0.98, 0.5, 0.9);
glVertex3f(-0.98, 0.0, 0.9);
glEnd();

glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.5, 0.9);
glVertex3f(-0.98, 0.5, -0.2);
glVertex3f(-0.98, 0.4, -0.2);
glVertex3f(-0.98, 0.4, 0.9);
glEnd();

glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.5, -0.3);
glVertex3f(-0.98, 0.0, -0.3);
glVertex3f(-0.98, 0.0, -0.2);
glVertex3f(-0.98, 0.5, -0.2);
glEnd();
//--------------grey mid--------------

glColor3f(0.69, 0.58, 0.56);
glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.0, 0.9);
glVertex3f(-0.98, 0.4, 0.9);
glVertex3f(-0.98, 0.4, 0.7);
glVertex3f(-0.98, 0.0, 0.7);
glEnd();

glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.4, 0.7);
glVertex3f(-0.98, 0.4, 0.0);
glVertex3f(-0.98, 0.25, 0.0);
glVertex3f(-0.98, 0.25, 0.7);
glEnd();
glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.4, -0.2);
glVertex3f(-0.98, 0.0, -0.2);
glVertex3f(-0.98, 0.0, 0.0);
glVertex3f(-0.98, 0.4, 0.0);
glEnd();

//--------------black----------------
glColor3f(0.15, 0.11, 0.11);
glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.0, 0.7);
glVertex3f(-0.98, 0.25, 0.7);
glVertex3f(-0.98, 0.25, 0.65);
glVertex3f(-0.98, 0.0, 0.65);
glEnd();
```

```
glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.25, 0.65);
glVertex3f(-0.98, 0.25, 0.0);
glVertex3f(-0.98, 0.2, 0.0);
glVertex3f(-0.98, 0.2, 0.65);
glEnd();

glBegin(GL_POLYGON);
glVertex3f(-0.98, 0.25, 0.0);
glVertex3f(-0.98, 0.0, 0.0);
glVertex3f(-0.98, 0.0, 0.05);
glVertex3f(-0.98, 0.25, 0.05);
glEnd();


//-------------black drawer---------

glColor3f(0.0, 0.0, 0.0);
glPushMatrix();                 // scale, translate  rotate matrix
glTranslatef(-0.95, 0.06, 0.8);   //(z,y,x)
glScalef(1.1, 0.4, 1.5);    //(z,y,x)
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 0.0, 0.0);
glPushMatrix();
glTranslatef(-0.92, 0.06, 0.8);
glScalef(0.56, 0.15, 1.45);
glutSolidCube(0.1);
glPopMatrix();

//---------bed(1st layer)------------
glColor3f(0.15, 0.11, 0.11);
glPushMatrix();
glTranslatef(-0.7, 0.01, 0.36);
glScalef(6, 0.25, 5);
glutSolidCube(0.1);
glPopMatrix();
//----------(2nd layer)---------------
glColor3f(0.17, 0.13, 0.12);
glPushMatrix();
glTranslatef(-0.7, 0.04, 0.36);
glScalef(6, 0.3, 5.2);
glutSolidCube(0.1);
glPopMatrix();

//----------bed-----------

glColor3f(0.84, 0.72, 0.69);
glPushMatrix();
glTranslatef(-0.7, 0.065, 0.36);
glScalef(5.9, 0.3, 5.1);
glutSolidCube(0.1);
```

```
	glPopMatrix();
	//-------------bedsheet----------------
	glColor3f(0.86, 0.52, 0.55);
	glPushMatrix();
	glTranslatef(-0.5, 0.069, 0.36);
	glScalef(2.5, 0.3, 5.2);
	glutSolidCube(0.1);
	glPopMatrix();


	//--------------pillow---------------

	glColor3f(0.86, 0.52, 0.55);
	glPushMatrix();
	glTranslatef(-0.9, 0.1, 0.5);
	glScalef(1, 0.2, 1.5);
	glRotatef(45, 0, 0, 1);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.86, 0.52, 0.55);
	glPushMatrix();
	glTranslatef(-0.9, 0.1, 0.25);
	glScalef(1, 0.2, 1.5);
	glRotatef(45, 0, 0, 1);
	glutSolidCube(0.1);
	glPopMatrix();
	//-------light-------

	glColor3f(1.0, 1.0, 0.6);
	glPushMatrix();
	glTranslatef(-0.98, 0.3, -0.1);
	//glScalef(2.5, 0.3, 5.2);
	glRotatef(90, 1, 0, 0);
	glutSolidCone(0.03, 0.07, 100, 100);  //(base,height,slice,stack)
	glPopMatrix();

	glColor3f(1.0, 1.0, 0.6);
	glPushMatrix();
	glTranslatef(-0.98, 0.3, 0.8);
	//glScalef(2.5, 0.3, 5.2);
	glRotatef(90, 1, 0, 0);
	glutSolidCone(0.03, 0.07, 100, 100);
	glPopMatrix();
}
void greywall()
{
	glColor3f(1.0, 1.0, 0.6);		//yellow box design
	glPushMatrix();
	glTranslatef(-0.55, 0.32, 0. - 0.9);
	glutSolidCube(0.08);
	glPopMatrix();
```

```
        glColor3f(1.0, 1.0, 0.6);
        glPushMatrix();
        glTranslatef(-0.43, 0.28, 0. - 0.9);
        glutSolidCube(0.08);
        glPopMatrix();

        glColor3f(1.0, 1.0, 0.6);
        glPushMatrix();
        glTranslatef(-0.31, 0.32, 0. - 0.9);
        glutSolidCube(0.08);
        glPopMatrix();

        glColor3f(1.0, 1.0, 0.6);
        glPushMatrix();
        glTranslatef(-0.2, 0.28, 0. - 0.9);
        glutSolidCube(0.08);
        glPopMatrix();
}
void box()
{


        //----------TV -------------
        glColor3f(0.18, 0.12, 0.13);
        glPushMatrix();
        glTranslatef(0.46, 0.25, 0.5);
        glScalef(0.1, 1.5, 3);
        glutSolidCube(0.1);
        glPopMatrix();


        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.47, 0.25, 0.5);
        glScalef(0.1, 1.55, 3.3);
        glutSolidCube(0.1);
        glPopMatrix();


        //--------------table-----------

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.46, 0.1, 0.5);
        glScalef(0.4, 0.4, 3);
        glutSolidCube(0.1);
        glPopMatrix();


        glColor3f(1.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.455, 0.1, 0.5);
        glScalef(0.41, 0.2, 2.9);
```

```
        glutSolidCube(0.1);
        glPopMatrix();

        //-----------speaker-----------

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.47, 0.08, 0.2);
        glScalef(0.1, 1.55, 0.7);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.47, 0.08, 0.8);
        glScalef(0.1, 1.55, 0.7);
        glutSolidCube(0.1);
        glPopMatrix();

        //--------wall design light-------------


        glColor3f(1.0, 1.0, 0.6);
        glPushMatrix();
        glTranslatef(0.47, 0.4, 0.1);
        glutSolidCube(0.08);
        glPopMatrix();

        glColor3f(1.0, 1.0, 0.6);
        glPushMatrix();
        glTranslatef(0.47, 0.35, 0.2);
        glutSolidCube(0.08);
        glPopMatrix();
}
void sofa()
{


        glColor3f(0.5,0.2,0.0);
        glPushMatrix();             //brown portion
        glTranslatef(-0.32, 0.065, -0.3);
        glScalef(4, 0.65, 1.7);
        glutSolidCube(0.1);
        glPopMatrix();

        //------base-black-----------
        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(-0.32, 0.015, -0.3);
        glScalef(4.1, 0.15, 1.71);
        glutSolidCube(0.1);
        glPopMatrix();
```

```
	//-------back
	glColor3f(0.0, 0.0, 0.0);
	glPushMatrix();
	glTranslatef(-0.32, 0.1, -0.36);
	glScalef(4.1, 1.3, 0.61);
	glutSolidCube(0.1);
	glPopMatrix();

	//---------seat----------

	glColor3f(0,0,0);
	glPushMatrix();
	glTranslatef(-0.41, 0.092, -0.25);
	glScalef(1.7, 0.15, 1.0);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.0, 0.0, 0.0);
	glPushMatrix();
	glTranslatef(-0.21, 0.092, -0.25);
	glScalef(1.7, 0.15, 1.0);
	glutSolidCube(0.1);
	glPopMatrix();

	//------side rest--------

	glColor3f(0.0, 0.0, 0.0);
	glPushMatrix();
	glTranslatef(-0.55, 0.065, -0.3);
	glScalef(1, 1.4, 1.1);
	glutSolidCube(0.1);
	glPopMatrix();

}

void wardrobe()
{
	//--------------wall--------
	glColor3f(0.47, 0.35, 0.26);    // wardrobe left side brown wall
	glPushMatrix();
	glTranslatef(-0.99, 0.25, -0.8);
	glScalef(0.5, 5, 4);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.47, 0.35, 0.26); //right side brown wall
	glPushMatrix();
	glTranslatef(0.99, 0.25, -0.8);
	glScalef(0.5, 5, 4);
	glutSolidCube(0.1);
	glPopMatrix();

	//-----------top-------------
```

```
        glColor3f(0.47, 0.35, 0.26);
        glPushMatrix();
        glTranslatef(0.0, 0.5, -0.8);
        glScalef(20, 0.5, 4);
        glutSolidCube(0.1);
        glPopMatrix();

        //-----------wardrobe-------------
        glColor3f(0.38, 0.27, 0.19);
        glPushMatrix();
        glTranslatef(0.4, 0.2, -0.9);
        glScalef(8, 4, 2);
        glutSolidCube(0.1);
        glPopMatrix();

        //----------design-------------

        glColor3f(0.9, 0.9, 0.9);
        glPushMatrix();
        glTranslatef(0.1, 0.2, -0.89);
        glScalef(1.5, 3.5, 2);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.9, 0.9, 0.9);
        glPushMatrix();
        glTranslatef(0.4, 0.2, -0.89);
        glScalef(1.5, 3.5, 2);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.9, 0.9, 0.9);
        glPushMatrix();
        glTranslatef(0.6, 0.3, -0.89);
        glScalef(0.7, 1.5, 2);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.9, 0.9, 0.9);
        glPushMatrix();
        glTranslatef(0.7, 0.3, -0.89);
        glScalef(0.7, 1.5, 2);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.47, 0.35, 0.26);
        glPushMatrix();
        glTranslatef(0.65, 0.1, -0.89);
        glScalef(1.8, 1.5, 2);
        glutSolidCube(0.1);
        glPopMatrix();
}
```

```
void room1()
{

  //--------------floor-----------------
        glColor3f(0.7, 0.4, 0.3);
        glBegin(GL_POLYGON);
        glVertex3f(1, 0.0, -1);
        glVertex3f(1, 0.0, 1);
        glVertex3f(-1, 0.0, 1);
        glVertex3f(-1, 0.0, -1);
        glEnd();

        lines();


        //--------wall(back)grey1----------------
        glColor3f(0.69, 0.58, 0.56);  //front of door
        glBegin(GL_POLYGON);
        glVertex3f(1, 0.5, -1);
        glVertex3f(1, 0.0, -1);
        glVertex3f(-1, 0.0, -1);
        glVertex3f(-1, 0.5, -1);
        glEnd();


        //----------wall(right)------
        glColor3f(0.69, 0.58, 0.56);  //wardrobe ke right wali
        glBegin(GL_POLYGON);
        glVertex3f(1, 0.5, -1);
        glVertex3f(1, 0.0, -1);
        glVertex3f(1, 0.0, 1);
        glVertex3f(1, 0.5, 1);
        glEnd();



        //----------wall(left)red------
        glColor3f(1.0, 0.0, 0.0);
        glBegin(GL_POLYGON);
        glVertex3f(-1.0, 0.5, -1);  //(z,y,x)
        glVertex3f(-1.0, 0.0, -1);
        glVertex3f(-1.0, 0.0, 1);
        glVertex3f(-1.0, 0.5, 1);
        glEnd();



        //----------roof------
        glColor3f(0.80, 0.60, 0.58);
        glBegin(GL_POLYGON);
        glVertex3f(-1, 0.5, -1);
        glVertex3f(-1, 0.5, 1);
        glVertex3f(1, 0.5, 1);
```

```
glVertex3f(1, 0.5, -1);
glEnd();

//--------wall(back)grey1----------------
glColor3f(0.69, 0.58, 0.56);
glBegin(GL_POLYGON);
glVertex3f(1, 0.5, 1);
glVertex3f(1, 0.27, 1);
glVertex3f(-1, 0.27, 1);
glVertex3f(-1, 0.5, 1);
glEnd();

glColor3f(0.69, 0.58, 0.56);
glBegin(GL_POLYGON);
glVertex3f(0.2, 0.0, 1);
glVertex3f(0.2, 0.27, 1);
glVertex3f(-1, 0.27, 1);
glVertex3f(-1, 0.0, 1);
glEnd();

glColor3f(0.69, 0.58, 0.56);
glBegin(GL_POLYGON);
glVertex3f(0.4, 0.0, 1);
glVertex3f(0.4, 0.27, 1);
glVertex3f(1, 0.27, 1);
glVertex3f(1, 0.0, 1);
glEnd();

//----------------door main-------------
glColor3f(0.01, 0.012, 0.012);
glPushMatrix();
glTranslatef(0.15, 0.135, 1.055);
glRotatef(45 , 0, 1, 0);
glScalef(1.5, 2.7, 0.1);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.901, 0.9012, 0.9012);
glPushMatrix();
glTranslatef(0.15, 0.135, 1.055);
glRotatef(45 , 0, 1, 0);
glScalef(1.2, 2.2, 0.11);
glutSolidCube(0.1);
glPopMatrix();

//--border------

glColor3f(0.38, 0.27, 0.19);
glBegin(GL_POLYGON);
glVertex3f(0.4, 0.285, 1.0101);
glVertex3f(0.4, 0.27, 1.0101);
glVertex3f(0.2, 0.27, 1.0101);
glVertex3f(0.2, 0.285, 1.0101);
```

```
glEnd();

glColor3f(0.38, 0.27, 0.19);
glBegin(GL_POLYGON);
glVertex3f(0.395, 0.0, 1.0101);
glVertex3f(0.395, 0.285, 1.0101);
glVertex3f(0.41, 0.285, 1.0101);
glVertex3f(0.41, 0.0, 1.0101);
glEnd();



glColor3f(0.38, 0.27, 0.19);
glBegin(GL_POLYGON);
glVertex3f(0.2, 0.0, 1.0101);
glVertex3f(0.2, 0.285, 1.0101);
glVertex3f(0.185, 0.285, 1.0101);
glVertex3f(0.185, 0.0, 1.0101);
glEnd();

// back border-----

glColor3f(0.38, 0.27, 0.19);
glBegin(GL_POLYGON);
glVertex3f(0.4, 0.285, 0.999);
glVertex3f(0.4, 0.27, 0.999);
glVertex3f(0.2, 0.27, 0.999);
glVertex3f(0.2, 0.285, 0.999);
glEnd();

glColor3f(0.38, 0.27, 0.19);
glBegin(GL_POLYGON);
glVertex3f(0.395, 0.0, 0.999);
glVertex3f(0.395, 0.285, 0.999);
glVertex3f(0.41, 0.285, 0.999);
glVertex3f(0.41, 0.0, 0.999);
glEnd();



glColor3f(0.38, 0.27, 0.19);
glBegin(GL_POLYGON);
glVertex3f(0.2, 0.0, 0.999);
glVertex3f(0.2, 0.285, 0.999);
glVertex3f(0.185, 0.285, 0.999);
glVertex3f(0.185, 0.0, 0.999);
glEnd();




//----------tv wall------
```

```
	glColor3f(0.53, 0.12, 0.12);
	glPushMatrix();
	glTranslatef(0.74, 0.25, 0.5);
	glScalef(5.1, 5, 9.99);
	glutSolidCube(0.1);
	glPopMatrix();

	//---------door------------

	glColor3f(0.0, 0.0, 0.0);
	glPushMatrix();
	glTranslatef(0.74, 0.18, -0.005);
	glScalef(2, 3.5, 0.2);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.47, 0.35, 0.26);
	glPushMatrix();
	glTranslatef(0.74, 0.18, 0.0);
	glScalef(2.1, 3.6, 0.2);
	glutSolidCube(0.1);
	glPopMatrix();

	//---------roof lamp-------------------------------

	glColor3f(1.0, 1.0, 0.6);
	glPushMatrix();
	glTranslatef(-0.2, 0.5, 0.0);
	glRotatef(90, 1, 0, 0);
	glutSolidTorus(0.03, 0.41, 100, 100);
	glPopMatrix();




	redwall();

	greywall();

	box();

	wardrobe();

	sofa();

	glFlush();

}

//---------------------------------------room2-----------------------------------------------------------

void room2()
```

```
{
//-----------------entrane--------------

        glColor3f(1.0, 1.0, 0.9);
        glBegin(GL_POLYGON);
        glVertex3f(0.99, 0.5, 1);
        glVertex3f(0.99, 0.5, 3);
        glVertex3f(0.99, 0.0, 3);
        glVertex3f(0.99, 0.0, 1);
        glEnd();



        glColor3f(0.5, 0.8, 0.85);
        glBegin(GL_POLYGON);
        glVertex3f(0.99, 0.0, 2.8);
        glVertex3f(0.99, 0.0, 3);
        glVertex3f(0.99, 0.0, 3);
        glVertex3f(0.99, 0.0, 2.8);
        glEnd();

}
void mtv()
{
        glColor3f(0.7, 0.4, 0.2);  //TV wall
        glPushMatrix();
        glTranslatef(0.96, 0.2, 1.73);
        glScalef(0.1, 4, 7);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.0,0.0,0.0);  // Tv table
        glPushMatrix();
        glTranslatef(0.92,0.05,1.73);
        glScalef(0.6,0.6,0.6);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.64, 0.64, 0.64); //tv table shadow
        glPushMatrix();
        glTranslatef(0.97, 0.05, 1.73);
        glScalef(0.6, 0.65, 0.65);
        glutSolidCube(0.1);
        glPopMatrix();

        //shadow

        glColor3f(0.0, 0.01, 0.0);
        glPushMatrix();
        glTranslatef(0.97, 0.2, 1.73);
        glScalef(0.001, 4, 7);
        glutSolidCube(0.1);
        glPopMatrix();
```

```
//line

glColor3f(0.5, 0.8, 0.2);
glPushMatrix();
glTranslatef(0.97, 0.425, 1.8);
glScalef(0.001, 0.05, 23.80);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.5, 0.8, 0.2);
glPushMatrix();
glTranslatef(0.97, 0.325, 1.8);
glScalef(0.001, 0.05, 23.80);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.5, 0.8, 0.2);
glPushMatrix();
glTranslatef(0.97, 0.225, 1.8);
glScalef(0.001, 0.05, 23.80);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.5, 0.8, 0.2);
glPushMatrix();
glTranslatef(0.97, 0.125, 1.8);
glScalef(0.001, 0.05,23.80);
glutSolidCube(0.1);
glPopMatrix();



//tv

glColor3f(0.0990, 0.099901, 0.09909);
glPushMatrix();
glTranslatef(0.9596, 0.2, 1.73);
glScalef(0.1, 1.5, 3);
glutSolidCube(0.1);
glPopMatrix();

// speaker

glColor3f(0.0990, 0.099901, 0.09909);
glPushMatrix();
glTranslatef(0.9596, 0.075, 2.0);
glScalef(0.6, 1.5, 0.5);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.64, 0.64, 0.64);
```

```
        glPushMatrix();
        glTranslatef(0.9599, 0.075, 2.0);
        glScalef(0.6, 1.52, 0.52);
        glutSolidCube(0.1);
        glPopMatrix();


        glColor3f(0.0990, 0.099901, 0.09909);
        glPushMatrix();
        glTranslatef(0.9596, 0.075, 1.47);
        glScalef(0.6, 1.5, 0.5);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.64, 0.64, 0.64);
        glPushMatrix();
        glTranslatef(0.9599, 0.075, 1.47);
        glScalef(0.6, 1.52, 0.52);
        glutSolidCube(0.1);
        glPopMatrix();
}
void msofa()
{
        //base
        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.0, 0.002, 1.7);
        glScalef(1.2, 0.5, 3);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.0, 0.002, 2.1);
        glScalef(1.2, 0.5, 3);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.065, 0.002, 2.2);
        glScalef(2.5, 0.5, 1.2);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(0.065, 0.002, 1.6);
        glScalef(2.5, 0.5, 1.2);
        glutSolidCube(0.1);
        glPopMatrix();

        //layer 2
```

```
glColor3f(0.5, 0.2, 0.0);
glPushMatrix();
glTranslatef(0.0, 0.05, 1.7);
glScalef(1.2, 0.5, 3);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.5, 0.2, 0.0);
glPushMatrix();
glTranslatef(0.0, 0.05, 2.1);
glScalef(1.2, 0.5, 3);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.5, 0.2, 0.0);
glPushMatrix();
glTranslatef(0.065, 0.05, 2.2);
glScalef(2.5, 0.5, 1.2);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.5, 0.2, 0.0);
glPushMatrix();
glTranslatef(0.065, 0.05, 1.6);
glScalef(2.5, 0.5, 1.2);
glutSolidCube(0.1);
glPopMatrix();

//rest

glColor3f(0.0, 0.0,0.0 );
glPushMatrix();
glTranslatef(-0.06, 0.05, 1.7);
glScalef(0.4, 1.5, 3.1);
glutSolidCube(0.1);
glPopMatrix();


glColor3f(0.0, 0.0, 0.0);
glPushMatrix();
glTranslatef(-0.06, 0.05, 2.115);
glScalef(0.4, 1.5, 3.1);
glutSolidCube(0.1);
glPopMatrix();

//pillow

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-0.03, 0.115, 1.7);
glScalef(0.2, 0.5, 0.5);
glRotatef(45, 1, 0, 0);
```

```
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-0.03, 0.115, 1.8);
glScalef(0.2, 0.5, 0.5);
glRotatef(45, 1, 0, 0);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-0.03, 0.115, 1.6);
glScalef(0.2, 0.5, 0.5);
glRotatef(45, 1, 0, 0);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-0.03, 0.115, 2.0);
glScalef(0.2, 0.5, 0.5);
glRotatef(45, 1, 0, 0);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-0.03, 0.115, 2.1);
glScalef(0.2, 0.5, 0.5);
glRotatef(45, 1, 0, 0);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-0.03, 0.115, 2.2);
glScalef(0.2, 0.5, 0.5);
glRotatef(45, 1, 0, 0);
glutSolidCube(0.1);
glPopMatrix();

//table

glColor3f(1.0, 0.6, 0.4);
glPushMatrix();
glTranslatef(0.3, 0.045, 1.9);
glScalef(1.5, 0.8, 1.5);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 0.9, 0.2);   //Teapot on table
```

```
        glPushMatrix();
        glTranslatef(0.3, 0.10, 1.9);
        glScalef(0.2, 0.2, 0.2);
        glutSolidTeapot(0.1);
        glPopMatrix();

        glColor3f(0.8, 0.4, 0.0);
        glPushMatrix();
        glTranslatef(0.3, 0.005, 1.9);
        glScalef(1.55, 0.2, 1.55);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(1.0, 0.6, 0.4);    //torus on wall
        glPushMatrix();
        glTranslatef(0.25, 0.40, 1.045);
        glScalef(0.2, 0.2, 0.2);
        glutSolidTorus(0.05,0.1,5,25);
        glPopMatrix();

        glColor3f(1.0, 0.6, 0.4);  //2nd torus
        glPushMatrix();
        glTranslatef(0.309, 0.40, 1.045);
        glScalef(0.2, 0.2, 0.2);
        glutSolidTorus(0.05, 0.1, 5, 25);
        glPopMatrix();

        glColor3f(1.0, 0.6, 0.4); //3rd torus
        glPushMatrix();
        glTranslatef(0.28, 0.35, 1.045);
        glScalef(0.2, 0.2, 0.2);
        glutSolidTorus(0.05, 0.1, 5, 25);
        glPopMatrix();



        //floor

        glColor3f(1.0, 1.0, 0.7);
        glPushMatrix();
        glTranslatef(0.15, 0.0005, 1.9);
        glScalef(7, 0.1, 10);
        glutSolidCube(0.1);
        glPopMatrix();

}
void mdesign()
{
        //1st row
        //red
        glColor3f(0.95, 0.95, 0.95);
        glPushMatrix();
```

```
glTranslatef(-0.47, 0.4, 1.04);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.53, 0.12, 0.12);
glPushMatrix();
glTranslatef(-0.47, 0.4, 1.041);
glScalef(0.8, 0.8, 0.01);
glutSolidCube(0.1);
glPopMatrix();

//black

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.3, 0.4, 1.04);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.053, 0.012, 0.012);
glPushMatrix();
glTranslatef(-0.3, 0.4, 1.041);
glScalef(0.8, 0.8, 0.01);
glutSolidCube(0.1);
glPopMatrix();

//2nd row

//black

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.47, 0.25, 1.04);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.053, 0.012, 0.012);
glPushMatrix();
glTranslatef(-0.47, 0.25, 1.041);
glScalef(0.8, 0.8, 0.01);
glutSolidCube(0.1);
glPopMatrix();

//red

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.3, 0.25, 1.04);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
```

```
glPopMatrix();

glColor3f(0.53, 0.12, 0.12);
glPushMatrix();
glTranslatef(-0.3, 0.25, 1.041);
glScalef(0.8, 0.8, 0.01);
glutSolidCube(0.1);
glPopMatrix();

//3rd row

//red
glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.47, 0.1, 1.04);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.53, 0.12, 0.12);
glPushMatrix();
glTranslatef(-0.47, 0.1, 1.041);
glScalef(0.8, 0.8, 0.01);
glutSolidCube(0.1);
glPopMatrix();
//black

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.3, 0.1, 1.04);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.053, 0.012, 0.012);
glPushMatrix();
glTranslatef(-0.3, 0.1, 1.041);
glScalef(0.8, 0.8, 0.01);
glutSolidCube(0.1);
glPopMatrix();


//big

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.7, 0.25, 1.04);
glScalef(2, 2, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.095, 0.095, 0.095);
glPushMatrix();
```

```
glTranslatef(-0.7, 0.25, 1.041);
glScalef(1.7, 1.7, 0.01);
glutSolidCube(0.1);
glPopMatrix();


glColor3f(0.53, 0.12, 0.12);
glPushMatrix();
glTranslatef(-0.7, 0.25, 1.042);
glScalef(1.4, 1.4, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.7, 0.25, 1.043);
glScalef(1.1, 1.1, 0.01);
glutSolidCube(0.1);
glPopMatrix();
//line

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.32, 0.25, 1.04);
glScalef(0.03, 5, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.28, 0.25, 1.04);
glScalef(0.03, 5, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.45, 0.25, 1.04);
glScalef(0.03, 5, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.95, 0.95, 0.95);
glPushMatrix();
glTranslatef(-0.49, 0.25, 1.04);
glScalef(0.03, 5, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.0, 0.0, 0.0);
glPushMatrix();
glTranslatef(0.93, 0.245, 1.05);
glScalef(0.03, 4.0, 0.01);
```

```
        glutSolidCube(0.12);
        glPopMatrix();

        glColor3f(0.0, 0.0, 0.0);
        glPushMatrix();
        glTranslatef(-0.97, 0.245, 1.05);
        glScalef(0.03, 4.0, 0.01);
        glutSolidCube(0.12);
        glPopMatrix();

}
void mtop()
{
        glColor3f(0.34, 0.25, 0.13);
        glPushMatrix();
        glTranslatef(0.0, 0.49, 1.9);
        glScalef(3, 0.3, 5);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.9985, 0.9985, 0.9985);
        glPushMatrix();
        glTranslatef(0.0, 0.488, 1.9);
        glScalef(2.7, 0.3, 4.5);
        glutSolidCube(0.1);
        glPopMatrix();


        glColor3f(0.34, 0.25, 0.13);
        glPushMatrix();
        glTranslatef(0.35, 0.49, 1.9);
        glScalef(3, 0.3, 5);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.9985, 0.9985, 0.9985);
        glPushMatrix();
        glTranslatef(0.35, 0.488, 1.9);
        glScalef(2.7, 0.3, 4.5);
        glutSolidCube(0.1);
        glPopMatrix();

        GLUquadricObj* quadratic;
        quadratic = gluNewQuadric();

        glColor3f(0.0 + cr, 0.0 + cg, 0.0 + cb);
        glPushMatrix();
        glTranslatef(0.17, 0.42, 1.95);
        glutSolidSphere(0.04, 100, 100);
        glPopMatrix();

        glColor3f(0.09, 0.09, 0.09);
        glPushMatrix();
```

```
        glTranslatef(0.175, 0.48, 1.999);
        glScalef(1, 0.15, 1);
        glutSolidSphere(0.02, 100, 100);
        glPopMatrix();

        glPushMatrix();
        glTranslatef(0.175, 0.5199, 1.95);
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.002, 0.002, 0.07, 100, 100);
        glPopMatrix();

        glPushMatrix();
        glTranslatef(0.17, 0.42, 1.95);
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.0415, 0.0415, 0.002, 100, 100);
        glPopMatrix();

        glPushMatrix();
        glTranslatef(0.17, 0.40, 1.95);
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.0355, 0.0345, 0.003, 100, 100);
        glPopMatrix();

        glPushMatrix();
        glTranslatef(0.17, 0.445, 1.95);
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.0355, 0.0345, 0.003, 100, 100);
        glPopMatrix();

}

void mid()
{
        //cyan

        glColor3f(1.0, 1.0, 0.9);
        glBegin(GL_POLYGON);
        glVertex3f(0.985, 0.5, 1);
        glVertex3f(0.985, 0.5, 2.6);
        glVertex3f(0.985, 0.0, 2.6);
        glVertex3f(0.985, 0.0, 1);
        glEnd();


        glColor3f(0.85, 0.8, 0.85);
        glBegin(GL_POLYGON);
        glVertex3f(-1, 0.0, 1);
        glVertex3f(1, 0.0, 1);
        glVertex3f(1, 0.0, 3);
        glVertex3f(-1, 0.0, 3);
        glEnd();

        glColor3f(0.85, 0.8, 0.85);
```

```
glBegin(GL_POLYGON);
glVertex3f(-1, 0.5, 1);
glVertex3f(1, 0.5, 1);
glVertex3f(1, 0.5, 3);
glVertex3f(-1, 0.5, 3);
glEnd();

//white wall

glColor3f(1.0, 1.0, 0.9);
glBegin(GL_POLYGON);
glVertex3f(1, 0.5, 1.01);
glVertex3f(1, 0.27, 1.01);
glVertex3f(-1, 0.27, 1.01);
glVertex3f(-1, 0.5, 1.01);
glEnd();

glColor3f(1.0, 1.0, 0.9);
glBegin(GL_POLYGON);
glVertex3f(0.2, 0.0, 1.01);
glVertex3f(0.2, 0.27, 1.01);
glVertex3f(-1, 0.27, 1.01);
glVertex3f(-1, 0.0, 1.01);
glEnd();

glColor3f(1.0, 1.0, 0.9);
glBegin(GL_POLYGON);
glVertex3f(0.4, 0.0, 1.01);
glVertex3f(0.4, 0.27, 1.01);
glVertex3f(1, 0.27, 1.01);
glVertex3f(1, 0.0, 1.01);
glEnd();

//black background

glColor3f(0.09, 0.09, 0.09);
glBegin(GL_POLYGON);
glVertex3f(-0.2, 0.0, 1.011);
glVertex3f(-0.2, 0.5, 1.011);
glVertex3f(-0.9, 0.5, 1.011);
glVertex3f(-0.9, 0.0, 1.011);
glEnd();

//------entrance--------


glColor3f(1.0, 1.0, 0.9);    //WALL
glBegin(GL_POLYGON);
glVertex3f(-0.7, 0.0, 3);
glVertex3f(-1, 0.0, 3);
glVertex3f(-1, 0.5, 3);
glVertex3f(-0.7, 0.5, 3);
glEnd();
```

```
glColor3f(1.0, 1.0, 0.9);    //WALL
glBegin(GL_POLYGON);
glVertex3f(-0.4, 0.0, 3);
glVertex3f(1, 0.0, 3);
glVertex3f(1, 0.5, 3);
glVertex3f(-0.4, 0.5, 3);
glEnd();

glColor3f(0.0, 0.0, 0.0);      //design wall
glPushMatrix();
glTranslatef(0.5, 0.4, 2.99);
glScalef(1, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 0.1, 0.0);      //design
glPushMatrix();
glTranslatef(0.4, 0.3, 2.99);
glScalef(1.0, 1.0, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.0, 0.0, 0.0);      //design
glPushMatrix();
glTranslatef(0.3, 0.2, 2.99);
glScalef(1.0, 1.0, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.0, 0.0, 0.0);      //design
glPushMatrix();
glTranslatef(0.3, 0.4, 2.99);
glScalef(1.0, 1.0, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.0, 0.0, 0.0);      //design
glPushMatrix();
glTranslatef(0.5, 0.2, 2.99);
glScalef(1.0, 1.0, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.0, 0.0, 0.0);      //design
glPushMatrix();
glTranslatef(0.985, 0.2, 2.99);
glScalef(0.03, 5.0, 0.01);
glutSolidCube(0.2);
glPopMatrix();

glColor3f(0.0, 0.0, 0.0);      //design
glPushMatrix();
```

```
        glTranslatef(-0.996, 0.2, 2.99);
        glScalef(0.03, 5.0, 0.01);
        glutSolidCube(0.2);
        glPopMatrix();

        glColor3f(1.0, 1.0, 0.9);   //WALL
        glBegin(GL_POLYGON);
        glVertex3f(-0.7, 0.3, 3);
        glVertex3f(-0.4, 0.3, 3);
        glVertex3f(-0.4, 0.5, 3);
        glVertex3f(-0.7, 0.5, 3);
        glEnd();

        glColor3f(0.1,0,1.0);  //shield
        glPushMatrix();
        glTranslatef(-0.55,0.4,3.0);
        glScalef(0.1,0.1,0.1);
        glutSolidTorus(0.07,0.2,5,25);
        glPopMatrix();

        glColor3f(0.9, 0.9, 0.9);
        glPushMatrix();
        glTranslatef(-0.55, 0.4, 3.0);
        glScalef(0.1, 0.1, 0.1);
        glutSolidTorus(0.08, 0.08, 5, 25);
        glPopMatrix();


        glColor3f(0.1, 0.0, 1.0);
        glPushMatrix();
        glTranslatef(-0.55, 0.4, 3.0);
        glScalef(0.1, 0.1, 0.1);
        glutSolidTorus(0.0799, 0.0799, 5, 25);
        glPopMatrix();




        mtop();
        mtv();
        mdesign();
        msofa();

}
void kbox()
{
        //--front-----
        glColor3f(0.22, 0.22, 0.22);    //corner box
        glPushMatrix();
        glTranslatef(-2.7, 0.17, 1.585);
        glScalef(5, 3.5, 1.6);
        glutSolidCube(0.1);
        glPopMatrix();
```

```
glColor3f(0.22, 0.22, 0.22);   //first box
glPushMatrix();
glTranslatef(-1.7, 0.17, 1.585);
glScalef(5, 3.5, 1.6);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.85, 0.69, 0.35);    //oven box
glPushMatrix();
glTranslatef(-2.2, 0.11, 1.585);
glScalef(5, 2.5, 1.6);
glutSolidCube(0.1);
glPopMatrix();

//oven

glColor3f(0.58, 0.58, 0.58);   //lower box border
glPushMatrix();
glTranslatef(-2.2, 0.07, 1.68);
glScalef(1.93, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.058, 0.058, 0.058); //lower black box
glPushMatrix();
glTranslatef(-2.2, 0.07, 1.681);
glScalef(1.92, 0.7, 0.01);
glutSolidCube(0.1);
glPopMatrix();


glColor3f(0.58, 0.58, 0.58); //upper box border
glPushMatrix();
glTranslatef(-2.2, 0.18, 1.68);
glScalef(1.93, 1, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.058, 0.058, 0.058);  //upper black box
glPushMatrix();
glTranslatef(-2.2, 0.18, 1.681);
glScalef(1.92, 0.7, 0.01);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.58, 0.58, 0.58);   //upper mid grey box
glPushMatrix();
glTranslatef(-2.2, 0.18, 1.682);
glScalef(1.5, 0.4, 0.01);
glutSolidCube(0.1);
glPopMatrix();
//----back-----------------
```

```
glColor3f(0.85, 0.81, 0.72);    //yellow table background
glPushMatrix();
glTranslatef(-2.9, 0.06, 2.25);
glScalef(2, 2, 10);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.1, 0.1, 0.1);      //yellow table top
glPushMatrix();
glTranslatef(-2.9, 0.166, 2.25);
glScalef(2, 0.1, 10);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.1, 0.1, 0.1);     //base black
glPushMatrix();
glTranslatef(-2.88, 0.00, 2.25);
glScalef(2, 0.1, 10);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.1, 0.1, 0.1);    //side black border
glPushMatrix();
glTranslatef(-2.9, 0.06, 2.755);
glScalef(2, 2.1, 0.1);
glutSolidCube(0.1);
glPopMatrix();

//-----------side

glColor3f(0.22, 0.22, 0.22);       //chair ke samne wali table
glPushMatrix();
glTranslatef(-1.8, 0.06, 2.93);
glScalef(7, 2, 1.6);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.22, 0.22, 0.22);       //kitchen ki first table
glPushMatrix();
glTranslatef(-1.4, 0.06, 2.7);
glScalef(1.6, 2, 6);
glutSolidCube(0.1);
glPopMatrix();

//white
glColor3f(0.9, 0.9, 0.9);
glPushMatrix();                    //samne wali table ka white design
glTranslatef(-1.32, 0.07, 2.71);
glScalef(0.1, 0.8, 5.5);
glutSolidCube(0.1);
glPopMatrix();
```

```
glColor3f(0.9, 0.9, 0.9);              //samne wali table ki peeche ka white design
glPushMatrix();
glTranslatef(-1.485, 0.07, 2.71);
glScalef(0.1, 0.8, 5.5);
glutSolidCube(0.1);
glPopMatrix();


glColor3f(0.9, 0.9, 0.9);             //chair ki samne wali table ka white design
glPushMatrix();
glTranslatef(-1.83, 0.07, 2.85);
glScalef(5.6, 0.8, 0.1);
glutSolidCube(0.1);
glPopMatrix();

//black

glColor3f(0.05, 0.05, 0.05);          //chair ke samne wali table ka top
glPushMatrix();
glTranslatef(-1.8, 0.165, 2.93);
glScalef(7, 0.1, 1.75);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.05, 0.05, 0.05);          //samne wali table ka top
glPushMatrix();
glTranslatef(-1.4, 0.165, 2.7);
glScalef(1.75, 0.1, 6);
glutSolidCube(0.1);
glPopMatrix();



//------design---------

 //wardrobe lines
glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-1.59, 0.17, 1.672);
glScalef(0.01, 3.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-1.8, 0.17, 1.672);
glScalef(0.01, 3.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-2.6, 0.17, 1.672);
```

```
glScalef(0.01, 3.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(1.0, 1.0, 1.0);
glPushMatrix();
glTranslatef(-2.8, 0.17, 1.672);
glScalef(0.01, 3.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();

//black lines
glColor3f(0.22, 0.22, 0.22);   //oven wali lines
glPushMatrix();
glTranslatef(-2.1, 0.11, 1.672);
glScalef(0.01, 2.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.22, 0.22, 0.22);
glPushMatrix();
glTranslatef(-2.3, 0.11, 1.672);
glScalef(0.01, 2.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();

//handel wardrobe and oven wale

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.36, 0.21, 1.671);
glScalef(0.5, 0.15, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.0, 0.21, 1.671);
glScalef(0.5, 0.15, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-1.85, 0.3, 1.671);
glScalef(0.5, 0.15, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-1.535, 0.3, 1.671);
glScalef(0.5, 0.15, 0.05);
```

```
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.65, 0.3, 1.671);
glScalef(0.5, 0.15, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.55, 0.3, 1.671);
glScalef(0.5, 0.15, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-1.63, 0.19, 1.671);
glScalef(0.15, 0.5, 0.05);
glutSolidCube(0.1);
glPopMatrix();


//black wall design
glColor3f(0.22, 0.22, 0.22);    //kitchen ke samne wali
glPushMatrix();
glTranslatef(-3.0, 0.44, 2.25);
glScalef(0.1, 1.2, 15);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.22, 0.22, 0.22);
glPushMatrix();
glTranslatef(-3.0, 0.2, 2.88);
glScalef(0.1, 4, 2.5);
glutSolidCube(0.1);
glPopMatrix();

//----yellow tabel boxes

glColor3f(0.1, 0.1, 0.1);         //middle line lambi wali
glPushMatrix();
glTranslatef(-2.79, 0.08, 2.25);
glScalef(0.05, 0.05, 10);
glutSolidCube(0.1);
glPopMatrix();


glColor3f(0.1, 0.1, 0.1);   //partitions vertical lines
glPushMatrix();
glTranslatef(-2.8, 0.06, 2.5);
```

```
glScalef(0.05, 2.1, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.1, 0.1, 0.1);
glPushMatrix();
glTranslatef(-2.8, 0.06, 2.25);
glScalef(0.05, 2.1, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.1, 0.1, 0.1);
glPushMatrix();
glTranslatef(-2.8, 0.06, 2.1);
glScalef(0.05, 2.1, 0.05);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.1, 0.1, 0.1);
glPushMatrix();
glTranslatef(-2.8, 0.06, 1.95);
glScalef(0.05, 2.1, 0.05);
glutSolidCube(0.1);
glPopMatrix();

// yellow table handels

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.8, 0.13, 2.45);
glScalef(0.05, 0.15, 0.55);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.8, 0.06, 2.45);
glScalef(0.05, 0.15, 0.55);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.8, 0.06, 2.55);
glScalef(0.05, 0.15, 0.55);
glutSolidCube(0.1);
glPopMatrix();

glColor3f(0.56, 0.56, 0.52);
glPushMatrix();
glTranslatef(-2.8, 0.13, 2.55);
glScalef(0.05, 0.15, 0.55);
glutSolidCube(0.1);
```

```
	glPopMatrix();

	glColor3f(0.56, 0.56, 0.52);
	glPushMatrix();
	glTranslatef(-2.8, 0.13, 2.05);
	glScalef(0.05, 0.15, 0.55);
	glutSolidCube(0.1);
	glPopMatrix();


	glColor3f(0.56, 0.56, 0.52);
	glPushMatrix();
	glTranslatef(-2.8, 0.06, 2.05);
	glScalef(0.05, 0.15, 0.55);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.56, 0.56, 0.52);
	glPushMatrix();
	glTranslatef(-2.8, 0.06, 2.15);
	glScalef(0.05, 0.15, 0.55);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.56, 0.56, 0.52);
	glPushMatrix();
	glTranslatef(-2.8, 0.13, 2.15);
	glScalef(0.05, 0.15, 0.55);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.56, 0.56, 0.52);
	glPushMatrix();
	glTranslatef(-2.8, 0.13, 1.9);
	glScalef(0.05, 0.15, 0.55);
	glutSolidCube(0.1);
	glPopMatrix();

	glColor3f(0.56, 0.56, 0.52);
	glPushMatrix();
	glTranslatef(-2.8, 0.06, 1.9);
	glScalef(0.05, 0.15, 0.55);
	glutSolidCube(0.1);
	glPopMatrix();
}
void kchair()
{
	GLUquadricObj* quadratic;
	quadratic = gluNewQuadric();

	glColor3f(0.58, 0.56, 0.56);    //grey wala chair support
	glPushMatrix();
	glTranslatef(-1.8, 0.13, 2.63);
```

```
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.01, 0.005, 0.125, 100, 100);
        glPopMatrix();


        glColor3f(0.09, 0.09, 0.09);     //chairs ka base
        glPushMatrix();
        glTranslatef(-1.8, 0.01, 2.63);
        glScalef(1, 0.15, 1);
        glutSolidSphere(0.04, 100, 100);
        glPopMatrix();

        glColor3f(0.85, 0.69, 0.35);    //chair main seat
        glPushMatrix();
        glTranslatef(-1.8, 0.15, 2.63);
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.05, 0.05, 0.02, 100, 100);
        glPopMatrix();


        glColor3f(0.85, 0.69, 0.35);      //chair ke neeche ka yellow(pole wala part)
        glPushMatrix();
        glTranslatef(-1.8, 0.13, 2.63);
        glRotatef(90, 1, 0, 0);
        gluCylinder(quadratic, 0.01, 0.009, 0.035, 100, 100);
        glPopMatrix();
}

void kitchen()          //2 X 1.5 dimension   0.5 ki height
{

        glColor3f(0.68, 0.54, 0.32);  //kitchen ka brown floor
        glBegin(GL_POLYGON);
        glVertex3f(-1, 0.0, 3);
        glVertex3f(-3, 0.0, 3);
        glVertex3f(-3, 0.0, 1.5);
        glVertex3f(-1, 0.0, 1.5);
        glEnd();

        glColor3f(0.9, 0.9, 0.9);     //right wall grey
        glBegin(GL_POLYGON);
        glVertex3f(-3, 0.0, 1.5);
        glVertex3f(-1, 0.0, 1.5);
        glVertex3f(-1, 0.5, 1.5);
        glVertex3f(-3, 0.5, 1.5);
        glEnd();

        glColor3f(0.9, 0.9, 0.9);       //left wall grey wali
        glBegin(GL_POLYGON);
        glVertex3f(-3, 0.0, 3);
        glVertex3f(-1, 0.0, 3);
        glVertex3f(-1, 0.5, 3);
        glVertex3f(-3, 0.5, 3);
```

```
glEnd();


glColor3f(0.85, 0.8, 0.85);  //roof kitchen
glBegin(GL_POLYGON);
glVertex3f(-1, 0.5, 3);
glVertex3f(-3, 0.5, 3);
glVertex3f(-3, 0.5, 1.5);
glVertex3f(-1, 0.5, 1.5);
glEnd();

glColor3f(0.9, 0.9, 0.9);    //back wall kitchen
glBegin(GL_POLYGON);
glVertex3f(-3, 0.0, 1.5);
glVertex3f(-3, 0.5, 1.5);
glVertex3f(-3, 0.5, 3);
glVertex3f(-3, 0.0, 3);
glEnd();

//---entrance

glColor3f(0.75, 0.80, 0.46);     //entrance ka right wali hari patti
glBegin(GL_POLYGON);
glVertex3f(-1, 0.0, 1.5);
glVertex3f(-1, 0.5, 1.5);
glVertex3f(-1, 0.5, 1.85);
glVertex3f(-1, 0.0, 1.85);
glEnd();

glColor3f(0.75, 0.80, 0.46);         //entrance ka top hari patti
glBegin(GL_POLYGON);
glVertex3f(-1, 0.35, 1.5);
glVertex3f(-1, 0.5, 1.5);
glVertex3f(-1, 0.5, 3);
glVertex3f(-1, 0.35, 3);
glEnd();

//design

glColor3f(0.0, 0.0, 0.0);             //bottom ka cube
glPushMatrix();
glTranslatef(-0.99, 0.14, 1.66);
glScalef(0.1, 1, 1.5);
glutSolidCube(0.1);
glPopMatrix();


glColor3f(0.0, 0.0, 0.0);       //middle cube black
glPushMatrix();
glTranslatef(-0.99, 0.275, 1.66);
glScalef(0.1, 1, 1.5);
glutSolidCube(0.1);
glPopMatrix();
```

```
        glColor3f(0.0, 0.0, 0.0);    //top ka black cube
        glPushMatrix();
        glTranslatef(-0.99, 0.4, 1.66);
        glScalef(0.1, 1, 1.5);
        glutSolidCube(0.1);
        glPopMatrix();

        glColor3f(0.8, 0.4, 0.3);            //side wali kite
        glPushMatrix();
        glTranslatef(-1.03, 0.34, 1.45);
        glScalef(0.08, 0.08, 0.08);
        glutSolidOctahedron();
        glPopMatrix();

        glColor3f(0.8, 0.4, 0.3);
        glPushMatrix();
        glTranslatef(-1.03, 0.18, 1.45);
        glScalef(0.08, 0.08, 0.08);
        glutSolidOctahedron();
        glPopMatrix();

        glColor3f(0.8, 0.4, 0.3);
        glPushMatrix();
        glTranslatef(-1.03, 0.34, 1.29);
        glScalef(0.08, 0.08, 0.08);
        glutSolidOctahedron();
        glPopMatrix();

        glColor3f(0.8, 0.4, 0.3);
        glPushMatrix();
        glTranslatef(-1.03, 0.18, 1.29);
        glScalef(0.08, 0.08, 0.08);
        glutSolidOctahedron();
        glPopMatrix();

        kbox();

        glPushMatrix();
        kchair();
        glPopMatrix();

        glPushMatrix();
        glTranslatef(-0.2, 0.0, 0.0);
        kchair();
        glPopMatrix();
}
void swimming()
{
        glColor3f(0.75, 0.80, 0.46);
        glBegin(GL_POLYGON);
        glVertex3f(-1, 0.0, 1);
        glVertex3f(-1, 0.5, 1);
```

```
        glVertex3f(-1, 0.5, 1.5);
        glVertex3f(-1, 0.0, 1.5);
        glEnd();

}
void maindoor()
{
        //door
        glColor3f(0.095, 0.095, 0.095);
        glBegin(GL_POLYGON);
        glVertex3f(-0.7, 0.3, 3);
        glVertex3f(-0.55 - xx, 0.3, 3 + zz);  //update left door values
        glVertex3f(-0.55 - xx, 0.0, 3 + zz);
        glVertex3f(-0.7, 0.0, 3);
        glEnd();

        glColor3f(0.095, 0.095, 0.095);
        glBegin(GL_POLYGON);
        glVertex3f(-0.4, 0.3, 3);   //update right door values
        glVertex3f(-0.55 + xx, 0.3, 3 + zz);
        glVertex3f(-0.55 + xx, 0.0, 3 + zz);
        glVertex3f(-0.4, 0.0, 3);
        glEnd();

        //border

        glColor3f(0.23, 0.20, 0.23);
        glBegin(GL_POLYGON);
        glVertex3f(-0.401, 0.3, 3.01);
        glVertex3f(-0.38, 0.3, 3.01);
        glVertex3f(-0.38, 0.0, 3.01);
        glVertex3f(-0.401, 0.0, 3.01);
        glEnd();

        glColor3f(0.23, 0.20, 0.23);
        glBegin(GL_POLYGON);            //left border
        glVertex3f(-0.699, 0.3, 3.01);
        glVertex3f(-0.72, 0.3, 3.01);
        glVertex3f(-0.72, 0.0, 3.01);
        glVertex3f(-0.699, 0.0, 3.01);
        glEnd();

        glColor3f(0.23, 0.20, 0.23);
        glBegin(GL_POLYGON);
        glVertex3f(-0.38, 0.299, 3.01);
        glVertex3f(-0.72, 0.299, 3.01);
        glVertex3f(-0.72, 0.32, 3.01);
        glVertex3f(-0.38, 0.32, 3.01);
        glEnd();

        if (xx1)   //xx1=0 remain constant
        {
```

```
//door design

glColor3f(0.35, 0.34, 0.35);
glBegin(GL_POLYGON);
glVertex3f(-0.65, 0.2, 3.01);
glVertex3f(-0.6, 0.2, 3.01);
glVertex3f(-0.6, 0.1, 3.01);
glVertex3f(-0.65, 0.1, 3.01);
glEnd();

glColor3f(0.35, 0.34, 0.35);
glBegin(GL_POLYGON);
glVertex3f(-0.45, 0.2, 3.01);
glVertex3f(-0.5, 0.2, 3.01);
glVertex3f(-0.5, 0.1, 3.01);
glVertex3f(-0.45, 0.1, 3.01);
glEnd();


//line

glColor3f(0.9095, 0.9095, 0.9095);
glBegin(GL_POLYGON);
glVertex3f(-0.45, 0.099, 3.01);
glVertex3f(-0.55, 0.099, 3.01);
glVertex3f(-0.55, 0.1, 3.01);
glVertex3f(-0.45, 0.1, 3.01);
glEnd();

glColor3f(0.9095, 0.9095, 0.9095);
glBegin(GL_POLYGON);
glVertex3f(-0.45, 0.3, 3.02);
glVertex3f(-0.449, 0.3, 3.02);
glVertex3f(-0.449, 0.05, 3.02);
glVertex3f(-0.45, 0.05, 3.02);
glEnd();

glColor3f(0.9095, 0.9095, 0.9095);
glBegin(GL_POLYGON);
glVertex3f(-0.65, 0.2, 3.02);
glVertex3f(-0.55, 0.2, 3.02);
glVertex3f(-0.55, 0.201, 3.02);
glVertex3f(-0.65, 0.201, 3.02);
glEnd();

glColor3f(0.9095, 0.9095, 0.9095);
glBegin(GL_POLYGON);
glVertex3f(-0.65, 0.0, 3.02);
glVertex3f(-0.651, 0.0, 3.02);
glVertex3f(-0.651, 0.25, 3.02);
glVertex3f(-0.65, 0.25, 3.02);
glEnd();
```

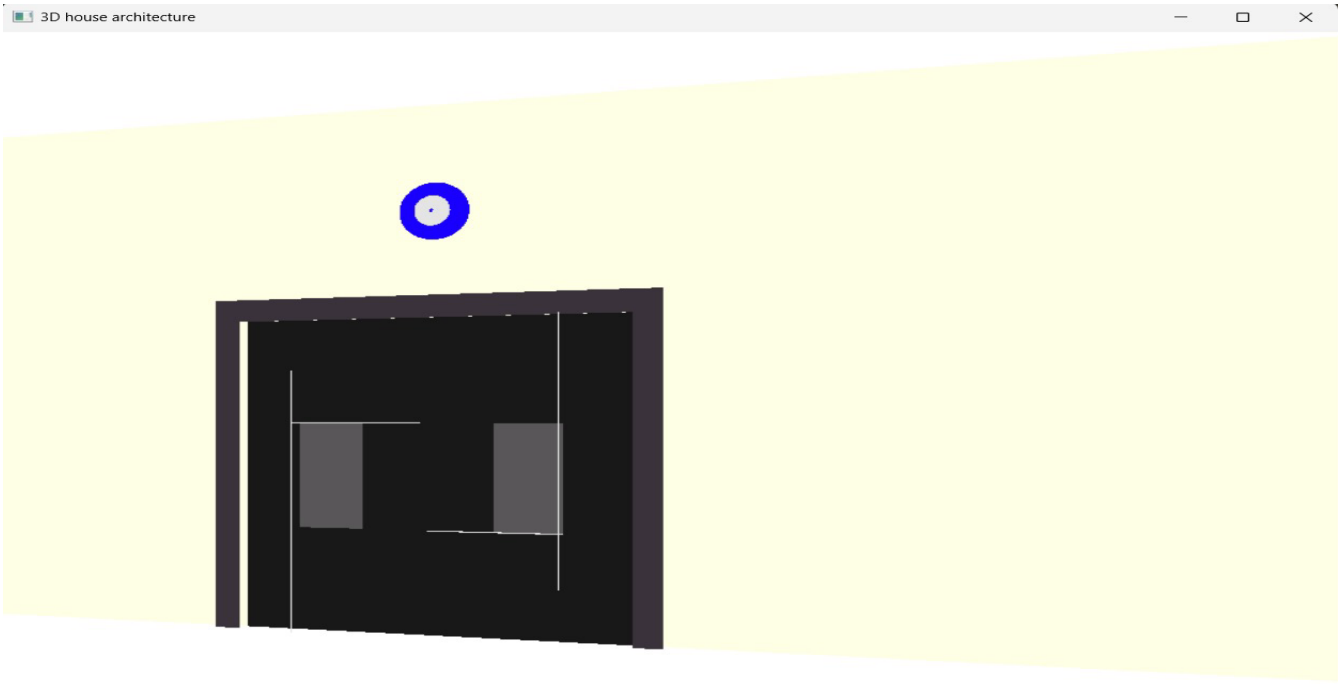
```
        }

}
void outercover()
{

        glColor3f(0.095, 0.095, 0.095);
        glBegin(GL_POLYGON);
        glVertex3f(-1, 0.0, 3);
        glVertex3f(-1, 0.5, 3);
        glVertex3f(-1, 0.5, 4);
        glVertex3f(-1, 0.0, 4);
        glEnd();


}
void house()
{
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

        gluLookAt(viewer[0], viewer[1], viewer[2], viewer[0] + lx, lat[1], viewer[2] + lz, 0.0, 1.0, 0.0);


        outercover();
        maindoor();
        room1();
        room2();
        kitchen();
        mid();
        swimming();
        glutSwapBuffers();
}
void keys(unsigned char key, int x, int y)
{
        if (key == 'x')viewer[0] -= 0.1;
        if (key == 'X')viewer[0] += 0.1;
        if (key == 'y')viewer[1] -= 0.1;
        if (key == 'Y')viewer[1] += 0.1;
        if (key == 'z')viewer[2] -= 0.1;
        if (key == 'Z')viewer[2] += 0.1;
        if (key == 'o') doort(0);

        glutPostRedisplay();
}

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

        int q = viewer[0];
```

```
	int w = viewer[1];
	int e = viewer[2];
	float fraction = 0.1f;

	switch (key) {
	case GLUT_KEY_LEFT:
		angle -= 0.02f;
		lx = sin(angle);
		lz = -cos(angle);
		break;
	case GLUT_KEY_RIGHT:
		angle += 0.02f;
		lx = sin(angle);
		lz = -cos(angle);
		break;
	case GLUT_KEY_UP:

		viewer[0] += lx * fraction;
		viewer[2] += lz * fraction;

		break;
	case GLUT_KEY_DOWN:
		viewer[0] -= lx * fraction;
		viewer[2] -= lz * fraction;
		break;

	case GLUT_KEY_PAGE_UP:
		viewer[1] += 0.1;
		lat[1] += 0.1;
		break;
	case GLUT_KEY_PAGE_DOWN:
		viewer[1] -= 0.1;
		lat[1] -= 0.1;
		break;
	}
	glutPostRedisplay();
}
void changeSize(int w, int h)
{
	if (h == 0)
		h = 1;
	float ratio = w * 1.0 / h;


	glViewport(0, 0, w, h);
	glMatrixMode(GL_PROJECTION);


	glLoadIdentity();


	gluPerspective(40.0f, ratio, 0.1f, 10.0f);   //3D object projected on 2D screen (ver field of view,aspect
ratio,distance to nearest clipping plane
```

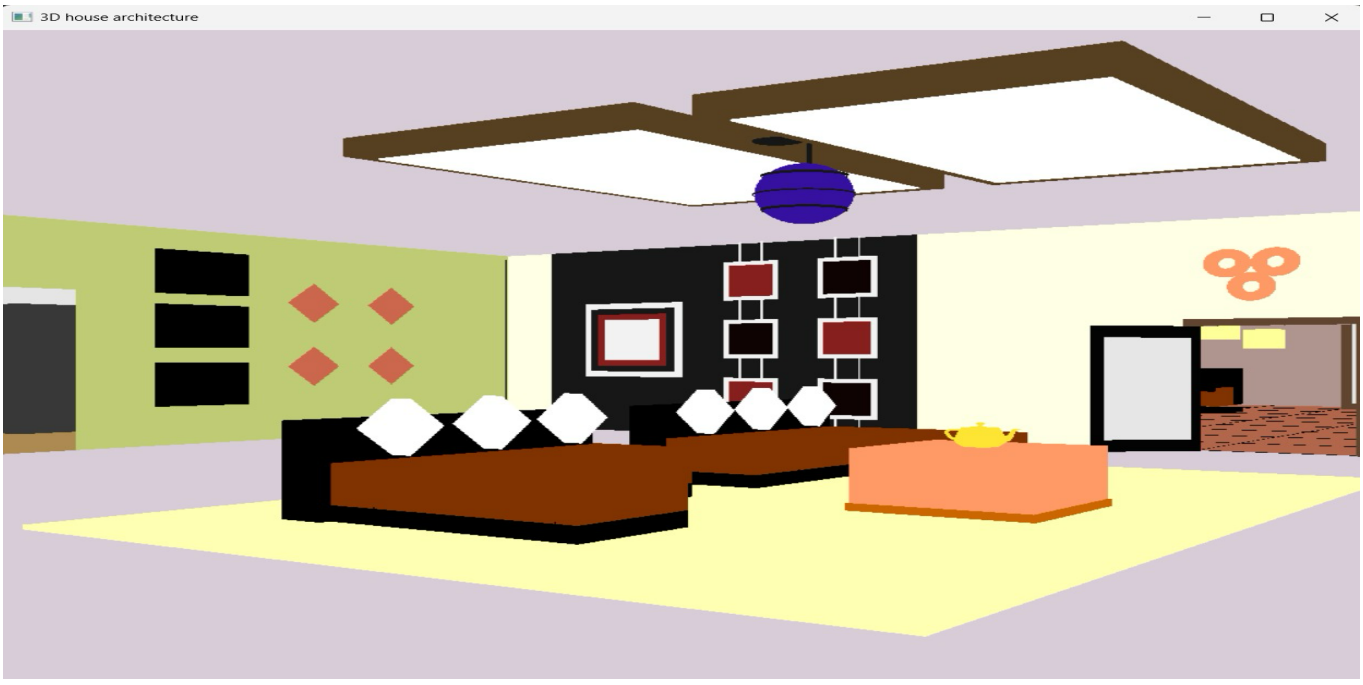
```
                                        //dist to far clipping plane
        glMatrixMode(GL_MODELVIEW);
        glutPostRedisplay();
}

int main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH | GLUT_ALPHA);
        glutInitWindowSize(1000, 800);
        glutInitWindowPosition(0, 0);
        glutCreateWindow("3D house architecture");

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();
        glutDisplayFunc(house);
        glutReshapeFunc(changeSize);
        glDepthFunc(GL_LEQUAL);
        glutSpecialFunc(specialKey);
        glutKeyboardFunc(keys);
        glEnable(GL_BLEND);
        glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
        glEnable(GL_DEPTH_TEST);
        tfs(0);
        glAlphaFunc(GL_GREATER, 0.1);
        glEnable(GL_ALPHA_TEST);

        glClearColor(1.0, 1.0, 1.0, 1.0);
        glutMainLoop();
        return 0;
}
```

5. Output / Screenshots

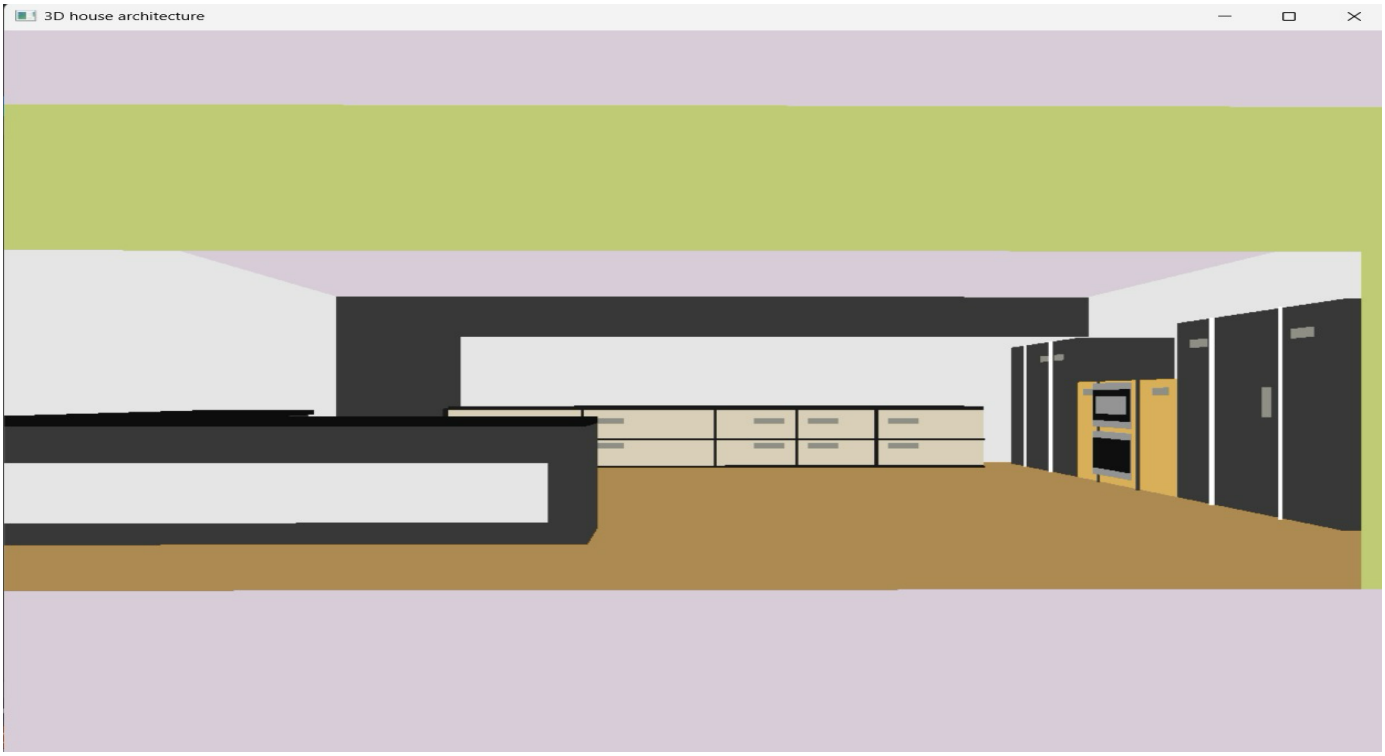
Main Door View :



Living Room View:



Kitchen View :



Room View :