Project Title : 3D Computer Lab

CS352:ComputerGraphics&Visualization Lab

Project Report

Course Instructor:

Dr. Somnath Dey

Submitted By:

Allu Mrudula– 200001002

Padamata Kanishka Sai–200001058

Rayala Navya Harshitha– 200003064

Introduction:

Our project title is 3D Computer Lab. OpenGL is a powerful graphics library that allows developers to create stunning 3D visualisations in real-time.

One of the most exciting applications of OpenGL is in the creation of virtual environments, such as computer labs. With the ability to create 3D models of objects, apply textures and materials, and set up lighting and camera angles, it is possible to create a realistic and immersive experience for users.

In this project, we will explore the process of designing a 3D computer lab using OpenGL. This will involve creating 3D models of the lab equipment which includes a Monitor ,CPU ,Keyboard ,Mouse, Cables and a 3D Room, Blackboard , Furniture like Chairs, Tables, Door , Windows ,Fans and setting up lighting and textures, and implementing OpenGL code to create the virtual environment.

For the model of the lab we took our campus computer lab as one of the references to make our design.

By the end of this project, we will have a better understanding of the capabilities of OpenGL and how it can be used to create realistic and interactive virtual environments.

Specifications

Libraries :

Here the libraries used mainly are of OpenGL such as <GL/glut.h> for keyboard,mouse input functions and windows resizing . Another library that is used is <SOIL/SOIL.h> for image rendering using textures and <bits/stdc++.h> header for writing the OpenGL code using C++ language.

The <GL/glut.h> header file is a library file in OpenGL.The acronym "GLUT" stands for "OpenGL Utility Toolkit".Thisprovides an interface for creating and managing windows and menus for OpenGL applications.

The OpenGL Utility Toolkit (GLUT) handles most of the system dependent actions required to display a window, put OpenGL graphics in it, and accept mouse and keyboard input. It is used to create the skeleton for all the assignments that hides Windows programming issues from you so you can concentrate on programming graphics. It also provides a set of predefined shapes and materials that can be used to create 3D models.

In other words the <GL/glut.h> header file is included in most OpenGL applications and is often used in conjunction with other header files such as <GL/gl.h> and <GL/glu.h>. It provides a convenient and easy-to-use interface for creating interactive 3D applications with OpenGL.

Some of the most commonly used functions in the <GL/glut.h> header file include:

glutCreateWindow(): creates a new window with the specified title.
glutDisplayFunc(): sets the function to be called whenever the window needs to be redrawn.
glutKeyboardFunc(): sets the function to be called when a keyboard key is pressed.
glutMouseFunc(): sets the function to be called when a mouse button is pressed.
glutIdleFunc(): sets the function to be called when there are no events to handle.

Other functions we used include
glPushMatrix(): Set current matrix on the stack and starts pushing the functions after this
glPopMatrix(): Pop the functions matrix till it is empty
glTranslatef(X, Y, Z): Function to translate *(X, Y, Z).*
glRotatef(angle,axis): Function to rotate by *angle* along *axis*. Ex: gl*Rotatef(*90,0,1,0).
glScalef(Sx, Sy, Sz): Scale the size of the geometry by *Sx, Sy,Sz* along *X, Y, Z* axis.
Example:

```
glPushMatrix();
glTranslatef(1, 2, 3);
glRotatef(90,0,1,0);
glScalef(2, 2, 1);
glPopMatrix();
```

In this, first we scale, then rotate and then translate.

*glLightfv(*GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
For Ambient light, diffuse, specular light

glLightfv(GL_LIGHT0, GL_POSITION, light_position);
For Light position

glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, texture);
glDisable(GL_TEXTURE_2D);
For Image rendering

How to run the program?

Step 1: To run the program first download the folder containing our program file and then open it in the terminal (For linux).
Step 2: Now run the command mentioned below
g++ cl.cpp -lGL -lGLU -lglut -lSOIL
which runs the c++ OpenGL code and creates the output file
Step 3: Then run *./a.out* to get the output window.

Key controls :

Along with the output window, the terminal also mentions the key controls as output in our program . Below mentioned are controls used
Press : 1 -> Left Light ON
Press : 2 -> Right Light ON
Press : w -> move Up
Press : s -> move Down
Press : a -> move left
Press : d -> move Right
Press : + -> Zoom In
Press : - -> Zoom Out

Press : i -> increase fan speed
Press : u -> decrease fan speed
Press : o -> Turn on / off the computers

Functionalities Implemented

Below are some functions used to create objects.here i am mentioning some inbuilt function and the objects they created in the model.

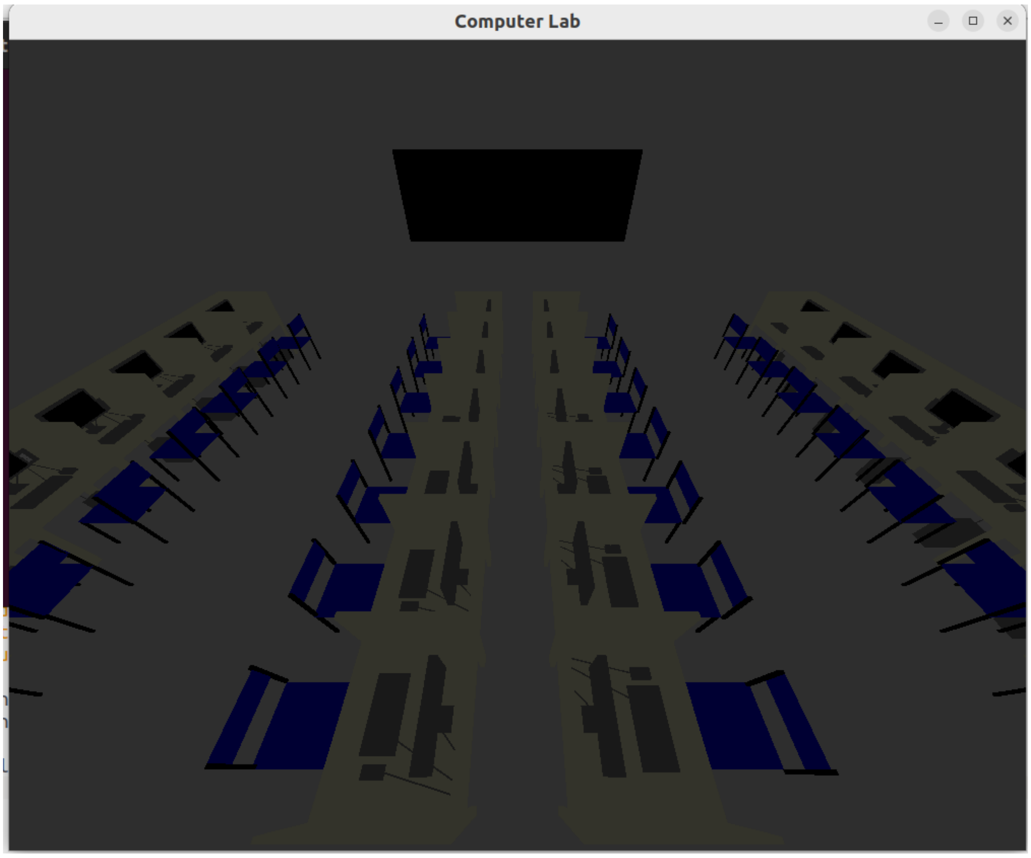
```
void cube(R,G,B){

// This function is used to create a cube of size 1  throughout the code to modify the objects as we
desired such as translate ,scale,rotate etc

}

void wall_light(){
//right wall light
  // left wall light
}
void wall_floor(){
//floor
  //left wall
    // right wall
    // Up (roof wall)
    // Front wall
    //back wall
    // Black board
}
void chair(){
 //base seat
// leg base 1
// leg base 2
// leg base 3
// leg base 4
//chair back
//chair sides
}

void computer(){
 //desktop
//screen
//stand
//keyboard
//cpu
//mouse
//cables
//switchboard
//above said objects are created in this function
}
```

```
void table(){

    // socket upper hole
    // socket left hole
    // socket left hole
    // left switch
    // right switch
    // middle switch
  }
void fan_rotation()
{
// this is for fan speed
    fan_rt = fan_rt+ speed;
    if(fan_rt>360)
        fan_rt =0;
    glutPostRedisplay();
}

void fan(){
  //head
//round base
//fan-leg  {3 fan wings}
}
void light_function_0(){
    For Light 1
}
void light_function_1(){
    For Light 2
}
void display_setting()
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();
    // xmin, xmax, ymin, ymax, near, far
    glFrustum(-5,5,-5,5, 4, 100);

    glMatrixMode( GL_MODELVIEW );
    glLoadIdentity();
    // eye, look, head
    gluLookAt(eyeX,eyeY,eyeZ, lookX,lookY,lookZ, 0,1,0);

    //glViewport(0, 0, windowHeight, windowWidth);
    glRotatef(theta,axis_x,axis_y,0);
```

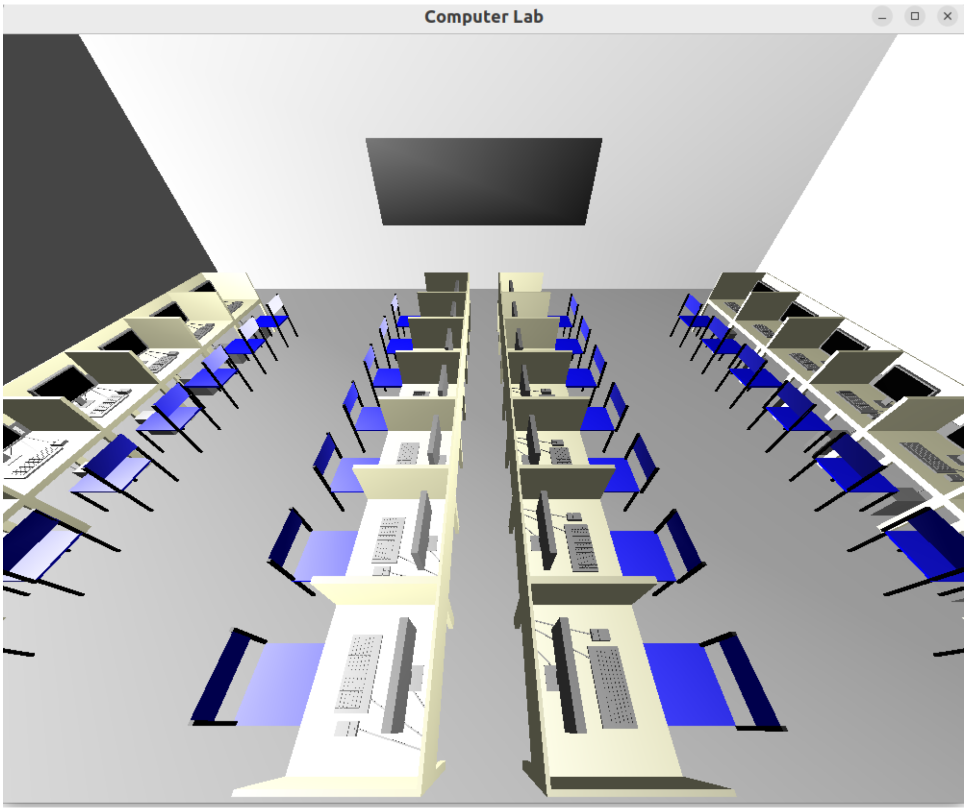
```
   glTranslatef(0,0,Tzval);
}

void display(void){
  Includes all the functions in display
  wall_floor()
  wall_light()
  fan()  {2 fans}
  light_function_0()
  light_funciton_1()

  table()
  chair()
  computer() {table, chair, computer for 4 rows}
}
```

Screenshots

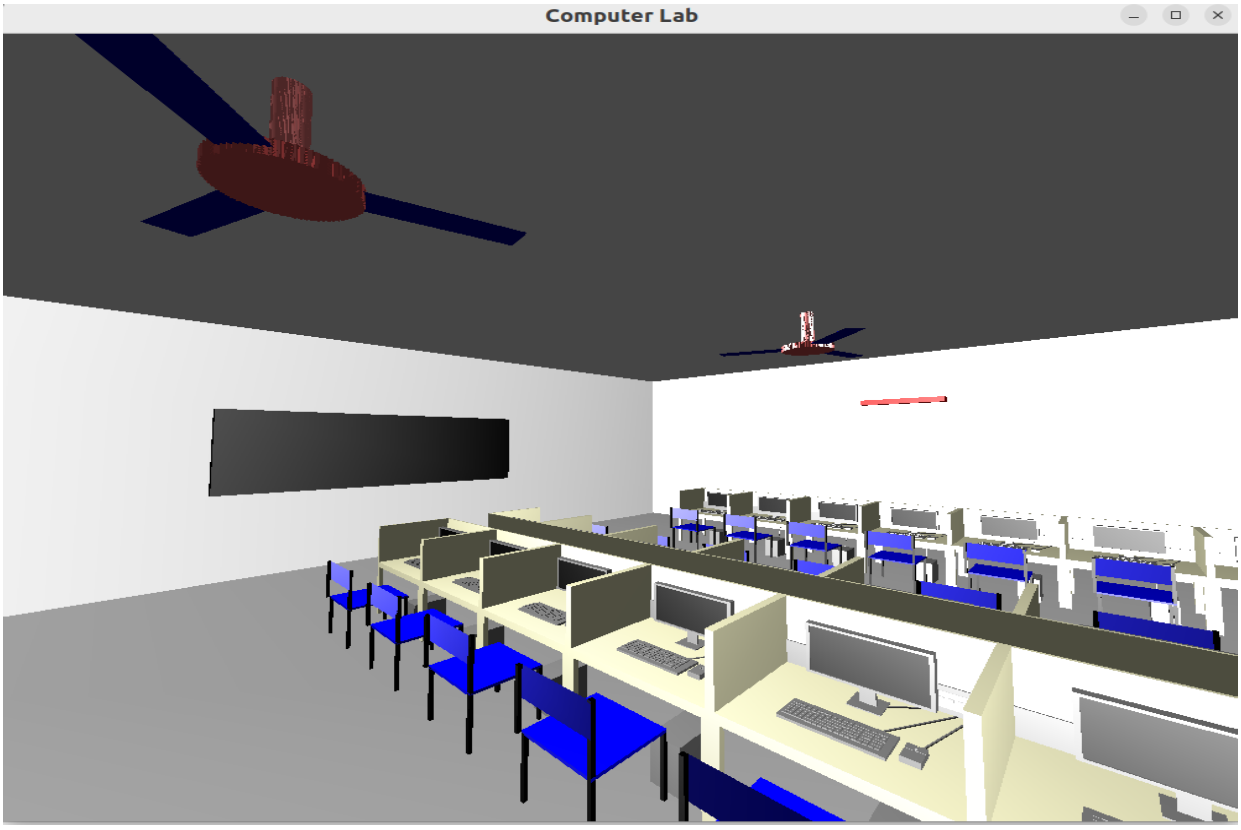
Initially lights are off



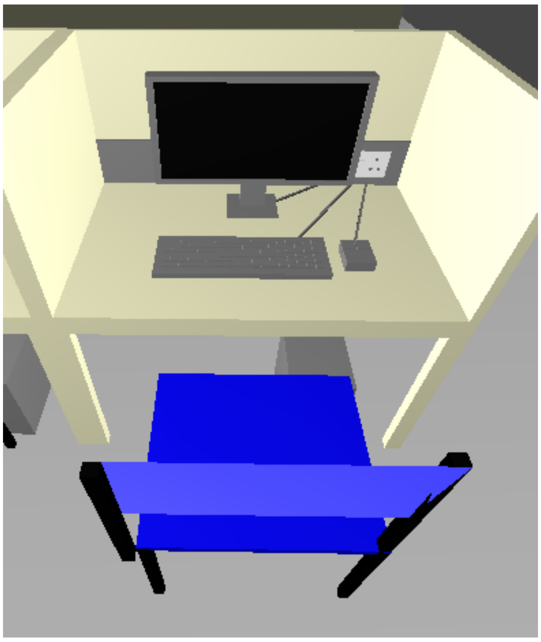
When lights are turn on



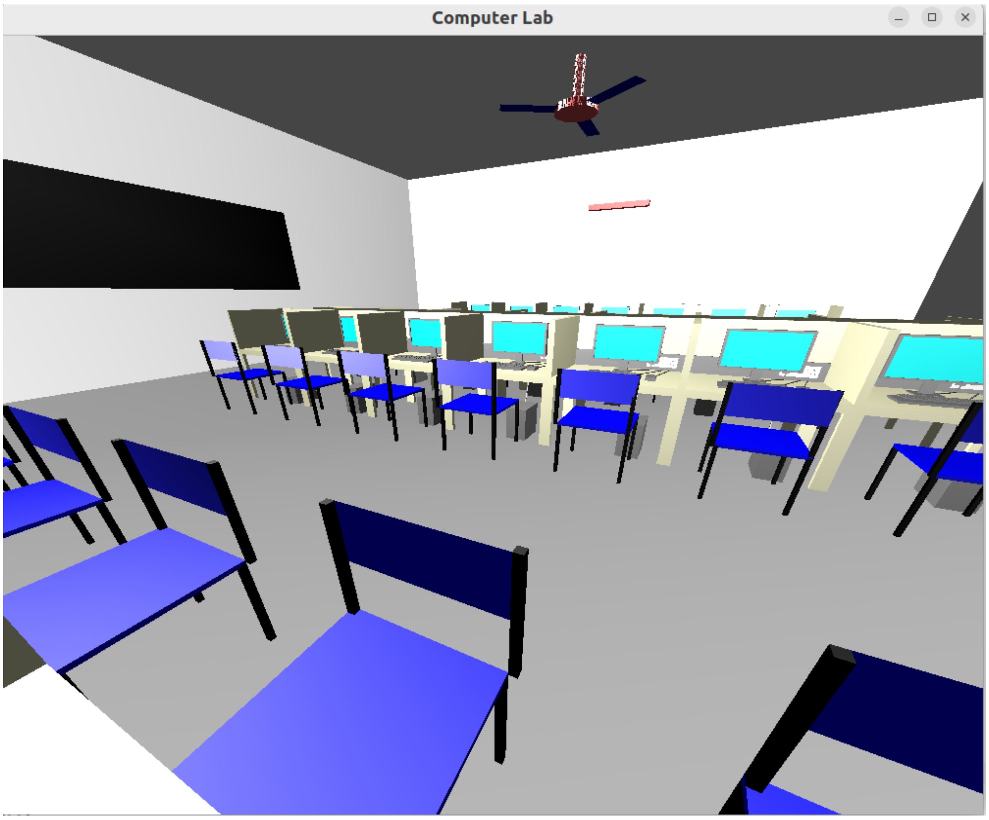
From this point of view we can see two fans ,blackboard, computers,tables, light,chairs



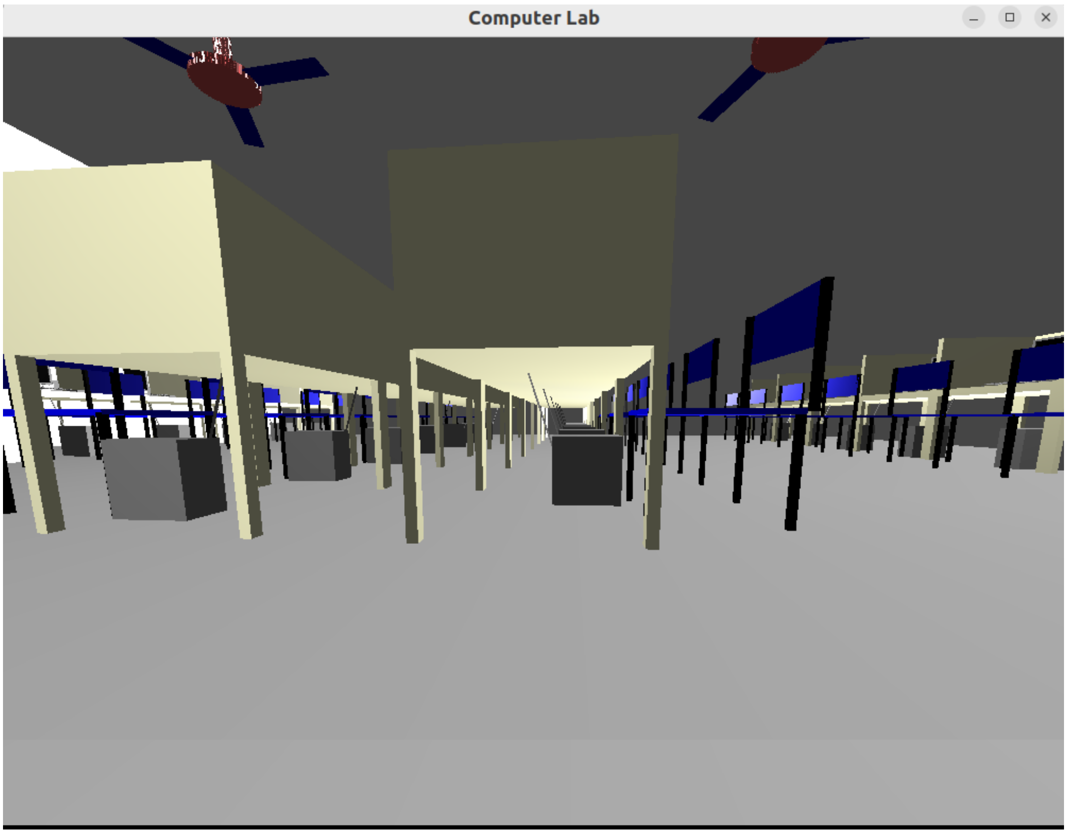
A computer unit with a monitor which is off ,keyboard ,CPU,cables ,mouse,chair, switchboard ,tables with partitions



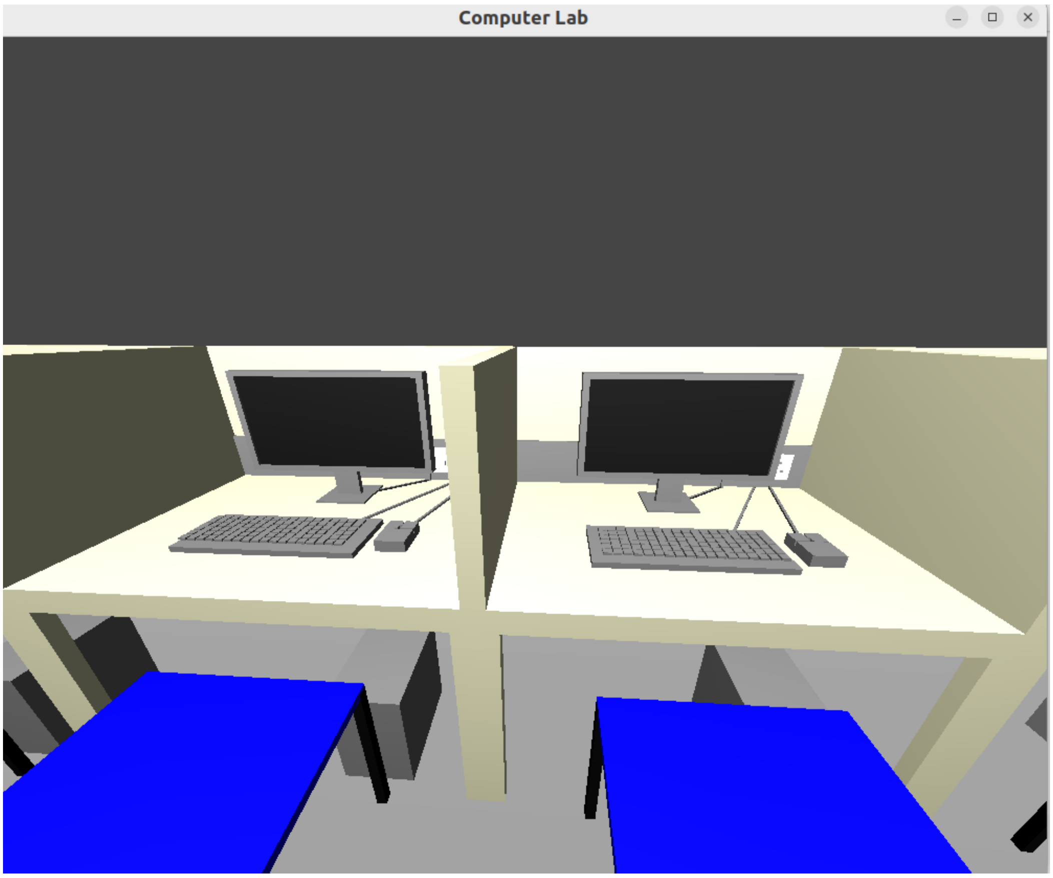
When monitors are on



We can CPU s under the clearly from this angle which are connected through cables



Clear view of keyboard ,mouse from this angle


Computer Lab