

# Modeling Of Pod Building

**CS352: Computer Graphics &  
Visualization Lab**

Project Report

Course Instructor:  
Dr. Somnath Dey

Submitted By:

Group Member 1: Likhith Raj DV - 200001039

Group Member 2 : M.Bhanu Prakash- 200001048

Group Member 3 : Abhishek Jaiswal - 200001001

## Introduction

POD modeling is a challenging process that demands careful planning and visualization to guarantee that the finished result fits the needs of its inhabitants. With the development of computer graphics and technology, we can now have access to strong tools like OpenGL, which let them make 3D environments that are both immersive and realistic. We can produce precise and detailed cityscapes that can be explored and enjoyed from a first-person perspective by making use of OpenGL's capabilities. OpenGL can also be used to render text and other information in the urban cities in addition to these visual aspects. This enables us to design educational and entertaining user experiences, whether users are exploring the city for fun or for utilitarian needs like navigation and wayfinding.

And as our project suggests, our goal is to build a 3D model of 'POD building'. It also includes some other features like cloud movement, tree construction.

Our Project primarily focuses on simulating pod construction, but it also incorporates trees, roads and moving clouds. Perspective projection was implemented. We also incorporated day and night effects. It also includes camera implementation about its own Axis.

## Specifications

Making a 3D model with finished structures, roads, trees, and other items is a part of POD building using OpenGL. As a result, viewers are able to observe and explore this model in real-time while looking at the city from various angles and viewpoints. The creation of realistic lighting and the simulation of environmental elements like the weather (clouds) and the time of day are also possible with OpenGL.

- **LIBRARIES REQUIRED:**

1. **OpenGL Library:** The fundamental library needed to render 3D visuals with OpenGL is this one. We can build and work with 3D visuals because of the collection of functions and data types it offers.
2. **GLU (OpenGL Utility Library):** It offers a number of useful features not included in the basic OpenGL library. Primitive shapes like spheres, cylinders, and cones can all be drawn using these functions.
3. **GLM (OpenGL Mathematics)** offers classes and functions for carrying out typical mathematical operations needed in 3D graphics programming, like matrix and vector operations.
4. **GLUT (OpenGL Utility Toolkit):** The task of constructing interactive 3D applications can be made simpler by GLUT's set of utilities for handling user input and creating windows.
  - I. **Window management:** When generating and managing windows, GLUT offers tools for adjusting the window's size, orientation, and title.
  - II. **User input handling:** For handling user input, including keyboard and mouse input, GLUT offers many functions. As a result, We may make dynamic 3D programmes that react to user input.
  - III. **Timer functions:** To build animations and other dynamic effects, GLUT has utilities for setting up timers inside the programme.
  - IV. **Menu creation:** In order to add more functionality and interactivity, GLUT offers tools for building menus inside the programme window.

Many developers choose to utilize more up-to-date libraries like GLFW because GLUT is an older library that is not actively updated.

NOTE: All the functionalities only the keys in the keyboard(No Mouse based functionalities).

- KEY CONTROLS:

1. KEY A : For left shifting the whole Camera.
2. KEY D : For right shifting the whole Camera.
3. KEY W : For forward shift in the Camera.
4. KEY S: For backward shift in the Camera.
5. KEY T: For downward shift in the Camera.
6. KEY G: For upward shift in the Camera.
7. KEY Q : For left rotation in the camera.
8. KEY E: For right rotation in the camera.
9. KEY L: For activating the night light effect.

- KEY CONTROL FUNCTIONALITIES :

1. on pressing the key 'A' the camera position will move towards the left by 5 units.
2. on pressing the key 'D' the camera position will move towards the right by 5 units.
3. on pressing the key 'W' the camera position will move away by 5 units.
4. on pressing the key 'S' the camera position will move towards the camera by 5 units.
5. on pressing the key 'Q' the camera position will be rotated by 5 degrees in anticlockwise direction.
6. on pressing the key 'Q' the camera position will be rotated by 5 degrees in clockwise direction.
7. on pressing the key 'T' the camera position will move downwards by 5 units.
8. on pressing the key 'G' the camera position will move upwards by 5 units.

## Functionalities Implemented

The functionalities in POD construction using OpenGL:

1. Building creation: OpenGL can be used to create and render 3D models of POD buildings, which allows us to design and visualize the structures of the city. This includes creating walls, roofs, windows, doors. For creating floor blocks and the doors we use default `glutSolidCube()` and for the window creation we used default functions like `drawCircle()` and also `glutSolidCube()`.

2. Lighting: OpenGL can simulate realistic lighting in the POD building city environment, including day and night light. This helps to create a more immersive and realistic environment in the POD modeling.

3. Object placement: In addition to POD buildings, OpenGL can also be used to place other objects in the POD building environment, such as trees and also clouds. This helps to create a more complete and interactive environment.

6. Navigation: OpenGL enables users to navigate the city environment in real-time, from a first-person perspective. This includes the ability to move in between the POD buildings, look around.

7. Text rendering: In addition to rendering 3D models and objects, OpenGL can also be used to render 2D text in the POD construction environment. This includes showing data like street names,

building names, and other pertinent data. We can easily include readable, high-quality text into the 3D environment using OpenGL. Bitmap fonts, texture fonts, and vector fonts—each of which offers varying degrees of fidelity and flexibility—can all be used to produce text in OpenGL. Whether viewers are exploring the city for fun or for utilitarian reasons like navigation and wayfinding, we can make their experience more educational and interesting by incorporating text into the urban landscape.

8. Road creation: We can utilize OpenGL to make unique meshes and objects for roads, including textures and other visual components like lane markers, in order to create roadways inside the city environment.

9. Lighting sources: We may design and manage various lighting sources for the urban area using OpenGL. This covers spotlighting, point lighting, and directional lighting. The effects that can be produced by these lighting sources include shadows, highlights, and shading.

10. Light intensity and color: We can modify the brightness and color of lighting sources in urban environments using OpenGL. This enables precise control over the environment's overall appearance and feel and can be utilized to produce various moods and ambiances.

- Functionalities used for the project:
  1. default functions like: `glTranslatef` , `glScalef` , `glutSolidCube`(for building each block in the pod).
  2. `drawCircle()` used for constructing windows.
  3. `glutSolidSpher()` used for tree pattern construction and also for cloud formation.
  4. `glutTimerFunc()` function used for iteratively redrawing the cloud structure after every 16 ms duration.
  5. `glutKeyboardFunc()` function used for utilizing the keys in the keyboard.

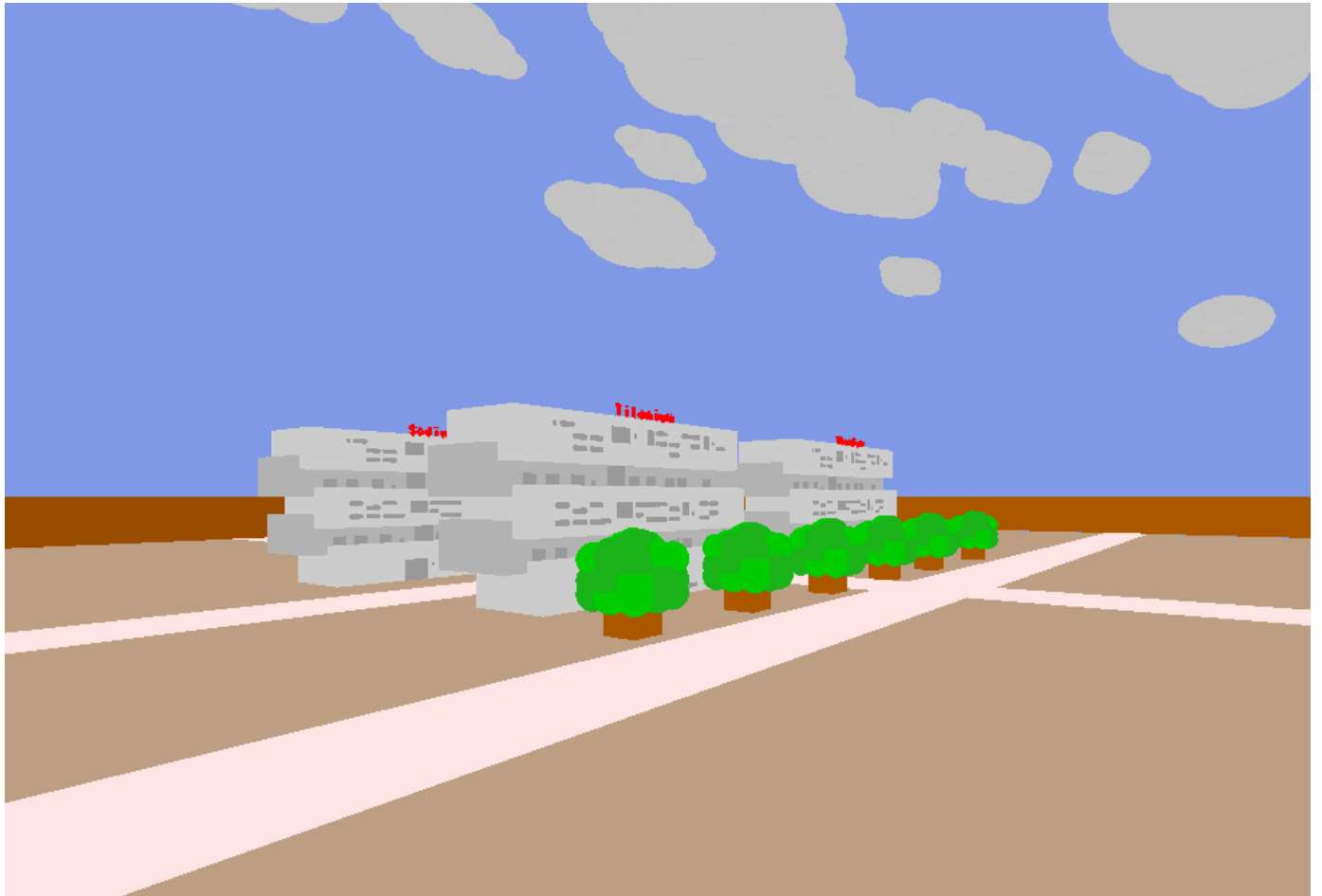
## Output

### 1. FrontView of the Project ( Modeling of POD Building )

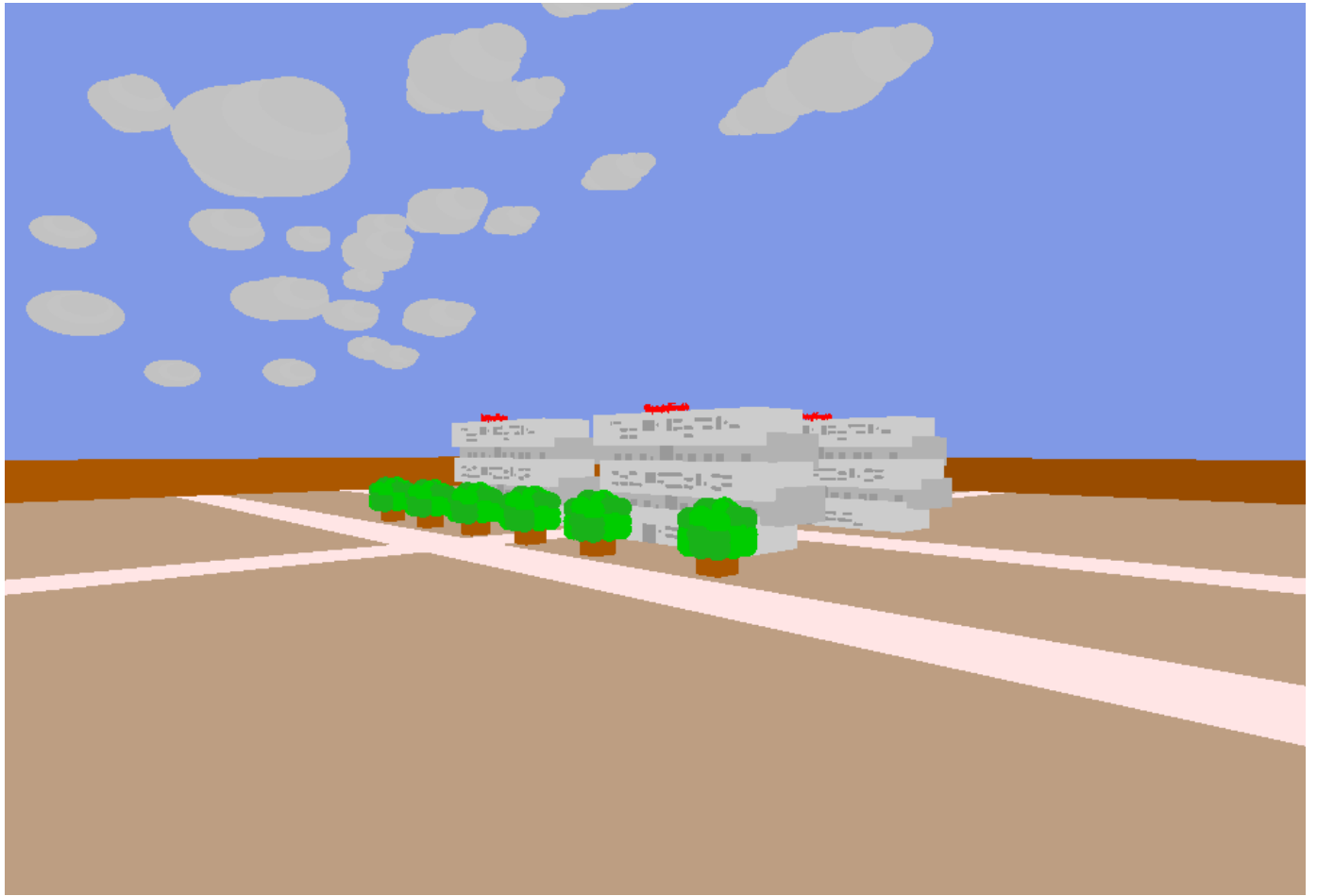
( POd building / road / moving cloud / Tree/Texture)



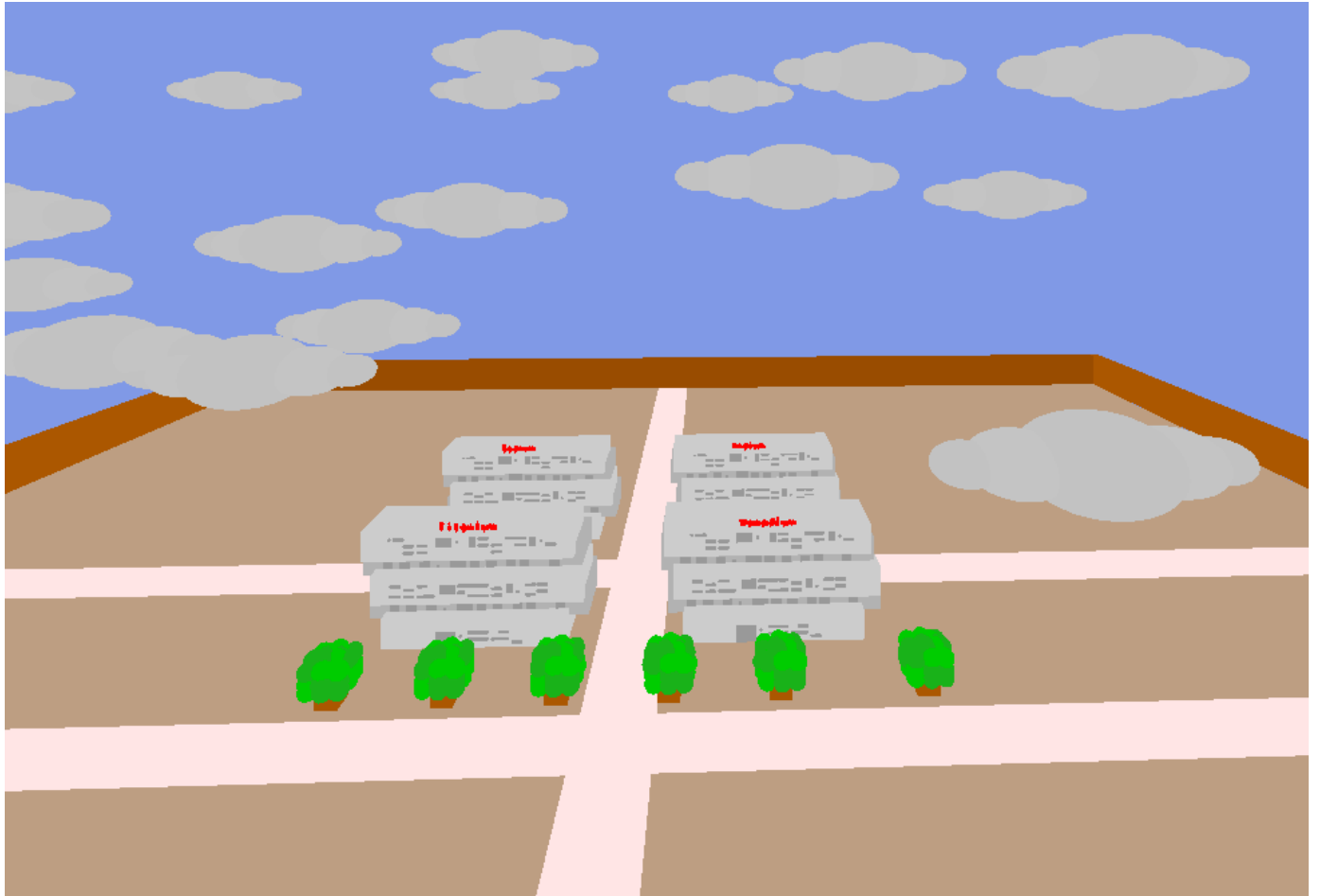
## 2. Left Rotation of Camera using key 'Q/q'



### 3. Right Rotation of Camera using key 'E\e'



#### 4. Moving Upward of Camera using key 'G\g'





5. After on the light main screen

