City and Village

Submitted By

Student Name	Student ID
Jafor Sadak	212-15-4159

MINI LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE422: Computer Graphics Lab in the Computer Science and Engineering Department**

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

April 11, 2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Deawan Rakin Ahamed Remal**,**Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Deawan Rakin Ahamed Remal

Lecturer

Department of Computer Science and Engineering Daffodil International University

Submitted by

Jafor Sadak
ID:212-15-4159
Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

Table 1: Course Outcome Statements

CO's	Statements
CO1	**Understand** computer graphics system and implement graphics primitives for drawing a graphics scene.
CO2	**Apply** appropriate OpenGL programming techniques, resources and modern engineering and IT tools to solve graphics programming issues including different shapes, 2D and 3D transformation
CO3	**Perform** effectively as an individual or a member or a leader of diverse teams through proper documentation and initialization of project work
CO4	**Create** a project by explaining complex computer engineering activities with the computer engineering community by performing effective communication through effective reports, design documentation, make effective presentations and give and receive clear instructions.
CO5	**Understand** computer graphics system and implement graphics primitives for drawing a graphics scene.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2,	K1-K4	EP1
CO2	PO5	C3, C4	K1-K4, K6	EP1 ,EP3
CO3	PO9	C4, A2	K6	EP1 ,EP3
CO4	PO10	C6, P3, A2	K4	EP3 ,EP5

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

Table of Contents

GitHub Project Link :
https://github.com/jaforsadakdiu/Village-and-city-OpenGl-project.git

Chapter 1

Introduction

1.1 Introduction

We use OpenGL to create a virtual village and city scenery. Learning 2D rendering helps understand the basic graphical drawing [1]. We develop a realistic 2D environment in opengl. OpenGL provides machine-based pipeline. Each function modifies the globally to control how objects are drawn [2]. Modern OpenGL lets developers use small programs called shaders to make graphics faster [3]. OpenGL gives you lots of control to user [4]. OpenGL's power lies in its low-level control [5]. We can see different shape buildings. We can see different shape streets. We can see the horizon. We see deep forest in the very far sights. We show a rocket moving vertically to upside and downside. We see different village house , ship, and river. The paddy fields, bird shapes, human shapes are shown. It helps us learn about designing. The design architecture and draw complex scenes in OpenGL. We learn how to manage textures, lighting.We learn how to draw the moving objects, the key board moving objects and small integration of drawings. We show day night and season change scenery.

1.2 Motivation

We want to upskill ourselves in graphics design. We want to improve our skills in designing. We want to make realistic 2D environments in OpenGL. By making the project, we learn how to manage complex scenes. We know to make textures and give moving objects. It enhances our understanding of graphics. We improve our programming knowledge , visual design. It will also help us create more dynamic virtual environments.

1.3 Objectives

We develop various shapes for buildings, streets other structures. We make the image 1000 pixels to x axis and 1000 pixels to y axis. we work in the every 100 pixel images. We give building, sun, moon, clods shapes. We gave different shape trees. We give different shapes roads. We gave rivers and bridge. We gave lightings. We draw a well. We make simple human shapes.We have to gather knowledge about different components and draw different color shades. We use different color and we make the project dynamic. We gives us basic knowledge about computer graphics designing.

1.4 Feasibility Study

OpenGL provides all the necessary tools and libraries to create the project. We have access to resources. It allows us to design various components buildings, roads, trees, and moving objects. The project is manageable. We can see the scope. It is suitable for what we want to achieve. We can basic programming techniques to handle things. It is well supported in OpenGL. It will help us develop foundational skills in computer programming.

1.5 Gap Analysis

We need to improve our knowledge of advanced lighting and shading techniques. We need to make environment more realistic. It is for the day night cycle. The seasonal changes has also limitations. We could create more smooth structures. We could integrate miore user input. Adding 3d animations and objects gives more dimension [6]. The color shades and color combinations should have chosen more accurately.

1.6 Project Outcome

We see the clouds are moving. The different clouds are moving at different speed. The sun in the daytime with a keyboard controlled aeroplane moving . And a rocket is also keyboard controlled. We see deep forest at the very far place. Mountains are seen. We can also see buildings far away. The river is blue there are three ships and they are keyboard controlled. The river is seen in shades with sunlight. The bridge is seen on the river. The train is running in the track. On the other track, the cars are running in the opposite direction. We see the different shapes building as circle , triangles, quads. A well , industrial buildings, cricket ground , football ground, paddy fields, lights , tractors are seen. We can see the lighting along with the roads.

Chapter 2

Proposed Methodology/Architecture

2.1 Requirement Analysis & Design Specification

Using OpenGL to create a realistic 2D environment adds a lots of value with respect in terms of designing [7]. The components buildings, streets, roads, rivers, moving objects are shown. We focus 1000x1000 pixels. We will implement different shapes for buildings, trees, clouds, and roads by polygons, triangle and quads shape. Switch function, loops and glutpush and pop matrix are used to create this. A day-night cycle, seasonal changes need color changes and different function calls.Smooth transitions between various objects are shown like bird flying are also presentend. We uses parametric equations with trigonometric functions. We draw ellipses and applies transformations glTranslatef. The randomColor() function introduces visual variety. It assigns random RGB values. It assigns unique body and roof colors for their index. It enhancs the aesthetics. Nested loops makes a grid of windows to make variation in buildings. The use of glPushMatrix() and glPopMatrix() ensures isolated transformations. Changing position variables and redrawing the scene by glutPostRedisplay(). It gives life to the cars.

2.1.1 Overview

We add buildings, roads, rivers, trees, and moving objects cars, rockets, and ships. We showed day-night changes, seasons. We learned how to draw shapes. We use colors, make things move with the keyboard.

2.1.2 Proposed Methodology/ System Design

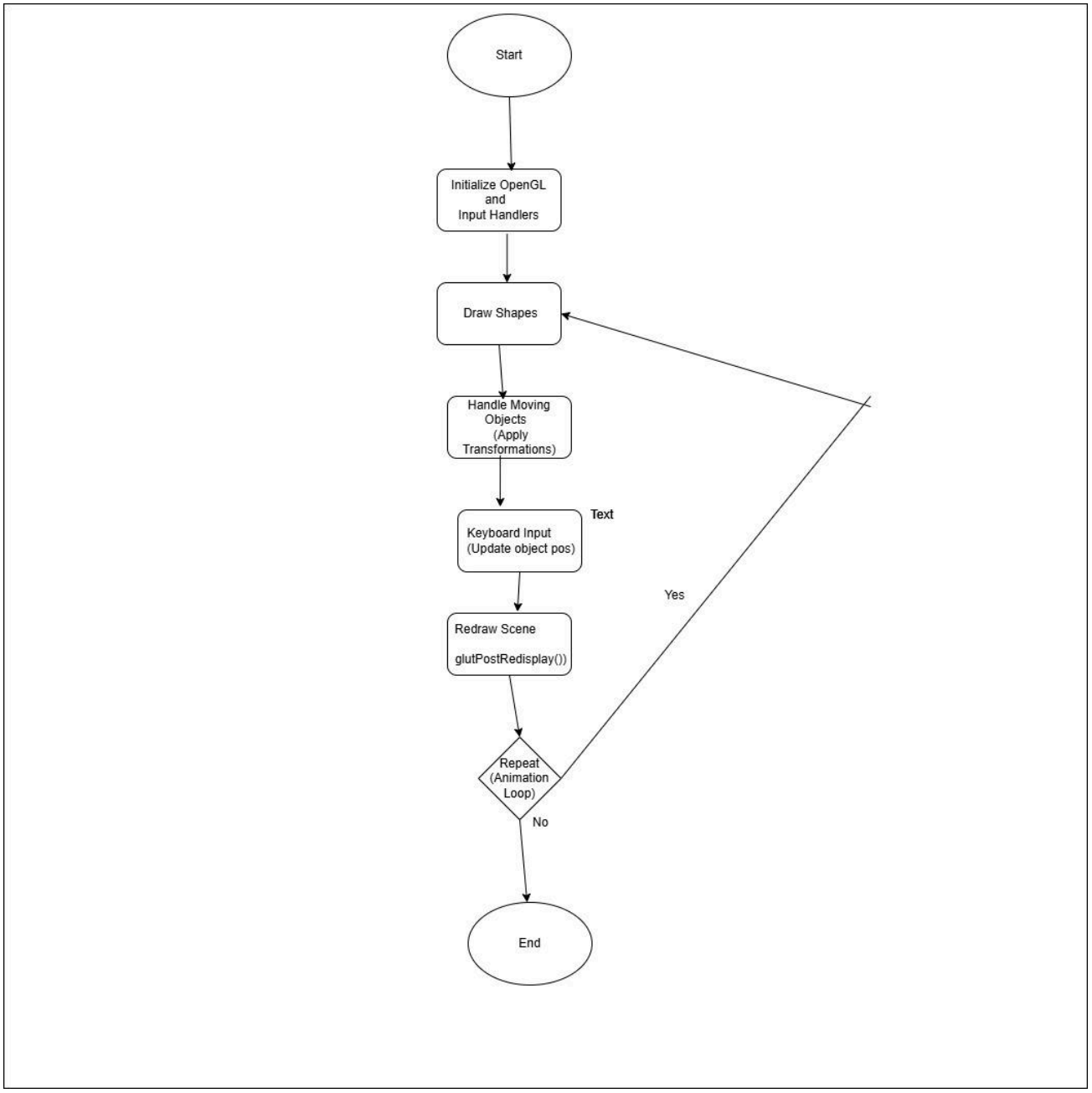

Figure 2.1: Flowchart of algorithms

2.1.3 UI Design

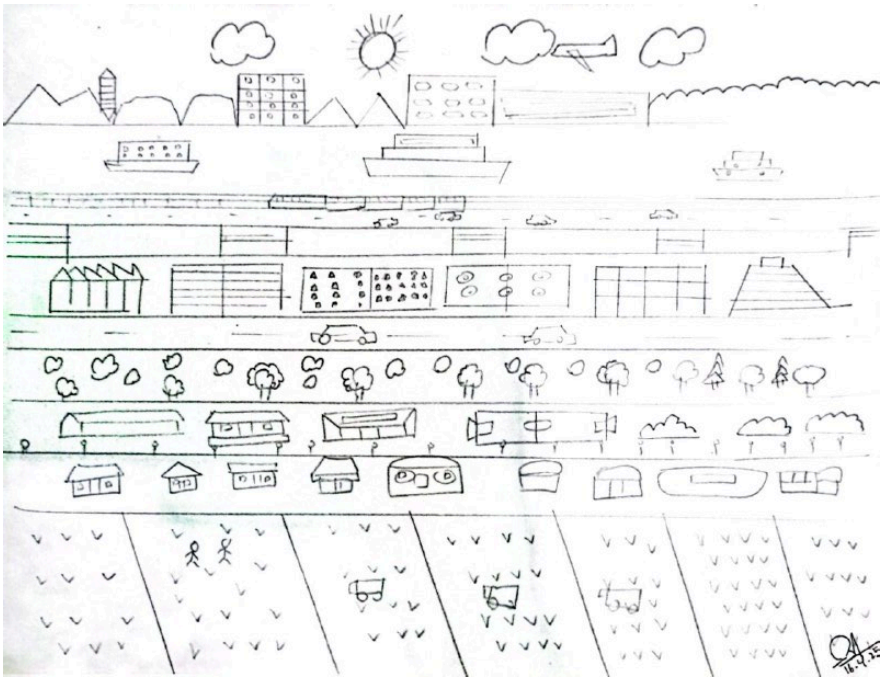

Figure : Rough Plan

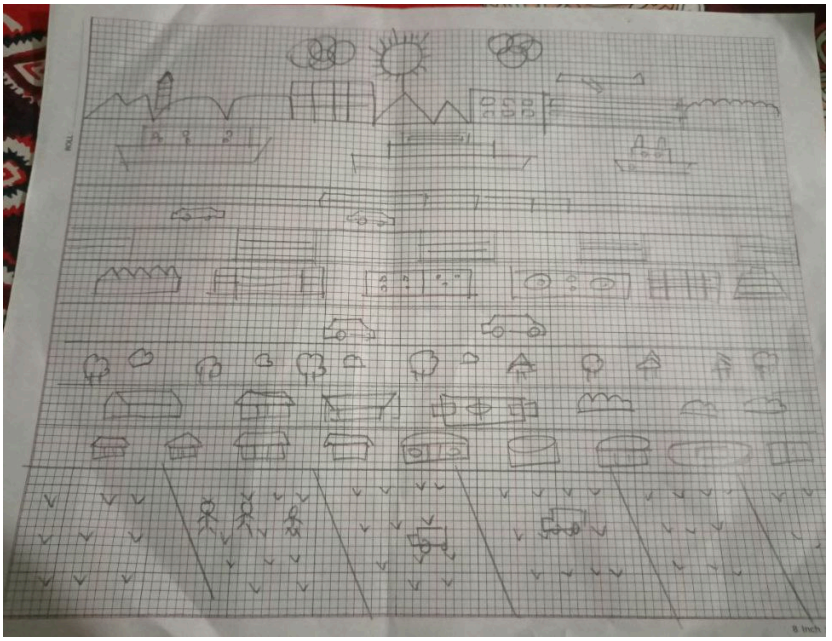

Figure :The graph

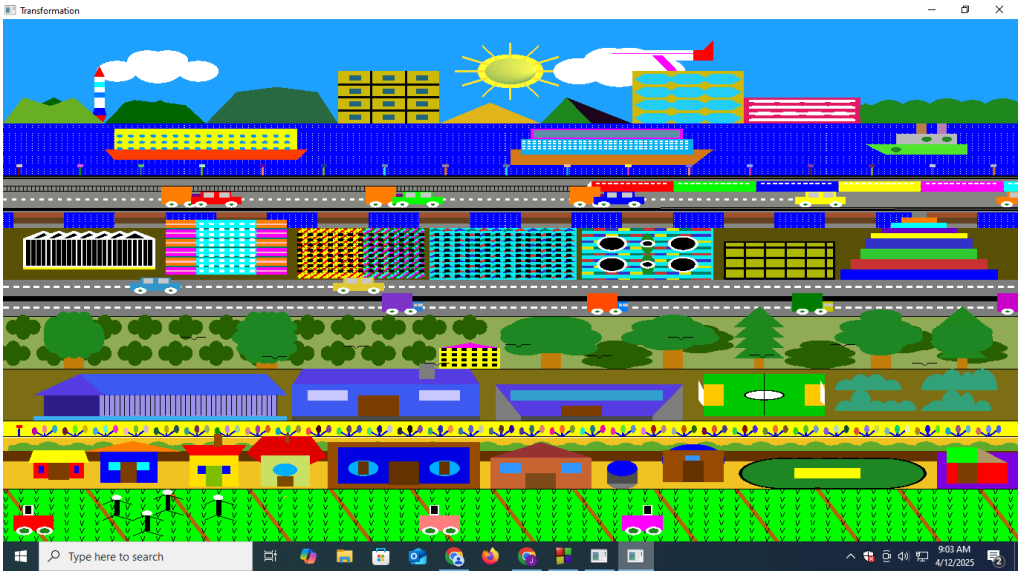

Figure 2.2: Day-time

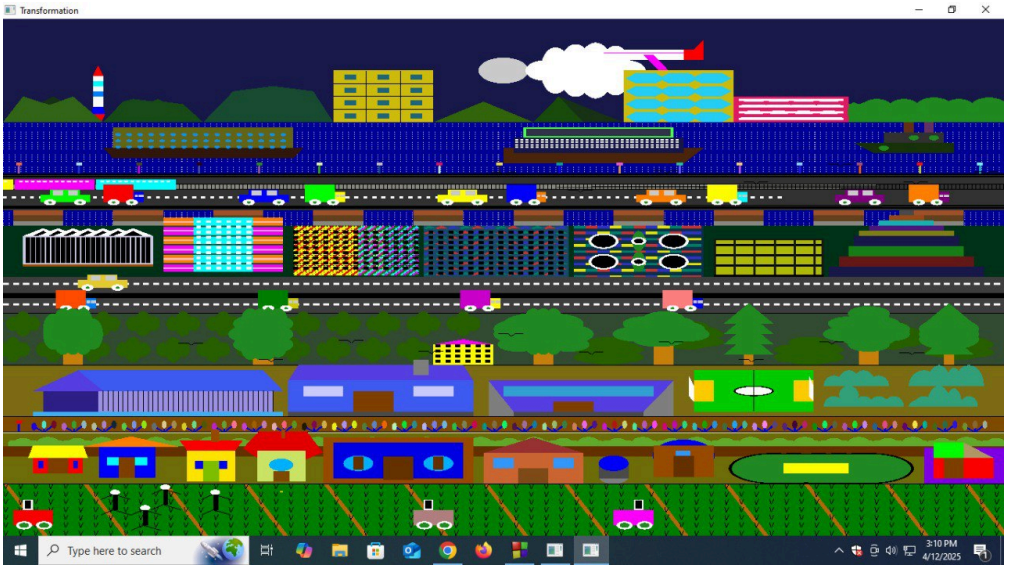

Figure 2.3: Night-time

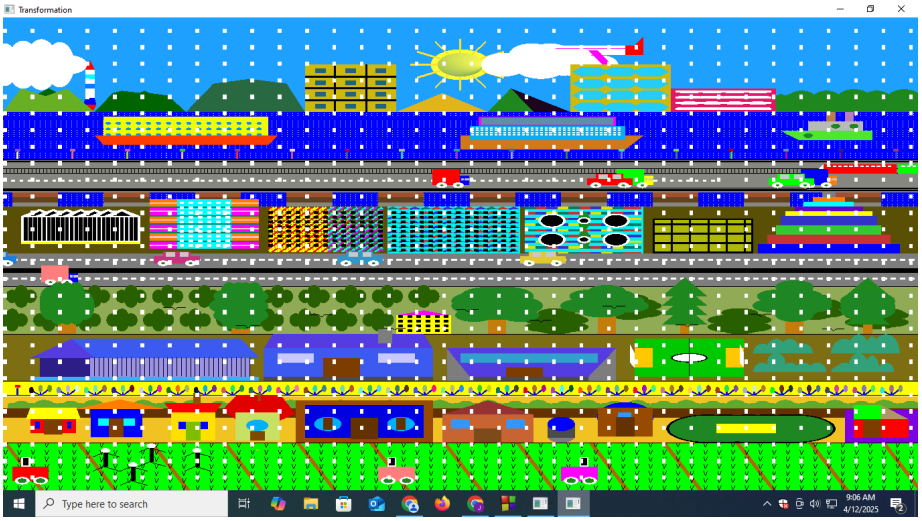

Figure 2.4: Day-time - snow

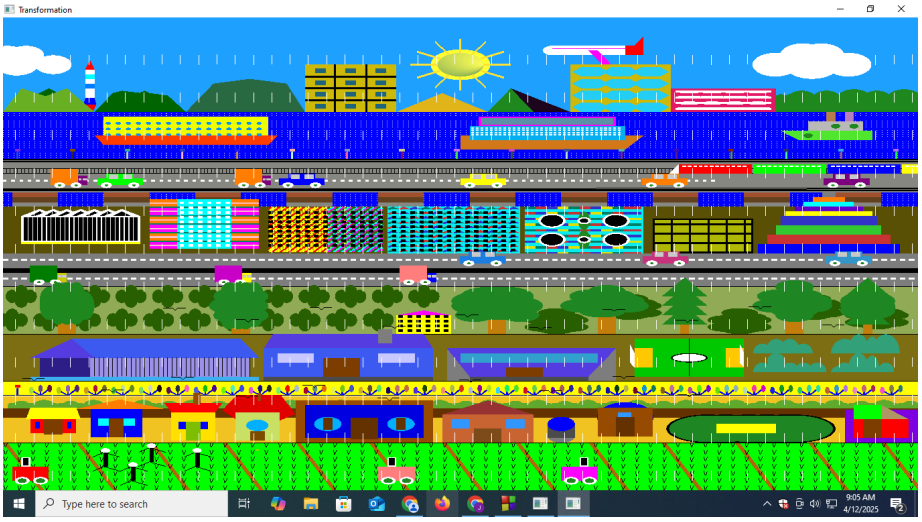

Figure 2.5: Day-time Raining

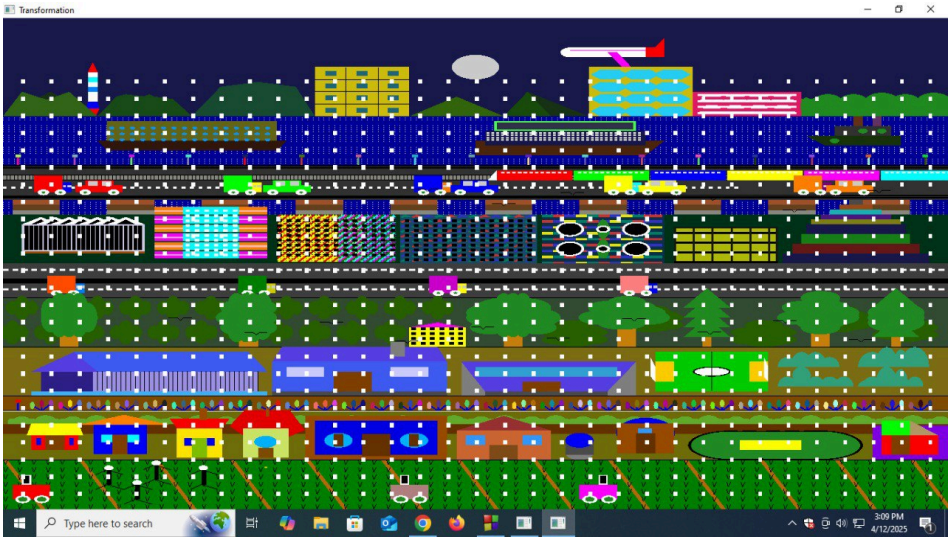

Figure 2.6: Night-time-snow

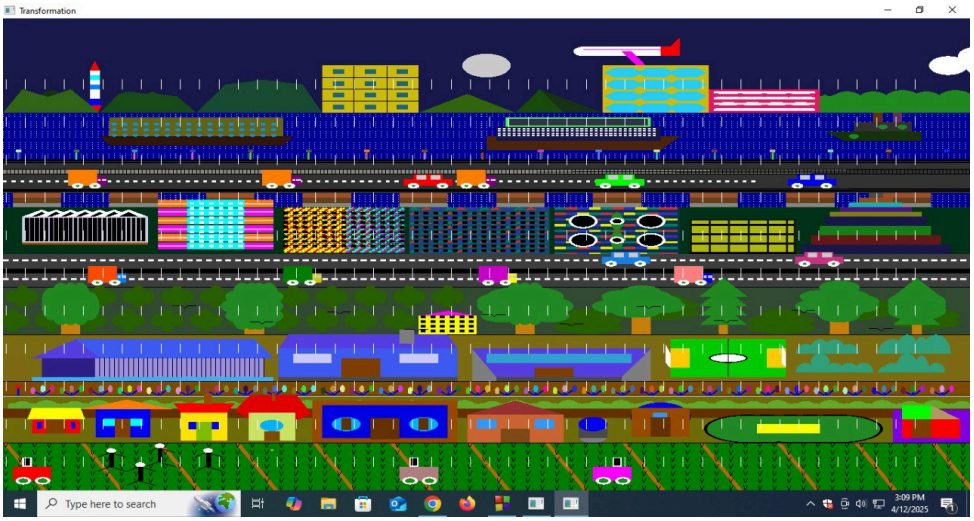

Figure 2.7: Night -time rain

2.2 Overall Project Plan

We made a colorful and animated OpenGL scene. Function draws moving train compartments. Another function draws a large grid of small white rectangles.It resembles buildings, windows. We made multiple decorative polygons. We arrange in neat grids. We use some shaped hexagons, triangles. Several trees are drawn rectangles for trunks. Multiple shape of green circles for leaves. Circles are used repeatedly to form clouds. We made move things across the screen updating their positions. We use glutPostRedisplay() to refresh the frame.

Chapter 3

Implementation and Results

3.1 Implementation

We made blue sky. We use shades to draw sun. We use multiple circle shapes to draw the clouds. We adjust the circles radius ratio to draw the. In first 200 pixels , we made this. Then we draw rivers by quads shape. Then we draw bridge on the river. The bridge has trains and cars. They are also moving. The cars are made with quads and circles. The multiple cars are made horizontally with loops.

We made three different shape ships . We secorated the windows and pools and ships. We made different colored hills with circles. We made green and yellow dessert like mountain. There are rail lines made with gl_Lines. The other part different shape buildings are made with quad and polygon. different sized polygons. We made those buildings with different ratio polygons. Then the roads also begins. The different cars and cargos are plying on the road . They are made with quads and polygons. Then we made a jungle . We can see trees and a cottage. Each tree is different shaped. We see the tree with triangle shaped. We can also see the mills and factories in the picture.

We can see the cricket ground. and footbal; ground made with quads and circle.The we show a village. Different shapes are seen in the village. The paddy fields we can see the crops. A group of people can walk.The different tractors were deployed. We create a new quadric object. We use transformation matrix. We draw circle with center points and radius. We save and restore previous transformation matrix. For moving object, we use glTranslatef() to move them to different axis. We move the cloud and reset if it crosses the screen width.

When the objects move off of the screen, conditions are reset on the screen. glutPostRedisplay() redraws the shapes. Pressing 0,1,2,3,4,5,6,7,8,9,a,w,u,i,j n,d different features. glBegin() makes polygon, quads, triangle, lines.glutInit(&argc, argv) initializes GLUT. It sets the display mode with glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB). The window size is set with glutInitWindowSize(1000, 500). window is created using glutCreateWindow(""). The init() function handles initialization. The glutDisplayFunc(drawScene) function draws the scene. The glutKeyboardFunc(key) function handles keyboard input. It uses

glClearColor(0.0f, 0.0f, 0.0f, 1.0f) to set the color to black. rand() function makes random value. We use shifting property to move aside an object. We make objects vertically and horizontally to present a dynamic scenery. The
objects move automatically or the keyboard controlled.

We draw simple bird shapes using lines. Each shape is made of four black lines. We arrange to draw wings. It moves from right to left across the screen. they reappear on the right after leaving the screen. The effect is controlled by timeElapsed variable. It increases with each frame. The movement and vibration are applied using glTranslatef. glPushMatrix/glPopMatrix makes bird's position is handled separately.

3.2 Performance Analysis

We run the project successfully. The objects are moving with desired positions. The day night comes accurately. And the combination of rain and snowfall is flowing accordingly. The performance of the project is efficient. It effectively manages various moving objects and scenes.The use of transformation matrices and GLUT ensures smooth transitions and real-time updates. Keyboard inputs are seamlessly for controlling objects cars, ships, and planes. The dynamic nature of the environment clouds, ships, vehicles, runs smoothly.

3.3 Results and Discussion

We perfectly implemented dynamic 2D virtual environment.Various moving objects, cars, ships, planes were controlled by keyboard. The day-night cycle and seasonal changes were displayed.We successfully show transitions between different environments rain and snowfall. Buildings, trees, rivers, and roads created a realistic

village and city scenery. It meets its objectives providing engaging virtual environment. It offers valuable insights into graphics design and programming techniques.

Chapter 4

Engineering Standards and Mapping

4.1 Impact on Society, Environment and Sustainability

It has a positive impact on society. It enhancs the understanding of graphics design. Computer programming helps improve skills. It is related to creating dynamic, interactive environments. It can be used in various fields. Gaming, education, and urban planning is those sectors.. The environment portrayed in the project is realistic. It can raise awareness about environmental issues. Deforestation or urbanization are main problem nowadays.We felt we need to show a demonstration to make it look good.Besides, we can send a meesage to society that we are able to do such environmental planning that can make a better place to live in.

4.1.1 Impact on Life

It helps individuals to develop practical programming skills. It also enhances creativity. Users can visualize complex 2D environments. It is possible to manipulate virtual spaces. It can be used to raise awareness. It balances between urban development and nature. By designing realistic environments, it can ensure sustainable city planning. It says importance of natural landscapes. Technology can help us understand environmental changes.

4.1.2 Ethical Aspects

It encourages the responsible use of resources. It aims to create awareness around environmental conservation. It also emphasizes the importance of respecting nature.It opens up new opportunities for education and entertainment. It helps individuals to experience different environments and scenarios.

4.1.3 Sustainability Plan

It contributes to sustainability. Users can create virtual environments. It does not rely on physical resources. It can reduce the physical models. Future developments could focus on environmental aspects. It needs prototyping making complex project.

4.2 Project Management and Team Work

The success of the project depends on effective teamwork [1]. It needs clear communication. It needs proper task allocation. The group can tackle complex challenges. Regular meetings and progress updates ensure the progress. The team work makes successful implementation of this project.

Complex Engineering Problem

It solves complex engineering problem. Interactive and dynamic virtual environment is complex task.It can manage multiple moving objects, transformations. It requires knowledge of computer graphics, programming. Resource management ensures smooth performance.

4.2.1 Mapping of Program Outcome

It demonstrates proficiency in OpenGL. The project helps build teamwork, communication, and project management skills. It is essential in real-world engineering and software development careers.

Table 4.1: Justification of Program Outcomes

PO's	**Justification**
PO1	We can demonstrate opengl and draw shapes. We can set the environment. Knowledge gain by utilizing different shape structures. We add different color combination drawing it.
PO2	We draw line drawing and other objects. We use lines, quads , triangles to draw different shape. We also implement circle to draw real life objects 2d , 3d transformation.
PO3	We works as team. We collaborate on documentation and initialization of project works.
PO4	We solved complex engineering problem by shifting different structures of building and house. The paddy fields and other staffs as road line were drew using it.We are slo designing documentation and presentation.

4.2.2 Complex Problem Solving

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	**EP2** Range of Conflicting Requirements	**EP3** Depth of Analysis	**EP4** Familiarity of Issues	**EP5** Extent of Applicable Codes	**EP6** Extent Of Stakeholder Involvement	**EP7** Inter-dependence
We gather knowledge of graphics design component. The drawing functions. And art alignment and different objects that are portrayed on 2d.	Conflicting requirements were limited. We although find complexity to make day-night sequence. The night sequence and session sequence made that could be conflicting by declaration.	We do lot analysis drawing them. Cause the space management. The space complexity and objects complexity made us analyze depth.	We found familiar problems. Cause most of they are a shape that is in a quads or triangle or circle. These makes familiarity issues.	Codes need to be reused. Wer reuse the code of circle. We made different architect using different shape . It was made with changing functions.	Strong involvement. We make sure oir stakeholders give us opinion on the project . We built it according to the system.	The connection is light. The functions are interconnected. To change something we need detail involvement of changing the conflicted part.

4.2.3 Engineering Activities

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
We are using windows 11 operating system, We need code blocks and glut tools of opengl.We had to Install the gnu gcc compiler.	The interaction was very strong with activism. Each function is connected to others. The glutinitdisplaymode, glutfunction, keyboard controls and the vertical movements and horizontal movements ext are connected to each other.	We explore things like moving objects and keyboard control objects . The color combination things and objects coordination things.	The consequences were the awareness of environment. A virtual system that can show us a design for a planned and environment friendly eco-system.	The familiarity between ground knowledge and implementation was important. We reuse some of the particular codes of circles and pattern based codes.

Chapter 5

Conclusion

5.1 Summary

We create a realistic 2D virtual village and city scenery. It involved designing various components s buildings, streets, rivers, trees, moving objects cars, planes, and ships. It has dynamic features day-night cycle, seasonal changes. We make interactive elements by keyboard. The goal was to enhance graphics design skills. We improve programming knowledge. We develop a deeper understanding of computer graphics. Moving objects, transformations, and environmental changes makes the difference.

5.2 Limitation

There are several limitations. The environment remains in 2D. We do not make 3D objects. It could enchance the depth . The day-night and seasonal changes could be more beautiful. The use of lighting and shading techniques are limited. It limits the realism.Advanced techniques could be explored . It gives better visual effects. It relies on keyboard input for movement. It might restrict user interaction options.

5.3 Future Work

we will include 3D environments. It would add more realism to the scenes.We wiil integrate more user interaction methods, The mouse controls , touch inputs will be added. It could make the simulation more engaging. Characters moving , more complex weather effects could also be applied to the project. Performance optimization techniques could be applied to smooth performance

References

1. Shreiner, D., Sellers, G., Kessenich, J. & Licea-Kane, B., 2009. *OpenGL programming guide: The official guide to learning OpenGL, version 1.1*. 7th ed. Boston: Addison-Wesley.

2. Sellers, G., Wright, R. & Haemel, N., 2016. *OpenGL SuperBible: Comprehensive tutorial and reference*. 7th ed. Boston: Addison-Wesley Professional.

3.Benstead, L. & Lee, J., 2011. *Beginning OpenGL game programming*. 2nd ed. Boston: Cengage Learning.

4. Angel, E. & Shreiner, D., 2014. *Interactive computer graphics: A top-down approach with WebGL*. 7th ed. Boston: Pearson.

5. Wright, R.S., 2012. *OpenGL development cookbook*. Birmingham: Packt Publishing.

6. Rost, R.J., Licea-Kane, B., Ginsburg, D., Kessenich, J., Sellers, G., Shreiner, D. & Lipchak, N., 2009. *OpenGL shading language*. 3rd ed. Boston: Addison-Wesley.

7. Götz, D., 2016. *Learn OpenGL: Beginner's guide*. Self-published. Available at: https://learnopengl.com [Accessed 12 Apr. 2025].

8. McShaffry, M. & Graham, D., 2013. *Game coding complete*. 4th ed. Boston: Cengage Learning.