American International University- Bangladesh (AIUB)
Faculty of Science and Technology (FST)

Course Name:	**Computer Graphics**	**Course Code:**	**CSC4118**
Semester:	**Fall 2025-2026**	**Section:**	**J**
Faculty Name:	**Dipta Justin Gomes**		

Project Title:	**Dhaka City**
Group:	**C**

SUBMITTED BY

Student Name	Student ID	Department	Contribution
Chowdhury Maheen Shahriyar	22-48381-3	CSE	Scene-1: Dhaka Metro Rail
MD. Fardin Rahman Mridul	22-49521-3	CSE	Scene 3: Bangladesh National Museum
Nazmul Hossain Khan	22-49520-3	CSE	Scene-2: Shaheed Minar

INTRODUCTION

This project presents a 2D graphical visualization of three iconic landmarks of Dhaka City: the Dhaka Metro Rail, Shaheed Minar, and the Bangladesh National Museum. Each scene reflects a unique aspect of the city, combining modern infrastructure, national heritage, and cultural significance through interactive computer graphics.

Scene 1: Dhaka Metro Rail

Scene 1 visualizes a modern urban environment centered on the Dhaka Metro Rail system. The scene depicts roads, moving cars, a running metro train, buildings, traffic signals, street lights, trees, and dynamic environmental elements such as clouds, rain, snow, lightning, rainbow, and day–evening–night transitions. The animation reflects real-life city movement and traffic behavior.

The motivation behind this scene is to represent Dhaka's rapid urban development and the importance of the metro rail in reducing traffic congestion and improving public transportation. As Dhaka is one of the most densely populated cities in the world, visualizing its evolving infrastructure helps demonstrate how technology contributes to smarter urban living.

The novelty of this scene lies in its interactivity and environmental realism. Multiple weather conditions, time transitions, traffic signal logic, vehicle motion, and user-controlled lighting are integrated into a single cohesive scene. These features create a lively and realistic simulation rather than a static graphical display.

The background of the scenario is inspired by real urban roads of Dhaka, where metro rail lines run alongside busy streets under varying weather and lighting conditions. This reflects everyday city life and enhances visual authenticity.

The platform used for this scene is OpenGL with GLUT in a 2D orthographic projection. OpenGL was chosen for its efficiency in rendering graphics, handling animations, and supporting real-time interaction, making it suitable for educational visualization projects.

Scene 2: Shaheed Minar

This scene represents the Shaheed Minar, one of the most important national monuments of Bangladesh. The Shaheed Minar symbolizes the sacrifice of the martyrs of the Language Movement of 1952, who gave their lives to protect the Bangla language. In this scene, the monument is placed at the center with proper symmetry, surrounding pathways, trees, bushes, lamp posts, people, and the national flag. The environment changes smoothly from day to night, making the scene realistic and meaningful.

The motivation of this scene is to visually present national respect and remembrance using computer graphics. Instead of showing the Shaheed Minar as a static image, this project brings it to life through animation. People stand in front of the monument during the day as a sign of respect, while at night the lamps glow and stars appear, creating a calm atmosphere

The novelty of this scene lies in its smooth day–night transition and realistic environmental behavior. The sun sets completely before the moon rises, stars appear gradually, clouds move slowly, the flag waves gently, and people disappear at night. These features make the scene visually appealing while keeping the design simple.

The background of this scenario is inspired by the real Shaheed Minar area in Dhaka. Bushes, pathways, and lighting are added to match the real surroundings. This scene is developed using OpenGL with GLUT, which is suitable for learning computer graphics because it supports drawing, animation, and interaction.

Scene 3: Bangladesh National Museum

This scene presents a 2D animated graphical simulation of a National Museum environment using OpenGL and GLUT. The scenario represents a real-life outdoor setting where both natural and man-made elements coexist harmoniously. Key components of the scene include the museum building, sky, sun, moon, clouds, birds, trees, vehicles, roads, the national flag, and lamp posts. The environment dynamically changes over time, displaying smooth transitions between day, sunset, night, and sunrise. Each object in the scene is animated to create a lively, visually appealing, and realistic atmosphere that reflects real-world behavior.

The main motivation behind this scene is to understand and apply fundamental computer graphics concepts through hands-on implementation. Rather than producing static drawings, this scene emphasizes animation and interaction, allowing better visualization of motion, transformation, and time-based changes. It also aims to strengthen practical knowledge of OpenGL primitives, transformations, animation loops, and structured programming techniques.

The novelty of the scene lies in the integration of multiple animated elements within a single coherent scene. Features such as a gradual day–night cycle, slowly moving clouds, twinkling stars using alpha blending, a waving national flag, flocking birds, and moving vehicles contribute to enhanced realism. Additionally, the crescent-shaped tilted moon and smooth sky color transitions further enrich the visual experience.

The National Museum is selected as the central structure to symbolize a cultural and educational landmark. Surrounding features such as roads, greenery, and public infrastructure are included to simulate a realistic urban environment.

OpenGL with GLUT is used as the primary graphics platform due to its simplicity and efficiency in rendering 2D graphics, handling transformations, animations, and user interaction. GeoGebra Classic was also used to model and plan the overall scene layout accurately.

RELATED WORKS

This section reviews 10 existing projects and research works related to computer graphics–based city visualization, traffic simulation, and environmental animation using OpenGL and similar technologies. These works provide insight into how graphical scenes, object movement, and visual effects can be implemented efficiently and help position the Dhaka City project within the broader context of graphical simulation systems.

1. Mohakhali Bus Terminal:

URL: https://github.com/ahsan-habib7/Computer-Graphics-Project-Mohakhali-Bus-terminal-

This project simulates the *Mohakhali Bus Terminal* — one of the busiest transport hubs in Dhaka — using 2D OpenGL/GLUT. It features multiple bus counters with signboards of popular transport services and buses parked at their respective counters. The scene includes dynamic vehicles such as buses, trucks, cars, CNGs, and animated motion with wheel rotation, providing a realistic effect. Additional animated elements like trains on railway tracks and an airplane in the sky make the visualization comprehensive. The project demonstrates urban transportation simulation with multiple moving objects and layered scene rendering, conceptually similar to the *Dhaka City* project's focus on motion and animation in a city context.

2. City Traffic Simulation:

URL: https://github.com/elkaRD/CityTrafficSimulation

This project implements a city traffic simulation using C++ and OpenGL. It focuses on rendering road networks, intersections, and moving vehicles in a virtual city environment. The simulation supports camera navigation and dynamic vehicle movement, allowing users to observe traffic flow in real time. The work is closely related to the Dhaka City project as it demonstrates how OpenGL primitives and transformation functions can be used to animate vehicles and simulate real-world traffic behavior. The idea of visualizing traffic flow in a city context directly aligns with the metro rail and road-based animation concepts used in the Dhaka City scenes.

3. Moving Car:

URL: https://github.com/khaledbnmohamed/Moving-car-openGL

This project presents a simple 2D car movement simulation using OpenGL and GLUT. It includes animated vehicles moving along a road and basic interaction through keyboard controls. Although the visual complexity is lower, the project demonstrates core animation principles such as translation, timing control, and continuous rendering. These principles are fundamental to the Dhaka City project, particularly for animating metro trains, vehicles, and other moving objects within different scenes.

4. Road Traffic Simulation:

URL: https://github.com/Ralweena/Road_Traffic_Simulation

This OpenGL-based traffic simulation focuses on vehicle movement, road layout, and traffic signal control in a 2D environment. Vehicles respond to traffic lights and follow predefined paths, mimicking real-world traffic behavior. This work is relevant to the Dhaka City project because it illustrates structured scene design and object interaction, which are essential when representing busy urban environments like Dhaka with realistic motion and flow.

5. Car Lanes Project:

URL: https://github.com/muhammed-emad621/OpenGL-GLUT-car-lanes-project

This project simulates multiple car lanes using OpenGL and GLUT, including animated vehicles and environmental changes such as day and night modes. The project highlights scene control, color changes, and animation loops. Similar techniques are applied in the Dhaka City project to manage scene transitions and environmental effects, making this work a useful reference for understanding scene-based rendering.

6. Snow Simulation:

URL: https://github.com/lscholte/SnowSimulation

This project focuses on real-time snow simulation using particle systems in OpenGL. It demonstrates how environmental effects such as snowfall can be created using repeated rendering of small particles with random motion. This work is relevant to the Dhaka City project's weather-based visual effects, as similar techniques are used to simulate rain or other atmospheric elements to enhance realism.

7. Rain Simulation:

URL: https://github.com/Daxelman/Rain

This OpenGL project simulates rainfall using animated particles and blending techniques. The work emphasizes visual realism and smooth animation through continuous updates using rendering loops. The Dhaka City project applies similar concepts for weather effects, showing how particle systems can significantly improve the visual quality of a city scene.

8. Particle Fountain Simulation:

URL: https://gist.github.com/Cyndir/712d25f3610af1b8c0b2

This particle fountain simulation demonstrates how particle systems can be used to generate continuous visual effects in OpenGL. The project focuses on object spawning, velocity control, and rendering optimization. Such techniques are directly applicable to weather effects like rain and snow in the Dhaka City project, where large numbers of small animated elements must be rendered efficiently.

9. RainGL – Rain Simulation:

URL: https://github.com/sanketsudake/RainGL

RainGL is an advanced rain simulation project that uses OpenGL to render realistic rain effects. It highlights the use of shaders, animation loops, and performance optimization. While more advanced, it provides insight into how realistic environmental effects can be achieved in graphical applications. The Dhaka City project follows a simplified version of these ideas to maintain performance while improving visual realism.

10. Highway Scene:

URL: https://github.com/Alimul-Mahfuz/CG-Project

This repository implements a 2D graphics simulation of a highway scene using OpenGL and C++. The project features a highway environment including day, night, and rainy weather conditions, moving objects, and visual elements like buildings and natural scenery. It demonstrates the use of OpenGL for day/night transitions, object animation, and simple environmental effects — concepts that are closely

related to the *Dhaka City* project's implementation of multiple scenes with animated elements and weather modes.

APA References:

1. ahsan-habib7. (n.d.). *Computer Graphics Project – Mohakhali Bus Terminal* [Source code]. GitHub.
2. elkaRD. (n.d.). *CityTrafficSimulation* [Source code]. GitHub.
3. Khaledbnmohamed. (n.d.). *Moving-car-openGL* [Source code]. GitHub.
4. Ralweena. (n.d.). *Road Traffic Simulation* [Source code]. GitHub.
5. Ttusk. (n.d.). *OpenGL GLUT car lanes project* [Source code]. GitHub.
6. Scholte, L. (n.d.). *SnowSimulation* [Source code]. GitHub.
7. Daxelman. (n.d.). *Rain* [Source code]. GitHub.
8. Anonymous. (n.d.). *Particle fountain simulation using OpenGL* [Source code]. GitHub Gist.
9. Sudake, S. (n.d.). *RainGL* [Source code]. GitHub.
10. Alimul-Mahfuz. (n.d.). *CG-Project: Country Highway Scene (OpenGL)* [Source code]. GitHub.

IMPLEMENTATION

The implementation of the Dhaka City project is divided into three independent OpenGL scenes, each representing a significant landmark of Dhaka. Every scene is implemented using 2D orthographic projection, modular drawing functions, animation logic, and user interaction. The scenes are merged into a single program using a scene-switching mechanism while maintaining independent logic and behavior.

Scene 1: Dhaka Metro Rail

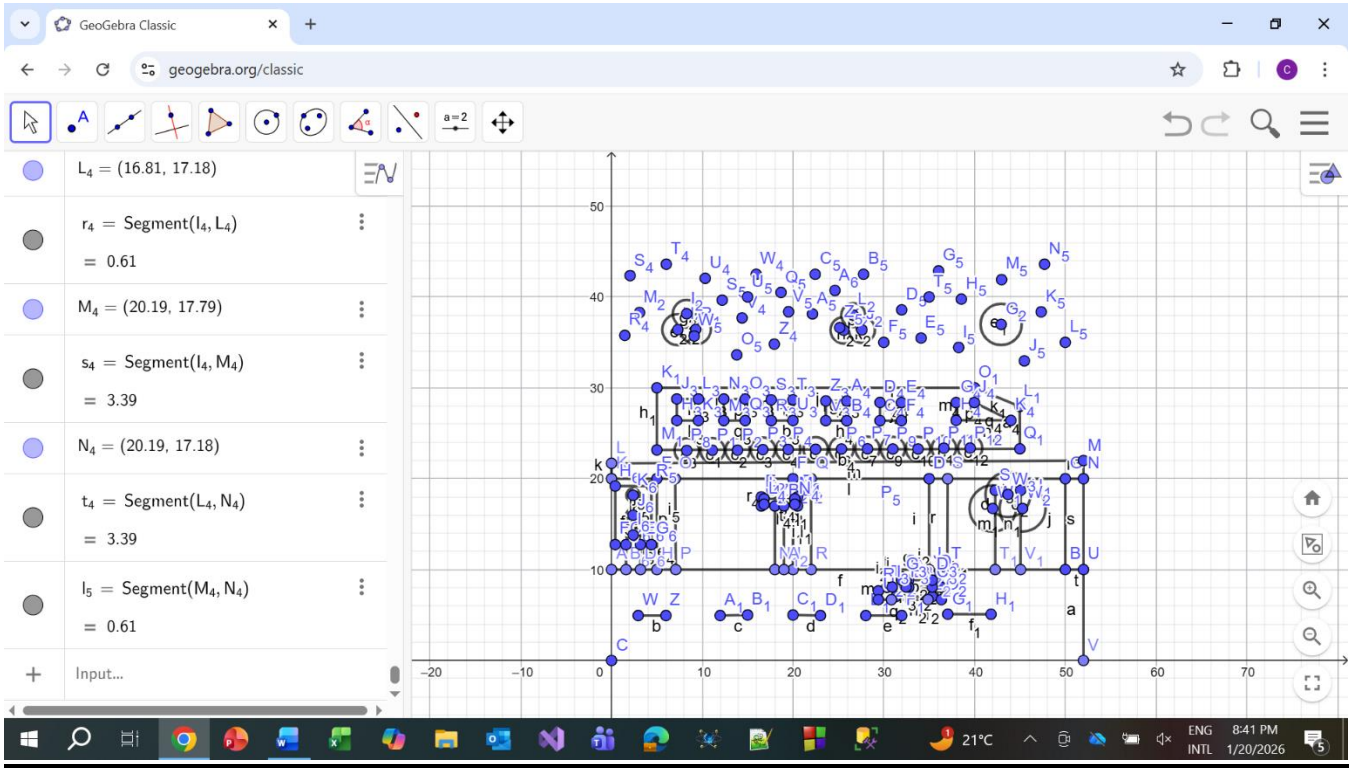
Key Features:
Scene 1 includes a fully animated urban environment focused on the Dhaka Metro Rail. The key features include moving metro trains, multiple cars with wheel rotation, traffic signals with timing logic, street lights with automatic and manual control, and dynamic environmental conditions. The scene supports day, evening, and night modes, along with weather effects such as rain, snow, lightning, and rainbow. User interaction is enabled through keyboard and mouse inputs to control weather, lighting, traffic behavior, and scene transitions. Real-time animation creates a realistic city atmosphere.

Key Technologies Used:
The scene is implemented using OpenGL with the GLUT (OpenGL Utility Toolkit) library in C++. A 2D orthographic projection (gluOrtho2D) is used to maintain a flat yet visually rich city layout. OpenGL's immediate mode (glBegin, glEnd) is used for drawing shapes, while blending (glEnable(GL_BLEND)) is applied for transparency effects such as fog, rain, lightning flash, and light glow. GLUT callback functions handle display rendering, idle-based animation, keyboard input, mouse interaction, and timers for smooth motion.

List of Key Functions:

- scene1() – Renders the complete metro rail scene
- idleScene1() – Updates animations such as cars, clouds, weather, and traffic
- drawCar(), drawCar2(), drawCar3() – Draw moving vehicles
- updateCar(), updateCar2(), updateCar3() – Control vehicle motion and speed
- metroRail(), metroWheels(), metroWindows() – Draw metro components
- drawRain(), drawSnow(), updateRain(), updateSnow() – Weather simulation
- drawRoadLight(), drawRoadLight2() – Street lighting system
- trafficSignal() – Traffic signal logic and rendering
- handleKeyScene1() – Keyboard interaction
- handleMouseScene1() – Mouse-based lighting control

Screenshot of GeoGebra Design:



Scene 2: Shaheed Minar

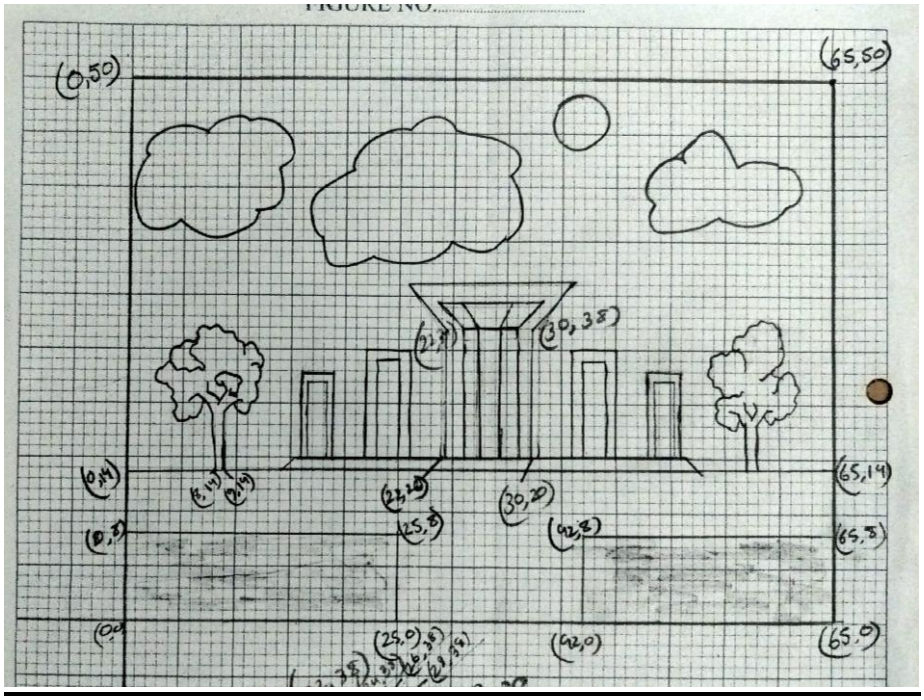
The implementation of Scene 2 focuses on creating a respectful and animated visualization of the Shaheed Minar using simple OpenGL techniques. The scene uses a 2D orthographic projection, and all

objects are drawn using basic shapes such as rectangles, circles, lines, and polygons.
A key feature of this scene is the day-to-night animation system. A variable controls the time of day, allowing the sun to slowly set and the moon to rise afterward. Stars appear only after the moon reaches halfway, creating a natural night effect.
Environmental animation is another important feature. Clouds move slowly across the sky, birds fly only during daytime, and the Bangladesh flag waves gently using a sine wave. Lamp posts automatically glow at night to enhance realism.
The Shaheed Minar structure is built using multiple rectangular pillars. The center structure contains three tall pillars connected by a bridge and a trapezoid top, while medium and short pillars are placed symmetrically on both sides. A red circular symbol is added to represent the monument's symbolic element.
People are drawn using simple line figures. They appear only during the daytime and stand respectfully in front of the monument and on the pathways. At night, the people disappear, showing silence and respect. Bushes and pathways are added to improve depth and balance.
OpenGL with GLUT is used to handle drawing, animation, and keyboard interaction. The idle function updates movement, and keyboard keys control sunset and sunrise animation.

Key Functions:

scene2() – Draws the complete Shaheed Minar scene
idleScene2() – Updates all animations
drawSunMoon() – Controls sun and moon movement
drawStars() – Displays stars at night
drawBangladeshFlag() – Draws the waving national flag
drawPerson() – Draws people figures
drawBush() – Draws bushes around the monument
handleKeyScene2() – Handles keyboard input

Image of Design:



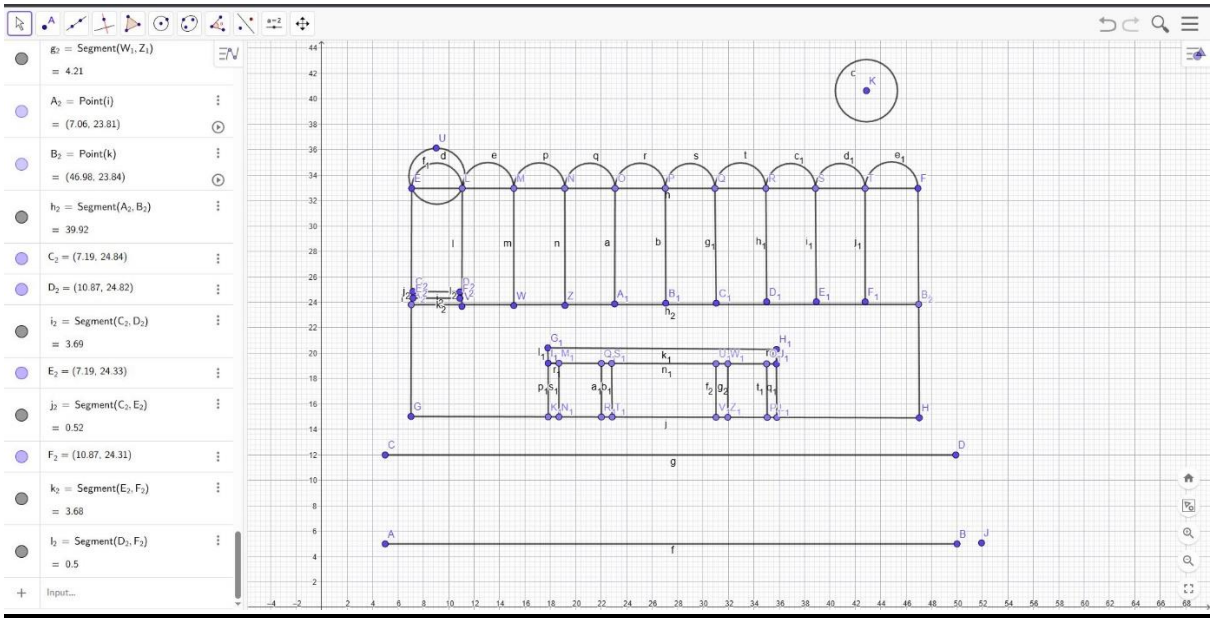
Scene 3: Bangladesh National Museum

The project is implemented as a 2D animated scene using OpenGL and GLUT, focusing on realism through motion and time-based transitions. One of the main features is the day–night cycle, which smoothly transitions between day, sunset, night, and sunrise by updating sky colors and object visibility. The sun and moon animations simulate rising and setting using vertical translation, while the moon is designed as a tilted crescent using overlapping circles. Clouds move horizontally at different speeds to create depth, and stars appear only at night with fading and twinkling effects using alpha blending. The scene also includes moving cars on a road, birds flying in a flock, a waving national flag created using sine wave deformation, and lamp posts that light up at night. All objects are modularly designed for better structure and reusability.
The project uses OpenGL as the core graphics library for rendering 2D shapes and animations. GLUT (OpenGL Utility Toolkit) is used to manage the window, handle keyboard input, and control animation through the idle function. OpenGL transformations such as glTranslatef, glRotatef, and glScalef are used to animate objects. Alpha blending is enabled to create smooth fading and twinkling effects for stars. Mathematical functions like sine and cosine are used for circular shapes and wave motions.

List of Key Functions Used in the Project:

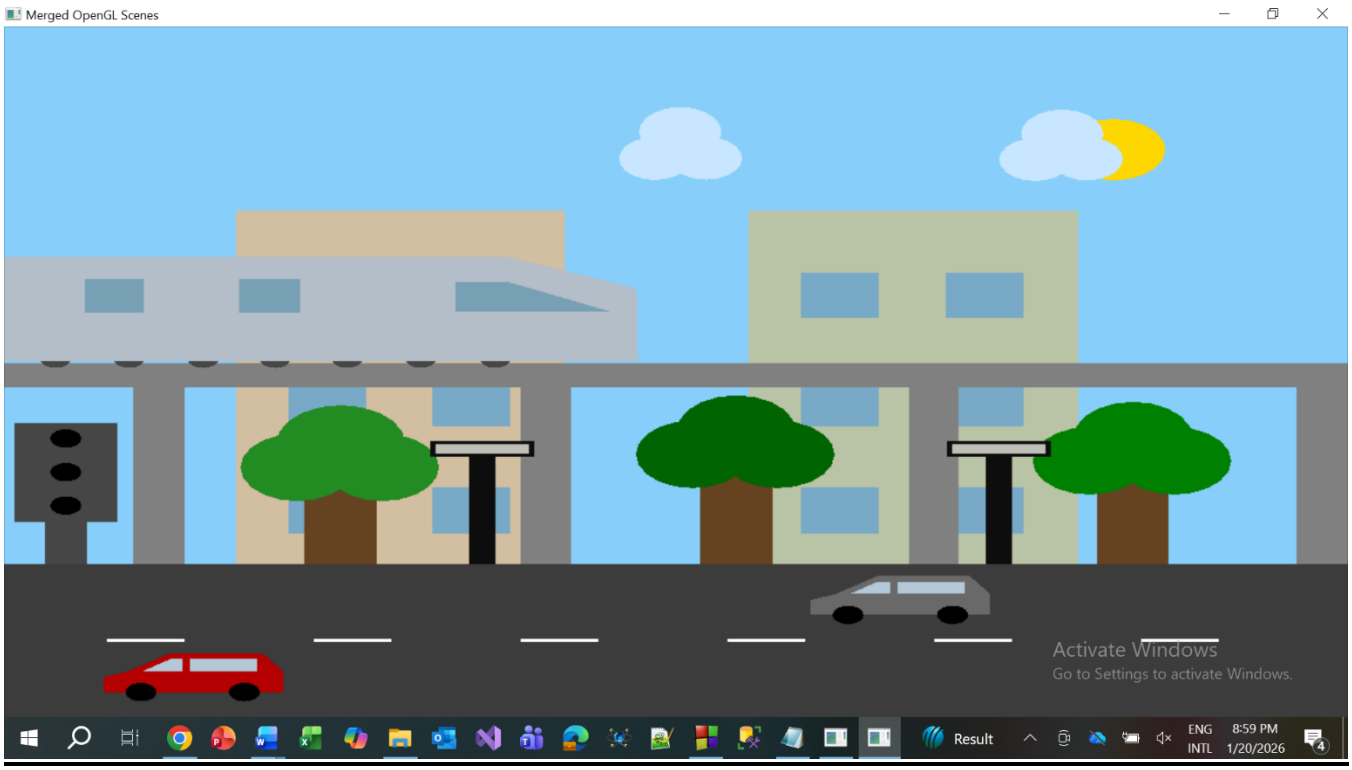
- display() – renders all scene elements
- idle() – controls animation and movement
- sun() and moon() – draw and animate celestial bodies
- cloud1(), cloud2(), cloud3() – animate clouds
- stars() – draw night stars with blending
- bird() and drawBird() – animate birds
- drawBangladeshFlag() – waving flag animation
- handleKeypress() – manages user interaction

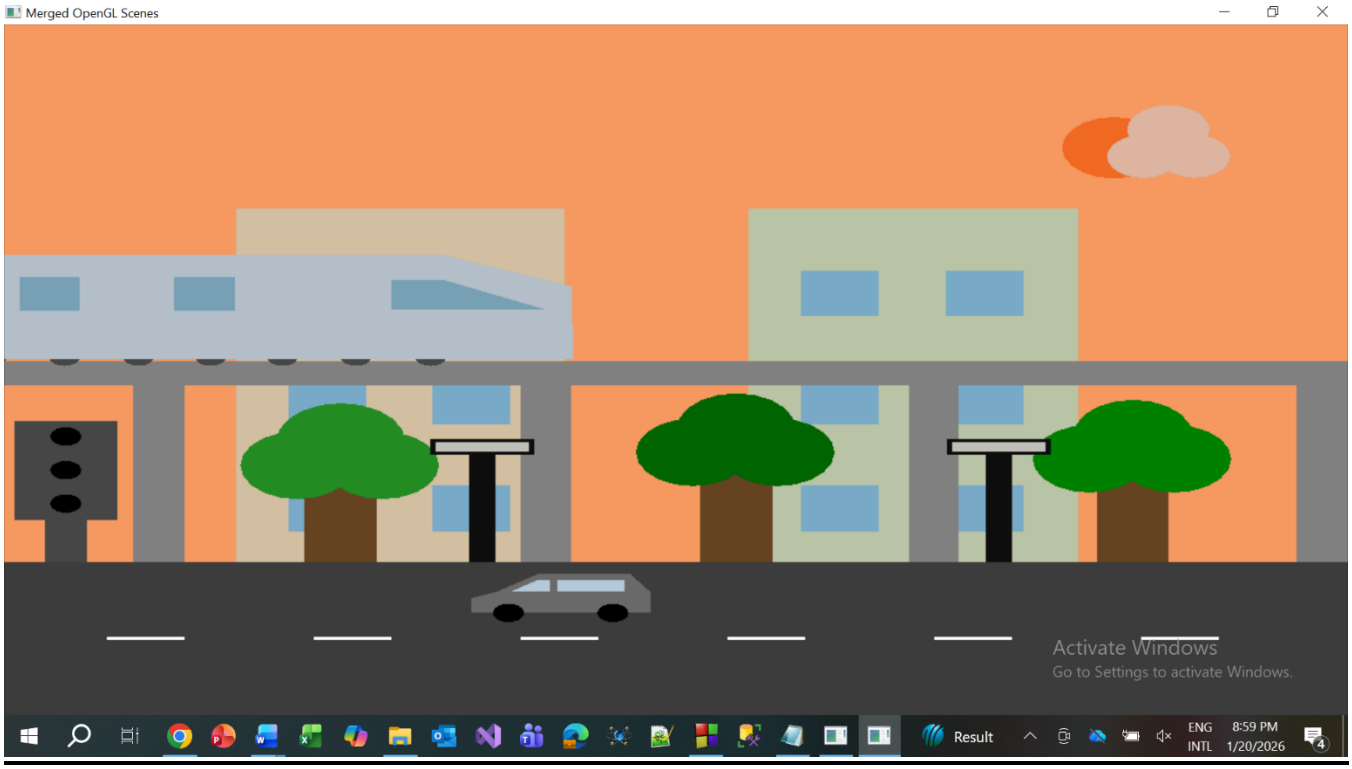
Screenshot of GeoGebra Design:

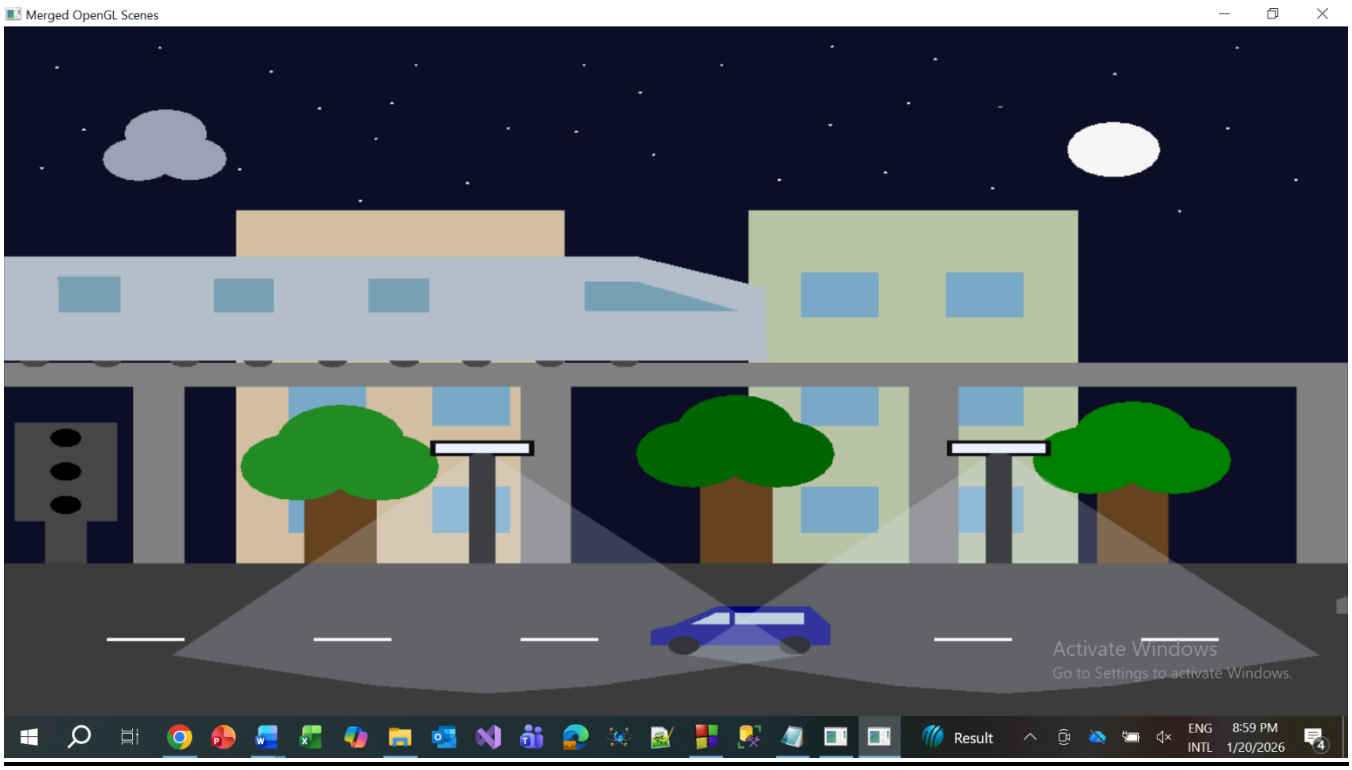


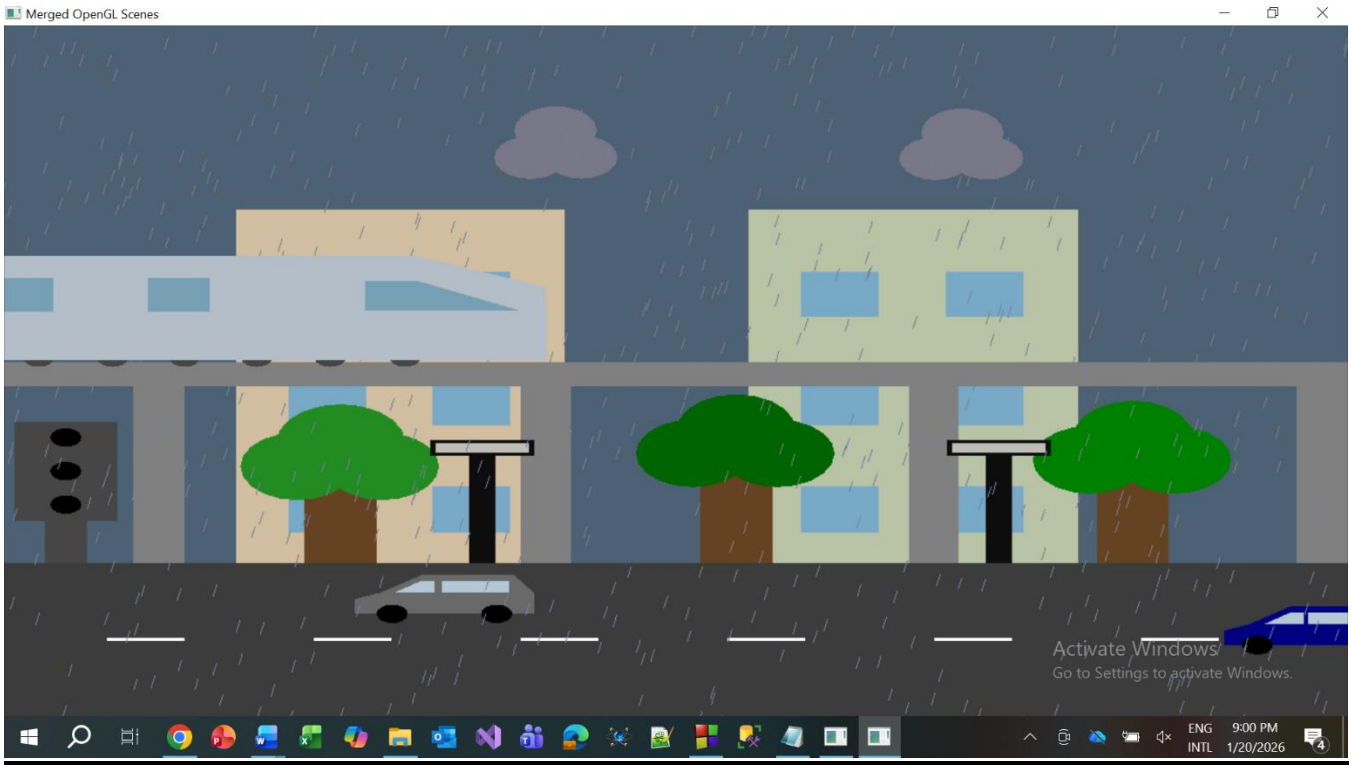
Screenshots of Project:

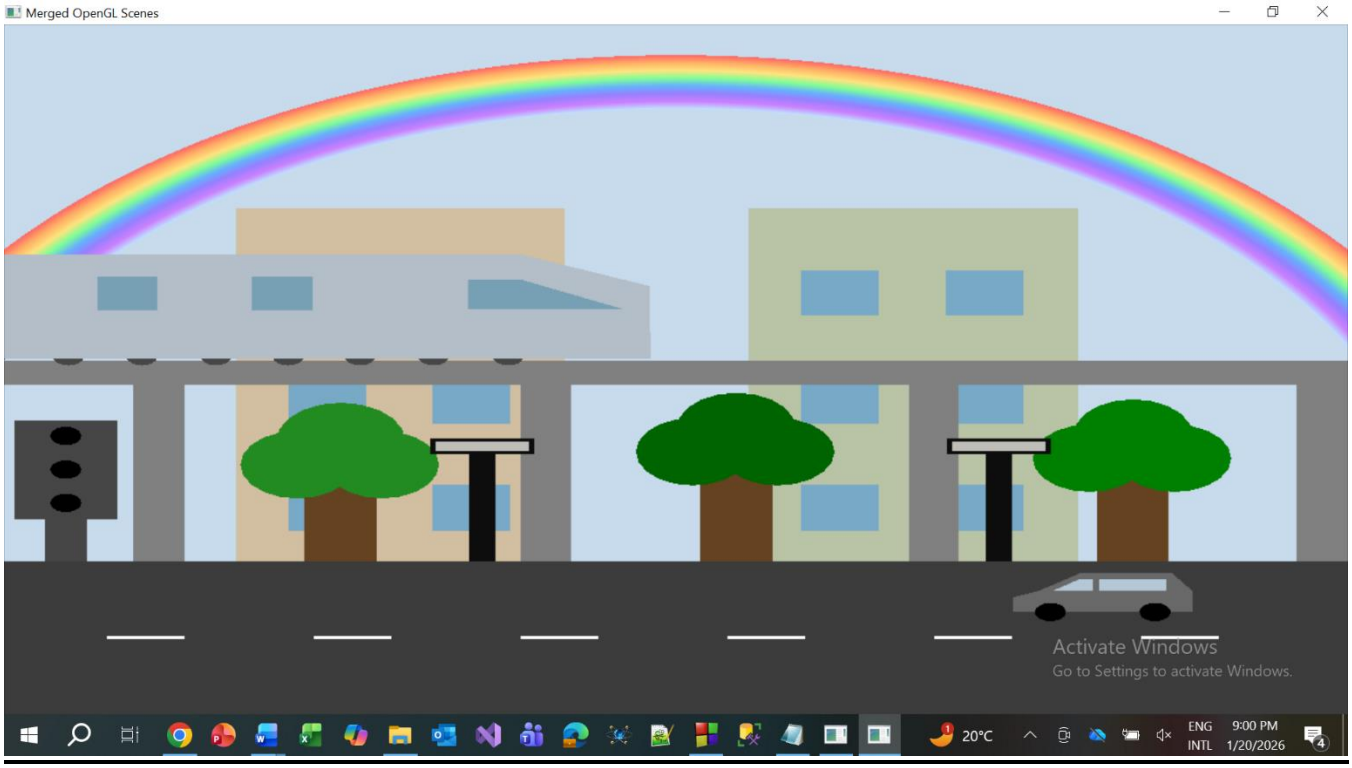
Scene 1:



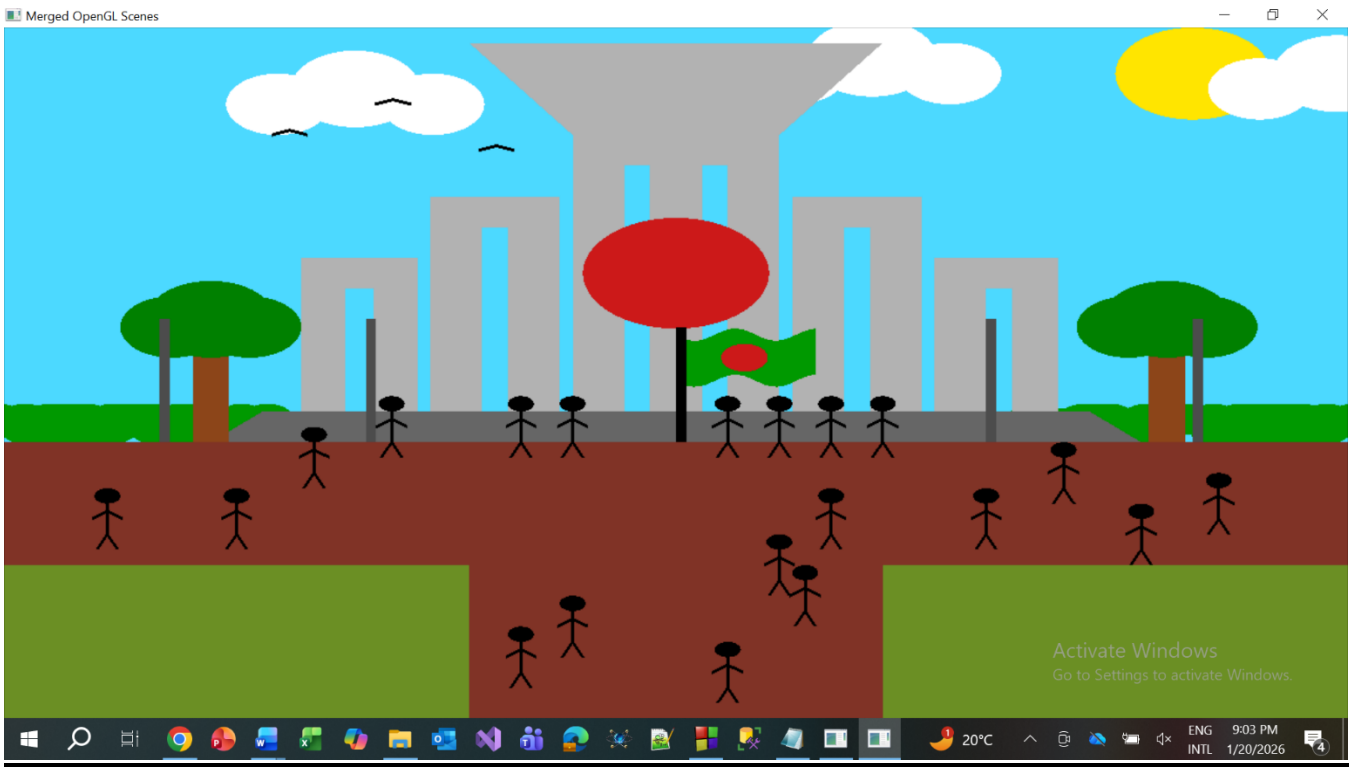


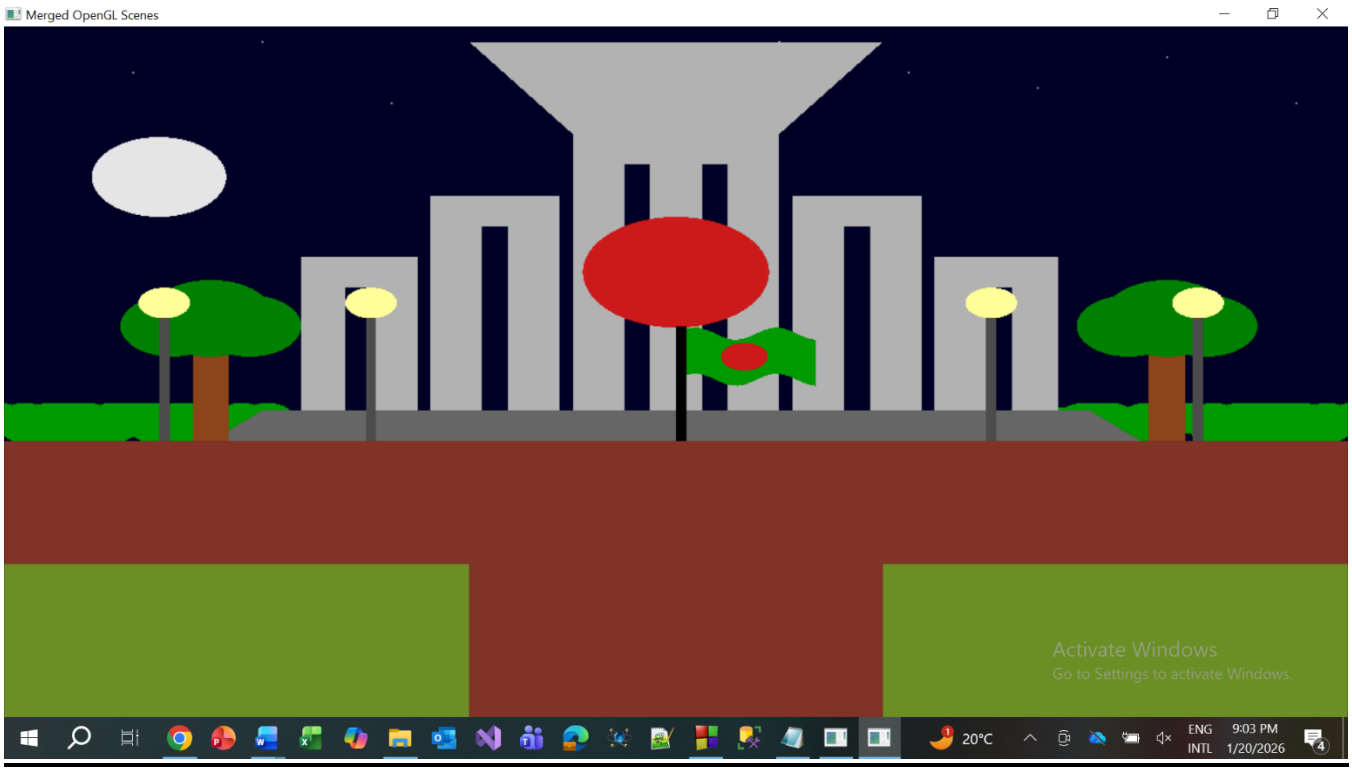

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
Result
ENG
INTL
8:59 PM
1/20/2026

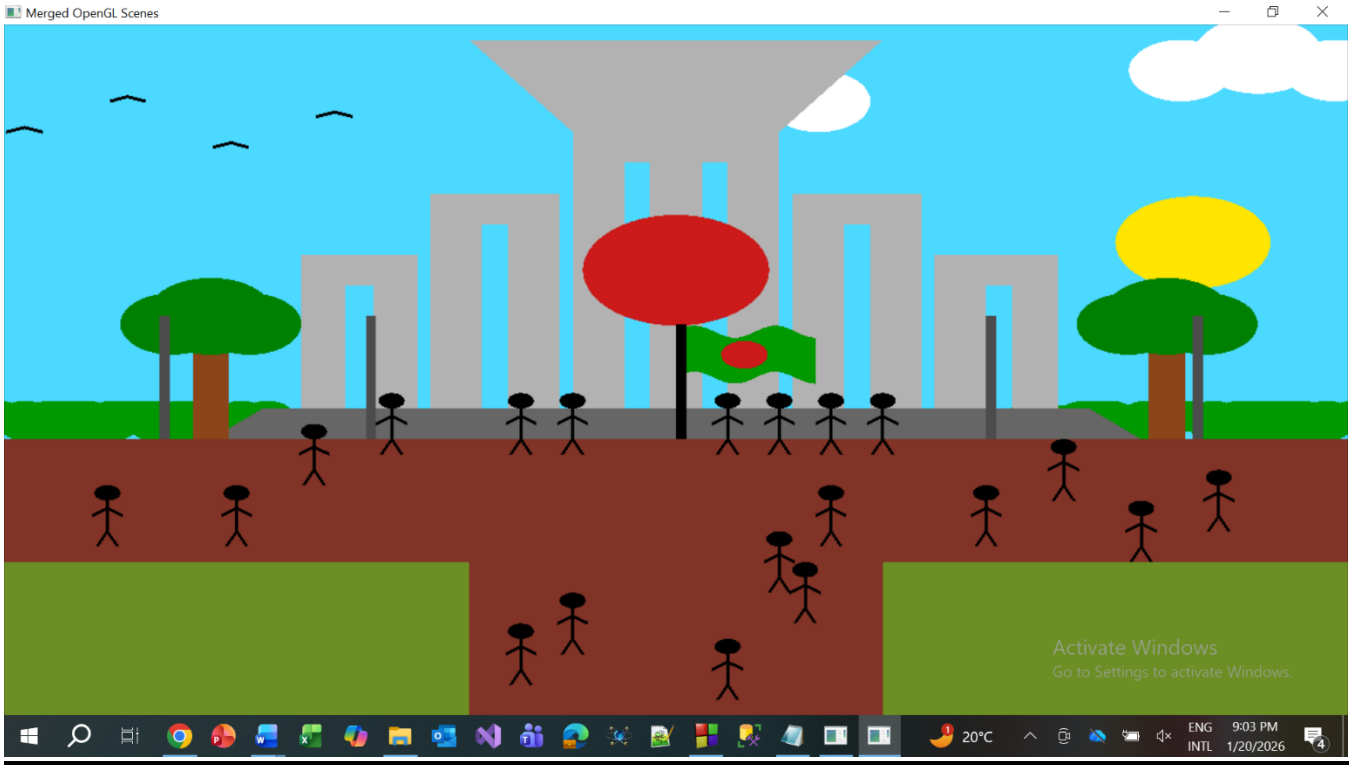

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
ENG
INTL
9:00 PM
1/20/2026

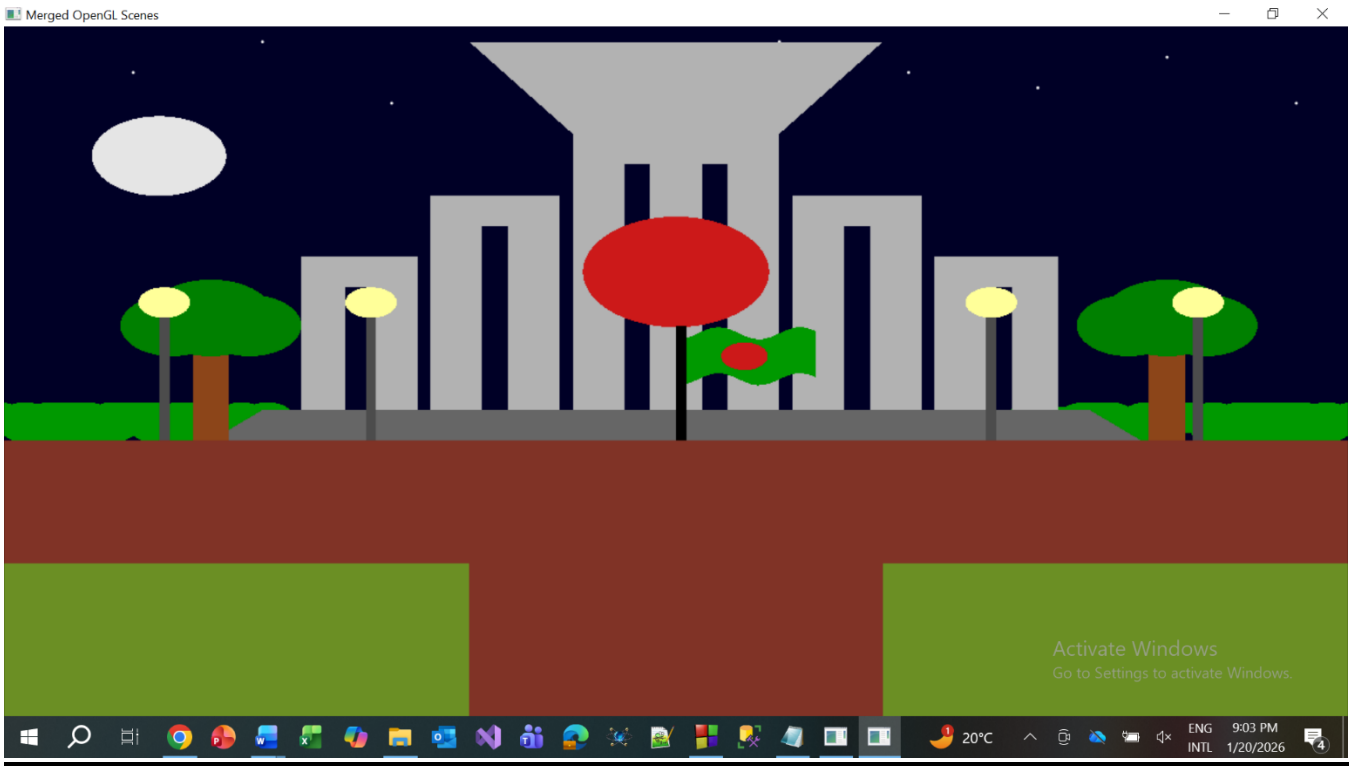

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:00 PM
1/20/2026

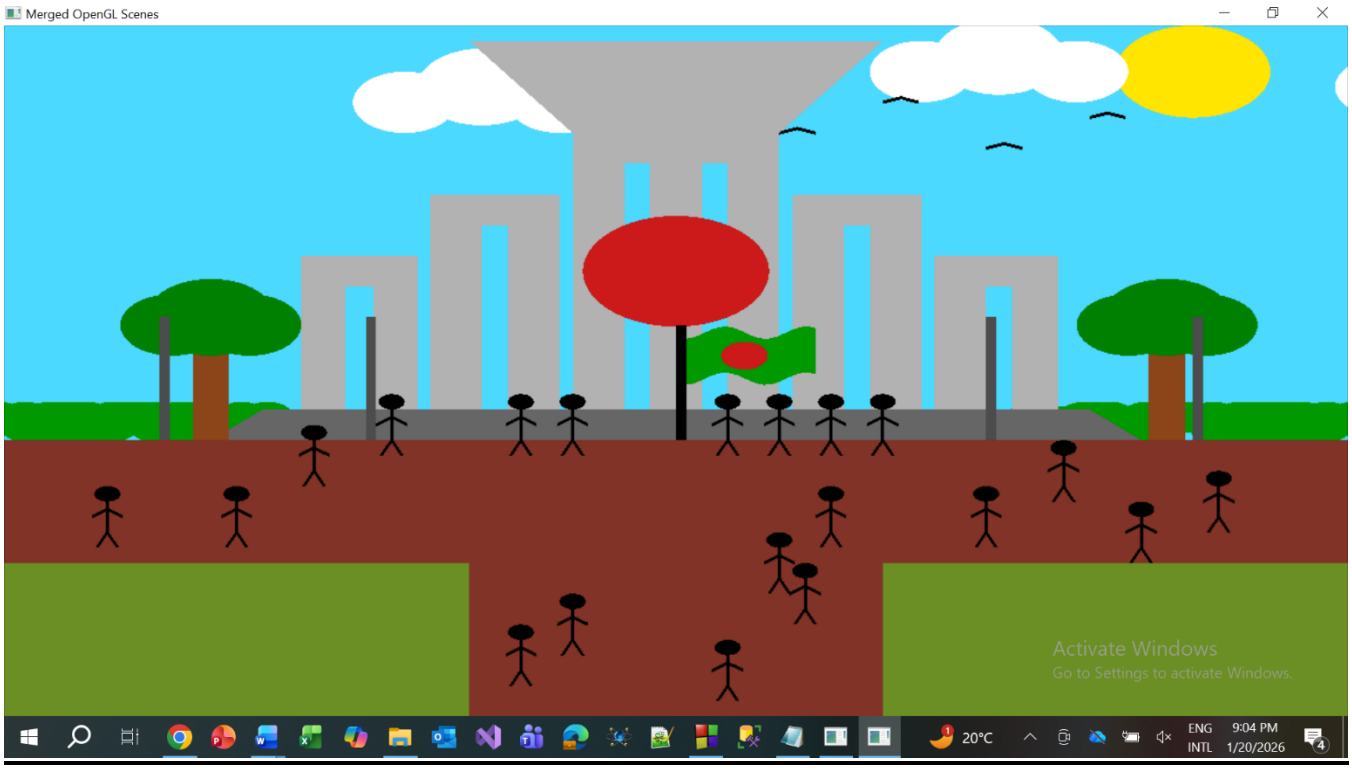
Scene 2:


Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:03 PM
1/20/2026

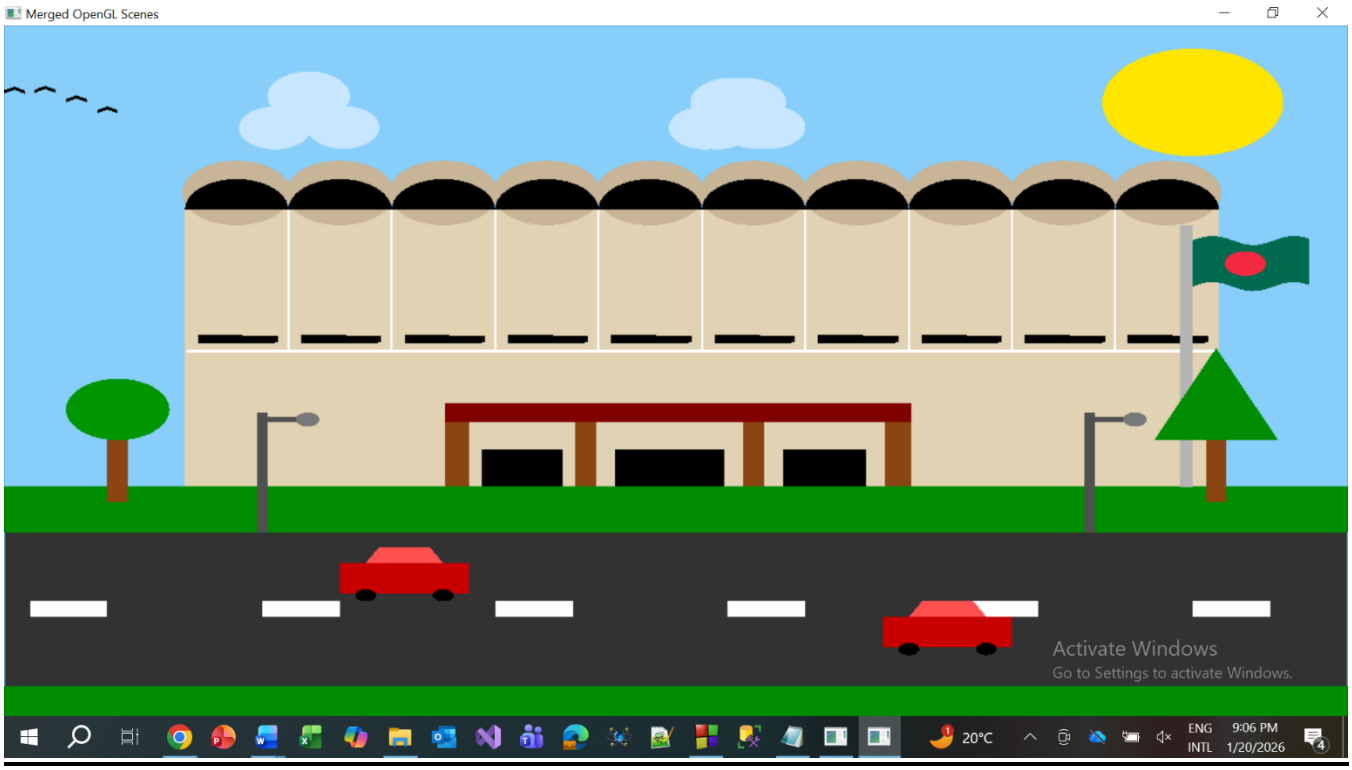

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:03 PM
1/20/2026

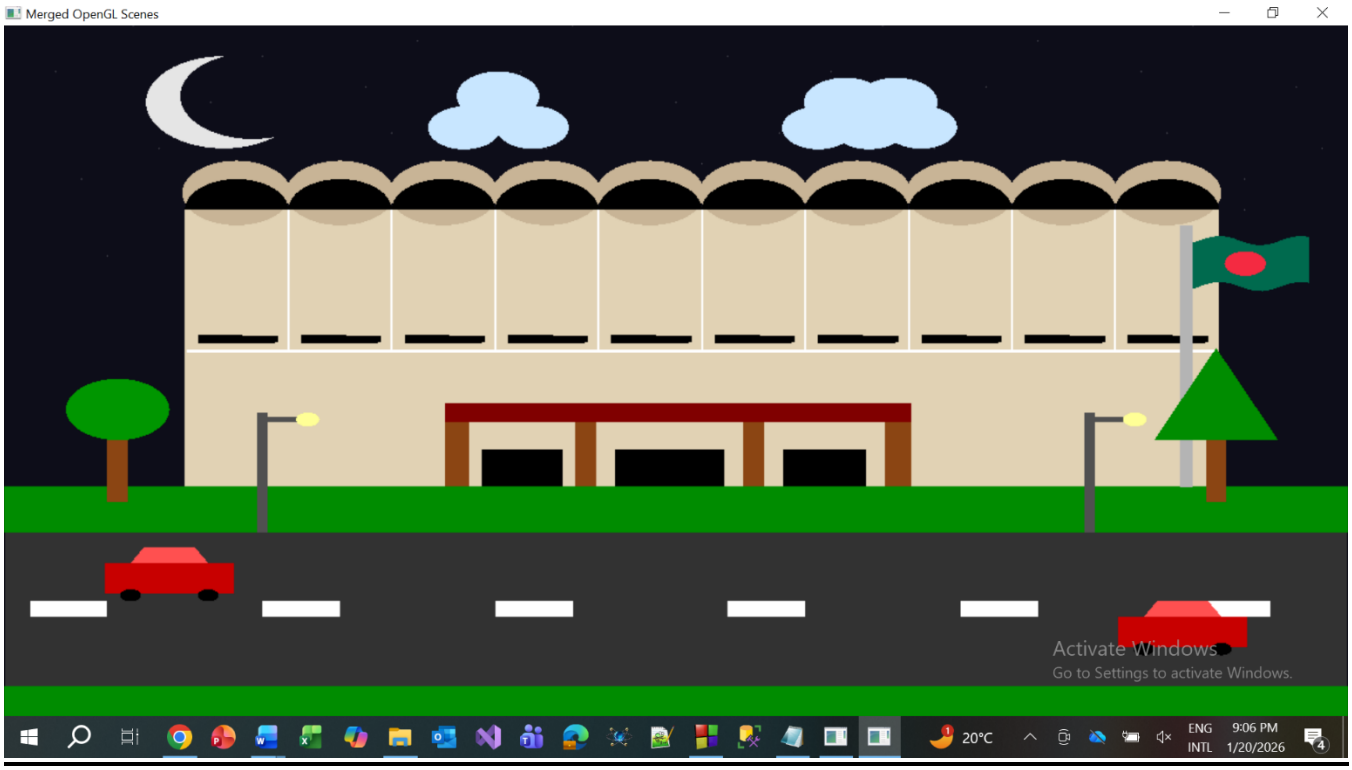

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:03 PM
1/20/2026

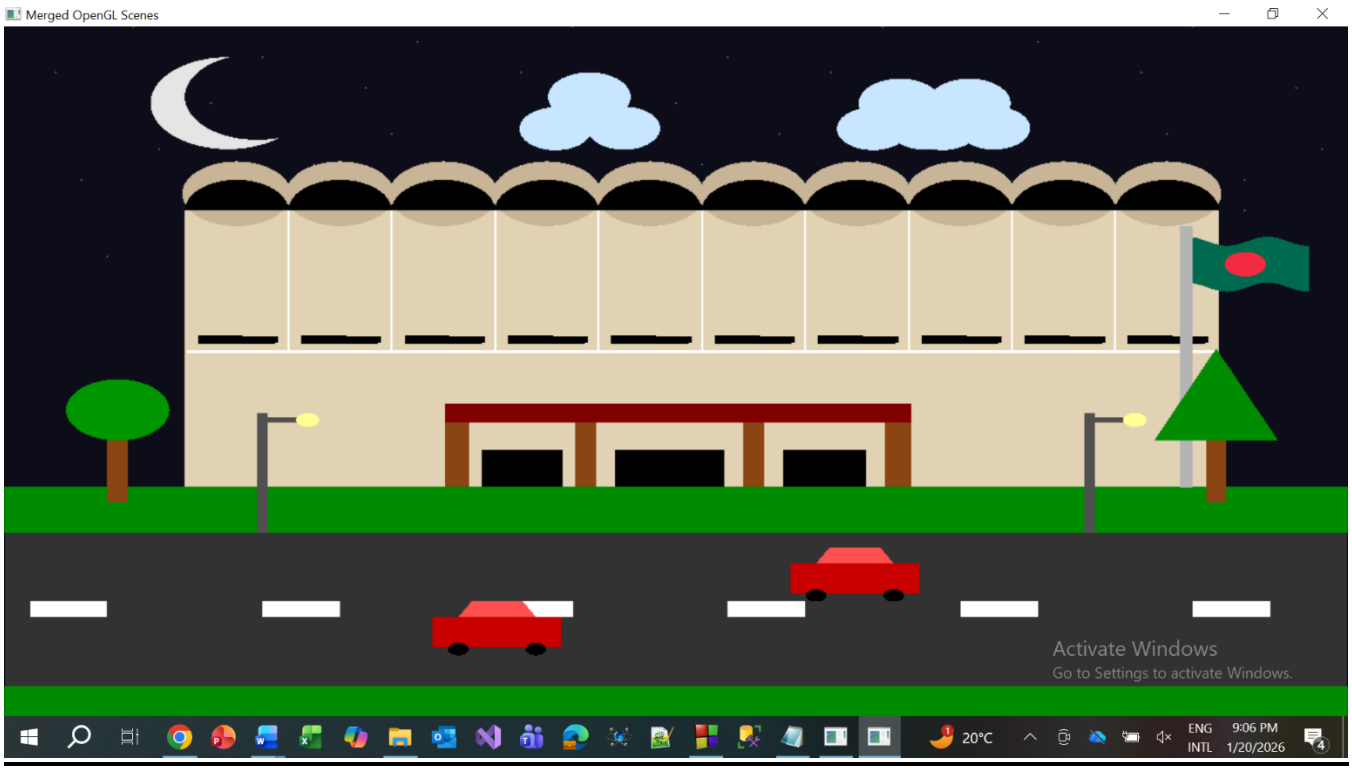

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:03 PM
1/20/2026

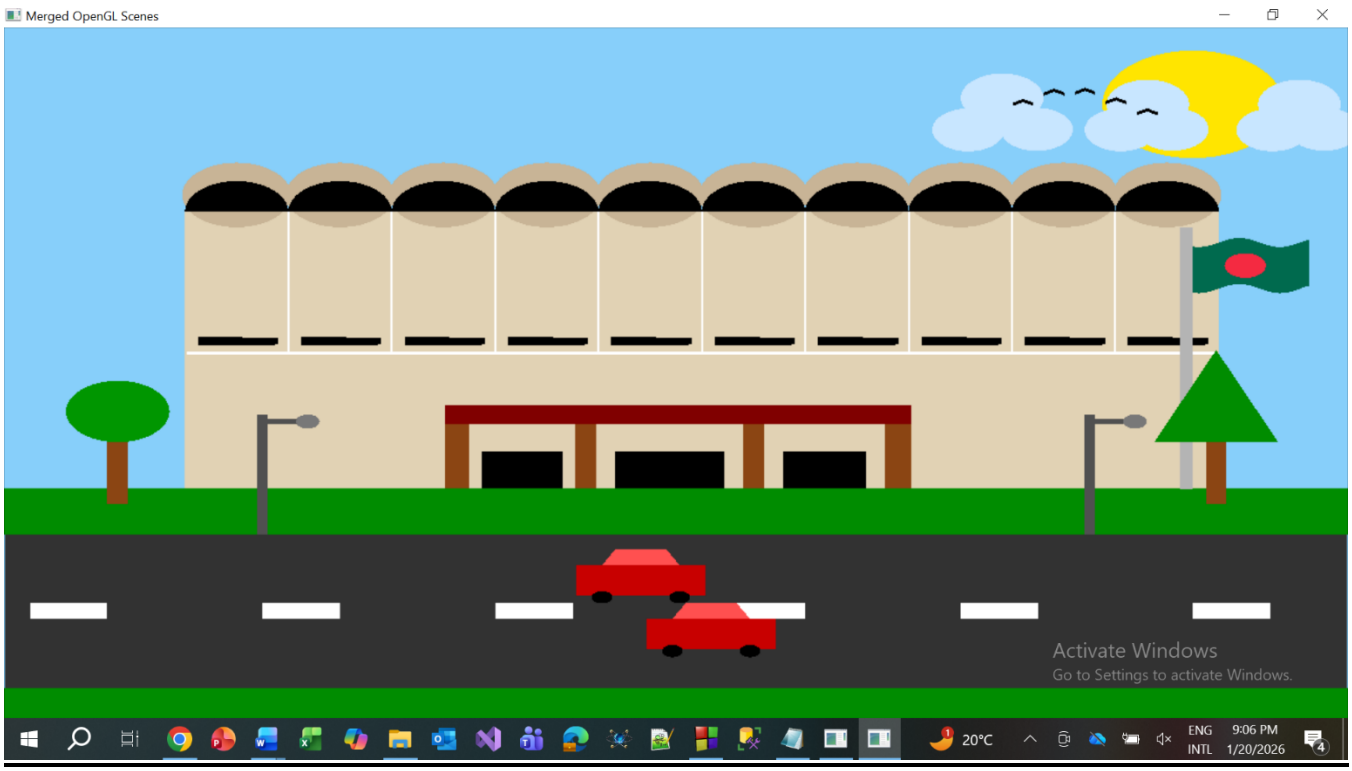

Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:04 PM
1/20/2026

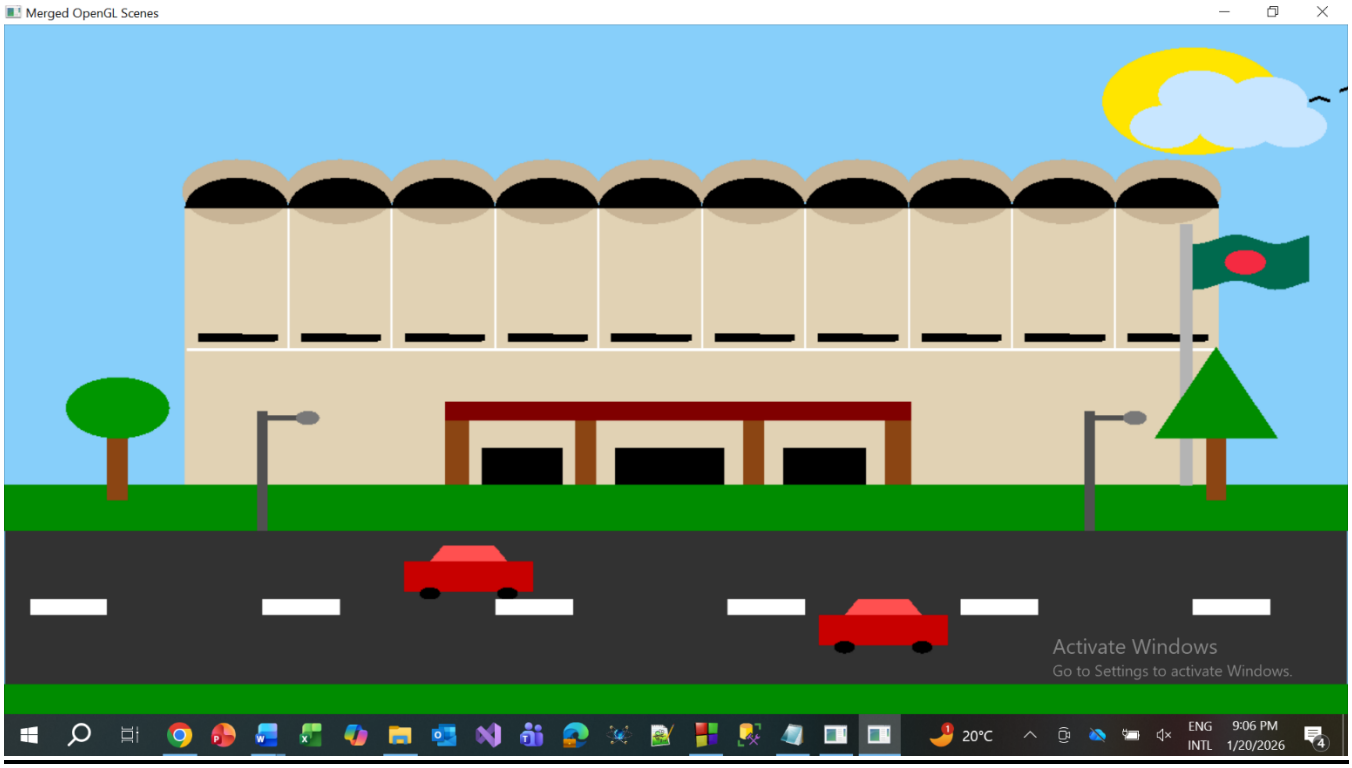
Scene 3:


Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:06 PM
1/20/2026


Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:06 PM
1/20/2026


Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:06 PM
1/20/2026


Merged OpenGL Scenes
Activate Windows
Go to Settings to activate Windows.
20°C
ENG
INTL
9:06 PM
1/20/2026



Code Availability:

GITHUB Link:

https://github.com/Fardin-Rahman-Mridul/Computer-Graphics.git

Future Work:

Although the Dhaka City project successfully presents a graphical visualization of major landmarks using 2D OpenGL, several enhancements can be considered for future development. One major improvement would be upgrading the project from 2D to 3D graphics to achieve more depth, realism, and immersive visualization. Incorporating advanced camera movement such as zoom, pan, and rotation would further enhance user interaction.

The project can be extended by adding real-time traffic behavior, including collision avoidance, traffic congestion simulation, and AI-based vehicle movement. Weather effects can be made more dynamic by introducing gradual transitions, wind influence, and seasonal changes. Sound effects such as traffic noise, rain, and metro movement could be integrated to improve realism.

Another future direction is performance optimization using modern OpenGL techniques such as Vertex Buffer Objects (VBOs) and shaders. The project could also be adapted for cross-platform deployment or converted into an interactive educational or tourism-based application showcasing Dhaka city landmarks.

Finally, adding a graphical user interface (GUI) for scene selection and controls, along with user-guided walkthroughs, would make the project more user-friendly and suitable for larger-scale visualization systems.

Conclusion:

The Dhaka City project successfully demonstrates a 2D graphical visualization of important landmarks of Dhaka using OpenGL and GLUT. Through multiple interactive scenes, the project presents realistic elements such as metro rail movement, traffic systems, weather effects, and time-based transitions. The integration of multiple scenes into a single application highlights effective modular design and coordination among team members.

Despite its strengths, the project has certain limitations. Since it is developed using 2D graphics, the depth and realism are limited compared to modern 3D simulations. The animations are frame-dependent, which may cause variations in speed on different systems. Additionally, the project relies on legacy OpenGL functions, restricting advanced graphical optimizations and shader-based effects.

In conclusion, this project serves as a strong foundational visualization of Dhaka city landmarks and urban behavior. It effectively combines graphical concepts, animation, and user interaction, while leaving ample scope for future enhancement using advanced graphics techniques and real-time simulation features.