



**American International University-Bangladesh**

Faculty of Science & Technology (CSE)

## **Computer Graphics**

Fall 2025-26

**Section: I, Group:**

**JourneyByCarCityToVillageViaBridge**

**Supervised by**

**Dipta Justin Gomes**

<b>Group Members</b>	<b>ID</b>
Mominul Islam	23-51372-1
MD. Sharif Iqbal	22-49589-3

## **1. Introduction**

This project is a 2D Computer Graphics Simulation developed using OpenGL and GLUT. It focuses on rendering two contrasting environments: a bustling Main City and a serene Village. The simulation features a real-time day-night cycle, dynamic weather elements like moving clouds, and interactive animated vehicles such as trains, buses, trucks, and cars. By utilizing geometric primitives, transformations, and timer-based animations, the project creates a lively and interactive visual representation of urban and rural life.

## **2. Problem Statement / Literature Review / Background**

Problem Statement: The goal of this project is to simulate the transition between an urban city environment and a rural village landscape. A key challenge is implementing a realistic Day-Night transition where the lighting, celestial bodies (Sun/Moon), and vehicle behaviors (headlights) change dynamically. The simulation aims to provide a smooth, interactive user experience that highlights the distinct visual characteristics of city versus village life.

### Background & Literature Review:

- Computer Graphics in Urban Simulation: Visualizing cityscapes and rural areas is essential in gaming and architectural modeling.
- OpenGL for 2D Rendering: OpenGL provides the necessary tools (primitives, color blending) to render complex 2D scenes efficiently.
- Day-Night Cycle: Dynamic lighting changes are used in simulations to create atmospheric depth and realism.

## **3. Objective of the Project**

### The main objectives of this project are:

- To develop a dual-scenario simulation (Main City and Village) using OpenGL and GLUT.
- To implement a seamless Day-Night transition, featuring the movement of the sun and moon and dynamic sky color changes.
- To animate scenario-specific elements:

- City: High-rise buildings, flyovers, moving trains, and heavy traffic (buses, cargo trucks).
- Village: Traditional huts, agricultural fields, rivers, hills, and local traffic.
- To allow user interaction via keyboard and mouse to control weather elements and vehicle speeds.
- To demonstrate proficiency in using OpenGL geometric primitives (GL\_QUADS, GL\_TRIANGLES, GL\_POLYGON, GL\_LINES).

## 4. Methodology / System Implementation

### 4.1 Functionality (User Controls)

General Controls:

- D / d: Switch to Day Mode (Bright sky, Sun visible).
- N / n: Switch to Night Mode (Dark sky, Moon visible, Vehicle headlights on).
- Space Key: Pause/Resume all global animations.
- V / v: Switch to Main City scenario.
- S / s: Switch to Village scenario.

### 4.2 Scenario-Specific Controls:

Scenario	Key / Input	Action
Main City	KEY_UP / KEY_DOWN	Increase/Decrease Train speed (Day).
	KEY_LEFT / KEY_RIGHT	Increase/Decrease Moon speed (Night).
	MOUSE_LEFT	Start/Stop the Train (Day).
	MOUSE_RIGHT	Start/Stop the Sun (Day).

Scenario	Key / Input	Action
	MOUSE_LEFT	Start/Stop the Moon (Night).
	MOUSE_RIGHT	Start/Stop the Cargo Truck (Night).
Village	KEY_UP / KEY_DOWN	Increase/Decrease Car 1 Speed.
	KEY_LEFT / KEY_RIGHT	Increase/Decrease Car 2 Speed.
	Space Bar	Toggles Sun, Moon, Clouds, and Cars simultaneously.
	MOUSE_LEFT	Pause/Resume Sun animation (Day).
	MOUSE_RIGHT	Pause/Resume Car 1 (Day) or Moon (Night).

## 5. Implementation Steps

**Step 1: Setting Up the Scene** The system initializes the window using GLUT and defines the viewport. A global variable tracks the current state (isDay), toggling background colors between cyan (Day) and dark blue (Night).

### Step 2: Drawing Static Elements

- **Main City:** Complex structures like high-rise buildings, residential apartments, and a concrete flyover are rendered using GL\_QUADS. A railway track is drawn using GL\_LINES.
- **Village:** The rural landscape is built using GL\_POLYGON for huts and hills, and GL\_TRIANGLES for trees and haystacks. A river is rendered using geometric curves.

### Step 3: Creating Moving Elements

- **City Vehicles:** A train moves along the elevated track, while buses and cargo trucks move along the road. The cargo truck's headlights are rendered as yellow cones during Night mode.
- **Village Vehicles:** Local cars travel along the winding village roads.

- Celestial Bodies: The Sun and Moon translate across the sky in opposite cycles.
    - Step 4: Implementing the Day-Night Cycle
  - Sky: The background color transitions from bright blue to dark navy.
  - Lighting: In Night mode, the sun is hidden, the moon appears, and vehicle headlights (e.g., on the Cargo Truck and Cars) become visible yellow polygons.
- Step 5: Animation Logic
- glutTimerFunc() is utilized to incrementally update variables controlling the position of clouds, vehicles, and celestial bodies (e.g., moveTrain, moveSun), ensuring smooth 60 FPS animation.
5. Significance of the Project
- Simulation Accuracy: Accurately represents the contrast between the verticality of city infrastructure and the horizontal, open nature of village landscapes.
  - Technical Demonstration: Showcases the ability to manage multiple scenes and complex state transitions (Day/Night) within a single OpenGL program.
  - Interactive Graphics: Demonstrates how user input can manipulate graphical elements in real-time, a core concept in game development and simulation software.

## 6. Conclusion

The "Journey By Car: City to Village" project successfully simulates two distinct environments—the Main City and the Village. By implementing a robust Day-Night cycle, the simulation offers a dynamic visual experience where the environment changes drastically between light and dark modes. The project fulfills all objectives by rendering complex static structures (buildings, flyovers, huts) and dynamic animated entities (trains, trucks, celestial bodies) using C++ and OpenGL. The inclusion of night-specific features, such as vehicle headlights and moon phases, adds a layer of depth and realism to the graphical representation.

## 7. References

1. Angel, Edward. *Interactive Computer Graphics: A Top-Down Approach with OpenGL*. Addison-Wesley.
2. Shreiner, Dave. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*.

3. OpenGL Documentation: <https://www.opengl.org/>

## 8. Screenshots of the System





