Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavrepalanchowk

Project Report
On
"COMP 342"

Submitted by:
Aaditya K.C (05)
Jagat K.C (07)
Saurav Bhujel (23)

Level:UNG CS(III/I)

Submitted to: Dhiraj Shrestha
Department of Computer Science and Engineering

Submission Date: 2/8/2026

Abstract

The Day–Night Transition Simulation is a computer graphics project that models the natural cycle of day and night using mathematical and rendering techniques. The simulation represents the apparent motion of the sun and moon through rotational movement along semi-circular paths, where time is mapped to angular displacement to ensure smooth transitions. Variations in lighting, sky color, and shadow behavior are dynamically calculated based on the height of the sun.

In addition to the daily cycle, the system supports **seasonal changes**, allowing the user to switch between summer and winter. Seasonal variation affects sunrise and sunset timings, daylight duration, sky color tones, and environmental elements such as snow. By integrating trigonometric formulas, interpolation methods, and time-based animation, the project demonstrates how real-world astronomical and environmental changes can be effectively visualized using 2D computer graphics and OpenGL.

1. Introduction

The day–night cycle and seasonal variations are fundamental natural phenomena caused by the Earth's rotation and its tilt around the Sun. These phenomena influence light, temperature, shadows, and environmental appearance. In computer graphics, simulating such real-world behavior is an effective way to understand animation, transformations, lighting models, interpolation, and mathematical modeling.

This project, *Day–Night Transition Simulation Using PyOpenGL*, is a 2D computer graphics simulation developed in Python using the PyOpenGL library. The simulation visually represents a complete 24-hour day–night cycle with smooth transitions between daytime and nighttime. It also incorporates seasonal variation by allowing users to switch between summer and winter modes, affecting sunrise and sunset timing, daylight duration, sky color, and environmental elements such as snow.

The primary motivation of this project is to bridge theoretical computer graphics concepts with practical implementation. By using trigonometric functions, time-based animation, and color interpolation, the project demonstrates how realistic natural cycles can be modeled and visualized in a graphical environment.

2. Project Description

2.1 Overview of the System

The Day–Night Transition Simulation models the apparent motion of the Sun and Moon across the sky. Instead of rotating the Earth, the simulation rotates the Sun and Moon along semi-circular orbital paths. Time is represented as a continuous variable mapped to angular displacement, ensuring smooth and realistic motion.

The environment consists of a sky, ground, sun, moon, and surrounding elements. As time progresses:

- The Sun rises, reaches its peak at noon, and sets.
- The Moon follows a similar path during nighttime.
- Sky color transitions smoothly between morning, noon, evening, and night.
- Brightness and lighting intensity change dynamically.
- Shadows change direction and length based on the Sun's position.

Seasonal changes modify the length of daytime and nighttime. In summer mode, days are longer, while in winter mode, nights are longer and snow effects are enabled.

2.2 Time Modeling and Animation

Time in the simulation is not handled as discrete hours but as a continuously increasing variable. A full 24-hour day is mapped to angular motion using trigonometric functions.

For daytime:

- Day progress is calculated as a ratio of elapsed daytime to total daytime duration.
- The Sun's angle is computed using this ratio and mapped over a semi-circular arc.

For nighttime:

- Night progress is calculated similarly.
- The Moon's position is computed using the same mathematical approach.

This approach ensures smooth animation without abrupt transitions.

2.3 Mathematical Modeling

The position of the Sun and Moon is calculated using trigonometric equations:

- **x-position:**
  $x = center\_x + radius \times \cos(angle)$
- **y-position:**
  $y = center\_y + radius \times \sin(angle)$

Brightness is derived from the vertical height of the Sun or Moon:

- $brightness = \max(0, \sin(angle))^{1.5}$

This brightness value is used to control sky color, ground color, and overall lighting intensity.

2.4 Lighting, Color, and Shadows

Lighting plays a crucial role in realism. As the Sun rises, brightness increases gradually, reaching its maximum at noon and decreasing towards sunset. Sky and ground colors are interpolated based on brightness to simulate dawn, daylight, dusk, and night.

Shadows are computed opposite to the Sun's direction. Their length depends on the Sun's height:

- Long shadows during sunrise and sunset
- Short shadows at noon

For clarity and simplicity, shadows are disabled during winter mode.

2.5 Seasonal Variation

The simulation supports two seasons:

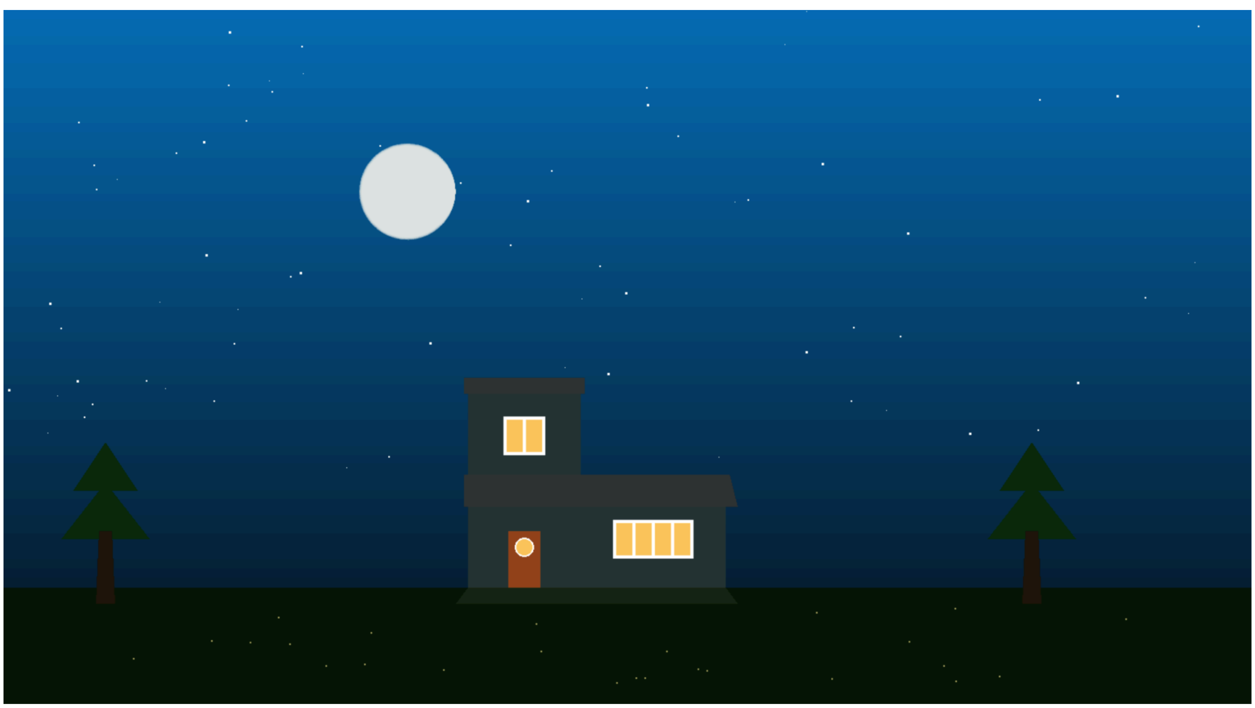
- **Summer:** Longer daytime, shorter nights, brighter sky tones
- **Winter:** Shorter daytime, longer nights, snow-covered ground, and snowfall effects

Users can toggle between seasons using keyboard input, instantly observing changes in lighting and environmental behavior.

2.6 Screenshots (Snapshots)

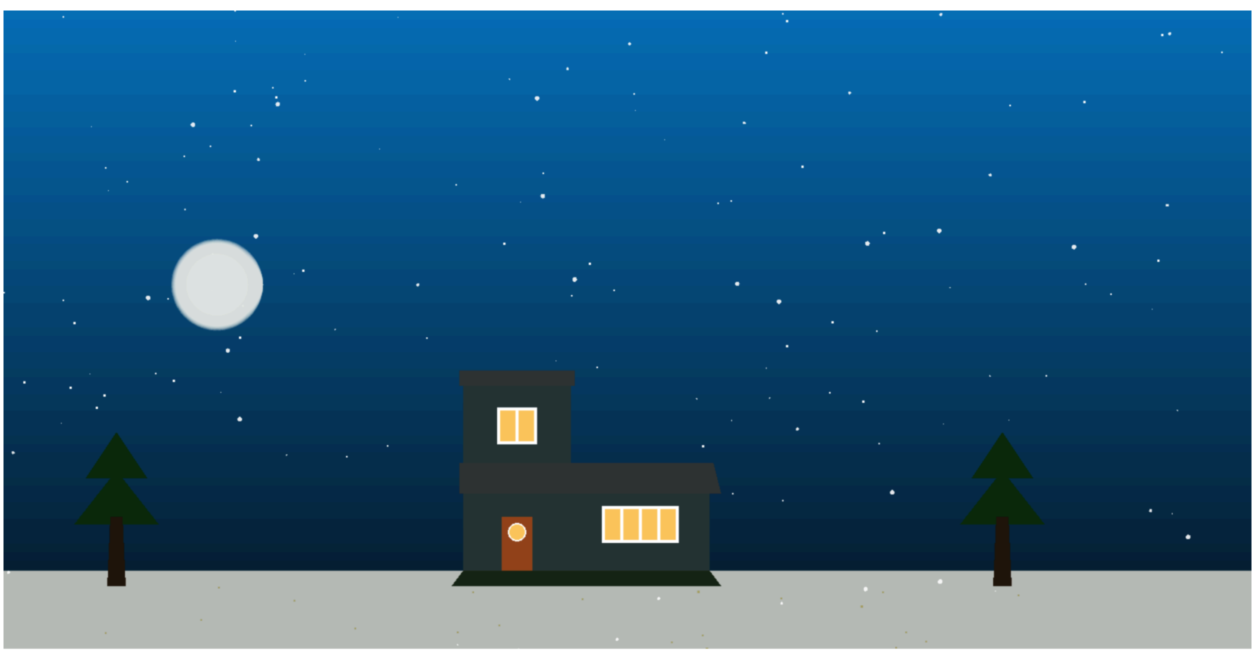
Summer Day View

Summer Night View



Winter Day View

Winter Night View



3. Technologies Used

- **Programming Language:** Python
- **Graphics Library:** PyOpenGL (OpenGL + GLUT)
- **Mathematical Tools:** Trigonometric functions (sine, cosine), interpolation techniques
- **Supporting Libraries:** NumPy, Pillow

4. System Requirements

- Python 3.8 or later
- OpenGL-capable graphics hardware
- Installed libraries: PyOpenGL, PyOpenGL-accelerate, NumPy, Pillow

5. Conclusion

The Day–Night Transition Simulation successfully demonstrates how real-world natural phenomena can be modeled using computer graphics techniques. By integrating mathematical modeling, time-based animation, lighting calculations, and color interpolation, the project provides a realistic visualization of day–night cycles and seasonal changes.

This project enhanced our understanding of core computer graphics concepts such as transformations, animation loops, lighting models, and user interaction. Although the simulation is limited to 2D rendering, it establishes a strong foundation for more advanced 3D simulations. Overall, the project fulfills its objectives and serves as an effective educational tool for learning graphics programming with OpenGL.

6. References

- LearnOpenGL – https://learnopengl.com/
- PyOpenGL Documentation – https://pyopengl.sourceforge.net/
- OpenGL Programming Guide

7. Source code link

- https://github.com/aadukcr7/computer-graphics-mini-project.git