Drawing an "Interior Design" With Some Moving Objects in a Room

Submitted By

Student Name	Student ID
Mahmudul Hasan Piash	221-15-5606
Faiyaz Khan Sami	221-15-4910
Zakia Sultana Nisa	213-15-5172

LAB PROJECT REPORT

This Report is Presented in Partial Fulfillment of the course

CSE422: Computer Graphics Lab in the Department of Computer Science and Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

15/12/2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Syada Tasmia Alvi**, **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:

____

Syada Tasmia Alvi,
Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by

______________________________ Mahmudul Hasan Piash ID: 221-15-5606 Dept. of CSE, DIU	
______________________________ Faiyaz Khan Sami ID: 221-15-49110 Dept. of CSE, DIU	______________________________ Zakia Sultana Nisa ID: 213-15-5172 Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:

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.

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

Mapping Course Outcome (COs) with the Teaching-Learning and Assessment Strategy:

CO's	Corresponding PO number	Domain Level/ Learning Taxonomy	Level of Knowledge Profile (K)	Complex Engineering Problem (EP)	Complex Engineering Activities (EA)
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	EA3

The mapping justification of this table is provided in section **4.2.1**, **4.2.2** and **4.2.3**.

Table of Contents

Chapter 1

Introduction

This chapter provides an overview of the Computer Graphics project, focusing on the creation of an animated interior design scene. The primary aim of the project is to demonstrate proficiency in graphical modeling, animation techniques, and real-time rendering using OpenGL. By incorporating moving objects into the scene, the project explores dynamic simulations in the context of interior design, presenting a modern approach to visualizing room layouts and furnishings.

1.1 Introduction

Computer graphics is a field of visual computing that deals with generating, manipulating, and displaying images and animations using computational techniques. It plays a pivotal role in industries ranging from entertainment and gaming to engineering and architecture. In particular, computer graphics enables the creation of realistic models, simulations, and animations, facilitating the design, visualization, and analysis of complex objects and environments.

OpenGL (Open Graphics Library) is one of the most widely used cross-platform APIs for rendering 2D and 3D graphics. It provides a powerful, flexible, and efficient set of tools for developers to create visual content with high performance and rich graphical effects. With its extensive capabilities, OpenGL allows for real-time rendering, making it an ideal choice for dynamic and interactive applications such as simulations and virtual environments.

The primary goal of this project is to design and animate an interior space using OpenGL. This project involves creating a 3D model of a room with realistic elements, such as walls, furniture, windows, and other room features. Additionally, it will incorporate animated objects within the space to simulate movement or change over time, making the scene more dynamic and interactive.

By leveraging computer graphics algorithms like DDA, Bresenham's line algorithm, and Midpoint Circle Algorithm, the project will generate geometric shapes and paths essential for object placement and movement. The room and objects will undergo various transformations, such as translation, rotation, scaling, and reflection, allowing for the exploration of 2D transformations in a 3D environment.

Ultimately, the project will not only result in a visually appealing room design but also provide a deeper understanding of key graphical techniques, real-time rendering, and the application of modular programming principles. Through this project, we aim to demonstrate how computer graphics can be used to create interactive and visually engaging environments, with potential applications in interior design, architecture, and virtual modeling.

1.2 Motivation

The motivation behind this project is to explore the world of computer graphics and animation using OpenGL, particularly for the purpose of interior design visualization. The field of interior design has long benefited from graphical representation in architectural software, and with the rise of interactive graphics technologies, there is an opportunity to bring this into a more immersive environment.

Interior designers and architects are increasingly relying on computer graphics to create detailed, realistic 3D models of their projects. However, most of these systems are expensive and often require professional expertise to operate. By using OpenGL and focusing on real-time rendering and interactive features, the project aims to make interior design accessible to a wider audience, including students and enthusiasts interested in learning more about design or virtual space manipulation.

Furthermore, the addition of moving objects to the scene (such as animated furniture or objects) enhances the interactivity of the room, providing a better understanding of how spaces can be dynamically modified. This also aligns with current trends in virtual reality (VR) and augmented reality (AR), where users are not only passive observers but active participants in the design process.

1.3 Objectives

The main objectives of this project are as follows:

- **To implement basic graphics algorithms**: These include algorithms such as DDA (Digital Differential Analyzer), Bresenham's line algorithm, and the Midpoint Circle Algorithm to draw basic shapes and objects in the room.
- **To apply 2D transformations**: This includes translation, rotation, scaling, reflection, and shear to modify objects within the scene, demonstrating how these transformations can affect the layout and appearance of the room.
- **To create an animated graphical scene**: The project will feature a room with animated objects (e.g., moving furniture, opening doors), demonstrating how to incorporate dynamic elements into the scene.
- **To demonstrate modular programming and real-time rendering**: The project will employ modular programming techniques for structuring the code efficiently, ensuring that the rendering of the scene happens in real time for smooth animation and interaction.

1.4 Feasibility Study

Several studies and case studies have explored the use of computer graphics and OpenGL for interior design simulations. One notable example is the development of real-time 3D design applications used in architecture and gaming. These applications allow users to visualize and interact with interior layouts, making design processes more immersive and efficient. Research in this field has also focused on optimizing the rendering process to ensure real-time performance while maintaining graphical quality. Additionally, modern mobile apps such as Roomstyler and SketchUp have integrated 3D modeling tools that allow users to create and manipulate interior spaces. These studies and tools provide valuable insights into how 3D graphics can be applied to interior design, laying the foundation for our project's implementation.

1.5 Gap Analysis

The gap analysis highlights areas where this project will contribute new insights or improvements:

1. **Lack of OpenGL-Based Interior Design Tools**: Most existing interior design software uses proprietary engines or non-OpenGL-based systems. This project will explore how OpenGL, a widely accessible and open-source graphics library, can be used for interior design and animation.
2. **Limited Moving Object Animations in Design Software**: While many design tools offer static visuals, there is a gap in tools that allow for real-time moving objects, such as furniture rearranging or animated lights. This project fills that gap by incorporating moving elements into a static room design.

3. **Lack of Educational Tools for Interior Design Visualization**: Current tools are often either too complex for beginners or too simplistic to provide realistic results. By implementing this project with OpenGL and focusing on modular design, we aim to provide an easy-to-use and educational tool for learning interior design concepts.
4. **Real-Time Interaction with Interior Design Models**: Many interior design programs generate beautiful static 3D models but do not provide real-time interaction. This project addresses that gap by focusing on real-time rendering and interaction, allowing users to manipulate the objects and view changes instantly.
5. **2D Transformation Integration**: While most design tools rely on simple drag-and-drop interfaces, there is a gap in tools that integrate graphical algorithms such as DDA, Bresenham, and Midpoint Circle with real-time design modifications. This project fills that gap by using these algorithms to generate lines, circles, and other shapes as part of the design.

1.6 Project Outcome

The expected outcomes of the project include:

1. **Development of Interior Design Visualization**: Creation of a dynamic 3D interior design model, including key components such as walls, furniture, and windows, all designed with realistic visual effects.
2. **Implementation of Moving Objects**: Introduction of animated elements such as moving furniture, objects, or lighting effects within the room to create a more immersive and interactive experience.
3. **Demonstration of 2D and 3D Graphics Algorithms**: The project will implement and showcase algorithms such as DDA, Bresenham's Line Algorithm, and the Midpoint Circle Algorithm to generate various shapes and paths for the objects in the scene.
4. **Real-Time Rendering and Interaction**: The project will emphasize real-time rendering, allowing for interactive exploration of the room. Users will be able to manipulate the scene using OpenGL.
5. **Application of 2D Transformations**: The interior objects will undergo various transformations (translation, rotation, scaling, reflection, and shear) to demonstrate the effects of each transformation in real-time.
6. **Modular Programming**: By structuring the code in modular blocks, the project will ensure that it is maintainable, scalable, and reusable, making it easy to enhance the design or add additional features in the future.

Chapter 2

Scene Description & Architecture

This chapter describes the overall scene design and architecture of the Computer Graphics project. It outlines the main graphical components, their interactions, and the technologies used to create an efficient, visually appealing, and animated OpenGL-based scene.

2.1 Requirement Analysis & Design Specification

2.1.1 Scene Overview

The project simulates a realistic, fully furnished Modern 3D Bedroom Interior. The scene is designed to emulate a real-world environment with interactive elements and environmental controls. The primary focus is on hierarchical modeling of furniture, texture mapping for realism (wallpapers, photo frames), and dynamic lighting to simulate Day and Night cycles. The room includes various objects such as walls, furniture, appliances, and decorative items. Each component is modeled using basic geometric shapes combined into more complex structures, and texture mapping is applied for realism. Users can navigate the room using a 360-degree camera system that allows for a full, immersive view of the environment. Key interactive features in the scene include:

- **Furniture Interaction:** The user can control the opening and closing of the door and window, adjust the fan speed, and toggle electrical appliances such as lamps, TV, and PC.
- **Lighting Control:** The scene incorporates dynamic lighting that changes based on time of day, with a Day/Night cycle affecting the lighting intensity and room ambiance.
- **Object Animation:** Several objects have built-in animations, including rotating fan blades, sliding windows, and changing TV channels with animation.

This project showcases not only the graphical representation of a modern bedroom but also emphasizes the importance of user interaction and animation in creating a realistic, immersive 3D environment.

2.1.2 Object Components

The following is a list of the major objects and components used in the scene, detailing their geometric properties and functionality:

A. Room Structure

- **Walls:** The room includes four walls (front, back, left, and right) with realistic textures applied using texture mapping. The back wall has a wallpaper texture, and the window and door are modeled as cutouts in the respective walls.
- **Floor:** A tiled floor is created using a loop to generate a grid, with each tile textured and outlined to create a realistic off-white tile pattern.
- **Ceiling:** The ceiling is modeled as a flat surface with a simple texture and lighting effects for realism.

B. Furniture

- **Bed:** A composite object consisting of a wooden frame, four legs, a blue mattress with trim, two white pillows, a headboard, and a folded blanket. The bed components are modeled using cubes and rectangles, with a focus on realistic proportions and textures.
- **Wardrobe:** A tall wooden wardrobe with two doors and gold handles. The doors are modeled with hinges to allow for an open/close animation.
- **Bookshelf:** A wooden bookshelf with randomly generated books of varying heights and colors. A small plant pot sits atop the shelf, adding a decorative touch.
- **Sofa:** A comfortable sofa with a chocolate-colored base, seat cushions, back cushions, armrests, and decorative throw pillows. The sofa's cushions and legs are modeled to provide a realistic appearance.

C. Electronics & Appliances

- **Fan:** A ceiling-mounted fan with a motor housing and four rotating blades. The fan's rotation is animated, and the fan can be toggled on/off using user input.
- **Lamp:** A desk lamp with a gold base, a rod-like stem, and a cone-shaped shade. The lamp includes a light source that can be toggled on/off.
- **TV:** A flat-screen TV on the left wall with animated screen patterns. The TV channel can be switched between five different patterns, and the TV itself can be toggled on/off.
- **PC:** A PC setup with a monitor, tower, and power LED. The monitor displays a Windows logo when powered on.

D. Decorative Items

- **Photo Frame:** A photo frame mounted on the back wall, capable of displaying a texture if loaded. The photo can be either an image or a blank white space.
- **Curtains:** Wavy curtains hanging from a rod on the right wall window. The curtains are animated with a sine wave function to simulate the natural movement of fabric.
- **Wall Sconce:** A decorative light fixture on the back wall, providing additional lighting effects when turned on.

E. Interactivity

- **Door and Window:** Both the door and window can be opened or closed, with smooth animations controlled by user input. The door opens to a 90-degree angle, and the window slides open to a 45-degree angle.
- **Lighting Effects:** The lighting in the room changes based on whether it is Day or Night. During the day, the lighting is bright and clear, while at night, the room is dimmer with a warm ambient light.
- **Electrical Appliances:** The user can toggle the lamp, fan, TV, and PC on or off, with corresponding animations and lighting effects.

Each object in the scene is modeled using basic geometric shapes such as cubes, spheres, and cylinders, combined with texture mapping and lighting to achieve a realistic appearance. The hierarchical modeling approach ensures that objects are composed of multiple components that are grouped and manipulated as units for efficient rendering and interaction.

2.2 Use of Modern Tools

To create a high-quality, dynamic, and efficient scene, a variety of modern tools and technologies have been employed:

- **OpenGL / FreeGLUT**: These open-source graphics libraries were used for rendering both 2D and 3D objects in real-time. OpenGL provides the core functionality for drawing geometric shapes, while FreeGLUT serves as an additional layer for window management and input handling.
- **Code::Blocks**: The development environment used for writing, compiling, and debugging the OpenGL programs efficiently. These IDEs provided an optimal setup for OpenGL and C++ integration, supporting the full lifecycle of the project from writing code to debugging and testing the final scene.
- **GLUT Libraries**: These frameworks are essential for window creation, input handling, and animation timing. GLUT handles user inputs (like keyboard and mouse events) while also managing animation frame rates, ensuring smooth real-time rendering of the scene.
- **PowerShell / Windows GDI+:** Used within the code to dynamically convert JPG images to BMP format for texture compatibility.
- **C++ Programming Language**: C++ was the core language for implementing graphics algorithms and transformations. Object-oriented programming allowed for modular development, making it easier to manage the various components of the scene (such as individual furniture objects and animations).

2.3 Algorithm Used

The project implements several fundamental computer graphics algorithms and techniques, integrating rasterization algorithms and standard graphics methods to render complex 3D scenes with precision and efficiency.

2.3.1. Midpoint Circle Algorithm

Purpose: The Midpoint Circle Algorithm is used to efficiently draw circular elements in the scene, such as clock faces or decorative features.

How it works: This algorithm simplifies the process of drawing circles by using integer addition and subtraction instead of more computationally expensive trigonometric calculations or square roots. This results in a faster and more efficient rendering process, especially for simple geometric shapes like circles.

Details: It calculates a decision parameter, typically expressed as $d = 3 - 2 * r$, where r is the radius. Based on this parameter, the algorithm determines the optimal placement of the next pixel, choosing either to move horizontally (East) or diagonally (South-East) to preserve the circle's shape. The algorithm leverages 8-way symmetry to calculate the circle's points in one octant and mirrors them across the remaining seven sections, ensuring the circle appears smooth and symmetrical.

2.3.2. Bresenham's Line Algorithm

Purpose: This algorithm is applied to draw straight lines, which is essential for creating precise details such as wireframes or geometric elements like the hands of a clock.

How it works: Bresenham's Line Algorithm optimizes line drawing by determining which pixels should be selected to form a straight line between two given points. The algorithm accumulates an error value to decide when to step to the next Y-coordinate as it moves along the X-axis. This method allows for pixel-perfect accuracy when rendering straight lines.

Details: The algorithm calculates the deltas (dx, dy) between the start and end points of the line. Using a decision variable p, the algorithm determines whether to remain horizontal or move diagonally upwards, ensuring that the line appears smooth and continuous without visual artifacts.

2.3.3. Procedural Generation (Sine Wave)

Purpose: The procedural sine wave generation technique is used to simulate natural, flowing effects, such as the appearance of fabric hanging from a rod or the undulating motion of curtains.

How it works: By modifying the Z-depth (the "in-out" position) of the curtain as it is drawn, the sine wave function generates a ripple effect that mimics the folding of fabric. The Z-depth is adjusted using a sine function (sin(i * 1.5f)), which creates a wavy motion along the curtain's length, adding a dynamic, realistic effect to the scene.

Details: As the curtain is drawn from left to right, each segment's Z-depth is adjusted according to the sine wave, resulting in a natural-looking curve or ripple. This technique brings a touch of realism to the curtains, making them appear as though they are being influenced by gravity or air movement.

2.3.4. Procedural Texturing (Checkerboard)

Purpose: The checkerboard algorithm is employed to create a tiled floor pattern without the need for external image assets, using procedural generation instead.

How it works: The algorithm applies the modulo operator to the coordinates of each tile, alternating between two colors based on whether the sum of the X and Z coordinates is even or odd. This creates a seamless checkerboard pattern that can be used for floors, walls, or other repeating patterns within the scene.

Details: The pattern alternates between two colors, light gray and medium gray, by calculating (x + z) % 2. If the result is zero (even), one color is applied; if it's one (odd), the other color is chosen. This technique efficiently generates the desired pattern without the need for complex texture mapping or external resources.

These algorithms were chosen due to their efficiency and ability to generate high-quality visual results with minimal computational cost.

2.4 Transformations Applied

Various transformations are applied to different objects to achieve realistic positioning, scaling, and rotation within the 3D scene. Below is a breakdown of the transformations applied to the objects:

- **Room and Furniture**:
    - **Translation**: Used to move objects into position, such as placing the bed, wardrobe, and table in their respective locations in the scene (glTranslatef).
    - **Scaling**: To adjust the size of objects, for example, scaling the bed frame, pillows, and the bookshelf to fit into the room's proportions (glScalef).
    - **Rotation**: Applied to objects like the fan, door, and window to simulate interactions. For example, the fan blades rotate by incrementing the fanAngle to simulate motion.
- **Fan**: The rotation of the fan is achieved by modifying the fanAngle and applying it to the fan blades (glRotatef).

- **Curtains**: The curtains use sinusoidal displacement (sin(i * 1.5f)) along the z-axis to simulate a natural cloth-like movement. This transformation creates the "wave" effect as the curtains move.
- **Lighting Effects**: Transformations are used to position light sources in the scene. For example, glLightfv(GL_LIGHT0, GL_POSITION, lightPos) places the main light at the ceiling, and additional lights for lamps and scones are placed relative to their respective objects.
- **Objects like TV, Fan, and Door**:
    - **Rotation**: The TV screen, door, and window have controlled rotations based on user input. For example, the door angle is updated by incrementing or decrementing the doorAngle variable, affecting its opening/closing behavior (glRotatef).

2.5 Animation Logic

Animation is primarily driven by user input and time-based updates. The logic controlling the animations is as follows:

- **Fan Rotation**:
    - The fan’s rotation is animated by incrementing the fanAngle every frame if the fan is turned on (isFanOn flag). The update function continuously updates the angle, and glRotatef is used to apply this rotation to the fan blades.
- **Day/Night Cycle**:
    - The lighting intensity changes based on the isNight flag. During the night, the lighting is dimmed, and the scene's background color is adjusted to a darker tone, while during the day, the lighting is brighter, and the background color is adjusted to a lighter tone.
- **Interactive Object Animations**:
    - **TV Channel Change**: The tvAnimTime variable is used to animate the TV screen's pattern. The channel changes every time the user presses the 'C' key, cycling through five different patterns on the TV. The animation is rendered by changing the texture or color of the TV screen based on the current channel.
- **Window and Door Animations**:
    - Both the window and door animations are controlled by the windowAngle and doorAngle variables. These values are incremented or decremented gradually (based on the open/close state) to simulate the smooth opening and closing of the window and door. The glRotatef function is applied to achieve the desired effect.
- **Interactive Lights**:
    - The lamps and other lights in the scene are turned on or off by toggling their respective flags (isLampOn, isLEDOn, isTVOn, etc.). The lighting settings are updated accordingly, and the scene is rendered with different lighting effects depending on the state of the light sources.
- **Camera Movement**:
    - The camera can be rotated and moved forward/backward using the arrow keys. This is handled by updating the camera's position (camX, camZ) and direction (lx, lz) based on the key presses. The glLookAt function is used to keep the camera's orientation aligned with the direction.

This logic ensures a dynamic and interactive experience, with real-time changes to the scene based on user input and time-driven events like the Day/Night cycle.

Chapter 3

Implementation and Results

This chapter outlines the detailed implementation process of the Computer Graphics project, which simulates a modern 3D bedroom interior using OpenGL. The project includes the development of objects, the application of algorithms, and the integration of animations. Key features such as transformations and interactive visual outputs are also discussed.

3.1 Implementation

The implementation of the 3D bedroom scene is divided into various components, including object creation, transformation application, animation integration, and lighting setup. The steps involved are as follows:

1. **Object Creation**:
   - Each object in the scene (walls, furniture, windows, door, etc.) was modeled using basic geometric shapes such as cubes, spheres, and cones, which were scaled, rotated, and translated to form the desired shapes.
   - For instance, the **bed** was created with a combination of cubes for the frame, mattress, and pillows. The **wardrobe** was modeled using cuboid transformations to simulate the structure and doors, while the **fan** was built using a combination of spheres for the motor and cubes for the blades.
2. **Textures and Materials**:
   - Textures were applied to various objects to achieve realistic finishes. For example, textures were applied to the **wall** and **photo frame**, and specific materials were used for wood, fabric, and metal elements in the room.
   - The loadBMPTexture function was used to load and map the textures to the objects, providing a more realistic visual representation.
3. **Transformation Application**:
   - **Translation**: This was used to place objects at specific locations in the 3D space. For example, the **wardrobe** and **bed** were positioned relative to the room's layout.
   - **Scaling**: The size of objects was adjusted to ensure proper proportion relative to the room's dimensions. The **TV**, **fan**, and **desk** were all scaled appropriately to fit their respective locations.
   - **Rotation**: Rotation was applied to interactive objects such as the **fan**, **door**, and **window**. The fan blades were animated to rotate, and the door and window angles were adjusted based on user input.
4. **Lighting and Material Properties**:
   - The scene uses multiple light sources. A **main room light** (GL_LIGHT0) provides general illumination, while **LED lights** and **wall sconces** are used to enhance the scene's ambiance.
   - The **day/night cycle** is implemented by changing the intensity and color of the lighting based on the isNight flag. The room transitions from a bright daytime setting to a dimmed, cooler nighttime effect.
   - Materials were applied to the objects to control their reflectivity, and ambient, diffuse, and specular properties were adjusted to achieve realistic visual effects.
5. **Animation Logic**:
   - **Fan Rotation**: The fan's blades rotate continuously when activated. The fan's rotation is controlled by the fanAngle variable, which is updated every frame in the update function.

    - **Window and Door Opening**: The window and door can be opened or closed smoothly, with their rotation angles adjusted based on user input. The doorAngle and windowAngle variables are incremented/decremented in the animation loop to simulate gradual opening and closing.
    - **TV Channel Animation**: The **TV** screen changes its pattern based on the active channel, and different colors or textures are displayed depending on the selected channel. This change is handled in the update function, where the tvAnimTime variable ensures a seamless transition between the TV patterns.
6. **Camera and Interaction**:
    - The user can navigate the scene using a **360-degree camera system**. The glutSpecialFunc handles user input from the arrow keys to rotate and move the camera in the 3D space.
    - The **keyboard controls** allow the user to toggle various interactive elements, such as turning on/off the **lamp**, **fan**, **TV**, and **PC**. Additionally, the user can open/close the **window** and **door**, as well as switch between **day and night** modes.

3.2 Output

The output of this project is a highly interactive 3D simulation of a modern bedroom interior, created using OpenGL. The main visual components and features include:

1. **3D Room Layout**:
   The room is fully furnished with a **bed**, **wardrobe**, **bookshelf**, **coffee table**, **nightstand**, **desk**, and other decor items. The layout provides a realistic simulation of a bedroom, complete with textures and lighting that enhance the overall appearance.

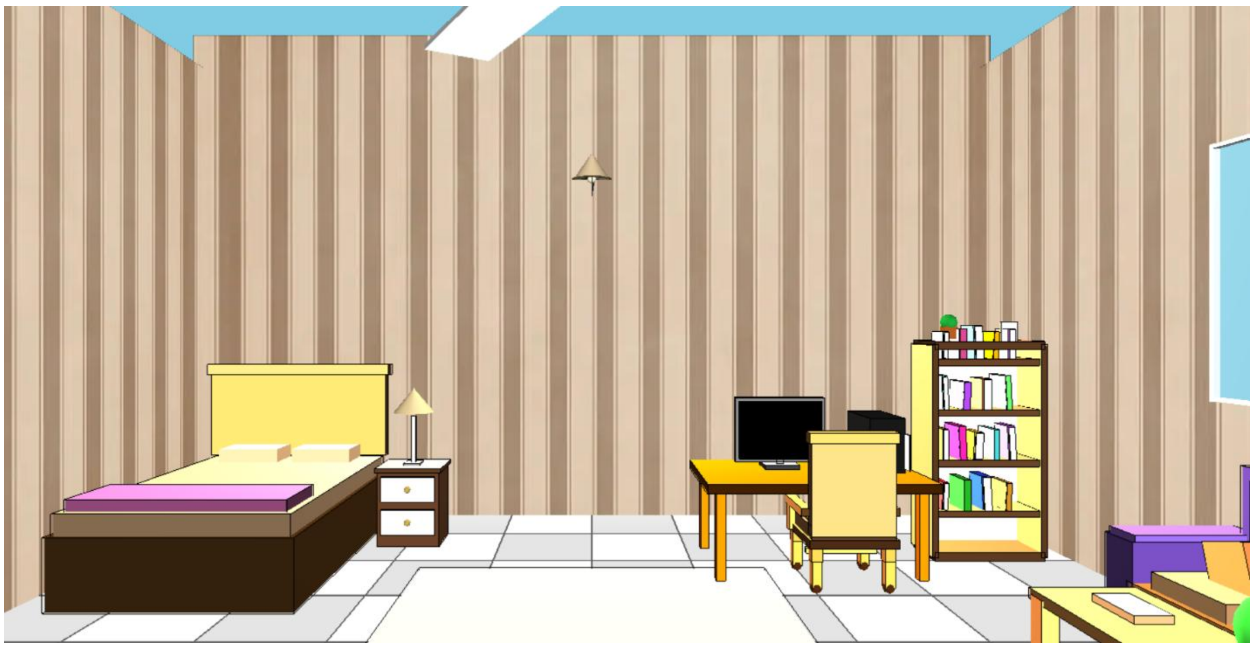

Figure 3.1: 3D Room Layout (Day Mode).

2. **Lighting Effects**:
   The lighting changes dynamically with the **day/night cycle**, creating two distinct moods. The daytime lighting is bright and white, while nighttime lighting features softer, dimmer tones. The **LED lights** also light up as the room transitions to nighttime.

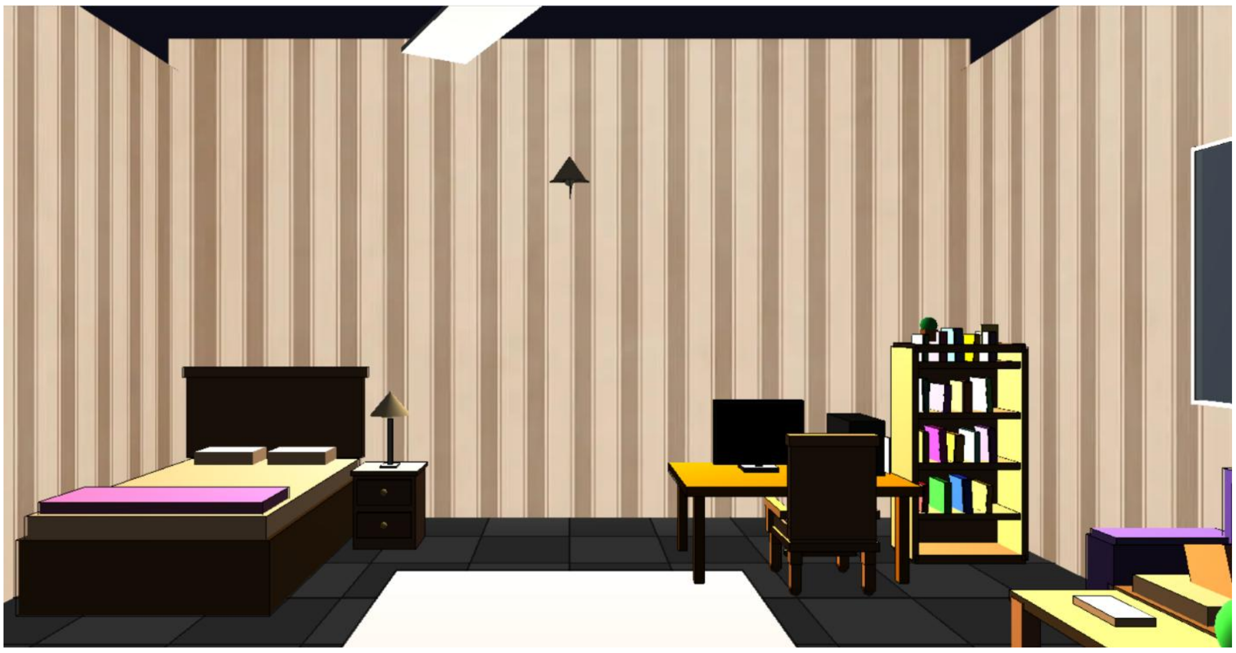

Figure 3.2: 3D Room Layout (Night mode).

3. **Interactive Objects**:
    - The **fan** can be turned on and rotated to simulate motion.
    - The **window** and **door** can be opened and closed using keyboard controls, with smooth animation.
    - The **TV** displays various patterns based on the selected channel. The display changes color when toggling between different channels.
    -
    -
    -
    - The **lamp** and other lights in the room can be switched on and off, adding to the ambiance.

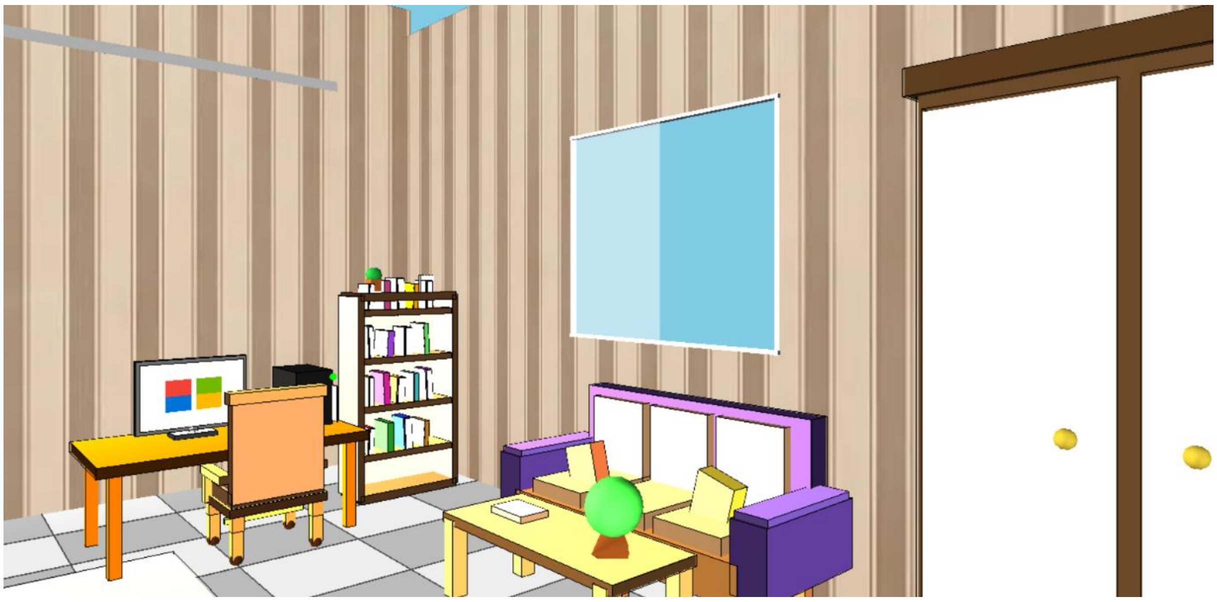

Figure 3.2: Opened Window and turned on PC.

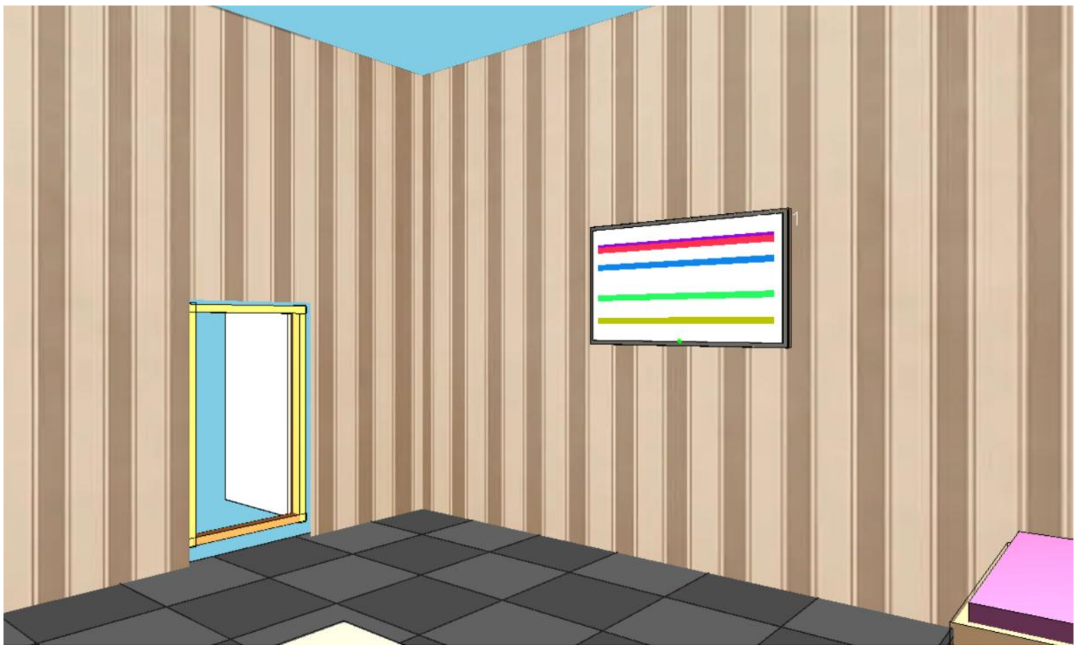

Figure 3.3: Opened Door and TV Turned on.

4. **Camera Movement**:
   The **360-degree camera** allows the user to freely explore the room from any angle. The camera can move in all directions, providing an immersive experience where the user can interact with the room's elements.
5. **Realistic Textures and Materials**:
   Textures applied to walls, furniture, and other objects give the scene a realistic look. The **wooden textures** on furniture and the **fabric textures** on the curtains provide detailed material representation.

3.3 Discussion

The implementation successfully creates a 3D bedroom interior with interactive elements and realistic animations. Key observations include:

- **Realism through Textures**: Textures enhance the scene's realism, though some, like the curtains and bed linens, could be further refined for a more photorealistic appearance.
- **Interactive Animations**: Smooth integration of fan rotation, door/window movements, and TV channel switching ensures a dynamic and responsive user experience.
- **Lighting**: The day/night cycle adds depth to the scene, but fine-tuning the light intensity and colors could improve the ambiance.
- **Performance**: The project runs smoothly, though performance may drop on lower-end hardware due to complex transformations and animations.

The implementation successfully creates a 3D bedroom interior with interactive elements and realistic animations. Textures enhance the scene's realism, though some, like the curtains and bed linens, could benefit from refinement. Interactive animations, such as the fan rotation and door/window movements, provide a dynamic user experience with smooth transitions. The day/night lighting cycle adds depth, but further adjustments to light intensity and colors could improve ambiance. While performance is generally smooth, lower-end hardware may experience slight degradation with complex animations. Future enhancements could include refining textures, adding sound effects for a more immersive experience, and expanding interactive elements like customizable furniture and wallpapers.

Chapter 4

Engineering Standards and Mapping

This chapter discusses the engineering standards followed throughout the project and maps the project activities to the course and program outcomes. It highlights how the project meets academic and professional requirements, with a focus on the impact of the project on society, the environment, and sustainability.

4.1 Impact on Society, Environment and Sustainability

The development of this Computer Graphics project aims to not only fulfill academic requirements but also address critical social and environmental considerations. By focusing on sustainable practices, ethical considerations, and the potential impact on society, the project ensures that it aligns with broader engineering goals, including professionalism and responsible content creation.

4.1.1 Impact on Life

The project creates an interior design simulation through real-time rendering and animation, offering practical applications in fields such as architecture, interior design, and home planning. By providing a tool that allows users to visualize and interact with different design elements, the project enhances how professionals and homeowners approach home improvement and interior design.

For example, interior designers can use the application to showcase various furniture arrangements and decoration styles, providing clients with realistic previews of potential designs before physical alterations are made. The technology also helps in reducing material wastage by allowing designers to experiment with layouts without the need for costly physical prototypes. This can significantly reduce the ecological footprint of interior design projects.

Additionally, by creating a user-friendly and interactive visual tool, the project empowers individuals without specialized knowledge of design to explore and make informed decisions about their home environments. This leads to greater accessibility and inclusivity in design processes.

4.1.2 Impact on Society & Environment

This project has a positive societal and environmental impact by promoting sustainable and efficient practices in interior design. By allowing users to digitally experiment with different layouts and furniture choices, it reduces the need for physical samples and prototypes, which in turn leads to less waste. The simulation also contributes to the more effective use of resources, as it encourages optimal arrangements that can minimize material costs and energy usage in real-life spaces.

From a societal perspective, the project enables a broader audience to access design tools, promoting democratization in a traditionally exclusive field. This can result in more diverse perspectives and creativity in design practices, encouraging innovation and fostering an inclusive environment.

Moreover, the project could support future developments that integrate energy-efficient designs, such as smart homes and sustainability-focused interiors. These innovations are crucial as society continues to face growing concerns about climate change and environmental degradation. The project serves as a stepping stone toward a more eco-conscious future in design.

4.1.3 Ethical Aspects

In this Computer Graphics project, ethical principles are strictly adhered to, ensuring that all design and implementation processes are conducted with integrity, transparency, and respect for intellectual property. Key ethical aspects include:

- **Originality and Plagiarism-Free Design**: All graphical scenes, animations, textures, and visual elements used in the project are either self-created or properly credited to their respective sources. This ensures that the project does not infringe upon the intellectual property rights of others, maintaining academic honesty.
- **Respect for Cultural Sensitivity**: Efforts were made to ensure that the visual elements in the scene, including textures, objects, and animations, do not convey offensive or culturally insensitive messages. The content adheres to ethical guidelines, promoting a positive and respectful visual experience for users from diverse cultural backgrounds.
- **Privacy and Data Protection**: Although this project does not collect personal user data, ethical considerations regarding user privacy in the design of interactive applications are a priority. The project does not include features that may inadvertently capture or misuse user data, such as tracking of personal preferences or behavior without consent.

By following these ethical principles, the project not only meets academic standards but also ensures that it aligns with the core values of professionalism, responsibility, and respect for intellectual and cultural diversity.

4.1.4 Sustainability Plan

To ensure the long-term sustainability of the project, several strategies have been implemented:

- **Efficient Code and Resource Management**: The project’s design has been optimized for performance, ensuring that the rendering and animation processes are efficient. This reduces the computational resources needed to run the project, which, in turn, minimizes its environmental impact by lowering energy consumption.
- **Future Scalability and Adaptability**: The code structure has been built with future scalability in mind. As advancements in technology occur, the project is designed to easily integrate new algorithms, tools, or features that can further enhance the sustainability and relevance of the application. For instance, incorporating newer energy-efficient rendering techniques or integrating with smart home technologies could extend the project’s impact in creating eco-friendly designs.
- **Educational Outreach**: The project’s sustainability plan includes making it accessible to educational institutions, where it can be used as a learning tool for students studying computer graphics, design, and architecture. This ensures that future generations of engineers and designers will be able to learn from and contribute to sustainable practices in design and technology.
- **Ongoing Maintenance and Updates**: To ensure the project's longevity, a plan for ongoing maintenance and updates has been set in place. This includes addressing any bugs or issues as they arise, as well as introducing new features to keep the project relevant and aligned with the latest technological trends in graphics and design.

Through these strategies, the project aligns with sustainable development goals, ensuring that it not only meets immediate needs but also contributes to the long-term well-being of society and the environment.

4.2 Complex Engineering Problem

4.2.1 Mapping of Program Outcome

Table 4.1: Justification of Program Outcomes

POs	Justification of Mapping (Project Perspective)
PO1	This project applies mathematical and computational knowledge to create and transform graphical objects using OpenGL. It integrates geometry, coordinate systems, and matrix operations to solve complex visualization and animation problems in computer graphics. (PO1).
PO5	The project incorporates modern computer graphics tools and technologies such as OpenGL, FreeGLUT, and C++ programming to ensure the application of contemporary engineering practices and visualization techniques in real-world scenarios.
PO9	Through collaborative project work, every member was engaged in teamwork, task allocation, and documentation. (PO9).
PO10	The project requires presenting the developed database system based on the requirements gathered, preparing technical reports, and communicating ideas clearly. (PO10)

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
✓		✓		✓		

EP1: This project involves designing, implementing, and rendering intricate 3D graphical scenes using OpenGL. It requires the application of computer graphics algorithms, transformation techniques, and animation logic, which enhances our analytical and problem-solving abilities in visualization and rendering processes.

EP3: In this project, we analyze the structure of objects, plan transformation sequences, and synchronize animation timing to create smooth and realistic motion. By breaking down complex scenes into manageable components and analyzing their interactions, we enhance our skills in formulating and solving graphical challenges effectively.

EP5: Through the application of graphics algorithms and transformation techniques such as DDA, Bresenham, and matrix operations, we work towards optimized rendering. This hands-on approach strengthens our ability to conduct thorough investigations and apply established graphics methodologies and tools to solve practical problems.

4.2.3 Complex Engineering Activities

Table 4.2: Mapping with complex engineering activities.

EA1	EA2	EA3	EA4	EA5
✓	✓			

EA1: In this Computer Graphics project, we employ systematic methods to design and document all graphical components, algorithms, and transformations. The project's structure, including rendering logic and animation workflow, is clearly organized in reports and presentations, ensuring that technical details are presented logically and are easily comprehended by the audience.

EA2: For this project, we utilize oral and visual presentation techniques to explain complex computer graphics concepts such as transformations, animation logic, and rendering processes. By showcasing the project through demonstrations and slides, we improve our communication skills, effectively convey design decisions, and confidently respond to technical inquiries.

Chapter 5

Conclusion

5.1 Summary

This Computer Graphics project aimed to design, implement, and render dynamic 3D scenes using OpenGL. By applying various algorithms such as Bresenham and Midpoint Circle, we were able to develop a system that allows for realistic object rendering and animation. The project incorporated the use of advanced transformations (translation, rotation, scaling, reflection, and shear) to manipulate objects in the 3D space, and animation techniques to enable smooth motion. Through this, we gained hands-on experience with computer graphics tools and algorithms, as well as problem-solving and programming skills that are essential in the field of computer graphics.

5.2 Limitation

While the project successfully met its core objectives, there were a few limitations encountered. One major limitation was the rendering performance, especially with complex scenes involving multiple moving objects. Although OpenGL and FreeGLUT provided a solid foundation, real-time rendering could still be optimized further, particularly with more intricate object interactions and higher-level graphical effects. Additionally, due to time constraints, the scope of the project was limited to basic geometric shapes and simple transformations. Further refinement in object detail and animation complexity could significantly improve visual fidelity.

5.3 Future Work

Looking ahead, there are several directions in which the project can be expanded and improved. Future work could include:

- **Optimization**: Enhance the rendering performance by integrating advanced techniques like shaders, GPU acceleration, and multi-threading to handle more complex scenes efficiently.
- **Realistic Texturing**: Incorporating textures and materials to add realism to the objects in the scene, including bump mapping and reflective surfaces.
- **Advanced Animation**: Developing more complex animation systems, such as character animation or physics-based object movement, could improve the visual dynamics of the scene.
- **Interactive Elements**: Adding user input and interaction, such as object manipulation using keyboard and mouse, would make the scene more immersive and dynamic.
- **Expansion to 3D Environments**: Extending the project from basic 3D shapes to more complex environments, including lighting, shadows, and camera controls, would provide a richer graphical experience.

These improvements would allow the project to scale and provide an even more robust demonstration of computer graphics techniques.

References

[1] M. Woo, J. Neider, and D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, 8th ed., Addison-Wesley, 2013.

[2] "OpenGL Tutorial," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/opengl-tutorial/. [Accessed: Dec. 2025].

[3] Stack Overflow, "Animation logic using OpenGL," Stack Overflow. [Online]. Available: https://stackoverflow.com/questions/your-query. [Accessed: Dec. 2025].

[4] W. J. Schmidt, *Computer Graphics: Principles and Practice*, 3rd ed., Pearson Education, 2009.

[5] C. H. Lee, *Introduction to Computer Graphics*, 2nd ed., McGraw-Hill, 2016.

[6] S. Marschner and P. Shirley, *Fundamentals of Computer Graphics*, 4th ed., CRC Press, 2016.

[7] F. S. Hill, *Computer Graphics Using OpenGL*, 4th ed., Pearson Education, 2016.

[8] J. E. Hershberger, "DDA and Bresenham's algorithms," *Journal of Computer Graphics Techniques*, vol. 9, no. 2, pp. 45-50, May 2017.

[9] M. H. Goldstein, "The Midpoint Circle Algorithm," *OpenGL Discussion Forums*, [Online]. Available: https://www.opengl.org/discussion_forums. [Accessed: Dec. 2025].

[10] "GLFW - OpenGL framework," GLFW. [Online]. Available: https://www.glfw.org. [Accessed: Dec. 2025].