CSE410 - Computer Graphics Sessional

OpenGL Assignment

Deadline: 02 May 2025, 11:59 PM

Overview

This assignment consists of the following tasks:

- **Task 1**: Implementing a fully controllable camera in 3D space.
- **Task 2**: Creating a real-time analog clock.
- **Task 3**: Simulating a bouncing ball in a cube with gravity, collisions, and rolling behavior.

You must recreate the behavior demonstrated in the provided reference output files.

Task 1: Camera Movement

Implement a camera that can **translate** and **rotate** freely in 3D space.

Translation Controls

- **Up arrow**: Move forward
- **Down arrow**: Move backward
- **Left arrow**: Move left
- **Right arrow**: Move right
- **Page Up**: Move upward
- **Page Down**: Move downward

Rotation Controls

- **1**: Look left (Yaw)
- **2**: Look right (Yaw)
- **3**: Look up (Pitch)
- **4**: Look down (Pitch)
- **5**: Tilt clockwise (Roll)
- **6**: Tilt counterclockwise (Roll)
- **w**: Move upward without changing reference point
- **s**: Move downward without changing reference point

Demo

Check the **balldemo** executable from "Demo Executables" folder on Moodle for reference.

Task 2: Real-Time Analog Clock

Implement a real-time analog clock showing hour, minute, and second hands.

Description

- Draw a circular clock face with hour and minute markers.
- Draw three hands: Hour (short, thick), Minute (long, medium), Second (thin, long).
- Animate the clock hands based on system time.
- Clock should update in real-time.

Demo

Check the **clockdemo** executable from "Demo Executables" folder on Moodle for reference.

Task 3: 3D Bouncing Ball Simulation

Simulate a ball bouncing inside a cube under gravity, with realistic rolling and collision handling.

Visual Elements

- 3D Ball
- Checkered floor
- Colored cube walls (sides and ceiling)
- Velocity direction arrow (toggleable)

Physics Simulation

- Gravity acting downward (9.8 m/s^2)
- Bounce on collisions with restitution coefficient (0.7∼0.8) for damping effect
- Ball comes to rest if vertical velocity is too low
- Rolling rotation based on displacement
- Proper spinning behavior for the ball
- You do not necessarily need to implement friction mechanism for the rolling ball

Simulation Controls

- **Space**: Toggle simulation on/off
- **r**: Reset ball position and velocity (randomly, but the velocity direction should be upward)
- **+/-**: Increase or decrease initial speed (only applicable after reset and before release)
- **v**: Toggle velocity arrow

Demo

Check the **balldemo** executable from "Demo Executables" folder on Moodle for reference.

Submission Instructions

Important Submission Guidelines

1. Create a directory named with your 7-digit student ID.
2. Add all relevant source files.
3. Zip the directory (`.zip` format only).
4. Upload to Moodle before the deadline.

Special Instructions

- **No plagiarism**: Strictly enforced.
- **Use variables**: Avoid hardcoding.
- **Clean code**: Structure your program properly.
- **Incremental workflow**: Keep adding components gradually.

Mark Distribution

Task	Subtask Description	Subtask Mark	Task Mark
Task 1	Camera Movement	20	20
Task 2	Drawing clock face (markers)	10	30
	Drawing and updating hands correctly	10	
	Smooth animation using system time	10	
Task 3	Scene rendering (cube, floor, ball visuals)	10	50
	Resetting and control functionality	10	
	Proper gravity and collision handling	15	
	Ball realistic rolling and bouncing	15	

Total: 100 Marks