3D Game Animation

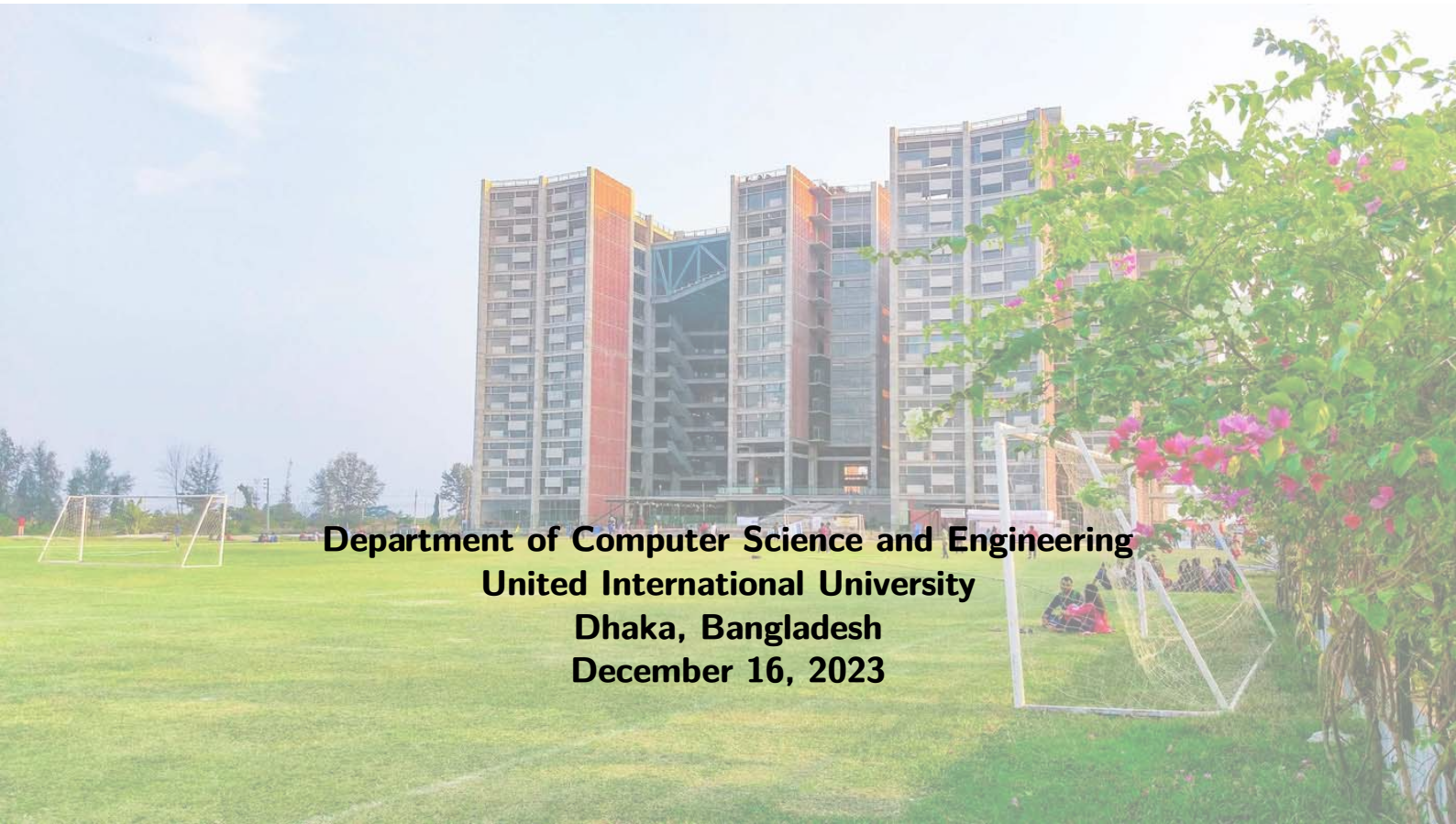
Report/Assignment No. 2

CSI422: Computer Graphics Lab

Group Information

Student Name	Student ID
Iftekhar Mahmud	011182073
Md. Abid Hasan Saikot	011182064

Course Teacher: Md Hasan Al Kayem, Lecturer



Department of Computer Science and Engineering
United International University
Dhaka, Bangladesh

December 16, 2023

1. Introduction

The project focused on utilizing simple geometric shapes to design and animate a cricket player's movements within a 3D environment. The animation was implemented using OpenGL, leveraging its capabilities to render and animate 3D objects.

2. Animation Description

The animation depicts a cricket player engaging in the act of playing the ball. It begins with the cricket player in a static position, ready to hit the ball. The player's body is represented using cylinders, cones, and spheres, creatively arranged to illustrate the various body parts.

The animation sequence starts when the player initiates the swing. The limbs are manipulated using translation and rotation transformations to simulate the realistic movement of swinging the bat. The cone-shaped bat is shown to intersect with the spherical ball, illustrating the action of hitting the ball.

The player's movements are synchronized throughout the animation to convey the fluidity and coherence expected in a cricketing action. Keyframes and time-based calculations ensure a smooth and natural animation sequence.

3. Operation Instruction

Starting the Animation:

- Open the application or run the program that executes the cricket player animation.
- A window displays the 3D environment with the cricket player model and the ball.

Interacting with the Animation:

Use the keyboard controls defined in the application to navigate and view the animation:

- Press the '1' and '2' keys to adjust the viewing position along the X-axis.
- Press the '3' and '4' keys to adjust the viewing position along the Y-axis.
- Press the '5' and '6' keys to adjust the viewing position along the Z-axis.

Observing the Animation:

As the animation runs, observe the cricket player's movements:

- Notice the player's preparation stance.
- Observe the swing movement using cylinders and cones to represent arms and the bat.
- Observe the ball represented by a sphere.

Controlling the Animation:

The animation runs continuously, demonstrating the cricket player's action.

Exiting the Animation:

Close the application or terminate the program to end the animation session.

4. Code Explanation

Header Files and Constants:

The code includes necessary header files like <windows.h>, <GL/glut.h>, and defines constants like pi and other initializations for mathematical calculations.

Structures:

Defines Point3D and Point13D structures to represent points in 3D space.

Variables and Initializations:

Initializes variables like eye, pos, eye1, pos1, and other parameters related to the animation setup.

Utility Functions:

Includes utility functions for drawing axes, grids, and basic shapes like circles, grids, cuboids, cylinders, cones, and spheres.

Rendering Functions:

drawPlayer() and drawBall() functions define the rendering of the cricket player and the ball using combinations of shapes like cylinders, cones, and spheres.

Display Function:

display() function configures the OpenGL display settings, sets up the camera, and renders the player and ball by calling corresponding drawing functions.

Animation Logic:

- animate() function implements the animation logic by calculating the position of the ball based on projectile motion equations.
- It updates the ball's position and triggers a redisplay using glutPostRedisplay() to render the updated scene.

Initialization and Input Handling:

- init() sets up the projection matrix and initializes the OpenGL environment.
- keyboard_action() handles user keyboard input to interact with the animation, allowing movement along the X, Y, and Z axes.

Main Function and Application Loop:

main() initializes the OpenGL window, sets up the callbacks, and enters the main loop for the application.

OpenGL Functions:

Throughout the code, OpenGL functions like glBegin(), glEnd(), glVertex(), glTranslated(), and glRotated() are used to define shapes, and transformations, and render the scene.

Comments and Structure:

The code contains comments for better readability, explaining the purpose and functionality of various sections.

5. Github Link for the project :

https://github.com/iftekhar-mahmud/3D-Animation