



# AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

**Course:**

COMPUTER GRAPHICS

**Project Title:**

**FLAPPY BIRD**

Group No.10

Summer 2024-2025

Section: L

**Submitted to**

Noboranjana Dey

**Submitted By**

NAME	ID
Efty, Md. Emran Nazir	22-47802-2
Sudipto Mondal	23-50021-1
Mist Sayma Akter	23-52022-2

**Date of Submission:**

September 15th, 2025

## 1. Introduction

We implemented this game to create a basic yet fully functional Flappy Bird clone using OpenGL and the GLUT library. This project serves as a hands-on exercise in computer graphics, demonstrating how to use low-level APIs to build a 2D interactive application. It allowed us to explore fundamental concepts like rendering primitives, managing game state, and handling real-time user input in a familiar and engaging context.

The motivation behind this project was to provide a practical example of computer graphics principles in action. While modern game engines abstract away many complexities, this project forces a deeper understanding of how graphics are rendered, how objects are transformed, and how game loops function. It is a valuable learning tool for anyone looking to transition from theoretical knowledge to practical application in graphics programming.

The significance of the project is that it bridges the gap between theory and practice in computer graphics and game development. By building a complete, playable game from scratch, we gain a comprehensive understanding of the entire development pipeline. This includes setting up the rendering environment, creating and animating game objects, implementing physics, and handling user interactions and collision detection. The code serves as a clean, commented reference for future projects.

The target audience for this project is primarily students and hobbyists interested in computer graphics, game development, or C++ programming. It's particularly useful for those who want to learn how graphics software works at a foundational level without the steep learning curve of a complex engine. It provides a simple but complete case study that can be studied, modified, and built upon.

## 2. Tools Used in the Project

The project uses the following tools and techniques:

- **OpenGL / GLUT:** For rendering 2D graphics, drawing objects, and handling window management.
- **glBegin()/glEnd() and glVertex2f():** Used to draw shapes like quads, triangles, and circles.
- **glColor3f():** For coloring objects.

- **glVertex2f()**: This specifies a 2D vertex, which is a point in the coordinate system, for drawing shapes.
- **glRasterPos2f() & glutBitmapCharacter()**: To display text such as score, lives, and level.
- **Timer Functions (glutTimerFunc)**: To update game physics and animations periodically.
- **Keyboard Input Handling (glutKeyboardFunc)**: To control the bird and restart the game.
- **Trigonometric Functions (cos, sin)**: To draw circular shapes for bird, clouds, and flowers.
- **Randomization (rand())**: To create dynamic pipe positions, clouds, and rain drops.
- **GluOrtho2D()**: This sets up a 2D orthographic projection, which maps the world coordinates (0 to windowWidth, 0 to windowHeight) directly to the screen pixels, simplifying 2D rendering.
- **Sound API (PlaySound)**: To add jump and hit effects (Windows only).

### 3. Knowledge Applied

#### a) For Job Market Purposes:

This project demonstrates a range of skills highly valued in the tech industry:

- **Foundational Computer Graphics**: Proficient understanding of low-level graphics programming using APIs like OpenGL. This is a direct skill for roles in game development, simulation, and visualization.
- **C++ Programming**: Strong command of C++ concepts, including structs, pointers, and memory management.
- **Game Development Fundamentals**: Experience with core game logic, including event handling, physics simulation (gravity), collision detection, and state management (game over, score, levels).
- **Problem-Solving**: The ability to break down a complex task (creating a game) into smaller, manageable components (drawing a bird, animating pipes, handling input).

- **Object-Oriented Principles:** The code uses structs to organize game objects like Pipe, Cloud, and Flower, showcasing a modular approach to programming.

## b) For Higher Study Purposes

The project serves as a strong foundation for advanced academic work:

- **Real-time Rendering:** It provides a practical understanding of how real-time rendering works, where the scene is redrawn multiple times per second to create the illusion of motion.
- **Physics Simulation:** The implementation of gravity (`birdVelocity += gravity`) and a simple jump mechanic demonstrates basic physics simulation techniques.
- **Computational Geometry:** The collision detection logic between the bird and the pipes involves simple geometric calculations.
- **Procedural Content Generation:** The random generation of pipe positions, cloud locations, and flower styles illustrates a basic form of procedural generation, a key concept in computer graphics and procedural modeling.
- **Human-Computer Interaction (HCI):** The project explores basic HCI principles through a simple, responsive user interface driven by keyboard input.

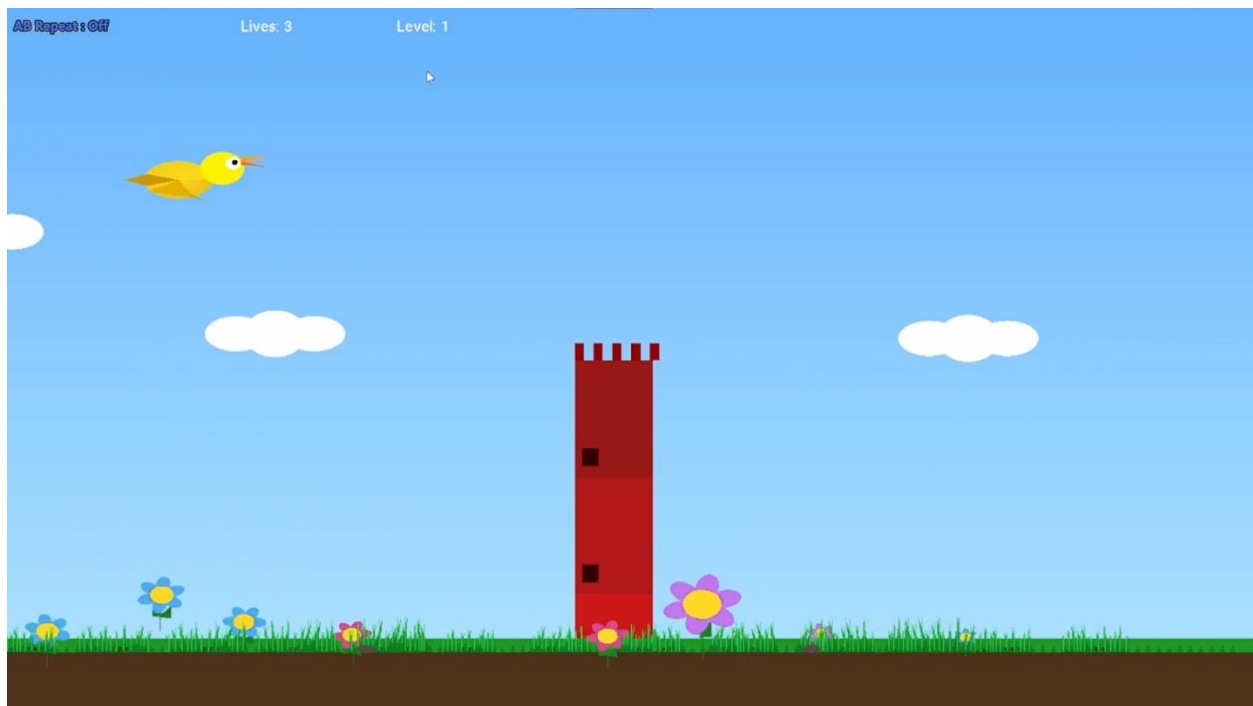
## 4. Reference Materials

- **GitHub Repository:** <https://github.com/mdemrannazirefty/Flappy-Bird-Game-GLUT.git>
- **Graph / Visualization:**
- **YouTube Demonstration:**  
[https://www.youtube.com/watch?v=ZSmQ\\_o1Cbq4](https://www.youtube.com/watch?v=ZSmQ_o1Cbq4)

## 5. Screenshots of The Project



Figure: Level-1



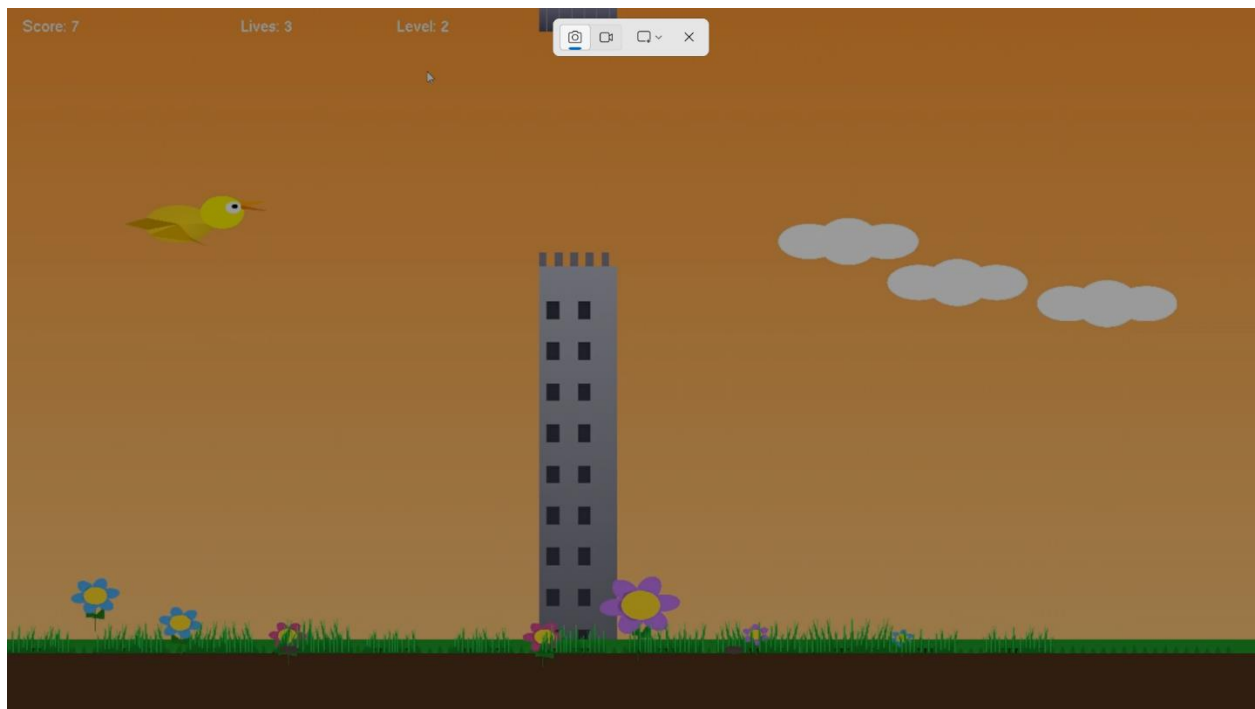


Figure: Level-2

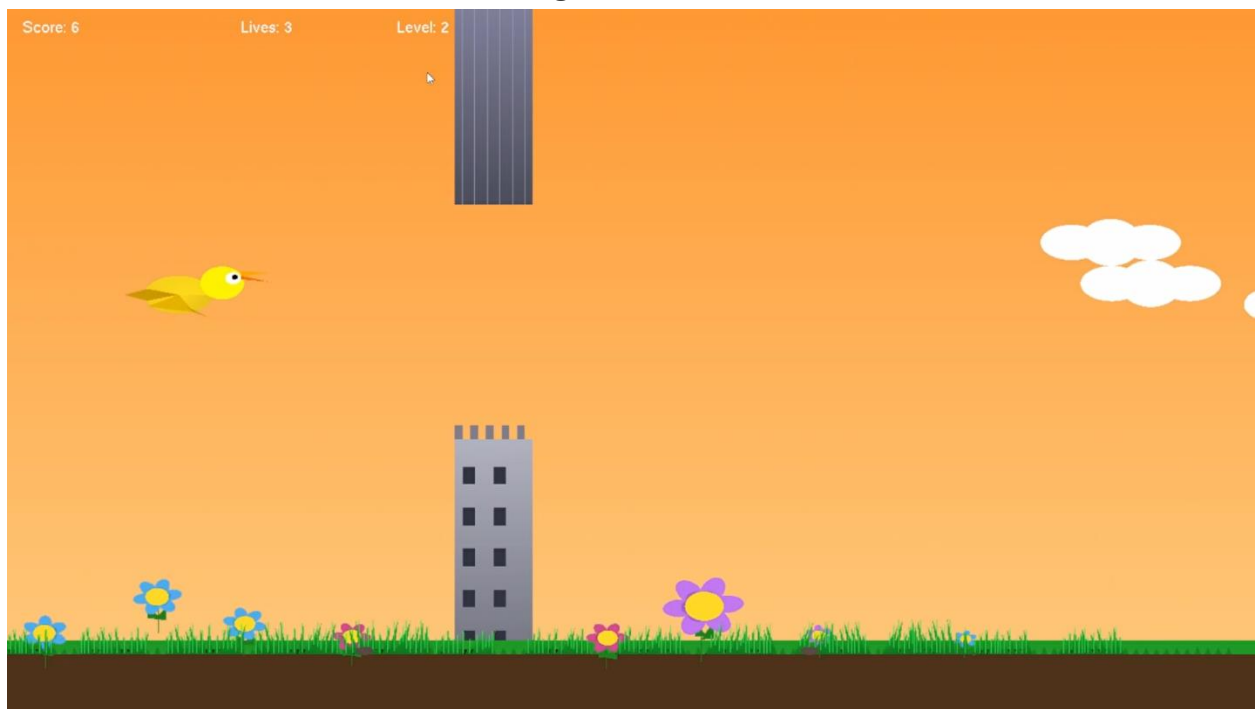




Figure: Level-3

