# UNIVERSITY OF LAYYAH

## PROJECT OVERVIEW

**Project Title :**

FLAPPY BIRD GAME

**Submitted To :**

Sir Faisal Hafeez

**Submitted By :**

Fasihuddin(09)
Ameer Hamza(22)
M .islam (26)
M.Asif(27)

## INTRODUCTION

In the realm of computer programming, languages like C++ and Java have been staples for game development. With their robust features and extensive libraries, these languages provide a solid foundation for creating engaging and interactive games. In this project, we will be utilizing C++ as our primary language, leveraging the SFML (Simple and Fast Multimedia Library) for graphics and game development. We will also explore the use of compilers such as VS Code and Dev C++ to bring our game to life.

## Project Description

The objective of this project is to design and develop a Flappy Bird game, a popular mobile game that challenges players to navigate a bird through obstacles. Our game will feature similar gameplay mechanics, including procedurally generated obstacles, scoring, and collision detection. We will utilize SFML for graphics, sound effects, and game logic, ensuring a smooth and engaging gaming experience.

## Key Features

- Procedurally generated obstacles for endless gameplay
- Scoring system with collision detection
- Simple yet addictive gameplay mechanics
- SFML-powered graphics and sound effects
- Support for multiple platforms (Windows, macOS, Linux)

## Technical Requirements

- Programming language: C++
- Compiler: VS Code, Dev C++
- Graphics library: SFML
- Operating system: Windows, macOS, Linux

## Data Structures:

Here are some data structures that can be used in the game:

**1. Linked List for Obstacles**

- A linked list can be used to manage obstacles (pipes) in the game.
- Each node in the list represents an obstacle, containing attributes like position, size, and velocity.
- New obstacles can be added to the end of the list, and obstacles that are no longer visible can be removed from the list.

**2. Array for Bird's Movement**

- An array can be used to store the bird's movement patterns, such as its position, velocity, and acceleration.
- The array can be updated regularly to reflect changes in the bird's movement.

**3. Stack for Game States**

- A stack can be used to manage game states, such as the game over screen, pause screen, and game play screen.
- Each state can be pushed onto the stack when it becomes active, and popped off the stack when it becomes inactive.

**4. Queue for Scoring**

- A queue can be used to manage scoring, such as storing the points earned by the player.
- Points can be added to the end of the queue, and the total score can be calculated by summing up all the points in the queue.

**5. Hash Table for Collision Detection**

- A hash table can be used to detect collisions between the bird and obstacles.
- The hash table can store the positions of obstacles, and the bird's position can be checked against the hash table to detect collisions.

By the end of this project, we aim to deliver a fully functional Flappy Bird game that showcases our programming skills, creativity, and attention to detail.