Physics Game Engine and Angry Birds Clone in C++

Project Proposal

Sabin Katwal (081BEI029) Safal Neupane (081BEI030)
Sparsha Bhusal (081BEI040)

1. Introduction

This project aims to develop a custom 2D physics game engine from scratch using Object-Oriented Programming (OOP) principles in C++. The engine will simulate basic physics interactions such as gravity, projectile motion, and collisions. A playable *Angry Birds* clone will be developed using this engine to showcase its capabilities. Although no full game engines or external physics libraries will be used, frameworks such as SFML or SDL will be utilized for rendering, input, and audio handling.

2. Objectives

- To implement OOP principles such as encapsulation, inheritance, and modularity.
- To design and build a reusable 2D physics engine from scratch.
- To simulate motion, forces, and collisions using physics fundamentals.
- To build a functional *Angry Birds* style game using the engine.
- To utilize C++ frameworks (e.g., SFML or SDL) for rendering and user interaction.
- To improve understanding of physics, simulation, and modular game development.

3. Scope of the Project

- Implement a 2D vector-based physics simulation system.
- Apply gravity, force, and projectile motion to dynamic bodies.
- Create game elements: birds, blocks, pigs, structures, etc.
- Include mouse-based slingshot launching mechanics.
- Provide real-time visual feedback using a rendering framework (SFML or SDL).
- Design basic levels with destructible environments and scoring.

4. Tools and Technologies

- **Language:** C++
- **Frameworks:** SFML (preferred), SDL, or OpenGL (for graphics, input, sound)
- **Compiler/IDE:** g++, Visual Studio, Code::Blocks, CLion
- **Platform:** Windows / Linux
- **Version Control:** Git (GitHub, GitLab)

5. Methodology

5.1. Class Design

The main classes will be designed as follows:

```
class Vector2D {
private:
    float x, y;
public:
    float magnitude();
    Vector2D normalize();
    Vector2D add(Vector2D v);
};

class RigidBody {
private:
    Vector2D position;
    Vector2D velocity;
    float mass;
public:
    void applyForce(Vector2D force);
    void update(float deltaTime);
};

class PhysicsEngine {
public:
    void simulate(float deltaTime);
    void detectAndResolveCollisions();
};
```

5.2. Functional Modules

- Vector mathematics and physics calculations
- Motion integration and force application
- Collision detection and resolution (bounding box or circle-based)

- Mouse input for launching birds
- Rendering game world using SFML/SDL
- Sound effects and scoring
- Level loading, reset, and game state management

6. Expected Output

- A playable clone of *Angry Birds* with intuitive slingshot controls
- Realistic projectile motion and collision simulation
- Visual effects using a C++ graphics framework
- Sound effects and scoring mechanism
- Fully self-written physics engine (no external physics/game engines)

7. Timeline

Week	**Task Description**
Week 1	Requirements Gathering and Planning
Week 2	Vector Class, Physics Base Classes Design
Week 3	Implement Motion Simulation and Gravity
Week 4	Collision Detection and Response System
Week 5	Game Mechanics: Slingshot, Levels, Score
Week 6	Rendering and Sound Integration using SFML/SDL
Week 7	Final Testing, Polishing, and Report Writing