Problem Statement: Physics Game - Collisions and Gravity

Develop a physics-based gaming application using C++/C# or JavaScript that focuses on realistic collision dynamics and gravitational interactions. The goal is to create an engaging gameplay experience where players can manipulate objects, analyze trajectories, and master the application of physics principles in a challenging gaming environment.

Showcasing:

realistic collision dynamics, gravitational forces, and the innovative incorporation of momentum.

Requirements:

1. **Game Elements:**
    - **Projectile:** Represented as an object (e.g., bird, character) with an initial position, velocity, and mass.
    - **Targets/Obstacles:** Include multiple targets or obstacles with different shapes and sizes.
2. **Physics Simulation:**
    - Implement realistic collision interactions between rigid bodies (Projectile and Targets/Obstacles) based on initial conditions.
    - Integrate gravitational forces that influence the trajectory and motion of the rigid bodies.
    - Consider additional forces (e.g., wind) to introduce complexity and challenge.
    - Provide a physics engine that takes specified inputs and calculates the new configuration of the scene for rigid bodies.
3. **Physics inputs:**
    - Define initial conditions for the scene, assuming rigid bodies. This includes specifying the locations of bodies (Projectile, Targets/Obstacles), their initial velocity, and forces acting upon them.
    - Allow the specification of additional parameters such as launch angle, force, and time step size for simulation accuracy.
4. **Physics Outputs:**
    - Output the updated positions and velocities of the rigid bodies after each simulation step.
    - Visualize the scene's new configuration, considering any changes resulting from rigid body collisions, gravitational effects, or other forces.

5. **Controls:**
    - Enable user input controls for setting initial conditions and parameters.
    - Allow users to specify launch angle, force, and other relevant physics parameters for rigid bodies.
    - Implement controls for adjusting the time step size to control simulation accuracy.
6. **Graphics and Sound:**
    - Utilize graphics to visually represent the physics-based interactions in the game, emphasizing rigid body collisions and movements.
    - Implement sound effects corresponding to key physics events such as rigid body collisions and launches.

Project Goal

The primary ambition of this project is to achieve a breakthrough in physics-based computation within the gaming realm. By focusing on the intricacies of realistic collision dynamics, gravitational interactions, and pioneering the implementation of dynamic momentum disruptions, the project aims to elevate the gaming experience to a level where players actively engage with and manipulate fundamental physics principles. When a bird collides, the goal is to authentically break the gravity momentum, simulating a quantum leap in computational realism. This project aspires to contribute to the advancement of physics understanding in the gaming community, pushing the boundaries of what is achievable in virtual worlds through cutting-edge computational physics.

Key Implementation Guidelines:

- Develop classes and functions to manage physics inputs and outputs for rigid bodies.
- Implement a physics engine that accurately models collision dynamics and gravitational interactions for rigid bodies.
- Provide a user-friendly interface for adjusting physics parameters.
- Utilize graphics libraries or frameworks to visualize the physics-based interactions of rigid bodies.
- Implement sound effects corresponding to key physics events involving rigid bodies.