ANGRY BIRDS GAME

Overview

This project recreates the fun and challenge of the classic Angry Birds game, developed using the LibGDX framework and Box2D physics engine. The goal is to use a slingshot to launch birds and destroy pigs hiding in structures, progressing through increasingly complex levels.

The game focuses on realistic physics, dynamic interactions, and intuitive gameplay mechanics to create an engaging experience.

Features

Birds

The game includes three bird types, each with unique attributes influencing gameplay:

1. **Red Bird**: A balanced bird with medium attack power, ideal for general-purpose destruction.
2. **Black Bird**: The most powerful bird, effective against tough structures like steel.
3. **Yellow Bird**: The least powerful bird, best suited for lighter structures like wood.

- **Bird-Slingshot Relationship**: Birds are launched via the slingshot. The drag-and-release mechanics determine their trajectory and speed.

Pigs

Targets in the game are classified into three types based on durability:

1. **Minion Pig**: Weak, requiring minimal force to eliminate.
2. **Corporal Pig**: Moderate resistance to damage.
3. **King Pig**: Strongest and typically the final challenge.

- **Pig Interactions**: Pigs are destroyed when impacted by birds or collapsing structures, with damage calculated using Box2D's physics engine.

Structures

The game features destructible structures made of materials such as:

- **Wood**: Low durability, easy to destroy.
- **Steel**: High durability, resistant to impacts.
- **Dirt**: Medium durability.

Each material reacts differently to bird collisions, affecting gameplay strategy. Structures are arranged to protect pigs, adding a layer of challenge.

Gameplay Mechanics

1. **Slingshot Mechanics**: The slingshot allows players to adjust the launch angle and power by dragging and releasing the bird. The longer the drag, the more force applied to the bird.
2. **Collision Detection**: Managed by the `GameContactListener` class. Detects when a bird collides with a pig or structure, applying damage based on impact force.
3. **Scoring System**: Points are awarded for:
    - Destroying pigs.
    - Breaking parts of the structure.
    - Bonus points for unused birds.

Levels and Challenges

- The game includes two main levels (`Level1` and `Level2`) with increasing difficulty.
- **Level Design**: Each level introduces new layouts and materials, requiring players to adapt their strategies.
- **Victory/Defeat Conditions**:
    - Victory: All pigs are eliminated.
    - Defeat: All birds are used without eliminating all pigs.

Screens

The game employs multiple screens for a seamless experience:

1. **HomePage**: Displays the game logo, instructions, and a "Play" button.
2. **Level Selection Screen**: Allows players to choose their level.
3. **Game Screen**: Where the core gameplay takes place, including slingshot mechanics and collision physics.
4. **Win/Loss Screens**: Appear based on the outcome of the level, with options to retry or proceed.

Code Structure

Core Components

- **Entities**:
    - `Bird`: Base class for all bird types (`RedBird`, `BlackBird`, `YellowBird`).
    - `Pig`: Represents pigs in the game with health attributes.
    - `Structure`: Defines destructible objects (wood, steel, dirt).
    - `Slingshot`: Manages bird launching mechanics.
- **Game Logic**:
    - `GameContactListener`: Handles interactions between entities (e.g., collisions).

- `LevelManager`: Loads and manages levels, including object placement and win/loss conditions.

- **Game Screens**:
  - `HomePageScreen`: Displays the main menu.
  - `GameScreen`: Contains gameplay logic and rendering.
  - `WinScreen` and `LossScreen`: Handle post-level outcomes.

Assets

- **Textures**: Includes sprites for birds, pigs, and materials.
- **Level Maps**: TMX files defining level layouts.

Technical Details

Technology Stack

1. **LibGDX**: A robust framework for cross-platform game development, providing tools for rendering, input handling, and resource management.
2. **Box2D**: A physics engine used to simulate realistic interactions between game entities.
3. **Asset Management**: Textures and maps are preloaded to ensure smooth gameplay.

Physics Implementation

- Box2D ensures realistic responses for:
  - Bird trajectories (based on launch power and angle).
  - Structural collapses upon impact.
  - Collision effects between birds, pigs, and structures.

How to Play

1. **Setup**:
   - Drag the bird on the slingshot to adjust aim and power.
   - Release to launch the bird toward the pigs.
2. **Objective**:
   - Eliminate all pigs before running out of birds.
3. **Scoring**:
   - Destroy pigs and structures to earn points.
   - Save birds for bonus points.

How to Run

Prerequisites

- Java 8 or higher installed.
- Gradle installed or included in your IDE.

Steps

1. Clone the repository:

   ```
   git clone <repository-url>
   ```

2. Build and run the game using Gradle:

   ```
   ./gradlew desktop:run
   ```

Future Enhancements

- Add new bird types with special abilities.
- Introduce more levels with diverse layouts.
- Implement a leaderboard for high scores.

Contributors

Developed as a physics-based game development project, integrating realistic mechanics and engaging gameplay.

Sidharth Kumar - 2023526

Rishabh Dwivedi - 2023434
