Project Document

Overview:

Functional game recreation of "Angry Birds."

Develop a robust slingshot mechanic to launch birds with varying trajectories.

Create different bird types, each possessing unique abilities or characteristics that influence their flight and impact. In a repo, We created a split bird class, explosive bird class, speed boost bird class, bigger bird class, and vertical moving bird class. Each of them has a unique ability after clicking the mouse.

Design and implement destructible structures that house enemy pigs, ensuring they react realistically to impacts based on their materials and construction. We create stone and wood classes as destructible structures; they have different degrees of hardness.

Develop 3 levels, each offering a unique challenge to the player. Levels are different regarding layout, structure design, and the number and type of enemy pigs.

Introduce various game modes like timed challenges and star collections. Define and implement rules for each game mode. Allow users to select game modes from the level selection screen.

Create a user-friendly interface that allows players to easily check through levels, access game options, and view their progress, scores, and ranking.

Utilize SFML to implement appealing graphics, ensuring that game elements such as birds, pigs, and structures are easily distinguishable. The view follows the bird as it moves sideways.

Incorporate fitting audio elements, including background music, sound effects for various actions like launching birds and structure collapses, star collection, and feedback sounds for successful or failed level attempts. We also have trajectory tracking and explosion animation, which is useful for players to adjust birds' launching positions.

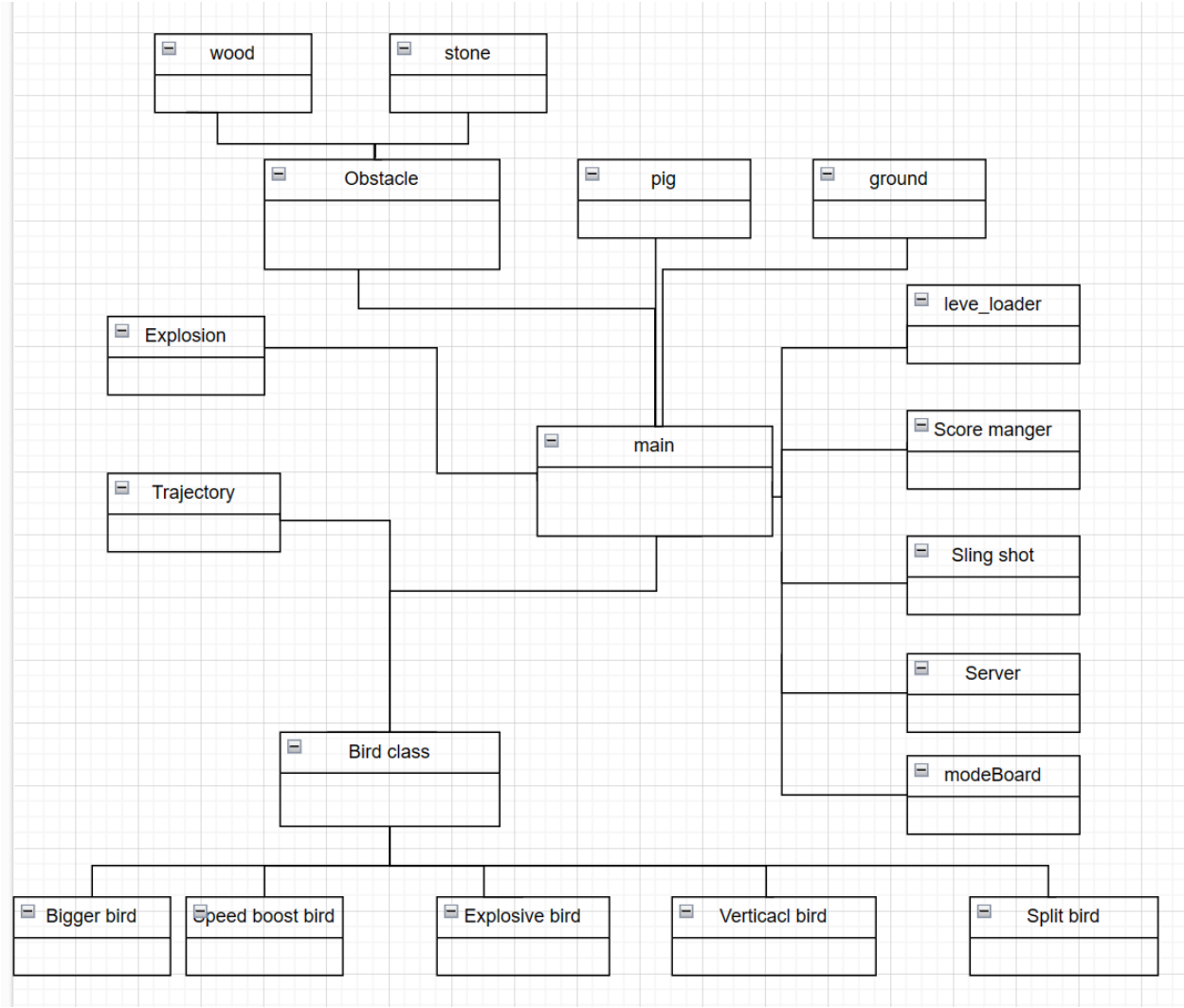
Use BOX2D to handle the game's physics, ensuring that collisions, gravity, and object movements behave realistically.

We did not pay too much attention to the level editor to create levels and save them in a file.

Software Structure:

We have a libs folder including external libraries like BOX2D and SFML and level-loading files. The media folder includes pictures of birds, pigs, obstacles, and background; it also includes background music and music for launching birds. The Src folder includes the main scripts for this project; the bird class works as the base class for all kinds of birds, the obstacle class works as a base class for wood and stone class, the sling_shot class for sling and launch, the scoreManger class for score savings, modeBoard class to choose different modest, trajectory class for bird's fly tracking and we have some other classes like pig, ground, and level load class.

Class Relationships Diagrams:



- Instructions for Building and Using the Software
  - Interfaces to External Libraries:
    - SFML for multimedia processing.
    - BOX2D for physics simulation.
  - Building Instructions:
    - Compiling the program using 'make', as detailed in the git repository.
    - Requirements for external libraries (SFML, BOX2D).
  - Usage Instructions:

        - Input ./Angrybird
- Testing
    - birds Testing
        - Individual Testing:
            - Each bird type is tested individually in-game to verify unique abilities (speed, split, explosion, vertical moving, etc.). Also, ensuring that they can work with trajectory class.Environment Interaction:
            - Testing bird interactions with obstacles and pigs to ensure realistic physics responses.
        - User Control:
            - Validating the user's control over the bird's trajectory and special abilities activation.
    - Pigs Testing
        - Destruction Testing:
            - Testing different scenarios where pigs are hit by birds or debris, ensuring they react and get destroyed as expected.
        - Scoring Validation:
            - Ensuring the correct allocation of scores upon pig destruction.
    - ScoreManager and modeBoard Testing
        - Score Calculation:
            - Verifying accurate score calculation based on bird usage, pig destruction, and leftover birds.
        - Level Progression:
            - Testing the score-based level progression system.
        - Mode choice:
            - Ensuring players can choose the mode as they wish.
    - Slingshot Testing
        - Mechanics Validation:
            - Ensuring the slingshot mechanics (pulling back, aiming, releasing) are smooth and responsive.
        - Interaction with Birds:
            - Testing the compatibility of the slingshot with different bird types.
    - Level Loader Testing
        - Level Initialization:
            - Verifying that each level loads with the correct layout, bird types, pig placement, and obstacles.
        - Transition Testing:

            - Testing transitions between levels, ensuring consistency and proper loading.
    - Obstacles Testing
        - Destruction Physics:
            - Testing the physical behavior of obstacles when hit by birds or falling debris.
        - Variability in Materials:
            - Ensuring that different materials (wood, stone) react realistically according to their properties.

    Outcomes:
    - Birds:
        - All bird types successfully realized their expected functions with correct physics, animations, and user interactions.
    - Pigs:
        - Pigs responded correctly to impacts, getting destroyed and scoring appropriately.
    - ScoreManager and modeBoard:
        - Score calculations were accurate and consistent across levels, with proper level progression. Players can choose 3 kinds of mode.
    - Slingshot:
        - Slingshot mechanics were responsive and compatible with all bird types.
    - Level Loader:
        - Levels loaded correctly with proper layouts and elements. Smooth transitions between levels were achieved.
    - Obstacles:
        - Obstacles exhibited realistic physical responses to impacts, with material-based variations in behavior.
- Work Log
    - Week1:
        - Jinxiong Lu

          Implementing a Bird base class with box2d world setup and SFML setup and rendering methods.

        - Qingyun Guo

          Implementing pig and obstacle base classes like wood and stone.

    - Nelly Nie

      Implementing different level maps.

    - Lujie Ban

      Implementing level loader class.
- Week2:
    - Jinxiong Lu

      Implementing a Speedboost Bird class and Explosive Bird class based on the Bird base class. Implementing a ScoreBoard class that handles user interactions for inputting nicknames, uploading and retrieving scores, and displaying level scores. It manages both local score data and online score synchronization.
    - Qingyun Guo

      Implementing a Vertical Movement Bird class, and Split Bird class based on the Bird base class.

    - Nelly Nie

      Implementing a trajectory class based on the Bird base class.

    - Lujie Ban

      Implementing a Bigger Bird class and modifying the level 1 json file.

- Week3:
    - Jinxiong Lu

      Implementing the SlingShot class with methods including setup, bird launching, rendering, and managing the slingshot state.
    - Qingyun Guo

      Implementing the modeBoard to manage the mode choice, one for a time limit, one for star collection, and improving the split bird.
    - Nelly Nie

Implementing the explosion class that has methods including playing animation.

- Lujie Ban

  Debugging the bigger bird class and modify corresponding loaders.