Project Plan

1. Scope of the work

Game logic

A classic Angry Birds game will be developed. The game's mechanism is to throw bird objects and get points based on performance, where a higher score is given to:

- Destroy more structures and pigs, or cause a chain reaction with a single shot
- Use fewer birds to clear the level
- Utilize birds' special abilities effectively
- Time bonus for quickly clearing the level

The player will be asked to input his nickname first. In the main menu, the user chooses the game mode:

- Reach the goal within the time limit
- Kill all the pigs

After choosing the game mode, the game levels will be loaded. The user can choose the highest level unlocked so far or any previously unlocked ones. The higher the level, the more difficult the game becomes. Higher difficulty in game level refers to:

- More pigs to kill with fewer birds available
- More complex construction structure

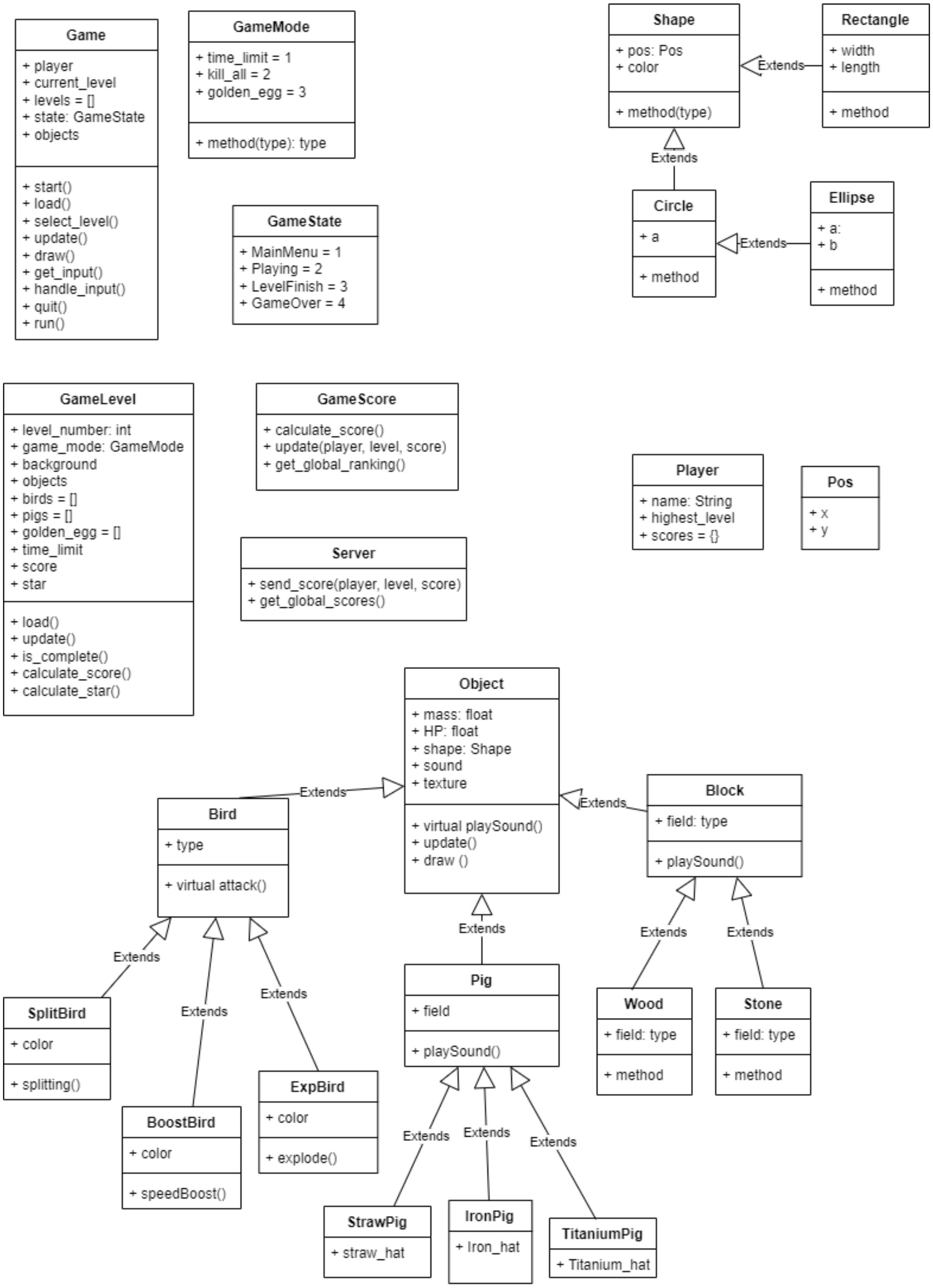
Players can see their scores in real-time in the game window. The total score will be saved if it is the highest for that level. Once the game is finished, the player will get a star rating (maximum 3 starts) to show how well the player played. Moreover, all players' scores are stored on the server side to generate a global score rank, on which players can see how well they played compared to other players.

The program has a Game class that handles basic game activities such as user interactions and the game state. When selected from the game window, the game can create a Level object to load level data from a file. The level is simulated by the physics engine and separately displayed as graphics in the game window. The window includes UI elements such as the score and a pause menu button. Physics objects have the same base class to enable easy game interactions. More detail about the program structure is shown in the class diagram.

Game features

Our goal is to implement all basic and additional features. The plan is to create at least 2 types of birds with different attack abilities.

2. High-level structure



https://app.diagrams.net/#G1vbzWpq90szBUUoeXQHzRgBlY5bYp9aQi#%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D

3. External libraries

- SFML(Simple and Fast Multimedia Library)

SFML is a comprehensive, cross-platform multimedia library in C++. Our plan is to use SFML for creating windows, processing events and drawing.

- Box2D - physics simulation

Box2D is an industry standard library widely used in 2D games, including the original Angry Birds. It provides realistic simulations and is optimized for game development

- GoogleTest - unit testing

GoogleTest is Google’s C++ testing and mocking framework.

All chosen libraries are C++ based, allowing deep integration. Besides, all these libraries are optimized for real-time applications, making the combination ideal for game development.

4. Division of Work and Responsibilities

&

5. Planned Schedule and Milestones

To manage the project efficiently, we’ll work in three sprints with defined milestones:

Sprint 2: Initial implementation (1.11. - 15.11.). By the end of the sprint there should be an initial running version of the project.

Sprint 2: Basic Mechanics and UI Setup (2 weeks)

- Doni: Implement basic bird launching and physics simulation.
- Soma & Yalda: Set up SFML-based graphics rendering and basic UI elements.

- Xin & Jaakko: Create a simple, functional level that loads from an external file.
- Milestone: Functional prototype with basic level and score display.

Sprint 3: Features complete (15.11. - 29.11.). By the end of the sprint the main features should be implemented, even if not finalized.

Sprint 3: Enhanced Features and Level Loading (2 weeks)

- Yalda: Add scoring mechanics based on throws and efficiency.
- Soma & Jaakko: Implement additional birds with unique special actions (e.g., speed boost).
- Xin & Doni: Develop a file-based level loader for multi-level gameplay.
- Milestone: Multi-level gameplay with different bird types and scoring.

Sprint 4: Finalization (29.11. - 13.12.). The project output, including documentation, is finalized, and tested.

Sprint 4: Optional Features and Refinements (2 weeks)

- Jaakko & Doni: Integrate additional features (e.g., sound effects, high score tracking, improved textures).
- Yalda: Conduct user testing and debug any gameplay issues.
- Xin & Soma: Finalize UI and overall presentation.
- Milestone: Completed game with additional features ready for submission.

6. Project Management

The team attempts to meet approximately once a week to review, plan and discuss progress and potential obstacles. In addition, progress can be tracked informally in a chat or more formally with an issue board.