Matilda Stendahl, Linnea Haapio, Casper Tillander, Julius Halmela
ELEC-A7151, Object Oriented Programming with C++

Angry Birds – Project plan

Project Scope

The objective of this project is to develop a game that resembles Angry Birds [1]. The objective of the game is for the player to throw objects at a fortress to destroy enemy targets. The game will include simple graphics, different levels, and a basic user interface that displays the statistics of the game.

The gameplay will be centered around the mouse. The player can use a drag and click mechanism to launch the birds, and a secondary click will activate a special action. The gameview follows the bird as it moves sideways. The game will include three levels, loaded from files in order to promote easy scalability of the game. A user interface will display the players points and the number of throwable birds left. We intend to implement at least a high score list, as well as a rating system using stars.

Structure

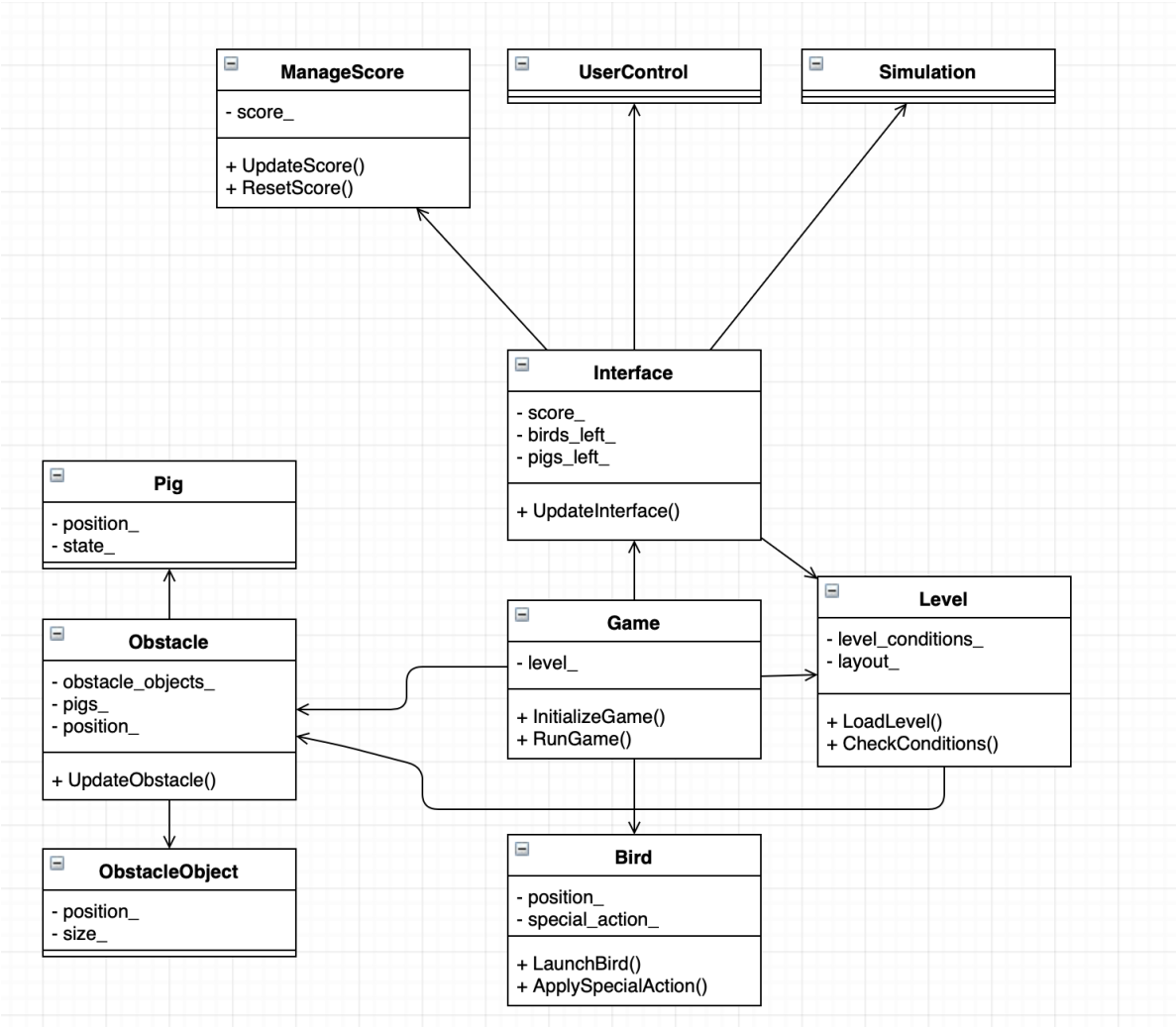
The project will be structured using different modules, which will help us maintain an organized codebase. The main modules will be the Game, Bird, Simulation, Enemies and Obstacles modules. Within these modules, there will be several classes.

The Game module will be the biggest one. It contains the Game, Level, ManageScore, UserControl and Interface classes. The Game class will manage the game overall, including levels, score and user interface. The Level class will create tree individual game levels with different difficulties. The ManageScore class keeps track of the player’s score. The UserControl handles user input and interaction in the game and the Interface class will handle the user interface, so displaying the game and creating the graphics.

The Bird Module will include the Bird class, which represents the player’s projectile and defines its special action. The Simulation module includes the Simulation class, which manages the object interactions and the physics of the game. The Enemy module will contain the Pig class, which creates and manages the enemy pigs. Lastly, the Obstacles module will

contain the Obstacle class, which creates and manages obstacles and the ObstacleObject class defines the individual objects within the obstacles.

The primary UML diagram is shown below.



External libraries

The physics of the game will be implemented using the Box2D library [2]. SFML will be used for the graphics and the multimedia [3]. The application framework Qt will be used for implementing the user interface [4].

Matilda Stendahl, Linnea Haapio, Casper Tillander, Julius Halmela

Division of work and responsibilities

Dividing the project into separate areas is difficult at this stage, but we have primarily divided the project into these following areas of responsibility:

Matilda: Bird and enemy implementations
Linnea: Physics simulation and obstacles
Casper: Game logic and score management
Julius: Graphics and user interface

More precise division of work will be decided at the weekly meetings.

Planned schedule

Week 43: October 23 to October 29, 2023

- Familiar ourselves with the project, research and brainstorm ideas
- Create the project plan

Week 44: October 30 to November 5, 2023

- Project plan review with our advisor
- Finalizing the plan with the given feedback
- Starting off the coding of the project; getting the structure done
    - Creating modules, directories, files and classes

Week 45: November 6 to November 12, 2023

- Implement the basics of the game, including the player, player actions, interactions with the objects and basic rules of the game

Week 46: November 13 to November 19, 2023

- Continuing with the basics of the game
- Implementing the basic user interface and graphics of the game; bird, obstacles, pigs, points and throwables left (very simple)

Week 47: November 20 to November 26, 2023

- Implement the mouse controls; developing the controls for the player and player interactions with the game
- Continuing with the interface and graphics, adding more details

Week 48: November 27 to December 3, 2023

- Creating the different game levels with different difficulties
- Adding additional features

Week 49: December 4 to December 10, 2023

- Project demo to our advisor, getting feedback
- Finalizing project based on feedback and returning the final version

References

1. *Play* (2023) *Angry Birds*. Available at: https://www.angrybirds.com/play/ (Accessed: 25 October 2023).
2. Erincatto (no date) *Erincatto/box2d: Box2D is a 2D physics engine for games*, *GitHub*. Available at: https://github.com/erincatto/Box2D (Accessed: 25 October 2023).
3. *Simple and fast multimedia library* (no date) *SFML*. Available at: https://www.sfml-dev.org/ (Accessed: 25 October 2023).
4. *Tools for each stage of software development lifecycle* (no date) *Qt*. Available at: https://www.qt.io/ (Accessed: 25 October 2023).