Advanced Programming in Java
Project Proposal: Pong Game

Mohammad Zain Yousuf, Saad Fahim and Abeera Alam

November 2022

1 Abstract

The aim of the project is to create a 2-player multi-threaded ping pong game using Swing. The GUI of the game will display two paddles, each at one end of the screen and a ball placed in the middle. When the game will start, the ball will start moving, and the movement of the ball initially is kept random, it can move towards any one of the two players. Users can move their paddles up and down on the screen. The ball upon collision with any one of the paddles will change direction and move to other player's side. The rules for assigning score will be the same as those followed in a regular ping pong game. The game will keep track of each player's score and run for three consecutive rounds. The player who wins 2 out of these 3 rounds wins the game.

2 Scope and Target Audience

2.1 Objects and Deliverables

- Utilize Java data structures and collections to store and manipulate the data for the game.
- Use JavaFx or any other library such as Swing to build the frontend and UI of the pong game.
- Optionally, include multithreading into the codebase and implement the game with multiple threads to concurrently perform the tasks and make our architecture efficient.

2.2 Expected Completion Date

The project is expected to be completed by the end of Week 16.

2.3 Target Audience

The target audience for this project will be gamers looking to compete with their friends in a virtual ping pong game. It will bring a sense of deja vu for those who are fans of the old arcade games.

3 Modules and Architecture

3.1 Modules

The project consists of 3 main classes:

- Paddle Class for Player 1 and Player 2- implements runnable interface and defines the functionality for the paddle.
- Ball Class for the Ping Pong Ball - implements runnable interface and defines the required functionality for the ball
- Pong Class to Implement the Game - uses paddle class and pong class to run the game. This is the main running module.

3.2 Architecture

3.2.1 UML Diagram

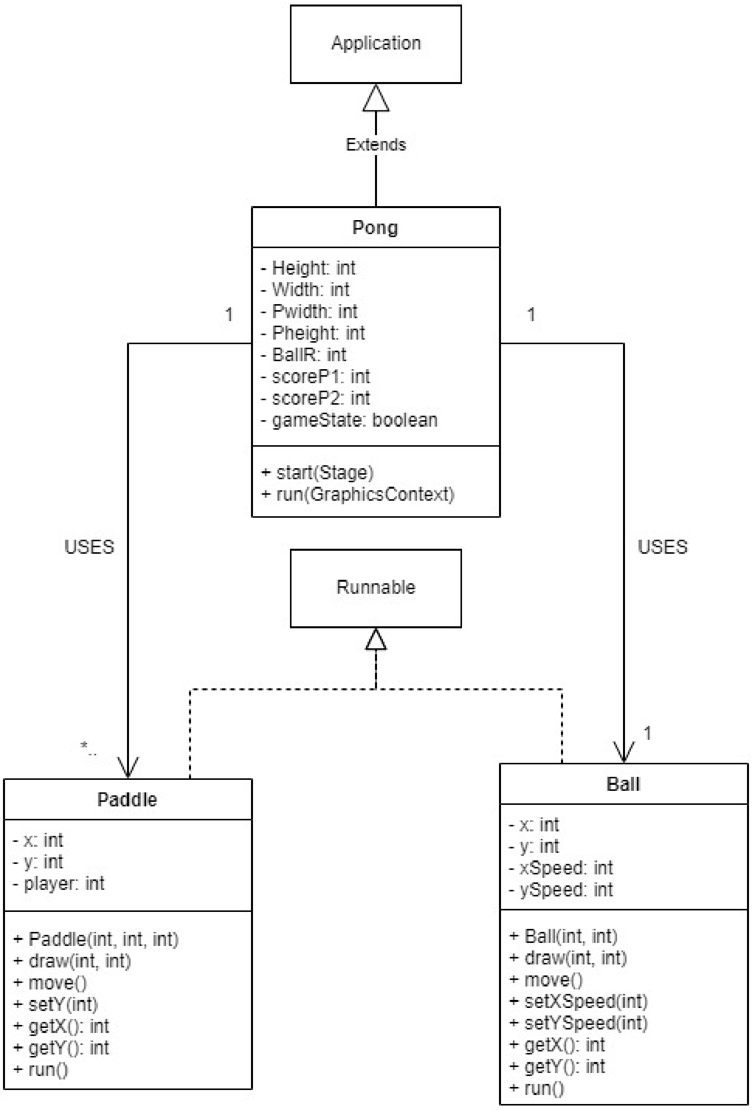

Figure 1: UML Diagram

3.2.2 Use Case Diagram

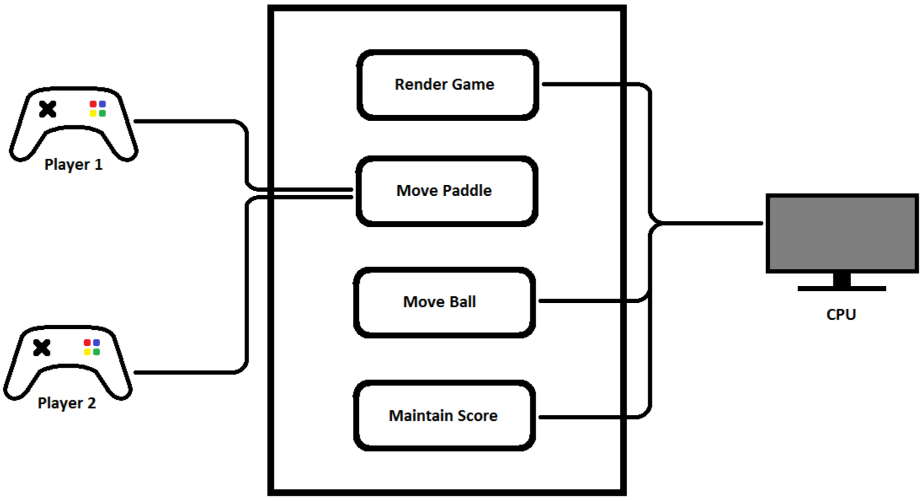

Figure 2: Use Case Diagram