Project 7 - Final Report
Peer to Peer Two-Player Gomoku (Five in a row)

Aishwarya Paithankar
Varad Raut

December 4, 2022

Project Name and Team 2

Final State of System 2

Final Class Diagram and Comparison 2

Third-Party Code vs. Original Code 5

OOAD Process Statement 5

Project Name and Team

Project Name: Peer to Peer Two-Player Gomoku (Five in a row)
Team Members: Aishwarya Paithankar and Varad Raut

Final State of System

We have successfully completed the implementation of all the features that were promised in Project 5. These include:

- Player discovery on the local network
- Rendering a 15x15 Gomoku board along with the stones
- Real time communication of moves between the participating devices

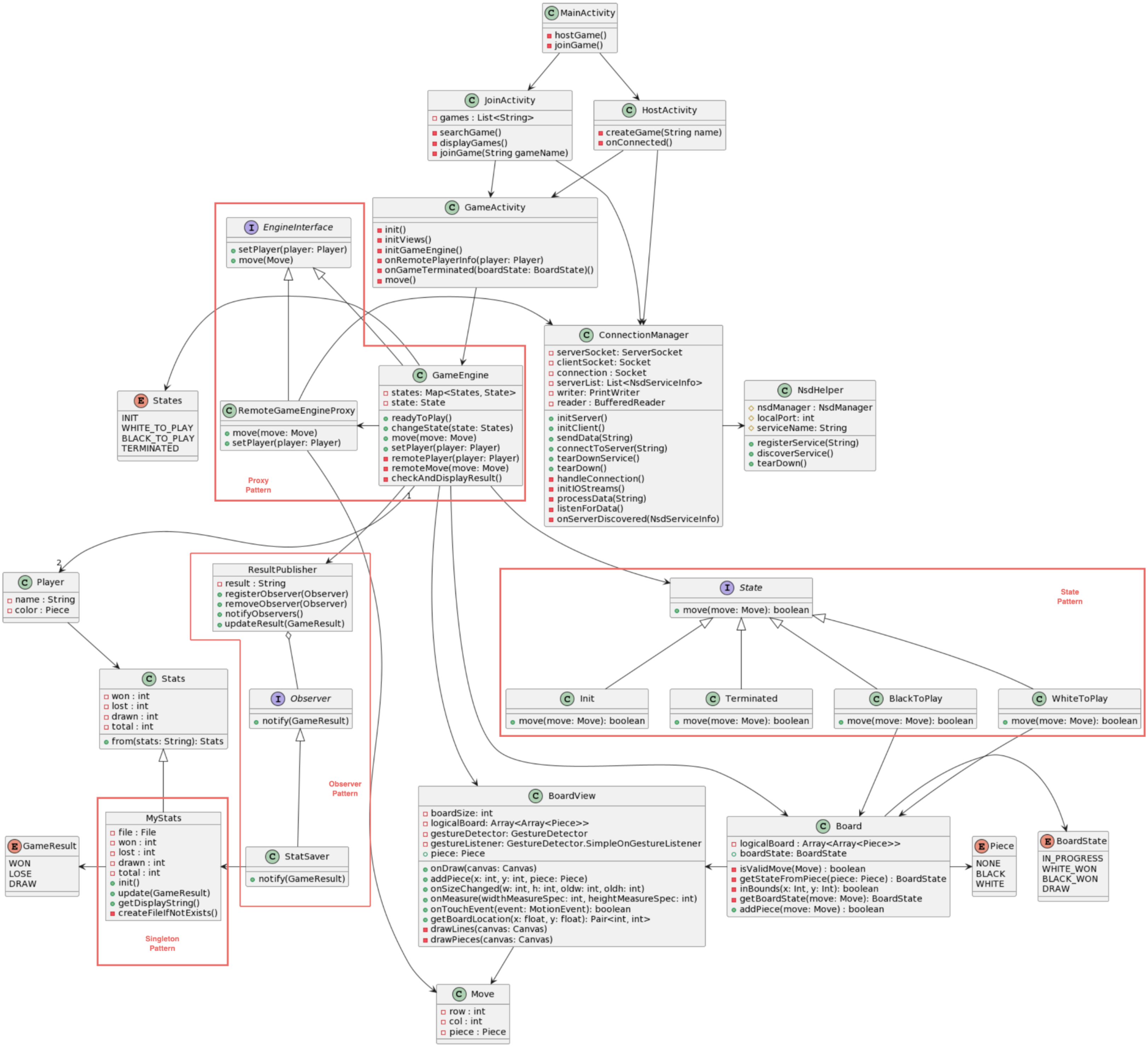
Feature wise, nothing has changed since the original requirements of project 5.

Final Class Diagram and Comparison

The overall structure of the classes did not change between projects 5 and 7. We did, however, add a few new helper classes:

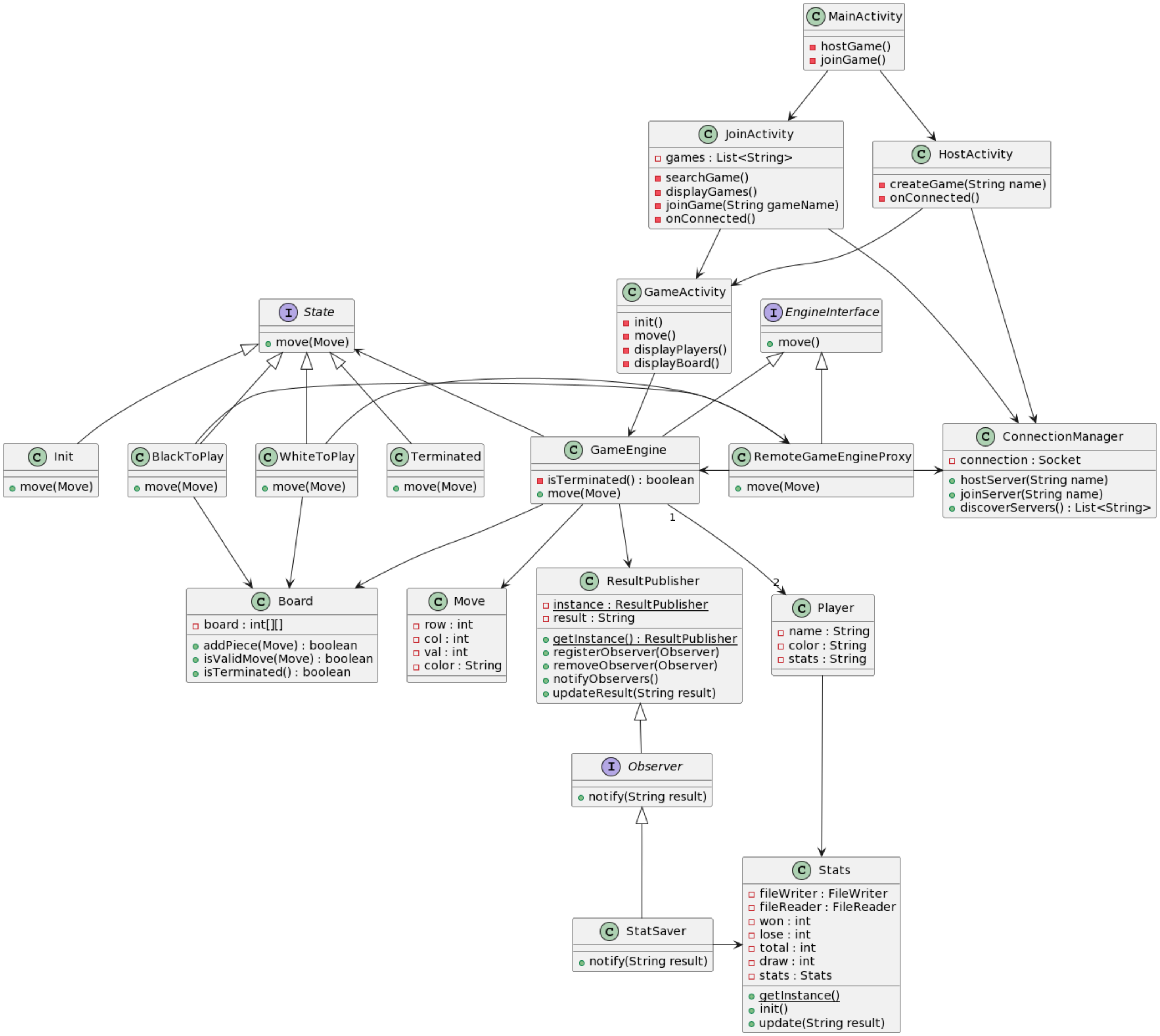
1. **BoardView:** This class extends Android's View class. It is responsible for rendering the board and stones on the screen.
2. **NsdHelper:** A helper class branched out of ConnectionManager. It contains code for registering and discovering DNS-based network services.
3. **MyStats:** A subclass of the Stats class, MyStats is used to represent stats of the current player. The Stats class is used for representing remote player stats.
4. **Enums:** We added a few enums to our code: Piece, BoardState, GameResult, States. These improved code readability and maintainability.

Final Class Diagram



Link to full-size image of class diagram

Project 5 Class Diagram



Third-Party Code vs. Original Code

All of the code in our project is original except for a library (android-ripple-background) that we used for showing a ripple effect animation when the user is waiting for another user to connect. Apart from that, we made use of external websites for tutorials and references. Our references are as follows:

- Kotlin and Android - Android development using Kotlin tutorial
- GitHub - skyfishjy/android-ripple-background - Library for adding ripple effect
- Develop - Android - Material Design - App UI design reference
- Use network service discovery | Android Developers - Android NSD API tutorial
- Brandmark Logo Maker - For making the home screen title image

OOAD Process Statement

1. The requirements phase in which we discussed and came up with use cases and the class diagram was the most important one. It became much easier to write code based on the planning done earlier.
2. Due to the complexities in using the Canvas and NSD APIs, certain classes were not added in the project 5 class diagram. We had to update the class diagram based on the complexities that we encountered while writing the code.
3. Identifying and implementing the State and Proxy patterns was a rewarding experience. Both these patterns made it easy to maintain and understand the code, in our opinion.