Cocharge Doc - Rock Paper Scissors Implementation

Requirements Met

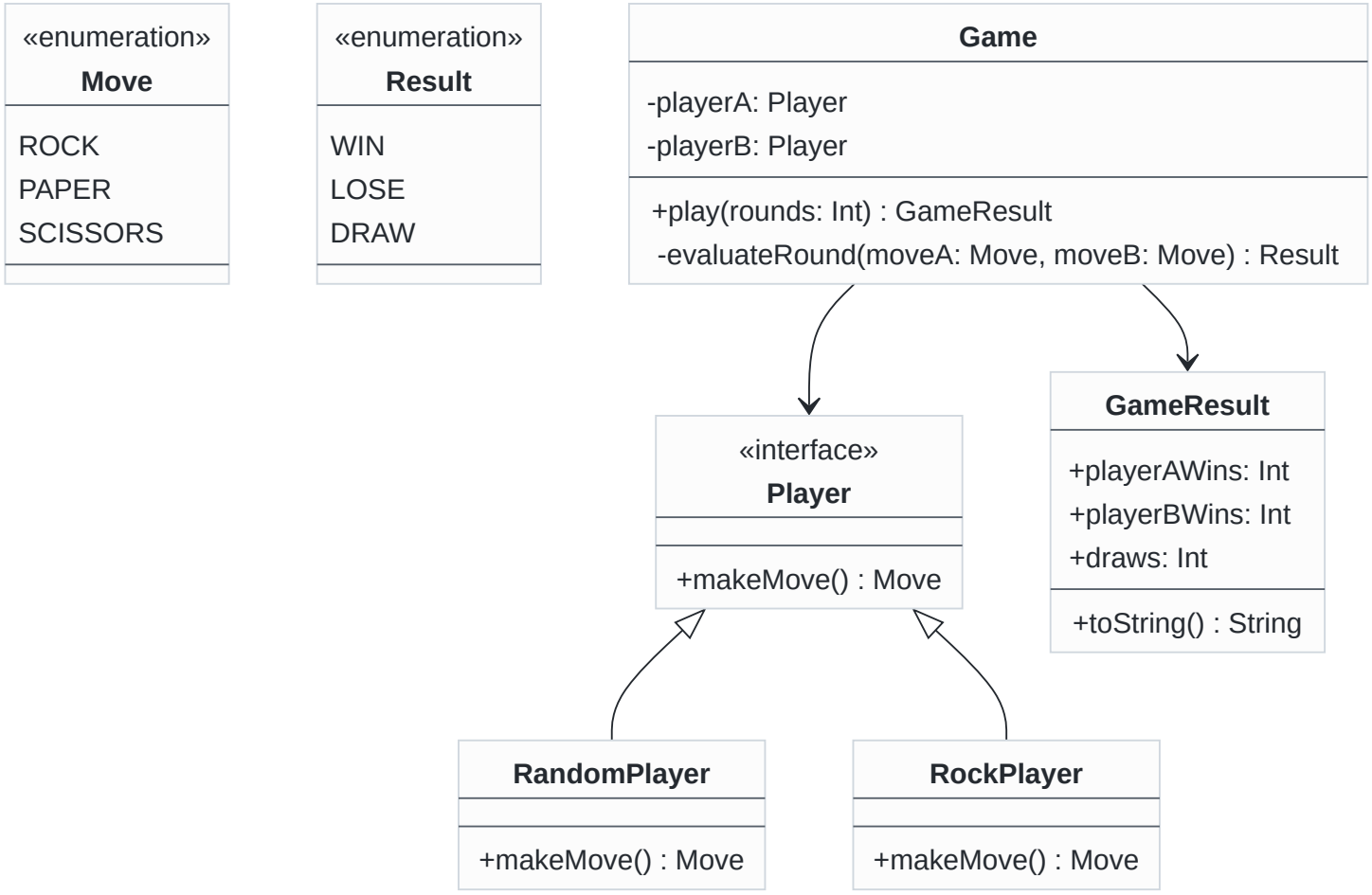
- Random vs Rock player implementation ✓
- 100 rounds of gameplay ✓
- Game logic evaluation ✓
- Win/loss/draw tracking ✓
- Test coverage ✓
- Clean code principles ✓

Technical Documentation

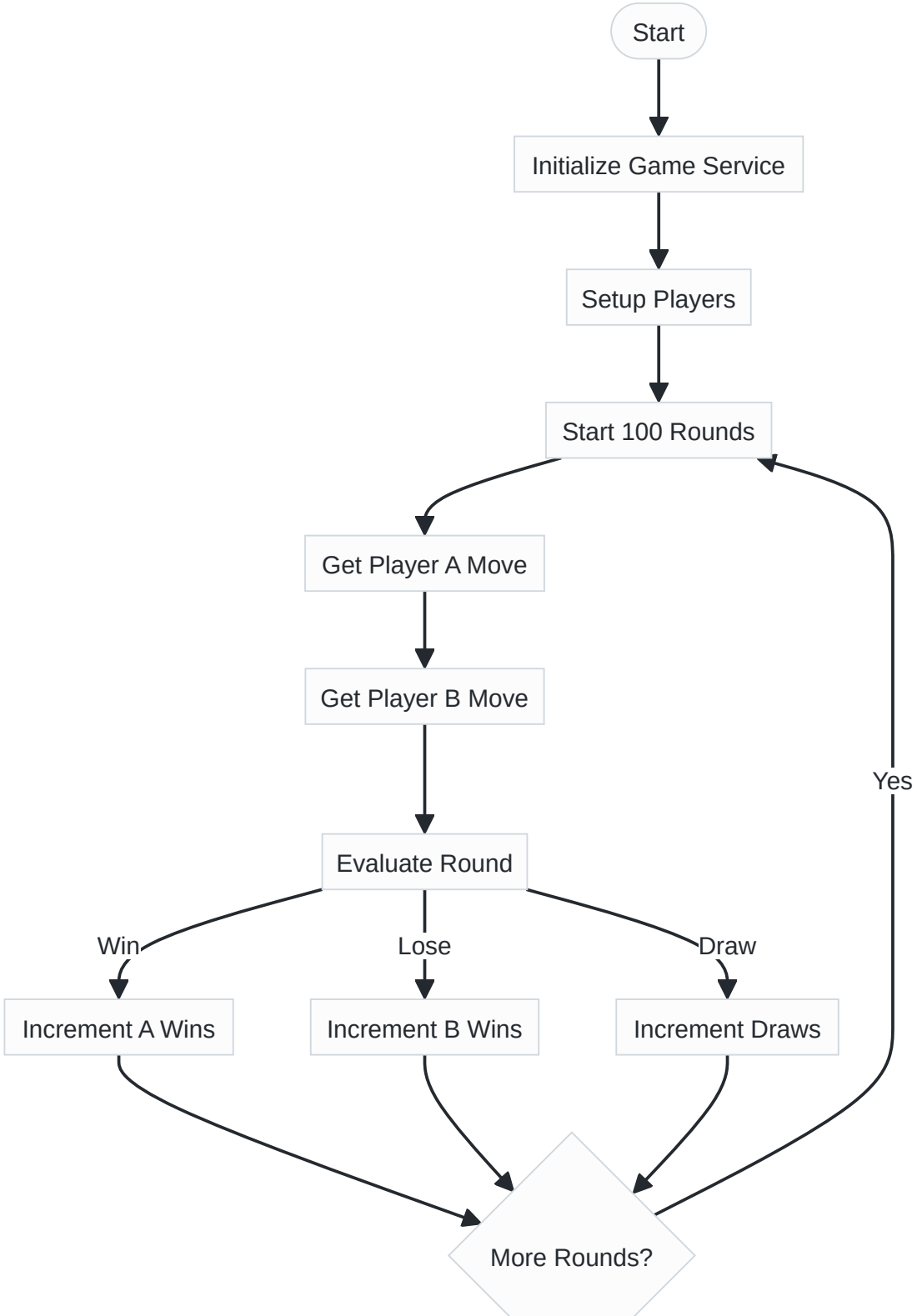
Project Structure

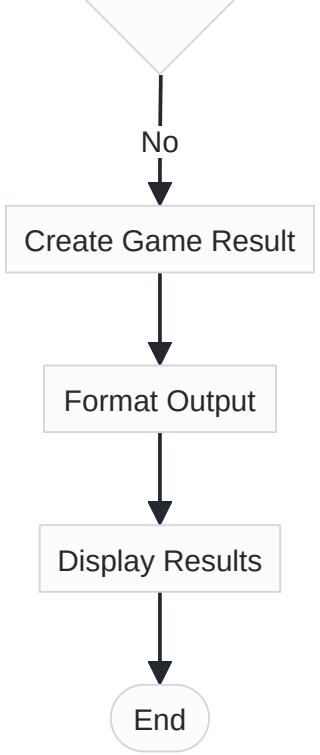
```
src/
├── main/kotlin/com/example/rps/
│   ├── domain/
│   │   ├── model/
│   │   └── player/
│   └── application/
│       ├── game/
│       └── service/
└── test/kotlin/com/example/rps/
    ├── domain/
    └── application/
```

Class Diagram

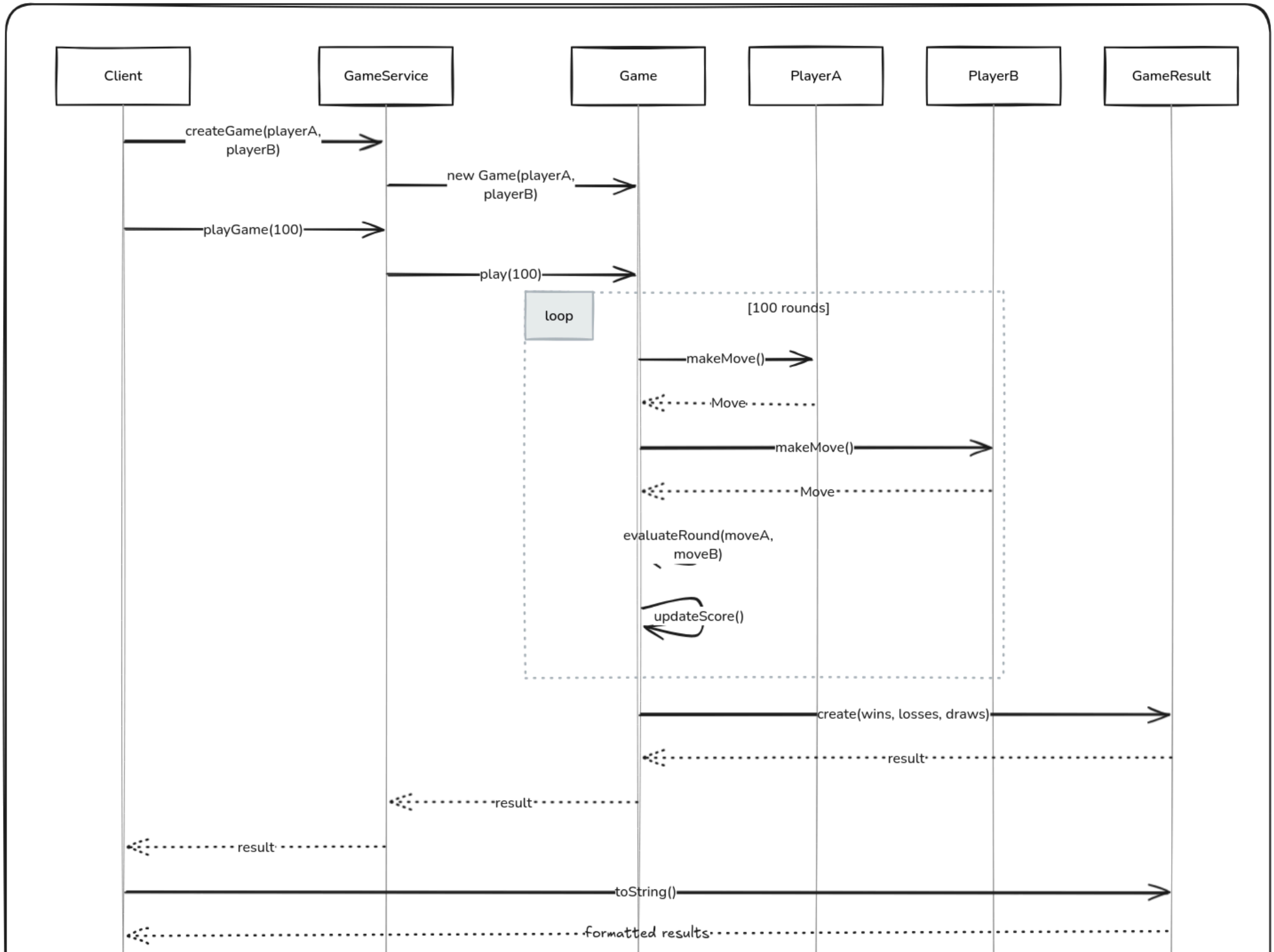


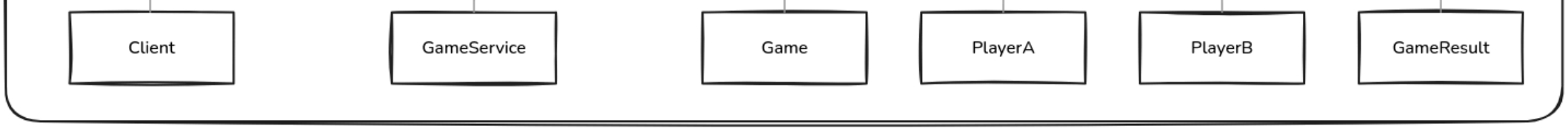
Flow Diagram


Start
Initialize Game Service
Setup Players
Start 100 Rounds
Get Player A Move
Get Player B Move
Evaluate Round
Win
Lose
Draw
Increment A Wins
Increment B Wins
Increment Draws
More Rounds?
Yes



Sequence Diagram

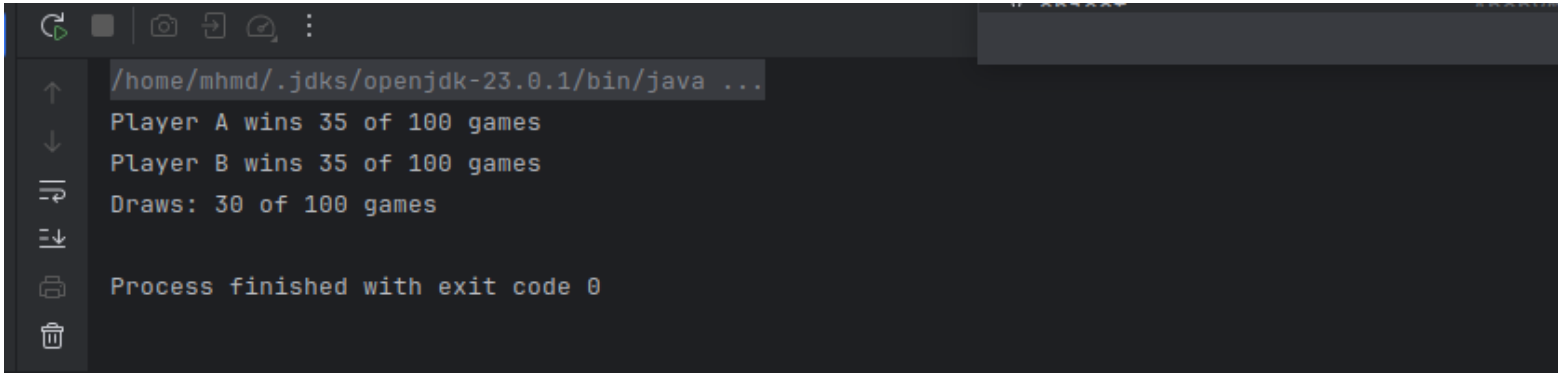

Client
GameService
Game
PlayerA
PlayerB
GameResult
createGame(playerA, playerB)
new Game(playerA, playerB)
playGame(100)
play(100)
loop
[100 rounds]
makeMove()
Move
makeMove()
Move
evaluateRound(moveA, moveB)
updateScore()
create(wins, losses, draws)
result
result
result
toString()
formatted results



Test Coverage

- Unit tests for all components
- Mock player interactions
- Game logic validation
- Result calculation verification

Build and Run

```
./gradlew build
./gradlew test
./gradlew run
```

Sample Output