Paul Council
Anand Patel
Jonathan Gonzoph

Project Specifications

- Project Title
    - Adaptable R.P.S.
- Project Summary
    - Provides a robust game framework for hosting competitions of games akin to Rock Paper Scissors. In such games, players move simultaneously and the rules of the game decide who wins. This framework will be able to recover from network issues and support player ai regardless of language implementation.
- Project Goals
    - Play a game of RPS with multiple people (running on different machines)
    - Run a single elimination tournament of RPS with more than two people (running on different machines)
    - Actively display results to the tournament director and display errors (if any) to the players.
    - The tournament should be able to attempt recovery from network errors.
    - The Player class can be implemented in multiple languages, specifically Java, Python, and C++.
    - The tournament allows multiple matches to be played simultaneously.
- Project Features
    - The Game, Tournament, and Player are interfaces with comments describing what should be provided by the programmer for optimal implementation. This

style of coding should open our project up to people who would normally shy away from coding for fun.
    - Users can easily create their own player with their custom designed decision making Player “AI” written in (common) language of their choice.
    - Games supported are easy to learn and therefore easy to code for the game maker.
    - All tournament styles can be implemented with the provided interface for the tournament holder to easily code.
    - This framework connects across a network to allow multiple people to connect and play.
    - Relevant wins and losses in a match are recorded and accessible by the participating Player’s AI which allows strategies to be coded into a Player.
    - The interfaces allow this project to grow as new games and tournament styles are invented.
    - The tournament server supports swapping of different styles of Python implemented networking connections.
- {Limitations}
    - The Player “AI” must be implemented using one of the common languages supported by json (listed on json.org).
    - The Tournament Director and individual players have to be located on the same network.