Amos Byon

Anne Kao

Jess Lai

Jon Park

Kevin Kim

**Technical Specifications**

**Hardware Requirements**

* 1 Server computer to network the client computers
* 4 client computers to run an instance of the game

**Software Requirements**

* Java
* Eclipse
* Github
* libGDX framework
* SmartFoxServer
* More details to be covered in Design document

**Server**

* Will contain all the data from the running game
  + Characters
    - Position
    - Health
    - Items
    - Attacks
  + Map
  + Statistics
    - Later to be exported and sent to each character / client computer as a CSV
* Clients will send player input to the server to then be transmitted to other clients in the game
* Server will contain a log of all chat logs that players input
* Server will ensure that all players are synchronized and show events as they are happening with little to no latency
* Server will randomly spawn items throughout the map that will give a temporary buff to the player that picks it up

**Client**

* Each client computer will have an instance of the game running on their machine
* Each client must receive packets of data from the server to continuously update the game on their screen with the actions from other players
* Each client is able to send their player commands to the server to be transmitted to the other three client computers
* Each client is able to send a message to the other players (to all, private, team only, etc)
* Each client should only be able to see a portion of the entire map that they are in

**Export Statistics**

* Each client will keep a track of their statistics throughout the game (Kill/Death/Assist ratio, damage dealt, damage received, time alive, accuracy, etc)
* At the end of the game, players will send their personal stats to the server
* The server will display the end game statistics to all players
* End game statistics can also be exported as a CSV if players would like a record of their game

**Estimate of hours**

* Across 5 programmers, approximately 200 hours