How to compile and run the program

1. Please use the terminal to go to the src file
2. Type “make”
3. Type “make run”
4. And you can see the result

The main is in the /src/test/java/main.java

You can specify how many match you want to and how many people are going to be inserted into the pool by specifying

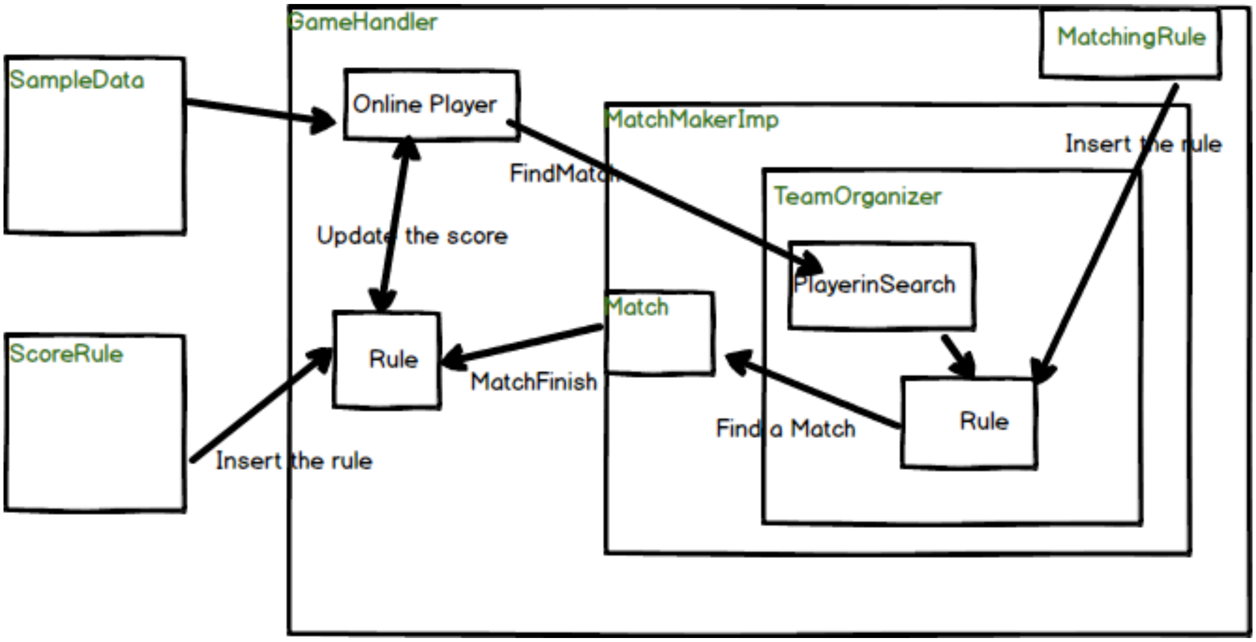
```
int howManyMatch=100;
int retieveHowManyPeople=200;
```

*I’ve add a attribute rankScore in Player class which indicate the ranking of each player

Basic Overview

The basic structure is described as below.

Overview



Everything is handled by GameHandler.

GameHandler receive the SampleData and the ScoreRule we want to use

And it will init a MatchMaker Object . Match maker will return a match object.

After that the GameHandler will update the result of the match into the database.

Strategies

There are two strategies that can be change in the program.

1. ScoreRule:
   This rule is in charged of update the player’s score after a match
   You can implement new rule by implement this ScoreRule class.
   You can change the scoreRule while init the GameHandler

   src/test/java/main.java

   ```
   GameHandler gameHandler= new GameHandler(onlinePlayers , new BasicScoreRule());
   ```

2. MatchingRule
   This rule is in charged of selecting the players into the matching system.
   You can implement new rule by implement this matchingrule class.

   You can change the Matching in
   `src/com/riotgames/interview/hongkong/gamehandle/GameHandler.java`

   ```
   Match now=matchmaker.findMatch(playersPerTeam);
   ```

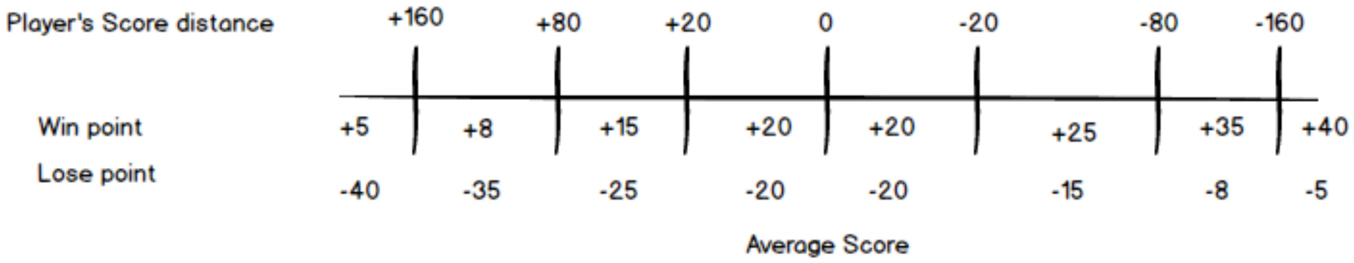
   change to

   ```
   Match now=matchmaker.findMatch(playersPerTeam, new BasicMatchingRule(playersPerTeam));
   ```

Strategies Overview

1. Basic Score Rule
   The algorithm of BasicScoreRule is described below
   After a match finished, we can know which team wins. Moreover, we can get the average score of each team.
   Then we calculate player.score-average score

   Base on the picture above, we give different points to each player.



2. Matching rule
   After we retrieve the player in the searching pool.
   We sort the list based on player’s score.
   And we use a random number to pick the first player randomly.
   After that we find other players that has the closest score the first player.