PART 1 – APP

Imagine a game like League of Legend or Overwatch. They use a matchmaking service to match users with the same skill. So each user calls the matchmaker and after a reasonable amount of time, he receives the address of a match-server to connect to, and play a match with some other players with about the same skill.

A- SERVICE DESIGN

1. Please describe how you envision the subsystems of the matchmaking application.
2. How will those subsystems communicate with other services ?
3. What could be the interface with the game ?

B- PROGRAMMING

You can answer this part in C++ or python.

1. Simple Matchmaking.

You receive a list of 1000 to 2000 entries of the form (PlayerID, Skill) as the input.

Example:

[{ID: 0x123A89, Skill: 264},
{ID: 0x4E3A12, Skill: 812},
...
{ID: 0x12BA13, Skill: 12}]

You should return the best match that is composed of 2 lists of 6 players (= 2 teams of 6), that verify those rules :

- Delta Skill between two players in a match have to be minimal (id : Players want to play with near skill players).
- Delta between the two team sum skills have to be minimal (id : team are balanced).
- If no match satisfies a "quality criteria", it has to return nothing.

Write code that returns the possible matches using the list of players as input. After the call, only players that don't belong to a match have to stay in that list.

2. Matchmaking improvements.

Improve the previous function for squads management (group of friends playing together). Now you receive a list of 1000 to 2000 entries (PlayerID, Skill, SquadId).

Example:

[{ID: 0x123A89, Skill: 264, SquadId : 148},

{ID: 0x4E3A12, Skill: 812, SquadId : -1},

...

{ID: 0x12BA13, Skill: 12, SquadId : 148}]

If SquadId is -1, Player doesn’t belong to any Squad, while all players with the same SquadId have to be in the same match, and the same team.

Modify the previous code.

3. Bonus : Further matchmaking improvements.

What could you add to this system to prevent players from waiting too long for a match?

Modify the previous code.

PART 2 - CONCLUSIONS

1. How long did it take you to complete this test ?

2. Remarks and suggestions ?

Thank you for your participation !