There are three kinds actors.

1. Mobile app client (mobile)
2. Game client (game)
3. Firebase server (firebase)

**Mobile**

Mobile interactions

1. Create lobby set
2. Get available lobbies
3. In lobby
4. player number ie. joining order.
5. your player number.
6. Get other player avatar infomation.(picture, name, wins/losses)
7. Change game variables (ex. difficulty)
8. In game
9. Get current money.
10. Get lives left
11. Get kill count
12. set cursor position
13. place tower(tap screen with a tower selected)
14. get win/loss result from match.

**Create lobby**

The player will set a name, and game variables(ex. difficulty, number of lives). Or the player can use the default settings, this is pre-defined in the app. Then the player publiches the lobby to firebase. There should be created a unique lobby id. This id will be used as a reference to exchange information. All listeners should be on this id. The lobby should be put in the list of available lobbies.

**Get available lobbies**

Query firebase for a list of available lobbies, the names of them,how many players are in the game and the lobby’s id.

**Joining a lobby**

The player joins a lobby by choosing a lobby. Using that lobby’s id the app makes sure the lobby is not full, puts the player name in the player spot available to claim it and places listeners at the appropriate places??(under the player spot?? ie player1, player2, player3, player4) so maybe ref.child(gameId).child(“player4”);

When someone presses ready the creator of the lobby(or other if that person has left). will see if all have pressed ready and if they’re all ready the creator will change a value from false to true that the game server will be subscribing to.

In the lobby the players can change setting for the game. these should be saved on the server so the game can get them when the game launches.

**In Game**

In the game the players mobiles will subscribe only to the variables of their particular player child node ex .child(“Player4”).child(“money”)

{

“gameId123”:

{

“player4”:

{

“money”: 100

“kills”: 14

“lives\_left”: 50

}

}

}

**Game**

Game interactions

1. Get game variables
2. get player avatars
3. get game launch
4. In game
5. players cursor position
6. players tower placement
7. set money
8. set lives left
9. set kill count
10. player connection status?

When we want to remove the game we just set the game id to null.

Database structure

{

“gameId123”:

{

“player1”:

{

“playerId”: “ShadowMaster”,

“money”: 100,

“kills”: 14,

“lives\_left”: 50,

“X”: 100,

“Y”: 200

},

“player2”:

{

“playerId”: “Jonas”,

“money”: 100,

“kills”: 14,

“lives\_left”: 50,

“X”: 100,

“Y”: 200

},

“player3”:

{

“playerId”: “007”,

“money”: 100,

“kills”: 14,

“lives\_left”: 50,

“X”: 100,

“Y”: 200

},

“player4”:

{

“playerId”: “Fish”,

“money”: 100,

“kills”: 14,

“lives\_left”: 50,

“X”: 100,

“Y”: 200

}

}

}

Nödvändig? om vi bara ska ha ett game gående i taget.

{

“active\_games”:

{

“gameId123”: “active”,

“gameId124”: “empty”,

“gameId125”: “active”,

…….

}

}

annars så kan vi ha:

{

“lobby\_ready”:

{

“player1”: true,

“player2”: false,

“player3”: true,

“player4”: false

}

}

{

“game\_rules”:

{

“difficulty”: “medium”,

“starting\_cash”: 1000,

…….

}

}

Funkar detta med att lobbyskaparen skapar hela strukturen på firebase och de andra bara uppdaterar värdena?

Och när spelet är färdigt så töms game informationen och sätts till default värde tex strängen “empty”

Borde Positionen separeras till en egen nod eftersom de har högre uppdaterings frekvens?

Blir kladdigt i sådana fall.

Vi har inga security conserns väl?

Vi kan sköta read write tillåtelse från appen.