Communication framework

Client-server communication and error handling

The way communication works in our implementation, is by sending location updates with a fixed time interval. These serve as acknowledgements that the client has a reliable connection. For this app, a time interval of 3 seconds is chosen in order to reduce stress of the server and client, but still have timed updates on all player’s locations. If the server has not received a ping after 30 seconds, the player is removed from the locations list and is regarded as ‘inactive’. The server won’t be asking for new positions or calculating its current position, until the client reconnects and sends a new request. The player is then added back to the ‘active locations’ list.

**Dropping connections**

When the client does not send an acknowledgement within 3 seconds (due to a communication error or bad signal), the player is still regarded as active. Other players can initiate a battle, tap on the player, etc. (but the other player is still put on hold until a two way acknowledgement is set-up for battle). If the communication error persists for 30 seconds or more, the player is regarded as inactive. If other players were waiting for a battle to instantiate, they will be notified that the player is offline, and any communication is terminated.

**New implementations**

In the second part of the project, the team is asked to still keep track of ‘older locations’ of players. Removing an inactive player from the locations list is not feasible in this scenario, as the information of the last location is lost. But this sort of location can be kept separately from other active locations that do need to be updated every few seconds, as these remain constant anyway. This will reduce server stress.