Assignment 5: RMI Game (40 Points)

In assignment 5 you should re-write the communication logic of your client-server game from assignment 4 to use the Java RMI technology. You can use the same game implementation (UI) etc, you should **just** re-write the part of the application doing the communication between clients and server. You may extend the scope or your game logic, however it is **not** required.

The application should still support the same basic concepts as for assignment 4:

- Login via unique Name
- Player movement
- Logout of client

If not already done for assignment 4 you may add additional concepts (not mandatory):

- Different kinds of players
- Winning
- Limited field of vision
- …

You should use **RMI** to implement a **non-polling**, **asynchronous** communication scheme between the server and the clients. Clients login to the server with their name, the server must save a remote reference of the client to realize non-blocking callbacks to the client. Add proper logic to ensure your application can do callbacks asynchronously. Clients hold a remote reference to the server and the server holds a reference to the client (per client). Other types may be realized using the **Serializable** interface.

Hints

- Revisit your implementation from assignment 4 and do a clean architecture dedicated to the RMI framework.
- Define proper types for remote objects and serializable data types.
- Beware of blocking callbacks they may result in deadlocks.
- Decouple computations from the RMI threads by scheduling them separately to avoid blocking other clients (i.e. by using an ExecutorService).
- **Synchronize** your data structures if need to avoid data races (try to reduce the locking to a minimum)