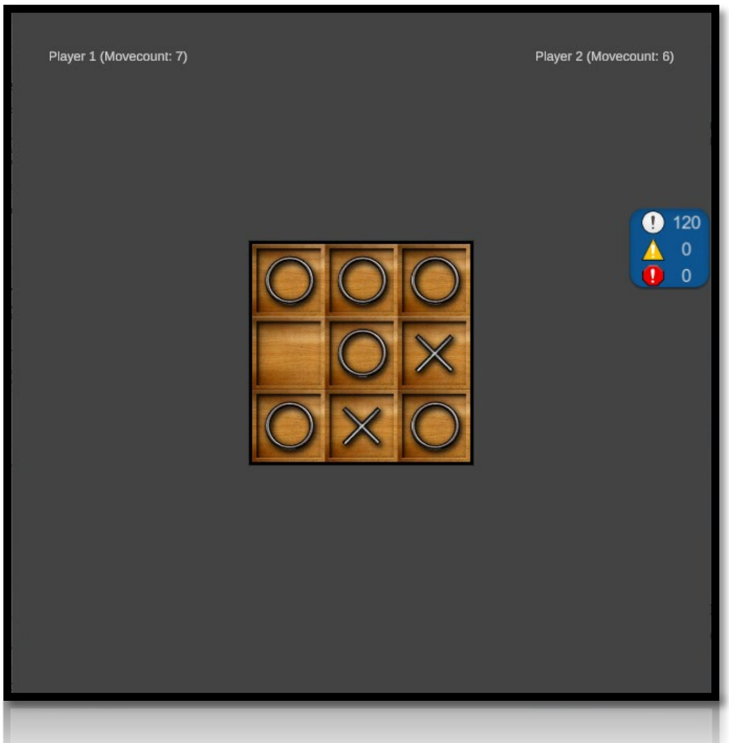
Networking - Assignment 4 - Turn-based Multiplayer Game



Getting started

In this 4th & last assignment you have to implement/extend a turn-based multiplayer game. A starting project is available for download in a single zip on BlackBoard. Making the graphics beautiful and unique is nice for your portfolio, but ***not a requirement*** to pass the assignment.

Since all of lecture 4 was about getting started with this assignment, this time around the 'Getting started' section for the assignment is fairly small and consists of just a couple of tips:

- Start by running the server and a couple of clients (build if necessary).
  Test out all that works, and all that doesn't work.
- Go through the lecture again if required, to gain a general understanding of the application(s), investigate all classes as they are discussed in the lecture, ask question during lab if anything is unclear.
- Investigate the different state related GameObjects in the client Hierarchy and how the states are connected to the views (and vice versa).
- Restart the client and the server and by using the Server log and the Unity inspector, try to follow along with the flow of messages/states & rooms that occur in response to your actions (see slides as well!).

Sufficient requirements (basic gameplay with multiple clients):

- Clients need to enter a unique nickname to 'log in' on the server.
  If a client tries to log in with a nickname that is already in use, the client should be warned appropriately.
- The lobby correctly displays the name of the client that has joined.
- Clients can chat in the lobby and their messages will be prepended with their nickname by the server.
  (same as assignment 1, but in a different context and only basic chat is required.)
- The game screen displays the names of the clients that are playing (e.g. P1 Bob - P2 Anna)
- The game has an end condition:
  - The game ends when one player has 'claimed' all tiles.
  - On game end, clients are returned to the lobby and see in their chat whether they've won or not (Note: clients might still be sending MakeMoveRequests at the moment a win condition is reached. The server doesn't need to handle this, as long as it doesn't crash).
  - *You may assume players do not leave the game/server while the game is still in play.*
- Whenever 2 clients in the lobby are ready to play a game, their game starts immediately.
  In other words: the server runs multiple games at once, without a hard limit to the amount of games.
- Finished game 'instances' must be destroyed or reused and your approach should be visible in the log.

Good requirements (improved client handling):

- The server uses heartbeats to detect and remove faulty clients.
- The client can actively leave a game by 'conceding'.
- The client can passively leave a game by no longer 'existing'.
- Whenever a client leaves a game, the game ends.
- When the game ends, clients get to see an intermediate end screen with the final game results.
  They can leave the end screen through whatever mechanism you devise to be returned to the lobby.
- You are free to assume clients do not go /afk (you do not have to handle that situation)

Very good requirements (full Tic Tac Toe):

- You've implemented the real TicTacToe game with **turns** and **rules** (read Excellent part first).
  - Clients can see whose turn it is.
  - Clients should not be able to take actions, when it is not their turn.
  - Server should not accept moves from clients if it is not their turn.

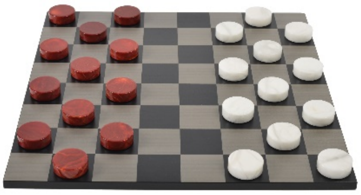
Excellent requirements (implementing a game of your choice):

- You've implemented a different game than TicTacToe or did something equally impressive (networking course related :)) for which you deserve an excellent. In case of the latter, discuss your plans up front with your lab teacher.

Game ideas

Here are just some ideas for games, some more complicated than others, but all demonstrate the level of complexity you should aim for (definitely not make it more complicated than this):

Connect 4

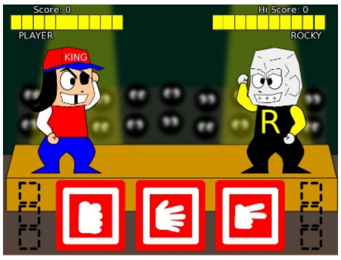


Checkers

WII Party 'Strategy steps'

WII Party
'Choose a platform'

Nintendo switch
'Air to a fortune'



Some sort of rock paper scissor/timebased battle with x rounds.