# PCSS_Group10 hand-in

For our hand-in we wanted to focus on the code rather than the graphics. For that reason, we made a command prompt application that allows 3 clients to join a server that is split up into more lobbies that the players can open and join.

The game concept is very simple. You join a server and roll a dice (1-100)– highest dice roll wins the game. This might not be the most interesting game concept, but it proves that we can make a simple network application.

The solution is created in C# and Visual Studio 2017.

For more in-depth explanation, refer to comments in the code.

Use case diagram
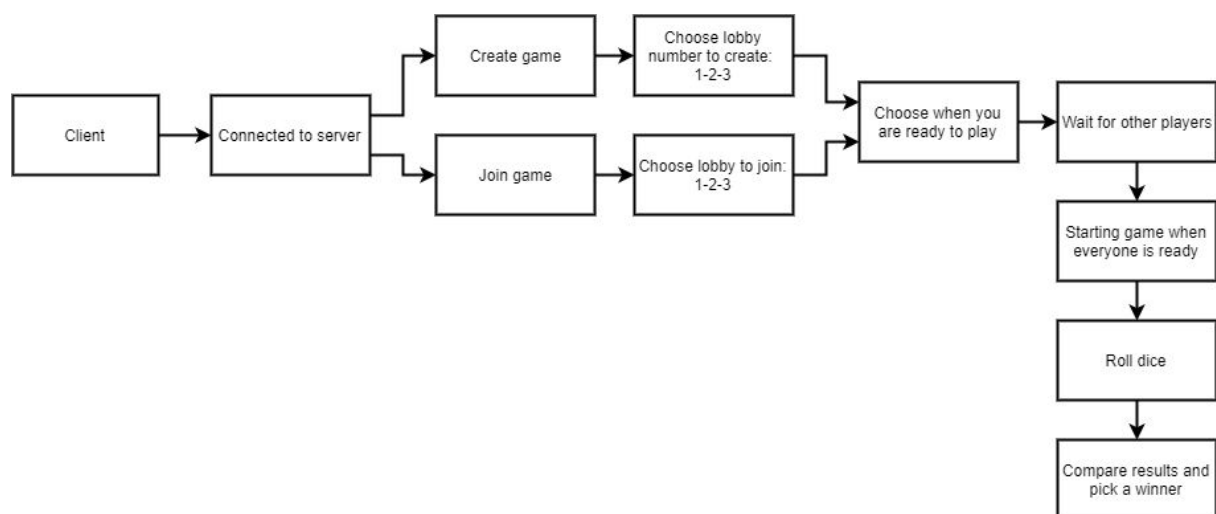
How to use solution

Step 1. Open server
Step 2. Open 3 clients
Step 3. Follow on screen instructions. Start by choosing a name. All of them are "admins" and they can all open or close the 3 lobbies. Type lobbylist to see which lobbies are open, and if any are open you can see how many players are in them. When 3 people join a lobby, the game starts.

Notes regarding GitHub and group work

Our group was split into these:

SERVER: Bjørn - Mathias - Jens
CLIENT: Alexander - Josef - Simon

However it was very hard for us to do it like that, so the lines have been very blurred out and we have mostly all worked together. We have done most of the work together, and for that reason we have used our group blackboard rather than the GitHub ticket systems - we failed to recognize that that was a requirement until right now.. (05-11-2017) and it's pretty much too late to do it now. We hope it is not a problem.

Mathias Kjeldsen has a Mac and for that reason he has been unable to use C# and GitHub due to compatibility problems with Visual Studio 2017. For that reason, he has worked closely with Bjørn Skalkam in person (they live close to each other) and through screen-sharing on Google Hangouts. Approximately one fourth of the commits that are on GitHub in Bjørns name should be viewed as something Mathias has created.

Additionally, when we have worked together in school we have had a hard time working on several computers at once, since the code messes up when more than one person is working at the same script at the same time due to overwriting each other's code and things like that. So we worked together on the code on Bjørns computer most of the time, and did most of the commits from his computer, instead of switching computers all the time for no reason.

We feel that the project is simply too small for 6 people to work on simultaneously – with too few scripts to work on.

Furthermore, while some of us worked on the solution that we ended up submitting to GitHub, others have spent a lot of time working on an alternate, different approach using sockets, which would allow the server to reply to multiple clients via socket, and not just on client at a time. This code did not work out in the end, even though it was very close to working like intended, so all of this hard work was discarded and is not reflected in the GitHub point system.

Last but not least none of us knew C# before starting this project, and since all of the classes this semester have been in C++ and we are suddenly supposed to use C# for this, we have spent a lot of hours and had a very hard time trying to figure it out.