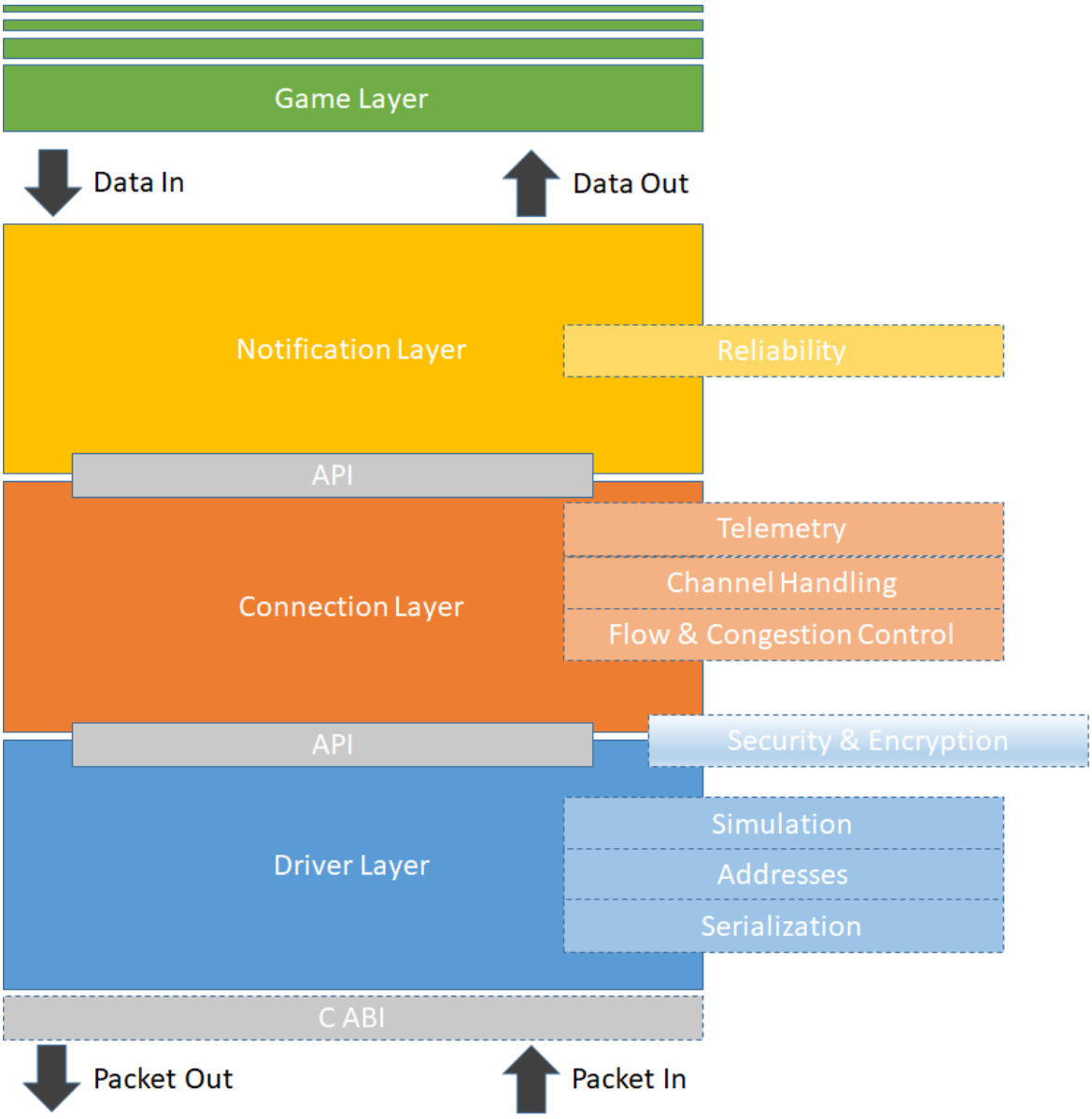
Game Development 4 (Herkansing)

Dylan Smit

3034198

17-6-2022



Contents

Links

Git Repository: https://github.com/smitdylan2001/KM4-Redo

Windows build: https://github.com/smitdylan2001/KM4-Redo/releases/tag/v1.0

PHP Files: https://github.com/smitdylan2001/KM4-Redo/tree/main/DatabasePHP

SQL Dump: https://github.com/smitdylan2001/KM4-Redo/blob/main/DatabasePHP/DatabaseDump.sql

See Leaderboard: https://studenthome.hku.nl/~dylan.smit/Database/get_leaderboard.php

Demo Sign in:

test@test.nl – testPass

mark@test.nl – testPassMark

dyl@fhfh.com – passwords

dylly@smit.com - newpassword

Concept:

I tried to keep my concept very simple, complying with the minimal requirements from the assignment. This was to reduce stress next to Context III, which is a full-time project.

My game is playable with 2-4 people, but scalable to as many users as needed.

The core gameplay loop is based on reaction time. 1 player at a time gets to see a grid of buttons. You must press the buttons as fast as possible. If you do this 3 times you get to see your combined time of the button presses. Now the next player will do the same until all players are done.

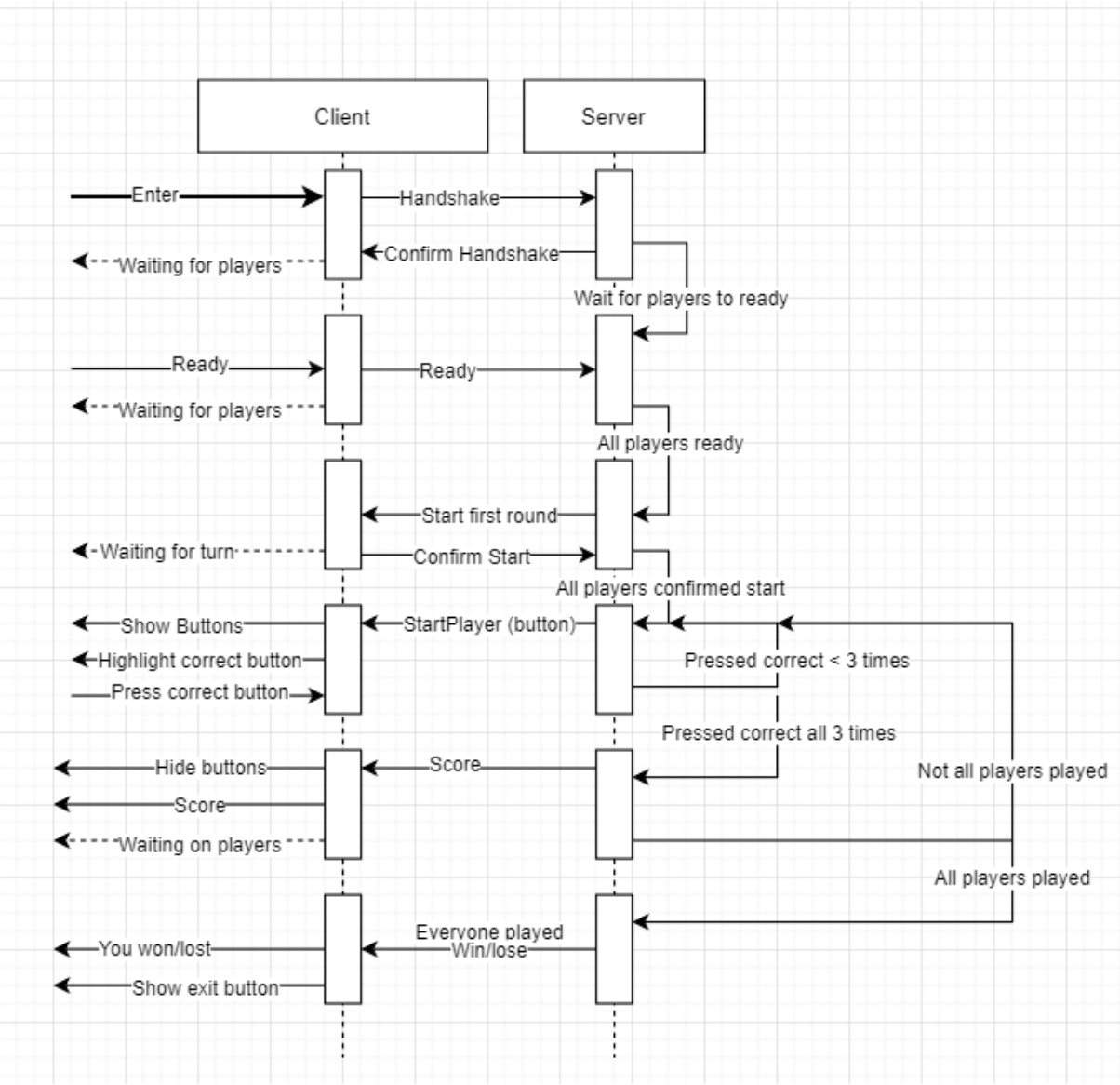
Game scores will be uploaded to an online leaderboard in milliseconds (since the database stores scores in integers). This is visible via URL.

Game Flow:

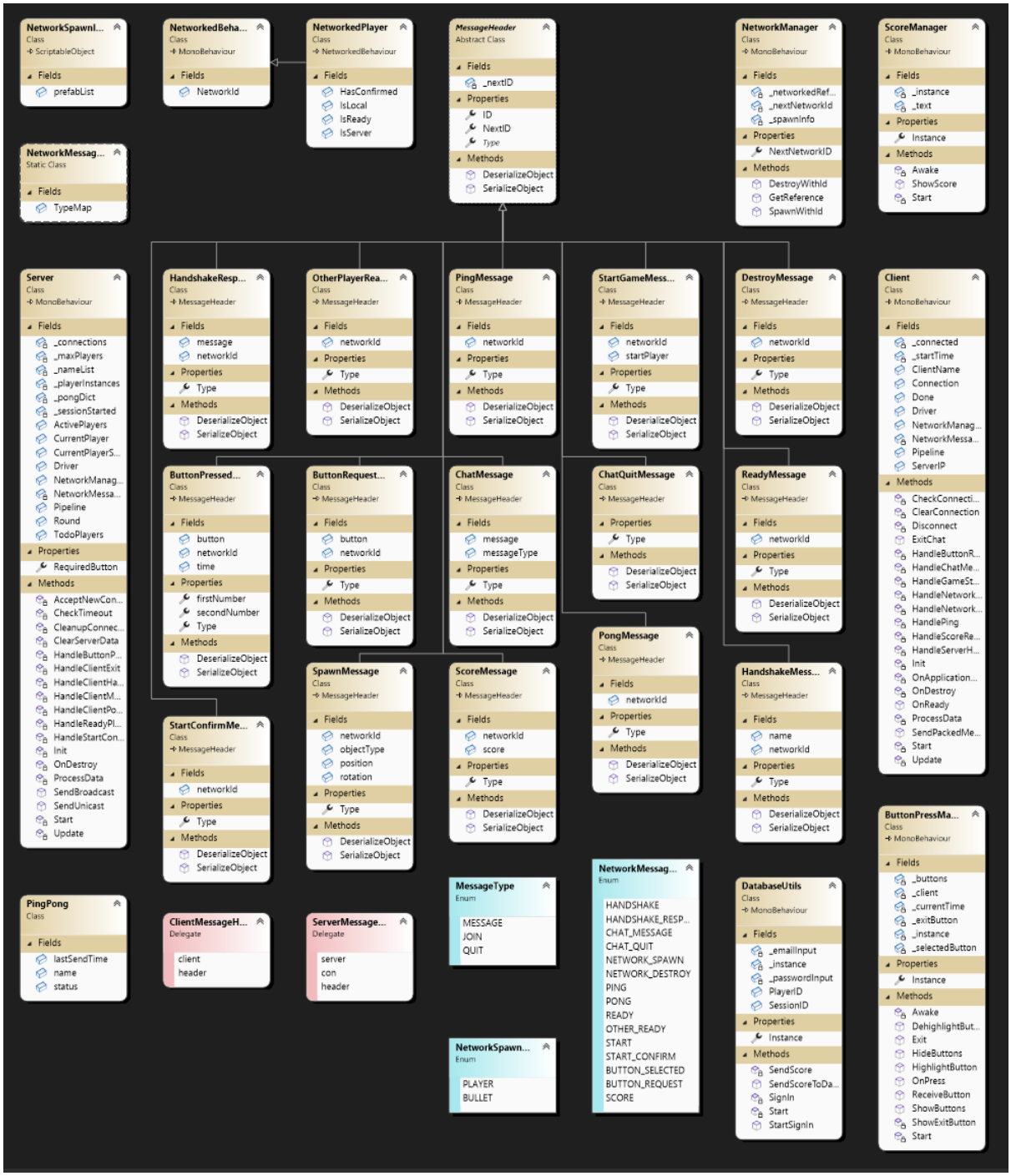
- Start Game
- See sign in screen
    - Sign Up button
    - Fill in info in sign up screen
    - Press submit
- Sign in
- 1 person press host (can still play), 1-3 press client (can play)
- Click ready when everyone is in
- Get grid of buttons when it's your turn
- Required pressed button turns black
- Turn black button
- Repeat last 2 steps 2 times
- Get timed score when done
- Wait for all players to finish
- Click exit to return to sign in screen
- Return to step 1

Diagrams

Message Flow:



Class Diagram:



Technological Process:

The whole networking setup is made using the Unity Transport API 1.0.0.
The basic setup can be seen in the Transport Personal Setup folder in the GitHub Repository. This system was made from scratch to make the Transport API more understandable for me.

To speed up my development process I left this setup alone in favor of the basic chatclient setup from Canvas. I do want to include this in the repository since I spent a lot of time on this system. This system mostly lacked the ability to properly send messages to desired clients, which worked flawlessly in the chatclient setup. During the process of making my game in it I properly understood how the new system was set up and changed the system to my personal preference.

I used the NetworkedPlayer to store variables per player, like isReady and hasConfirmed.

All my network messages are set up using the MessageHeader base class to streamline the message (de)serialization on both client- and server side.

Most Input is handles via UI Buttons, managed by the ButtonPressManager script.

The connection between Unity and the PHP SQL database is handles via DatabaseUtils. Here I use UnityWebRequests to send and receive information from and to the database.

I added a sign-up system inside of unity for new users. This is not included in the diagrams!

Reflection:

P	M	I
I got the Unity side working faster than I expected once the first few messages were working	I had a really hard time starting my Unity networking setup, because all guides online have a very basic setup, which was not enough to start developing further, and the demo Unity project on Canvas was a few steps further than I understood, which made it harder to write my own code for it	I noticed I prefer work which is between Programming and Designing. This course was challenging because it had decently low-level code, which I didn't enjoy working with
I did not have much difficulty with PHP and SQL	I did not enjoy working with the transport API	
Great to have physical lessons again	The requirements for a pass are different between the Unity and PHP assignments in Canvas (new user sign up is a requirement on the unity part, but not php part)	
	Context III took up a lot of my time, since it is a full-time project. I missed a lot of lessons because of this	