

Miikka Räsänen

Mobiiliohjain Linna Games Oy:lle

Tradenomi
Tietojenkäsittely
Syksy 2020



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä(t): Räsänen Miikka

Työn nimi: Mobiiliohjain Linna Games Oy:lle

Tutkintonimike: Tradenomi

Asiasanat: Unity, mobiiliohjain, peli, lähiverkko, unet, android

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa mobiiliohjain Linna Games Oy:lle heidän nykyiseen projektiinsa. Työssä käytiin läpi erilaisia tapauksia, missä mobiiliohjain voisi toimia ja vertailla sen hyviä ja huonoja puolia, kuten asioita, joita ei löydy normaalista ohjaimesta, mutta löytyisi mobiiliohjaimesta.

Mobiiliohjainta käytetään mobiililaitteella, kuten älypuhelimella tai tabletilla. PC:llä toimii itse peli, johon mobiiliohjaimella yhdistetään WiFi:n kautta, PC-peli on itse myös palvelin. Koska peli on käynnissä PC:llä, tapahtuu kaikki grafiikka ja laskenta siinä samalla ja mobiiliohjain yritetään pitää mahdollisimman kevyenä, niin sitä voi käyttää vanhemmillakin malleilla. Tässä opinnäytetyössä tehtiin mobiiliohjain vain Android-käyttöjärjestelmälle.

Linna Games Oy:llä oli peliprojekti, joka toimi langattomilla bluetooth-ohjaimilla. Tämä systeemi piti saada vaihdettua mobiiliohjaimiin, jotta päästäisiin eroon omista ohjaimista. Suunnittelussa lähdettiin siitä oletuksesta, että pelaajilla on käytössään älypuhelin. Tarkoituksena oli siis tehdä mobiililaitteelle sovellus, jonka kautta tätä peliprojektia pystytään pelaamaan. Mobiiliohjain tehdään Unity-pelimoottorilla.

Peliprojekti koostuu monesta pienestä pelistä, jotka on tarkoitettu olemaan nopeita ja yksinkertaisia pelattaessa. Jokaiseen pienempään peliin tehtiin toimiva mobiiliohjain ja havainnollistettiin niiden toiminta kunkin pelin kohdalla. Suurimassa osassa on käytetty liikkumisessa kiihtyvyyssanturia tai virtuaalista ohjaussauvaa. Toiminnot tehdään painamalla virtuaalisia painikkeita.

Abstract

Author(s): Räsänen Miikka

Title of the Publication: Mobile Controller for Linna Games LLC

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: Unity, mobile controller, game, wifi, unet, android

The purpose of this Bachelor's thesis was to plan and implement a mobile controller for Linna Games LLC to their current project. The work examined different cases where a mobile controller could work and compared its pros and cons, such as things not found in a normal controller but found in a mobile controller.

The mobile controller is used with a mobile device such as a smartphone or tablet. The game itself runs on a PC, which is connected to the mobile controller via WiFi, the PC game itself is also a server. Since the game is running on a PC, all the graphics and calculations take place in it at the same time. Thus the mobile controller was made to be as light as possible, so it can be used with older models, as well. In this thesis, a mobile controller was made only for the Android operating system.

Linna Games LLC had a game project that worked with wireless Bluetooth controllers. This system had to be switched to mobile controllers in order to get rid of its own controllers. The design was based on the assumption that players have a smartphone. The intention was, therefore, to make an application for the mobile device through which this game project can be played. The mobile controller was made with a Unity game engine.

The game project consists of many small games designed to be fast and simple to play. A working mobile controller was made for each smaller game and their operation was shown for each game. Most games use an accelerometer or a virtual joystick to move around. Actions are performed by pressing the virtual buttons.

Sisällys

1	Johdanto	1
2	Unity	2
2.1	Unityn ominaisuuksia	2
2.2	Unity 5.1 -version verkkojärjestelmä	3
3	Android	5
3.1	Mikä on Android.....	5
3.2	Android SDK.....	5
4	Mobiiliohjain.....	6
4.1	Mikä mobiiliohjain on.....	6
4.2	Mobiiliohjaimen etuja	6
4.3	Mobiiliohjaimen haittoja.....	7
4.4	Miten aiotaan käyttää mobiiliohjainta projektissa	7
5	Toimivuus palvelimen ja asiakasohjelman välillä	9
5.1	Yhdistäminen.....	9
5.2	Avainkoodi.....	10
5.3	Kanssakäyminen palvelimen ja asiakasohjelman välillä.....	11
5.4	Moninpeli	12
5.4.1	Yhdistäminen ja oman pelaajaidentiteetin saaminen.....	12
5.4.2	Yhteyden katkaisu ja uudelleen yhdistäminen	12
5.5	Palvelimen uudelleenkäynnistys	13
6	Mobiiliohjain toiminnassa peleissä	14
6.1	Ready Check	14
6.2	Dare Roulette	15
6.3	RO_BUGS.....	15
6.4	Neon ²	17
6.5	Quick Draw	18
6.6	Mad Bash.....	19
6.7	Hockey Boys	20

7	Yhteenveto ja jatkokehitys	21
	Lähteet	22

Symboliluettelo

Android	Käyttöjärjestelmä, jota käytetään mobiililaitteissa.
API	Application Programming Interface eli ohjelmointirajapinta.
Asiakasohjelma	Sovellus, jolta pelaaja käyttää mobiiliohjainta mobiililaitteella.
Bluetooth	Langattomaan yhdistämiseen käytettävä systeemi mobiililaitteilla, tietokoneilla ja muilla elektronisilla laitteilla.
C#	Ohjelmointikieli, jota käytetään Unityssä.
D-pad	Peliohjaimen ristiohjain, jolla liikutaan pelissä.
Debuff	Vastustajaan käytettävä voima, joka heikentää vastustajan pelaamista.
Input	Tarkoitetaan komennon syöttämistä laitteelle, yleensä nappulan muodossa.
Joystick	Peliohjaimen ohjaussauva, jolla liikutaan pelissä.
Network	Tietoverkko, jonka kautta kaikki viestien lähettäminen tapahtuu.
Palvelin	Palvelimeen yhdistetään mobiililaitteet, ja se jakaa tietoa asiakasohjelmille.
Scene	Ikkuna Unityssä, josta näkee kameran kautta, miltä peli näyttää.
SDK	Software Development Kit eli asennettava kokoelma ohjelmistokehityksen työkaluja.
Tilt	Ohjaimen kallistaminen.
UI	User interface eli käyttöliittymä.
Unity	Pelimoottori, jolla peli tehdään.
Unity hub	Unityn versioiden hallintasovellus, jolla myös hallitaan Unity projekteja
Unet	Unityn oma tietoverkkosysteemi.
WiFi	Langaton tietoverkkoprotokolla, jota käytetään lähiverkon luomiseen.

1 Johdanto

Opinnäytetyö on toimeksianto Linna Games Oy:ltä. Linna Games Oy on perustettu vuonna 2019 Kajaanissa. Yritys tekee yhteistyötä pienien ja keskikokoisten tiimien kanssa, jotka tuottavat sisältöä sekä sovelluksia asiakkaiden tarpeisiin.

Linna Games Oy:llä on asiakkaalle peliprojekti, joka on tehty toimivaksi fyysisten ohjainten kanssa. Pelissä on useita pieniä pelejä, jotka ovat yksinkertaisia ja nopeita pelata. Jokaisessa pienessä pelissä on oman tyylinen tapa ohjata, ja kaikkia näitä on vaikea tehdä fyysisen ohjaimen kanssa. Tarkoituksena on rakentaa Android-laitteeseen mobiiliohjain-sovellus, jolla voidaan ohjata tätä nykyistä peliprojektia. Mobiiliohjain rakennetaan käyttäen Unity-pelimoottoria ja tämän sisältämää Unet-verkkojärjestelmää.

Aluksi käydään läpi, mitä Unity ja Unet ovat ja miten niitä käytetään hyväksi tässä opinnäytetyössä. Näiden jälkeen esitetään katsaus itse mobiiliohjaimeen, mikä se on, mitkä ovat sen hyvät ja huonot puolet, verrataan normaaliin peliohjaimeen ja miten mobiiliohjainta käytetään hyväksi peliprojektissa.

Opinnäytetyössä esitellään, miten kommunikointi toimii asiakasohjelman ja palvelimen välillä, miten tieto siirtyy mobiiliohjaimesta itse peliin. Miten yhdistys tapahtuu, miten moninpeli toimii ja mitä siinä pitää ottaa huomioon, että kaikki toimii kuten pitäisi.

Työssä kuvataan myös, miten mobiilipeliohjain toimii pelien kanssa, joita varten ohjain on tehty. Mitä ohjaustyyliä käytetään missäkin pelissä ja miten kyseistä peliä pelataan sillä tyyllillä.

2 Unity

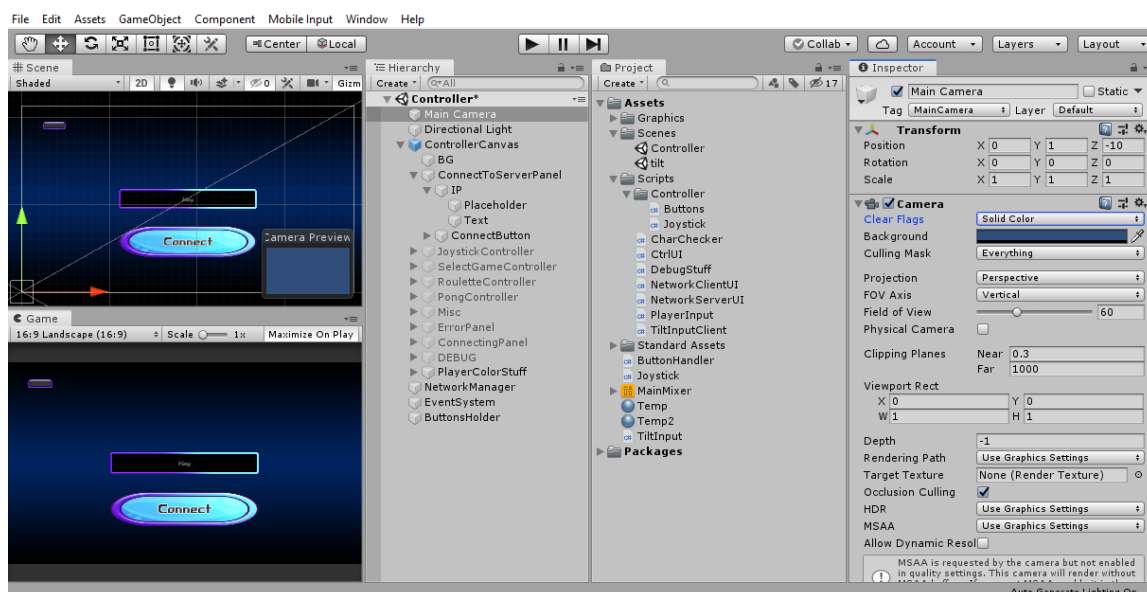
2.1 Unityn ominaisuuksia

Unity on pelimoottori, jolla voidaan tehdä 3D- ja 2D-pelejä tai sovelluksia seuraaville alustoille: PC, Playstation 3, Playstation 4, Xbox 360, Xbox One, Wii, WiiU, Android, iPhone sekä internetselaimelle. Unity sisältää graafisen käyttöliittymän, jolla on yksinkertaista tehdä nopeasti prototyypppejä ja kunnollisia tripla-A-pelejä. Käyttöliittymä on hiottu erittäin helpoksi myös niille, jotka ovat uusia pelialalla. [1.]

Unityssä on erikseen ikkuna, josta näkee suoraan, millaiselta peli näyttää kameran kautta pelaajalle. Tätä kutsutaan Scene-ikkunaksi, ja se helpottaa tekijää asettelemaan asioita tarkasti, jotta peli näyttää hyvältä. Peliin voi lisätä suoraan kansioista Scene-ikkunaan objekteja, jotka sitten erikseen näkyvät käyttäjälle hierarkialistassa.

Unityssä pystyy painamalla play-nappulaa milloin vain testaamaan, miten peli toimii. Pelin pysäytys tapahtuu samasta nappulasta, kun peli on päällä. On myös mahdollista keskeyttää peli, kun peli on päällä, jotta käyttäjä voi tarkastella, miltä peli pysäytetyllä hetkellä näyttää.

Unity on tehty C- sekä C++-ohjelmointikielillä, ja se tukee C#- sekä Javascript-ohjelmointikieliä [2]. Kuvassa 1 on kokonaiskuva Unityn käyttöliittymästä.



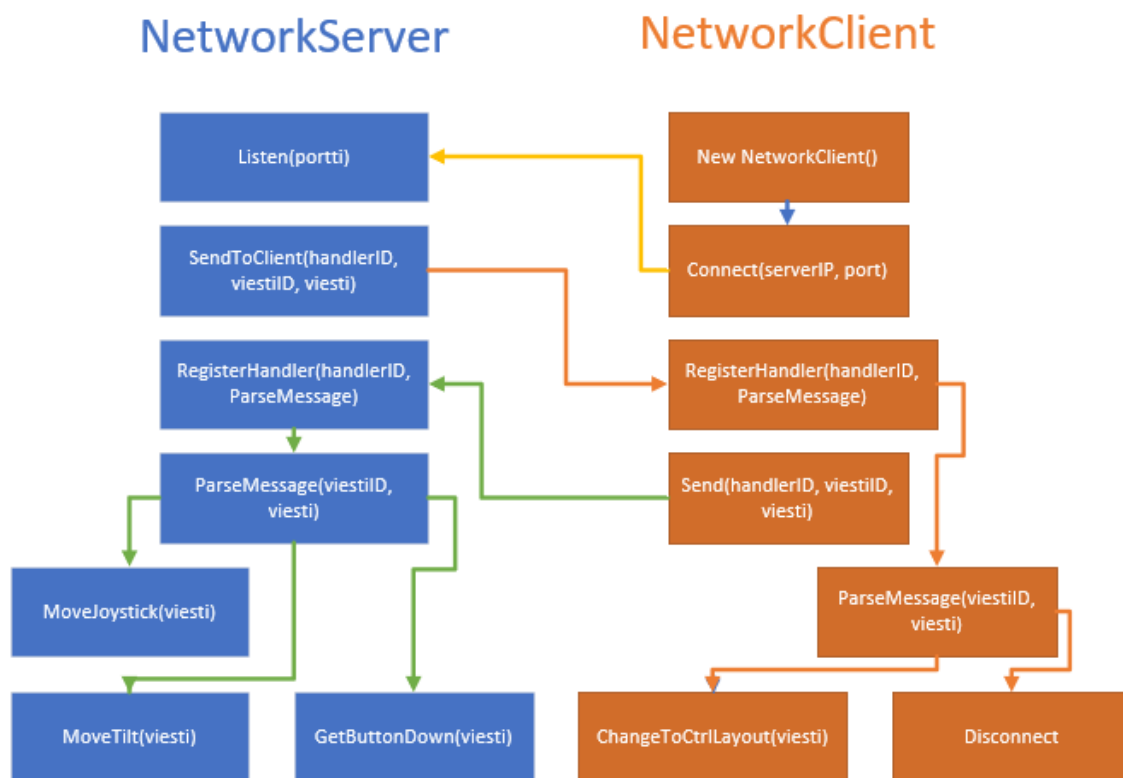
Kuva 1. Unityn käyttöliittymä.

2.2 Unity 5.1 -version verkkojärjestelmä

Unet on Unity 5.1-version uusi ja paranneltu verkkojärjestelmä, joka on joustavampi ja tehokkaampi kuin aiemmissa versioissa toiminut verkkojärjestelmä. Unet on poistumassa käytöstä vuonna 2021, jolloin on tulossa korvaava järjestelmä. Tässä työssä on käytetty Unetia, koska sen kautta voidaan nopeasti tehdä tarpeeksi hyvä kommunikaatio asiakasohjelman ja palvelimen välille käyttäen WiFi-yhteyttä.

Unet koostuu korkean ja alemman tason API:sta eli ohjelmointirajapinnasta, joilla on omat funktionsa Unetin pyörittämisessä [3]. Unet sisältää paljon erilaisia ominaisuuksia, ja näistä on tehty myös ihan omia opinnäytetöitään. Koska tässä projektissa riittää, että pelkästään asiakasohjelma ja palvelin voivat muodostaa yhteyden toisiinsa ja lähettää merkkijonoja molempiin suuntiin, käytetään vain muutamaa korkean tason API-luokkaa, joihin keskitytään tässä osiossa.

NetworkServer on korkean tason API-luokka, joka hallinnoi asiakasohjelmien yhteyksiä. Tätä luokkaa käytetään aloitettaessa kuuntelemaan uusia yhteyksiä tietyltä portilta lähiverkossa ja asetetaan rekisterin käsittelijä ohjaamaan sille osoitetut viestit sen käsittelemään funktioon. Kun palvelin ottaa vastaan viestin asiakkaalta, käydään ensimmäiseksi läpi, minne viesti on tarkoitettu ja lähetetään se oikeaan osoitteeseen, tai funktioon tässä tapauksessa. Kyseinen funktio käy viestin lävitse ja jäsentää sen luettavaan muotoon. Kun viestistä on saatu tarvittavat tiedot irti, voidaan niitä käyttää pelin ohjaukseen. [4.] Kuvassa 2 näkyy, miten interaktio asiakasohjelman ja palvelimen välillä tapahtuu.



Kuva 2. Asiakasohjelman ja palvelimen interaktio.

NetworkClient on toinen korkean tason API-luokka, jota käytetään aloittamaan ohjelmalla toiminto, jolla ohjelmasta tulee asiakas ja otetaan yhteys palvelimeen. Kun yhteys on muodostettu, asiakasohjelma pystyy lähettämään ja vastaanottamaan viestejä palvelimelta [5]. Asiakasohjelman tärkein funktio on pystyä lähettämään tietoa siitä, miten pelaaja ohjaa ohjainta. Jokainen kallistuksen kulma lähetetään eteenpäin palvelimelle, jotta pystytään tarkasti antamaan ohjeet, miten pelaajan ohjaama peli käyttäytyy. Viestejä palvelimelta asiakasohjelmaan lähetetään tarkistamaan, onko asiakas vielä paikalla, jotta voidaan määrittää oikea ohjaimen ulkoasu käyttöön tai käskä katkaisemaan yhteys, jos siihen on tarvetta.

Nämä ovat ainoat funktiot, joita tarvitaan Unetiltä käyttöön tässä projektissa.

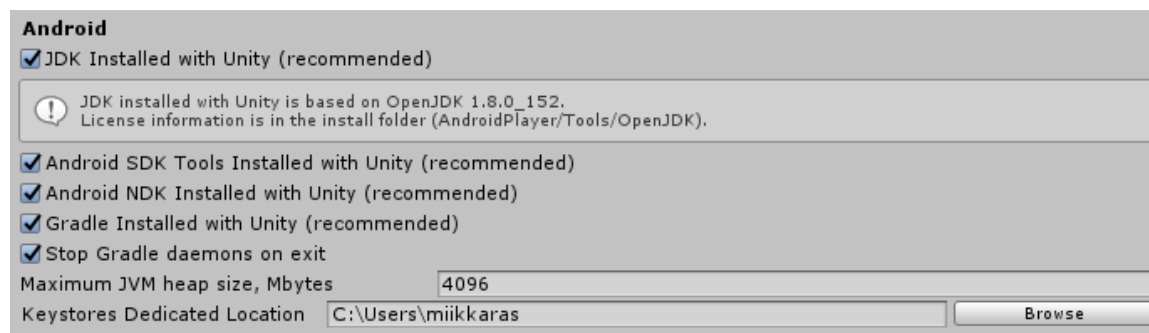
3 Android

3.1 Mikä on Android

Android on avoimeen lähdekoodiin perustuva käyttöjärjestelmä, joka on tarkoitettu pienenä ja kevyenä käytettäväksi älypuhelimiin, tabletteihin ja televisioihin [6]. Android valittiin, koska toimeksiannon kohteena olevat testilaitteet käyttävät Androidia. On myös mahdollista kääntää ohjelma iOS-käyttöjärjestelmälle, mutta tämän opinnäytetyön puitteissa keskitytään Android-pohjaan.

3.2 Android SDK

Unity tarvitsee Androidilta SDK-työkalut, jotta Unity pystyy rakentamaan sovelluksen Android-laitteelle [7]. Nykyään voidaan Unity Hubin kautta asentaa erikseen nämä tarvittavat työkalut oikeine versioineen. Ennen tämä prosessi vei enemmän aikaa, koska piti etsiä oikeat versiot itse ja eri versiot eivät aina toimineet keskenään. Kuvassa 3 näkyy asetukset, joita Unity tarvitsee, että Android-kehitys onnistuu.



Kuva 3. Android-asetukset Unityssä.

4 Mobiiliohjain

4.1 Mikä mobiiliohjain on

Mobiiliohjaimella tarkoitetaan ohjainta, joka on periaatteessa sovellus kosketusnäyttöisessä mobiililaitteessa, kuten esimerkiksi älypuhelimessa. Ohjelma itsessään antaa kosketusnäytölle esimerkiksi virtuaalisen ohjaussauvan liikkumiseen ja nappuloita eri toimintojen käyttämiseen. Mobiiliohjaimessa on periaatteessa loputon määrä vaihtoehtoja, miten sitä käytetään, eli sen ei tarvitse olla aina perusohjaussauvalla ja napeilla varustettu. Mobiiliohjainta voi periaatteessa käyttää melkein minkä vain pelin pelaamiseen, jos siihen voi implementoida ohjaimen, mutta vähänkin toimintapainotteisissa peleissä tämä ei ole kovin hyvä ratkaisu, sillä kosketusnäytön kautta pelaaminen on hankalampaa ja enemmän viivettä aiheuttavaa kuin peliohjaimella pelaaminen. Kuvassa 4 on erittäin yksinkertainen mobiiliohjain ohjaussauvalla ja kolmella nappulalla varustettuna.



Kuva 4. Yksinkertainen mobiiliohjain ohjaussauvalla ja nappuloilla.

4.2 Mobiiliohjaimen etuja

Hyvä puoli mobiiliohjaimessa nykypäivänä on se, että melkein kaikki omistavat jonkinlaisen mobiililaitteen, oli se sitten älypuhelin tai tabletti, jossa on Android-käyttöliittymä. Jos haluaa

liittyä mukaan yhdessä pelaamiseen, tarvitsee vain ottaa oma laite ja ladata mobiiliohjain-sovellus sille ja liittyä peliin.

Peleissä mobiiliohjaimen hyviä puolia on se, että jokaisella on oma ruutunsa. Sinne voi asettaa tietoa, jota ei tahdo jakaa muiden pelaajien kanssa tai oman näkymän pelipaikalta yms. Myös UI eli käyttöliittymä voidaan asettaa mobiilinäyttöön, ettei PC-näytöllä ole liikaa asiaa estämässä pelin näkymää. Erilaisiin skenaarioihin voi vaihtaa sitä vastaavan UI:n.

Voidaan myös käyttää mobiililaitteen omia ominaisuuksia pelien pelaamiseen, kuten kiihtyvyyssanturi, jolla pystytään saamaan arvo, missä kulmassa ohjain on kädessä ja nämä arvot voidaan yhdistää peliin käytettäviksi erilaisiin tapauksiin.

4.3 Mobiiliohjaimen haittoja

Huonoja puolia mobiiliohjaimessa on, että useimmiten kun tekee muutoksia peliin, niin pitää myös mobiiliohjain päivittää. Päivittäminen voi olla ikävää, jos tahtoo testata monella laitteella yhtä aikaa, että kaikki toimii niin kuin pitäisi ja tämä myös hidastaa projektin tekemistä.

Kun mobiililaitteessa on näyttö koko ajan päällä pelatessa, kuluttaa se laitteen akun nopeasti tyhjäksi, jos se ei ole laturissa kiinni. Tämä rajoittaa pitempiä sessioita kavereiden kanssa, varsinkin jos on vanhempimallinen laite.

Mobiiliohjaimella pitää olla oma Unity-projekti erikseen, mikä tarkoittaa, että pitää muokata koko ajan kahta eri projektia. Varsinkin pienten virheiden korjaus voi olla ongelmallista, kun pitää siirtyä kahden projektin välillä.

4.4 Miten aiotaan käyttää mobiiliohjainta projektissa

Jokaiselle pelille tulee oma ohjaimensa. Vaikka olisi kaksi samalla tavalla ohjattavaa peliä, niin vaihdetaan vähintään teema ohjaimessa tämän pelin kaltaiseksi.

Ohjaimiksi tulevat joystick, D-pad ja tilt-ohjain. Joystick-ohjaimessa käytetään virtuaalista ohjaussauvaa, joka lähettää akselin arvot sen mukaan, kuinka pitkälle sitä vedetään tietyistä pisteistä. D-pad-ohjain on taas perusristiohjain, jolla annetaan myös liikkumisen akseli, mutta sellaisessamuodossa, että jokainen suunta on joko 1 tai 0, eikä välimuotoja ole. Tilt-ohjainta

kallistetaan, jotta saataisiin eri akselin arvot liikkumiseen, esimerkiksi vasemmalle kallistaminen liikuttaa vasemmalle.

Jokaisella pelillä on myös oma määrä nappuloita, joita niissä voi käyttää, joten jokaiseen ohjaimeen pitää katsoa, mitä nappuloita siihen sisällyttää, ettei tulisi turhia nappuloita tai liian vähän. Jos tahdotaan, että ohjaukset ovat helppoja, pitää ensimmäiseksi ottaa huomioon käden rajoitukset [8]. Tärkeää on tehdä ohjaimista helppokäyttöiset sekä oikea- että vasenkätisille.

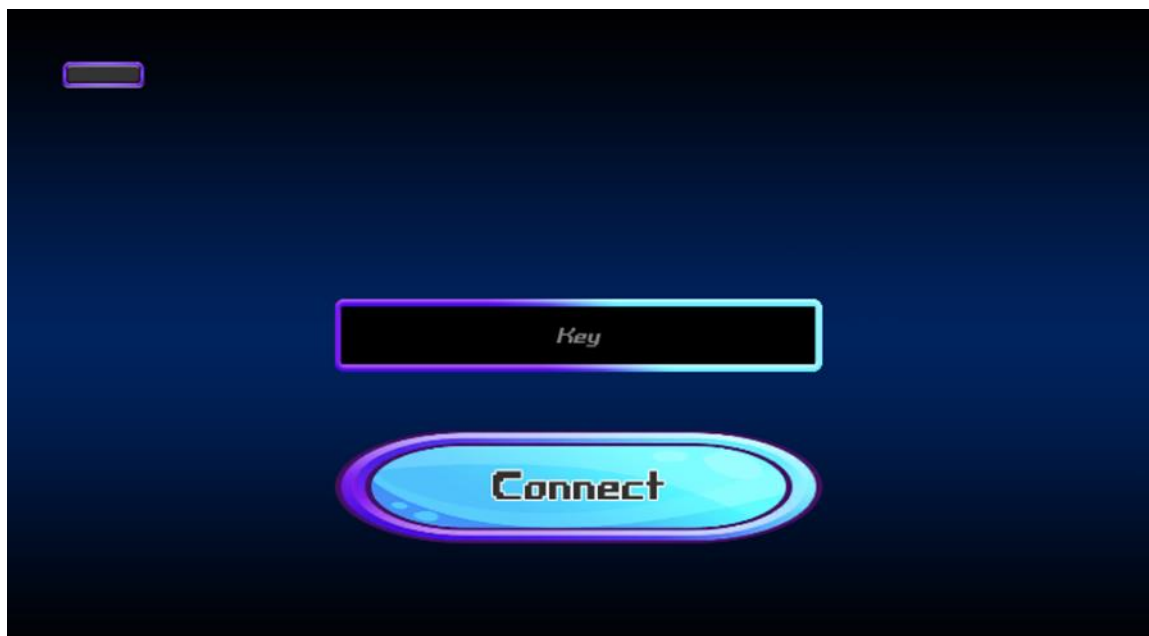
Lopussa käydään läpi kaikki pelit, joita tällä hetkellä on projektissa ja minkälaista ohjainta missäkin tilanteessa käytetään.

5 Toimivuus palvelimen ja asiakasohjelman välillä

Pelissä on itse palvelin, joka pyörittää kaikkea, mitä tässä pelissä tapahtuu. Tarkoitus on pitää mobiilipuoli erittäin kevyenä, ja kaikki graafiset ja laskennalliset asiat tapahtuvat PC-puolella. Jos tekee pelin, jossa on vain mobiiliohjaimet, voi sen testaaminen olla hankalaa, joten kannattaa erikseen asettaa myös nappulat ohjaimelle tai näppäimistölle testauskäyttöön. Jokaiseen skriptiin, joka käyttää verkko-ominaisuuksia, pitää lisätä Networking-nimiavaruus [9].

5.1 Yhdistäminen

Kun ohjelma on päällä, voi palvelimeen yhdistyä mobiiliohjaimella, jonka jälkeen ensimmäinen sisään yhdistynyt voi valita, mitä peliä pelataan. Palvelimeen voi yhdistää vain tietty määrä pelaajia kerralla; tällä hetkellä pienin määrä on yksi ja suurin määrä on neljä. Yhdistämistä varten pitää mobiiliohjaimen ja palvelimen olla samassa lähiverkossa. Yhdistäminen tapahtuu syöttämällä avainkoodi. Kuvassa 5 näkyy yhdistysruutu, missä syötetään avainkoodi, jonka jälkeen voi painaa "connect"-nappulaa, joka hoitaa yhdistämisen palvelimelle.



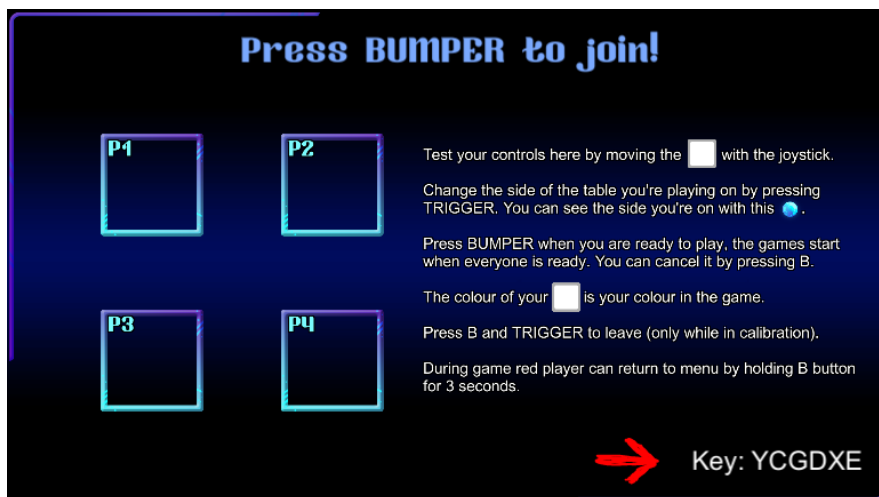
Kuva 5. Prototyypin yhdistysruutu mobiiliohjaimessa.

5.2 Avainkoodi

Yhdistämisen helpottamiseksi voidaan myös tehdä IP ja portin tilalle avainkoodi. Tämä mahdollistaa sen, ettei pelaajan tarvitse saada IP-osoitetta, vaan palvelimella näkyisi näytöllä jokin avain, minkä kautta voidaan yhdistää palvelimeen. Kuvassa 6 on mobiiliohjaimen aloitusruutu, jossa yhdistäminen tapahtuu syöttämällä avainkoodi, joka näkyy kuvassa 7, joka on pelipalvelimen aloitusruutu.



Kuva 6. Mobiiliohjaimen aloitusruutu, johon on syötetty avainkoodi palvelimelle.



Kuva 7. Pelin palvelimen aloitusruutu.

Vaikka kyseessä onkin vain kirjainsarja, niin oikeasti se on IP-osoitteen kaksi viimeistä osaa. Ohjelmassa on koodi, joka muuttaa IP-loppuosan kirjaimiksi, ja sama koodi on myös asiakasohjelmassa, joka taas muuttaa sen takaisin numeroiksi, mutta sitä itse pelaaja ei näe. Koska sekä asiakasohjelma että palvelin ovat samassa lähiverkossa, on molempien ensimmäinen

osa IP-osoitetta täysin sama, joten kun asiakasohjelmasta painaa yhdistä-nappulaa, ottaa se koodilla alkupään oman IP-osoitteen mukaan ja loppupään muuttaa avainkoodin kautta kokonaiseksi IP-osoitteeksi. Tämä estää pelaajia näkemästä palvelimen omaa IP-osoitetta, mikä lisää vähän palvelimen suojausta ja on myös kätevämpi tapa muutenkin yhdistää, kun pitää vain muutamia kirjaimia syöttää, eikä kokonaista IP-osoitetta. Palvelimen asetusten säätäminen oikeiksi on tärkeä osa suojausta tärkeämpää informaatiota [10].

5.3 Kanssakäyminen palvelimen ja asiakasohjelman välillä

Koska ei haluta turhaan kuormittaa viestintää palvelimen ja asiakasohjelman välillä, tehdään erilaiset ratkaisut erilaisiin tilanteisiin.

Kun kyseessä on vain perusohjaimen käyttö asiakasohjelmalla, pitää palvelimelle lähettää tietoa aina, kun painetaan jotain nappulaa, mutta itse palvelimen ei tarvitse tässä tilanteessa lähettää takaisin mitään. Tietysti perusasiat, kuten se, että pelin päättäminen ja aloittaminen pitää lähettää, mutta itse ohjauksen kanssa riittää lähetys vain palvelimelle päin.

Jos taas on kyseessä tilanne, että pelaajalla on oma pelikuva omassa ruudussaan, pitää siinä lähettää paljon enemmän tavaraa edestakaisin. Esimerkiksi jos vaikka toinen pelaaja pommittaa jonkin debuffin eli vastustajaa heikentävän ominaisuuden toiseen pelaajaan, voi pelaajan näytölle ilmestyä jotain ohjaamista häiritsevää tai muuta samantapaista. Tässä tilanteessa pitää lähettää dataa asiakasohjelmalle, että käynnistetään debuff-toiminto ruudulla ja voidaan lähettää takaisin tieto toiselle pelaajalle, että hän onnistui debuffin toimittamisessa.

Asioita, joita lähetetään palvelimelta asiakasohjelmalle:

- Mikä peliohjain tarvitaan nykyistä peliä varten
- Nykyisen ruudun ID, että osataan näyttää oikea ohjain
- Oma pelaajaväri
- Virhetapahtumat
- Katkaise yhteys, jos tarvetta

Asioita, joita asiakasohjelmalta lähetetään palvelimelle:

- Nappulan painaminen ja sen ID
- Ohjaussauvan akselin kulma
- Kiihtyvyyssanturin luvut
- Yhteyden katkaisu omasta tahdosta

5.4 Moninpeli

Moninpeli on tärkein ominaisuus tässä projektissa, ja kaikki on rakennettu sen ympärille, jotta se saataisiin hyvin toimimaan ja tuntumaan mukavalta pelaajille. Kun moninpeli on kyseessä, täytyy ottaa huomioon enemmän asioita kuin yhden ohjaimen yhdistämisessä.

5.4.1 Yhdistäminen ja oman pelaajaidentiteetin saaminen

Yhdistämisen yhteydessä pitää ottaa huomioon muutamia tärkeitä asioita. Kun pelaaja yhdistyy mobiiliohjaimella palvelimelle, tehdään hänelle automaattisesti oma pelaajaidentiteetti, johon annetaan palvelimessa playerId, mikä on pelaajan numero; numeroita voi olla enintään neljä. Pelaaja saa myös mobileID:n, joka on tunnus siihen, mikä on hänen mobiiliohjaimensa, että osataan lähettää oikeaan ohjaimeen viestiä muutoksista. Molempia näitä tarvitaan, koska kun pelaaja yhdistää, Unet networking tekee online-pelaajista oman listan, jota ei pääse niin helposti muokkaamaan, joten mobileID pitää huolen siitä, että lähetetään aina oikealle päätteelle viestit. PlayerID taas voi vaihtua pelaajilla, joten sitä ei voida käyttää viestien lähettämisessä.

5.4.2 Yhteyden katkaisu ja uudelleen yhdistäminen

Kuten aikaisemmassa kohdassa mainittiin, mobileID on tärkeä pitää yllä, jotta tiedetään, kenelle viestit menevät. Kun asiakasohjelma yhdistää uudelleen pelaajan yhteyden katkaisun jälkeen,

hakee se vanhan paikan Unet networking -listasta ja tämän ID pysyy aina samana. Tämän takia mobileID on tärkeä, jotta osataan aina palata oikealle kohdalle Unet networking -listassa.

Pelisession aikana tapahtuva yhteyden katkaisu on varsin haastava, jos halutaan, että pelaajan voi yhdistää takaisin, eikä jokin toinen laite pysty viemään hänen paikkaansa. Työssä päädyttiin tekemään lista, johon lisätään se tai ne pelaajat, joilta katkeaa yhteys. Tämän aikana peli pysähtyy, ettei peli etene sillä aikaa, kun yhteyden menettäneet pelaajat yrittävät yhdistää takaisin. Listan tarkoitus on ottaa näiden pelaajien järjestysnumero ylös, monentena he yhdistivät palvelimelle. Kun seuraava asiakasohjelma yrittää yhdistää, katsotaan listalta, onko kyseessä täysin uusi asiakasohjelma vai vanha, jolla on numero jo valmiiksi ja on valmis palaamaan. Kun kaikki pelaajat ovat yhdistäneet takaisin, voidaan peliä jatkaa.

5.5 Palvelimen uudelleenkäynnistys

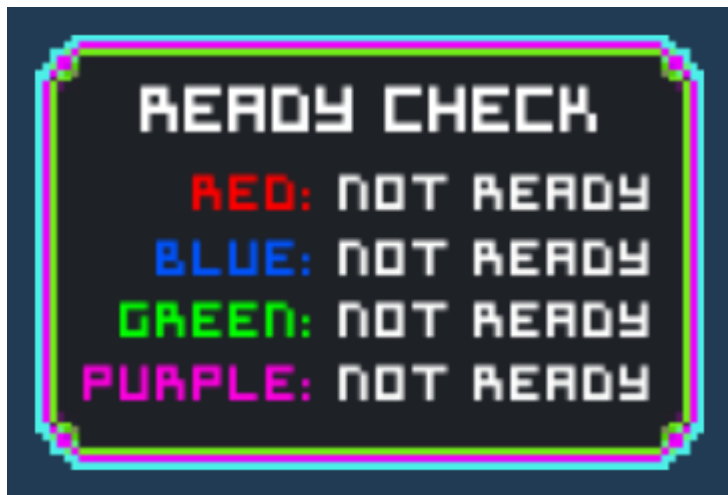
Koska kaikki ei aina mene, niin kuin pitäisi, peliin on lisätty palvelimen uudelleenkäynnistys toiminto, jos esimerkiksi jonkun mobiiliohjaimesta loppuu akku ja yhteys katkeaa kesken pelisession. Jotta ei tarvitse jäädä odottamaan, että kyseinen mobiililaite käydään asettamassa lataukseen ja pystytään jatkamaan peliä, on jokaisella asiakasohjelmalla mahdollisuus painella tietty koodi, joka käynnistää uudelleenkäynnistyksen, joka katkaisee yhteyden kaikilta pelaajilta ja nollaa koko palvelimen.

6 Mobiiliohjain toiminnassa peleissä

Lopputuloksena on oikein hyvin toimiva mobiiliohjain, jolla pystyy pelaamaan kaikkia projektin nykyisiä pelejä ja myös tulevia. Eri peleihin on erilaiset ohjaimet, jotta käyttäjille on mahdollisimman yksinkertaista tajuta, mitä mikäkin nappula tekee. Nappien nimeämiset muuttuvat pelien mukana, ettei pelaaja sekoita, mitä mikäkin nappula tietyssä tilanteessa tekee.

6.1 Ready Check

Ready check eli valmiustarkastus on jokaiseen peliin lisätty ominaisuus, joka tarkistaa pelin alkaessa, että jokainen pelaaja on varmasti paikalla ja valmis aloittamaan kyseisen pelin. Kun peli käynnistetään, hyppää ruudulle ikkuna, mistä kaikki näkevät, ketkä pelaajat ovat valmiita ja ketkä eivät. Mobiiliohjaimesta painamalla "Accept"-nappulaa voidaan ilmoittaa olevansa valmis pelin alkamiseen. Kuvassa 8 ready check toiminnassa.



Kuva 8. Ready check, jossa pelaajat eivät ole valmiita.

6.2 Dare Roulette

Dare Roulette on yksinkertainen ruletti, joka on tarkoitettu nopeaan juomapeliin shoteilla, pelin pelaaminen on vain ruletin pyörittämistä, joka tapahtuu painamalla vain yhtä nappulaa, joten yksi ohjain on tarpeen, vaikka olisikin useampi pelaaja. Kuvassa 9 näkyy Dare Roulette -peli käynnissä. Kuvassa 10 näkyy Dare Rouletten mobiiliohjain käytössä.



Kuva 9. Dare Roulette -peli käynnissä.



Kuva 10. Dare Roulette -pelin mobiiliohjain.

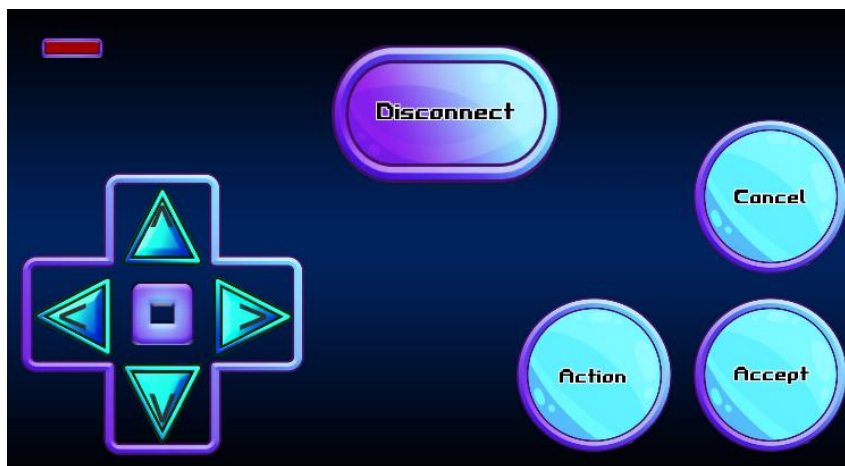
6.3 RO_BUGS

RO_BUGS on aika pitkälti kloonin pacmanista, mutta moninpelinä, tarkoittaen sitä, että jokainen pelaaja ohjaa omaa hahmoaan yhtä aikaa. Ensimmäiseksi aloittaa punainen pelaaja tai 1 ID-

pelaaja hyvänä ötökkänä, jonka tarkoitus on kerätä voimapilleri, jonka jälkeen hän voi syödä pahoja ötököitä.

Muut pelaajat ohjaavat pahoja ötököitä, jotka yrittävät saada syötyä pelaajan, kun hänellä ei ole voimapilleriä käytössä. Se paha ötökä, joka syö pelaajan pääsee itse tämän pelaajan tilalle ohjaamaan hyvää ötökkää.

Ohjaimet ovat siis kaikilla samanlaisia, mutta mobiiliohjaimessa näkyvä pelaajaväri kertoo mitä ötökkää pelaaja ohjaa. Ötököillä siis vaihtuu väri aina, kun niiden pelaajat vaihtuvat. Kuvassa 11 näkyy RO_BUGSin mobiiliohjain, vasemmassa yläkulmassa näkyy pelaajan väri.



Kuva 11. Kuvassa RO_BUGSin mobiiliohjain prototyyppissä.

6.4 Neon²

Neon on Pong-klooni, jossa pelikentän molemmilla puolilla on pelaajilla mailat ja yritetään estää saamasta palloa omaan maaliin. Tässä pelissä on käytössä myös voimia, joita käyttämällä pelaaja pystyy vahvistamaan itseään tai heikentämään vastustajaansa. Pelistä on myös nelinpeliversio, jossa kahden pelaajan sijasta neljä pelaajaa toimivat kahden pelaajan tiiminä toisiaan vastaan. Kuvassa 12 näkyy Neon²-peli käynnissä.



Kuva 12. Neon²-peli käynnissä.

Tässä pelissä käytetään normaalin ohjauksen sijasta tilt-ohjausta, joka tarkoittaa sitä, että pelaajat liikuttavat mailojaan kallistamalla mobiiliohjaintaan eri suuntiin, kallistettaessa nolla-arvo joko kasvaa tai laskee puhelimen kallistuskulman mukaan. Tämä ei perinteisesti ole käytössä normaaleissa ohjaimissa, joten tässä mobiiliohjain pääsee vahvuuksiinsa.

Toimintoja varten mobiililaitteen näytössä on interaktiivinen "action"-painike. Toimintoja ovat pelin aloitus, pallon syöttäminen ja erikoiskykyjen käyttäminen. Toinen pelissä käytettävä painike on "Action"-painikkeen yläpuolella sijaitseva "tilt reset"-nappula, jolla voidaan määrittää kallistuksen oletuskulma, joka toimii nollatasona. Kuvassa 13 näkyy pelin mobiiliohjain.



Kuva 13. Neon²-pelin mobiiliohjain, jossa nuolet kallistussuuntia.

6.5 Quick Draw

Quick Draw on nimensä mukaisesti peli, jossa yritetään reagoida vastustajaa nopeammin ampumaan. Peli toimii niin, että kentällä pelaajat ovat vastakkain lännen nopein -kisassa. Pelaajat odottavat signaalia, milloin saa ampua, ja silloin pitää painaa action-nappulaa mobiiliohjaimessa. Nopeampi pelaaja voittaa ja ruudulle tulee näkyviin pelaajien tulokset. Jos pelaaja ottaa varaslähdön ennen signaalia, niin hän häviää tämän erän. Kuvassa 14 näkyy Quick Draw -peli käynnissä.

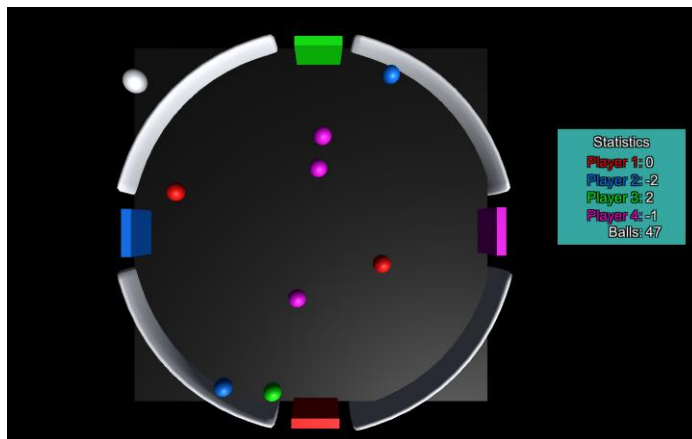


Kuva 14. Quick Draw -peli käynnissä.

Pelissä reagointiajan vertaaminen tapahtuu palvelimen kautta eikä mobiiliohjaimen sisällä, jotta saataisiin mahdollisimman nopeasti nähtyä reagointiaika pelissä. Tässä vaihtoehdossa mukaan tulee hiukan enemmän viivettä, mutta se vaikuttaa samoin molempiin pelaajiin.

6.6 Mad Bash

Mad Bash on neljän pelaajan peli, jossa pelaajat kilpailevat toisiaan vastaan koittaen suojella omaa maalia. Pyöreälle pelialueelle putoaa useita palloja tietyn ajan välein. Pelaajat voivat maalata pallot oman värinsä mukaan osumalla niihin mailalla, ja tällöin pallon mennessä toisen pelaajan maaliin pallon värjännyt pelaaja saa itselleen pisteen. Jokainen pelaajan omaan maaliin mennyt pallo vähentää pelaajalta pisteen, ja pelaaja, jolla on korkein pistetulos viimeisen pallon mentyä maaliin, voittaa pelin. Kuvassa 15 näkyy Mad Bash -peli käynnissä.



Kuva 15. Mad Bash -peli käynnissä.

Vaikka pelialue on pyöreä, pelin ohjaus on periaatteessa samanlainen kuin Neon²-pelissä, mutta tällä kertaa "action"-painikkeesta pelaaja pystyy käyttämään pulssi-kykyä, joka luo alueen pelaajan mailan ympärille, työntäen palloja voimalla pois päin.

6.7 Hockey Boys

Hockey Boys on kahdesta neljään pelaajalle tarkoitettu jääkiekkopeli, jossa tarkoitus on tehdä maali vastustajan maaliin. Pelissä on molemmilla joukkueilla kaksi hyökkääjää ja maalivahti, pelaajat pelaavat hyökkääjinä ja tietokone toimii maalivahteina. Hyökkääjät liikkuvat kallistettaessa mobiiliohjainta siihen suuntaan, mihin pelaaja tahtoo liikkua. Liikkumista pystyy nopeuttamaan pitämällä "Sprint"-painiketta pohjassa mobiiliohjaimessa, ja kiekon laukominen tapahtuu painamalla "Shoot"-painiketta; tätä painiketta voi pitää pohjassa ladatakseen kovemman lyönnin. Hyökkääjillä on kestävyys-palkki kertomassa, pystyykö pelaaja lataamaan laukomista tai luistelemaan nopeammin. Kuvassa 16 näkyy Hockey boys -pelin mobiiliohjaimen ulkoasu.



Kuva 16. Hockey Boys -mobiiliohjain.

7 Yhteenveto ja jatkokehitys

Itse ensimmäinen yhdistys mobiililaitteelta palvelimelle onnistui hyvinkin äkkiä ja pääsi nopeasti tekemään ja testaamaan. Yhden ohjaimen testikäyttöön ohjelmointi oli nopeata ja sujuvaa.

Koska nappulat on tehty yhdelle pelille, on niitä helppo käyttää uusiksi toiselle pelille suoraan kopioimalla.

Heti alkuun huomasin, että moninpeliin oli helppo nopeasti yhdistää monta pelaajaa, mutta itse pelaajien hallinnointi ei ollut niin helppoa. Heti kun yksi pelaaja yhdisti, lähti pois pelistä ja tuli takaisin, meni networking sekaisin, koska Unet yritti palauttaa vanhaan asemaan takaisin yhdistänyttä pelaajaa, mutta tätä ei ollut ja piti luoda uusi.

Pelkästään hallinnoinnin takia moni asia kesti viikkoja tehdä, jotta saisin valmiin ja toimivan tuotteen. Pitää ottaa niin paljon asioita huomioon, joita ei ennalta osaa arvata, ellei ole aikaisemmin tehnyt jotain network-pohjaista työtä.

Projektin kannalta aion jatkaa mobiiliohjaimen ominaisuuksien kehittämistä ja hioa tätä vielä paremmaksi ja toimivammaksi, kuin sen nykyinen versio on.

Lähteet

- [1] Brian Moakley. 2018. Unity Tutorial Part 1: Getting Started. Viitattu 19.2.2020, <https://www.raywenderlich.com/5589-unity-tutorial-part-1-getting-started>
- [2] Concept Art Empire. What is Unity 3D & What is it Used For. Viitattu 9.9.2020, <https://conceptartempire.com/what-is-unity/>
- [3] Unity Technologies. 2016. Introduction to a Simple Multiplayer Example. Viitattu 19.2.2020, <http://web.archive.org/web/20161006112931/https://unity3d.com/learn/tutorials/topics/multiplayer-networking/introduction-simple-multiplayer-example?playlist=29690>
- [4] Unity Technologies. 2019. NetworkServer. Viitattu 9.9.2020, <https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Networking.NetworkServer.html>
- [5] Unity Technologies. 2019. NetworkClient. Viitattu 9.9.2020, <https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Networking.NetworkClient.html>
- [6] Android Authority. 2020. What is Android? Here's everything you need to know. Viitattu 9.9.2020, <https://www.androidauthority.com/what-is-android-328076/>
- [7] FaiiD. How to Make an Android App Using Unity. Viitattu 19.2.2020, <https://www.instructables.com/id/How-to-Make-an-Android-App-Using-Unity/>
- [8] Andrew Dotsenko. 2017. Designing Game Controls. Viitattu 19.2.2020, https://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php
- [9] Renan Oliveira. 2019. How to Create a Multiplayer Game in Unity. Viitattu 19.2.2020, <https://gamedevacademy.org/how-to-create-a-multiplayer-game-in-unity/>
- [10] Andrew Kirmse, Chris Kirmse. 1997. Security in Online Games. Viitattu 14.9.2020, https://www.gamasutra.com/view/feature/131620/security_in_online_games.php