Mobile controller

Dokumentaatio

**Valmiit Scriptit**

## Joystick

UpdateVirtualAxes metodin loppuun lisätään rivi jossa kutsutaan SendJoystickInfo nimistä metodia jonka parametreiksi laitetaan -delta.x ja delta.y. Tämä metodi löytyy NetworkClientUI:sta.



## ButtonHandler

SetDownState ja SetUpState metodien sisään laitetaan rivi jossa kutsutaan samalla tapaa kuin Joystick scriptissä mutta omaa metodia nimeltä SendButtonInfo, parametreiksi se ottaa nimen (ja myöhemmin boolin)



## NetworkClientUI

Kyseisellä hetkellä tämä scripti hoitaa Clientin eli Mobiili controllerin yhteyden serverin kanssa, yhdistyy siihen ja kertoo milloin se saa inputtia.

**OnGUI**

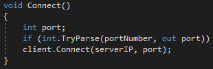
Tekee debug ui:n jotta voidaan muokata IP-osoitetta ja porttia mihin otetaan yhteys. Tekee myös napit joilla yhdistetään client serveriin ja lähetetään ensimmäinen viesti

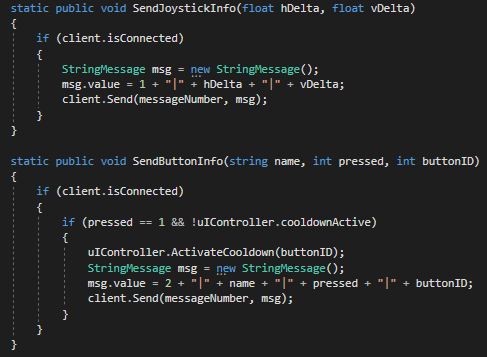
**Start**

Tehdään uusi client ja registeröidään viestinumerot joita kuunnellaan ja mitä tehdään, kun saadaan viestiä sitä kautta.

**Connect**

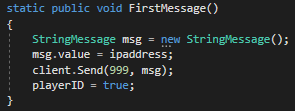
Otetaan yhteys serveriin sen jälkeen kun ollaan luettu pelaajan antama tieto ja muutettu se oikeaan muotoon jos tarpeellista.



**SendJoystickInfo** & **SendButton**

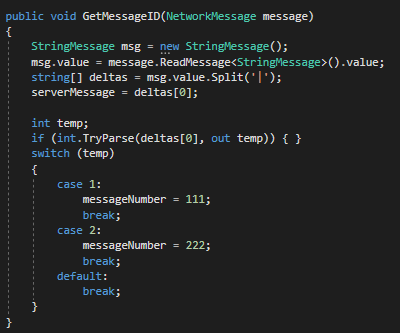
Jos client on yhdistynyt serveriin, se pakkaa tiedon joka eritellään toisistaan ”|” – merkillä. Paketin ensinmäinen numero on paketin id, jolloin katsotaan onko se Joystick – tietoa vai button-tietoa. Pelaaja yks saa viestin numerolla ”111”, pelaaja kaks numerolla ”222”, jne.

**FirstMessage**



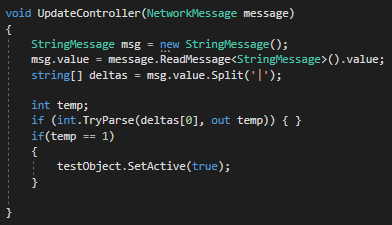
Lähetetään ensimmäinen viesti serverille joka sisältää oman IP-osoitteen

**GetMessageID**



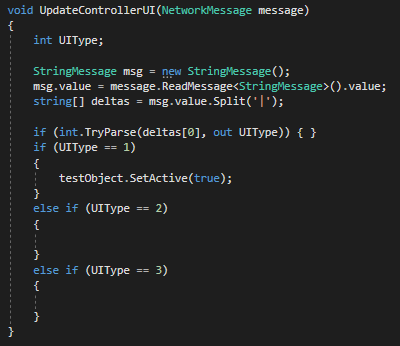
Tämä purkaa serveriltä saadun viestin joka kertoo mones pelaaja client on. Sitten päivittää oman viesti numeron jonka kautta client lähettää tulevat viestinsä

**UpdateController**

´

Tämä tehtiin testi tarkoitukseen, että testataan saadaanko clientilla peli objecti näkyviin serverin käskystä

**UpdateServerUI**



Aloitus siihen että client osaa ottaa vastaan viestiä serveriltä ja sen mukaan päivittään omaa User Interfacea

## NetworkServerUI

**OnGUI**

Näyttä serverin tietoja ja antaa mahdollisuuden muuttaa porttia jos on tarvetta, ilmoittaa oman ip:n. Oikealla ala laidassa on myös kaksi buttonia testi tarkoitusta varten. Toinen button lähettää viestin clienteille ja toinen tulostaa yhdistyneiden clienttien tiedot.

**Start**

Tekee jokaiselle pelaajalle oman virtuaalisen joystickin ja yhdistää sen omaan axiseen. Sitten aukaistaan oletus portti, jota kuunnellaan, rekisteröidään pelaajien viesti numerot.



**PlayerOne & PlayerTwo**

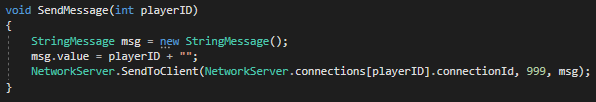
Käsittelevät tiedon jonka saavat, ensin purkaen sen ja sitten tarkistetaan ensimmäinen luku, joka on viestin id. Tällä katsotaan sisältääkö viesti button - vai joystick tiedon. Ja lähettää sen sitten eteenpäin oikeille scripteille ja metodille

**NewPlayer**



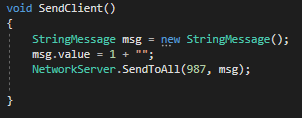
Kun serveri saa ensimmäisen viestin clientilta se käy tämän metodin läpi, joka määrittää monesko pelaaja client on. Tässä metodissa kutsutaan myös SendMessage metodia johon annetaan pelaaja numero

**SendMessage**



Tehdään viesti joka lähetetään clientille joka lähetti ensimmäisen viestin. Tämä viesti kertoo clientille mikä/mones pelaaja se on.

**SendClient**



Tämä tehtiin nopeaa testausta varten että nähdään voiko clientilla laittaa jonkin peli objektin näkyviin.

## GameObjectInput (CarInput)

GameStuff/Make input do something

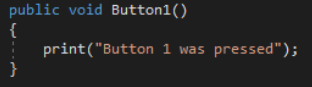
**Update**

Hae tieto CrossPlatforInputManager:lta



and do stuff with it

**Button1**, **Button2**, **Button3**…(muokattu lisäämällä booli jolla seurataan painetaanko sitä vai ei)



## Ohjeistus

**Add new button**

* Tee uusi button
* lisää button handler scripti
* Unityssa scriptille nimi ja id lisää Pelaajille metodi siihen
* lisää sen kutsuminen serverin pelaaja switchille

**Add new player**

* Tee uusi axis-pari ja nimeä ne
* Tee serverille uusi virtual axis
* Ja pelaajalle metodi sinne.

**UI-modding**