

Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2022

Melissa Laakso

Laajan ohjaintuen kehittäminen mobiilipeleille Unity-pelimoottorilla



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2022 | 31 sivua

Melissa Laakso

Laajan ohjaintuen kehittäminen mobiilipeleille Unity-pelimoottorilla

Mobiilipelejä tulee markkinoille paljon, ja kilpailu pelaajista on kovaa. Mobiilipelin ohjauksen täytyy tuntua pelaajasta miellyttävältä, jotta pelaaja palaa pelin pariin. Pelin ohjaaminen kosketusnäytöllä ei välttämättä kuitenkaan ole paras mahdollinen tapa suorituskyvyn kannalta. Moni pelaaja suosii peliohjaimen käyttöä mobiilipeleissä, ja tämä antaa suosiota mobiilipeleille, joissa on ohjaintuki.

Opinnäytetyön tarkoituksena oli selvittää, pystyykö Unity-pelimoottorilla luomaan mahdollisimman laajan ohjaintuen kustannustehokkaasti käyttäen pelimoottorin Input System -pakettia ja sen sisältämää Gamepad-luokkaa. Näillä työkaluilla toteutettiin testisovellus Android-laitteelle, joka näyttää visuaalisesti, jos se vastaanottaa signaalia yhdistetyltä Bluetooth-ohjaimelta. Testaukseen käytettiin Sony DualShock 4 -ohjainta, Sony DualSense-ohjainta ja Microsoft Xbox One -ohjainta.

Testisovellus vastaanotti signaalit oikein DualShock 4 -ohjaimesta ja Xbox One -ohjaimesta. Unity ei ole vielä kehittänyt mobiilialustoille tukea DualSense-ohjaimelle. Vaikka Input System -paketin Gamepad-luokka toimi ohjaintuen toteutukseen hyvin ja kehitys oli kustannustehokasta, pitäisi sen toimivuutta testata laajemmalla valikoimalla eri Bluetooth-ohjaimia, jotta saadaan tietää, kuinka laaja ohjaintuki sen avulla on mahdollista luoda.

Asiasanat:

videopelit, mobiilipelit, peliala

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and communications technology

2022 | 31 pages

Melissa Laakso

Developing an extensive controller support for mobile games with Unity game engine

There is an abundance of mobile games on the market, and the competition for players is fierce. The controls of a mobile game must feel pleasant to the player so that the player returns to the game. However, controlling the game with a touch screen may not be the best possible way in terms of performance. Many players prefer to use a game controller in mobile games, and this gives popularity to mobile games with controller support.

The purpose of the thesis was to find out if Unity can be utilized to create an extensive controller support cost-effectively using the game engine's Input System package and the Gamepad class it contains. These tools were used to create a test application for an Android device that visually displays if it receives a signal from a connected Bluetooth controller. A Sony DualShock 4 controller, a Sony DualSense controller, and a Microsoft Xbox One controller were used for testing.

The test application correctly received signals from the DualShock 4 controller and the Xbox One controller. Unity has not yet developed support for the DualSense controller for mobile platforms. Although the Gamepad class of the Input System package worked well for the implementation of controller support and the development was cost-effective, its functionality should be tested with a wider selection of different Bluetooth controllers to find out how extensive the controller support is.

Keywords:

video games, mobile games, game sector

Sisältö

1 Johdanto	6
2 Mobiilipelilaitteiden ja peliohjainten historia ja kehitys	8
2.1 Mobiilipelien syntyminen ja ohjaus fyysisillä näppäimillä	8
2.2 Mobiilipelien siirtyminen älypuhelimiin ja kosketusnäyttöohjaukseen	9
2.3 Peliohjainten historia ja kehitysprosessi	10
3 Peliohjaimen käyttö mobiilipeleissä	11
3.1 Peliohjaimia käyttävät mobiilipelit ja peligenret	11
3.2 Peliohjaimen käytön hyödyt verrattuna kosketusnäytöllä ohjaukseen	14
4 Työn kuvaus ja käytetyt teknologiat	16
4.1 Suunniteltu työprosessi ja sen tavoitteet	16
4.2 Käytetyt laitteet ja teknologiat	16
4.2.1 Unity-pelimoottori	16
4.2.2 Unity Input System -paketti	17
4.2.3 Android-käyttöjärjestelmä	17
4.2.4 Bluetooth-tiedonsiirtotekniikka	18
4.2.5 Käytetyt peliohjaimet	18
5 Testisovelluksen luominen ja eri ohjainten toimivuuden testaus sovelluksessa	19
5.1 Ohjainten tekniset erot ja yhdistäminen Android-laitteeseen	19
5.2 Ohjainten toimivuus testisovelluksessa	20
5.2.1 Testisovelluksen luominen	20
5.2.2 Ohjainten perusnäppäinten testaus testisovelluksessa	25
5.2.3 Ohjaimien erikoisnäppäimet	26
6 Tulokset ja pohdinta	27
Lähteet	29

Kuvat

Kuva 1. Peliohjaimen asennettava teline mobiililaitteelle. [10]	12
Kuva 2. Call of duty mobile pelin pelaajanäkymä. [11]	13
Kuva 3. Brawlhalla pelin pelaajanäkymä. [12]	14
Kuva 4. Työssä käytettävät ohjaimet.	18
Kuva 5. PlayerInputActions-objekti	21
Kuva 6. FlashImage-metodi.	22
Kuva 7. TriggerEvent-metodi.	23
Kuva 8. Sovelluksessa napin painallusta vastaavat ohjausnäppäimet.	24

1 Johdanto

Mobiilipelaaminen on suosittua ajanvietettä 2020-luvulla. Älypuhelimien ja tablettien yleistyminen on tehnyt mobiilipelaamisesta helposti saavutettavaa. Videopelejä on tehty jo vuosikymmeniä eri laitteille, ja pitkään niiden yhteinen tekijä oli pelien ohjaus jonkinlaisella fyysisellä peliohjaimella. Vaikkakin alkuun mobiilipelien ohjaus tapahtui laitteen numeronäppäimillä, nykyään mobiilipeleissä ohjaustapana käytetään kuitenkin monesti oletusarvoisesti laitteen kosketusnäyttöä, joka ei välttämättä ole aina pelaajalle paras mahdollinen tapa ohjaukseen. Kosketusnäyttö ei salli pelaajalle tarkkuutta ja tuntumaa, jonka fyysinen peliohjaus ja napit voivat tarjota. Monelle pelaajalle ohjaimen käyttö on myös tutumpaa kuin kosketusnäytöllä pelaaminen, mikä saattaa myös parantaa pelaajan suoritusta. Pelaajan suorituskkyky voi vaikuttaa oleellisesti siihen, nauttiiko pelaaja pelistä ja jatkaako hän pelin pelaamista. Tämän takia olisi tärkeää mahdollistaa pelaajalle optio käyttää fyysistä peliohjainta mobiilipelin ohjaukseen. Ohjaintuki löytyy tänä päivänä usein mobiilipeleistä, jotka ovat osa pelisarjaa, joka on aiemmin ilmestynyt jo konsolilla tai tietokoneella. Pelisarjojen pelaajat ovat jo tottuneet käyttämään fyysistä ohjainta tai näppäimiä pelin pelaamiseen, ja moni haluaa käyttää samaa ohjaustapaa myös pelin mobiiliversiossa. Myös tietyissä peligenreissä pelaajat saattavat olla tottuneempia käyttämään ohjainta kuin kosketusnäyttöä ohjaamiseen.

Erilaisia Bluetooth-yhteydellä mobiililaitteeseen kytkettäviä peliohjaimia on paljon. Tunnetuimpia valmistajia ovat esimerkiksi Sony, Microsoft ja Nintendo. Jos mobiilipelille luo ohjaintuen vain yhdelle tietylle ohjaimelle, voi peli menettää pahimmillaan pelaajia, sillä pelaajat eivät välttämättä ole valmiita hankkimaan uutta ohjainta pelin takia, jos pelaajalla on jo valmiiksi jonkun toisen valmistajan ohjaus. Isommat pelikehitysstudiot ovat pystyneet luomaan mobiilipeilleen kattavan ohjaintuen, joka mahdollistaa pelin pelaamisen mahdollisimman monella eri valmistajan ohjaimella. Itsenäisillä pelikehittäjillä ei kuitenkaan ole käytössään samanlaisia resursseja kuin isoilla pelikehitysstudioilla.

Useat pienet pelikehitysstudiot ja itsenäiset kehittäjät käyttävät pelikehitykseen Unity-pelimootoria. Opinnäytetyössä halutaan tutkia itsenäisten kehittäjien kannalta, onko Unity-pelimootorilla ja sen sisältämän Input System -paketin Gamepad-luokan avulla mahdollista luoda kustannustehokkaasti kattava ohjaintuki mobiilipelille. Opinnäytetyössä luodaan testisovellus näillä työkaluilla Android-laitteelle. Testisovelluksen on tarkoitus näyttää visuaalisesti, jos se vastaanottaa signaalia laitteeseen kytketyltä Bluetooth-ohjaimelta.

Opinnäytetyössä käytetään Sonyn DualShock 4 -ohjainta, DualSense-ohjainta ja Microsoftin Xbox One -ohjainta. Opinnäytetyössä vertaillaan kevyesti käytettäviä ohjaimia, jotta selviää, onko niissä eroja, jotka saisivat kehittäjän suosimaan tiettyä ohjainta.

Luvuissa 2 ja 3 käydään läpi mobiilipelien ja peliohjainten historiaa ja selvitetään, onko mobiilipelien ohjaus peliohjaimella hyödyllisempää kuin kosketusnäytöllä. Luvussa 4 käydään läpi työprosessi ja työssä käytettävät teknologiat. Luvussa 5 vertaillaan käytettäviä ohjaimia, luodaan testisovellus Unity-pelimootorilla ja testataan ohjainten toimivuus testisovelluksessa. Luvussa 6 käydään läpi työn tulokset.

2 Mobiilipelilaitteiden ja peliohjainten historia ja kehitys

2.1 Mobiilipelien syntyminen ja ohjaus fyysisillä näppäimillä

Ensimmäinen puhelin, jolla oli peli, oli Hagenuk MT-2000. Peli oli versio Tetriksestä ja se ilmestyi vuonna 1994. Mobiilipelit tunnetuksi teki Nokia vuonna 1997, kun Nokia 6110 -puhelimelle julkaistiin Snake. Snake oli myös ensimmäinen moninpeli, sillä peliä pystyi pelaamaan yhdessä kahdella laitteella infrapunan välityksellä. [1] Snaken inspiraationa toimi vuonna 1976 Gremlin Interactiven julkaisema peli Blockade. Blockade oli kaksinpeli, jossa pelaajat ohjasivat omia käärmeitään. Snaken kehittäjä Taneli Armanto on kertonut, että alun perin Nokia 6110 -puhelimelle oli kehitetty peliksi jo hyvin tunnettu Tetris, mutta koska Tetriksen oikeudet omistava yritys olisi halunnut jokaisesta pelin sisältävän laitteen myynnistä osion itselleen, Nokia hylkäsi idean ja tilalle kehitettiin Snake. Snake sai jatkoa vuonna 2000, kun Nokia 3310 -puhelimelle julkaistiin Snake II. [2]

Snake-pelien ilmestymisen aikaan puhelimissa ei ollut kosketusnäyttöjä vaan fyysiset näppäimet, yleensä numeronäppäimet ja muutamia muita toimintonäppäimiä. Snake ja Snake II pelejä ohjattiin puhelimen numeronäppäimillä, muodostaen numeroista 2,4,6 ja 8 suuntanäppäimet ohjaukselle, tai puhelimen ylöspäin ja alaspäin suuntanäppäimillä, joilla liikkuaikseen oikealle tai vasemmalle suuntanäppäintä piti painaa kahdesti. [3] [4]

Vuonna 2003 julkaistiin maailman ensimmäinen pelaamiseen tarkoitettu puhelin, Nokia N-Gage, ja vuotta myöhemmin julkaistiin sen jatkaja Nokia N-Gage QD. Puhelinta käytettiin poikkeavasti vaakasuunnassa, vasemman käden puolella sijaitsivat nuolinäppäimet ja oikean käden puolella numeronäppäimet, joita käytettiin pelaamiseen. Tämä suunnittelutapa muistutti aikansa videopelikonsolien ohjainten mallia. N-Gage puhelimet eivät kuitenkaan

menestyneet markkinoilla, koska niissä oli lukuisia käyttäjiä häirinneitä suunnitteluvirheitä. Samanaikaisesti julkaistut kannettavat videopelikonsolit, kuten Nintendo Gameboy Advance ja Playstation Portable houkuttelivat ostajia enemmän matalamman hintansa takia. [5]

2.2 Mobiilipelien siirtyminen älypuheliin ja kosketusnäyttöohjaukseen

Vuonna 2007 Apple mullisti puhelinmarkkinat julkaisemalla ensimmäisen iPhone-puhelimen. iPhone toi mukanaan elementtejä, jotka toimivat tänäkin päivänä älypuhelimien perustana. Suurimpana muutoksena iPhone valtavirtaisti kapasitiivisten kosketusnäyttöjen käytön puhelimissa, ja toi ensimmäisenä kuluttajille mahdollisuuden ohjata kosketusnäyttöä useammalla kuin yhdellä kosketuspisteellä. Tämä mahdollisti esimerkiksi näytön sormilla nipistyksen ja vierityksen. Käyttöliittymä suunniteltiin sormiystävälliseksi ja mahdollisimman yksinkertaiseksi käyttää, joka oli suuri muutos aikaisempiin älypuheliin. Kosketusnäyttöjä oli puhelimissa jo ennen iPhone'n julkaisua, mutta monesti ne olivat resistiivisiä kosketusnäyttöjä, joita käytettiin erillisellä osoitinkynällä. iPhone'n menestystä kasvatti myös Applen vuonna 2008 iOS-käyttöjärjestelmilleen julkaisema sovelluskauppa App Store. App Store mahdollisti sovellusten ja pelien digitaalisen jakelun laitteilla nopeasti ja helposti. iPhone'n kosketusnäyttö ja App Storen suosio valtavirtaisti kosketusnäytöllä pelattavat mobiilipelit ja toi ne helposti kuluttajien saataville. [6] [7]

Nykyään mobiilipelien ohjaukseen kosketusnäytöllä on monta tapaa. Pelaaja voi sormillaan tehdä erilaisia eleitä näytön pintaan, esimerkiksi napauttaa yhden tai useamman kerran peräkkäin, pitää pohjassa, liu'uttaa, vierittää tai nipistää. Ohjaukseen voidaan myös käyttää puhelimesta olevaa kiihtyvyysanturia tai gyroskooppia, jotka havaitsevat puhelimen liikuttelun pelaajan toimesta.

2.3 Peliohjainten historia ja kehitysprosessi

Ensimmäinen kaupallinen pelikonsoli Magnavox Odyssey julkaistiin vuonna 1972. Konsolia ohjattiin hyvin yksinkertaisilla johdoilla liitetyillä erillisillä ohjaimilla, joissa oli yksi nappi, ja kolme pyöritettävää nuppia. Nupeilla pystyi ohjaamaan vertikaalista ja horisontaalista liikettä. Sauvaohjaimen suosituksi teki Atari vuonna 1977 julkaisemallaan Atari 2600 konsolilla. Konsolin ohjaimessa oli yksi sauvaohjain ja yksi nappi. Ohjain oli vuosia standardimalli muille konsoleille, kunnes 80-luvulla nappi-painotteiset ohjaimet yleistyivät. Vuonna 1983 Nintendo julkaisi Family Computer -konsolin, jonka suorakulmion mallinen ohjain sisälsi tänä päivänä standardina pidetyt nuolinäppäimet, pyöreät A- ja B-napit sekä Start- ja Select-napit. Samanlainen ohjain oli myös tunnetummassa vuonna 1985 julkaistussa Nintendo Entertainment System-konsolissa. Nämä ohjaimet toimivat pitkään malleina muiden yhtiöiden suunnitelmassa ohjaimia, ja niitä voidaan pitää nykyaikaisten ohjaimien lähtökohtina. Vuosien saatossa ohjaimet muuttuivat muodoiltaan pyöreämmiksi. Ohjaimien ulkomuodossa yleistyi molemmilla puolilla sijaitsevat ulokkeet, joista ohjainta pideltiin kämmenillä. Vuonna 1994 ilmestynyt Sony Playstation konsolin ohjain toi mukanaan standardin neljästä toimintonapista, syrjäyttäen vain pelkkien A- ja B-nappien käytön. Ohjaimessa oli myös uutuutena kaksi olkapäänappia ohjaimen yläosassa, joka muutti tapaa, jolla ohjaimia pidellään ja mitä sormia yleisesti mihinkin nappeihin käytetään. Vuonna 1996 markkinoille tulleen Nintendo 64 konsolin ohjain sisälsi ensimmäisenä peukalolla käytettävän ohjaussauvan. Kaikki nämä kehitysvaiheet kulminoituvat vuonna 1998 ilmestyneeseen Sony Playstation DualShock-ohjaimeen. Tämä ohjain sisälsi tullessaan kaikki ne toiminnot, joita tämän päivän peliohjaimista pidetään standardina löytyvän. Ohjaimesta löytyy nuolinäppäimet, molemmille puolille sauvaohjaimet, 4 toimintonäppäintä, Start- ja Select-napit ja molemmilla puolilla olkapäänappi sekä liipaisin. Edelleen 2020-luvun suositut konsolit kuten Microsoft Xbox, Sony Playstation ja Nintendo Switch käyttävät tätä pohjaa ohjaimissaan. [8]

3 Peliohjaimen käyttö mobiilipeleissä

3.1 Peliohjaimia käyttävät mobiilipelit ja peligenret

Mobiilipelit, joissa on ohjaintuki ovat useimmiten pelejä, joissa pelaaja ohjaa pelihahmoa tai ajoneuvoa. Tällaiset pelit ovat olleet olemassa jo pitkään pelattavina tietokoneilla, pelikonsoleilla ja muilla fyysisiä näppäimiä hyödyntävillä alustoilla, joten monelle pelaajalle niiden ohjaaminen peliohjaimella on paljon luontevampaa ja tutumpaa kuin kosketusnäytöllä ohjaaminen. Tällaisten pelien vakioituneet ohjausmekaniikat ovat kehittyneet aikana, kun kosketusnäytöllä pelien ohjaaminen ei ollut vielä mahdollista. Monesti mobiilipelit, joissa on ohjaintuki, ovatkin juuri tietokoneella ja pelikonsoleilla tunnetuiksi tulleet pelejä ja pelisarjoja, joista on tehty oma versionsa mobiilille. Ohjaintuellisten mobiilipelien genrejä yhdistää se, että niiden ohjaus vaatii pelaajalta tarkkuutta ja nopeutta, jotta pelaaja menestyy pelissä.

Suosituimpia 2020-luvun mobiilipelejä, joissa on ohjaintuki, ovat muun muassa Call of duty mobile, Fortnite, Roblox, Minecraft, Brawlhalla ja Dead cells. Näissä jokaisessa pelissä pelaaja ohjaa pelihahmoa, ja jokainen peli tai pelisarja on saatavilla myös tietokoneille ja pelikonsoleille. [9]

Ohjainta ja mobiililaitetta voi käyttää toisistaan erillään esimerkiksi niin että mobiililaitte, joka toimii näyttönä, on pöydällä tai muulla tasolla ja ohjain pelaajan käsissä. Suositumpaa on kuitenkin käyttää ohjainta, jossa on valmiiksi teline mobiililaitteelle, tai käyttää tavallista ohjainta, johon on asetettu erikseen ostettava teline mobiililaitteelle (Kuva 1).



Kuva 1. Peliohjaimen asennettava teline mobiililaitteelle. [10]

Call of duty mobile ja Fortnite ovat peligenreltään Shooter-pelejä, suomessa genren pelejä kutsutaan yleisesti räiskintäpeleiksi. Yleisesti näissä peleissä pelaaja ohjaa aseistettua pelihahmoa, ja objektiivina on päihittää vastustajat ampumalla (Kuva 2). Räiskintäpeleissä yleisiä ohjauskomentoja pelihahmolle ovat muun muassa liikkuminen, hyppääminen, kyykkääminen ja makaaminen, tähtääminen ja ampuminen, kranaattien heitto, lataaminen, aseiden vaihtaminen, lyöminen ja muut pelikohtaiset toiminnot. Näille toiminnoille on peliohjaimille vakiintuneet mallit siitä, mikä toiminto tapahtuu mistäkin napista.



Kuva 2. Call of duty mobile pelin pelaajanäkymä. [11]

Taistelupelit ovat myös yleisiä ohjaintuellisia mobiilipelejä. Taistelupeleissä, kuten edellä mainitussa Brawlhalla pelissä, ohjataan pelihahmoa taistelussa toista hahmoa vastaan (Kuva 3). Mekaniikat pohjautuvat usein hahmon liikuttamiseen, ja monimutkaisiin näppäinkombinaatioihin, joilla suoritetaan taisteluliikkeitä, joilla tehdään vastustajalle vahinkoa. Juuri harjoittelua vaativan ja monimutkaisen ohjauksen takia peliohjain on suosittu tapa pelata näitä pelejä mobiililla. Kosketusnäytöllä eri toimintoja tai suuntia vastaavien nappien sijoittelu vie näytöltä paljon tilaa, joka saattaa vaikuttaa myös negatiivisesti pelaajan sormien liikkeiden tarkkuuteen.



Kuva 3. Brawlhalla pelin pelaajanäkymä. [12]

Muita suosittuja peligenrejä, joiden mobiilipeleistä löytyy ohjaintuki, ovat esimerkiksi roolipelit ja ajopelit.

3.2 Peliohjaimen käytön hyödyt verrattuna kosketusnäytöllä ohjaukseen

Fyysisen ohjaimen käytöstä mobiilipeleissä saattaa olla hyötyä, kun tarvitaan ohjaukseen tarkkuutta ja varsinkin moninpeleissä etulyöntiasemaa niihin pelaajiin, jotka käyttävät kosketusnäyttöä ohjaamiseen. Jos pelaaja on valmiiksi tottunut käyttämään pelatessaan fyysistä ohjainta, voi sen käyttäminen mobiilipeleissä olla tehokkaampaa kuin opetella tuntematon ohjaustapa kosketusnäytöllä.

Kanadassa Yorkin yliopistossa vuosina 2010 ja 2013 tehdyissä tutkimuksissa vertailtiin kosketusnäytön eroja fyysisten näppäinten sisältävään ohjaimeen ja erilaisiin kosketusnäyttöön kiinnitettäviin fyysisiin näppäimiin. Vuoden 2010 tutkimuksessa 12 henkilöä pelasivat Assassin's Creed: Altair's Chronicles -peliä kahdella eri laitteella, iPhone 3G -puhelimella käyttäen sen kosketusnäyttöä, ja Nintendo DS Lite -käsikonsolilla käyttäen laitteen omia fyysisiä ohjainnäppäimiä. DS Lite -käsikonsolissa on 4 suuntanäppäintä vasemman käden puolella ja 4 toimintanäppäintä oikean käden puolella. Peli valikoitui

yliopiston tutkimukseen koska peli oli molemmilla alustoilla samanlainen pieniä graafisia eroja lukuun ottamatta. Tutkimuksessa pelaajien piti suorittaa pelistä tietty kenttä molemmilla laitteilla 4 kertaa. Pelaajien tulisi yrittää suorittaa kenttä mahdollisimman nopeasti ja mahdollisimman pienellä määrällä pelihahmon kuolemia. Pelaajat saivat aluksi harjoitella suorittamalla kentän yhden kerran läpi. Suoritusten aikana kirjattiin ylös määrä, kuinka monta kertaa pelaajan hahmo kuoli, ja kuinka monta sekuntia pelaajalla kesti päästä kenttä loppuun asti. Tutkimuksessa osoitettiin pelaajien hahmon kuolemien määrän olevan 150 % korkeampi kosketusnäytöllä pelatessa, ja että aika sekunneissa, joka meni kentän suorittamiseen loppuun asti, oli 42 % pienempi, kun käytettiin Nintendoa ja sen fyysisiä näppäimiä. [13] Tutkimusta haluttiin täydentää, ja vuonna 2013 tehtiin uusi tutkimus, jossa vertailtiin neljää eri ohjausmenetelmää. Tutkimuksessa käytettiin Google Nexus One -puhelinta, ja verrattavat ohjausmenetelmät olivat pelkkä kosketusnäyttö, vasemman käden puolella olevien suuntanuolien päälle kosketusnäyttöön kiinnittyvä sauvaohjain, kosketusnäyttöön liimautuvat vaahtomuovinapit ja kokonaan erillinen fyysinen Wiimote-ohjain. 12 testaajaa suoritti Metal Slug-Super Vehicle-001 -pelistä ensimmäisen kentän kolme kertaa jokaisella testattavalla ohjaustavalla. Tämäkin tutkimus osoitti selkeästi erillisen fyysisen Wiimote-ohjaimen olevan paras menetelmä pelin ohjaukseen. Ohjaimella pelatessaan pelaajat kuolivat pelissä huomattavasti harvemmin, kentän suoritus aika oli lyhyempi ja pelaajat keräsivät enemmän pisteitä. Suurin suoritus aika ja eniten kuolemia tuli pelkkää kosketusnäyttöä käyttäessä. [14]

4 Työn kuvaus ja käytetyt teknologiat

4.1 Suunniteltu työprosessi ja sen tavoitteet

Opinnäytetyön tarkoituksena on kehittää Unity-pelimootorilla testisovellus Android-puhelimelle käyttäen Unityn Input System -pakettia ja sen sisältämää Gamepad-luokkaa, ja testata, ottaako sovellus vastaan signaaleja eri Bluetooth-ohjaimilta. Testillä halutaan tutkia kehittäjän näkökulmasta, onko Unityllä ja sen Input System -pakettia ja Gamepad-luokkaa hyödyntäen mahdollista kehittää Android-sovellus, jonka ohjaaminen onnistuisi monella eri valmistajan Bluetooth-ohjaimella. Opinnäytetyö voi toimia ohjeena kehittäjille ohjaintuen luomiseen Unityllä. Eri valmistajien Bluetooth-ohjaimia on paljon, ja eikä kehittäjän kannalta ole ongelmattonta kehittää sovellukselle ohjaintukea, joka toimisi vain yhden tietyn ohjaimen kanssa. Kehittämällä sovelluksiin ohjaintuki, joka mahdollistaa mahdollisimman monen eri ohjaimen käytön, voi sovelluskehittäjä saada sovellukselleen enemmän käyttäjiä. Isommat videopelejä kehittävät yritykset ovat pystyneet kehittämään peleilleen mahdollisimman kattavan ohjaintuen, mutta opinnäytetyön tarkoituksena on selvittää, voivatko Indie-kehittäjät, joiden päätyökaluna toimii Unity, tehdä vastaavaa laajaa ohjaintukea nopeasti ja kustannustehokkaasti Unityn Input Systemia ja sen Gamepad-luokkaa hyödyntäen. Opinnäytetyössä myös kevyesti vertaillaan eri ohjaimia, jotta selviää, onko ohjaimissa eroja, jotka saisivat kehittäjän suosimaan vain tietylle ohjaimelle tuen tai ominaisuuksien kehittämistä sovellukseen.

4.2 Käytetyt laitteet ja teknologiat

4.2.1 Unity-pelimoottori

Unity on Unity Technologiesin vuonna 2005 julkaisema C++-pohjainen pelimoottori ja kehitysympäristö 2D ja 3D pelien luomiseen. Unityn ohjelmointikielenä toimii C#. Unity-pelimoottorilla pystyy luomaan pelejä monille

eri alustoille, muun muassa tietokoneille, mobiililaitteille ja videopelikonsoleille. Sitä käytetään pelien kehityksen lisäksi esimerkiksi elokuva- ja autoteollisuudessa. [15] Tämän työn testisovelluksen tekoon käytetään Unityn versiota 2021.3.8f1.

4.2.2 Unity Input System -paketti

Unity Input System on Unity-pelimoottoriin ladattava paketti, joka mahdollistaa erilaisista ohjauslaitteista, kuten peliohjaimesta tai näppäimistöstä, tulevien syötteiden käyttämisen ohjaukseen kehitettävässä sovelluksessa. [16] Työssä päätettiin hyödyntää Unity Input Systemiä, sillä se on Unityllä kehittäessä hyvin laajasti käytetty, ja se sisältää Gamepad-luokan, jonka on tarkoitus yhtenäistää erilaiset klassisen tyyppiset ohjaimet, jotka sisältävät nuolinäppäimet, sauvaohjaimet, 4 toimintonäppäintä, olkapäänäpit sekä liipaisimet. Työn testisovelluksessa on tarkoitus saada eri ohjaimet toimimaan samalla tavalla hyödyntäen vain Gamepad-luokkaa sen sijaan että jokaiselle erimerkkiselle ohjaimelle luotaisiin oma käyttösysteeminsä. Tämä nopeuttaisi kehittäjän työprosessia huomattavasti, jos kehitetään sovellusta, jossa on ohjaintuki mahdollisimman monelle ohjaimelle. Työssä käytetään Input Systemin versiota 1.4.2.

4.2.3 Android-käyttöjärjestelmä

Android on vuonna 2003 alkunsa saanut käyttöjärjestelmä puhelimille ja tableteille. Ensimmäinen kehittäjille tarkoitettu versio julkaistiin vuonna 2007, ja ensimmäinen Android-järjestelmällä toimiva puhelin, T-Mobile G1, tuli markkinoille vuonna 2008. Android ja Applen iOS ovat kilpailleet samanaikaisesta julkaisustaan saakka käytetyimmän puhelinkäyttöjärjestelmän tittelistä, mutta vuonna 2012 Android ohitti iOS järjestelmän maailman käytetyimpänä, ja vuonna 2020 jopa 75 prosenttia maailman puhelimista ja tableteista sisältää Android-käyttöjärjestelmän. [17] Työssä sovelluksen testaukseen käytettävän puhelimen Android-versio on 12.

4.2.4 Bluetooth-tiedonsiirtotekniikka

Bluetooth on vuonna 1994 alkunsa saanut tiedonsiirtotekniikka, joka perustuu lyhyen kantaman radiotekniikkaan. Bluetoothia käytetään vaihtoehtona langallisille yhteyksille. Bluetoothin standardien kehittämisestä on vastuussa Bluetooth Special Interest Group (SIG). [18] Tänä päivänä Bluetoothia käytetään esimerkiksi puhelimissa, kaiuttimissa, kuulokkeissa, näppäimistöissä ja ohjaimissa. Bluetoothin uusin versio on vuonna 2017 julkaistu versio 5, jonka maksimi lähetysopeus on 2 megabittiä sekunnissa (2 Mbps), ja maksimi kantama 240 metriä. [19] Työssä käytettävän puhelimen Bluetooth versio on 5.0.

4.2.5 Käytetyt peliohjaimet

Työssä käytettävät peliohjaimet ovat Sony DualShock 4 v2 -ohjain, Sony DualSense -ohjain ja Microsoft Xbox One Black (2016) -ohjain (Kuva 4).



Kuva 4. Työssä käytettävät ohjaimet.

5 Testisovelluksen luominen ja eri ohjainten toimivuuden testaus sovelluksessa

5.1 Ohjainten tekniset erot ja yhdistäminen Android-laitteeseen

Kaikissa työssä käytetyissä ohjaimissa on nuolinäppäimet, sauvaohjaimet molemmilla puolilla, 4 toimintonäppäintä, Start- ja Select-napit (napit ovat nimitykseltään DualShock 4 -ohjaimessa Share ja Options, DualSense-ohjaimessa Create ja Options ja Xbox One -ohjaimessa View ja Menu) ja molemmilla puolilla olkapäänappi sekä liipaisin. Xbox One -ohjaimessa on Xbox-nappi, ja vastaavasti Sonyn molemmissa ohjaimissa on Playstation-nappi. Erikoisuutena molemmista Sonyn ohjaimista löytyy kosketuslevy. Kaikista ohjaimista löytyy 3,5 mm kuulokeliitäntä. Tiedonsiirtoon ja mahdolliseen ohjaimen akun lataamiseen Xbox One -ohjaimessa ja DualShock 4 -ohjaimessa on micro-USB portti, kun taas DualSense ohjaimessa käytetään uudempaa ja nopeampaa USB Type-C liitäntää. Xboxin ohjain saa virtansa kahdesta AA-kokoisesta alkaliparistosta, mutta myös uudelleen ladattavia ohjaimeen tarkoitettuja akkuja on markkinoilla. Molemmat Sonyn ohjaimet sisältävät ladattavat akut. DualShock 4 -ohjaimessa on 3,7 V:n litiumpolymeeriakku, jonka kapasiteetti on 1000 mAh. DualSense-ohjaimen akku on muuten vastaava, mutta sen kapasiteetti on 1560 mAh.

Ohjaimet yhdistetään Android-laitteeseen Bluetoothin avulla. Käytettyjen ohjainten Bluetooth-versiot eivät ole samat. DualShock 4-ohjaimessa on Bluetooth-versio 2.1, Xbox One -ohjaimessa 4.0 ja DualSense-ohjaimessa 5.1. [20] [21] [22]

Ohjainten yhdistys työssä käytettyyn Android-laitteeseen tapahtui jokaisella ohjaimella samanlaisella kaavalla. Ensin Android-laitteen asetuksista kytketään Bluetooth päälle. Seuraavaksi ohjain käynnistetään, ja asetetaan pariliitostilaan. Xbox One -ohjaimella pariliitostila kytketään päälle ohjaimen yläosasta löytyvällä pariliitosnapilla. Sonyn molemmilla ohjaimilla pariliitostila kytketään päälle painamalla samanaikaisesti Playstation-nappia ja Share- tai Create-

nappeja. Kaikki ohjaimet indikoivat olevansa pariliitostilassa valon vilkkumisella. Pariliitostilassa ollessaan ohjaimet näkyvät Android-laitteen Bluetooth-valikon pariliitoshaussa olevassa listassa. Xbox One -ohjain näkyy listassa nimellä Xbox Wireless Controller, kun taas Sonyn ohjaimet näkyvät nimellä Wireless Controller. Haluttu ohjain valitaan listasta, jonka jälkeen Bluetooth-yhteys Android-laitteen ja ohjaimen välille on muodostettu.

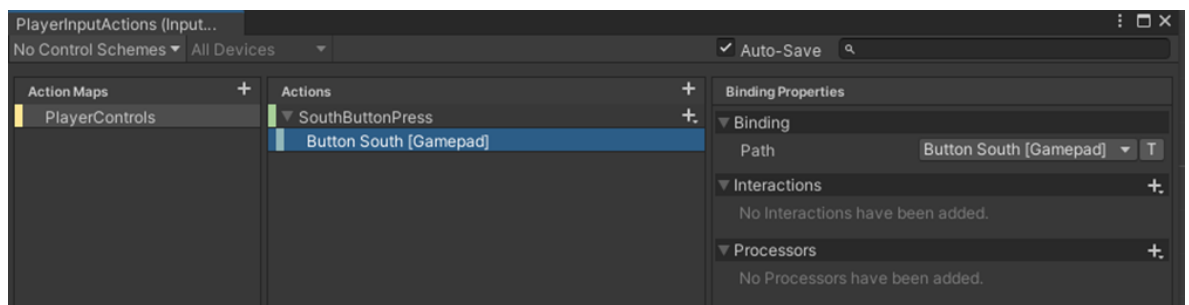
5.2 Ohjainten toimivuus testisovelluksessa

5.2.1 Testisovelluksen luominen

Testisovellus tehtiin Windows-pohjaisella tietokoneella käyttäen Unity-pelimoottorin versiota 2021.3.8f1. Projekti luotiin käyttäen Unityn valmista 2D-projektipohjaa, sillä 2D riittää sovelluksen käyttötarkoitukseen. Ensimmäiseksi projektiin lisättiin kuvatiedostot. Pelinäkökuvassa on ohjaimen kuva, jossa näkyvät kaikki käytettävät ohjausnapit. Ohjaimen napeista tulevaa signaalia indikoivat kuvat punaisista rasteista ja ympyröistä. Pelinäkökuvan kuvasuhde muutettiin vastaamaan testaukseen käytettävän Android-puhelimen ruudun kuvasuhdetta (20:9). Projektiin asennettiin Unityn package manager -ikkunan kautta Input System 1.4.1 -paketti. Alkuvaiheessa projekti pidettiin Windows-pohjaisena, jotta ohjainten toimivuutta pystyi pikaisesti testaamaan, ennen kuin projekti rakennettaisiin Androidille asennettavaksi APK-tiedostoksi.

Käyttäen ladattua Input System -pakettia, projektin Assets-kansioon luotiin PlayerInputActions-niminen Input actions -objekti. Tällä objektilla hallinnoidaan sovellukseen tulevia syötteitä. Objektin sisään luotiin PlayerControls-niminen Action map -objekti. Luomalla Action map -objekteja pystytään hallinnoimaan sovellukseen tulevia syötteitä sovelluksen eri vaiheissa. Esimerkiksi jos kyseessä olisi videopeli, voitaisiin kulloinkin aktiivisena olevan Action map -objektin kautta määrittää, että tietyistä syötteistä tapahtuu jokin toiminto, esimerkiksi painamalla nappia A pelaajan hahmo hyppäisi. Jos kuitenkin jossain pelin vaiheessa haluttaisiinkin, että painamalla nappia A pelaajan hahmo tekisikin jotain muuta, onnistuisi tämä vaihtamalla toiseen Action map -objektiin,

jossa napille A olisi määritetty eri toiminto. Tehdyssä testisovelluksessa ei ollut tarvetta useammalle kuin yhdelle Action map -objektille. Seuraavaksi PlayerControls-objektin sisään täytyi luoda Actions-objekteja. Nämä kuvastavat eri toimintoja, joita sovelluksessa halutaan tehdä. Esimerkiksi videopelissä nämä toiminnot voisivat olla hahmon liikkuminen, hyppääminen tai tähtääminen. Luotiin SouthButtonPress-niminen toiminto vastaamaan peliohjaimen eteläisimmän toimintonäppäimen painamista. Toiminnot täytyi määrittää sidos (Binding) johonkin syötteeseen. Sidoksen pystyi valitsemaan manuaalisesti alas vedettävästä listasta, joka näyttää eri luokkien lähdetyyppejä, tai valitsemalla kuuntelutilan, jossa Unity panee merkillä tulevan syötteen lähteen ja ehdottaa sitä sidokseksi. Valittiin listasta Button South (Gamepad) -sidos (Kuva 5). Koska kyseessä on painettava nappi, valitaan toimintotyyppi nappi (Button).



Kuva 5. PlayerInputActions-objekti

Testisovelluksessa päätettiin käyttää vain Gamepad-lähdeluokkaa, sillä Unity on luonut tämän luokan käytettäväksi tilanteissa, joissa syötteen lähteenä toimii klassisen mallinen peliohjain, joka sisältää nuolinäppäimet, sauvaohjaimet, 4 toimintonäppäintä, olkapäänapit sekä liipaisimet. [23] Testisovelluksen avulla haluttiin testata, onnistuuko kaikkien eri ohjaimien käyttö testisovelluksessa käyttämällä vain Gamepad-luokkaa, jolloin kehittäjän ei tarvitsisi tehdä jokaiselle ohjaimelle erillisiä tapoja syötteen vastaanottoon, joka säästäisi huomattavasti aikaa kehitystyössä.

Työtä varten luotiin kaksi metodia, FlashImage (Kuva 6) ja TriggerEvent (Kuva 7). Näistä ensimmäistä sovellus kutsuu silloin kun se saa peliohjaimelta signaalin napin painalluksesta, ja toista silloin kun signaali tulee liipaisimen

painalluksesta tai sauvaohjaimen liikutuksesta. Nämä metodit ovat siis vastuussa visuaalisesta signaalista käyttäjälle, että ohjaimesta on vastaanotettu signaalia. FlashImage-metodia kutsuttaessa se laittaa peliobjektin, johon koodikomponentti on liitetty, joko aktiiviseksi tai epäaktiiviseksi, riippuen kumpaa peliobjekti kutsumishetkellä on. Koodikomponentti liitettäisiin punaista X-kirjainta vastaaviin peliobjekteihin.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  @ Unity Script | 0 references
6  public class FlashThisImage : MonoBehaviour
7  {
8      0 references
9      public void FlashImage()
10     {
11         if (gameObject.activeInHierarchy == false)
12         {
13             gameObject.SetActive(true);
14         }
15         else
16         {
17             gameObject.SetActive(false);
18         }
19     }
20 }
```

Kuva 6. FlashImage-metodi.

TriggerEvent-metodia kutsuttaessa metodi ottaa vastaan InputAction-luokalta tiedon siitä, onko liipaisimen tai sauvaohjaimen liikutus aloitettu vai lopetettu. Jos liikutus aloitetaan, Punaista X-kirjainta tai punaista ympyrää vastaava peliobjekti, joka sisältää tämän koodikomponentin tulee aktiiviseksi. Jos liike lopetetaan eli liipaisin päästetään ylös tai sauvaohjain palautumaan takaisin keskipisteeseensä, peliobjekti kytketään pois päältä.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.InputSystem; // Lisätään tämä jotta saadaan pääsy InputAction-luokkaan

[UnityScript | 0 references]
public class TriggerFlashImage : MonoBehaviour
{
    0 references
    public void TriggerEvent(InputAction.CallbackContext value)
    {
        if (value.started)
        {
            gameObject.SetActive(true);
        }
        if (value.canceled)
        {
            gameObject.SetActive(false);
        }
    }
}

```

Kuva 7. TriggerEvent-metodi.

Pelinäkymään lisättiin tyhjä objekti, jolle asetettiin nimi ScriptHolder. Tähän objektiin lisättiin Player Input -komponentti. Komponentin Actions-kenttään linkattiin aikaisemmin luotu PlayerInputActions-niminen Input actions -objekti. Komponentin toimintatavaksi asetetaan Unity-tapahtumien (Unity Events) kutsuminen.

Aloitettiin testaamalla ensimmäiseksi pelkkien toimintonäppäimien toimivuus. Pelinäkymään lisättiin punaisia X-kirjaimia esittävät peliobjektit, jotka näkyvät pelinäkymässä olevan ohjaimen kuvan vastaavien nappien päällä. Näihin jokaiseen objekteihin lisättiin koodikomponentti, joka sisältää FlashImage-metodin. ScriptHolder-objektiin aiemmin lisätyn Player Input -komponentin kautta näiden peliobjektien sisältäviä metodeja voidaan kutsua käyttämällä Unity-tapahtumia. Player Input -komponentin Events-kohdan alla näkyvät kaikki toiminnot, jotka on luotu PlayerControls-nimisen Action map -objektin alle. Toimintonäppäinten painalluksia vastaaviin toimintoihin raahattiin toimintonäppäinten X-kuvia vastaavat peliobjektit. Tämän jälkeen toimintonäppäimen painamisen pitäisi kutsua Unity-tapahtumien kautta näppäintä vastaavan kuvan metodia, joka laittaa kuvan päälle ja pois vastaten näppäinten painalluksia. Toimivuus testattiin ensin Windows-ympäristössä liittämällä tietokoneeseen Xbox One -ohjain. Toimintonäppäinten painaminen

sai toivotusti aikaan X-kirjainten vilkkumisen päälle ja pois vastaavien nappien kohdalla. Tämän jälkeen sovelluksen alusta vaihdettiin asetuksista Androidiksi. Sovelluksesta rakennettiin apk-tiedosto, joka asennettiin Android-puhelimeen. Puhelimeen yhdistettiin testausta vasten Xbox One -ohjain. Sovellus otti ohjaimen signaalit toivotusti vastaan, ja näppäinten painallus sai aikaan X-kirjainten vilkkumisen päälle ja pois sovelluksessa.

Seuraavaksi lisättiin luotuun PlayerControls-objektiin toiminnot, jotka vastaisivat kaikki muita ohjaimen napin tavalla painettavia ohjaustapoja (Kuva 8). Nämä ovat nuolinäppäimet, sauvaohjainten painaminen, Start- ja Select-napit sekä olkapäänapit. Toimintoihin lisättiin Gamepad-luokasta niitä vastaavat sidokset. Toiminnot asetettiin kutsumaan oikeita metodeja ScriptHolder-objektin Player Input -komponentista Unity-tapahtumien avulla. Sovelluksesta rakennettiin uusi apk-tiedosto ja se asennettiin puhelimelle. Sovellusversio testattiin, ja kaikki napin painallusta vastaavat ohjaustavat antoivat oikean signaalin sovellukselle ja saivat oikeat X-merkit näkymään sovelluksessa.



Kuva 8. Sovelluksessa napin painallusta vastaavat ohjausnäppäimet.

Seuraavaksi implementoitiin toiminnot liipaisimille ja sauvaohjaimien liikutukselle. Nämä toiminnot eivät ole suoranaisesti napin painalluksia, vaan niissä täytyy pystyä rekisteröimään painalluksen aloitus ja lopetus. Näihin käytettiin luotua TriggerEvent-metodia napeissa käytetyn FlashImage-metodin sijasta. Toiminnot luotiin PlayerControls-objektiin ja niihin lisättiin Gamepad-luokasta niitä vastaavat sidokset. Sauvaohjaimien liikutukseen liittyviin toimintoihin sidoksia piti lisätä neljä, jokaiselle suunnalle omansa. Pelinäköymässä sauvaohjaiminen liikutuksen signaalia kuvastaa punainen ympyrä, ja liipaisimissa punainen X-kirjain. Tämän jälkeen rakennettiin viimeinen versio sovelluksesta apk-tiedostoksi, jota testattiin puhelimella. Sovellus luki oikein Xbox One -ohjaimesta tulevat signaalit myös näiden ohjauskomentojen osalta.

5.2.2 Ohjainten perusnäppäinten testaus testisovelluksessa

Jo sovelluksen kehitysvaiheessa testaamiseen käytettiin Xbox One -ohjainta. Sillä kaikki luodut toiminnot toimivat oikein, eli käytetty Gamepad-luokka osasi vastaanottaa ohjaimelta oikeita näppäimiä vastaavat toiminnot.

Sony DualShock 4 -ohjaimen toiminta testattiin sovelluksessa seuraavaksi. Sovellus rekisteröi ohjaimen signaalit oikein, ja vastasi testattua Xbox One -ohjainta toimivuudellaan.

Odotetusti Sony DualSense -ohjain ei menestynyt testauksessa. Unityn Input System on tämän ohjaimen osalta vielä kehitysvaiheessa kirjoitushetkellä syksyllä 2022, ja virallisesti Unity on luvannut täydellisen toimivuuden vain Windows- ja macOS-käyttöjärjestelmille. [24] Jos kehittäjä haluaisi siis sovelluksensa olevan käytettävä myös DualSense-ohjaimella, täytyisi sille kehittää erillinen tukensa jollain muulla keinolla kuin Input System -paketilla. Koska kuitenkin Sonyn DualShock 4 -ohjain toimi testauksessa, päätettiin saatavilla oleva DualSense-ohjain testata varmuuden vuoksi. Toimintonäppäimistä yksi ei antanut sovelluksessa näkyvää signaalia ollenkaan, kasi näppäimistä antoivat väärän signaalin, ja vain yksi toimi

toivotusti. Start- ja Select-napit eivät antaneet sovelluksessa näkyvää signaalia, mutta sauvaohjaimien painaminen rekisteröityi edellä mainittujen nappien painalluksena. Vasemman sauvaohjaimen liikuttaminen näkyi oikein, oikeanpuoleisen liikuttelusta ei tullut näkyvää signaalia. Liipaisimet näkyivät olkapäänappien painalluksena, kun taas olkapäänappien painallus ei minään. Hieman yllättäen kuitenkin nuolinäppäimet antoivat kaikki oikean signaalin sovellukselle.

5.2.3 Ohjaimien erikoisnäppäimet

Unity ei tällä hetkellä tue ohjaimista löytyvien tärinämoottoreiden käyttöä Android-alustalla. Sony DualShock 4 -ohjaimessa olevaa Playstation-nappia ja Xbox One -ohjaimesta löytyvää Xbox-nappia yritettiin saada sovelluksessa toimimaan tuloksetta. Gamepad-luokka ei sisällä niihin sopivaa sidosta, eikä sellaisia löytynyt myöskään ohjainten spesifimmistä sidoslukista. Testauksen yhteydessä havaittiin sattumalta, että Sony DualShock 4 -ohjaimessa oleva kosketuslevy toimii sovelluksessa osoitinlaitteen tavoin. Tätä toimintoa ei ollut millään tavalla suunniteltu eikä implementoitu sovellukseen erikseen, mutta sen löytyessä se päätettiin myös testata Android-alustalla. Sovellukseen lisättiin Unityn käyttöliittymävalikosta pelinäkymään painettava nappi ja liukusäädin. Android-alustalla testatessa sovelluksessa olevaa nappia pystyi painamaan ja liukusäädintä pystyi ohjaamaan kosketuslevyn avulla. Tämän toiminnon käyttö voisi soveltua peleihin, joissa tietokonehiiren kaltaisesta ohjausmekanismista voisi olla hyötyä.

6 Tulokset ja pohdinta

Opinnäytetyön tavoitteena oli tutkia, pystyykö Unity-pelimootoria ja sen sisältämää Input System -pakettia käyttäen luomaan mahdollisimman laajan ohjaintuen Android-alustalla toimivalle mobiilisovellukselle, jos käytetään vain Input System -paketin Gamepad-luokkaa. Opinnäytetyössä haluttiin selvittää, onko Indie-kehittäjillä mahdollisuutta luoda mobiilisovellukselleen kustannustehokkaasti laajaa ohjaintukea Unity-pelimootoria käyttäen. Opinnäytetyössä haluttiin vertailla kevyesti työssä käytettyjen kolmen eri Bluetooth-ohjaimen eroja ja tutkia onko ohjaimissa sellaisia ominaisuuksia, jotka saisivat kehittäjän suosimaan tekemään ohjaintuen sovellukseensa jollekin tietylle ohjaimelle.

Luvuissa 2 ja 3 tutustuttiin mobiilipelien ja ohjainten historiaan, jotta käsitetään, millaisia menetelmiä mobiilipelien ohjaukseen käytetään, millaisissa peleissä ja miten ohjaimia käytetään ohjaukseen sekä onko mobiilipelien ohjaukseen hyödyllistä käyttää erillistä fyysistä ohjainta kosketusnäytön sijaan. Luvussa 4 käytiin läpi suunniteltu työprosessi ja työssä käytetyt teknologiat. Luvussa 5 vertailtiin testissä käytettyjen ohjainten ominaisuuksia, luotiin testisovellus Unity-pelimootorilla ja ohjaintuki Input System -paketilla, ja testattiin eri ohjainten toimivuus luodussa sovelluksessa Android-alustalla.

Ohjainten ja mobiilipelien historian ja nykytilanteen tutkiminen loi tietopohjan sille, miksi mobiilipeleille on kannattavaa luoda ohjaintuki. Fyysisellä ohjaimella pelaaminen kosketusnäytön sijaan saattaa parantaa pelaajan suorituskyyä ja esimerkiksi antaa pelaajalle etulyöntiaseman moninpeleissä. Ohjaintuellisia mobiilipelejä on markkinoilla paljon, ja ohjaimen käyttö niissä on suosittua.

Opinnäytetyössä käytetyt ohjaimet eivät pääpiirteiltään eronneet toisistaan. Ohjainten huomattavimmat erot olivat ohjainten eri Bluetooth-versiot sekä Sonyn ohjaimissa erikoisuutena olevat kosketuslevyt. Opinnäytetyöhön ei sisältynyt tutkimusta siitä, onko eri Bluetooth-versioilla merkittävää eroa pelaajan suorituskyyyn esimerkiksi latenssin muodossa. Sonyn ohjaimista löytyvät kosketuslevyt saattavat olla tekijä, joka saisi kehittäjän suosimaan

ohjaintuen luomista näille ohjaimille. Testisovelluksessa DualShock 4 -ohjaimen kosketuslevyä onnistuttiin käyttämään tietokonehiiren tavoin napin painamiseen ja liukusäätimen ohjaamiseen. Tätä toimintoa voisi mahdollisesti hyödyntää mobiilipeleissä. Muita tekijöitä, jotka saisivat kehittäjän suosimaan vain tiettyä ohjainta, ei löytynyt.

Luodussa testisovelluksessa 2 ohjainta käytetystä 3:sta toimivat sovelluksessa moitteetta. Sonyn DualSense-ohjain ei toiminut toivotusti sovelluksessa, sillä Unity ei ole kehittänyt sille vielä kunnollista tukea Input System -pakettiin. Sonyn DualShock 4 -ohjain ja Microsoftin Xbox One -ohjain toimivat sovelluksessa, joka onkin tärkeämpää, sillä nämä ohjaimet ovat kirjoitushetkellä syksyllä 2022 paljon yleisemmässä käytössä kuin uudempi vuonna 2020 ilmestynyt DualSense-ohjain. DualSense-ohjainten yleistyessä olisi tärkeää, että kehittäjät huomioisivat myös tälle ohjaimelle tuen kehittämisen. Seuraamisen arvoista on, aikooko Unity kehittää tuen DualSense-ohjaimille mobiililaitteille tulevaisuudessa, jotta Unity-pelimootoria työkalunaan käyttävät kehittäjät voisivat luoda ohjaintuen sovelluksilleen myös DualSense-ohjaimille.

Unity-pelimoottorin Input System -paketti oli opinnäytetyössä ohjaintuen tekemiseen toimiva työkalu, mutta opinnäytetyöllä ei pystytä täysin vastaamaan kysymykseen, onko Gamepad-luokan käyttö hyvä tapa luoda kattava ohjaintuki. Gamepad-luokan käyttö mahdollisti opinnäytetyössä Xbox One -ohjaimen ja Sonyn DualShock-ohjaimen toimivuuden tehdyssä sovelluksessa, ja Gamepad-luokan avulla ohjaintuen luominen oli nopeaa ja kustannustehokasta. Opinnäytetyössä kuitenkin testattiin vain kolmea eri ohjainta, joten Gamepad-luokalla saavutettavan ohjaintuen laajuus ei opinnäytetyössä täysin selvinnyt, ja se olisi hyvä selvittää jatkotutkimuksella, jossa testattavia ohjaimia olisi enemmän.

Lähteet

- [1] Microsoft Devices Blog, 10 things you didn't know about mobile gaming. Saatavilla: <https://blogs.windows.com/devices/2013/01/16/10-things-you-didnt-know-about-mobile-gaming-2/> (Viitattu: 16.7.2022)
- [2] Krishnankutty, P., Nokia's Snake, the mobile game that became an entire generation's obsession. Saatavilla: <https://theprint.in/features/nokias-snake-the-mobile-game-that-became-an-entire-generations-obsession/462873/> (Viitattu: 16.7.2022)
- [3] Videonikkari, Nokia 3310 snake II game [Video]. Saatavilla: <https://www.youtube.com/watch?v=9Votks5-GhQ> (Viitattu: 17.7.2022)
- [4] Żurawka, A., Nokia 6110 Retro review. Old ringtones & snake game [Video]. Saatavilla: <https://www.youtube.com/watch?v=tMvls9RhiGk> (Viitattu: 17.7.2022)
- [5] Martini, R. 2007, Nokia and mobile gaming, sivut 2, 13–14. Saatavilla: <https://summit.sfu.ca/item/8186> (Viitattu 18.7.2022)
- [6] Whitwam, R., How Steve Jobs killed the stylus and made smartphones usable. Saatavilla: <https://www.extremetech.com/computing/98923-how-steve-jobs-killed-the-stylus-and-made-smartphones-usable> (Viitattu: 20.7.2022)
- [7] Alha, K., Mäyrä, F. 2020, Mobile Gaming, sivut 2–4. Saatavilla: https://trepo.tuni.fi/bitstream/handle/10024/140558/Mobile_Gaming_2020.pdf;jsessionid=1BEC190C10B5B6DF8B9E2055B691E3A1?sequence=1 (Viitattu: 23.7.2022)
- [8] Vincent, B., The Evolution of the Video Game Controller. Saatavilla: <https://www.popularmechanics.com/technology/gadgets/g34288261/evolution-of-the-video-game-controller/> (Viitattu 8.8.2022)
- [9] Petrovich, E., 13 Best Mobile Games With Controller Support. Saatavilla: <https://gamerant.com/best-mobile-games-controller-support/> (Viitattu: 17.8.2022)
- [10] 8BitDo, Mobile Gaming Clip for Xbox controllers. Saatavilla: <https://www.8bitdo.com/mobile-gaming-clip-for-xbox-controllers/> (Viitattu: 10.11.2022)

- [11] Activision, Call of duty mobile [Videopeli]. Saatavilla: <https://www.egames.news/noticias/Que-dispositivos-son-compatibles-con-Call-of-Duty-Mobile-20210119-0016.html> (Viitattu 20.8.2022)
- [12] Ubisoft, Brawlhalla [Videopeli]. Saatavilla: <https://mobilemodegaming.com/2020/08/10/brawlhalla-mobile-some-useful-tips-and-tricks-to-get-you-started/> (Viitattu 21.8.2022)
- [13] Zaman, L., Natapov, D., Teather, R., Touchscreens vs. traditional controllers in handheld gaming. Saatavilla: https://www.researchgate.net/publication/228744567_Touchscreens_vs_traditional_controllers_in_handheld_gaming (Viitattu: 13.7.2022)
- [14] Zaman, L., MacKenzie, S., Evaluation of nano-stick, foam buttons, and other input methods for gameplay on touchscreen phones. Saatavilla: <https://www.yorku.ca/mack/mhci2013-69.pdf> (Viitattu: 15.7.2022)
- [15] Wikipedia, Unity (game engine). Saatavilla: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) (Viitattu: 25.8.2022)
- [16] Unity technologies, Input System. Saatavilla: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.4/manual/index.html> (Viitattu: 25.8.2022)
- [17] The Editors of Encyclopedia Britannica, Android operating system. Saatavilla: <https://www.britannica.com/technology/Android-operating-system> (Viitattu: 26.8.2022)
- [18] Wikipedia, Bluetooth. Saatavilla: <https://en.wikipedia.org/wiki/Bluetooth> (Viitattu: 26.8.2022)
- [19] Triggs, R., Wankhede, C., A little history of Bluetooth. Saatavilla: <https://www.androidauthority.com/history-bluetooth-explained-846345/> (Viitattu: 26.8.2022)
- [20] Wikipedia, Xbox Wireless Controller. Saatavilla: https://en.wikipedia.org/wiki/Xbox_Wireless_Controller (Viitattu 24.9.2022)
- [21] Wikipedia, DualShock. Saatavilla: https://en.wikipedia.org/wiki/DualShock#DualShock_4 (Viitattu: 24.9.2022)

[22] Jones, R., DualSense: Everything you need to know about the PS5 controller. Saatavilla: <https://www.trustedreviews.com/news/dualsense-release-date-specs-preorder-news-3945005> (Viitattu 24.9.2022)

[23] Unity documents, Class Gamepad. Saatavilla: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.4/api/UnityEngine.InputSystem.Gamepad.html> (Viitattu 2.9.2022)

[24] Unity forum, Input System 1.2 released: More fixes and PS5 controller support for Desktop. Saatavilla: <https://forum.unity.com/threads/input-system-1-2-released-more-fixes-and-ps5-controller-support-for-desktop.1192894/> (Viitattu: 7.9.2022)