

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutus

Miika Piitulainen

Asymmetrisen VR-pelin tutkimus ja kehitys Unreal Engine 4-
Pelimoottorissa

Opinnäytetyö
Marraskuu 2019



Opinnäytetyö
Syyskuu 2019
Tietojenkäsittelyn koulutus
Alempi ammattikorkeakoulututkinto
Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä
Miika Piitulainen

Asymmetrisen VR-pelin tutkimus ja kehitys Unreal Engine 4-Pelimoottorissa

Toimeksiantaja
Vaki Oy

Tiivistelmä

Tässä opinnäytetyössä kehitän Unreal Engine 4 – pelimoottoria käyttäen asymmetrisen moninpelin. Moninpeli toimii kahden pelaajan välillä, missä toinen pelaaja pelaa näppäimistöllä ja hiirellä ja toinen pelaa virtuaalitodellisuuslaseilla. Moninpelistä tehdään kaksi versiota, missä moninpelin toteutustapa on eri. Opinnäytetyössäni vertaan versioita toisiinsa ja pyrin vastaamaan kahteen tutkimuskysymykseen: mitä pitää ottaa huomioon asymmetrisen VR moninpelin kehityksessä teknisen toteutuksen näkökulmasta ja mitä hyötyjä ja haittoja on paikallisen- ja verkkomoninpelin kehityksessä?

Opinnäytetyötä varten tein tutkimuksen, missä mitattiin tietokoneen näytönohjaimen ja suorittimen tarvitsemaa aikaa näyttääkseen yhden pelin ruudun. Tutkimuksen tavoitteena oli löytää versio, joka vie vähiten aikaa ruudun piirtämiseen. Hypoteesiksi esitin paikallista moninpeliversiota. Testitulosten perusteella optimoin lähiverkkomoninpeli-versiota ja tein testit uudestaan.

Lopullisten testaustulosten mukaan vähiten aikaa ruudun piirtämiseen menee lähiverkkomoninpeli-versiossa, mutta vain näppäimistö-hiiripelaajalla.

Kieli
suomi

Sivuja

45

Asiasanat

Asymmetria, VR, LAN, FPS



Thesis
September 2019
Degree Programme in Business IT
Bachelor's Thesis
Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600 (switchboard)

Author
Miika Piitulainen

Research and development of an asymmetrical vr game on unreal engine 4 – game engine

Commissioned by
Vaki Oy

Abstract

In this thesis, I will develop an asymmetric multiplayer game using the Unreal Engine 4 game engine. The game is between two players, with one player playing with the keyboard and mouse and the other playing with virtual reality glasses. There are two versions of the game where the multiplayer implementation is different. In my thesis I compare the versions to each other and try to answer two research questions: what should be taken into consideration in the development of an asymmetrical VR multiplayer from a technical implementation point of view and what are the advantages and disadvantages of local and online multiplayer versions?

For the thesis, I did a research test to measure the time it takes for the test computer's graphics card and processor to display a single frame in the game. The aim of the research was to find the version that takes the least time to draw a frame of the game. As a hypothesis, I proposed the local multiplayer version to be the fastest. Based on the test results, I optimized the LAN multiplayer version and did the tests again.

According to the final test results, the least amount of time it takes to draw a single frame in the game was the LAN multiplayer version, but only for the player playing with the keyboard and the mouse.

Language

Finnish

Pages

45

Keywords

Asymmetry, VR, LAN, FPS

Sisältö

1	Johdanto	6
2	Unreal Engine 4-pelimoottori	6
3	Virtuaalinen todellisuus	7
4	Asymmetrisyys moninpeleissa	9
4.1	Esimerkkejä asymmetrisistä VR moninpeleistä	10
4.2	Keep Talking And Nobody Explodes	10
4.3	Nemesis Perspective	12
5	Asymmetristen pelien suunnittelu	13
6	Asymmetristen pelien kehitys	19
6.1	Projektin aloitus ja tason muokkaus.....	19
6.2	Koodinsyöttöruudukko VR-pelaajan pelitilassa	22
6.3	Ongelmanratkaisunäytöt näppäimistö-hiiripelaajan pelitilassa.....	26
6.4	Näppäimistö-hiiripelaajan ohjelmointi	29
6.4.1	Pelihahmon ohjelmointi paikallisen moninpelellisen versiossa.....	29
6.4.2	Pelihahmon ohjelmointi verkkomoninpeliversiossa.....	30
6.4.3	Molempien versioiden yhteinen ohjelmointiosa	31
6.5	Verkkomoninpeliversioiden tietokoneiden yhdistäminen	32
6.6	Asymmetrisen pelin lopullinen pelitila	33
7	Kehitettyjen pelien vertailu	34
7.1	Työkalut	34
7.2	Testitapaukset	35
7.3	Testauksen hypoteesi.....	36
7.4	Testitulokset.....	36
7.5	Testitulosten vertailu	38
7.6	Optimointi.....	38
8	Testauksen yhteenveto	39
9	Pohdinta.....	40
9.1	Opinnäytetyötä varten kehitetyn pelin vertailu muihin asymmetrisiin VR peleihin	42
9.2	Itsearviointi.....	42

Lyhenteet

Asymmetria	Symmetrian vastakohta, epäsymmetrinen.
Moninpeli	Peli tai pelin osa, missä useampi kuin yksi pelaaja pystyy pelaamaan peliä samaan aikaan.
FPS	Frames Per Second, eli luku, joka ilmaisee ruudunpäivitysnopeuden sekunnissa
LAN	Local Area Network, eli lähiverkko. Rajatulla alueella toimiva verkko tietokoneiden välillä

1 Johdanto

Opinnäytetyön tavoitteena on kehittää kaksi eri versiota samasta asymmetrisestä moninpelistä ja vastata seuraaviin tutkimuskysymyksiin: mitä pitää ottaa huomioon asymmetrisen VR moninpelin kehityksessä pelattavuuden näkökulmasta ja mitä hyötyjä ja haittoja on paikallisen- ja verkkomoninpelin kehityksessä?

Opinnäytetyötä varten kehitetään kaksi eri versiota samasta asymmetrisestä moninpelistä. Molemmat pelit kehitetään Unreal Engine 4 - pelimoottorissa. Molemmissa peliversioissa pelin tapa käsitellä asymmetristä moninpeliä on eri. Ensimmäisessä versiossa (versio a) moninpeli on toteutettu yhdelle koneelle, johon molemmat ohjaimet ovat yhdistettynä. Toisessa versiossa (versio b) moninpeli on toteutettu kahdelle koneelle, jotka ovat yhdistettynä samaan paikalliseen verkkoon. Molemmat ohjaimet ovat tässä versiossa yhdistettynä vain yhteen tietokoneeseen. Molemmista versioista mitataan näytönohjaimen ja suorittimien tarvitsemaa aikaa näyttää pelin yksi ruutu näytöllä ja kokonaisruudunpäivitysnopeutta.

2 Unreal Engine 4-pelimoottori

Pelimoottoreilla tarkoitetaan sovelluksia, millä voi kehittää pääasiassa videopelejä. Pelimoottorin tehtävä on käsitellä esim. pelin fysiikoita, tekoälyä ja kuvan renderöintiä, jotta pelikehittäjä voi keskittyä ohjelmoimaan pelin logiikkaa. [1.]

Unreal Engine 4 on neljäs iteraatio Epic Gamesin kehittämästä Unreal Engine - pelimoottorista. Unreal Engine 4:ssä on mahdollista toteuttaa ohjelmointi kokonaan graafisesti Blueprint-mekaniikalla [2]. Opinnäytetyössä molempien pelien ohjelmointi on tehty kokonaan Blueprint-mekaniikkaa käyttäen.

Blueprint ohjelmointi on visuaalinen ohjelmointitapa Unreal Engine 4 - pelimoottorissa. Kirjoitetun koodin sijasta ohjelmointi tapahtuu visuaalisesti

käyttäen eri "solmuja". Solmuja voidaan yhdistää toisiinsa vetämällä solmuissa olevia sisääntulo- ja ulostuloverkkoja toisiinsa. Jokaisessa tasossa on oma Level Blueprint, minkä avulla voi vaikuttaa tason sisäisiin ominaisuuksiin. Tasolla tarkoitetaan ympäristöä, missä yhdistelmä eri pelin elementtejä kuten valot, blueprintit, esineet jne. luovat yhdessä pelaajalle halutun kokemuksen [3]. Class Blueprint on paljon monimuotoisempi ja niitä voidaan yhdistää tiettyihin elementteihin peleissä kuten esineisiin ja vaikuttaa vain niihin [4].

Blueprint Class tarvitsee toimiakseen kantaluokan. Käytetyimmät kantaluokat ovat Actor, Pawn, Character, PlayerController ja Game Mode kantaluokat. Actor kantaluokan avulla pystytään vaikuttamaan elementtiin, joka on sijoitettu tasoon. Pawn kantaluokka on sama kuin Actor kantaluokka, mutta voit hallita elementtiä esim. liikuttamalla sitä näppäimistön avulla. Character kantaluokka on Pawn kantaluokkaan perustuva kantaluokka, missä on liikkumiseen tarkoitettu ohjelmointi jo tehtynä. PlayerController on Actor kantaluokkaan perustuva kantaluokka, mikä vastaa Pawn-kantaluokkaisten blueprinttien hallinnasta. Game Mode kantaluokan avulla pystyy määrittelemään, mitä peliä pelataan, mitkä sen säännöt ovat ja muita itse peliin liittyviä asioita [5.]

3 Virtuaalinen todellisuus

Virtuaalinen todellisuus (engl. virtual reality), on simuloitu virtuaalinen maailma, jota pystyy katsomaan virtuaalitodellisuuslaseja käyttäen [6]. Yleensä virtuaalitodellisuuslasien sisällä on kaksi näyttöä molemmille silmille. Kun virtuaalitodellisuuslasit laitetaan päähän, molemmille lasien näytöille heijastetaan kuvaa, mikä antaa kolmiulotteisen kuvan vaikutelman. Useimmissa virtuaalitodellisuuslaseissa on myös sensorit, mitkä havaitsevat pään liikkumisen. Tämä mahdollistaa 360° liikkumisen virtuaalitodellisuusympäristössä. Virtuaalinen todellisuus voidaan rinnastaa lisättyyn todellisuuteen (Augmented Reality) ja yhdistettyyn todellisuuteen (Mixed Reality). Lisätyssä todellisuudessa fyysiseen todellisuuteen lisätään virtuaalisia elementtejä, joita voi katsoa lisälaitteen kautta [7]. Yhdistetty todellisuus on virtuaalinen tila, missä fyysisen

todellisuuden ja virtuaalitodellisuuden elementit pystyvät olemaan vuorovaikutuksessa toistensa kanssa [8].

Keskeisin ero edellä mainittujen teknologioiden välillä on niiden suhde fyysiseen todellisuuteen ja miten pelaaja uppoutuu pelimaailmaan, eli immersioon: Virtuaalisessa todellisuudessa pyritään täydelliseen immersioon virtuaalisessa maailmassa ja fyysisen maailman elementtejä ei hyödynnetä. Lisätyssä todellisuudessa pyritään "lisäämään" virtuaalisen todellisuuden elementtejä oikeaan maailmaan, jota pystyy katsomaan ja olemaan vuorovaikutuksessa esim. puhelimen kautta. Lisätyssä todellisuudessa immersio virtuaaliseen maailmaan on vähäistä. Yhdistetyssä todellisuudessa pyritään immersiiiviseen kokemukseen, missä fyysinen maailma ja virtuaalinen maailma pystyy olemaan vuorovaikutuksessa toistensa kanssa [9.]

Ensimmäinen konsepti virtuaaliselle todellisuudelle oli stereoskoopit 1800-luvulla. Stereoskoopissa näytetään yksi kuva molemmille silmille, jotka on otettu hieman eri kuvakulmasta. Kun kuvia katsoo molemmilla silmillä, syntyy illuusio kolmiulotteisuudesta. Ensimmäinen todellinen virtuaalitodellisuuskokemus oli Mort Heiligin patentoitu pelihallipelikabinetin tapainen laite, missä näytettiin 3d kuvaa pelin näytöltä [10.]

Ensimmäinen päässä pidettävä virtuaalitodellisuuslaite oli Heiligin vuonna 1960 kehittämä Telesphere Mask – virtuaalitodellisuuslasit. Nämä lasit olivat kuitenkin hyvin alkukantaisia, eikä niissä ollut minkäänlaista liikkeentunnistusta [10.]

Seuraava harppaus oli virtuaalitodellisuuslasit, missä on liikkeen tunnistus. Nämä lasit olivat kahden Philco Corporationin insinöörin kehittämiä "The Headsight" virtuaalitodellisuuslasit, mitkä kehitettiin vuonna 1961. Laseissa oli magneetti ja erillinen kamera seurasi pään liikkeitä sen avulla. Nämä olivat ensimmäiset virtuaalitodellisuuslasit, joissa ympäristöä voi vapaasti katsoa. [10.]

Peliyritys Sega yritti ensimmäisenä tuoda virtuaalitodellisuuslasit kuluttajille. Lasit yhdistettäisiin Segan julkaisemaan Sega Genesis konsoliin ja näyttäisi pelaajalle

kolmiulotteista kuvaa lasien avulla. Tämä kuitenkin jäi prototyypivaiheeseen, eikä ikinä julkaistun kuluttajille

Nintendo yritti seuraavaksi virtuaalitodellisuuslasien julkaisua vuonna 1995 ja onnistui julkaisemaan Virtual Boy-konsolin konsolimarkkinoille. Virtual Boy näyttää monokromista stereoskooppista punamustaa kuvaa näytöllänsä, mikä luo kolmiulotteisen vaikutelman käyttäjälle. Konsoli oli kuitenkin jäänyt myyntitavoitteistansa ja konsolin tuotanto ja myynti lopetettiin 1996 [10.]

Vuonna 2010 Palmer Luckey kehitti ensimmäisen prototyypin Oculus Rift-virtuaalitodellisuuslaseista. Virtuaalitodellisuuslasien kehitys alkoi edistyä enemmän prototyypin esittelyn myötä ja markkinoille syntyi enemmän virtuaalitodellisuuslaseja kuluttajille kuten HTC Vive, Playstation VR, Valve Index jne. [11.]

Virtuaalitodellisuuslasien kehitys on nykyisin keskittynyt helpottamaan liikkumista virtuaalisessa tilassa tekemällä VR-laseista johdottomia ja itsenäisiä laitteita, mitkä eivät tarvitse esim. ulkoisia sensoreita tai tietokonetta. Virtuaalista todellisuutta on myös alettu käyttää enemmän pelaamisen ulkopuolella esim. opiskeluissa. VR pelien määrä on myös noussut huomattavasti ja eri genrejen pelejä kehitetään enemmän [12.]

4 Asymmetrisyys moninpeleissa

Asymmetrisyydellä tarkoitetaan symmetrian eli yhdenmukaisuuden vastakohtaa [13]. Esimerkiksi symmetrisessä kuvassa kuva voidaan jakaa kahteen tai useampaan yhtäläiseen osaan, kun taas asymmetrisessä kuvassa niin ei voi tehdä.

Peleissä asymmetrisellä moninpelillä tarkoitetaan peliä, jossa joidenkin pelaajien näkymä tai rooli pelissä on erilainen kuin toisten pelaajien. Jotkin pelaajat pystyvät esim. olemaan vuorovaikutuksessa tiettyjen pelin elementtien kanssa, mihin toiset pelaajat eivät voi vaikuttaa [14.]

4.1 Esimerkkejä asymmetrisistä VR moninpeleistä

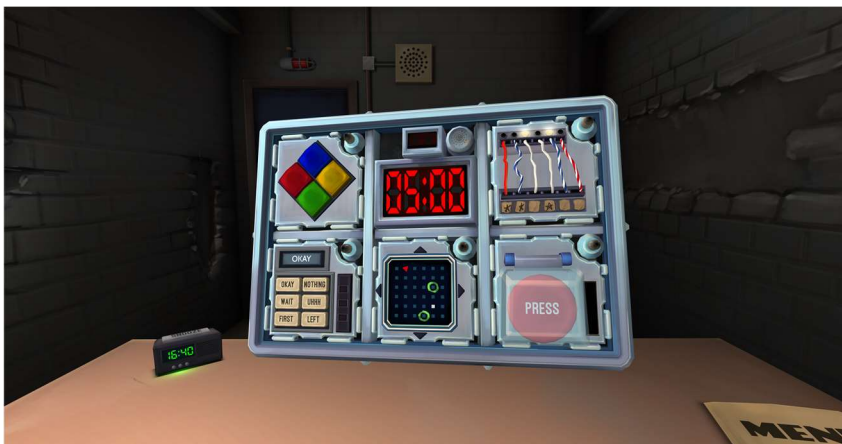
Näissä esimerkeissä pelit ovat asymmetrisiä moninpelejä joita voi pelata virtuaalitodellisuuslaseilla ja jollain muulla ohjaimella. Olen valinnut nämä kaksi peliä, koska olen itse kokeillut näitä pelejä ja ne ovat tuoneet inspiraatiota tähän opinnäytetyöhön.

Ensimmäiseksi esimerkiksi olen valinnut Keep Talking and Nobody Explodes-pelin, joka on asymmetrinen VR moninpeli. Peli on myös suosittu VR-peli, joka on saanut paljon myönteisiä arvosteluita Steam-kauppasivullansa [15].

Toinen esimerkki on Evocat Gamesin kehittämä Nemesis Perspective. Tämä peli on myös asymmetrinen VR-moninpeli, mutta lyhyempi pelattavuudeltaan.

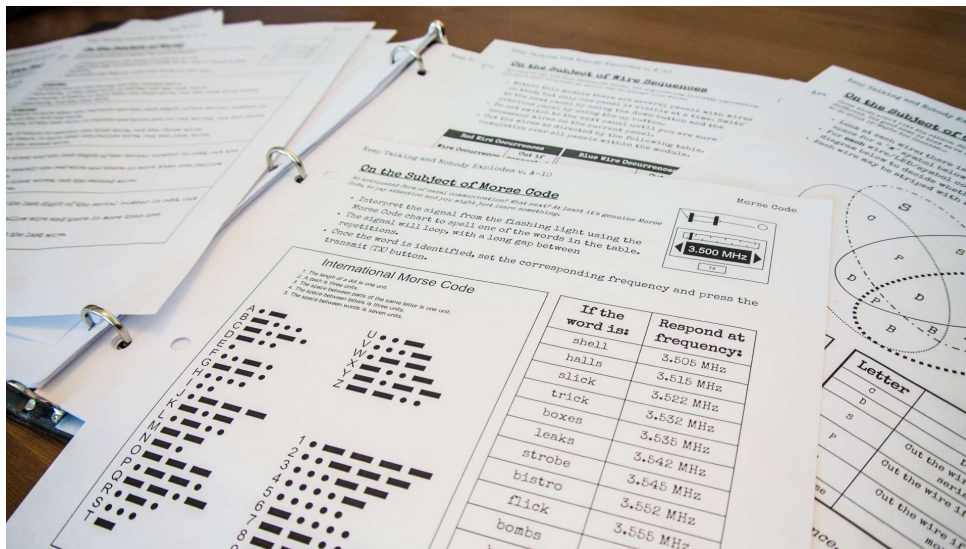
4.2 Keep Talking And Nobody Explodes

Keep Talking and Nobody Explodes on Steel Crate Gamesin kehittämä ja julkaisema peli. Peli on kehitetty Unity-pelimootorissa ja peli julkaistiin Gear VR laitteelle kesäkuussa 2015. Myöhemmin kyseinen peli on julkaistu useammalle alustalle. Peli on moninpeli, jossa toinen pelaa virtuaalitodellisuuslaseilla tai muulla ohjaimella ja toinen pelaaja antaa ohjeita toiselle pelaajalle ohjekirjan avulla. Peli on kehitetty virtuaalitodellisuutta varten, mutta myöhemmissä julkaisuissa pelaaminen onnistui ilman virtuaalitodellisuuslaseilla.



Kuva 1. Virtuaalitodellisuuslaseilla pelaavan pelaajan näkymä Keep Talking and Nobody Explodes pelissä.

Pelin tavoitteena on purkaa pommi yhdessä toisen pelaajan kanssa. Virtuaalitodellisuuslaseilla pelaavalla pelaajalla näkyy pommi, missä on eri osia. Osat täytyy deaktivoida ohjeiden mukaan. Ohjeet moduuleiden purkamiseen on toisella pelaajalla. Pelaajien täytyy kommunikoida pommin moduuleiden toiminnasta ja niiden mukaan purkaa pommi oikein. Jos pelaajille tapahtuu 3 virhettä pommin purun aikana, niin peli päättyy pommin räjähdykseen [16.]



Kuva 2. Ohjekirja pommin purkamiseen.

Pelin idea on lähtenyt Global Game Jam 2014-tapahtumasta, missä pitää kehittää tietyn teemainen peli 48 tunnissa. Teemana oli tuolloin ”Me emme näe asioita niin kuin ne ovat, me näemme ne niin kuin me olemme”. Ben Kane osallistui tapahtumaan ja halusi tehdä juuri VR-pelin, koska niitä ei ollut paljoa. Kehityksen aikana Ben antoi muiden kokeilla hänen VR-laseillansa vuoristoratademoa. Hän huomasi, että VR-pelaajalla oli erittäin hauskaa, mutta katsojilla ja jonottajilla ei ollut. Hän halusi kehittää pelin, missä katsojakin tulee mukaan peliin. Ben ja hänen tiimensä kehittivät ongelmanratkontapelin, missä molemmat pelaajat joutuvat auttamaan toisiansa läpäistäkseen pelin. Tämä peli oli erittäin suosittu ja sen pohjalta kehitettiin Keep Talking and Nobody Explodes-peli [17.]

4.3 Nemesis Perspective

Nemesis Perspective on Evocat Gamesin kehittämä ja Kajak Gamesin julkaisema asymmetrinen VR moninpeli. Pelissä VR pelaaja pelaa isolla hirviöllä, jonka tehtävänä on saada toisen pelaajan terveystmittari nolnaan. Toisen pelaajan terveystmittaria voi vähentää lyömällä toisen pelaajan pelihahmoa virtuaalitodellisuusohjaimilla.



Kuva 3. Nemesis Perspective: VR pelaajan näkökulma.

Toinen pelaaja ohjaa pienempää ihmishahmoa konsoliohjaimella ja pyrkii saamaan VR-pelaajan pelihahmon terveystmittarin nolnaan. Ihmishahmo pystyy hyppimään, tekemään miinoja, juoksemaan tasolla ja hyökkäämään VR-pelihahmoa kohti lyömällä.



Kuva 4. Nemesis Perspective: Peliohjain-pelaajan näkökulma.

Peli on saanut Steam sivuillansa myönteisiä arvioita. Peliä on keuhuttu esim. pelikonseptista ja moitittu hieman sisällön puutteesta [18; 19].

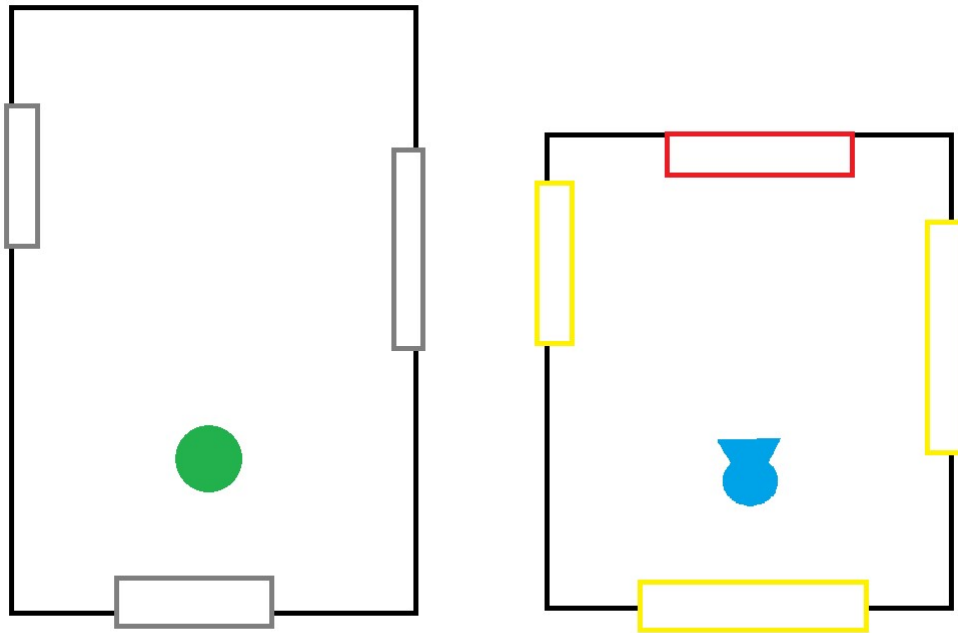
5 Asymmetristen pelien suunnittelu

Asymmetrisen pelin tavoitteena on syöttää tietty koodi VR-pelaajan pelitilassa olevaan ruudukkoon. Ruudukko on 3x3 ruudukko, minkä ruutuja voi VR-pelaaja painaa ohjaimen päässä olevan pallon avulla. Oikean koodin saamiseksi pelaajien täytyy yhdessä selvittää kolme laskutoimitusta, mitkä sijaitsevat VR-pelaajan seinillä. Laskutoimituksen vastaus tulee syöttää näyttöön, joka on saman puoleisessa seinässä VR-pelaajan laskutoimituksen kanssa. Esim. VR-pelaajan pelitilan vasemman seinän laskutoimituksen tulos tulee syöttää näppäimistö-hiiripelaajan pelitilan vasemmalla seinällä olevaan näyttöön.

Kun vastaus on syötetty, niin näyttöön ilmestyy koordinaatti. Tämä koordinaatti vastaa jotakin VR-pelaajan ruudukossa olevaa ruutua. Koordinaatin kirjaimet vastaavat ruudukon rivejä ja koordinaatin numerot sarakkeita. Kun oikea koodi on sijoitettu, niin peli onnittelee VR-pelaajaa ja sulkee pelin automaattisesti.

Pelin alussa pelaajat seisovat keskellä omaa pelialuetta. Jos peli on lähiverkkomoninpeli-versio, niin peli ensin yhdistää toiseen tietokoneeseen.

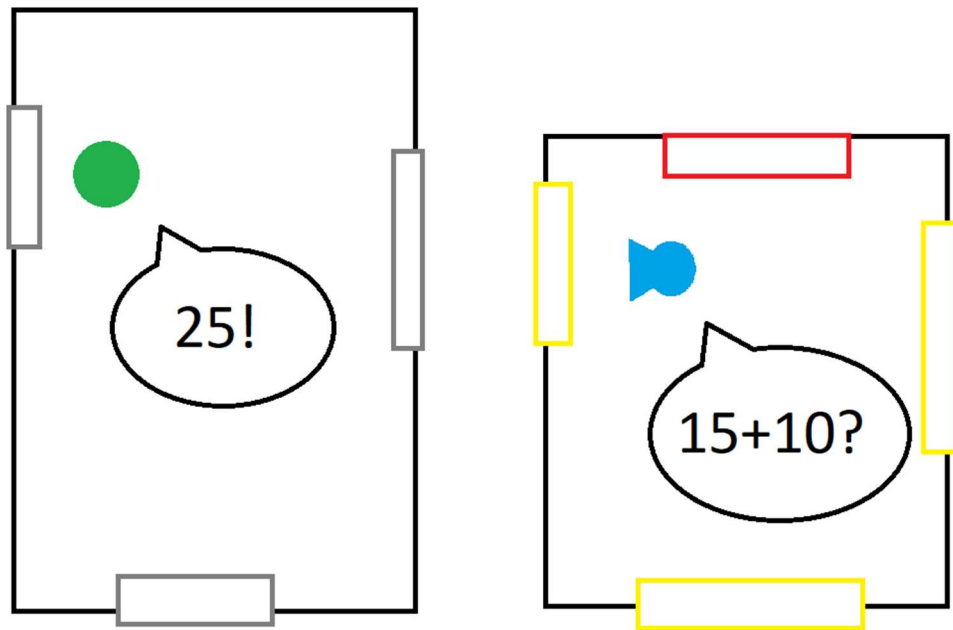
Pelaajat voivat vapaasti liikkua omassa pelialueessansa pelin aikana.



Kuva 5. Konsepti pelin alkutilanteesta ylhäältä päin katsottuna.

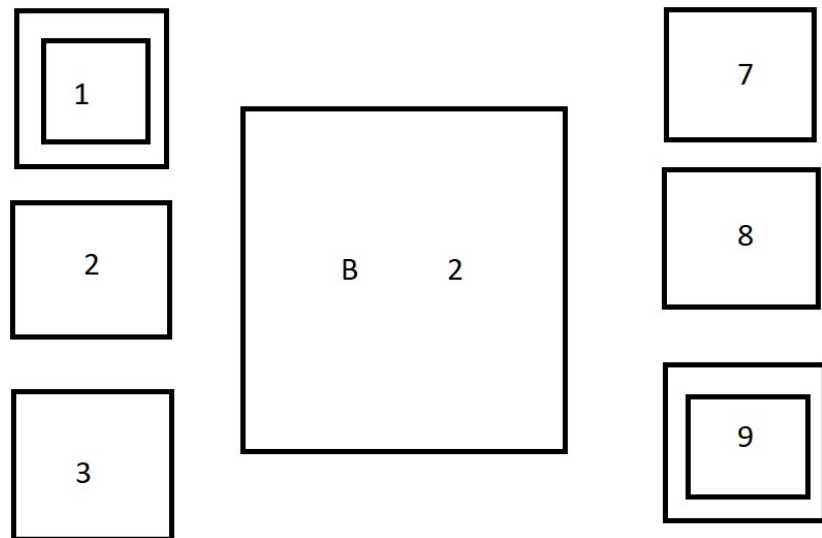
Kuvassa 5 nähdään pelin suunniteltu alkutilanne. Oikealla puolella on VR-pelihahmo, joka on esitetty sinisellä värillä. VR-pelihahmon pelialueella on kolme laskutoimitusta eri seinillä, mitkä on kuvassa 5 ilmaistu keltaisella värillä. VR-pelihahmon pelialueen yhdellä seinällä sijaitsee 3x3 taulu, mihin lopullinen koodi pitää sijoittaa. Tämä taulu on kuvassa 5 osoitettu punaisella.

Kuvan 5 vasemmalla puolella on näppäimistö-hiiripelaajan pelialue. Pelaaja on esitetty kuvassa 5 vihreällä pallolla. Pelialueessa on kolme eri näyttöä, johon tulee syöttää VR-pelaajan pelialueella olevan laskutoimituksen tulos. Näytöt on sijoitettu melkein samanlaisiin kohtiin seinillä kuin VR-pelialueen seinillä. Kun tulos on syötetty, niin näyttöjen keskelle ilmestyy koordinaatti. Tämä koordinaatti voi olla väärä, mikäli tulos on virheellinen.



Kuva 6. Konseptikuva pelitilanteesta, jossa ratkaistaan seinällä olevia pulmia.

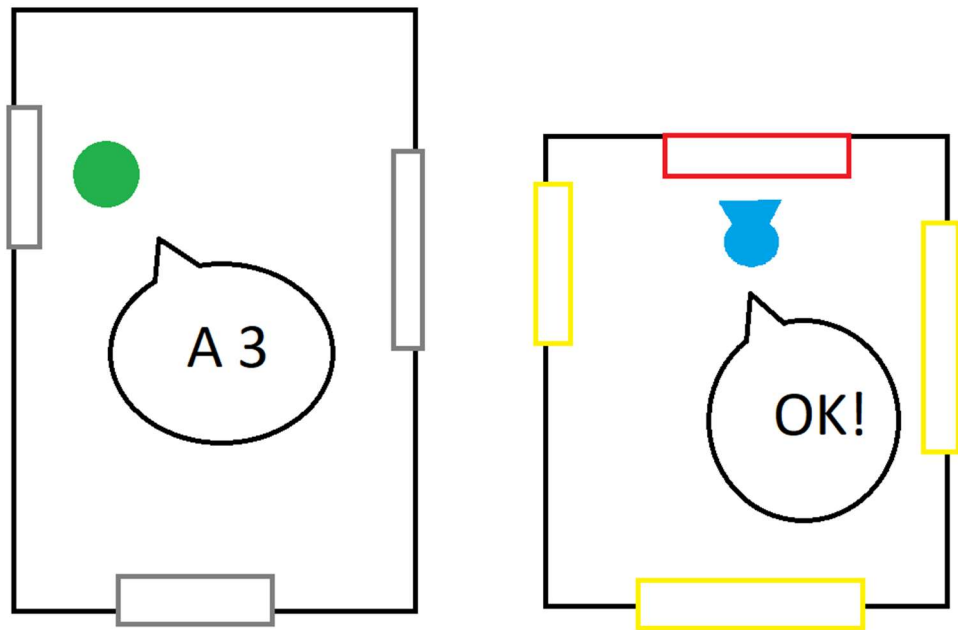
Kuvassa 6 kuvataan pelitilanne, missä pelaajat kommunikoivat pelin ongelmanratkaisutehtävistä toisillensa. VR-pelaajan vasemmalla seinällä oleva pulma kysyy vastausta laskutoimitukseen $15+10$. VR-pelaajan pitää kommunikoida fyysisesti puhumalla toiselle pelaajalle, mikä laskutoimitus on ja yhdessä selvittää sen tulos. Kun tulos on saatu, niin näppäimistö-hiiripelaaja syöttää tuloksen saman puolen seinän näyttöön.



Kuva 7. näppäimistö-hiiripelaajan pelialueella olevan näytön konseptikuva.

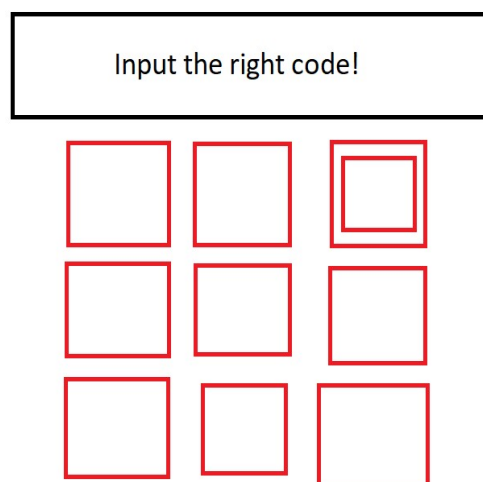
Kuvassa 7 esitetään yksi näppäimistö ja hiiri-pelaajan pelialueella olevista näytöistä. Tässä esimerkissä pelaaja on valinnut vasemmanpuoleisesta sarakkeesta numeron 1 ja oikeanpuoleisesta sarakkeesta numeron 9. Nämä numerot yhdistetään ja vastaukseksi tulee 19. Esimerkiksi jos VR-pelaajan alueella on ongelma $19+19$, niin pelaaja syöttäisi vasemmanpuoleiseen sarakkeeseen 3 ja oikeanpuoleiseen sarakkeeseen 8. Näin tulos olisi 38.

Kuvan 7 keskellä olevaan näyttöön syntyy koordinaatti valitun vastauksen perusteella. Vasemmanpuoleisen sarakkeen valinta muodostaa koordinaatin kirjaimen ja oikeanpuoleisen sarakkeen valinta muodostaa koordinaatin numeron. Molemmista sarakkeista voidaan vain valita yksi ruutu. Esimerkiksi jos pelaaja valitsee vasemmanpuoleisesta sarakkeesta 1, niin kirjainkoordinaatti olisi B. Jos pelaaja valitsee vasemmanpuoleisesta sarakkeesta 2, niin kirjainkoordinaatti on eri. Koordinaatin kirjaimet ovat A:sta C:en ja numerot ovat yhdestä kolmeen.



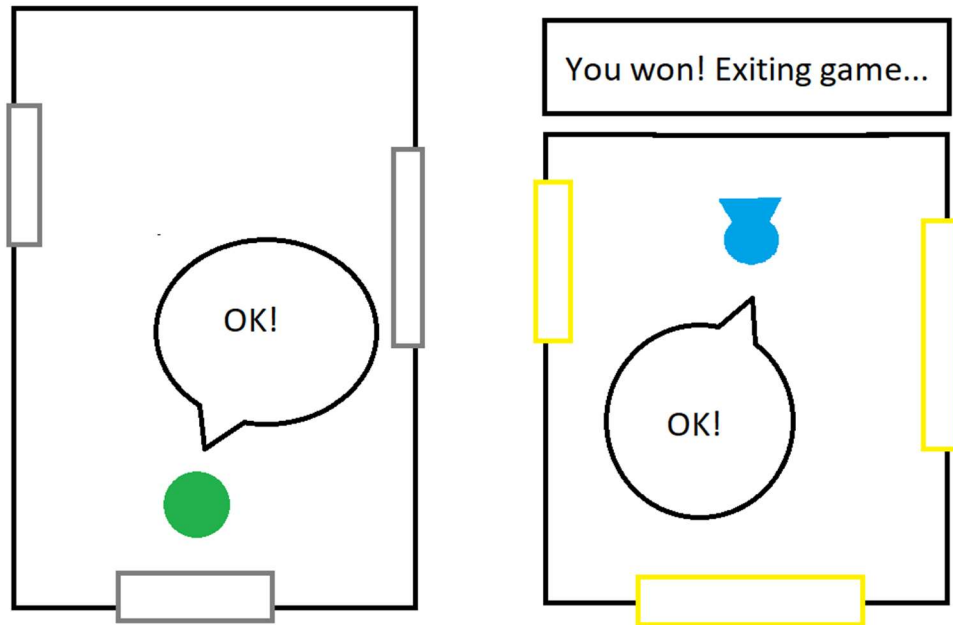
Kuva 8. Konseptikuva pelitilanteesta, jossa pelaajien täytyy kommunikoida pulman tulokseen viittaava koordinaatti.

Kun näppäimistö-hiiripelaaja syöttää tuloksen näyttöön, niin näyttö antaa tietyn koordinaatin. Kuvassa 8 näyttö näyttää koordinaatiksi A3. Näppäimistö-hiiripelaajan täytyy sanoa koordinaatti VR-pelaajalle ja VR-pelaajan täytyy syöttää tulos ruudukkoon.



kuva 9. Konseptikuva VR-pelaajan pelialueella olevasta ruudukosta

Kuvassa 9 kuvaillaan konsepti VR-pelaajan pelialueessa olevasta ruudukosta. Ruudukko koostuu kolmesta rivistä ja kolmesta sarakkeesta. Näppäimistö-hiiripelaajan näyttöjen koordinaatit vastaavat tiettyä riviä ja saraketta ruudukossa. Koordinaatin kirjaimet ilmaisevat ruudukon rivejä ja koordinaatin numerot ilmaisevat ruudukon sarakkeita. Kuvassa 9 VR-pelaaja on syöttänyt koordinaatin A3, eli ensimmäisen rivin kolmas sarake.



Kuva 10. Konsepti pelin lopputilanteesta.

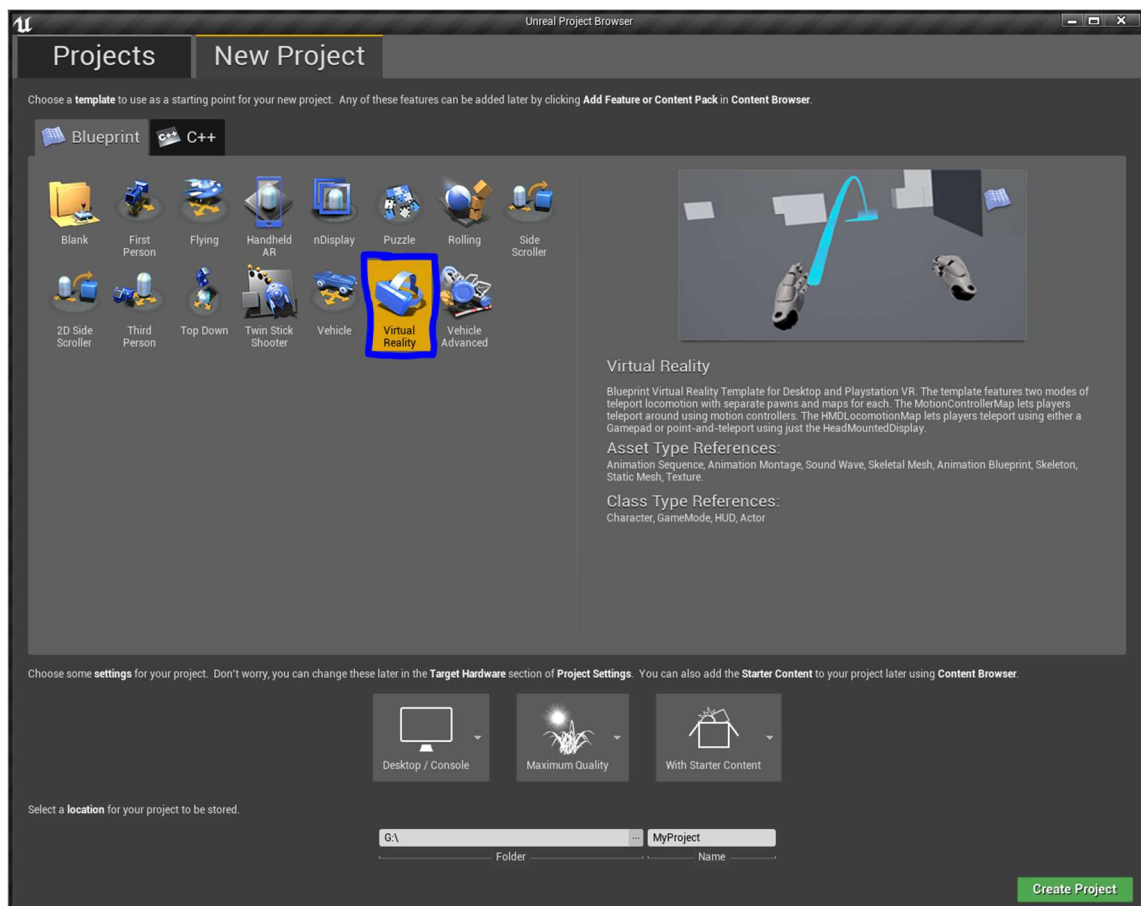
Kuvassa 10 kuvaillaan pelin lopputilannetta. Kun kaikki ongelmat on selvitetty, jokaiseen näyttöön on syötetty oikea tulos ja koordinaatit on syötetty VR-pelaajan ruudukkoon, niin peli onnittelee pelaajaa ja lopettaa pelin. Jos koodi on väärä, niin peli ei pääty. Jos koodi on väärä, niin molempien pelaajien täytyy tarkistaa, onko jokin pulman vastaus väärä. Väärä vastaus pulmaan antaa väärän koordinaatin VR-pelaajan ruudukkoon ja estää pelin voittamisen.

6 Asymmetristen pelien kehitys

Pelin kehitys aloitettiin ohjelmoimalla aluksi pelin hahmot ja pelimekaniikat konseptikuvien perusteella. Ohjelmoinnin jälkeen ohjelmoitiin moninpeliominaisuudet pelin eri versioihin. Ohjelmoinnissa on käytetty apuna Unreal Engine 4 - pelimoottorin omaa VR esimerkkiprojektia pelin pohjana.

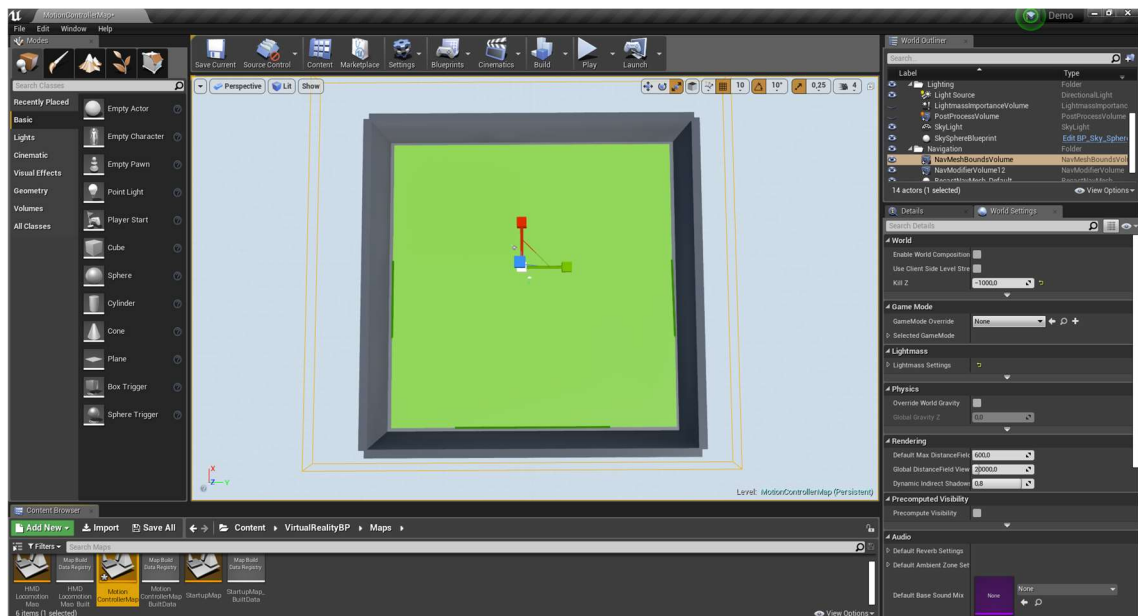
6.1 Projektin aloitus ja tason muokkaus

Molemmissa versioissa aloitin tekemällä uuden projektin, missä on käytetty Unreal Enginen omaa VR templatea, eli esimerkkiprojektia. Näissä esimerkkiprojekteissa on valmiina VR-pelaajalle vaaditut kontrollit ja asetukset.



Kuva 11. Uuden projektin aloittaminen.

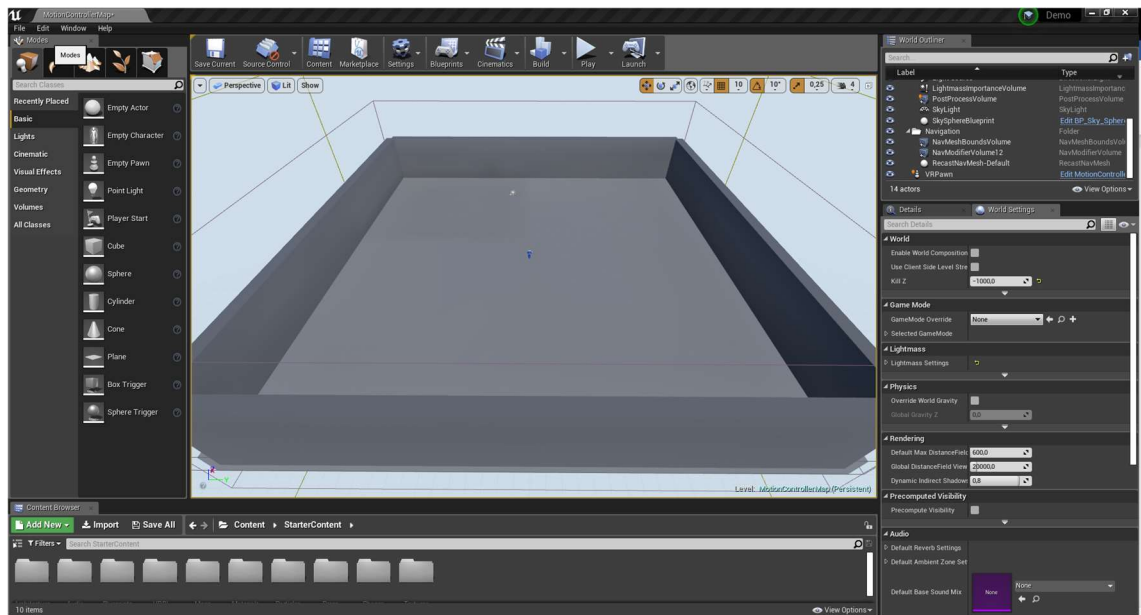
Projektin avattua esiin tulee esimerkkitaso. Lähden ensin poistamaan kaikki ylimääräiset tasot, joita en tarvitse projektissa. Päädyin tekemään pelin MotionControllerMap-nimisessä tasossa. Tässä tasossa on asetettu valmiiksi pelaajahahmo, jota voi VR-laseilla pelaava pelaaja kontrolloida. Esimerkkitasossa on myös valmiiksi navigaatioon tarvittava NavMeshBoundsVolume. NavMeshBoundsVolume on laatikko, joka osoittaa tietyn alueen, missä tekoäly tai pelaaja voi liikkua [20]. Tässä pelissä NavMeshBoundsVolume määrittää pelialueen, missä VR-pelaaja voi liikkua.



kuva 12. NavMeshBoundsVolume tasossa.

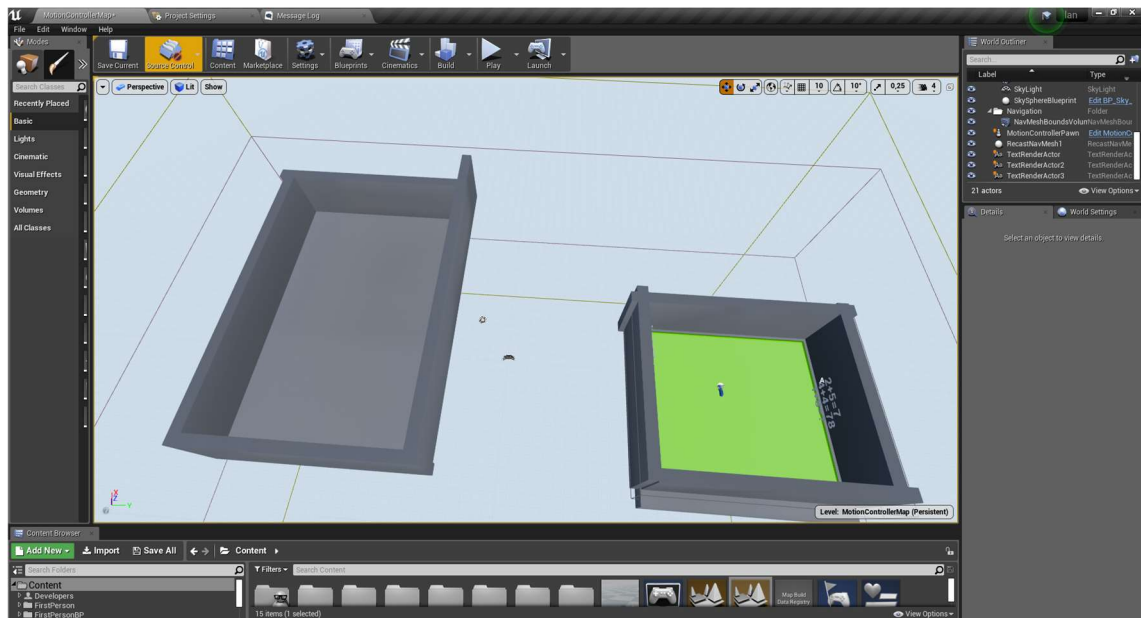
Kuvassa 12 NavMeshBoundsVolume on sijoitettu VR-Pelaajan pelialueelle. VR-pelaajan liikkumista on rajoitettu vain vihreälle alueelle tasossa.

Esimerkkitasossa lähdin poistamaan ylimääräiset kuutiot, törmäystarkastukset ja tekstit mitä löysin tasosta. tasoon jää vain itse pelaaja, valon lähde, taivaskupoli, seinät, lattia ja NavMeshBoundsVolume.



Kuva 13. VR-pelaajan siivottu pelitila

Seuraavaksi lähdin kopiaimaan tasossa olevat seinät ja lattian toisen pelaajan tilaa varten. Kopiointin jälkeen skaalasin molempien pelitiloja pienemmiksi, jotta pelaajien ei tarvitse niin pitkiä matkoja liikkua. Sen jälkeen laitoin VR-pelaajan pelitilan seinille tekstit, joissa on pelin laskutoimitukset.

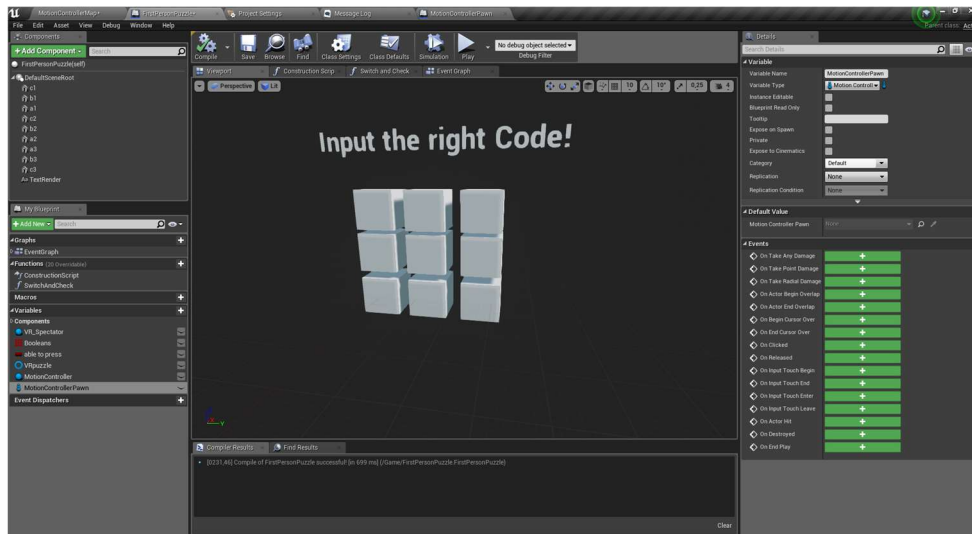


Kuva 14. Molempien pelaajien pelialue

Kuvassa 14 on molempien pelaajien pelikistetty pelialue. Oikealla puolella on VR-pelaajan pelialue ja vasemmalla puolella on näppäimistö-hiiripelaajan pelialue. VR-pelaajan pelialue on pienempi, koska VR-pelaajan liikkuminen alueella ei ole

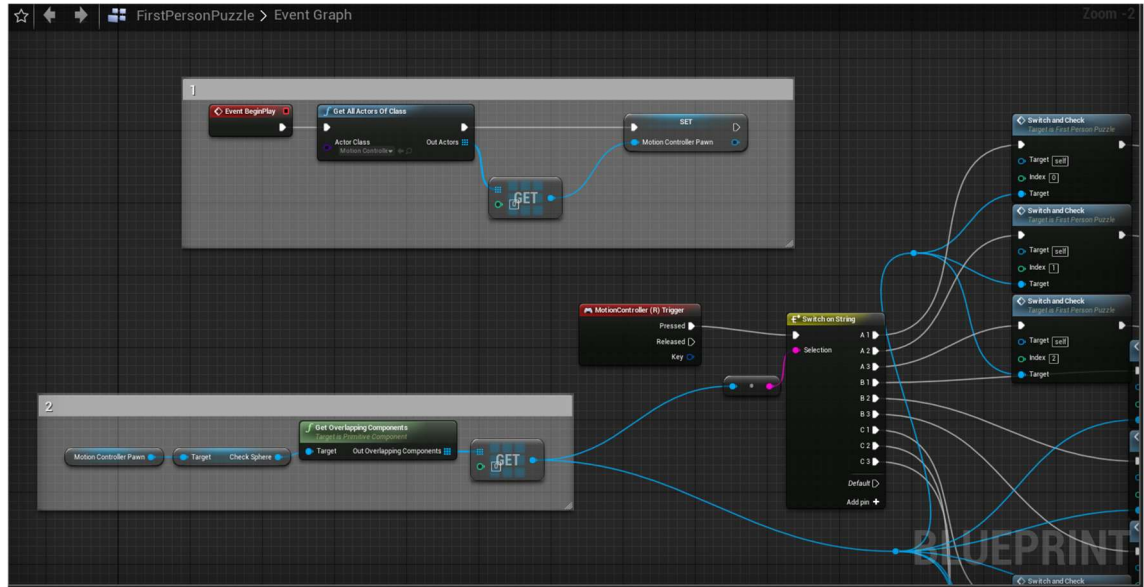
nopeaa. Näppäimistö-hiiripelaajan pelialue on isompi, koska pelaaja liikkuu nopeasti alueella.

6.2 Koodinsyöttöruudukko VR-pelaajan pelitilassa



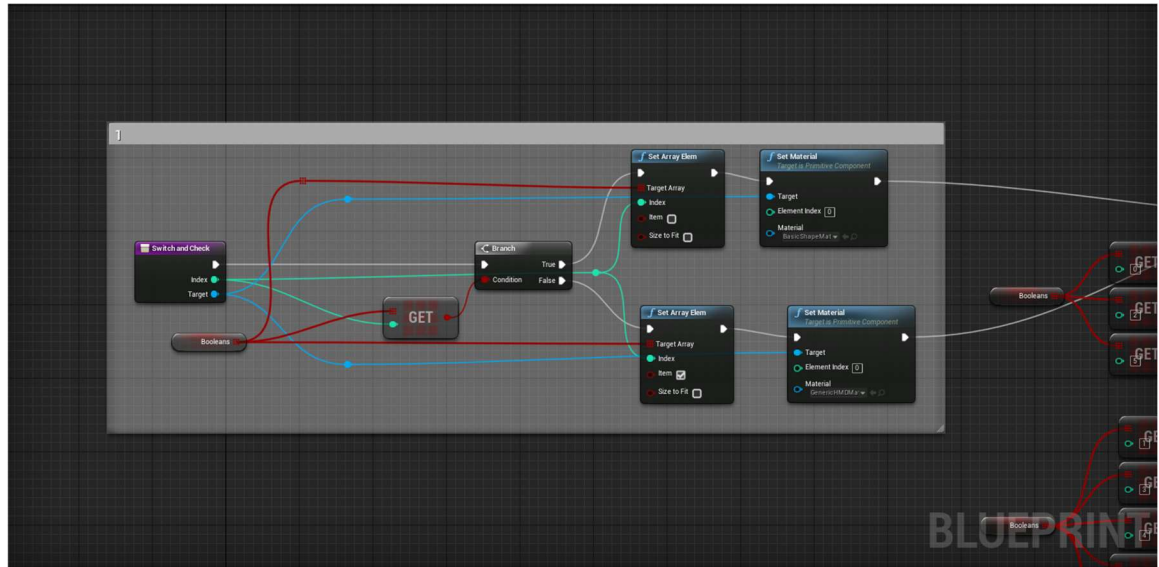
Kuva 15. Koodinsyöttöruudukon esikatselu.

Kuvassa 15 havaitaan VR-pelaajan pelitasossa olevan ruudukon blueprintti. Blueprintin nimi on FirstPersonPuzzle. Blueprint koostuu yhdeksästä Static Mesh-tyyppisestä kuutiosta, joita voi VR-lasien ohjaimen liipaisimesta painamalla aktivoida. Ruudukon yläpuolella on myös teksti "Input the right code!", mikä kehoittaa pelaajaa syöttämään oikean koodin.



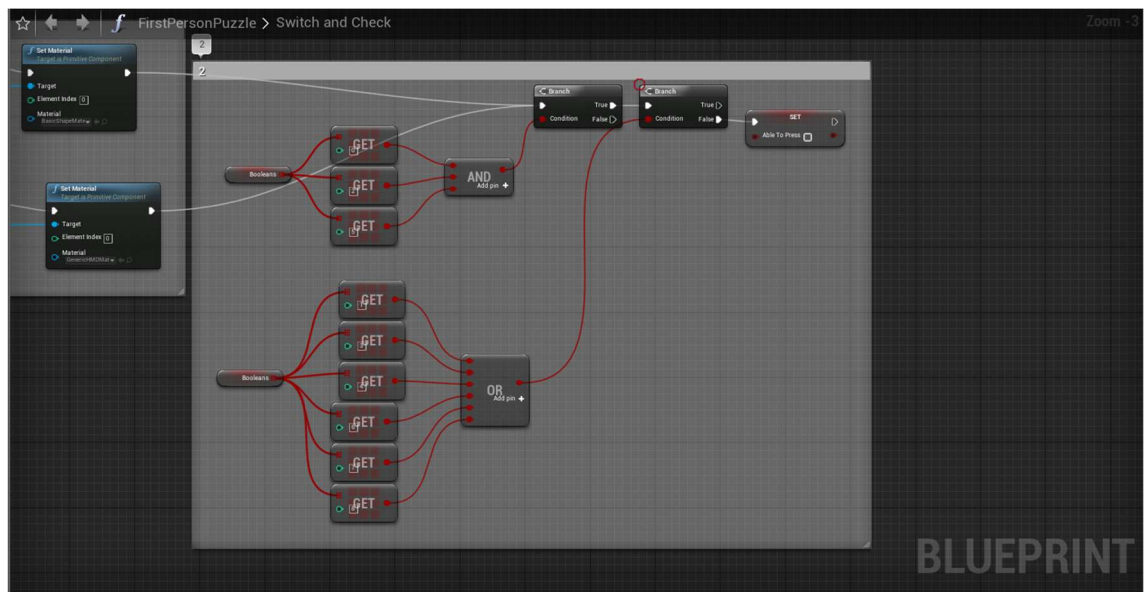
Kuva 16. Koodisyöttöruudukon FirstPersonPuzzle Blueprintin koodin alkupuoli

Kuvassa 16 ruudukon ohjelma aloitetaan ruudusta 1, missä se etsii VR-pelaaja tilasta ja muuttamalla se muuttujaksi "Motion Controller Pawn". Tämä auttaa blueprinttiä kommunikoimaan VR-pelaajan kanssa, mitä me tarvitsemme aktivoidaksemme kuutiot. Ruudussa 2 ohjelma etsii VR-pelaajan ohjaimessa olevan "Check Sphere"-pallon. Tarvitsemme viittauksen "Motion Controller Pawn" muuttujaan jossa pallo on. Sitten ohjelma katsoo, osuuko "Check Sphere" mihinkään esineeseen ja ottaa ensimmäisen esineen nimen, mihin se osuu. Kun VR-pelaaja painaa oikean ohjaimen liipaisinta, niin ohjelma katsoo, mikä on osuneen esineen nimen ja suorittaa tietyn "Switch And Check" funktion, mikäli esineen nimi on "Switch on String" listassa. Funktio ottaa vastaan osuneen esineen nimen ja indexin, joka riippuu esineen nimestä.



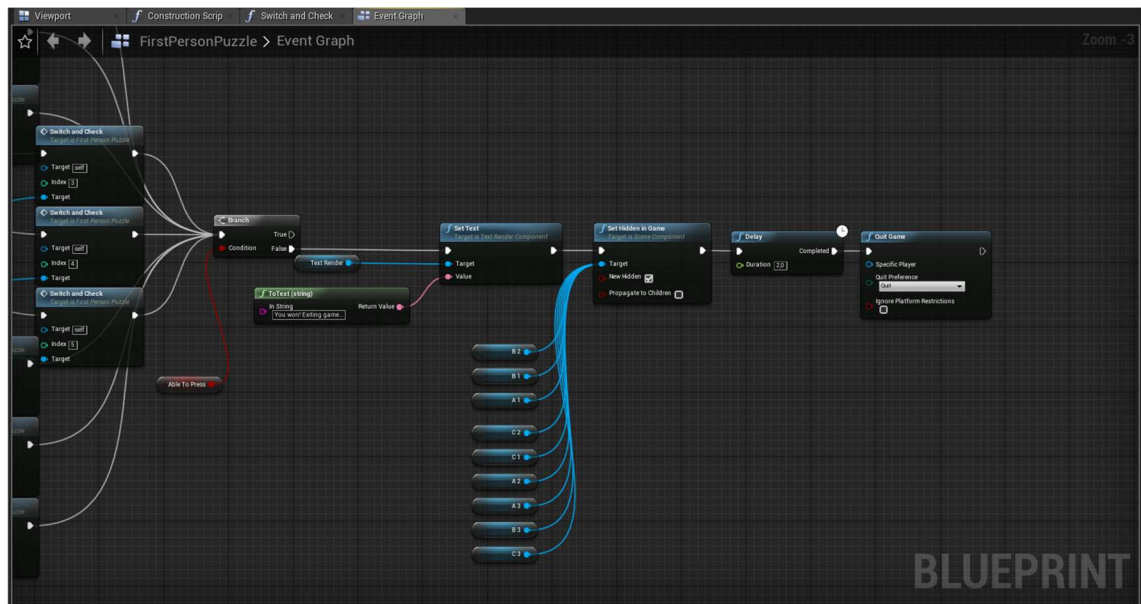
Kuva 17. Switch And Check funktion ensimmäinen osa.

Kuvassa 17 esitetään ensimmäinen osa Switch And Check -funktioista. Kuvan 19 funktiossa ensin tarkistetaan, onko osunut kuutio aktiivinen vai ei. Tämä tarkistetaan "Booleans" merkkijonon avulla. Funktio etsii funktion ottaman indeksin avulla merkkijonosta kyseistä numeroa. Jos numeron arvo jonossa on tosi, eli kuutio on aktiivinen, niin kuutio muutetaan epäaktiiviseksi. Sama myös toisinpäin, jos kuutio ei ole aktiivinen. Arvo sitten muutetaan päinvastaiseksi. Aktivointi ja deaktivointi ilmenee kuution väristä: aktiivinen kuutio on musta ja epäaktiivinen on valkea.



kuva 18. Oikean koodin tarkistus Switch and Check-funktiossa.

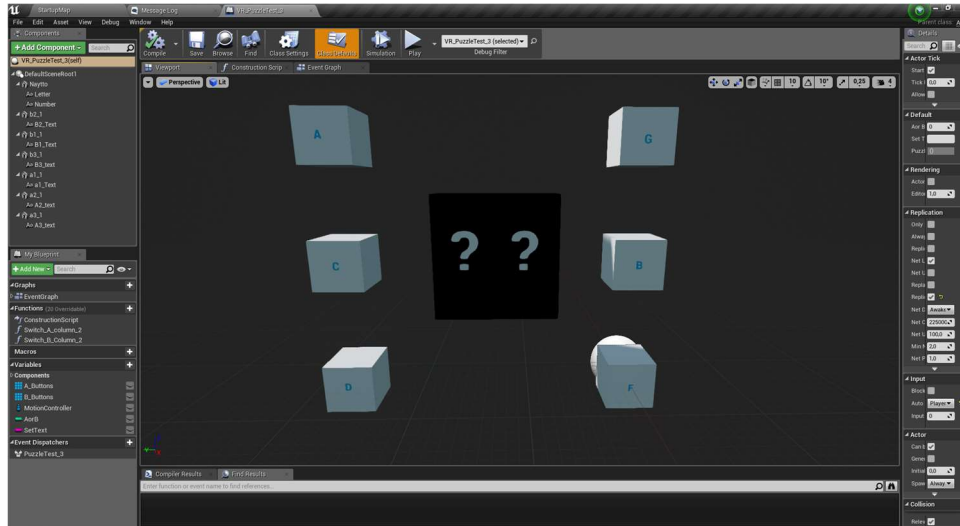
Kuvassa 18 esitetään Switch And Check -funktion koodintarkistusosa. Funktion osa katsoo, onko koodi oikea. Koodi on oikea, jos ”booleans” merkkijonon tietyt arvot ovat totta ja kaikki muut arvot ovat epätotta. Jos koodi on oikea, niin ”able to press” muutetaan epätodeksi ja funktio päättyy. Jos koodi on vielä väärin, niin funktio päättyy, eikä muuta ”able to press” -arvoa.



Kuva 19. FirstPersonPuzzlen loppuosa.

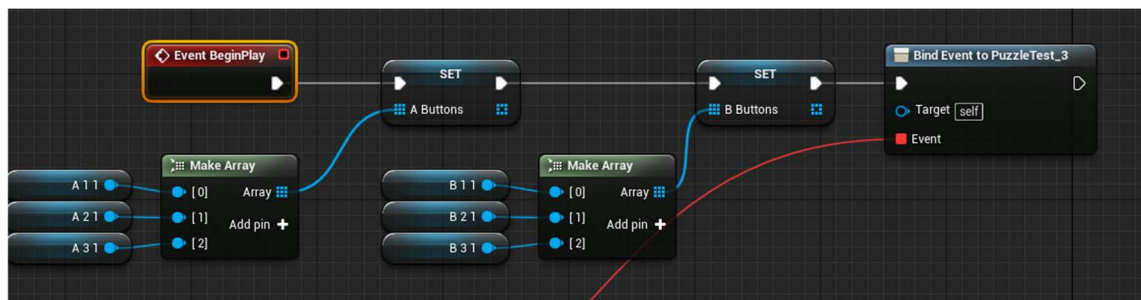
Kuvassa 19 esitetään FirstPersonPuzzlen pääkoodin loppuosa, mikä suoritetaan Switch And Check -funktion jälkeen. Kuvan 21 koodissa ensin tarkistetaan, onko ”Able To Press”-arvo False. Jos arvo on False, niin ruudun teksti muutetaan, ruudun ruudut hävitetään, odotetaan 2 sekuntia ja peli lopetetaan. Jos arvo on True, niin mitään ei tapahdu ja peli jatkuu.

6.3 Ongelmanratkaisunäytöt näppäimistö-hiiripelaajan pelitilassa



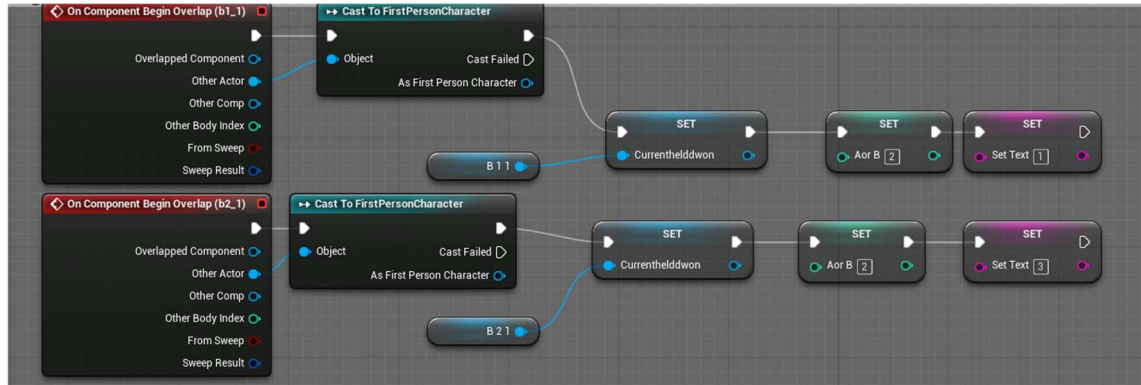
Kuva 20. ongelmanratkaisunäytön esikatselu.

Kuvassa 20 on esitetty yksi ongelmanratkaisunäyttö. Näppäimistö-hiiripelaajan pelitilassa on kolme näyttöä, jotka on johdettu samasta blueprintistä. Blueprintissä on 6 valkoista ja yksi musta Static Mesh kuutio. Kaikissa kuutioissa on myös osana tekstikomponentti.



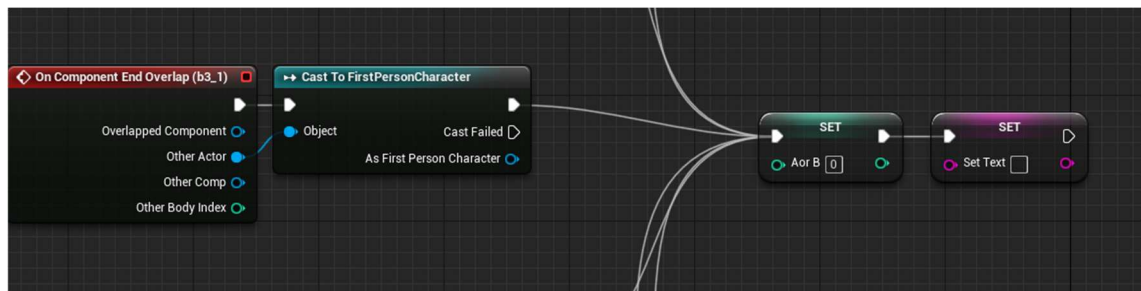
Kuva 21. Ongelmaratkaisunäytön alussa suoritettava ohjelma.

Kuvassa 21 havaitaan ongelmanratkaisunäytön alussa suoritettava koodi. Blueprint aloittaa tekemällä kaksi jonomuuttujaa. Toiseen muuttujaan tulee oikeanpuolimmaisiet näppäimet ja toiseen vasemmanpuoliset. Sitten tehdään muokattu tapahtuma, mikä suoritetaan, kun hiiren oikeaa painiketta klikataan.



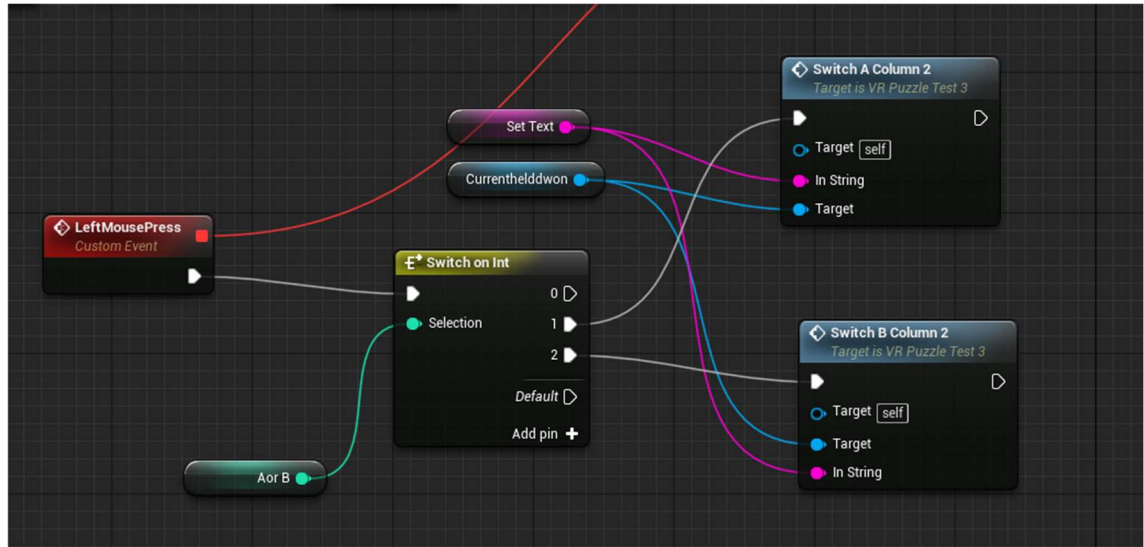
Kuva 22. Osoittimen kohdatessa näppäimen, arvot muuttuvat.

Pelaajan näytössä on hiiren osoittimen tapainen pallo, minkä tarkoitus on välittää viesti blueprintille, mitä nappia painetaan. Kun pallo osuu johonkin näppäimeen, niin se laittaa muuttujaan "Currently held down" kyseisen kuution arvon kuvan 22 mukaan. lisäksi se laittaa indeksiarvon 1 tai kaksi "A or B" muuttujaan. Lopuksi blueprint laittaa numeron tekstimuuttujaan "setText", mitä käytetään keskimmäisen näytön tekstin muuttamiseen. Jos osoitin menee pois näytöistä, niin arvot nollaantuvat kuten kuvassa 23 esitetään.

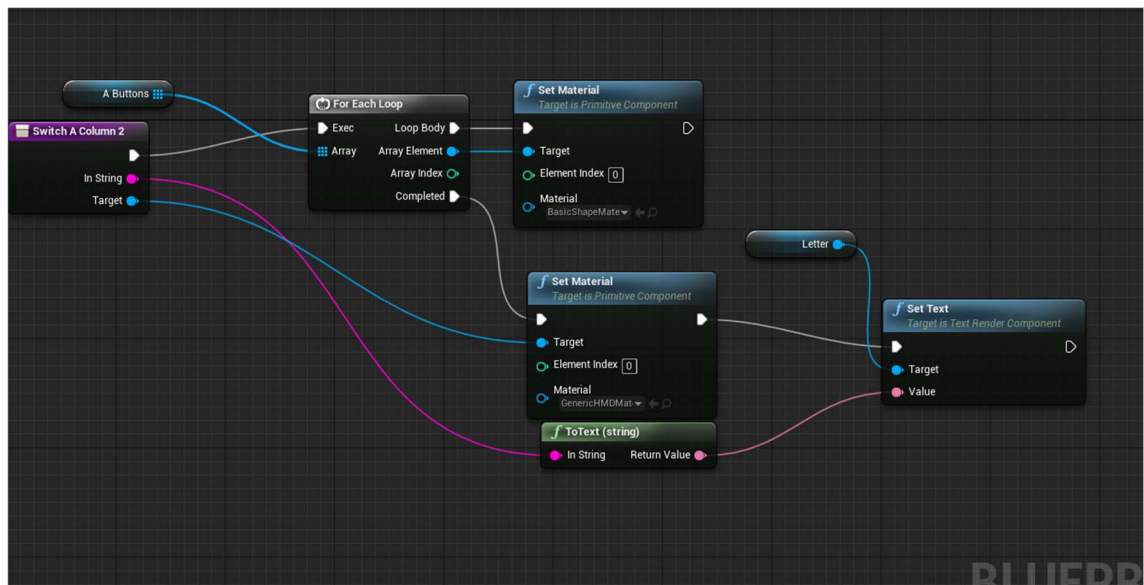


Kuva 23. Osoittimen poistuessa näppäimeltä, arvot nollaantuvat.

Kun arvot on saatu, eli pelaajan osoitin koskee jotakin näppäintä, niin pelaaja voi olla vaikutuksessa näytön kanssa. Jos pelaaja painaa oikeaa hiiren painiketta, niin blueprint katsoo, kumman puolen näppäimiä pelaaja painaa katsomalla "A or B" arvoa. Jos se on 1, niin pelaaja painaa vasemman puolen näppäimiä. Jos arvo on yksi, niin pelaaja painaa oikean puolen näppäimiä. Arvon mukaan blueprint suorittaa joko kuvassa 24 esitetyn funktion "Switch A column" tai "Switch B column".



Kuva 24. Vasemman hiiren painamisen jälkeen suoritettava ohjelma



Kuva 25. Funktio, millä vaikutetaan päänäytössä näytettäviin koordinaatteihin

Kuvassa 25 esitetään funktion "Switch A Column" koodia. Tätä koodia käytetään identtisesti myös "Switch B Column"-funktiossa. Molemmat funktiot ottavat arvot "Set text"-tekstin ja "Currently hold down"-static meshin. Molemmissa funktioissa muutetaan painetun näppäimen väri mustaksi ja muut valkoisiksi. Niin kuin edellisessä blueprintissäkin, musta väri tarkoittaa napin olevan painettuna. Sitten näytön keskellä oleva teksti vaihtuu "Set Text" muuttujassa olevaan tekstiin. Koska näyttö tarvitsee kaksioisaisen vastauksen, niin päänäytön molemmilla puolilla täytyy olla aktiivinen näppäin. Jotta vasemmalle puolelle ei tule esim.

kahta samaa näppäintä samanaikaisesti aktiiviseksi, niin funktiot ovat erikseen molempien puolien näppäinten aktivoimiseksi.

6.4 Näppäimistö-hiiripelaajan ohjelmointi

Molemmissa versioissa näppäimistö ja hiiri -pelaajan hahmo on ohjelmoitu hieman eri tavalla toisistaan. Varsinkin paikallisessa moninpeliversiossa pelaajahahmon ohjelmointi täytyy olla eri, jotta tietokoneen näytöllä näytetään vain yhden pelaajan näkymä.

6.4.1 Pelihahmon ohjelmointi paikallisen moninpelin versiossa

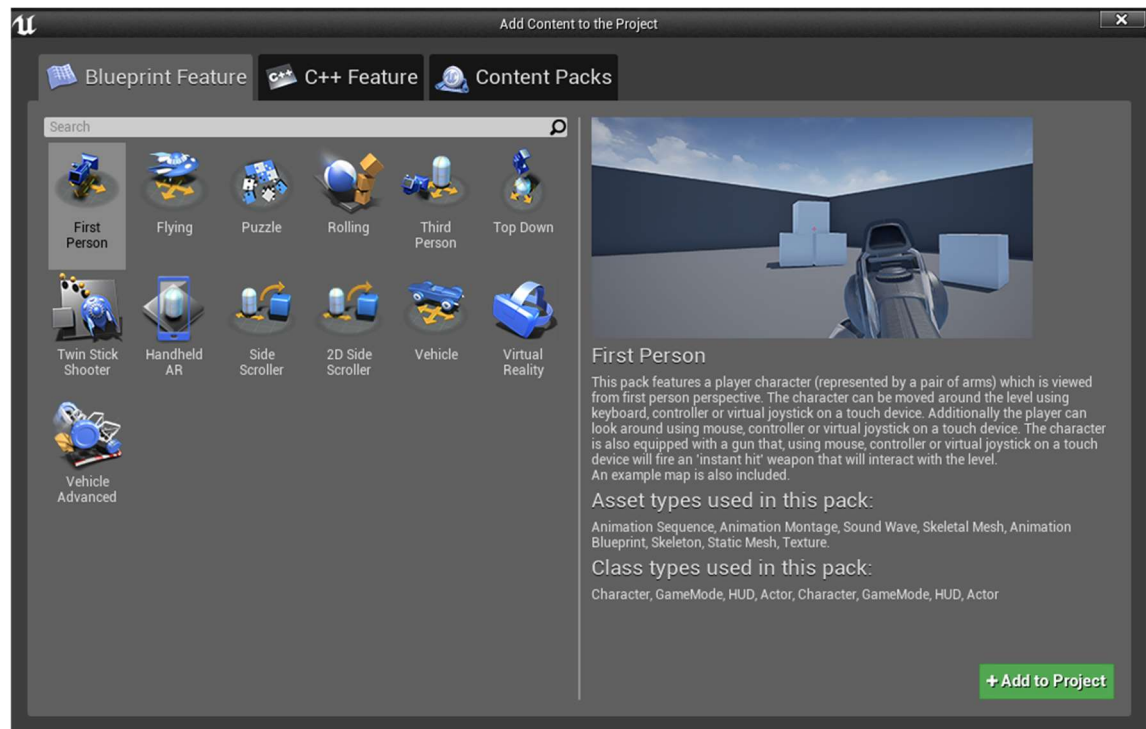
Paikallisessa moninpeli versiossa näppäimistö-hiiripelaajan blueprint koostuu SceneCaptureComponent2D-kamerakomponentista, Sphere-pallosta jossa on törmäyslaatikko ja kameran näköinen StaticMesh. SceneCapture2D komponentti on kaksiulotteinen nelikulmainen taso, mikä näyttää kuvaa tasolla jokaisella ruudulla pelin aikana [21]. Kuvaa otetaan jostakin määrätystä kamerakomponentista tai pelaajan edestä.

Blueprintin ohjelma alkaa alustamalla katsojan näkökentän. Tämä on ruutu, mitä muut pelaajat näkevät, kun VR-pelaaja pelaa peliä. Katsojan ruudulle näytetään tiettyä tekstuuria, joka on liitetty SceneCaptureComponent2D-komponenttiin. Tämä mahdollistaa katsojan näkymän olevan näppäimistö-hiiripelaajan pelihahmon näkökulma [22].

Pelin aikana pelihahmon jokaisella ruudulla katsotaan pelaajan liikkuminen. AddActorLocalOffset solmun avulla pelaaja pystyy liikkumaan nuolinäppäimillä tai ASDW-näppäimillä. Pelaaja voi katsoa ympärilleen AddActorWorldRotation solmun avulla, mikä ottaa hiiren x-koordinaatin, eli sivuttaissuunnan.

AddRelativeRotation solmulla pelaaja pystyy katsomaan ylös ja alas. Solmu ottaa hiiren Y-akselin [22].

6.4.2 Pelihahmon ohjelmointi verkkomoninpeliversiossa



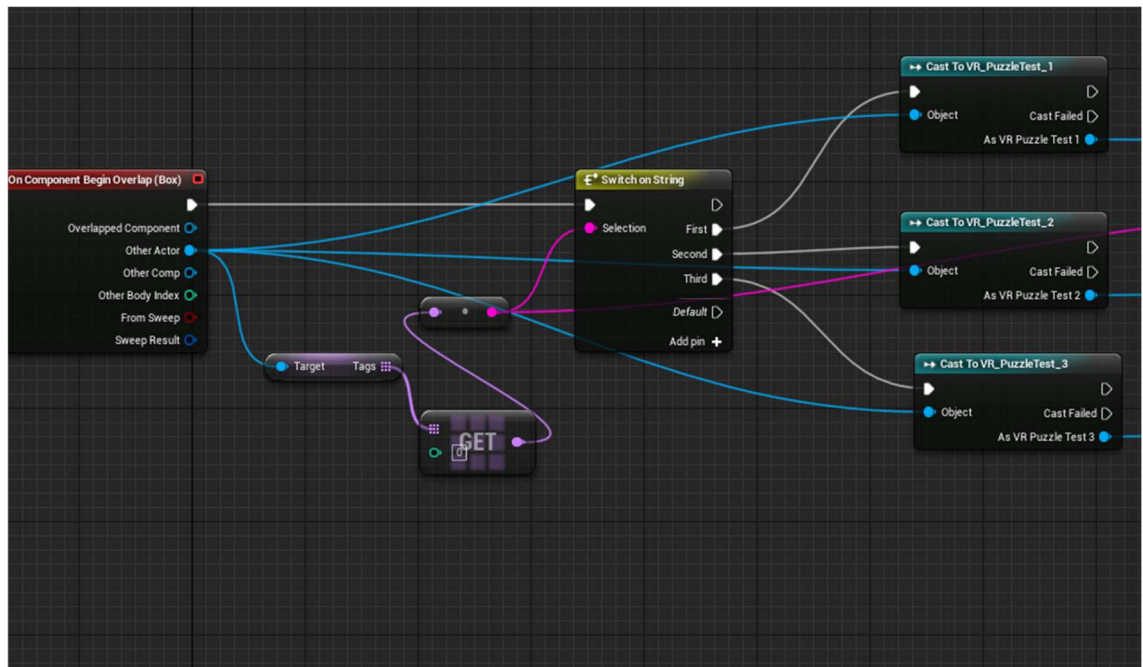
kuva 26. FirstPerson esimerkkiprojektin tuominen

Verkkomoninpeliversiossa näppäimistö-hiiripelaajan pelihahmo on ohjelmoitu hieman eri tavalla. Tässä versiossa on käytetty FirstPersonTemplate-esimerkkiprojektin valmista pelihahmoa. Pelihahmosta on poistettu ase ja siihen on lisätty sama Sphere-pallo ja sen Box törmäystarkastelulaatikko. Kaikki pelaajan liikkumiseen liittyvä ohjelmointi on jo valmiiksi tehtynä, joten lisäohjelmointia ei tarvita. Kuvassa 26. nähdään FirstPersonTemplate-esimerkkiprojektin tuomiseen liittyvä ikkuna.

6.4.3 Molempien versioiden yhteinen ohjelmointiosa

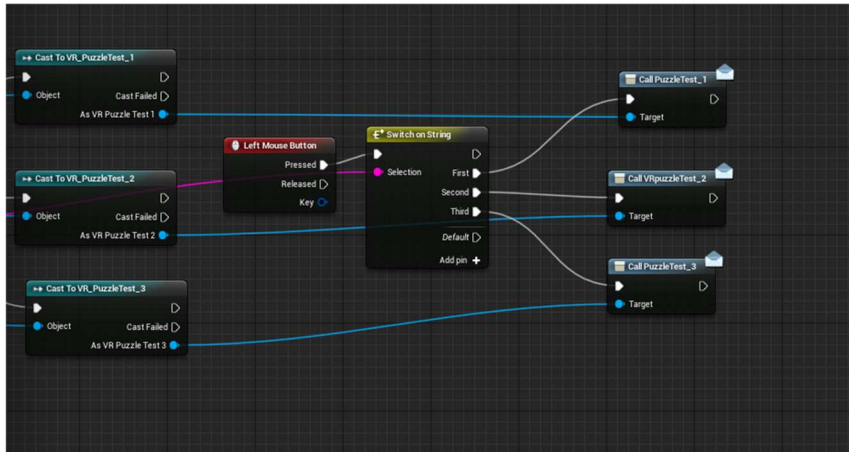
Molempien versioiden näppäimistö-hiiripelaajan pelihahmon blueprintissä on ohjelmoitu sama osuus, mikä liittyy näppäimistö-hiiripelaajan pelitilan ongelmanratkaisunäyttöihin. Tämän ohjelmointiosan tarkoituksena on välittää viesti tietylle ongelmanratkaisunäytölle siitä, että pelaajan edessä oleva pallo osuu näytöllä oleviin painikkeisiin.

Blueprintin ohjelma alkaa sillä, että pelaajan edessä olevan pallon törmäystarkastelu katsoo, mihin esineisiin se osuu. Jos se osuu johonkin esineeseen, niin se katsoo, mikä Tag-nimike on liitetty esineeseen. Tag-nimikkeet ovat tunnisteita esineille, mitä voi määrittää. First, Second ja Third tunnisteet ovat laitettu jokaiseen ongelmanratkaisunäyttöön: First on laitettu ensimmäiseen ongelmanratkaisunäyttöön, Second on laitettu toiseen ja Third kolmanteen. Jos Törmäyksessä esineen tunniste on First, Second tai Third niin ohjelma tekee tyyppimuunoksen. Tämän avulla ohjelma pystyy tekemään muutoksia muihin blueprintteihin.



Kuva 27. Näppäimistö-hiiripelaajan törmäystarkastelun koodi

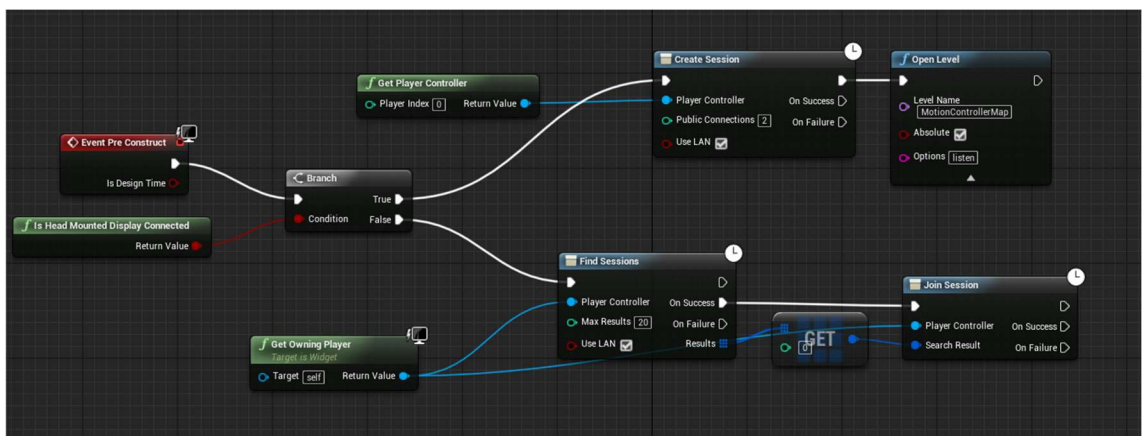
Seuraava osa aktivoituu, kun pelaaja painaa vasenta hiiren painiketta. Ohjelma tarkistaa, onko pallo vielä jonkun napin päällä ja lähettää viestin sen ongelmanratkaisunäytön blueprintille, mitä pelaajan hiiri osoittaa.



Kuva 28. Törmäystarkastelun koodi, kun hiiren vasenta painiketta painetaan

6.5 Verkkomonipeliversiön tietokoneiden yhdistäminen

Verkkomonipelissä molemmat tietokoneet ovat yhdistettynä LAN-kaapelilla, mutta pelin täytyy tietää, mitkä tietokoneet pitää yhdistää. Verkkomonipeliversiossa peli alkaa eri tasosta, missä kaikki toiseen tietokoneeseen yhdistämisen ohjelmointi tapahtuu. Tasossa on liitetty tyhjä UI-elementti, mikä tekee tietokoneiden yhdistämisen.

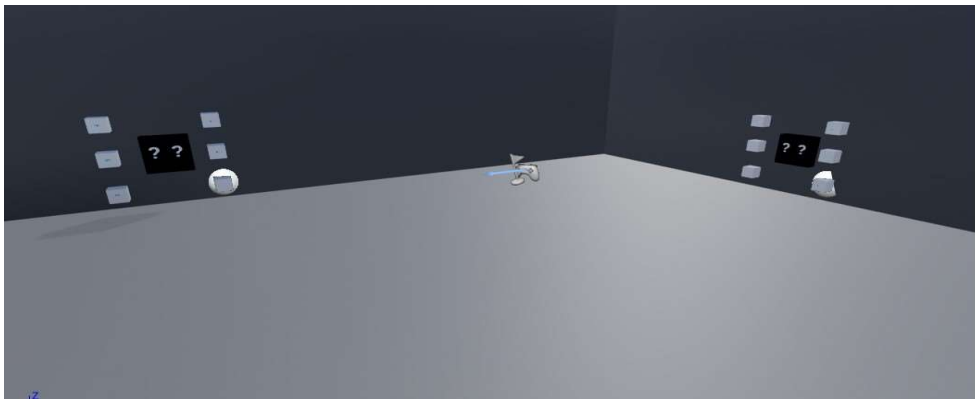


Kuva 29. Pelin yhdistäminen

Kun taso latautuu niin blueprint katsoo, onko VR-lasit yhdistettynä. Jos ne ovat, niin kyseinen tietokone luo peli-istunnon ja lataa pelin varsinaisen pelitason. VR-pelaaja toisinsanoen isännöi peli-istuntoa, mihin näppäimistö-hiiripelaaja sitten liittyy.

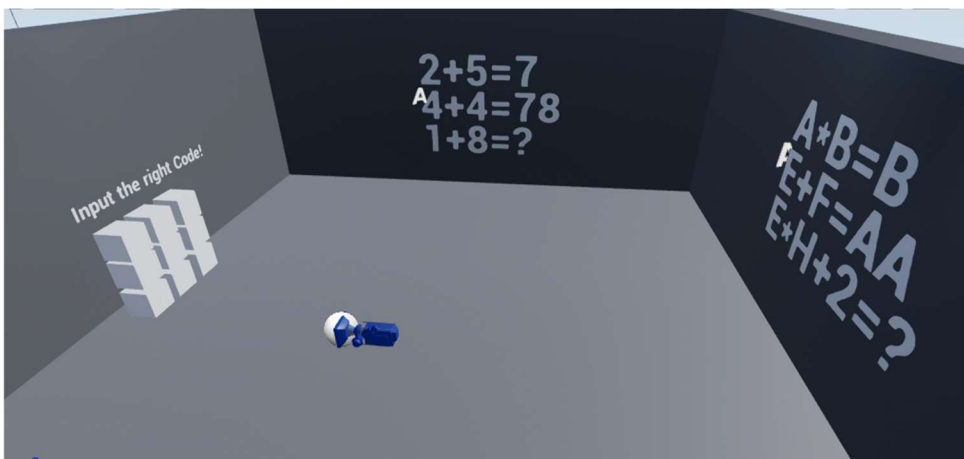
Jos pelaajan tietokoneeseen ei ole yhdistetty VR-lasit, niin peli pyrkii etsimään käynnissä olevia istuntoja. Kun blueprint löytää ensimmäisen käynnissä olevan istunnon niin pelaaja liittyy siihen näppäimistö-hiiripelaajana. Tämän jälkeen peliä voi pelata normaalisti.

6.6 Asymmetrisen pelin lopullinen pelitila



Kuva 30. Näppäimistö-hiiripelaajan viimeistelty pelialue.

Kuvassa 30 nähdään viimeistelty pelialue, missä näppäimistö-hiiripelaaja pelaa. Pelaaja voi liikuttaa pelihahmoa näppäimistön A, S, D, W näppäimillä ja katsoa ympärillensä hiirtä liikuttamalla. Painamalla hiiren vasenta painiketta ongelmienratkaisuruutujen lähellä, niin pelaaja voi valita ruudun. Pelaajan täytyy olla tarpeeksi lähellä, jotta pelaajan edessä oleva osoitin osuu haluttuun ruutuun.



Kuva 31. Virtuaalitodellisuuslaseilla pelaavan pelaajan viimeistelty pelialue

Kuvassa 31 nähdään lopullinen pelialue, missä VR pelaaja pelaa. Alue on harmaa ja pelkistetty. Pelaaja pystyy liikkumaan alueella painamalla ohjaimen kosketuslevyä. Kun kosketuslevyä painaa, ilmesty sininen kaari, minkä loppupiste määrää, mihin liikutaan. Kun kosketuslevy vapautetaan, niin pelaaja liikkuu määrättyyn kohteeseen. VR-pelaaja pystyy vaihtamaan ruudukossa olevan ruudun väriä painamalla ohjaimen liipaisinta.

7 Kehitettyjen pelien vertailu

Opinnäytetyötä varten mitattiin molempien pelien tarvitsemaa aikaa piirtää yksi ruutu pelaajan näytölle ja arvoja vertailtiin toisiinsa. Tutkimuksen tavoitteena on selvittää, kumpi versio tarvitsee tietokoneen näytönohjaimelta ja suorittimelta vähiten aikaa näyttääkseen yhden ruudun pelaajan näytössä. Mitä vähempi luku, niin sen tehokkaampi peli on ja ei vaadi paljoa aikaa tietokoneelta suorittaakseen peliä.

7.1 Työkalut

Testaukseen varten kummatkin versiot on rakennettu julkaisukelpoisiksi. Pelin ruudun piirtämiseen tarvitsemaa aikaa mitattiin Unreal Engine 4-pelimoottorin sisäisillä stat fps ja stat unit- komennoilla. Stat fps mittaa, kuinka monta ruutua sekunnissa näytölle päivitetään. Mitä suurempi luku, sen parempi. Stat unit komento antaa tarkempia mittauksia pelin tarvitsemasta ajasta näyttää yksi ruutu pelaajan näytöllä. Mitä suuremmat arvot, sen pidempään pelaajan tietokone tarvitsee aikaa suorittaakseen peliä.

Stat unit antaa eri metriikoita, jotka mittaavat tietokoneen näytönohjaimen ja suorittimen tarvitsemaa aikaa näyttää pelin ruutua pelaajan näytöllä millisekunneissa. Olli Mäntylän diplomityössä ”Suorituskyky Ja Unreal Engine 4” selitetään arvot seuraavalla tavalla:

"Arvoista "Frame" tarkoittaa yhden kuvan piirtämiseen kulunutta aikaa. "Game" tarkoittaa CPU:n suorittaman pelisäikeen vaatimaa aikaa (CPU Game Thread), "Draw" CPU:n suorittaman piirtosäikeen aikaa (CPU Render Thread) ja "GPU" nimensä mukaisesti näytönohjaimen käyttämää aikaa. Näiden lukujen pohjalta voidaan arvioida pullonkaulana toimivaa suoritusyksikköä senhetkisessä tilanteessa." [23.]

CPU on lyhenne Central Processing Unit:sta tarkoittaa tietokoneen prosessoria.

7.2 Testitapaukset

Testitapauksia on kolme. Yksi testitapaus suoritetaan yhdellä tietokoneella (tietokone a) ja kaksi suoritetaan kahdella tietokoneella (tietokone a ja tietokone b). Testitapausten tulokset otetaan tietokone a:sta. Testauksessa ohjaimina käytetään näppäimistö ja hiiri yhdistelmää ja HTC-Vive-virtuaalitodellisuuslaseja.

Ensimmäisessä testitapauksessa testataan paikallisen moninpelin versiota. Tässä testitapauksessa molemmat ohjaimet ovat yhdistettynä tietokone a:n. Pelin käynnistyttyä otamme arvot tietokone a:n ruudulta

Toisessa testitapauksessa testataan lähiverkkomoninpeli-versiota. Tässä testitapauksessa ohjaimet on yhdistetty eri tietokoneisiin. Molemmat tietokoneet ovat yhdistettynä samaan lähiverkkoon. Tietokoneeseen a on yhdistetty näppäimistö ja hiiri. Tietokoneeseen b on yhdistetty virtuaalitodellisuuslasit. Tässä testitapauksessa mittaustulokset otetaan näppäimistö-hiiripelaajan tietokoneesta.

Kolmas testitapaus on samanlainen kuin toinen testitapaus. Tässä testitapauksessa ohjaimet yhdistetään toisin päin. Näppäimistö ja hiiri yhdistetään tietokoneeseen b ja virtuaalitodellisuuslasit yhdistetään tietokoneeseen a. Tässä testitapauksessa mittaustulokset otetaan virtuaalitodellisuuslaseilla pelaavan pelaajan tietokoneesta.

7.3 Testauksen hypoteesi

Testauksen tavoitteena on löytää tapa toteuttaa moninpeli asymmetrisesti näppäimistön ja hiiren yhdistelmän ja virtuaaliodellisuuslasien välillä, joka ei vie tietokoneen näytönohjaimelta ja suorittimelta paljoa aikaa näyttääkseen yhden ruudun näytöllä. Esitän hypoteesiksi vähiten suorittimelta ja näytönohjaimelta aikaa ruudun näyttämiseen vieväksi peliksi lähiverkkomoninpeli-versiota. Oletan, että jos molemmat ohjaimet ovat yhdistettynä samaan tietokoneeseen, niin tietokoneen suoritin ja näytönohjain tarvitsee enemmän aikaa suorittaakseen peliä.

Lähiverkkomoninpeli-versiossa molemmat tietokoneet suorittavat vain oman ohjaimen näyttöä, joten oletan sen olevan tehokkaampi. Toisin sanoen, lähiverkkomoninpeli-versiossa pelin suoritustaakka jaetaan kahden koneen välille.

7.4 Testitulokset

Ensimmäisessä testitapauksessa testattiin paikallisen moninpelin versiota.



kuva 32. Ensimmäisen testitapauksen testitulokset.

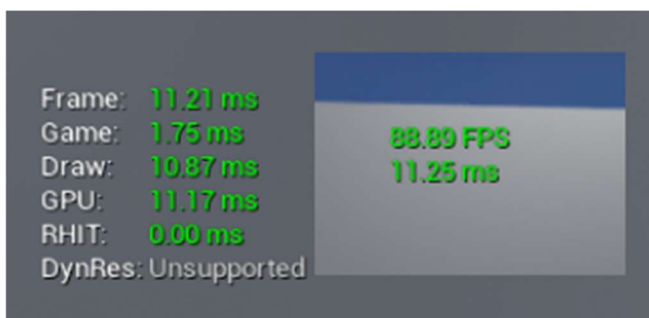
Kuvasta 32 näkyy, että Draw-aika on lähellä Frame aikaa, mikä johtuu pelin renderöimisen ongelmista. Näytönohjain tarvitsee myös saman verran aikaa renderöidäkseen pelin ruudun näytölle. Toisaalta FPS pysyttelee noin 90 FPS:n pinnalla, mikä on hyvä suoritus. Arvot ovat vihreällä alueella, joten pelin suorituskyky on hyvä.

Toisessa ja kolmannessa testitapauksessa testattiin lähiverkkomoninpeli-versiota. Molemmissa testeissä on käytössä 2 tietokonetta, mitkä ovat näytönohjaimeltaan ja suorittimeltaan erilaisia. Testitulokset otettiin samasta tietokoneesta, missä ensimmäinenkin testitapaus suoritettiin. Tämä antaa tarkan tuloksen, mitä voidaan verrata muihin testaustuloksiin. Toisessa testitapauksessa näppäimistö-hiiripelaaja pelaa testaukseen tarkoitetulla tietokoneella ja kolmannessa testitapauksessa VR-pelaaja pelaa testaukseen tarkoitetulla tietokoneella.



kuva 33. Toisen testitapauksen tulokset.

Kuvan 33 mukaan Game ja Draw arvot pysyvät erittäin matalina. FPS pysyy n.60 FPS:n tavoitearvossa. Peli suoriutuu arvojensa mukaan erittäin hyvin.



Kuva 34. Kolmannen testitapauksen tulokset.

Kuvasta 34 näemme, että kolmannen testitapauksen tulokset ovat erittäin samanlaisia ensimmäisen testitapauksen kanssa.

7.5 Testitulosten vertailu

Testauksen mukaan saamme taulukon 1 mukaisia tuloksia:

	FPS	Frame	Game	Draw	GPU
Paikallinen Moninpeli	89,73	11.16 ms	1.57 ms	10.81 ms	11.16 ms
Lähiverkkomoninpeli (näppäimistö ja hiiri)	62	16.13 ms	0.90 ms	2.21 ms	16.11 ms
Lähiverkkomoninpeli (VR)	88,89	11.21 ms	1.75 ms	10.87 ms	11.17

Taulukko 1. Testitulokset

Taulukko 1:stä voimme havaita, että lähiverkkomoninpeli-versiossa näppäimistö-hiiripelaajan on huonoin ruudunpäivitys sekunnissa ja ruutujen piirtämiseen ja renderöimiseen tarvitaan eniten aikaa. Kaikissa testitapauksissa näytönohjain tarvitsee eniten aikaa renderöidäkseen yhden ruudun.

7.6 Optimointi

Testaustuloksien mukaan lähdin selvittämään, miksi lähiverkkomonipelissä näppäimistö-hiiripelaajan FPS on matala verrattuna muihin testitapausten FPS arvoihin. Unreal Engine 4:n dokumentaatiossa lukee, että maksimi FPS määräytyy näytön päivitystiheyden mukaan, jos "Smooth Frame Rate" on päällä. Testauksen aikana, virtuaalitodellisuuslasien näytön päivitysnopeus oli 90 Hz ja päämonitorin 60 Hz [24].

Tämän tiedon mukaan, lähdin testaamaan vielä kerran toista testitapausta, mutta ottamalla "Smooth Frame Rate" valinnan pois käytöstä. Nostin myös ruudunpäivitysrajausta 90 FPS:in, jotta testitulosta voisi luotettavasti verrata muihin testituloksiin.



Kuva 35. Muokattu testitapaus 2.

Uusien testitulosten myötä saamme taulukon 2 mukaisia tuloksia:

Testattava versio	FPS	Frame	Game	Draw	GPU
Paikallinen moninpeli	89,73	11.16 ms	1.57 ms	10.81 ms	11.16 ms
Lähiverkkomoninpeli (Näppäimistö ja hiiri)	89.99	11.11 ms	1.07 ms	1.97 ms	11.10 ms
Lähiverkkomoninpeli (VR)	88,89	11.21 ms	1.75 ms	10.87 ms	11.17 ms

Taulukko 2. Ylimääräisen testauksen tulokset

Taulukko 2:en tuloksista voimme havaita, että Frame- ja GPU-arvo muuttuu maksimi FPS:n mukaan. Kaikissa testeissä näytönohjain tarvitsee edelleen suurimman ajan renderöidäkseen yhden ruudun pelaajan näytölle.

8 Testauksen yhteenveto

Testituloksista voidaan havaita, että kummatkin versiot tarvitsevat eniten aikaa näytönohjaimelta renderöidäkseen yhden ruudun pelaajalle. Molemmissa versioissa myös virtuaalitodellisuuslaseja käyttävä pelaajan tietokoneen prosessorin piirtosäie tarvitsee eniten aikaa ruudun piirtämisessä. Kaikissa testitapauksissa prosessorin pelisäie ei tarvitse paljoa aikaa.

Alustavan hypoteesin näkökulmasta vähiten näytönohjaimelta ja suorittimelta ruudun piirtämiseen menevä aika olisi ollut paikallisen moninpelin versiossa. Testien jälkeen hypoteesini on ollut väärä. Vähiten suorittimelta ja näytönohjaimelta ruudun piirtämiseen meni aikaa lähiverkkomoninpeli-versiossa, mutta vain näppäimistö-hiiripelaajalla. Toisaalta kyseinen versio tarvitsee kaksi tietokonetta. Virtuaalitodellisuuslaseilla pelaavan pelaajan tietokoneen näytönohjain ja suoritin vaatii enemmän aikaa näyttääkseen peliä ja vertailussa vie enemmän aikaa kuin näppäimistö-hiiripelaajan tietokone.

Testitulosten kannalta olisi parasta kuitenkin käyttää paikallismoninpeli-versiota. Vaikka tietokoneen suoritin ja näytönohjain tarvitsee enemmän aikaa näyttääkseen pelin ruutuja ruudulla, käyttäjän ei tarvitse hankkia toista tietokonetta. Lähiverkkomoninpeli-versiota voi käyttää, jos käyttäjä omistaa yhden tietokoneen, jonka suoritin ja näytönohjain eivät vaadi pelin ruudun piirtämiseen paljon aikaa ja toisen, mikä tarvitsee. Lähiverkkomoninpeli-version käyttöönottoa voi myös harkita, jos tietokoneet halutaan pitää kaukana toisistaan tai jos virtuaalitodellisuuslasien vaatimat USB ja HDMI-portit puuttuvat yhdestä tietokoneesta.

9 Pohdinta

Opinnäytetyössäni pyrin vastaamaan kahteen tutkimuskysymykseen: mitä pitää ottaa huomioon asymmetrisen VR moninpelin kehityksessä teknisen toteutuksen näkökulmasta ja mitä hyötyjä ja haittoja on paikallisen- ja verkkomoninpelin kehityksessä?

Asymmetrisen pelin kehityksessä pitäisi ensisijaisesti ottaa huomioon, mitä tekniikkaa hyödyntää pelaajien yhdistämiseen pelissä: pelataanko peliä vain yhdellä koneella vai yhdistetäänkö kaksi tai useampi tietokone toisiinsa jollakin tavalla. Opinnäytetyön aikana tehdyn tutkimuksen tulosten perusteella asymmetrisen moninpeli olisi parempi tehdä paikallisena monipelina yhdelle tietokoneelle. Paikallisen moninpelin suurin hyöty on yhden tietokoneen tarve pelin pelaamista varten. Päällimmäinen haitta paikalliselle monipeliversiolle on sen tarve eri USB- ja HDMI-porteille. Tämän opinnäytetyöhön kehitetyssä pelissä

paikallinen moninpeliversio tarvitsi kolme eri USB-porttia ja yhden HDMI-portin. Lähiverkkomoninpelin hyötyjä on tietokoneen työtaakan väheneminen näppäimistö-hiiripelaajalle ja tietokoneen eri sisääntuloporttien määrän vaatimus per tietokone. Päällimmäinen haitta on kahden tietokoneen tarve pelaamista varten.

Toinen huomioon otettava asia asymmetrisen pelin kehityksessä on pelitilan rakentaminen, varsinkin sellaisissa asymmetrisissä peleissä, missä käytetään virtuaalitodellisuuslaseja. Pelitila tulisi rakentaa siten, että pelaajien pelitilat ovat erillään toisistansa. Virtuaalitodellisuuslaseja käyttävä pelaaja pystyy mahdollisesti huijaamaan viemällä virtuaalitodellisuuslasit pelissä olevan seinän läpi. Jos tilat olisivat olleet vierekkäin, niin VR-pelaaja pystyy näkemään toisen pelaajan pelitilan. Tämän vuoksi pyrin omassa projektissani laittamaan välitilan pelaajien pelitilojen välille.

Kolmas huomioon otettava asia asymmetrisen VR pelin kehitykseen on virtuaalitodellisuuslasien valinta projektille. VR-lasit ovat erilaisia ohjaimiltaan ja peli ei toimi kaikilla VR-laseilla ilman ylimääräistä ohjelmointia. Opinnäytetyössä kehitetyssä pelissä peli on tehty HTC Vive-virtuaalitodellisuuslaseille. Peliä on kokeiltu myös HTC Vive Pro-virtuaalitodellisuuslaseilla, mutta peli ei toiminut kunnolla. Esim. VR-pelaaja ei pystynyt katsomaan ympärille ja liikkuminen tilassa ei onnistunut.

Neljäs huomioon otettava asia asymmetrisen pelin kehityksessä on pelin selkeys pelaajille. Opinnäytetyöni aikana toimeksiantajani kokeili lopullista peliä ja antoi arvokasta palautetta ja jotakin pohdittavaa. Peli itsessään oli erityisen alkukantainen ja pelkistetty. Peli oli hieman epäselvä pelaajalle ja jouduin selittämään pelin tavoitteen pelaajille. Pelin olisi voinut tehdä selkeämmin, esim. VR-pelitulassa olevaan ruudukkoon olisi voinut laittaa koordinaattien kirjaimet ja numerot helpottaakseen pelaajaa laittamaan koodin. Värejäkin olisi voinut käyttää pelissä tuodakseen viihtyvyyttä. Peliä myös moitittiin, koska siinä ei ollut mitään uudelleenpelattavuusarvoa. Pelissä on aina samat ongelmat ja samat ratkaisut, joten peliä voi pelata vain kerran.

9.1 Opinnäytetyötä varten kehitetyn pelin vertailu muihin asymmetrisiin VR peleihin

Verrattuna kappaleen 1.5 esimerkkipeleihin pelini oli erittäin yksinkertainen. Keep Talking and Nobody Explodes-peli on uudelleenpelattavuus-arvoltaan erittäin hyvä peli verrattuna opinnäytetyöhön kehitetystä pelistä. Pommit ovat aina erilaisia, mikä tuo peliin uudelleenpelattavuusarvoa. Peliä voi pelata yhdellä tietokoneella VR-laseja tai peliohjainta käyttäen. Toinen pelaaja voi vaan tulostaa ohjeet pommin purkuun ja auttaa siten. Opinnäytetyöhön kehitetyssä pelissä vaaditaan näppäimistö, hiiri ja VR-lasit.

Verrattuna toiseen esimerkkipeleihin; Nemesis Perspective-peliin, pelini oli visuaalisesti pelkistetty. Nemesis Perspective-pelissä uudelleenpelattavuusarvo on pieni, mutta visuaalisesti peli on hyvännäköinen. Pelin ihmishahmoa voi myös pelata joko peliohjaimella tai näppäimistöllä ja hiirellä. Pelissä oli myös selkeästi eroteltu molempien pelaajien kyvyt ja taidot pelissä. Opinnäytetyöhön kehitetyssä pelissä pelaajien ominaisuudet ovat melkein pä samanlaiset: molemmat pelaajat voivat liikkua vapaasti ja painaa omassa pelitilassa olevia näppäimiä.

9.2 Itsearviointi

Lähiverkkomoninpelin tietokoneiden yhdistämiseen liittyvä koodi on omasta mielestäni hyvin alkeellinen. Lähiverkkomoninpeli-versiossa näppäimistö-hiiripelaaja etsii ensimmäisen VR-pelaajan peli istunnon. Tämä voi tuottaa ongelmia, mikäli yhdistettyjä tietokoneita on useampi kuin kaksi ja näppäimistö-hiiripelaaja haluaa yhdistää tiettyyn VR-pelaajan peli istuntoon tai useampi näppäimistö-hiiripelaaja yhdistetään samaan istuntoon. Tämän voisi korjata tekemällä pelin alkuun listan kaikista löydetyistä peli istunnoista ja valitsemalla halutun peli istunnon, mihin toinen pelaaja haluaa yhdistää ja ohjelmoimalla rajoituksen pelaajien määrästä tietyssä peli istunnossa.

Mielestäni tutkimusta olisi voinut laajentaa tekemällä lähiverkkomoninpeli-versiosta palvelimessa toimiva versio. Nykyisessä versiossa pelaajat yhdistetään toisiinsa tietokoneisiin yhdistetyn LAN-kaapelin avulla. Palvelinta käyttävässä versiossa pelaajat yhdistävät tietokoneet toisiinsa ulkopuolisen palvelimen kautta. Testiä en voinut kuitenkaan tehdä tietotaidon puutteen vuoksi ja ulkoisen palvelimen saatavuuden puutteen vuoksi. Palvelinta käyttävä versio mahdollistaa pelaajien pelaavan erittäin kaukana toisistaan, mutta peliin täytyy kehittää jonkinlainen kommunikaatiokeino tai pelaajia täytyy kehoittaa käyttämään ulkoista keskustelupalvelua kuten Discord.

Pelin konsepti oli mielestäni huono ja tylsä pelattavuudeltaan, mutta sen asymmetrisyyden toteuttaminen oli mielestäni hyvin toteutettu. Toimeksiantajani oli myös kehunut pelin asymmetrisyyden toteuttamista ja ehdotti ominaisuuden kehittämistä johonkin toiseen projektiin. Tämän opinnäytetyön tavoite oli kuitenkin kehittää asymmetrinen VR-moninpeli, mitä voi pelata yhdellä tai kahdella koneella ja mielestäni olen siinä onnistunutkin.

Lähteet

1. Ward, J. 2008. What is a Game Engine?
https://www.gamecareerguide.com/features/529/what_is_a_game_.php. 11.9.2019
2. Evangelho, J. 2014. Why Is Epic Games Promoting Unreal Engine 4 With A 'Flappy Bird' Clone? <https://www.forbes.com/sites/jasonevangelho/2014/05/22/why-is-epic-games-promoting-unreal-engine-4-with-a-flappy-bird-clone/#467fa691acce>. 11.9.2019.
3. Unreal Engine. Levels. <https://docs.unrealengine.com/en-US/Engine/Levels/index.html>. 28.10.2019
4. Unreal Engine. 2015. Intro to Blueprints: Blueprint Introduction | 01 | v4.8 Tutorial Series | Unreal Engine. https://www.youtube.com/watch?v=EFXMW_UEDco. 28.10.2019
5. Unreal Engine. Blueprint Class. <https://docs.unrealengine.com/en-US/Engine/Blueprints/UserGuide/Types/ClassBlueprint/index.html>. 28.10.2019.
6. Strickland, J. How Virtual Reality Works. <https://electronics.howstuffworks.com/gadgets/other-gadgets/virtual-reality9.htm>. 6.9.2019
7. The Ultimate Guide to Understanding Augmented Reality (AR) Technology. Reality Technologies. <https://www.realitytechnologies.com/augmented-reality/>. 28.10.2019.
8. The Ultimate Guide to Understanding Mixed Reality (MR) Technology. Reality Technologies. <https://www.realitytechnologies.com/mixed-reality/>. 28.10.2019.
9. What really is the difference between AR / MR / VR / XR? Medium. <https://medium.com/@northof41/what-really-is-the-difference-between-ar-mr-vr-xr-35bed1da1a4e>. 28.10.2019
10. History Of Virtual Reality. Virtual Reality Society. <https://www.vrs.org.uk/virtual-reality/history.html>. 11.9.2019.
11. Dormehl, L. 2019. 8 virtual reality milestones that took it from sci-fi to your living room. <https://www.digitaltrends.com/cool-tech/history-of-virtual-reality/>. 11.9.2019.
12. Virtual Reality in Gaming. ThinkMobiles. <https://thinkmobiles.com/blog/virtual-reality-gaming/>. 28.10.2019.
13. Wikipedia. Symmetria. <https://fi.wikipedia.org/wiki/Symmetria>. 28.10.2019
14. Snyder, J. 2015. Asymmetrical Gameplay: Gimmick or Revolution? <http://www.theoryofgaming.com/asymmetrical-gameplay-gimmick-or-revolution/>. 4.9.2019.
15. Steam. Keep Talking And Nobody Explodes. https://store.steampowered.com/app/341800/Keep_Talking_and_Nobody_Explodes/?snr=1_7_15__13. 28.10.2019.
16. Wikipedia. Keep Talking And Nobody Explodes. https://en.wikipedia.org/wiki/Keep_Talking_and_Nobody_Explodes. 28.10.2019.
17. Graft, K. 2016. Road to the IGF: Steel Crate Games' Keep Talking and Nobody Explodes. https://www.gamasutra.com/view/news/264317/Road_to_IGF_Steel_Crate_Games_Keep_Talking_and_Nobody_Explodes.php. 28.10.2019.
18. Steam. Nemesis Perspective. https://store.steampowered.com/app/537140/Nemesis_Perspective/. 28.10.2019.
19. James The Maniac. 2017. VERSUS VR! - Nemesis Perspective. <https://www.youtube.com/watch?v=SmtWhUZKu5k>. 28.10.2019.
20. Wadstein, M. 2015. WTF Is? Volume - Nav Mesh Bounds in Unreal Engine 4. <https://www.youtube.com/watch?v=hE7aMBeT53o>. 28.10.2019.

21. Unreal Engine. 1.7 - Scene Capture 2D. https://docs.unrealengine.com/en-US/Resources/ContentExamples/Reflections/1_7/index.html. 28.10.2019.
22. Unreal Engine. 2019. Creating a VR spectator camera | Unreal Engine Livestream. <https://www.youtube.com/watch?v=0up1Lgmw-Lg>. 28.10.2019.
23. Mäntylä, O. 2016. Suorituskyky Ja Unreal Engine 4. Tampereen teknillinen yliopisto. Tietotekniikan koulutusohjelma. Diplomityö <https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/24429/Mantyla.pdf?sequence=1>. 4.9.2019.
24. Unreal Engine. Smooth Frame Rate. <https://www.unrealengine.com/en-US/blog/how-to-improve-game-thread-cpu-performance>. 1.9.2019.