

Virtuaalitodellisuussovelluksen kehitys Unity- pelimoottorilla

**Case: Konenäkö-hankkeen kuntouttava virtuaalitodellisuussovel-
lus**

LAB-ammattikorkeakoulu

Insinööri (AMK), Tieto- ja Viestintätekniikka (Ohjelmistotekniikka)

2021

Jonna Tuomisto

Tiivistelmä

Tekijä(t) Tuomisto, Jonna	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika 2021
	Sivumäärä 29	
Työn nimi Virtuaalitodellisuussovelluksen kehitys Unity-pelimoottorilla Case Konenäkö-hankkeen kuntouttava virtuaalitodellisuussovellus		
Tutkinto Insinööri (AMK)		
Toimeksiantajan nimi, titteli ja organisaatio Anna Lahti, Konenäkö-projektipäällikkö, LAB-Ammattikorkeakoulu		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli luoda kuntouttava virtuaalitodellisuussovellus. Työn toimeksiantajana toimi Konenäkö-hanke.</p> <p>Sovellus kehitettiin käyttäen Unity-pelimoottoria sekä langatonta Oculus Quest -virtuaalitodellisuuslaitetta. Unity-pelimoottori on joustava IDE-ohjelmisto. Opinnäytetyössä käsitellään Unityn käyttöä ja konsepteja, kuten peliobjekteja, scenejä ja skriptejä. Työssä käytettiin Oculus Integration Package -pakettia, joka on luotu Oculus-sovelluksien kehitykseen Unitylla. Lisäksi tutustuttiin virtuaalitodellisuuden historiaan sekä virtuaalitodellisuuslaitteistojen eroihin. Työssä myös käydään läpi VR:n ympärillä käytettyjä termejä.</p> <p>Työn tuloksena luotiin toimiva virtuaalitodellisuussovellus. Sovellusta testattiin kehityksen aikana, ja käyttökokemuksista saatiin kattavasti palautetta sovelluksen kehitykseen. Sovelluksesta jäi uupumaan joitakin suunniteltuja toiminnallisuuksia projektin loppumisen vuoksi. Sovellusta on mahdollista laajalti jatkokehittää lisäämällä puuttuvia ominaisuuksia sekä parantelemalla valmistuneita aktiviteettejä.</p>		
Asiasanat Oculus Quest, virtuaalitodellisuus, VR, Unity		

Abstract

Author(s)	Type of Publication	Published
Tuomisto, Jonna	Thesis, UAS	2021
	Number of Pages	
	29	
Title of Publication		
Virtual Reality software development with Unity game engine		
Case rehabilitation virtual reality application for Konenäkö project		
Name of Degree		
Engineer (UAS)		
Name, title and organization of the client		
Anna Lahti, Konenäkö-project manager, LAB University of Applied Science		
Abstract		
<p>The aim of the thesis is to create a rehabilitative virtual reality application. The work was commissioned by Konenäkö project.</p> <p>The application is developed using the Unity game engine and wireless Oculus Quest virtual reality headset. The Unity game engine is a flexible IDE software. The thesis goes through basic concepts of Unity such as game objects, scenes, and scripts. Oculus Integration Package is developed by Oculus for helping development of Oculus applications at Unity and it is used in development process. In addition, thesis goes through some history of virtual reality and the differences between virtual reality hardware. Thesis clarifies some terms used around VR.</p> <p>As a result of the thesis, a functional virtual reality application was achieved. The application was tested during development. Comprehensive feedback was obtained from user experiences. Some features have not been completed due to project ending. It is possible to further develop the application by adding missing features and improving the completed activities.</p>		
Keywords		
Oculus Quest, virtual reality, VR, Unity		

Sisällys

1	Johdanto.....	1
2	Virtuaalitodellisuus.....	2
2.1	Historia	2
2.2	Virtuaalitodellisuuden ja todellisuuden ero	4
2.3	Laitteistot	5
2.4	Oculus Quest.....	7
3	Unity-pelimoottori.....	9
3.1	Unity lyhyesti	9
3.2	Unity-ohjelmiston käyttöliittymä ja editori	10
3.3	Peliobjektit	12
3.3.1	Törmäys ja törmäyksen tunnistus	13
3.3.2	Rigidbody	14
3.3.3	Materiaalit.....	14
3.4	Scene	15
3.5	Skriptit	15
3.6	Partikkelit.....	15
3.7	VR-sovelluksen kehitys.....	16
3.7.1	Oculus Integration Package	16
3.7.2	Oculus Questin käyttöönotto.....	17
4	Case: kuntouttava virtuaalitodellisuussovellus	19
4.1	Suunnittelu	19
4.2	Ympäristö	20
4.3	Liikkuminen.....	20
4.4	Aktiviteetit	21
4.4.1	Tammenterhojen keräys	22
4.4.2	Nuotion teko	23
4.5	Testaus.....	23
5	Yhteenvedo ja pohdinta	25
	Lähteet	26

Lyhenneluettelo

AR - Täydennetty, laajennettu tai lisätty todellisuus (augmented reality)

AV – Lisätty virtuaalisuus (augmented virtuality). Lisätyssä virtuaalisuudessa tuodaan oikeasta maailmasta objekteja virtuaaliseen maailmaan

HMD – Head mounted display, päähän asetettava näyttölaite

E3 – Electronic entertainment expo, vuosittain järjestettävä pelialan tapahtuma, jossa esitellään muun muassa uusia videopelejä sekä laitteistoja

Framework – Ohjelmistokehys. Framework on ohjelmistossa ”apuväline”, joka helpottaa kehitettävän ohjelmiston kehitystyötä

IDE – Integroitu kehitysympäristö tai ohjelmointiympäristö (integrated development environment), on ohjelmisto, joka tarjoaa tarvittavat työkalut applikaation/ohjelmiston tekoon ja sen koontiin

Immersio – Kokemukseen upottava ja todellisuuden kadottava tunne

Low poly – 3D grafiikka tyyli, jossa objekteissa on suhteellisen vähän polygoneja, jolloin asiat ja esineet näyttävät kulmikkaammilta

MR – Yhdistetty todellisuus tai yhdistelmätodellisuus (mixed tai hybrid reality)

SDK – Kokoelma ohjelmistokehitys työkaluja, yhtenäisenä asennettavassa paketissa

UI – User interface, käyttöliittymä. Ohjelmiston osa, jonka avulla käyttäjä käyttää ohjelmitoa

VR – Virtuaalitodellisuus (virtual tai artificial reality), tietokoneen avulla luotu simulaatio

XR – Laajennettu todellisuus (extended reality), X:llä viitataan määrittelemättömään, joten voi tarkoittaa mitä tahansa todellisuutta

1 Johdanto

Virtuaalitodellisuus ja todellisuutta eri tavoin muokkaavat teknologiat ovat olleet jo muutamia vuosia teknologioita, joihin panostetaan koko ajan enemmän ja enemmän. Niiden potentiaalia on ymmärretty ja niitä halutaan tuoda niin teollisuuteen kuin terveydenhuoltoon ja ihan vain viihdekäyttöön. Hyvänä esimerkkinä toimii kesällä 2016 Niantic yhtiön julkaisema Pokemon Go -mobiilipeli, jossa voi pyydystää Pokemon-hahmoja oikeassa maailmassa älypuhelimien kameran läpi. Sovelluksen suosio oli suurta heti julkaisusta lähtien. (Nieminen 2016.) Myös terveydenhuollossa on ymmärretty ja otettu käyttöön virtuaalitodellisuuden hyötyjä. Takala (2017) on koontanut julkaisuunsa monia erilaisia tutkimuksia liittyen aiheeseen ja mainitsee, että virtuaalitodellisuutta käytetään muun muassa halvauspotilaiden kuntouttamisessa, psykiatrisessa hoidossa sekä helpottamaan ikäviä ja kivuliaita toimenpiteitä.

Nykypäivänä todellisuutta muokkaavat teknologiat eivät enää ole vain suurien yhtiöiden saatavilla siksi, että ne olisivat kalliita tai järjestelmät suuria ja tilaa vaativia. Nykypäivänä tällaisen laitteen hankkiminen on vaivattomampaa ja älypuheliiniin on saatavilla paljon erilaisia Pokemon Go:n kaltaisia pelejä ja sovelluksia. Myös nämä maailmaamme muokkaavat laitteistot kehittyvät koko ajan kevyemmiksi, pienemmiksi ja helppokäyttöisemmiksi. Osa näistä laitteistoista on jo nyt täysin langattomia.

Opinnäytetyö on toteutettu yhdessä Konenäkö-hankkeen kanssa. Konenäkö-hanke auttaa ja kannustaa sosiaali-, terveys- ja hyvinvointialojen yrityksiä löytämään sekä kokeilemaan erilaisia konenäköteknologioihin perustuvia ratkaisuja. Konenäkö-hanke aloitettiin vuonna 2019. Hankkeella on tavoitteena luoda konenäkösovelluksille yhteiskehittämissympäristö, joka palvelisi monipuolisesti ja -alaisesti opiskelijoita, sote- ja hyvinvointi alan yrityksiä, kolmannen ja julkisen sektorin toimijoita. Kehittämissympäristön on tarkoitus jakaa tietoa konenäkösovelluksien käytöstä ja kokemuksista, sekä uusista teknologioista.

Tässä opinnäytetyössä toteutettiin kuntouttava virtuaalitodellisuussovellus. Sovelluksen tarkoitus on toimia apuna kuntoutuksessa ja rentouttavana virtuaalitodellisuuskokemuksena. Sovellus luotiin käyttäen Unity-pelimootoria sekä Oculus Quest virtuaalitodellisuuslaitetta. Lisäksi työn tavoitteena oli selvittää virtuaalitodellisuuden historiaa ja alkuperää, sekä sen ympärillä käytettyjä erilaisia termejä. Opinnäytetyössä erotellaan ja selkeytetään termit toisistaan. Teoriaosuudessa käydään läpi virtuaalitodellisuuslaitteistojen toimintaperiaatteita sekä tutustutaan Unity-pelimootoriin, sen käyttöön ja toimintaperiaatteisiin.

2 Virtuaalitodellisuus

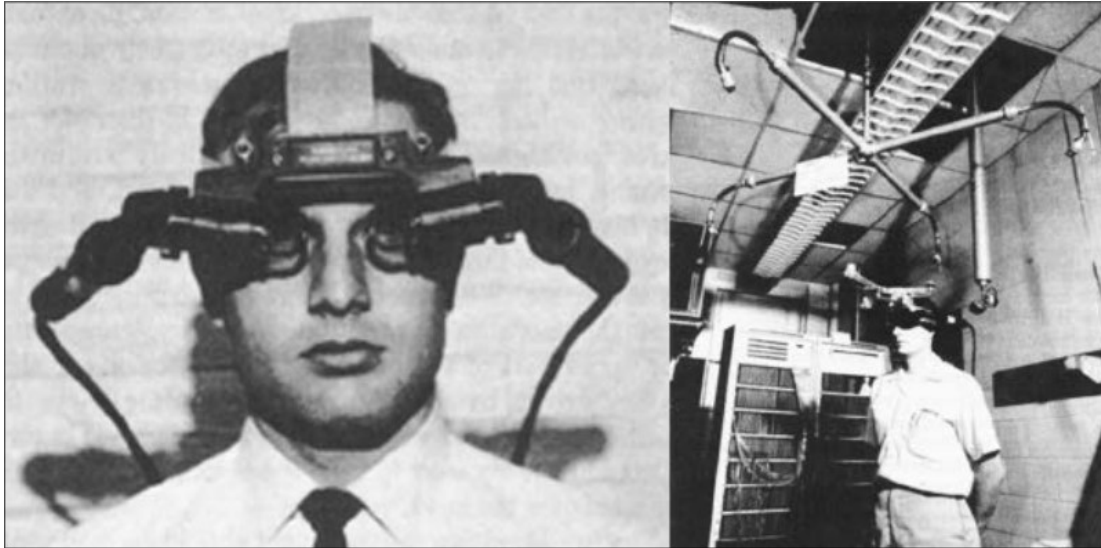
2.1 Historia

Vuonna 1955 elokuvakuvaaja Morton Heilig halusi kehittää erilaisen elokuvakokemuksen, "elokuvakokemuksen tulevaisuudesta", jossa olisi muitakin kuin vain näkö- ja kuuloaistin ärsykeitä. Hän kehitti Sensorama-nimisen Arcade pelikoneen kaltaisen laitteen (kuva 1), josta pystyi katselemaan muutamaa erilaista Heilighin ohjaamaa lyhytfilmiä. (Mealy 2018, 15.) Laite valmistui vuonna 1962. Sensorama antoi käyttäjälle immersioisen, eli upottavan ja todellisuuden kadottavan, elokuvakokemuksen. Laitteessa oli stereoskooppinen 3D näyttö, stereo äänet, penkin värinä, tuulettimien luoma tuuli sekä tuoksu. (Kipper G. & Rampolla J. 2012, 7.)



Kuva 1. Sensorama-laite (Mealy 2018, 15)

Vuonna 1968 Ivan Sutherland loi The Swords of Damocles -laitteen (kuva 2). Laitteessa oli päähän asetettava näyttölaite (head mounted display, lyhennettynä HMD), joiden optisten linssien läpi näki. Laite oli asennettava kattoon, ja se pystyi myös seuraamaan käyttäjänsä liikkeitä monipuolisesti. (Kipper G. & Rampolla J. 2012, 8.)



Kuva 2. The Swords of Damocles -laite (Linowes 2015, 3)

Monet peliyhtiöt yrittivät 1990-luvulla tuoda virtuaalitodellisuutta markkinoille. Sega yhtiö lupasi julkaista Sega VR -laitteen suosituille Sega Genesis -pelikonsolille, mutta kehitysongelmien takia laitetta ei julkistettu. Yhtiö kertoi laitteen aiheuttavan kovaa päänsärkyä ja pahoinvointia testausvaiheessa. Samoihin aikoihin Nintendo julkaisi Nintendo Virtual Boy -laitteen, joka oli ensimmäinen kannettava konsoli, joka näytti stereoskooppista 3D-kuvaa. Noin vuoden jälkeen Nintendo veti Virtual Boy -laitteen pois markkinoilta, sillä laitteen näyttämä kuva välkkyi ja sen pelättiin aiheuttavan pahoinvointia käytössä, sekä lapsille silmien karsastusta. (Mealy 2018, 16–17.)

Merkittävimpiä käännekohtia nykyaikaiselle virtuaalitodellisuudelle on vuoden 2012 E3-tapahtuma, jossa Palmer Luckey esitteli kehittelemäänsä virtuaalitodellisuuslaitetta, Oculus Rift Developer Kit, lyhyesti DK1. Luckey kehitti itse Oculus Rift -laitteiston konseptin, sillä hän ei ollut tyytyväinen sen ajan virtuaalitodellisuuslaitteiden tasoon. Siltikään DK1 ei ollut täydellinen, vaan aiheutti pidemmässä yhtäjaksoisessa käytössä huonovointisuutta ja päänsärkyä. Vuonna 2014 julkaistiin Oculus Rift Developer Kit 2, joka oli mukavampi käytössä sekä teknisesti kehittyneempi. (Alex 2018.)

2.2 Virtuaalitodellisuuden ja todellisuuden ero

Virtuaalitodellisuuden ja todellisuuden raja ei ole täysin yksiselitteisesti rajattavissa, sillä virtuaalisuutta voidaan lisätä eri tavoin ja eri volyymein todellisuuteen. Olisi todella epäselkeää puhua immersiiivisestä virtuaalitodellisuuskokemuksesta ja puhelimen kameran läpi katsottavasta Pokemon-hahmosta samana kokemuksena tai puhutella näistä samalla termillä. Usein virtuaalitodellisuutta käytetään kattoterminä kaikenlaisille immersioisille teknologialla tuotetuille kokemuksille.

Paul Milgram loi Virtuality Continuum-asteikon (kuva 3), jonka avulla voidaan määritellä virtuaalisuuden ja todellisuuden rajaa. Asteikolle on määritetty käsitteiden sijainteja toisiinsa nähden, sekä havainnollistettu mitkä termit toimivat yläkäsitteinä muille termeille. (Mealy 2018, 8)



Kuva 3. Paul Milgramin Reality – Virtuality continuum (Mealy 2018, 14)

XR

XR:stä puhutaan laajennettuna todellisuutena. Lyhenne tulee englanniksi sanoista extended reality, X reality ja cross reality. X-kirjaimella viitataan määrittämättömään, eli se voi käytännössä tarkoittaa mitä tahansa todellisuutta. (Rudnäs 2019.) XR-terminä toimii niin sanotusti yläkäsitteenä kaikille lyhenteille, eli VR, AR, AV ja MR (Mealy 2018, 14).

MR

MR (mixed tai hybrid reality) on yhdistetty todellisuus tai yhdistelmätodellisuus. Yhdistetty todellisuus on havaintojen kokonaisuus, jossa yhdistyy virtuaalitodellisuutta sekä fyysistä ympäristöä siten, että virtuaalisen ja fyysisen todellisuuden elementit ovat vuorovaikutuksessa toisiinsa. Yhdistettyä todellisuutta ovat esimerkiksi hologrammit. (Sanastokeskus 2017a.)

AR

AR (augmented reality) tarkoitetaan täydennettyä, laajennettua tai lisättyä todellisuus. Täydennyssä todellisuudessa fyysiseen ympäristöön tuotetaan tietotekniikan avulla havaittavaa tietoa. Täydennettyä todellisuutta on esimerkiksi, se että käyttäjä katsoo puhelimen kameran läpi ravintolaa ja puhelin antaa tiedot ravintolan ruokalistasta ja aukioloajoista. (Sanastokeskus 2017b.)

VR

Virtuaalitodellisuus (virtual tai artificial reality), lyhennettynä VR, on tietokoneen avulla luotu simulaatio, jossa yritetään luoda mahdollisimman todelliselta vaikuttava kokonaisuus aisteille virtuaalisesti. Virtuaalitodellisuudessa käyttäjä voi myös toimia vuorovaikutteisesti keinoitekoisen ympäristön kanssa. Virtuaalitodellisuus voi pyrkiä simuloimaan jotakin todellista tai kuvitteellista ympäristöä. (Sanastokeskus 2017c.)

2.3 Laitteistot

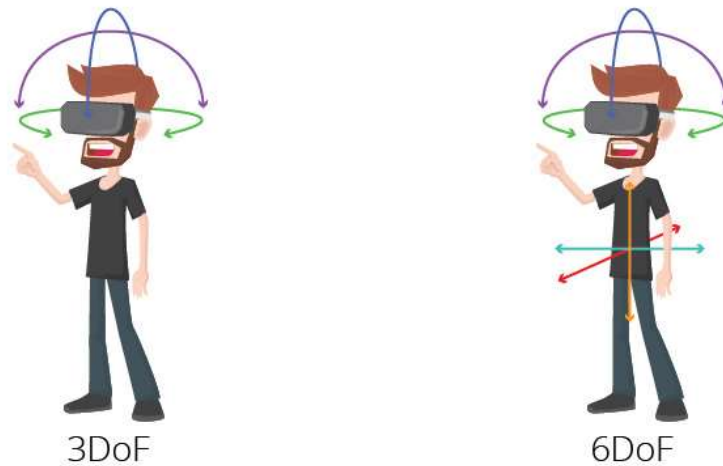
XR laitteita on monia erilaisia sekä eri valmistajilta. Perinteisissä virtuaalitodellisuuslaseissa on kaksi näyttöä, yksi molemmille silmille, joiden avulla luodaan käyttäjille syvyysvaikutelma. Tämä on myös saavutettavissa erilaisilla virtuaalitodellisuuslasikehikoilla, joihin älypuhelimien voidaan kiinnittää. Älypuheliiniin on saatavilla erilaisia virtuaalitodellisuuskokemuksia applikaatioita, jolloin puhelimen näyttö toimii näyttönä luomaan syvyysvaikutelman. Nykyaikana on myös saatavilla erilaisia AR ja MR luokiteltavia laseja, joiden läpi näkee ympäristön, jossa on ja liikkuu, ilman että HMD peittää käyttäjän näkyvyyttä. Koska virtuaalitodellisuuslaitteita löytyy niin monta erilaista mallia, on niiden toimintaperiaatteissa eroa.

3DoF ja 6DoF

DoF, eli Degree of Freedom tarkoittaa sitä, kuinka monen eri suunnan liikkeen virtuaalitodellisuuslasit pystyvät havainnoimaan 3D-avaruudessa. 3D-avaruudessa on kuusi eri akselia, joilla objekti voi liikkua. Nämä kuusi akselia voidaan karkeasti erottaa kahteen kategoriaan, objektin pyörimisliikkeisiin (rotational motion) sekä objektin etenemisliikkeisiin (translational motion). (Google VR 2018; Veative 2020.) Tätä on havainnollistettu nuolin kuvassa 4.

3DoF tarkoitetaan sitä, että voidaan seurata virtuaalitodellisuuslasien pyörimisliikkeitä, eli lasien käyttäjä voi kääntää päätään sivuille katsomalla oikealla ja vasemmalle, ylös ja alas sekä kallistaa päätään (Google VR 2018).

6DoF tukevat virtuaalitodellisuuslasit pystyvät seuraamaan kaikkia 3DoF seuraamia akseleita, mutta myös sen lisäksi käyttäjän etenemisliikkeitä, eli nousemista ja laskua, liikkeitä sivuille sekä liikkeitä eteen- ja taaksepäin (Google VR 2018).



Kuva 4. 3DoF ja 6DoF seurannan erot havainnollistettuna (Veative 2020)

Seurantajärjestelmät

Osa halvemmista virtuaalitodellisuuslaitteistoista sekä puhelimen avulla luodut virtuaalitodellisuuskokemukset käyttävät laitteen gyroskooppia sekä kiihtyvyyssanturia liikkeiden seurantaan. Näiden teknologioiden avulla voidaan saavuttaa vain 3DoF-kokemus. (Mechatech 2021a.)

Outside-in järjestelmässä erilliset laitteet, joita kutsutaan termeillä majakka tai sensori (base stations), seuraavat virtuaalitodellisuuslaseja ja niiden liikkeitä. Majakka termiä käytetään, sillä seurantaan tarvittu erillinen laite yleensä lähettää ihmissilmälle näkymätöntä valoa. Valo osuu virtuaalitodellisuuslasien ja ohjaimien antureihin, joiden avulla laitteisto pystyy analysoimaan ja laskemaan lasien sekä ohjainten asennon vektorimuutosten avulla. Outside-in järjestelmässä yleensä käytetään 2–4 sensoria, jotta sensorien näkemä alue on tarpeeksi laaja. Hyvin katetulla alueella virtuaalitodellisuuslasien käyttäjän on mahdollista liikkua vapaasti ja kääntyä 360 astetta, ja seurantajärjestelmä ei hävitä virtuaalitodellisuuslaseja tai ohjaimia näkyvistään. Outside-in järjestelmä voi toimia myös yhdellä sensorilla, mutta tällöin virtuaalitodellisuuslasien käyttöalue pienenee. Silti tällaisessa outside-in järjestelmässä on ongelmana se, että jos virtuaalitodellisuuslasien ja sensorin väliin tulee jokin

este, esimerkiksi huonekalu, seurantajärjestelmä hukkaa virtuaalitodellisuuslasit tai ohjaimen, jolloin virtuaalitodellisuuskokemuksen immersio kärsii. (Mechatech 2021b.)

Inside-Out järjestelmässä virtuaalitodellisuuslasit itsenäisesti käsittelevät liikkeen seurannan. Nykypäivän inside-out järjestelmät käyttävät muun muassa kameroita, valoja, gyro-skooppiä sekä kiihtyvyyssanturia. Näiden tietojen avulla virtuaalitodellisuuslasit luovat tilasta dataa, esimerkiksi 3D-kartan, jolla järjestelmä seuraa käyttäjän liikkeitä. (Mechatech 2021a&b.)

2.4 Oculus Quest

Oculus Quest on vuonna 2019 julkaistu Android-pohjainen, täysin itsenäinen ja langaton virtuaalitodellisuusjärjestelmä (kuva 5). Quest käyttää Inside-Out seurantajärjestelmää neljän laajakuvaa kameran sekä sensoreiden avulla, ja siinä on 6DoF seuranta. Laitteessa on Qualcomm Snapdragon 835 -mobiilipiirisarja ja sen akunkestoksi luvataan noin 2–3 tuntia. Laseissa on 1440x1600 resoluution OLED-näytöt sekä 72 Hz virkistystaajuus. (Oculus For Developers 2021a.)



Kuva 5. Oculus Quest virtuaalitodellisuuslaite

Oculus Quest -virtuaalitodellisuuslaitteen mukana tulee ohjaimet kumpaankin käteen, joilla kontrolloidaan toimintoja virtuaalitodellisuudessa. Oculus Quest -laite tukee myös käsiseurantaa, eli käyttäjä voi käyttää omia käsiään ohjaimina virtuaalitodellisuudella ilman erillisiä ohjaimia. Tosin käsiseurantaa ei tueta kaikissa virtuaalitodellisuussovelluksissa ja -peleissä. (Oculus 2021).

Vaikka Oculus onkin täysin langaton virtuaalitodellisuuslaite, on sitä mahdollista käyttää myös tietokoneeseen yhdistettynä. Oculus (2019) esitteli Oculus Link -ominaisuuden Oculus Quest -laitteistoon vuonna 2019. Oculus Link mahdollistaa sen, että Oculus Quest -laitetta voidaan käyttää tietokoneelta ja sen avulla saa pääsyn Oculusin omaan tietokonesovellukseen ja SteamVR kautta käytettäviin virtuaalitodellisuussovelluksiin. Tietokoneen tulee olla tarpeeksi tehokas sekä kaapelin laadukas. Tämä ominaisuus antaa Oculus Quest -laitteelle joustavuutta ja monipuolisuutta sekä mahdollisuuden saavuttaa sisältöä, jota ei muuten voi saada Oculus Quest -laseihin.

3 Unity-pelimoottori

3.1 Unity lyhyesti

Unity on 3D- ja 2D-grafiikkaa tukeva pelimoottori. Unity on suosittu ja tunnettu pelimoottori pelialalla, ja se on yhtä lailla suurien pelistudioiden kuin pienten indie-pelistudioiden suosiossa. Unity pelimoottorilla on kehitetty paljon tunnettuja pelejä, esimerkiksi Ori and the Blind Forest (kuva 6), Superhot, Subnautica sekä mobiilipeli Temple Run.



Kuva 6. Ori and the Blind Forest – videopeli (Moon Studios 2021)

Unity sisältää paljon sisäänrakennettuja ominaisuuksia, jotka helpottavat pelien kehitystyötä. Tällaisia sisäänrakennettuja ominaisuuksia ovat esimerkiksi fysiikkamoottori, 3D-renderöinti ja törmäyksen tunnistus. Unity-pelimoottori voidaan luokitella IDE-ohjelmistoksi (integrated development environment), eli Unity tarjoaa pelien kehitykseen lähes kaikki tarvittavat työkalut ohjelmistossaan. Unity-pelimoottorilla voi luoda pelejä monipuolisesti eri alustoille, muun muassa Windows-, MacOS- ja Linux-käyttöjärjestelmille, Android- ja iOS-mobiilikäyttöjärjestelmille, Oculus- ja HTC-virtuaalitodellisuusjärjestelmille sekä pelikonsoleille. (Sinicki 2021.)

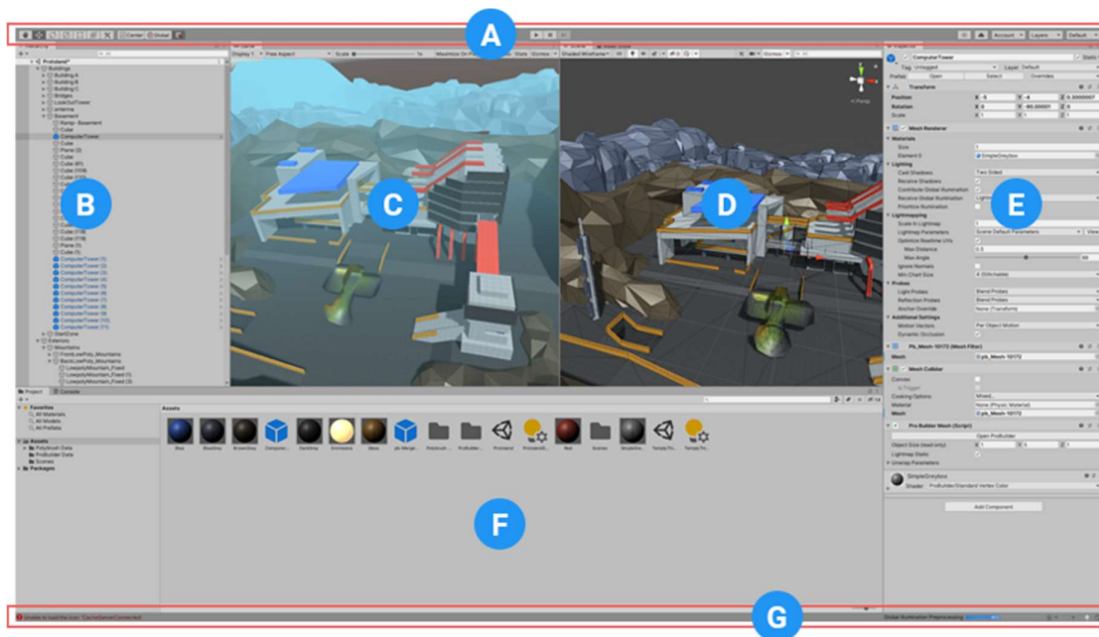
Unity tarjoaa myös Unity asset store palvelua. Unity asset store on kauppa, josta voi etsiä jo valmiita asetteja. Asetit ovat kattotermi Unity-ohjelmistossa toimiville nimikkeille. Asset

voi olla, esimerkiksi 3D-malli tai animaatiokontrolleri. Unity asset storesta löytyy muun muassa Unity technologiesin tarjoamia, sekä rekisteröityneiden Unity-käyttäjien luomia assetteja. (Unity 2021.)

3.2 Unity-ohjelmiston käyttöliittymä ja editori

Unity-ohjelmiston UI (user interface), eli käyttöliittymä, ja editori on karkeasti eroteltavissa seitsemään eri osaan, jotka on kuvassa 7 eroteltu seuraavasti:

- A – Työkalurivi (toolbar)
- B – Hierarkiaikkuna
- C – Pelinäkymä
- D – Scene-näkymä
- E – Tarkasteluikkuna
- F – Projekti-ikkuna
- G – Statusrivi (Unity Documentation 2021b.)



Kuva 7. Unity-pelimoottorin käyttöliittymä (Unity Documentation 2021b)

Työkalurivi

Työkalurivissä on koottuna editorin eri osissa tarvittavia erilaisia työkaluja. Työkalurivistä löytyy muuntotyökaluja (transform), joilla voi vaikuttaa scene-näkymässä sijaitseviin objekteihin. Työkalurivissä myös on play-, pause- ja stop-painikkeet, joilla kontrolloidaan pelinäkymää. Lisäksi työkalurivissä sijaitsee yhteistyönappula, joka avaa Unity collaborate -palvelun; Unity pilvipalvelu -nappula (Cloud Service) sekä Unity-tilinappula. Lisäksi työkalurivissä sijaitsee Unity projektiin luodut kerrokset-valintaikkuna (layers), joilla voi kontrolloida mitkä objektit näkyvät scene-näkymässä, sekä aseteluvälikkö (layout). Asetteluvälikköä löytyy muutamia erilaisia versioita Unity editorin ulkoasusta, sekä valikon kautta on mahdollista tallentaa kustomoitu editori. (Unity Documentation 2021c.)

Hierarkia - ikkuna

Hierarkiaikkunasta löytyvät kaikki scene-näkymässä sijaitsevat objektit. Objektien järjestyttä voi muuttaa ikkunassa vapaasti, tosin Unityssa on saatavilla asetus, jonka avulla objektit järjestyvät automaattisen aakkosjärjestykseen. Hierarkiaikkunassa on mahdollista muuttaa objektien näkyvyyttä (visibility) sekä määrittellä ovatko objektit valittavissa. Nämä asetukset vaikuttavat objektien toimintaan scene-näkymässä. Jos scene-näkymässä lisätään tai poistetaan objekti, muutokset tapahtuvat myös hierarkiaikkunassa. (Unity Documentation 2021d.)

Pelinäkymä

Pelinäkymä renderöi miltä kehitettävä peli näyttäisi scene-näkymään lisättyjen kameroiden läpi. Kun työkalurivin play-painiketta painetaan, pelinäkymä renderöinnin lisäksi pelin simulointi alkaa. Kun play-painike on painettuna, kaikki simuloinnin aikana tehdyt muutokset ovat väliaikaisia ja nollautuvat, kun simulointi lopetetaan. Pelinäkymässä on oma kontrollointirivi (Game view Control Bar), josta löytyy erilaisia pelinäkymään vaikuttavia asetuksia. (Unity Documentation 2021e.)

Scene-näkymä

Scene-näkymä on interaktiivinen näkymä, johon scenen peliobjektit luodaan. Näkymässä voi tarkastella peliin lisättyjä objekteja: kameroita, valoja sekä vastaavia peliin tarvittavia objekteja. Scene-näkymässä on mahdollista liikuttaa, käsitellä sekä muuttaa objekteja. (Unity Documentation 2021f.) Scene-näkymällä on myös kontrollointirivi (Scene view control bar), joka sisältää erilaisia asetuksia scene-näkymän katseluun. Kontrollointirivin avulla

voi myös määritellä ovatko peliin lisätyt valot ja audiot käytössä scene-näkymässä. (Unity Documentation 2021g.)

Tarkasteluikkuna

Tarkasteluikkunassa voi tarkastella ja muokata, muun muassa objekteja, asetteja, koodeja, sekä editorin asetuksia ja ominaisuuksia. Unityssa on mahdollista muokata montaa valintaa yhtä aikaa. Tarkasteluikkuna näyttää viimeisimmän valinnan tiedot, mutta tarkasteluikkunan voi lukita siten, että vaikka valinta muuttuisi, näkyy lukitun valinnan ominaisuudet edelleen. (Unity Documentation 2021h.)

Projekti-ikkuna

Projekti-ikkuna sisältää kaikki projektin tiedostot, kansiot sekä kansiorakenteen. Projekti-ikkuna on muutettavissa yhden tai kahden sarakkeen näkymään, jossa kahden sarakkeen näkymä tarjoaa esikatselu ominaisuuden joillekin tiedostotyypeille. (Unity Documentation 2021i.)

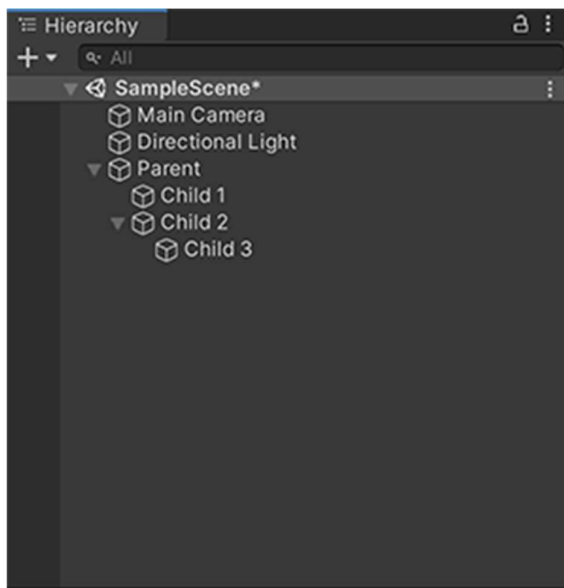
Statusrivi

Statusrivi antaa ilmoituksia ja lisätietoja liittyen Unityn prosesseihin, esimerkiksi konsolin viimeisin viesti. Statusrivistä on nopea pääsy ikonien avulla joihinkin Unity prosesseihin, esimerkiksi statusrivistä löytyy edistymispalkki, jota painamalla avautuu taustatehtävät-ikkuna, josta on mahdollista nähdä yksittäisten tehtävien ja alitehtävien edistyminen. (Unity Documentation 2021j.)

3.3 Peliobjektit

Unityssa kaikki pelissä olevat objektit ovat peliobjekteja (GameObject). Objektit eivät itsenäisesti ole toimivia, esimerkiksi pelihahmoja tai valoja, vaan objekteihin täytyy liittää komponentteja. Unity tarjoaa laajasti erilaisia komponentteja, joiden avulla luodaan objekteihin erilaisia toimintoja. Unity tarjoaa myös valmiiksi komponentteja sisältäviä objekteja, esimerkiksi valoja ja kameroita. Jokainen Unity peliobjekti sisältää pakollisen Transform-komponentin, joka määrittelee objektin sijainnin, rotaation sekä koon scene-näkymässä. (Unity Documentation 2021k.) Objektien komponenttien lisääminen, poisto ja muokkaus tapahtuu tarkasteluikkunassa. Komponentit ovat Unityssa hyvin joustavia, sillä niissä on runsaasti ominaisuuksia, joita voi säädellä ja muuttaa tarpeen mukaan. Jotkut komponentit tarvitsevat erillisiä tiedostoja toimiakseen, Unityssa näitä nimitetään viitatuiksi ominaisuuksiksi (reference property), esimerkiksi äänikomponentti tarvitsee äänitiedoston, jotta äänikomponentti voi soittaa äänen. (Unity Documentation 2021l.)

Unity käyttää peliobjektien kanssa vanhempi - lapsi hierarkiaa, joka on nähtävissä hierarkiaikkunassa, eli objekti voi sisältää muita objekteja, jotka sitten perivät ylimmän objektin, eli "vanhemman", ominaisuudet. Ominaisuuksia, jotka periytyvät objektin alaisuudessa oleville objekteille, eli "lapsille", ovat peliobjektin sijainti, koko ja rotaatio. Eli jos tätä ylimmän tason objektia liikutetaan scene-näkymässä, objektin lapset liikkuvat myös. Lapsiobjekteilla voi myös olla niin sanotusti sisäkkäin (nested) omia lapsiobjekteja, kuten kuvassa 8 child2-objektilla on alaisuudessaan ja lapsenaan child3-objekti. Tällaisissa tapauksissa nämä lapsien lapset perivät sekä oman vanhempansa että ylimmän tason vanhemman ominaisuudet, eli kuvan 8 child3-objekti perii parent- ja child2-objektien ominaisuudet. (Unity Documentation 2021d.)



Kuva 8. Unity objektien vanhempi - lapsi hierarkia (Unity Documentation 2021d)

Peliobjekteja on mahdollista merkitä passiiviseksi (inactive) tarkasteluikkunassa. Tämä poistaa objektin ja sen toiminnan scene-näkymästä, kunnes objekti muutetaan aktiiviseksi. Jos tämä objekti on vanhempi, objektin kaikki lapset passivoituvat myös. Yksittäistä lapsiobjektia ei voi aktivoida niin että lapsen vanhempiobjekti pysyisi passiivisena. (Unity Documentation 2021m.)

3.3.1 Törmäys ja törmäyksen tunnistus

Törmäyksen tunnistin (collider) on komponentti, joka tunnistaa objektien törmäyksen. Törmäys tapahtuu Unityssä kun objekti, jossa on törmäyksen tunnistin, törmäilee toiseen objektiin

jossa on myös tämä tunnistin. Unityn fysiikkamoottori tunnistaa tällöin törmäyksen. Mekaniikkaa voi käyttää peleissä esimerkiksi soittamaan äänen, kun objekti osuu johonkin. (Unity Documentation 2021n.)

Tunnistin on objektin ympärillä oleva näkymätön muoto. Unityn ja pelin kannalta on yleensä riittävää, että tunnistimen muoto mukailee löyhästi objektin alkuperäistä muotoa, sillä tarkasti objektin muotoa mukailevat pintacolliderit (mesh collider) vaativat paljon prosessointi tehoa. (Unity Documentation 2021n.)

Törmäystunnistimeen on mahdollista lisätä fysiikka materiaali (Physic Material), jonka avulla luodaan objekteihin kitkaa ja kimmoisuutta. Fysiikka materiaalissa voi määritellä objektin dynaamista sekä staattista kitkaa. Fysiikka materiaalissa voi myös määritellä, että kun objektit törmäävät toisiinsa, kuinka näiden kahden objektin välinen kitka ja kimmoisuus käytäytyvät. (Unity Documentation 2021o.)

Törmäyksen tunnistin voi olla tyypiltään myös staattinen. Silloin objektiin ei lisätä jäykän kappaleen fysiikkakomponenttia (rigidbody). Staattisia törmäyksen tunnistimia käytetään yleensä seiniin, lattioihin ynnä muihin vastaaviin liikkumattomiin elementteihin, joiden halutaan silti vaikuttavan liikkuviin objekteihin. (Unity Documentation 2021n.)

3.3.2 Rigidbody

Rigidbody on komponentti, joka mahdollistaa, että objektiin voi vaikuttaa fysiikan lait, esimerkiksi painovoima. Jos objektiin halutaan voida vaikuttaa ulkoisilla voimilla, tarvitsee objekti myös collider-komponentin.

Rigidbody on mahdollista irrottaa Unityn fysiikkamoottorista. Rigidbody-komponentissa on kinemaattisuusasetus (Is Kinematic), jonka ollessa aktivoituna, on objektia mahdollista siirtää kinemaattisesti esimerkiksi koodin avulla. Unityssa kinemaattisuus on peliohjelman liikkettä, joka ei ole fysiikkamoottorin tuottamaa. (Unity Documentation 2021.p)

3.3.3 Materiaalit

Materiaaleilla määritellään Unityssa miltä peliohjelmit näyttävät. Peliohjelman materiaali määritellään lisäämällä valittu materiaali objektin Mesh Renderer -komponenttiin. Materiaalit luokitella Unityssa aseteiksi, sillä ne sisältävät muun muassa tiedot miltä renderöity pinta näyttää, värin, sekä tekstuurin. Jos objektin materiaalia muuttaa tarkastelijaikkuna kautta, muutokset tapahtuvat koko materiaali aseteissa, esimerkiksi jos tarkasteluikkunan kautta vaihtaa objektin materiaalin väriä, kaikki objektit, joissa on sama materiaali, vaihtavat myös väriä. (Unity 2013.)

3.4 Scene

Scenet ovat säiliöitä, joihin Unityssa peli luodaan. Sceneihin sisällytetään kaikki kehitettävän pelin peliobjektit, kuten esimerkiksi ympäristö, pelihahmot, valot, kamerat sekä UI. Kehitettävään peliin on mahdollista luoda monta sceneä ja niitä voi ajatella pelin tasoina (level), joihin voi luoda esimerkiksi uuden ympäristön pelin edetessä. (Unity Documentation 2021q.)

3.5 Skriptit

Unityn tarjoamat komponentit tarjoavat laajasti mahdollisuuksia, mutta valitettavasti niissä on rajansa. Unityssa on mahdollista luoda skriptiedostoja (script file), joiden avulla voi luoda kustomoituja komponentteja. Skripti on lisättävä peliobjektiin, jotta sitä voi käyttää komponenttina. Unity tukee natiivisti C# koodikieltä. Lisäksi Unity tukee myös joitakin .NET kieliä. (Unity Documentation 2021r.)

Kun Unityssa luo uuden skriptitiedoston, työkalu luo tiedoston nimen perusteella tiedostoon alustavan sisällön. Skriptissä luokan nimen ja tiedoston nimen on oltava sama, jotta skripti on toimiva. (Unity Documentation 2021r.)

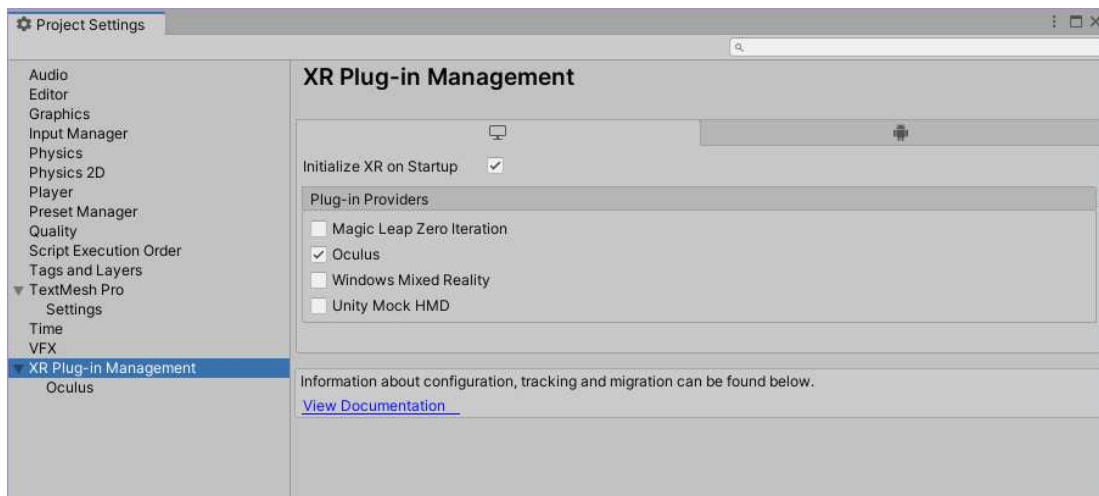
Unityn skripteissä on määriteltävä kaksi funktiota, Update ja Start. Update-funktio suoritetaan jokaisen frame-päivityksen aikana. Start-funktiota kutsutaan ennen pelin alkua, eli ennen kuin Update-funktiota kutsutaan ensimmäisen kerran. (Unity Documentation 2021r.)

3.6 Partikkelit

Partikkelisysteemin avulla voi simuloida ja renderöidä erilaisia efektejä, esimerkiksi tulta, savua tai nestettä, pienien kuvien tai 3d-objektien avulla generoimalla niitä suuri määrä kerralla. Partikkelisysteemikomponentilla on suuri määrä erilaisia muuteltavia ominaisuuksia ja asetuksia. Unityssa näitä ominaisuuksia ja asetuksia on niputettu, sekä eroteltu aiheen mukaan erillisiin moduuleihin. Partikkelisysteemillä on oma erillinen partikkeliefektipaneeli scene-näkymässä, joka sisältää muutamia lisäasetuksia liittyen partikkelien esikatseluun. Sen avulla voi muun muassa käynnistää, lopettaa ja aloittaa alusta partikkeliefektin simuloinnin; nähdä simuloitavien partikkelien määrän ja määritellä haluaako nähdä esikatselussa vain työstettävän partikkelisysteemin, vai myös scene-näkymässä sijaitsevia muita partikkelisysteemejä. (Unity Documentation 2021s.)

3.7 VR-sovelluksen kehitys

Virtuaalitodellisuussovelluksia pystyy luomaan Unity-pelimoottorilla. Unity editorissa itsessään on XR Plug-in Management framework -työkalu (kuva 9), jonka avulla voi asentaa tarvittavia työkaluja virtuaalisovelluksien luontiin. On hyvä huomioida, että virtuaalitodellisuussovelluksien kehityksessä virtuaalitodellisuusjärjestelmän alusta on oltava tiedossa, sillä eri valmistajien virtuaalitodellisuuslaitteistojen sovellukset toimivat eri tavoin. (Oculus for Developers 2021d.)



Kuva 9. Unity XR Plug-in Management -työkalu

3.7.1 Oculus Integration Package

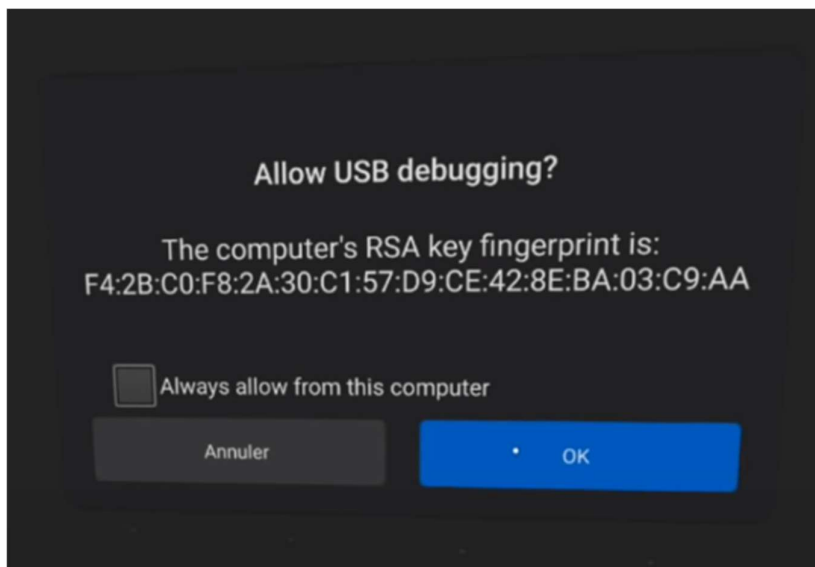
Oculus Integration package on Oculuksen tarjoama paketti, joka sisältää virtuaalitodellisuussovelluksen kehittämistä helpottavia ominaisuuksia, komponentteja, koodeja, plugineja sekä SDK-paketteja. Esimerkiksi Oculus Integration Package tarjoaa suoraan 3d-mallit ja assetit eri Oculus -laitteiden ohjaimille ja tarvittavat komponentit esineiden nosteluun virtuaalitodellisuussovelluksessa. (Oculus for Developers 2021b.)

Oculus Integration packagen saa Unityssa käyttöön lataamalla sen Unityn Asset Storesta. Lataus ei automaattisesti lisää pakettia Unity projektiin, vaan se täytyy manuaalisesti lisätä projektiin Unity editorissa. (Oculus for Developers 2021c.)

Oculus Questin sovellukset ovat Android pohjaisia. Android pohjaiset sovellukset tarvitsevat toimiakseen Android Manifest -tiedoston. Android Manifest -tiedosto sisältää Android-sovellukselle tarpeellista metadataa, esimerkiksi käyttöoikeuksia, konfigurointeja sekä tietoja tunnetuista Android-versioista. Oculus Quest automaattisesti luo ja lisää Android Manifest -tiedoston. Android Manifest -tiedostoa on päivitettävä erikseen myös Unity editorissa, sillä Unityssä tehdyt muutokset eivät automaattisesti päivity Android Manifest -tiedostoon. (Oculus for Developers 2021d.)

3.7.2 Oculus Questin käyttöönotto

Jotta Oculus Quest -laitetta voidaan käyttää sovelluksen kehitys tarkoituksiin, laitteelle pitää antaa lupa tehdä kehitysmuutoksia. Oculus Quest täytyy yhdistää Bluetooth-yhteydellä puhelimeen ja Oculus Appin asetuksista asettaa Developer Mode -asetus päälle. Vaikka puhelimen appin kautta on Developer Mode -asetus kytketty päälle, täytyy Oculus Quest -laitteeseen kytkeä vielä tietokoneeseen ja antaa lupa tietokoneen tehdä muutoksia Oculus Quest -laitteeseen (kuva 10). (Oculus for Developers 2021e.)



Kuva 10. Oculus Quest USB Debugging -ikkuna

Unity editorissa on myös tehtävä muutamia muutoksia, jotta Unity sovellus toimii Quest virtuaalitodellisuuslaitteessa oikein. Jotta sovellus voidaan asentaa Oculus Quest -laseihin, VR-lasien on oltava yhdistettynä tietokoneeseen USB-yhteydellä. Unityn Build Settings -asetuksista alusta (platform) täytyy olla valittuna Android-käyttöjärjestelmä, sekä Texture

Compression on vaihdettava ASTC, sillä kyseinen formaatti tukee korkeampia pakkaussuh-
teita sekä parempaa kuvanlaatua Android-laitteissa. (Oculus for Developers 2021d; Unity
Documentation 2021a.)

4 Case: kuntouttava virtuaalitodellisuussovellus

4.1 Suunnittelu

Virtuaalitodellisuussovellukseen oli tarkoitus luoda alun perin vain jonkinlainen luontoympäristö ja kehittää erilaisia toimintoja sen ympärille. Luontoalue vaikutti hyvältä idealta, sillä luonnolla koetaan olevan ihmiseen rauhoittava sekä rentouttava vaikutus, vaikka se olisikin vain esimerkiksi kuva metsästä. Luontoympäristöt, joita sovellukseen suunniteltiin, olivat muutamia erilaisia metsiä sekä ranta. Rantaympäristöstä luovuttiin pian, sillä aktiviteettien suunnittelussa tuntui, että rantaympäristössä aktiviteetit toistavat itseään. Metsäympäristöön oli helpompi keksiä monipuolisemmin erilaisia ja toisistaan eroavia aktiviteetteja.

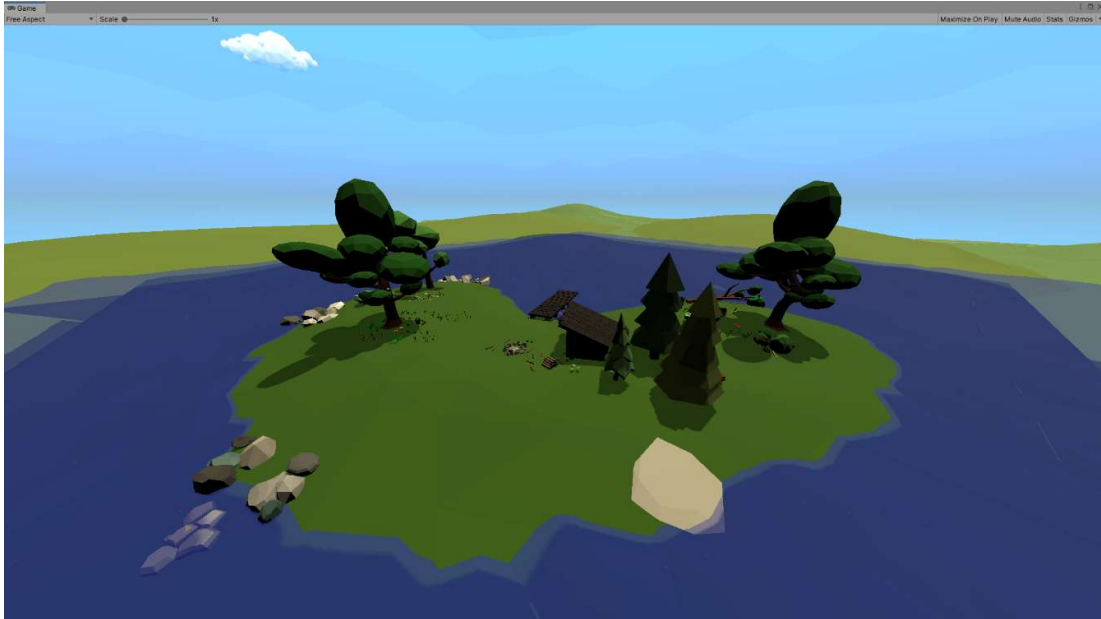
Virtuaalitodellisuussovelluksen laitteeksi valittiin Oculus Quest, koska laite on täysin langaton, kevyt ja laitteen toivottiin olevan helposti liikuteltavissa. Oculus Questin outside-in järjestelmä oli myös tarpeellinen ominaisuus, jotta laitteen käyttöönotto on helppoa sekä nopeaa, eivätkä virtuaalitodellisuuslasit tarvitse toimiakseen erillisiä sensoreita.

Kuntouttavan virtuaalitodellisuussovelluksen tehtävänanto oli löyhä, sillä Konenäkö-hanke ei ollut alusta asti projektissa mukana. Sovelluksen kehitys oli aloitettu jo toisessa projektissa. Konenäkö-hanke kiinnostui sovelluksesta kehityksen alkuvaiheessa. Kun virtuaalitodellisuussovelluksen kehitys liitettiin osaksi Konenäkö-hanketta, sovelluksen alkuperäinen suunnitelma oli hankkeen mielestä sopiva ja toimiva. Sovelluksesta lähinnä toivottiin, että se olisi käyttäjille miellyttävä sekä helppokäyttöinen niin, että mahdollisimman monet eri tasoiset ihmiset voisivat testata sovellusta. Konenäkö-hankkeessa sovellusta haluttiin myös testata kehityksen aikana, sekä saada palautetta suoraan, jotta kehityksen aikana voitiin reagoida palautteeseen ja tehdä tarvittavia muutoksia sovellukseen saadun palautteen pohjalta.

Sovelluksen suunnitteluvaiheessa aikaa kului runsaasti yleiseen tiedonetsintään virtuaalitodellisuussovellusten kehityksestä Unity-ympäristössä, sekä kuinka Oculus Quest -laitteen kanssa kehitystyötä tehdään. Ajan kulumista lisäsi se, että suurin osa ohjeistuksista olivat joko vanhempien Unity versioiden tai vanhempien Oculus Integration Package -versioiden ohjeistuksia.

4.2 Ympäristö

Pelialueena toimii veden ympäröimä saari (kuva 11). Ympäristöllä haetaan suomalaisen luonnon tuntua. Ympäristöön on sen takia sijoitettu suomalistyyppisellä arkkitehtuurilla mallinnettu laavu ja laituri. Alun perin ympäristöksi oli tarkoitus luoda isohko metsäalue, mutta kehityksen alkuvaiheella jo huomattiin, että niin suurelle pelialueelle ei olisi tarvetta.



Kuva 11. Virtuaalitodellisuussovelluksen saariympäristö

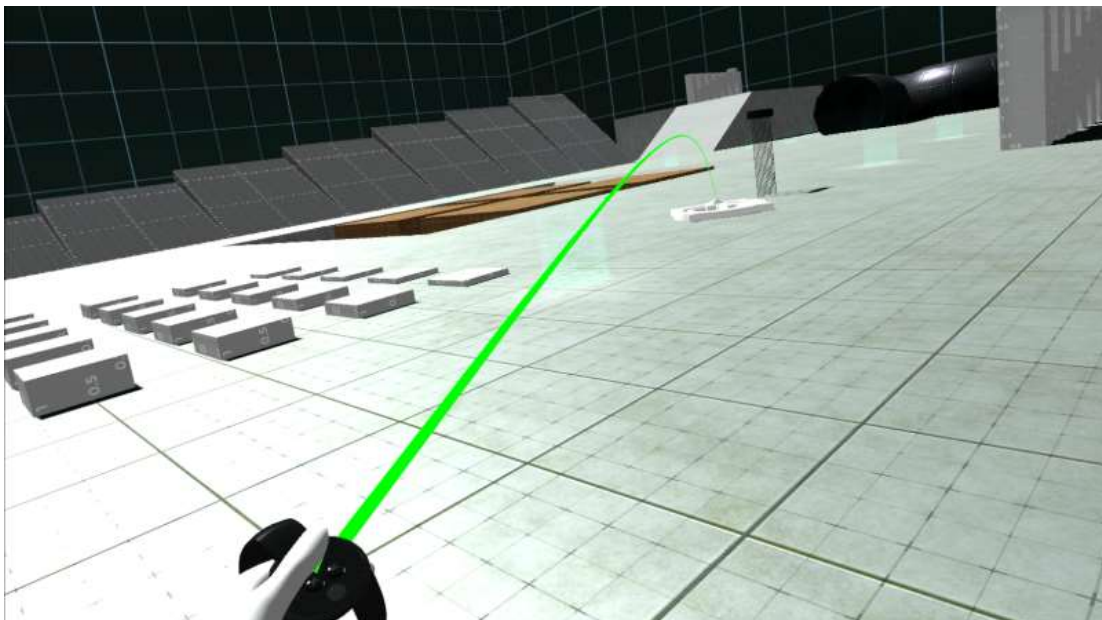
Vesielementti oli tarkoitus tuoda myös suunniteltuun isompaan metsäalueeseen, mutta ei niin näkyvästi kuin saarialueessa. Vesi toteutettiin Unity Asset Storesta saatavalla LowPoly Water -työkalulla. Työkalu generoi aaltoilevaa vettä matkivan peliobjektin halutulla leveydellä ja korkeudella, ja koska kyseessä on low poly -tyylinen vesielementti, voi segmenttien leveyttä ja korkeutta myös säätää. Generoinnin jälkeen veden aaltoilun korkeutta, taajuutta sekä pituutta pystyy muuttamaan.

4.3 Liikkuminen

Liikkumismuodoksi sovellukseen valittiin teleportointi (teleporting), sillä teleportointi aiheuttaa huomattavasti vähemmän pahoinvointia pelaajassa verrattuna siihen, jos liikkuminen tapahtuisi ohjainten ohjaussauvoja (joystick) käyttäen.

Kaikki teleportointiin tarvittavat komponentit olivat valmiiksi Oculus Integration Packagessa. Komponentit ovat hyvin suunniteltu ja ne tarjoavat eri vaihtoehtoja esimerkiksi, että onko teleportointi täysin vapaata pelaajalle vai voiko pelaaja liikkua vain tiettyjen pisteiden välillä.

Sovelluksessa teleportoinnissa pelaaja työntää ohjaimen ohjaussauvaa ylöspäin sekä osoittaa ohjainta suuntaan, johon pelaaja haluaa liikkua. Sovelluksessa ohjaimesta lähtee näkyvä kaari sekä kaaren päässä on ympyröity nuoli (kuva 12), joka näyttää pelaajalle mihin kohtaan pelaaja olisi siirtymässä. Teleportoitavan matkan pituutta säädellään nostamalla ja laskemalla ohjainta, jolloin kaari liikkuu lähemmäs tai kauemmas pelaajasta. Kun pelaaja on tyytyväinen teleportoitavan matkan pituuteen ja sijaintiin, pelaajan täytyy painaa ohjaimen liipaisinnappulaa, jotta teleportointi tapahtuu.



Kuva 12. Teleportoinnin kaari ja nuoli osoitin (Oculus for Developers 2021f)

4.4 Aktiviteetit

Sovelluksen aktiviteettien oli tarkoitus olla luotuun peliympäristöön sopivia ja niiden tarkoitus oli testata virtuaalitodellisuuden tarjoamia erilaisia mahdollisuuksia kuntoutuksen apuvälineenä. Alun perin sovellukseen oli suunniteltu neljä eri aktiviteettia: tammenterhojen poiminta, nuotion rakennus ja sytytys, seinäkiipeilyä kallion seinämää myöten sekä jousiammunta.

Näistä aktiviteeteistä tammenterhojen poiminta sekä nuotion rakennus ja sytytys saatiin toteutettua sovellukseen. Seinäkiipeily aloitettiin, mutta sen teko jäi kesken projektin loppumisen vuoksi. Jousiammuntaan oli valmiina tarvittavat 3D-mallit, mutta muuten aktiviteetin toteutusta ei saatu aloitettua.

4.4.1 Tammenterhojen keräys

Tammenterhojen keräyksessä pelaajan on tarkoitus kerätä tammenterhoja koriin. Korit ja terhot sijaitsevat pelimaailmassa valmiiksi (kuva 13). Pelaaja pystyy kantamaan koria ja tammenterhojaan mukanaan sovelluksessa.



Kuva 13. Virtuaalitodellisuussovelluksen tammenterhot ja kori

Korissa on neljälle tammenterholle paikat, johon tammenterho siirtyy. Kun koriin laitetaan tammenterho ja pelaaja on päästänyt siitä irti, skriptien kautta tammenterho liikutetaan oikealle paikalle. Kun korissa on neljä tammenterhoa, koriin voi edelleen laittaa tammenterhoja. Loput koriin laitettut terhot menevät toisten terhojen sisään, jota pelaaja ei huomaa. Jos koriin laittaa tammenterhon, sitä ei voi enää ottaa sieltä pois. Koriin on luotu laskuri, joka laskee montako terhoa korissa on. Laskurin ulkoasu on hyvin mitätön ja laskuri on piilotettu pelaajalta.

4.4.2 Nuotion teko

Nuotio sytytyksessä on muutama vaihe, jotka pelaajan täytyy suorittaa. Ensiksi pelaajan pitää laittaa nuotioon viisi oksaa, joita löytyy nuotion läheisyydestä. Nuotiossa on oksan muotoinen, violetin värinen, hiukan läpinäkyvä ja korostettu indikaattori pelaajalle. Kun pelaaja sijoittaa oksan tarpeeksi lähelle indikaattoria ja päästää oksasta irti, koodin avulla oksa siirretään ennalta määriteltyyn paikkaan nuotiossa. Indikaattori myös häviää ja seuraava indikaattori ilmestyy näkyviin.

Kun pelaaja on sijoittanut kaikki viisi oksaa nuotioon, nuotio on mahdollista sytyttää. Nuotion sytytys tapahtuu kopsauttamalla kahta nuotion vieressä sijaitsevaa irtokiveä keskenään yhteen. Kun kivet osuvat toisiinsa, se aktivoi kipinäpartikkelin lisäksi tulen syttymiseen luodun koodin. Tulen syttymiseen on koodi muodostettu siten, että tuli ei välttämättä aktivoidu ensimmäisestä kopsautuksesta, mutta mahdollisuus kasvaa jokaisesta kopsautusyrityksestä. Kun kivien välillä tapahtuu törmäys, koodissa satunnaisesti valitaan numero 1-10 väliltä ja jos numero on 9 tai suurempi, tuli syttyy. Lisäksi tähän satunnaisesti valittuun numeroon summataan pelaajan yritysten määrä. Tämän avulla eliminoidaan mahdollisuus, että pelaaja ei ikinä saisi tulta sytytettyä, vaikka mahdollisuus tähän onkin hyvin pieni.

Tulen syttyessä käynnistyvät tuliefektin partikkelit sekä valo, joilla tuliefekti luodaan. Partikkeleina tuliefektiin luodaan liekit, hohkaa, kipinöitä ja savua. Tuliefektille ei ole loppumisajankaa.

Nuotion teosta jäi puuttumaan ominaisuus, että tuli sammuisi tietyn ajan jälkeen, ellei nuotioon lisää puita. S sammumisen jälkeen nuotio oli tarkoitus voida rakentaa alusta asti ja sytyttää uudestaan.

4.5 Testaus

Virtuaalitodellisuussovellusta testattiin kehityksen aikana muutamilla eri ikäisillä henkilöillä sekä fysioterapiaoiskelijaryhmällä osana heidän Use of technology and supporting functioning of physiotherapy -kurssia.

Suurin osa virtuaalitodellisuussovellusta testanneista henkilöistä eivät olleet ennen testausta kokeilleet minkäänlaista virtuaalitodellisuuslaitteistoa. Testauksissa kävi ilmi, että vanhemmille henkilöille virtuaalitodellisuuslaitteen käyttö oli selkeästi hankalaa. Heidän oli hankala ymmärtää, kuinka laite ja sen ohjaimet toimivat, sekä mitä laitteella voi tehdä, esimerkiksi että ympäristöä voi katsella kääntämällä päätä.

Lähes kaikille testaajille Oculus Questin käyttö tuotti jonkinlaisia vaikeuksia jossain vaiheessa testausta. Ohjainten käyttö vaikutti tuottavan eniten vaikeuksia. Monet testaajat

unohtivat nopeasti, kuinka esimerkiksi liikkuminen tapahtui ja kuinka käteen otettiin esineitä suullisesta ohjeistuksesta huolimatta.

Fysioterapiaopiskelijaryhmän kanssa tuli hyvää keskustelua siitä, mikä olisi tällaisen virtuaalitodellisuussovelluksen oikea käyttäjäryhmä. Monien opiskelijoiden mielestä on tärkeää arvioida käyttäjän fyysiset ja kognitiiviset valmiudet käyttää virtuaalitodellisuussovellusta sekä -laitetta. Ylipäätään laitteen käyttöturvallisuus herätti paljon keskustelua. Opiskelijoilta tuli myös hyvää palautetta sovelluksen kehitykseen, esimerkiksi sovelluksessa olisi hyvä voida säätää vaikeusastetta ja sen tulisi sisältää enemmän pelillistäviä elementtejä.

5 Yhteenveto ja pohdinta

Työssä oli tavoitteena toteuttaa kuntouttava virtuaalitodellisuussovellus Konenäkö-hankkeelle. Toimiva sovellus saatiin luotua. Hankkeessa oltiin hyvin tyytyväisiä sovellukseen, vaikka projektin loppumisen vuoksi sovelluksesta jäi joitakin ominaisuuksia toteutumatta.

Sovelluksessa on potentiaalia jatkokehitykseen tulevaisuuden kannalta. Sovellukseen voi lisätä uusia erilaisia aktiviteetteja sekä suunnitellut aktiviteetit, jotka jäivät ajan puutteen vuoksi toteutumatta. Sovellusta voi myös parannella muuttamalla sekä jatkokehittämällä nykyisiä aktiviteetteja. Sovellusta voisi myös kehittää siten, että se olisi säädetävissä käyttäjän tasoon nähden, esimerkiksi aktiviteetteja olisi mahdollista suorittaa täysin istualtaan ja ilman kurkottelua. Sovelluksessa olisi myös hyvä olla helposti nähtävissä, mitä ohjainten eri nappulat tekevät.

Konenäkö-hanke sai sovelluksen kehityksen ja testausten aikana ymmärrystä virtuaalitodellisuudesta sekä siitä, kuinka sitä voitaisiin käyttää sote-alalla ja kuntoutuksen tukena. Testauksista saatiin kattavasti palautetta käyttökokemuksien kautta, ja ideoita sovelluksen kehitykselle. Tulevaisuudessa sovellusta olisi silti hyvä testata lisää. Sovellusta olisi myös hyvä testata yhteistyössä terveysalan ammattilaisten kanssa, jos sovellusta haluttaisiin laajemmin käyttää kuntoutuksen tukena. Tällaisen yhteistyön avulla olisi mahdollista saavuttaa laajempi ymmärrys millaisissa tapauksissa virtuaalitodellisuuskuntoutusta voitaisiin käyttää, sekä virtuaalitodellisuuskuntoutuksen hyödyistä ja haitoista.

Lähteet

Alex 2018. Oculus Rift History – How it All Started. Viitattu 21.3.2021. Saatavissa <https://rif-tinfo.com/oculus-rift-history-how-it-all-started>

Google VR 2018. Degrees of freedom. Viitattu 15.3.2021. Saatavissa <https://developers.google.com/vr/discover/degrees-of-freedom>

Kipper G. & Rampolla J. 2012. Augmented Reality: An Emerging Technologies Guide to AR. Saint Louis: Elsevier Science & Technology Books.

Linowes, J. 2015. Unity Virtual Reality Projects: Explore the world of virtual reality by building immersive and fun VR projects using Unity 3D. Community experience distilled. Birmingham: Packt Publishing

Mealy, P. 2018. Virtual & augmented reality for dummies. For dummies. Hoboken, New Jersey: John Wiley and Sons Inc.

Mechatech 2021a. How do common virtual reality tracking systems work? Viitattu 23.3.2021. Saatavissa <https://www.mechatech.co.uk/journal/how-do-common-virtual-reality-tracking-systems-work>

Mechatech 2021b. Why will inside-out tracking be the future of VR? Viitattu 23.3.2021. Saatavissa <https://www.mechatech.co.uk/journal/why-will-inside-out-tracking-be-the-future-of-vr>

Moon Studios 2021. Screenshots. Viitattu 28.5.2021. Saatavissa <https://www.orithegame.com/blind-forest/>

Nieminen, I. 2016. Uudesta Pokemon-pelistä jättihitti – Nintendon markkina-arvo hyppäsi yli 6 miljardia euroa. Yle. Viitattu 26.2.2021. Saatavissa <https://yle.fi/uutiset/3-9017208>

Oculus 2019. Play Rift content on Quest with Oculus Link, available now in beta! Oculus-blogi 18.11.2019. Viitattu 23.3.2021. Saatavissa <https://www.oculus.com/blog/play-rift-content-on-quest-with-oculus-link-available-now-in-beta/>

Oculus 2021. Tutustu Oculus Questiin. Viitattu 22.3.2021. Saatavissa <https://www.oculus.com/quest/features/>

Oculus for Developers 2021a. Oculus Device Specifications. Viitattu 22.3.2021. Saatavissa <https://developer.oculus.com/learn/oculus-device-specs/>

- Oculus for Developers 2021b. Understand Oculus Integration Package Components. Viitattu 24.3.2021. Saatavissa <https://developer.oculus.com/documentation/unity/unity-utilities-overview/>
- Oculus for Developers 2021c. Import Oculus Integration Package. Viitattu 24.3.2021. Saatavissa <https://developer.oculus.com/documentation/unity/unity-import/>
- Oculus for Developers 2021d. Configure Unity Settings. Viitattu 24.3.2021. Saatavissa <https://developer.oculus.com/documentation/unity/unity-conf-settings/>
- Oculus for Developers 2021e. Enable Device for Development and Testing. Viitattu 24.3.2021. Saatavissa <https://developer.oculus.com/documentation/unity/unity-enable-device/>
- Oculus for Developers 2021f. Locomotion Sample Scene. Viitattu 17.8.2021. Saatavissa <https://developer.oculus.com/documentation/unity/unity-sf-locomotion/>
- Rudnäs, N. 2019. Mitä on eXtended Reality (laajennettu todellisuus)? Seamk. Viitattu 26.2.2021. Saatavissa <https://lehti.seamk.fi/alykkaat-ja-energiatehokkaat-jarjestelmat/mita-on-extended-reality-laajennettu-todellisuus/>
- Sanastokeskus TSK ry 2017a. Tietotekniikan termitalkoot. Viitattu 26.2.2021. Saatavissa http://www.tsk.fi/tsk/termitalkoot/fi/haku-266.html?page=get_id&id=ID483&vocabulary_code=TSKTT
- Sanastokeskus TSK ry 2017b. Tietotekniikan termitalkoot. Viitattu 26.2.2021. Saatavissa http://www.tsk.fi/tsk/termitalkoot/fi/haku-266.html?page=get_id&id=ID105&vocabulary_code=TSKTT
- Sanastokeskus TSK ry 2017c. Tietotekniikan termitalkoot. Viitattu 26.2.2021. Saatavissa http://www.tsk.fi/tsk/termitalkoot/fi/haku-266.html?page=get_id&id=ID463&vocabulary_code=TSKTT
- Sinicki, A. 2021. What is Unity? Everything you need to know. Android Authority. Viitattu 20.5.2021. Saatavissa <https://www.androidauthority.com/what-is-unity-1131558/>
- Takala 2017. Virtuaaliodellisuus tuo uusia työvälineitä terveydenhoitoon. Lääketieteellinen aikakauskirja Duodecim. Viitattu 15.4.2021. Saatavissa <https://www.duodecimlehti.fi/duo13741>
- Unity 2013. Materials – Unity Official Tutorials. Youtube-video. Viitattu 25.5.2021. Saatavissa <https://youtu.be/IFIXvDZezBQ>

Unity 2021. Quick guide to the Unity Asset Store. Viitattu 21.5.2021. Saatavissa <https://unity3d.com/quick-guide-to-unity-asset-store>

Unity Documentation 2021a. Android. Viitattu 5.4.2021. Saatavissa <https://docs.unity3d.com/Manual/android.html>

Unity Documentation 2021b. Unity's Interface. Viitattu 12.5.2021. Saatavissa <https://docs.unity3d.com/Manual/UsingTheEditor.html>

Unity Documentation 2021c. The Toolbar. Viitattu 18.5.2021 Saatavissa <https://docs.unity3d.com/Manual/Toolbar.html>

Unity Documentation 2021d. The Hierarchy window. Viitattu 20.5.2021 Saatavissa <https://docs.unity3d.com/Manual/Hierarchy.html>

Unity Documentation 2021e. The Game view. Viitattu 21.5.2021 Saatavissa <https://docs.unity3d.com/Manual/GameView.html>

Unity Documentation 2021f. The Scene view. Viitattu 21.5.2021 Saatavissa <https://docs.unity3d.com/Manual/UsingTheSceneView.html>

Unity Documentation 2021g. Scene view control bar. Viitattu 21.5.2021 Saatavissa <https://docs.unity3d.com/Manual/ViewModes.html>

Unity Documentation 2021h. The Inspector window. Viitattu 22.5.2021 Saatavissa <https://docs.unity3d.com/Manual/UsingTheInspector.html>

Unity Documentation 2021i. The Project Window. Viitattu 22.5.2021. Saatavissa <https://docs.unity3d.com/Manual/ProjectView.html>

Unity Documentation 2021j. The status bar. Viitattu 22.5.2021. Saatavissa <https://docs.unity3d.com/Manual/StatusBar.html>

Unity Documentation 2021k. Gameobjects. Viitattu 24.5.2021. Saatavissa <https://docs.unity3d.com/Manual/GameObjects.html>

Unity Documentation 2021l. Using Components. Viitattu 24.5.2021. Saatavissa <https://docs.unity3d.com/Manual/UsingComponents.html>

Unity Documentation 2021m. Deactivating GameObjects. Viitattu 25.5.2021. Saatavissa <https://docs.unity3d.com/Manual/DeactivatingGameObjects.html>

Unity Documentation 2021n. Colliders. Viitattu 25.5.2021. Saatavissa <https://docs.unity3d.com/Manual/CollidersOverview.html>

Unity Documentation 2021o. Physic Material. Viitattu 25.5.2021. Saatavissa

<https://docs.unity3d.com/Manual/class-PhysicMaterial.html>

Unity Documentation 2021p. Rigidbody overview. Viitattu 25.5.2021. Saatavissa

<https://docs.unity3d.com/Manual/RigidbodyOverview.html>

Unity Documentation 2021q. Scenes. Viitattu 25.5.2021. Saatavissa

<https://docs.unity3d.com/Manual/CreatingScenes.html>

Unity Documentation 2021r. Creating and Using Scripts. Viitattu 25.5.2021. Saatavissa

<https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>

Unity Documentation 2021s. Particle System. Viitattu 26.5.2021. Saatavissa

<https://docs.unity3d.com/Manual/class-ParticleSystem.html>

Veative 2020. Degrees of Freedom... 3DoF vs 6DoF. Viitattu 15.3.2021. Saatavissa

<https://www.veative.com/blog/degrees-of-freedom-3dof-vs-6dof/>

