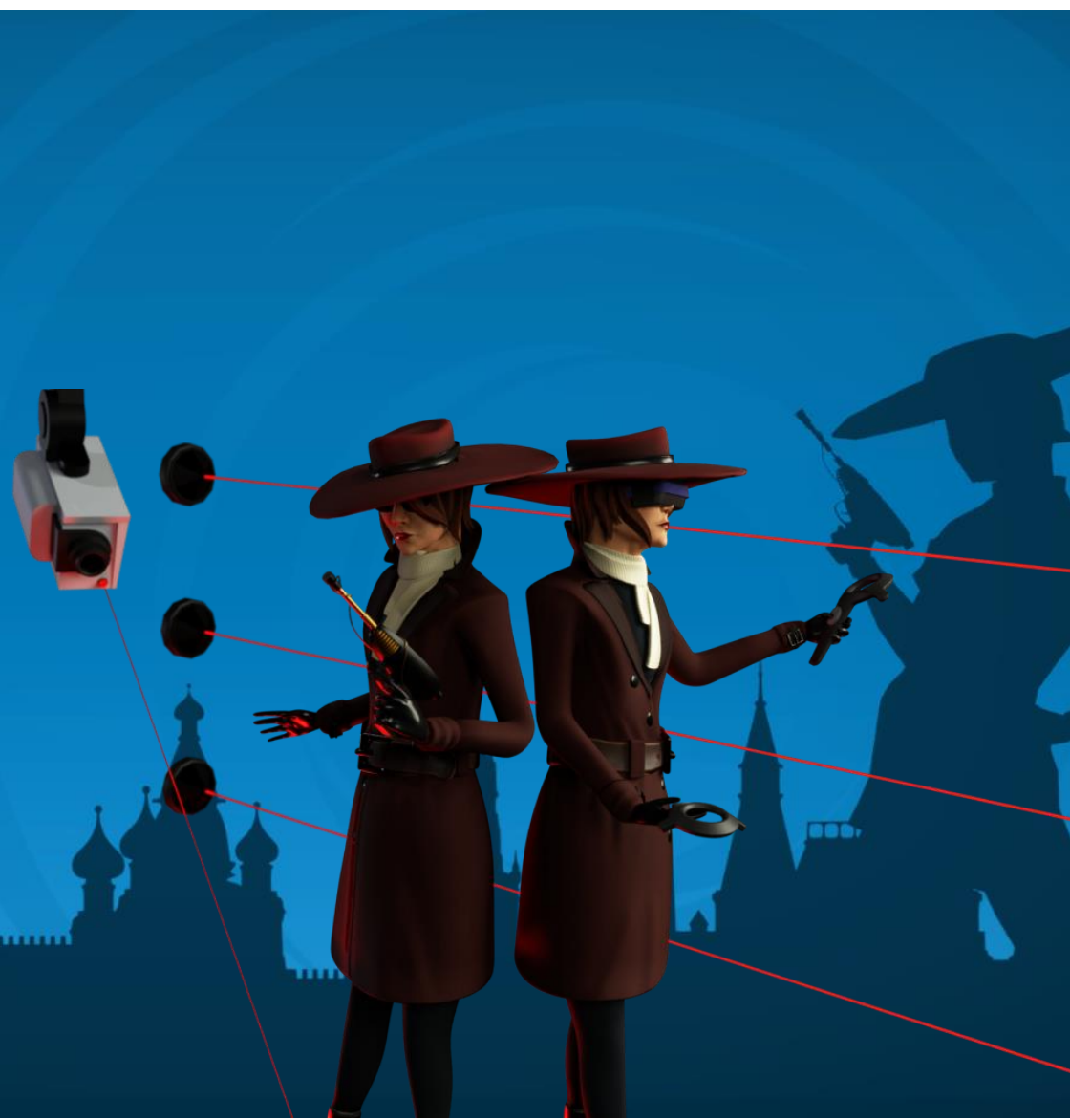


Mikko Juutinen

Videopelin muuntaminen

yhteensopivaksi virtuaalitodellisuuteen



Tradenomi
Tietojenkäsittely
Syksy 2019



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä: Juutinen Mikko

Työn nimi: Videopelin muuntaminen yhteensopivaksi virtuaalitodellisuuteen

Tutkintonimike: Tradenomi, tietojenkäsittely

Asiasanat: pelinkehitys, pelien siirtäminen, virtuaalitodellisuus

Tässä opinnäytetyössä käydään läpi virtuaalitodellisuuspelin tekemisen haasteet ja huomiokohdat, kun peli valmistetaan olemassa olevan peliin pohjautuen.

Teoriaosuudessa perehdytään lyhyesti virtuaalitodellisuuden historiaan sekä nykyhetken etenkin saatavilla olevien laitteiden osalta. Virtuaalitodellisuus kehittämistä pohjustetaan käymällä läpi Unity-käyttöympäristö ja sen tarjoamat työkalut virtuaalipelinkehityksen suhteen. Olennaisena osuutena työssä listataan suurimmat merkittävät huomiokohdat lisättäessä virtuaalitodellisuusominaisuuksia olemassa olevan pelin päälle.

Projektiosuus tehtiin osana tamperelaisen Catland-yrityksen Spy Who Shrank Me -videopelin kehitystä. Projektissa valmis PC-alustan peli muunnettiin työharjoittelun aikana täydeksi virtuaalipelikäännökseksi pelattavaksi useimmilla saatavilla virtuaalitodellisuuslaitteilla. Prosessista on kirjoitettu ylös useimmat merkittävät huomiokohdat, joihin törmäsin projektia tehdessä. Tärkeitä kohtia olivat muun muassa pelaajan liikkumisen toteutus eri tavoin sekä keinot, joilla pelaajan syöte erilaisiin käyttöliittymäelementteihin toteutettiin.

Abstract

Author: Juutinen Mikko

Title of the Publication: Transforming a Video Game into Virtual Reality Compatible Form

Degree Title: Bachelor of business administration

Keywords: game development, game porting, virtual reality

This thesis explores the challenges and considerations of making a virtual reality game when using an existing game as the base.

The theory section briefly introduces the history of virtual reality and the present, especially regarding the devices available. Virtual reality development will be outlined by going through the Unity platform and its tools for virtual game development. An essential part of the work is listing the major points of focus when adding virtual reality features to an existing game.

The project section was made as part of the development of Spy Who Shrank Me video game by Catland company in Tampere. In the project, the finished PC platform game was converted to a full virtual game for play on most available virtual reality devices. The most significant points of attention that were encountered while working on the project were recorded. These important points included, among other things, the implementation of the player's movement in different ways and the means by which the player's input in the various user interface elements was implemented.

Sisällys

1	Johdanto	1
2	Virtuaalitodellisuus.....	2
2.1	Historia	2
2.2	Laitteisto.....	4
2.2.1	Tietokonepohjaiset laitteet.....	4
2.2.2	Itsetoimivat laitteet.....	5
2.2.3	Mobiililaitteet.....	5
2.3	Käyttökohteet.....	5
3	Unity-käyttöympäristö.....	7
3.1	Ohjelmointi.....	7
3.2	Editori	7
3.3	VR-tuki	8
3.3.1	Asetukset.....	8
3.3.2	OpenVR.....	10
3.3.3	Oculus Integration for Unity.....	11
4	Huomioitavat komponentit	12
4.1	Käyttäjäsyoite.....	12
4.2	Liikkuminen	12
4.3	Vaikuttaminen ympäristöön.....	13
4.4	Käyttöliittymä	14
5	Projekti.....	15
5.1	Alustus	16
5.2	Käyttäjäsyoite	18
5.3	Liikkuminen	18
5.4	Pelaajan työkalut.....	20
5.5	Valikot.....	21
5.6	Heijastusnäyttö	23
6	Pohdinta	25
	Lähteet	26

Kuvalähteet.....	28
Liitteet.....	29

Symboliluettelo

SDK	Ohjelmiston kehityspakkaus, sisältää kaiken tarvittavan ohjelmiston rakentamiseen tietylle alustalle
Skene	Asetelma, kokoelma asioita, joista koostuu pelaajan pelimaailma
VR	Virtuaalitodellisuus
VRTK	Unity-alustalle tuotettu paketti, joka koostuu useista tavanomaisesti virtuaalipeleissä käytetyistä ominaisuuksista
Wrapper	Koodi, jonka tarkoitus on mahdollistaa ohjelmiston käyttö toisen ohjelmointikielen kanssa

1 Johdanto

Opinnäytetyön tarkoituksena on selvittää ja kuvailla, kuinka olemassa olevan tavanomaiselle pc-alustalle toteutetun peliprojektin voi muuntaa pelattavaksi useimmilla kaupallisilla virtuaalitodellisuuslaitteilla. Tavoitteena on kirjata tämän prosessin eri vaiheet hyvin ohjelmointiläheisestä näkökulmasta.

Työ on toteutettu pääosin Catland Oy:n palveluksessa, jonka aikana peliprojektia hallinnoi projektijohtaja Tomi Toikka. Työtehtäviini kuului virtuaalitodellisuusalustan suunnittelutyö ja toteutus. Projektin aikana toimin päävastuualueellani Unity-pelimoottorin päälle rakennettavan virtuaalitodellisuusalustan vastaavana projektijohtajana. Peliprojektin virtuaalitodellisuusalusta oli suunniteltu toimivaksi HTC Vive, Oculus Rift ja Microsoft Mixed Reality -alustoilla. Projektin loppupäässä tuimme myös Valve Index -virtuaalitodellisuuslaseja (mm. sormien seuranta Index -ohjaimien avulla).

2 Virtuaalitodellisuus

Virtuaalitodellisuus tarkoittaa täydellistä immersiota tavoittelevaa teknologiaa. Teknologian tavoitteena on mahdollistaa aitoa vastaavan sisällön lähettämisen käyttäjän aisteille. Keinoihin tällä hetkellä pääasiassa kuuluvat tiedon saaminen virtuaalisesta maailmasta näön ja kuulon avulla käytettäessä tarkoitukseen kehitettyä näyttölaitetta, sekä kyky vaikuttaa takaisin virtuaalimaailmaan erityisten ohjaimien avustuksella. [1.]

2.1 Historia

Tarve tehdä virtuaalitodellisuus muodostui aikoinaan kysymyksestä, kuinka on mahdollista tehdä ikkuna toiseen maailmaan, jossa ihminen voi nähdä, kuulla ja vaikuttaa samankaltaisesti kuin todellisuudessa.

Ensimmäinen yritys toteuttaa ajatus oli vuonna 1962 valmistunut Sensorama (kuva 1). Laitteen tarkoituksena oli näyttää värillistä stereovideota samaan aikaan esitettyjen ääni-, haju-, tuuli- ja värinävaikutteiden kanssa. Ensimmäinen versio päähän kiinnitettävästä, katsetta seuraavasta virtuaalitodellisuuslaitteesta puolestaan oli ”The Sword of Damocles” -laite. [2.]

Ennen vuotta 1990 virtuaalitodellisuusteknologia oli suuresti konsepteja tai sotilasteknologiaa. 1991 valmistui Sega VR, lisälaite Genesis-konsolille. Vaikka laitetta ei ikinä julkaistu markkinoille turvallisuushuolien vuoksi, oli tämä ensimmäisiä pelikäyttöön kehitettyjä virtuaalilaitteita. Kehityksessä ei näkynyt merkittäviä hyppyjä kaupallisella puolella ennen vuotta 2010, jolloin julkistettiin ensimmäinen nykyaikainen kaupallisen laitteen prototyyppi Oculus Rift. Tästä eteenpäin laitteiden kehitys lähti jyrkkään nousuun, kun markkinoille alkoi saapua useita kilpailevia laitteita. [3.]



Kuva 1. Sensorama-laite [1.]

2.2 Laitteisto

Virtuaalitodellisuuden kokemiseen tarvitaan erityistä laitteistoa. Vähimmäisvaatimuksena on päähän kiinnitettävä pään pyörimistä tunnistava näyttölaite, mutta monipuolisemmissa laitteistokokonaisuuksissa on myös mukana käsien sijaintia seuraavat käsiohjaimet sekä mahdollisesti muita sijaintia seuraavia paikannuslaitteita. Lähes poikkeuksetta kaikki korkeamman laadun virtuaalitodellisuuslaitteistot vaativat erillisen, keskimääräistä tehokkaamman tietokoneen tuottamaan laitteiden käyttöön vaatiman kuvasisällön tason.

2.2.1 Tietokonepohjaiset laitteet

HTC Vive on HTC:n ja Valven kehittämä virtuaalitodellisuusjärjestelmä PC:lle käyttäen Valven OpenVR-ohjelmistoalustaa. Laite oli yksi ensimmäisistä tehokkaammista virtuaalitodellisuuslaitteista. HTC Vive käyttää monista muista laitteista poiketen sisältä ulos tyylistä paikannustekniikkaa mahdollistaen kilpailijoitaan huomattavasti suuremman liikkumatilan. [4.]

Oculus Rift S on Facebook-yhtiön kehittämä laitteisto käyttäen omaa Oculus-alustansa sekä tarvittaessa myös SteamVR-alustaa. Rift S on jatkaja Oculusin alkuperäiselle laitteelle. Etuna laitteella on muun muassa HTC Viveä parempi kuvantarkkuus. [4.]

Windows Mixed Reality on Microsoftin vastine Oculus- ja SteamVR-alustoille. Suuri osa halvemmista virtuaalitodellisuuslaitteista käyttää tätä Microsoftin yhteistyösopimusten kautta. Merkittäviä valmistajia ovat muun muassa Acer, Dell ja Samsung. Täten Windows Mixed Reality on usein halvin vaihtoehto. [4.]

Valve Index on Valven kehittämä laitteisto. Se toimii hyvin samankaltaisesti ja osittain samoja osia käyttäen kuin edeltäjänsä HTC Vive, mutta on paranneltu kaikilla osa-alueilla, kuten kuvatarkkuudessa ja liiketunnistuksen tarkkuudessa sekä alueessa.

Playstation VR on Sonyn oma laitteisto käytettäväksi Playstation 4 -pelikonsolilla. Toimintatavaltaan se on hyvin samankaltainen Windows Mixed Reality -laitteiden kanssa käyttäessään ulkoisissaan -tyylillä toimivaa paikannusjärjestelmää. Tämän lisäksi ottaen huomioon Playstation 4 -pelikonsolin tehorojoitukset Playstation Virtual Reality -laite ei ole sovelias nopeitempisiin tai pelimaailmassa enemmän liikkumista vaativiin peleihin.

2.2.2 Itsetoimivat laitteet

Oculuksella on myös kaksi erilaista itsekseen toimivaa virtuaalilaitetta. Oculus Go on virtuaalitodellisuuslaite joka, uhraa tehoa ja ominaisuuksia saadakseen halvemman hinnan. Oculus Quest puolestaan on vastaavalla tekniikalla toteutettu, mutta huomattavasti tehokkaampi laite, joka kykenee hyödyntämään käsiohjaimia ja joitakin vaativampia pelejä. [4.]

Itsetoimivien laitteiden rajoituksena tällä hetkellä toimii niiden sisältämän prosessointiyksikön teho, joka ei pysty olemaan yhtä tehokas kuin vastaavassa erillisen tietokoneen vaativassa laitteessa.

2.2.3 Mobiililaitteet

Virtuaalitodellisuus on myös saatavilla useimmilla uusilla älypuhelimilla. Puhelimen näytölle piirrettävä kuva on muotoiltu siten, että sillä saavutetaan erityisen telineen avulla kasvojen eteen kiinnitettyinä monimutkaisempia laitteita vastaava virtuaalinäkymä, vaikka käytössä ei ole tarkoitukseen mukailtuja näyttöjä eikä linssejä. Kuitenkin tästä syystä kuvan laatu on huomattavasti huonompi kuin tarkoitukseen kehitetyillä laitteilla. Liikkeentunnistus rajoittuu täyden käsiseurannan sekä pään sijainti- ja pyörimisseurannan sijaan vain rajalliseen pään pyörimisen seuraamiseen.

2.3 Käyttökohteet

Virtuaalitodellisuutta voidaan hyödyntää jo nykyisellä teknologialla hyvin monipuolisesti. Tavalliselle kuluttajalle median kautta tunnetuin käyttökohde on videopelit. Useat 3d ympäristöön sijoittuvat ensimmäisen persoonan tavanomaiset videopelikonseptit sopivat hyvin suoraan toteutettaviksi VR-ympäristöön. Pelien kanssa samanlaiseen viihdekäyttöön on myös kehitetty kolmiulotteisia tai paljon kolmiulotteisilta vaikuttavia videoita, joita on mahdollista käyttää paljon pelejä pienemmällä kynnyksellä.

Teollisuuskäyttöön puolestaan on olemassa paljon käyttökohteita, joissa virtuaalitodellisuudella voidaan vähentää tai yksinkertaistaa työvaiheita. Esimerkiksi tuotteiden suunnittelun hoitaminen

kolmiulotteisessa ympäristössä, jota voi tarkastella VR-laitteilla, voi huomattavasti vähentää oikean maailman testikappaleiden hankintatarvetta [5.]

3 Unity-käyttöympäristö

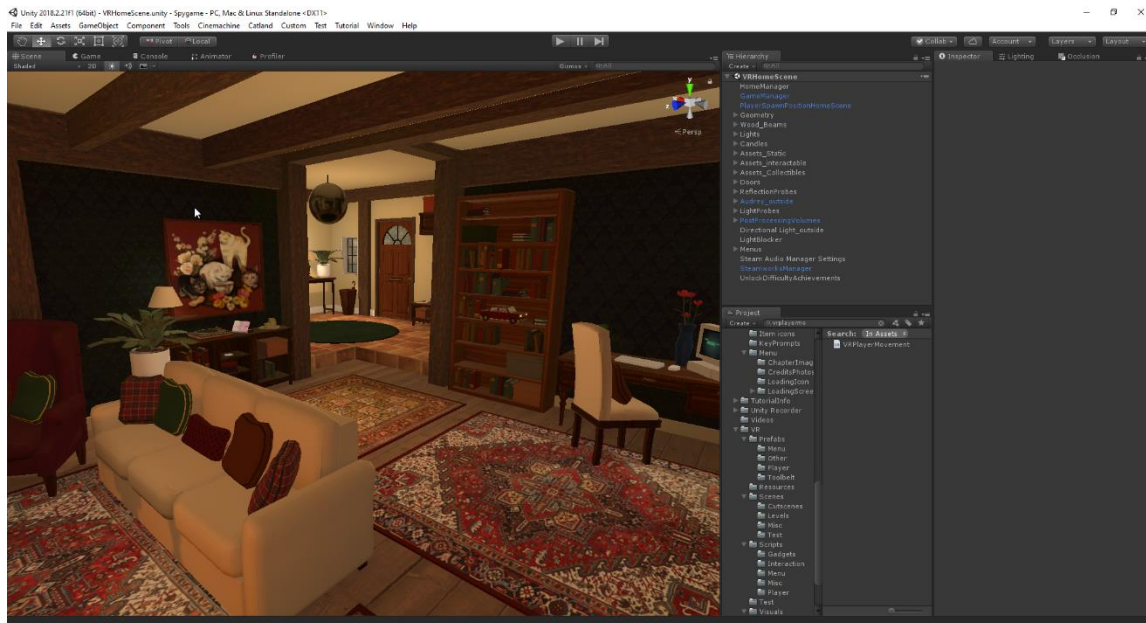
Unity on monikäyttöinen 2 ja 3-ulotteisten videopelien kehittämiseen tarkoitettu ohjelmisto eli pelimoottori, jota kehittää Unity Technologies. Ohjelmistolla on mahdollista valmistaa pelejä suoraan useimmille alustoille, esimerkiksi PC, PS4 tai Android. Unity on usein käytetty pelimoottori etenkin indiepelinkehityksen parissa sen valmiuden ja käytön helppouden vuoksi. Esimerkkinä moottorilla tuotetusta perinteisestä indiepelistä toimii Team Cherryn kaksiulotteinen seikkailu Hollow Knight sekä esimerkkinä virtuaalitodellisuuspelistä Beat Gamesin suosittu rytmipeli Beat Saber. [6.]

3.1 Ohjelmointi

Pelin skenet sisältävät peliobjekteja, jotka itsessään voivat sisältää koodinpätkiä antamaan toiminnallisuutta. Unityn tällä hetkellä pääasiassa tukema rakenne perustuu olio-ohjelmointiin sekä C#- että javascript-kielellä.

3.2 Editori

Editori sisältää kaikki työkalut kenttien tekemiseen ja pelilogiikan lisäämiseen. Editorinäkö koostuu muun muassa skenenäkymästä, pelin kameranäkymästä, objektien tarkkailuikkunasta sekä erinäisistä objektihierarkioista skenessä sekä tiedostoissa (kuva 2).



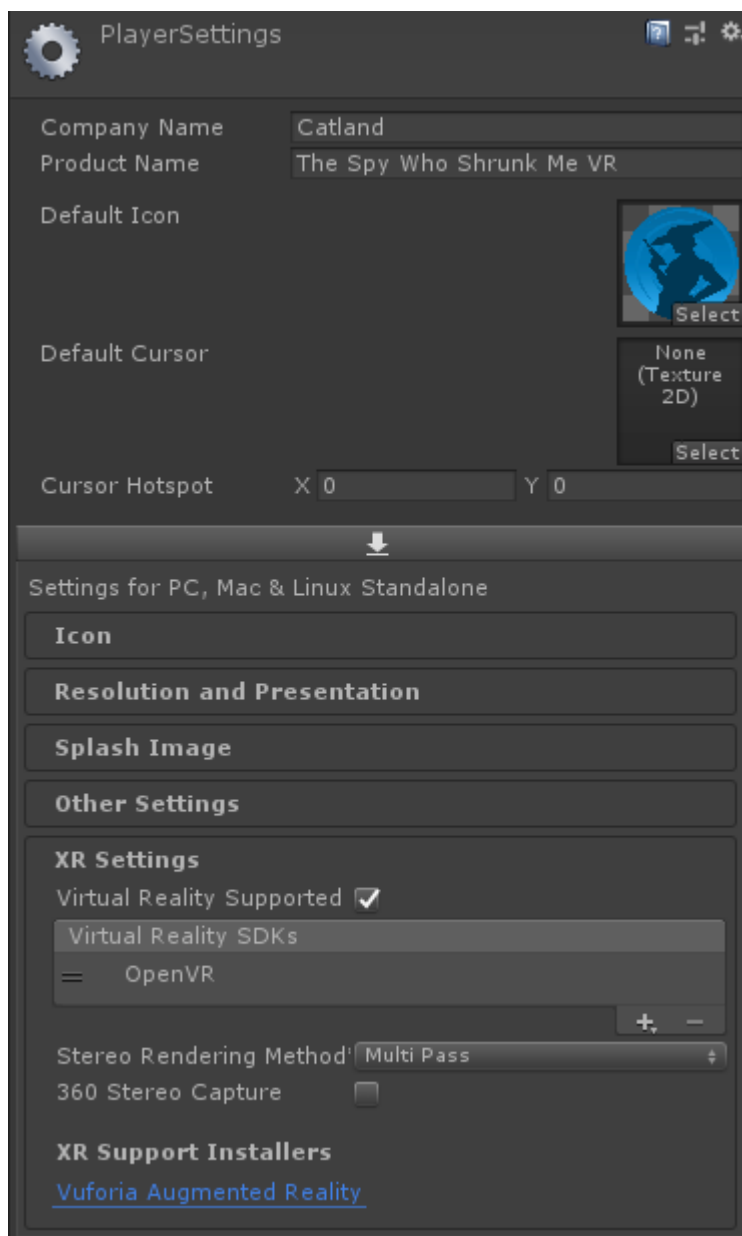
Kuva 2. Unity editorin tavanomainen skenenäkymä

3.3 VR-tuki

Unity itsessään ei anna täyttä tukea virtuaalitodellisuuslaitteille vaan tuen kuvata pelinäkömää tarvittavalla tavalla, että se voidaan piirtää oikein kahdelle VR-laitteen linsille.

3.3.1 Asetukset

Saatavilla olevat asetukset on kiteytetty Unityn Playerasetusten alavalikon XR Settings alle. Täältä on mahdollista kytkeä VR päälle sekä asettaa useita kehityspakkauksia tukemaan tiettyntyyppisiä VR-alustoja. Näitä ovat esimerkiksi lähes kaikkea tukeva OpenVR sekä Oculuksen oma sen omille laitteille (kuva 3). [7.]



Kuva 3. Unityn VR-asetukset

Lisäksi Unityn kameraobjektit tarjoavat jotain säätömahdollisuuksia kuvan ulostulon suhteen. Näitä voi muokata näyttölaitteelle piirtävän kameraobjektin asetuksista (kuva 4).



Kuva 4. Kameran asetukset

3.3.2 OpenVR

OpenVR on laajassa käytössä oleva Valven kehittämä virtuaalitodellisuuskirjasto, jonka tehtävänä on tehdä mahdollisimman monen erilaisen laitteen käyttäminen helpoksi yhteisen rajapinnan kautta.

Unity-ympäristössä OpenVR:n käyttö onnistuu SteamVR-paketin avulla, joka on vapaasti saatavissa Unity Asset Storesta. Tämä paketti sisältää OpenVR:n tarvittavat C#-kieliset wrapper-scriptit sekä muut Unityn ominaisuuksien kanssa vaikuttavat rajapinnat. Asennusprosessi asettaa tarvittavat asetukset sopiviksi ja muuntaa Unityn automaattisesti lähettämään ulos virtuaalilaitteille

sopivaa kuvaa. Lisäksi paketti sisältää valmiin objektihierarkiaesimerkin, joka on sellaisenaan mahdollista sulauttaa projektiin.

OpenVR on useimmiten käytetty virtuaalitukikirjasto Unity-alustalla sekä kehittyneisyyden vuoksi että sen yhdistäessä valtaosan laitteista saman järjestelmän alle vähentäen kehitysaikaa ja ylläpitoa. [8.]

3.3.3 Oculus Integration for Unity

Oculus Integration for Unity on Oculuksen oma virtuaalitukikirjasto Oculus Riftille, Questille ja Go:lle.

Unitylle tarkoitettu paketti sisältää kaiken tarvittavan näille laitteille kohdennetun ohjelman tekemiseen. Paketti sisältää muun muassa esimerkkiobjektihierarkian toiminnallisuuksineen, yleisimmin käytettyjen käsitoimintojen esimerkkitoiminnallisuudet sekä virheenkorjaustyökaluja.

Oculuksen virtuaalitukikirjastoa käytetään usein, jos halutaan joko rajoittaa peli vain Oculuksen laitteille tai halutaan lisätä alustan tuomat erityisominaisuudet mukaan, jos pelaaja pelaa peliä nimenomaan Oculus-laitteella. [9.]

4 Huomioitavat komponentit

Tavanomaiselle PC-alustalle tehdyssä projektissa on usein runsaasti järjestelmiä, jotka vastaavat esimerkiksi käyttäjän syötteestä sekä käyttöliittymän piirtämisestä. Nämä eivät yleensä toimi virtuaalitodellisuusympäristössä tarkoituksensa mukaisesti. Siten muuntaessa projektia virtuaalilaitteille useat järjestelmät täytyy poistaa tai rakentaa uusiksi. Uudelleen käytettävän materiaalin sekä tehtävän työn määrää määrittää hyvin se, kuinka hyvin erillään olemassa olevat järjestelmät on pidetty toisistaan.

Toinen työn laajuutta määrittävä tekijä on muuntoprosessin tavoiteltu laajuus. Vähimmillään VR-peli syntyy muuttamalla katsomistoiminto seuraamaan VR-pelaajan pään liikettä. Useat olemassa olevan pelin päälle tehdyt VR-pelit tähtäävät mahdollisimman helppoon toteutukseen. Kuitenkin voi olla tarpeellista tavoitella syvempää immersiota ja siten tähdätä tuottamaan kaikkia mahdollisia VR-laitteiden ominaisuuksia hyödyntävä peli. Tärkeää on tavoittaa sopiva tasapaino työn määrän ja laadun välillä.

4.1 Käyttäjäsyoite

Virtuaalitodellisuuslaitteet eivät pääasiassa käytä PC-alustalle tyypillisiä syöttölaitteita vaan erityisiä käsiohjaimia. Jos muuntoprojektissa on päädytty hyödyntämään käsiohjaimia, täytyy olemassa olevat syötejärjestelmät poistaa tai muokata tukemaan myös käsiohjaimien syötettä. Vaikka näiden erityisohjaimien syöte on suurimmalta osalta hyvin samankaltainen kuin tavanomaisen peliohjaimen, täytyy kaikessa syötteen vastaanottamisessa muun muassa huomioida, mistä kädestä tieto tulee.

4.2 Liikkuminen

Riippumatta pelilajista liikkuminen on yleensä yksi perustavimmista ominaisuuksista. Kuitenkin suurin osa virtuaalitodellisuuspeleistä sisältyy ensimmäisen persoonan pelilajin joukkoon. Näissä peleissä yleensä noudatetaan tavanomaista ihmismäistä liikkumismallia: eteen, sivuille, taakse, hyppy ja kyykky sekä kyky katsoa ympärilleen. Nämä ominaisuudet soveltuvat hyvin tämän het-

ken VR-laitteiden tarjoamaan mahdollisuuteen eläytyä nimenomaan pelaajan rooliin ensimmäisessä persoonassa. Tällöin jo olemassa oleva liikkuminen voidaan helposti säilyttää. Liikekomennot pysyvät samana ja missä aiemmin hahmo saattoi esimerkiksi kääntyä hiiren avulla, kääntyykin VR-laitteella päätä kääntämällä.

On myös mahdollista siirtyä kokonaan pois perinteisestä liikkumismallista, koska liikkuminen tällä tavalla virtuaalilaitteilla voi aiheuttaa osalle käyttäjistä pahoinvointia. Usein PC-version kaltainen liikkuminen on korvattava keinolla liikkua ilman liikettä eli teleportaatiolla. [10.]

Vaadittu laitteisto säätelee paljon mitä keinoja on käytettävissä. Pelkän näyttölaitteen sisältävällä laitteella ei usein voi liikkua lainkaan. Myös erillisillä käsiohjaimilla voidaan tehdä esimerkiksi käsien liikkeeseen perustuvaa liikkumista toisin kuin jos vaadittuna laitteistona on erityisten käsiohjaimien sijaan tavanomainen peliohjain. [11.]

On myös huomioitava, että virtuaalitodellisuudessa liikkuminen tapahtuu kahdessa eri tilassa. Sen lisäksi, että pelaajahahmon täytyy pystyä liikkumaan pelissä suuremmalla mittakaavalla, pelaajan on voitava myös liikkua virtuaalitodellisuuslaitteiden sallimissa puitteissa todellisuudessa. Myös tämän liikkeen täytyy siirtyä peliin. Tämä saattaa aiheuttaa tarpeen muuttaa jo olemassa olevaa pelaajan liikkumismekaniikkaa.

4.3 Vaikuttaminen ympäristöön

Pelimaailmassa saattaa usein olla runsaasti erinäisiä esineitä, joiden kanssa pelihahmon on kyettävä vaikuttamaan. Esimerkiksi ensimmäisen persoonan räiskintäpelissä hahmolla voi olla jatkuvasti käsissään ase. Ase on saatavilla näppäintä painamalla sekä muut tavanomaiset toiminnot, kuten aseensa lataaminen, toisesta näppäimestä. Lisäksi maailmassa saattaa sijaita lisäammuksia varten ammuslaatikoita, jotka ovat poimittavissa klikkaamalla. VR-ympäristöön suoraan käännettynä nämä eivät olisi suositeltavia mekaniikkoja. Aseen ilmestyminen käteen näppäintä painamalla on mahdollista, mutta pelaaja usein odottaa luontevampaa ja immersivistä keinoa, joka olisi mahdollista nimenomaan vain VR-laitteilla. Esimerkiksi aseensa käsien poimiminen vyötäröllä sijaitsevalta asevyöltä.

Selkein ero PC-alustaan siten tulee käsiin. Jokaista tarpeellista asiaa on voitava käyttää kummallakin kädellä. Usein PC-versiossa oleva klikkaaminen voi riittää ratkaisuksi. Esimerkiksi seinällä sijaitsevaa vipua voi kääntää oikealla kädenliikkeellä tai sitten klikata PC-alustamaisesti. Tärkeintä

on, että pelaaja voi tuntea tekevänsä toiminnon. Tällaisen immersion lisääminen kaikkialle voi olla suuri haaste, joten voi myös olla tärkeää käyttää useita erilaisia keinoja teknisten rajoitteiden puitteissa.

Toinen merkittävä ongelma muutokseen tulee kaksikäätisyydestä. VR-peleissä ei yleensä ole mahdollista mielekkäästi rajoittaa tai määrätä molempien käsien käyttöä. Tämä voi tilanteesta riippuen joko rikkoa tai parantaa pelattavuutta riippuen pelistä ja kuinka lähellä alkuperäistä peliä halutaan pysyä.

4.4 Käyttöliittymä

Kaikki pelaajan kameran pintaan piirrettävä ei ole sellaisenaan mahdollista piirtää virtuaalitodellisuusvisiirin näyttöihin. Esimerkiksi pelaajan heijastusnäytössä ollut elämämittari ja ammusmittari täytyy sulauttaa itse maailmaan erinäisten pienten näyttöjen muodossa. Yleisimpiä käytänteitä näiden toteuttamiseen ovat joko virtuaalisiin käsiin kiinnitetyt pienet näyttöpaneelit ja pelaajan katseen edessä leijuva pc-alustan heijastusnäyttöä jäljittelevä näyttö.

Heijastusnäytön elementtien lisäksi kaikki näkymät, jotka koostuvat yksinomaan kaksiulotteisista elementeistä, täytyy sulauttaa virtuaalimaailmaan. Yleinen ratkaisu tyyppillisen valikkonäkymän korvaamiselle on tehdä valikko kolmiulotteiseksi näytöksi joko tyhjäan tilaan tai muuhun ympäristöön. Luovempi ratkaisu voi esimerkiksi olla näyttöjen korvaaminen erinäisillä käytettävillä esineillä kuten vivuilla.

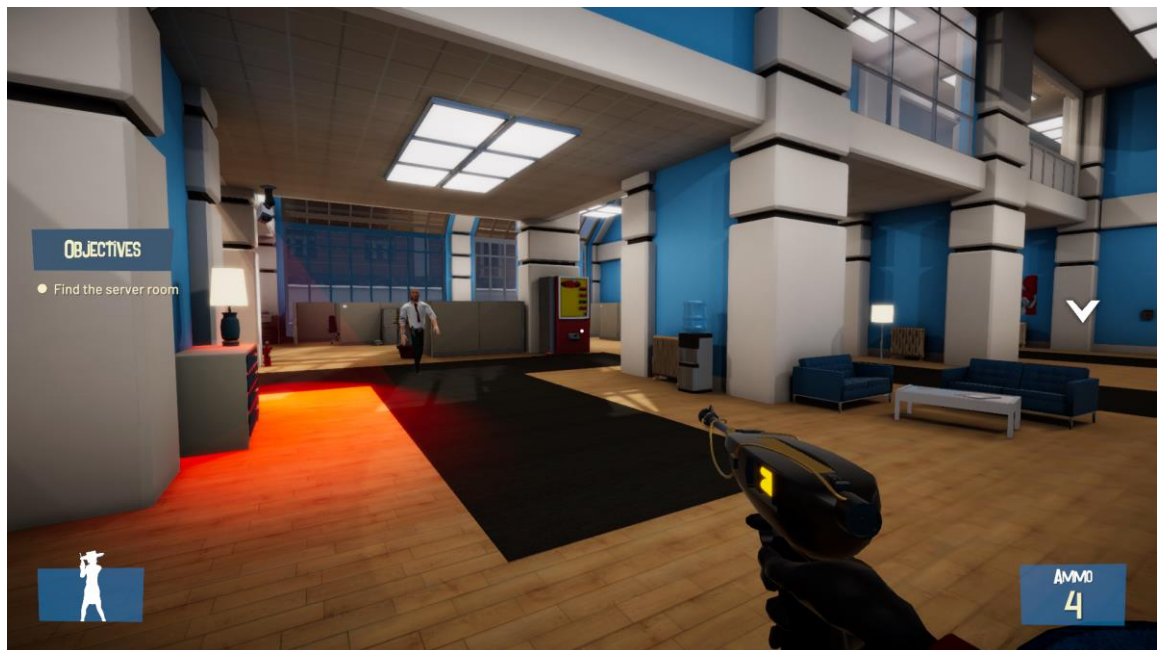
Erinäisten käyttöliittymäelementtien käyttämiseen ei virtuaalitodellisuudessa helposti ole mahdollista käyttää hiiren kaltaista osoitinlaitetta, koska se perustuu kaksiulotteiseen näkymään. Riippuen käytetyistä laitteista näppäimistön tai peliohjaimen käyttö on mahdollista tai käsiohjaimia käyttäessä osoittimena usein toimivat käsiin liitetyt sädetyyppiset osoittimet. Teknisesti tämän osoittimen toiminta usein emuloi tavanomaista hiiren osoitinta, jolloin saavutetaan suurin mahdollinen yhteensopivuus jo olemassa olevien valikkojärjestelmien kanssa.

5 Projekti

Projektin tarkoituksena oli toteuttaa mahdollisimman tehokas käännös kehityksessä olevasta Unity-moottorilla tehdystä pc-alustan pelistä virtuaalitodellisuuslustoille. Toimeksianto toteutettiin Catland yrityksen kehityksessä olleeseen Spy Who Shrank Me -peliin (kuva 5) kolmen kuukauden aikarajoitteen sisällä.

Peli itsessään oli jo alun perin teoreettisesti suunniteltu käännettäväksi virtuaalitodellisuuteen sitä kuitenkaan toteuttamatta. Vaatimuksina julkaisukelpoiseen tuotteeseen oli mahdollisuus pelata kaikkea jo olemassa olevaa sisältöä kaikilla PC-yhteensopivilla virtuaalitodellisuuslaitteilla. Pelissä jo olemassa olevan pelaajahahmon täytyi kyetä liikkumaan ja katsomaan maailmassa käyttäen virtuaalitodellisuusohjaimia ja -visiiriä. Myös kaikkien käytettävien esineiden täytyi olla käytettävissä vapaasti käsiä käyttäen. Näiden lisäksi kaikki täysin kaksiulotteinen sisältö piti sovittaa tai sulauttaa maailmaan kolmiulotteiseksi.

Täten ohjelmaan lisättiin rinnakkainen sovellusversio, joka sisälsi kaiken VR-erikoistuneen sisällön. Tämä sitten julkaistiin alkuperäisen pelin rinnalle lisäsisältömäisesti Steam-alustalle.



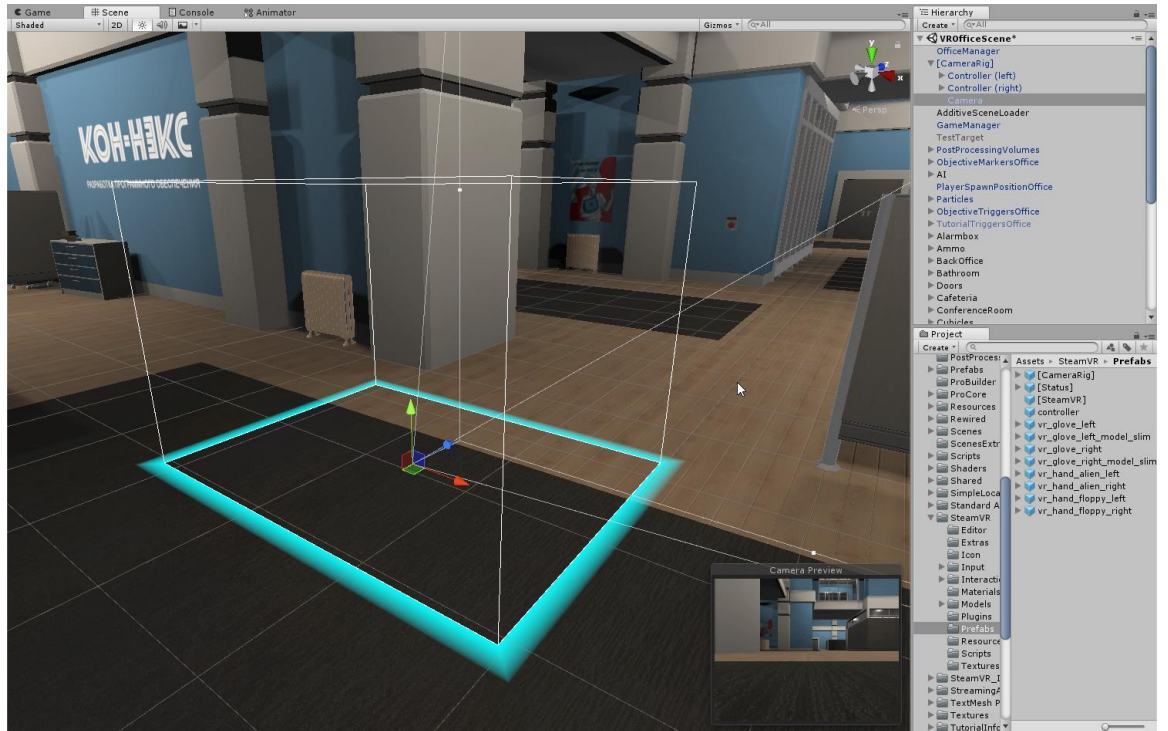
Kuva 5. Kuva pelin alkuperäisestä PC-versiosta

5.1 Alustus

Ensimmäinen vaihe käänösprosessissa oli huolellinen tutustuminen jo olemassa olevaan projektiin. Tällä tarkoitettiin sekä olemassa olevaan koodipohjaan tutustumista että yrityksessä aiemmin käytettyjen menetelmien ymmärtämistä, jotta saataisiin hahmotettua mahdollisimman hyvä yleiskuva tilanteesta sopivimpien ratkaisujen saavuttamiseksi. Seuraavaksi työhön sisältyi rakenteen ja komponenttien arviointi: montako järjestelmää oli riippuvaista vain pc-alustan vaatimista ominaisuuksista ja kuinka montaa eri järjestelmää voitaisiin käyttää suoraan ilman erityisiä muutoksia?

Varsinainen muokkausoperaatio alkoi täyttämällä VR-pelin minimivaatimukset mahdollisimman yksinkertaisesti eli näkeminen pelimaailmassa. Tähän sisältyi SteamVR-lisäosan asennus, joka antaisi projektille tarvittavan VR tuen sekä vanhan PC-version pelaajan poistaminen kokonaan. Tämä päädyttiin tekemään vanhan koodipohjan suhteellisen epäselvyyden vuoksi mahdollisimman puhtaalta pöydältä. Täten käytöstä poistettiin kaikki automaattisesti rikki menevät tai muuten käyttökelttomat komponentit, jotka olivat jollain tavoin sidoksissa pelaajaan. Tilalle sijoitettiin SteamVR-lisäosan vakiopelaaja, jolla pystyi näkemään jo olemassa olevassa pelimaailmassa, vaikkakin kaikki muut pelilliset ominaisuudet puuttuisivatkin (kuva 6).

Lisäksi tässä vaiheessa täytyi tehdä päätös, mitä muita vapaaehtoisia lisäosia projektiin käytettäisiin. Virtuaalipelaaja kokonaisuutena on usein hyvin samankaltainen monien muiden VR-pelien kesken, joten tähän löytyisi useita valmiita ratkaisuja, kuten esimerkiksi VRTK lisäosa. Kuitenkin projektissa päädyttiin olemaan käyttämättä muita lisäosia.



Kuva 6. SteamVR-lisäosan vakiopelaaja asetettuna kenttään

5.2 Käyttäjäsyoite

Projektin näppäinpainallusten vastaanotto hoidettiin pc-alustalla Rewired-järjestelmällä. Kaikki viittaukset tähän järjestelmään poistuivat automaattisesti uuden VRPlayer-luokan korvatesa vanhan Player-luokan. Rewired muutoin toimi täysin omana GameObjectinaan, jonka poistaminen skeneistä myös poisti toiminnallisuuden olemassaolon. Tilalle laitettava VR-laitteiden syöte puolestaan ei mene itse VRPlayer-luokan sisään, koska tässä ympäristössä itse pelaajan kädet eli VRHand-luokka hoitaa vastuun maailmaan vaikuttamisesta. [12.]

5.3 Liikkuminen

PC-versiota vastaava liikkuminen kaksiulotteisesti maan tasalla voitiin toteuttaa VR-versioon samankaltaisesti. Pelaajahahmon sijaan pelaajan pelialueen keskipiste liikkuu syötteiden mukaisesti. Vaikka hahmon liikkeen pohjana käytetään pelialuetta, otetaan laskutoimituksiin mukaan pelaajan kehon sijainti, joka ei välttämättä ole pelialueen keskellä. Tällä saavutetaan vastaava liikkuminen kuin PC-versiossa, mutta mahdollistaa pelaajan virtuaaliodellisuuslaitteen liikkeen siirtämisen peliin muun liikkumisen päälle. Pelialue ei kuitenkaan koskaan käänny samalla tavalla kuin PC-versiossa pelaajahahmo kääntyy hiirtä liikuttamalla. Siksi VR-version liikekontrollit ovat suhteellisia joko pelaajan katseeseen tai käden asentoon. Vaikka kääntymisen voi hoitaa kääntymällä todellisuudessa, se ei ole aina käytännöllistä laitteiden mahdollisten johtojen vuoksi. Siksi myös itse pelialuetta voi kääntää asteittain painalluksella (kuva 7).

Koska liikkuminen tällä tavalla virtuaalilaitteilla voi aiheuttaa joillekin pahoinvointia, peliin täytyi myös toteuttaa VR-peleille tyypillinen keino liikkua ilman liikettä eli teleportaatio (kuva 8). Kuitenkin pelissä teleportaatio oli jo yksi olennainen pelaajan työkalu, joten sen siirtäminen täysin tavanomaiseksi liikkumistavaksi poistaisi pelistä sisältöä ja mahdollisesti rikkoisi kenttäsuunnittelua. Täten tavallista liikkumista hoitava teleportaatio sisälsi kaksi tärkeää rajoitusta: sen suurin sallittu etäisyys oli lyhyt sekä lähtöpisteen ja määränpään välinen korkeusero oli rajattu.

Hyppytoiminto puolestaan toimi virtuaaliympäristössä hieman erilaisilla vaatimuksilla. Hyppiminen VR-laitteiden kanssa ei ole turvallista eikä pelaajalle voi antaa minkäänlaista palautetta esimerkiksi korkeammalle tasolle hypätessä. Lisäksi ainoasta kehon seurantapistestä, päästä, ei voi

luotettavasti laskea pelaajan aietta hypätä. Siten hyppy on mahdollista toteuttaa PC-versiota vastaavasti käyttämällä pelialuetta pelaajahahmon sijaan ja laittamalla sille hetkellistä voimaa ylöspäin.

Kyykkytoiminto puolestaan on mahdollista täysin VR-laitteen liikkeitunnistuksella. Projektissa pelaajahahmon korkeus on suoraan sidonnainen pelaajan pään sijaintiin tunnustustilassa. Siten kyykkyyyn meneminen suorittaa saman vaikutuksen kuin PC-version kyykkytoiminto. Kuitenkin tätä liikettä voidaan pitää myös epämiellyttävänä suorittaa, joten projektiin lisättiin mahdollisuus suorittaa sama vaikutus painikkeella. Koska pelaajahahmon kehoa ei voi pienentää pelaajan silti seistessä tunnistusalueella, suoritettiin vaikutus väliaikaisesti siirtämällä virtuaalisen pelialueen sijaintia maan pinnan alapuolelle, jolloin pelaajan pää jää kyykkytoiminnon vaatimalle korkeudelle maan pintaan nähden.

```
private Vector3 MovementFree(RelativeTo relativeTo)
{
    Vector2 trackpad = moveAction.GetAxis(mirrorMovementControls ? SteamVR_Input_Sources.RightHand : SteamVR_Input_Sources.LeftHand).Map(moveDeadzoneMin, moveDeadzoneMax, 0f, 1f);
    bool jump = jumpAction.GetStateDown(mirrorMovementControls ? SteamVR_Input_Sources.LeftHand : SteamVR_Input_Sources.RightHand);

    if (GameManager.Instance.IsPaused)
    {
        trackpad = Vector2.zero;
        jump = false;
    }

    Transform relativeTransform = null;
    switch (relativeTo)
    {
        case RelativeTo.Head:
            relativeTransform = VRPlayer.Instance.hmdTransform;
            break;
        case RelativeTo.Controller:
            relativeTransform = mirrorMovementControls ? VRPlayer.Instance.rightHand.hoverPointerTransform : VRPlayer.Instance.leftHand.hoverPointerTransform;
            break;
        default:
            break;
    }

    Vector3 relativeTransformForward = relativeTransform.forward;
    relativeTransformForward.y = 0f;
    Vector3 relativeTransformRight = relativeTransform.right;
    relativeTransformRight.y = 0f;

    if (IsGrounded() && jump)
    {
        ySpeed = jumpVelocity * (VRPlayer.Instance.isShrunken ? shrunkenSpeedMultiplier : 1);
    }

    Vector3 move = (relativeTransformForward.normalized * trackpad.y + relativeTransformRight.normalized * trackpad.x) * speed * (VRPlayer.Instance.isShrunken ? shrunkenSpeedMultiplier : 1);
    move *= Time.unscaledDeltaTime;
    return move;
}
```

Kuva 7. VR-pelaajan liikkumista hoitava funktio



Kuva 8. Teleportaatio-liikkumisvaihtoehto

5.4 Pelaajan työkalut

Pelaajan työkalut tai välineet olivat pelin keskeisin keino vaikuttaa maailmaan esimerkiksi tainuttamalla vartioita tehdäkseen liikkumisen helpommaksi tai pienentämällä pelaajan kokoa mahdollistaen uudenlaisia liikkumisvaihtoehtoja.

PC-alustalla nämä työkalut oli toteutettu hyvin samaan tapaan kuin lähes kaikissa markkinoilla olevissa ensimmäisen persoonan räiskintäpeleissä. Pelaaja piti kädessään yhtä välinettä kerrallaan vaihtaen eri vaihtoehtojen välillä, jotka on kaikki sidottu valmiisiin animaatioihin pelaajan näkökenttään. Vaihtoehdot olivat listattu ruudun alalaidassa.

VR-ympäristössä pelaaja ei enää voi olla sidottu toimimaan vain yhdellä kädellä eikä ole olemassa samaa ruutua, missä esittää erinäiset työkaluvaihtoehdot. Lisäksi välineet pitää pystyä ottamaan esille jostain maailmassa olevasta sijainnista. Siksi työkalut täytyi muuttaa omiksi esineikseen irti PC-alustan animaatioista ja näyttömalleista vastaamaan mitä tahansa muuta VR-käsillä vaikutettavaa esinettä, kuten esimerkiksi nostettavaa laatikkoa.

Koska näitä työkaluja ei mielellään voisi piilottaa PC-alustan tapaisesti niiden ollessa poissa käytöstä, luotiin ratkaisuksi työkaluvyö (kuva 9). Tältä pelaajan ruumiin edessä mukana liikkuvulta alueelta pystyi käytännöllisesti poimimaan ja sijoittamaan käytettäviä esineitä.



Kuva 9. Pelaajan työkalut työkaluvyöllä

5.5 Valikot

Projektissa valikot, niiden rakenteesta johtuen, jouduttiin rakentamaan uusiksi. Uudet valikot rakentuivat ulkonäön puolesta hyvin samalla tavalla kuin vanhat käyttäen Unity-moottorin olemassa olevia työkaluja. Kuitenkin valikkojen käyttämiseen tarvittavat järjestelmät täytyi rakentaa osittain uusiksi.

Käyttöliittymäelementtien käyttäminen Unityssä perustuu EventSystem-komponenttiin ja siihen sijoitettuihin BaseInputModule-luokasta periytyviin komponentteihin. BaseInputModulen perivät

luokat lähettävät niissä säädettyjen ehtojen mukaisesti käyttöliittymän käyttöön tarkoitetut komennot eteenpäin EventSystem-komponenttiin, joka puolestaan välittää ne käyttöliittymäelementeille.

Tähän tarkoitukseen täytyi luoda oma VRInputModule-luokka, jonka tehtävänä oli ottaa vastaan komentoja syöttölaitteelta. Tämä komponentti sisälsi melkein yksinomaan hyvin toisiaan muisuttavia komentojen uudelleenohjaukseen tarkoitettuja metodeja (kuva 10).

Osoittimen emulointi tapahtui VRPointer-luokassa. Luokan tehtävänä oli luoda säde käyttäen Unityn GraphicRaycaster-luokkaa kuvitteellisen pelaajan käteen kiinnitetyn kameran läpi sen osoittaessa kohta mahdollista valikkoa. Tällöin valikko luulee, että pelaaja osoittaa sitä tavanomaisella hiiriosoittimella.

Tätä tekniikkaa käyttäen rakennettiin PC-alustaa suurimmaksi osaksi vastaava maailmaan sulautettu näyttö. Kaikki pelin asetukset sekä kentänvalinta sijoitettiin pelaajahahmon kuvitteelliseen kotiin, joka siten toimi ikään kuin päävalikkona (kuva 11).

```
public void ClickPress(GameObject gameObject, Vector3 worldPointer)
{
    PointerEventData pointerEventData = new PointerEventData(eventSystem);
    RaycastResult raycastResult = pointerEventData.pointerCurrentRaycast;
    raycastResult.worldPosition = worldPointer;
    pointerEventData.pointerCurrentRaycast = raycastResult;
    ExecuteEvents.Execute(gameObject, pointerEventData, ExecuteEvents.pointerDownHandler);
}

public void ClickRelease(GameObject gameObject, Vector3 worldPointer)
{
    PointerEventData pointerEventData = new PointerEventData(eventSystem);
    RaycastResult raycastResult = pointerEventData.pointerCurrentRaycast;
    raycastResult.worldPosition = worldPointer;
    pointerEventData.pointerCurrentRaycast = raycastResult;
    ExecuteEvents.Execute(gameObject, pointerEventData, ExecuteEvents.pointerUpHandler);
}
```

Kuva 10. Esimerkkejä käyttäjäsyötteen emuloinnista VRInputModule-luokassa



Kuva 11. Pelaajan päävalikkona toimiva tila

5.6 Heijastusnäyttö

Kaikkea kaksiulotteista materiaalia ei kuitenkaan voinut tässä tapauksessa mielekkäästi sulauttaa ympäristöön. Tästä merkittävänä esimerkkinä oli pelissä suuressa roolissa olevan dialogin tekstitykset. Suoraan virtuaalilaitteen näyttöön piirtäminen on yleensä huono ajatus, koska silloin katsojan on vaikea arvioida eteen piirtyvän materiaalin syvyyttä. Ratkaisu ongelmaan löytyy sijoittamalla pelaajan eteen pään liikettä hitaasti perässä seuraava pinta (kuva 12).

```

private void LateUpdate()
{
    FollowHeadPosition();
    CheckNewSnapDir();
    RotateWithVelocity();
}

private void CheckNewSnapDir()
{
    Vector3 projHeadDir = GetProjectedHeadDir();
    if (Vector3.Angle(currentSnapDir, projHeadDir) > snapChangeAngle)
    {
        SetNewSnapDir(projHeadDir);
        //Debug.DrawLine(transform.position, transform.position + projHeadDir * 2, Color.green, Time.deltaTime);
        //Debug.DrawLine(transform.position, transform.position + currentSnapDir * 2, Color.red, Time.deltaTime);
    }
}

private Vector3 GetProjectedHeadDir()
{
    return Vector3.ProjectOnPlane(head.forward, Vector3.up);
}

private void SetNewSnapDir(Vector3 compareDir)
{
    int newSnapIndex = Mathf.RoundToInt((Vector3.SignedAngle(-Vector3.forward, compareDir, Vector3.up) + 180f) / (360f / snapPosCount));
    currentSnapDir = Quaternion.AngleAxis(360f / snapPosCount * newSnapIndex, Vector3.up) * Vector3.forward;
}

private void RotateWithVelocity()
{
    float angle = Vector3.SignedAngle(transform.forward, currentSnapDir, Vector3.up);
    float angleSign = Mathf.Sign(angle);
    float targetAngVelocity = angVelocityCurve.Evaluate(Mathf.Abs(angle)) * angleSign;
    angSpeed = Mathf.MoveTowards(angSpeed, targetAngVelocity, angAcceleration * Time.unscaledDeltaTime);
    Quaternion newRot = Quaternion.AngleAxis(angSpeed, Vector3.up) * transform.rotation;
    float newAngleSign = Vector3.SignedAngle(newRot * Vector3.forward, currentSnapDir, Vector3.up);
    if (Mathf.Abs(angSpeed) <= rotDeadzoneVelocity && (angleSign >= 0 && newAngleSign < 0 || angleSign < 0 && newAngleSign >= 0))
    {
        newRot = Quaternion.LookRotation(currentSnapDir, Vector3.up);
    }
    transform.rotation = newRot;
}

private void FollowHeadPosition()
{
    transform.position = head.position;
}

```

Kuva 12. Pelaajan katsetta seuraavan näytön toiminta

6 Pohdinta

Virtuaalitodellisuuspelin luominen jo tehdyn pelin pohjalta oli monella tapaa uusi ja mielenkiintoinen lähestymistapa verraten pelin luomiseen suoraan virtuaalitodellisuutta ajatellen. Proses-
sissa oli monta samankaltaista ongelmaa ratkaistavana kuin toisessa lähestymistavassa. Kehityk-
sessä korostui palapelimaisuus, koska ei itse pääse vaikuttamaan pelin perusmuotoihin, vaan
omat osat on pystyttävä sovittamaan olemassa oleviin koloihin.

Projektin aikana olin myös oppinut, kuinka asioista ei pidä tehdä liian vaikeita ja monimutkaisia
ilman hyvää syytä. Monet vaaditut virtuaalitodellisuustoiminnallisuudet olivat yleisesti käytettyjä
muissakin vastaavissa peleissä, joita olisi siten ollut helposti mahdollista ulkoistaa samalla saa-
dessa paremman laadun ja huomattavasti halvemman hinnan. Toisaalta mahdollisuus tehdä
enemmän itse antoi todennäköisesti enemmän hyödyllistä kokemusta, vaikka vastaavassa tilan-
teessa seuraavalla kerralla asia tulisi hoitaa toisin.

Lähteet

1. RealityTechnologies.com. The Ultimate Guide to Understanding Virtual Reality (VR) Technology. Viitattu 10.10.2019. <https://www.realitytechnologies.com/virtual-reality/>
2. Tomasz Mazuryk, Michael Gervautz. Virtual Reality - History, Applications, Technology and Future. Viitattu 10.10.2019. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7849>
3. Luke Dormehl, 2017. 8 virtual reality milestones that took it from sci-fi to your living room. Viitattu 11.10.2019. <https://www.digitaltrends.com/cool-tech/history-of-virtual-reality/>
4. Will Greenwald, 2019. The Best VR Headsets for 2019. Viitattu 11.11.2019. <https://www.pcmag.com/article/342537/the-best-virtual-reality-vr-headsets>
5. Mark Barry, 2016. The Future of Manufacturing Design is Virtual Reality. Viitattu 10.10.2019. <https://www.aberdeen.com/opspro-essentials/future-manufacturing-design-virtual-reality/>
6. Unity Technologies. Unity 2019. Viitattu 10.10.2019 <https://unity3d.com/unity>
7. Unity Technologies. VR Overview. Viitattu 10.10.2019. <https://docs.unity3d.com/Manual/VROverview.html>
8. Unity Technologies. Unity Asset Store. Viitattu 11.11.2019. <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>
9. Facebook Technologies, 2019. Unity integration. Viitattu 11.11.2019. <https://developer.oculus.com/downloads/package/unity-integration/>
10. Almar Suarez, 2018. How and why our experiments with virtual reality motion made us ill. Viitattu 11.11.2019. <https://venturebeat.com/2018/02/27/how-and-why-our-experiments-with-virtual-reality-motion-made-us-ill/>
11. Denis Tambovtsev, 2016. How Game Mechanics Change In Virtual Reality. Viitattu 21.11.2019. <https://uploadvr.com/virtual-reality-changes-game-mechanics/>

12. Rewired. Rewired Documentation. Viitattu 10.10.2019. <https://guavaman.com/projects/rewired/docs/>

Kuvalähteet

1. Kuva Sensorama. Viitattu 11.10.2019. Saatavissa: <http://www.mortonheilig.com/sensorama-1.jpg>

Litteet