LAMK Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

Automaatiojärjestelmän visualisointi

Orfer Oy

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Kone- ja tuotantotekniikan
koulutusohjelma
Opinnäytetyö
Kevät 2018
Olli Knuuttila

Lahden ammattikorkeakoulu
Kone- ja tuotantotekniikan koulutusohjelma

KNUUTTILA, OLLI:

Automaatiojärjestelmän visualisointi Orfer Oy

Mekatroniikan opinnäytetyö, 21 sivua

Kevät 2018

TIIVISTELMÄ

Opinnäytetyön tarkoituksena oli luoda visualisoitu malli jo olemassa olevasta 3D-mallista. Tätä visualisoitua mallia käytettäisiin Orfer Oy:n asiakkaiden vakuuttamiseen yrityksen kyvyistä toimittaa monipuolisia automaatiojärjestelmiä. Ennen opinnäytetyötä Orfer Oy omisti muitakin malleja, joiden käyttötarkoitus on sama.

Visualisaation vaatimuksiksi asetettiin liikkeiden samankaltaisuus olemassa olevaan automaatiojärjestelmään, jonka yritys oli toteuttanut, sekä virtuaalitodellisuuden hyödyntäminen mallin katselemisessa ja siinä liikkumisessa.

Toimeksiantaja toimi Orfer Oy, joka on erikoistunut robotiikkaan ja erilaisten automaatiojärjestelmien toimitukseen asiakkailleen turnkey-periaatteen mukaisesti.

Lopputuloksena saatiin toimiva visualisaatio, joka täytti asetetut vaatimukset liikkeiden samankaltaisuuden sekä virtuaalitodellisuuden puolesta.

Asiasanat: visualisointi, automaatio, virtuaalitodellisuus

Lahti University of Applied Sciences
Degree Programme in Mechanical and Production Engineering

KNUUTTILA, OLLI: Visualization of an automation system
Orfer Oy

Bachelor's Thesis in mechatronics 21 pages

Spring 2018

ABSTRACT

The commissioner of this thesis was Orfer Oy, a company specialized in robotics and in providing their customers a variety of different kinds of automatic systems in turnkey deliveries.The objective of this thesis was to create a visualized model from an existing 3D model. This visualization would be used to assure the customers of Orfer Oy that the company is capable in providing versatile automated systems. Before this thesis the company already had a few visualized models that had the same use.

Movements in the new model were to be similar to the ones in an existing automation system that the company had created. The new model should also utilize virtual reality in viewing the visualization and in viewer movement in the visualization.

As a result, the finished visualization answered all the demands in similarity of movements and in virtual reality requirements.

Key words: visualization, automation, virtual reality

SISÄLLYS

SANASTO

3D-STILL MALLI	3D-malli, jossa kaikki siihen kuuluvat komponentit mutta ei minkäänlaisia liikkeitä.
COLLIDER	Määrittelee kappaleen muodot fysiikkamoottoria varten, mahdollistaa kappaleiden osumisen toisiinsa. Näkymätön muualla kuin editorissa. Omistaa myös trigger-muodon.
EDITOR	Unityn käyttöliittymä, jossa projektit luodaan ja niitä muokataan.
TRIGGER	Poistaa fysiikkaominaisuudet colliderista, jolloin sitä voidaan käyttää tunnistamaan muut kappaleet sen alueella.
VR	Lyhenne sanoista Virtual Reality, virtuaalitodellisuus.

1 JOHDANTO

Opinnäytetyön tarkoituksena oli luoda olemassa olevasta 3D-still-mallista virtuaalitodellisuudessa toimiva malli. Tämä kyseinen still-malli oli luotu olemassa olevasta tuotantolinjastosta. Tähän malliin visualisoitiin liikkeitä, joita kyseisessä automaatiojärjestelmässä tapahtuu todellisuudessakin.

Valmiin mallin tarkoitus on vakuuttaa mahdolliset uudet asiakkaat Orfer Oy kyvykkyydestä toimittaa kokonaisia laitteistoja sekä antaa käsitystä, siitä minkä kaltaisia toimintoja laitteistoilla on jo toteutettu.

Visualisaation luomisalustana toimi Unity-pelimoottori, joka jäi ainoaksi vaihtoehdoista ilmaisen käyttömahdollisuutensa takia. Toisena mahdollisuutena oli Visual Components, mutta lisenssien puute esti tämän käytön.

Unity toimi hyvänä alustana visualisaation luomiseen. Ohjelmalla on mahdollista luoda erittäin toimivia malleja sen sisältämien toimintojen ansiosta. Unityssä luodussa visualisaatiossa ohjelmointikielenä toimi C sharp.

Työlle asetettiin kolme päätavoitetta ja rajaukset, jotka sen tulisi täyttää:

– Mallin tulisi toimia virtuaalitodellisuudessa ja sen sisällä tulisi voida liikkua, pelkkä visualisoitu malli ilman virtuaalitodellisuutta ei riitä.

– Kaikki mallin liikkeet tulisi pohjautua oikeisiin liikkeisiin, joita tapahtuu yrityksen tuottamassa olemassa olevassa järjestelmässä. Liikkeiden ei kuitenkaan täydy vastata täydellisesti oikeita liikkeitä, sillä 3D-mallikin poikkeaa hiukan oikeasta laitteistosta.

– Visualisaation tulee olla ennen kaikkea toimiva, toteutetaan vain sellaiset liikkeet ja laitteistot, jotka mahdollista toteuttaa.

2 ORFER OY

2.1 Historia

Orfer Oy perustettiin vuonna 1970 Orimattilassa. Sen tuotteina tuolloin olivat alihankintana valmistetut osat sekä sahateollisuuden koneet ja laitteet. Vuonna 1974 perustettiin toimipiste Lapinlahdelle asiakkaan tarpeesta johtuen. (Orfer 2018a.)

1980-luvulla tuotanto painoittui automaattisiin materiaalinkäsittelyjärjestelmiin. Vuonna 1989 Lapinlahden toimipiste myytiin sen tarpeellisuuden muututtua. (Orfer 2018b.)

Lapinlahdella ollut sarjavalmistus siirtyi Orimattilan tiloihin. Vuonna 1995 aloitetettiin Kawasaki-robottien maahantuonti. (Orfer 2018c.)

2000-luvulla alkoi robottiautomaation kasvu, joka johti Toshiba scara -robottien maahantuontiin sekä vuonna 2006 yrityksen ensimmäisen RoWA-keräilyvarastoautomaatin toimitukseen (Orfer 2018d).

2010-luvulla robottiautomaaatio oli muuttunut pelkästä tuotteesta olemassaolon elinehdoksi. Vuonna 2010 yritys toteutti lukuisia uudistuksia, joilla pyrittiin kehittämään asiakkaiden palvelukykyä. Tämän lisäksi yritys investoi oman tuotantonsa automatisointiin erilaisten laitteistojen ja henkilöstön kouluttamisen muodossa. (Orfer 2018e.)

2.2 Liiketoiminta

Liiketoimintanaan Orfer Oy kehittää, suunnittelee ja valmistaa robottiteknologiaa hyödyntäviä kappaaleenkäsittelyjärjestelmiä asiakkailleen. Laitteistot toimitetaan asiakkaille Turnkey-kokonaisuuksina. Toimitustapa tarkoittaa sitä, että laitteisto toimitetaan asiakkaille koeajettuina ja valmiina tuotantokäyttöön. Yhteistyötä jatketaan toimitusten jälkeen asiakkaan halutessa monenlaisina laitteistojen käyttö- sekä kunnossapitopalveluina. (Orfer 2018f.)

Automaatiokokonaisuudet suunnitellaan sekä valmistetaan Orfer Oy:n tehtailla alusta loppuun. Suunnitteluprosessi sisältää mekaniikan, ohjauksen, ohjelmistot ja käyttöliittymät. Valmistusprosessi kattaa teräksen esikäsittelyn, koneistuksen, hitsauksen, maalauksen sekä laitekokoonpanon. (Orfer 2018f.)

3 VISUALISOINTI

3.1 Yleistietoa 3D-visualisoinnista

3D-visualisaatiolla tarkoitetaan yleensä kuvaa tai videota, joka on luotu työpiirustusten tai muiden suunnitelmien pohjalta kolmiulotteiseksi malliksi (Deemec 2018). Visualisointeja käytetään paljon tehostamassa markkinointia sekä luomaan asiakkaille halutunlainen mielikuva yrityksestä (Design Reform 2018).

Visualisoidut mallit havainnollistavat ja selkeyttävät suunnitelmia, sekä myös vähentävät väärinkäsityksiä (Symetri 2018). Visualisoinneilla on useita eri käyttömahdollisuuksia, niillä voidaan esimerkiksi tarkastella ja arvioida tuotteita, tiloja sekä palvelukokonaisuuksia, joita ei ehkä vielä ole toteutettu (Design Reform 2018).

3.2 3D-animaatiot ja virtuaalitodellisuus

3D-animaatioiden avulla voidaan tehdä monimutkaisista asioista helpommin ymmärrettäviä. Prosesseja tai laitteistoja voidaan yksinkertaistaa näyttämällä laitteen sisällä tapahtuva toiminta sekä selostamalla, kuinka se toimii. Mahdollista on myös poistaa kaikki tarpeeton ja esittää vain haluttu osio. (Design Reform 2018.)

Virtuaalitodellisuuden avulla on mahdollista luoda suuri määrä erilaisia toteutuksia. Mahdollisia toteutuksia ovat koulutukset, opasteet, simuloinnit, prosessin kuvaukset sekä vain myyntisimulaatiot. Virtuaalitodellisuudella luodaan aidon tuntuinen tila ja ympäristö, jossa olevat kohteet ovat oikeassa mittakaavassa. (Design Reform 2018.)

4 VIRTUAALITODELLISUUS

Virtuaalitodellisuudella tarkoitetaan kolmiulotteista, tietokoneellisesti luotua ympäristöä, jossa käyttäjän on mahdollista liikkua ja joissakin tilanteissa jopa vaikuttaa ympäristöönsä. Tähän virtuaalitodellisuuteen päästään sisään käyttämällä esimerkiksi markkinoilla olevia erilaisia headset-järjestelmiä. Näissä järjestelmissä on myös suuria eroja, niin hinnassa kuin lisävarusteissakin. (Virtual Reality Society 2018b.)

Headset-järjetelmät sisältävät useita erilaisia antureita, niissä on erilliset näytöt sekä linssit molemmille silmille, sekä muita järjestelmästä riippuvia komponentteja. Antureista yleisimmät ovat magnetometrit, kiihtyvyysanturit ja gyroskoopit. Näiden antureiden avulla pystytään seuraamaan käyttäjän liikkeitä ja katselusuuntia. Niiden avulla on tarkoitus saavuttaa kuusi vapausastetta (6DoF). (Reality Technologies 2018.)

Virtuaalitodellisuudella on suuri määrä käyttökohteita, joista esimerkkeinä lääketiede-, urheilu-, arkkitehtuuri-, sotilas- sekä viihdekäyttö (Virtual Reality Society 2018a). Virtuaalitodellisuuden avulla voidaan luoda riskitön, mutta todentuntuinen koulutusympäristö useita hankalia tehtäviä varten (Virtual Reality Society 2018b).

5 UNITY-PELIMOOTTORI

5.1 Yleistietoa

Unity on Unity Technologiesin kehittämä pelimoottori, jolla on mahdollista luoda kaksi- ja kolmiulotteisia pelejä. Myös virtuaalitodellisuus sekä lisätty todellisuus ovat tuettuja. (Unity 2018a.)

Unity on laajasti käytetty pelimoottori, jonka avulla on mahdollista luoda monenlaisia eri projekteja. Unity tarjoaa mahdollisuuden ilmaiseen käyttöön, kuitenkin jos käyttävä yritys tuottaa yli satatuhatta dollaria, tulee heidän siirtyä kuukausimaksullisiin versioihin. Nämä versiot myös tarjoavat lisätoimintoja ja laajennettua tukea ostajilleen. (Unity 2018h.)

Unity tarjoaa tuen useille eri alustoille, mahdollistaen yhden projektin luomisen kaikille yleisimmille mobiili-, virtuaalitodellisuus-, tietokone-, konsoli- sekä tv-alustoille (Unity 2018f). Tuettuja alustoja on yli kaksikymmentäviisi kappaletta, enemmän kuin yhdelläkään muulla pelimoottorilla (Unity 2018c).

Unity Editor on saatavilla Windows sekä Mac laitteistoille, se tarjoaa suuren määrän käyttäjäystävällisiä työkaluja projektien luontiin. Olemassa olevien työkalujen lisäksi on myös mahdollista luoda uusia tai ottaa niitä käyttöön. (Unity 2018c.)

Unity tarjoaa myös mahdollisuuden hyödyntää Asset Store palvelua, josta löytyy tuhansia valmiita työkaluja, resursseja sekä laajennuksia (Unity 2018c).

Unityn fysiikkamoottoreina toimivat Box2D sekä NVIDIA PhysX (Unity 2018c).

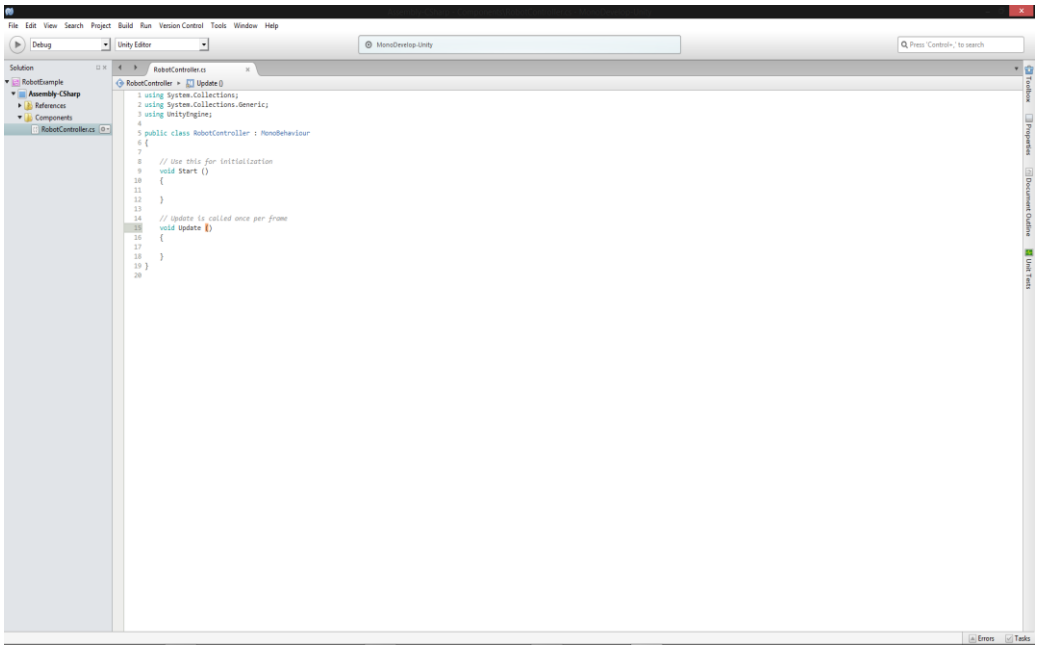
5.2 Ohjelmointikielet

Unity tukee kahta eri ohjelmointikieltä natiivisti. Nämä kielet ovat C sharp sekä UnityScript. C sharp on yleisesti käytössä oleva ohjelmointikieli. UnityScript on kehitetty ainoastaan Unityä varten ja se on tehty JavaScriptin pohjalta. (Unity 2018b.)

Näiden kielien lisäksi muitakin .NET kieliä on mahdollista käyttää, jos ne täyttävät vaaditut vaatimukset toiminnallisesti (Unity 2018b).

5.3 MonoDevelop

MonoDevelop on ohjelmointiympäristö, joka toimitetaan Unityn mukana. Ohjelmointiympäristö tarjoaa tekstieditorin erilaisilla lisätoiminnoilla, kuten virheenkorjauksella ja muilla projektin hallinta toiminnoilla. (Unity 2018e.)



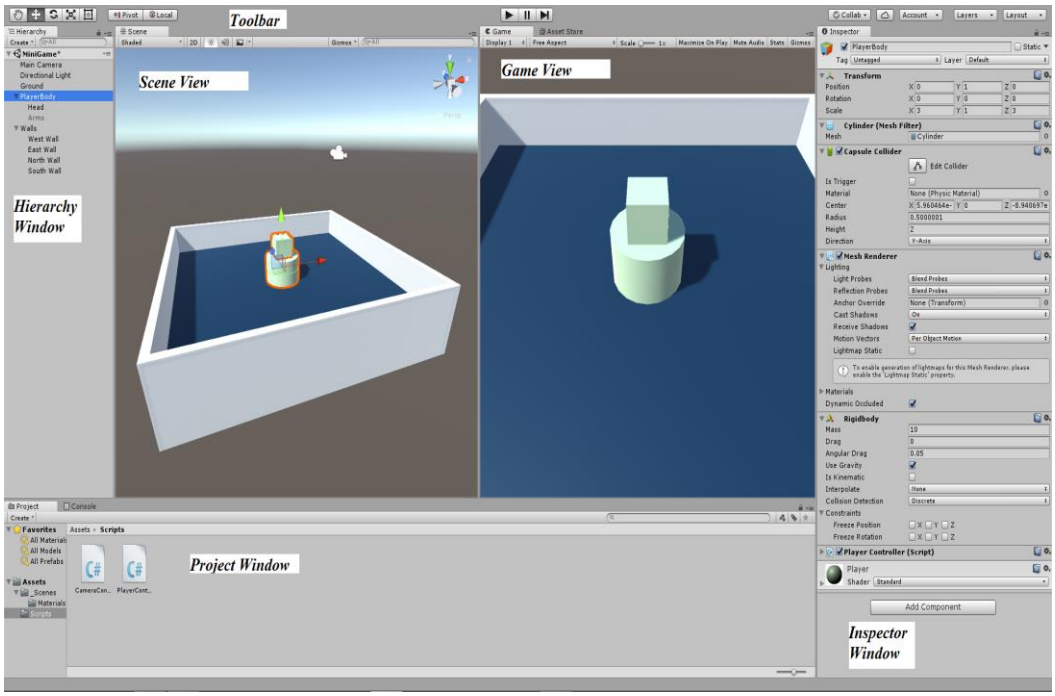
KUVA 1. MonoDevelop-ohjelmointiympäristö

Kuvassa 1 nähtävissä ohjelmointiympäristö, tätä tarkastellessa voidaan havaita miten ohjelmointiympäristö muistuttaa hyvin paljon normaalia tekstieditoria.

5.4 Editor-käyttöliittymän esittely

Editorin käyttöliittymä on opinnäytetyön tekijän omien kokemuksien mukaan todella selkeä, se tarjoaa lähes kaikki tarpeelliset toiminnot suoraan käynnistettäessä ilman etsimistä valikoista. Käyttöliittymä sisältää useita toimintoja, mutta läpikäydään vain opinnäytetyössä käytettyjä toimintoja.

Kuvassa 2 on nähtävissä Unityn käyttöliittymä sekä tärkeimmät toimintoikkunat otsikoituna.



KUVA 2. Unityn käyttöliittymä

Project Window -näkymästä on nähtävissä projektin resurssikirjasto. Se sisältää komponentit, ohjelmakoodit, mahdolliset lisätyökalut sekä laajennukset joita projektissa on käytettävissä. Kaikki projektiin tuodut resurssit näkyvät tässä ikkunassa. (Unity 2018d.)

Scene view -ikkunassa on mahdollista visuaalisesti muokata, sekä liikkua luodussa peliympäristössä. Riippuen projektista tämä näkymä on joko 3D- tai 2D-perspektiivissä. (Unity 2018d.)

Game view -ikkunassa näkyy itse peli renderöitynä siihen asetetuista kameroista. Tämä näkymä on sama kuin mikä se olisi, jos sillä hetkellä loisit projektista peliversion. (Unity 2018g.)

Hierarchy Window on hierarkkinen esitys kaikista projektissa olevista komponenteista. Jokaisella osalla on oma nimi hierarkiassa. Tiettyyn komponenttiin kuuluvat osat linkkautuvat toisiinsa ja tästä paljastuu miten osat ovat kiinnittyneet toisiinsa. (Unity 2018d.)

Inspector Window antaa mahdollisuuden muokata ja tarkastella valitun komponentin ominaisuuksia. Nämä ominaisuudet vaihtelevat komponenttien välillä, joten ikkunan tiedot eivät aina ole samanlaiset. (Unity 2018d.)

Toolbar sisältää kaikki tärkeimmät toiminnot. Vasemmalla ovat perustyökalut pelimaailman ja sen komponenttien manipulointiin. Keskellä ovat play-, pause- ja step-painikkeet. Oikealla sijaitsevat pilvipalvelut ja Unity tili. Lisäksi löytyy vielä tasojen näkyvyyspainike ja editorin asetelma valikko jonka avulla mahdollista muokata editor ikkunoita. (Unity 2018d.)

6 AUTOMAATIOJÄRJESTELMÄN VISUALISOINTI

6.1 Alkutilanne

Projektin alkutilanteessa Orfer Oy:llä oli tarve saada uusi visualisointi heillä jo käytössä olevien rinnalle. Kyseistä uutta visualisaatiota käytettäisiin vanhojen rinnalla asiakkaiden vakuuttamiseen Orfer Oyn kyvykkyydestä toimittaa erilaisia automaatioratkaisuja.

Tähän visualisointiprojektiin valikoitui jo eräälle asiakkaalle toimitetun linjaston 3D-malli. Tässä linjastossa on useita robotteja, kääntöpöytiä, siirtopöytiä, käärintälaitteita sekä kuljettimia. Roboteilla suoritetaan erilaisia lavaustehtäviä, ja kuljettimet ja muut pöydät yksinkertaisesti liikuttavat kappaleita.

Alkuperäinen malli oli pelkkä still-malli, tarkoittaen sitä, että se sisälsi vain paikoillaan olevia laitteistoja, sekä sinne kuuluvia kappaleita kuten lavoja sekä muita niillä olevia esineitä. Kyseinen still-malli oli luotu Visual Components -ohjelmalla.

6.2 Alkuselvitys

Jotta projekti voitiin aloittaa, tuli selvittää, oliko kyseistä 3D-mallia edes mahdollista avata muissa ohjelmissa kuin itse Visual Components -ohjelmassa. Koska visualisaation luontiin ohjelmaksi valikoitui Unity, tehtiin selvitys Unityn ja 3DSMAX -ohjelmiston tukemien tiedostotyyppien pohjalta. 3DSMAX on mallinnukseen luotu ohjelma, jonka avulla voidaan tiedostot muuntaa Unityn tukemaan .FBX-tiedostomuotoon.

Selvityksessä saatiin selville, että Visual Components -ohjelmistolla voidaan tallentaa malleja viiteen eri tyyppiin. Näistä kolme muotoa on yhteensopivia Unityn ja 3DSMAXin kanssa.

Sopivat tiedostotyypit olivat 3D Studio, Stereo Lithography sekä Wavefront. Toisessa vaihtoehdossa eli Stereo Lithography muodossa ongelmaksi ilmeni se, että kyseinen muoto yhdistää kaikki grafiikat yhdeksi kappaleeksi, mikä estää sen käytön tässä tarkoituksessa.

Tiedostotyypeistä kokeilujen jälkeen toimiviksi ratkaisuiksi jäivät 3D Studio sekä Wavefront.

Tiedostot muunnettiin näihin muotoihin, jolloin havaittiin Wavefront tyypin tiedostokoon kasvavan kohtuuttoman suureksi, yli 8 Gt. 3D Studio tiedostokoko oli 1 Gt. Yritettäessä avata tätä Wavefront tiedostoa Unityllä tai 3DSMAXilla ongelmaksi ilmaantui järjestelmän suorituskyvyn riittämättömyys, mikä johti järjestelmän kaatumiseen. Tästä syystä Wavefront malli hylättiin.

Viimeisenä kokeiluna yritettiin avata 3D Studio mallia 3DSMAX -ohjelmassa, mikä ei onnistunut sillä ohjelma ei tunnistanut mallin tiedostomuotoa. Mallin avaaminen suoraan Unityyn kuitenkin onnistui ja kaikki komponentit säilyivät mallissa.

6.3 Still-mallin muokkaus

Still-mallia tuli muokata, jotta sen käyttö visualisaatiossa olisi mahdollista. Mallin tuonti suoraan Unityyn 3D Studio muodossa aiheutti sen, että kappaleet eivät säilyttäneen hierarkiaan. Tällöin jokainen pienikin osa listattiin erillisenä kappaleena, eikä vain tietyn kappaleen osana. Erillisiä kappaleita hierarkiapuussa oli tällöin yli viisikymmentätuhatta.

Aloitin mallin muokkauksen poistamalla turhia komponentteja mallista, esimerkiksi kuljettimilla olevia lavoja. Työ oli hidasta, sillä esimerkiksi lavan jokainen eri osa piti valita erikseen ja poistaa. Ennen poistoa loin näistä kuitenkin ensin mallit, joissa eri osat olivat valmiiksi samassa hierarkiassa. Tämänlaiset mallit tallensin myöhempää käyttöä varten.

Kun turhat komponentit oli poistettu, siirryin luomaan roboteille oikeanlaista hierarkiaa. Robottien eri niveliin kuuluvat komponentit tuli olla oikeassa järjestyksessä, tämä oikeanlainen järjestys on havaittavissa kuvassa 3. Hierarkiassa ensimmäinen oli robotin alusta, josta siirryttiin koko robottiin, josta taas varsiin ja lopulta tarttujaan.

KUVA 3. Esimerkki siitä millainen robotin hierarkiasta luotiin

Roboteille täytyi myös luoda rotaatiopisteet niiden liikkeiden mallintamista varten. Jokaiselle nivelpisteelle luotiin omansa, kuten myös tarttujalle ja koko robotin rotaatioon. Näitä pisteitä apuna käyttäen robottien liikkeet saatiin mallinnettua.

Samankaltainen hierarkia luotiin myös kääntö- ja siirtopöydille. Ilman tätä ei liikkeiden tai rotaatioiden luonti onnistu.

6.4 Liikkeiden visualisointi

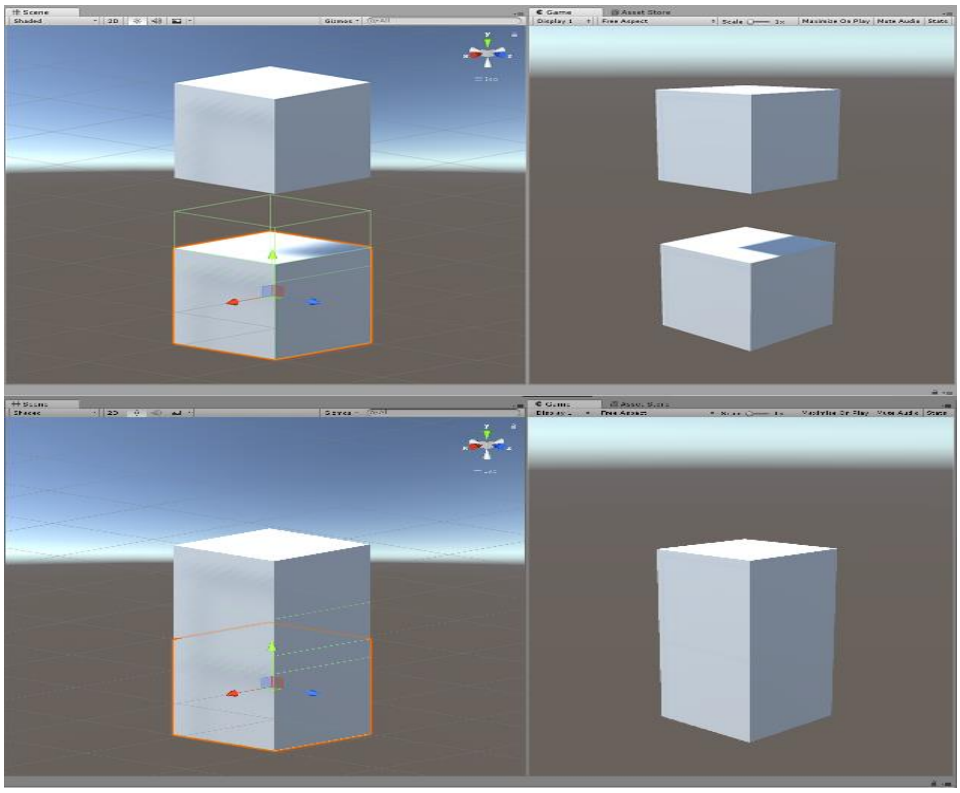
Liikkeiden visualisointi tapahtui robottien kohdalla muuttamalla tietyn osan rotaatiota. Esimerkiksi, jos koko robottia haluttiin kääntää, muutettiin hierarkiapuussa olevan kaikkia robotin osia sisältävän osion rotaatiota.

Halutut liikkeet ja osien rotaatiot toteutettiin luomalla jokaiselle robotille ohjelmakoodi, jonka alkuun määriteltiin näitä osia joiden haluttiin liikkuvan. Näitä osia sitten liikutettiin muuttamalla niiden rotaatiota koodillisesti.

Liikekoodiin annettiin liikutettavan osion sen hetkinen sijainti sekä haluttu kohdesijanti, ohjelmaa toteutettaessa muuttaa se osan sijainnin kyseiseen kohdesijaintiin. Kun liike on toteutettu, verrataan onko kyseisen osan nykyinen sijainti sama kuin annettu haluttu kohdesijainti. Sijainnin ollessa oikein annetaan ohjelman jatkua eteenpäin.

Robottien tartunta toteutettiin varsin yksinkertaisesti. Tarttujan tarttumaosaan luotiin collider, jonka tehtävänä on toimia tartuntapintana. Tämä collider asetettiin trigger-tyyppiseksi, jolloin sitä voidaan käyttää tunnistamaan sen alueen sisälle tulevat komponentit.

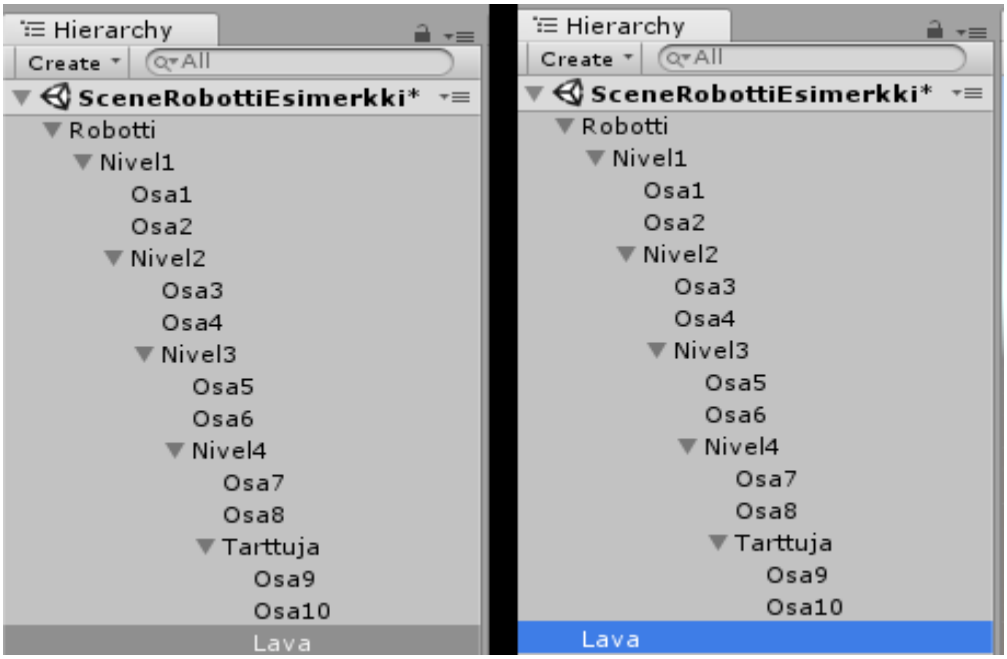
Kuvassa 4 on havaittavissa trigger-tyyppisen colliderin toiminta. Tästä kuvasta nähdään, että fysiikka moottori ei katso kyseistä collideria palikan rakenteeksi, jolloin toinen palikka tippuu colliderin sisään. Tätä collideria voidaan nyt kuitenkin käyttää tunnistamaan jos palikka on alueen sisällä, tulee alueelle tai poistuu alueelta.



KUVA 4. Trigger tyypin collider, kuvasta havaittavissa, että kyseinen collider ei kannattele toista palikkaa vaan se putoaa colliderin sisälle.

Tartuntatoiminto luotiin ohjelmakoodillisesti. Ensin tarttujaan luotiin oma ohjelmakoodi. Tätä ohjelmakoodia sitten kutsuttiin robotin liikekoodista, kun robotin sijainti sattui olemaan paikassa, jossa tartuntaa tarvittiin. Tartuntakoodin toiminta on varsin yksinkertainen. Se yksinkertaisesti muuttaa kappaleet, jotka ovat sen colliderin sisällä osaksi omaa hierarkkiaan. Esimerkiksi, jos tartutaan lavaan, muuttaa koodi lavan hierarkiassa osaksi tarttujaa ja sen alapuolelle sen osaksi.

Kun taas halutaan irrottaa ote toisessa sijainnissa, muutetaan tartuttu kappale omaksi hierarkiakseen, tällöin siitä tulee erillinen komponentti ja se voi taas liikkua itsekseen. Kuvassa 5 nähtävissä miten kappale muuttuu osaksi osaksi robotin tarttujaa tartunnan ollessa päällä ja päästettäessä irti se muuttuu erilliseksi komponentiksi.



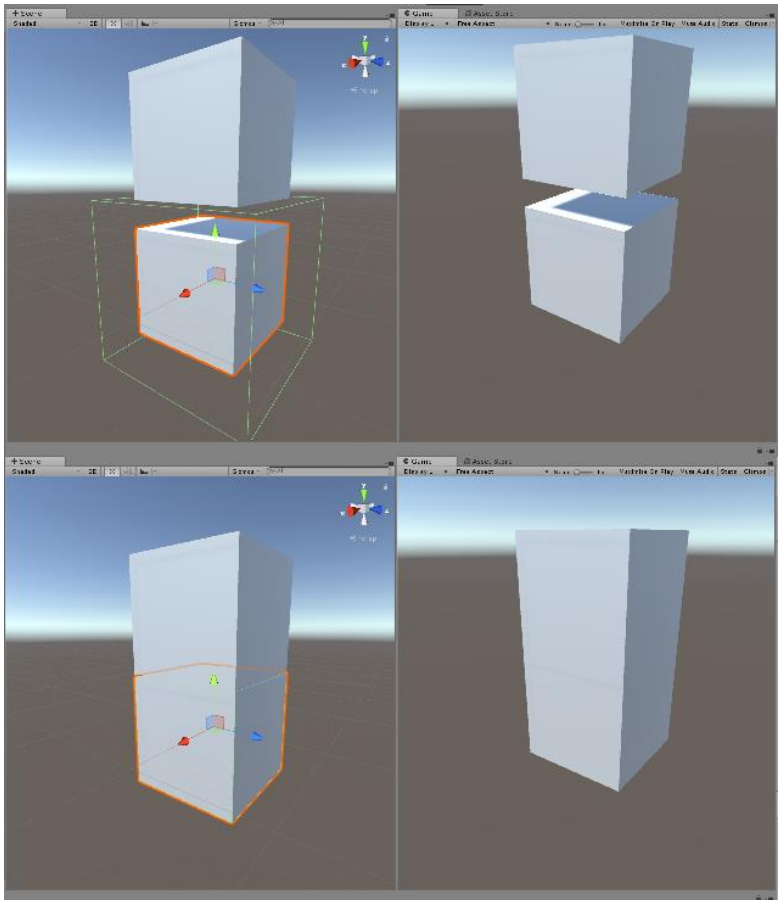
KUVA 5. Esimerkki tartuntatoiminnosta tartuttaessa lavaan, vasemmalla tartunta on asetettu päälle ja oikealla pois päältä.

Kuljettimien liikkeet luotiin myös collider perusteella. Ohjelmakoodillisesti osa kuljettimista oli varsin yksinkertaisia, toiset hiukan monimutkaisempia. Kaikkien koodi kuitenkin sisältää ehdot sille milloin kuljetin toteuttaa liikkeet ja milloin ei. Tämä tieto otetaan esimerkiksi siitä onko robotti lavannut tarpeeksi tai onko jokin anturi aktiivinen. Liike toteutettiin antamalla liikutettavalle kappaleelle liikekäsky haluttuun suuntaa sekä nopeus jolla se toteutetaan.

Kääntö- sekä siirtopöydät toteutettiin asettamalla niihin myös tartuntatoiminto. Tämä toiminto on samanlainen kuin roboteillakin. Tartunta on pakollinen lisä näille, sillä jos kappaleisiin ei tartuta ennen kääntöä tai liikuttamista ne eivät liiku itse pöydän mukana.

Kaikille komponenteille, joiden täytyy kannatella tai voida osua muihin kappaleisiin täytyy luoda myös collider rakenteet. Nämä collider rakenteet määrittävät varsinaiset fyysiset pinnat, joihin toiset kappaleet voivat osua ja niiden liikkeet pysähtyä. Esimerkiksi jos kuljettimeen ei ole näitä luotu putoavat lavat sen lävitse, mutta on huomioitava, että myös lavat tarvitsevat tällaiset rakenteen tai sama toteutuu uudestaan.

Kuvasta 6 havaittavista normaalin colliderin toiminta, collider kannattelee toista palikkaa. Collider ei näy itse lopullisessa tuotoksessa, mutta editorissa se näkyy vihreinä rajoina. Vasemmalla editori näkymä, oikealla miltä tilanne näyttää Game View -näkymässä, joka vastaa lopullista tuotosta.



KUVA 6. Kuvasta havaittavista normaalin colliderin toiminta, tämä collider määrittää laatikon rajat fysiikkamoottorille

Colliderit voi luoda usealla eri tapaa, yhtenä tapana on käyttää automaattista luomistoimintoa, joka luo colliderin kappaleen visuaalisen mallin mukaan. Toinen tapa on luoda ne itse Unitystä löytyvien neliskanttisten tai pyöreiden mallien avulla.

Automaattinen luomistoiminto ei tässä visualisaatiomallissa onnistunut. Toiminto aiheutti virheellisiä rakenteita, mikä johti ei-toivottuihin liikkeisiin visualisaatiossa. Näitä liikkeitä olivat esimerkiksi lavan äkillinen pysähtyminen sen osuttua collideriin, jota ei kyseissä paikassa tulisi olla.

Tästä johtuen kaikki visualisaatiomallin colliderit on luotu manuaalisesti. Kuljettimille luotiin neliskanttiset colliderit kannattelemaan kappaleita niiden päällä. Tämä poisti kaikki ei-halutut liikkeet ja virhetilanteet.

6.5 Virtuaalitodellisuuden lisääminen malliin

Unity tarjoaa valmiita ratkaisuja virtuaalitodellisuuden lisäämiseksi sillä luotuihin projekteihin. Virtuaalitodellisuus otettiin tässä tapauksessa käyttöön SteamVR-paketin avulla. Tähän syynä oli käytettävän virtuaalitodellisuusjärjestelmän tyyppi ja valmiiksi asennetut ohjelmistot. SteamVR-paketti tarjoaa valmiin tuen suurimmalle osalle VR-järjestelmiä.

Käyttöönotto on varsin yksinkertainen. Virtuaalilasituki tuodaan projektiin sisään Asset Storesta, tämän jälkeen lisätään halutut toiminnot projektin hierarkiaan ja muokataan niitä. Haluttuja toimintoja tässä tapauksessa on liiketoiminnot ja virtuaalijärjestelmän tuki. Toiminnoissa olevia asioita, joita täytyy muokata, ovat lasien piirtoetäisyys, ohjainten korkeus maatasosta sekä liiketoimintojen nopeuksia ja liikealueiden kokoja.

Kuvassa 7 nähtävissä SteamVR-paketin sisältö, paketti tarjoaa kaikki tarvittavat komponentit ja ohjelmakoodit virtuaalitodellisuuden lisäämiseksi yksinkertaisiin projekteihin.

KUVA 7. SteamVR-paketin sisältö

Liiketoimintona käytettiin portaaliliikkumista. Tämä tarkoittaa ennalta määrätyille alueille asetettuja kenttiä, joille liikkuminen on sallittua. Käyttäjä osoittaa kenttää virtuaalistodellisuudessa ja napinpainalluksella liikkuu sinne. Näiden kenttien koko tuli säätää oikeanlaiseksi, jotta niiden välillä liikkuminen onnistui. Ohjainten korkeus täytyi muuttaa, jotta liike olisi mahdollista sillä ne olivat oletusarvoilla maatason sisällä. Piirtoetäisyyden muutoksilla mahdollistettiin koko mallin näkeminen, sillä liian lyhyellä etäisyydellä laitteistoja ei piirretty.

7 YHTEENVETO

Opinnäytetyön tarkoituksena oli luoda visualisaatio tuotantolinjastosta. Visualisaatiossa liikkeet rajattiin totuudenmukaisiksi, mutta ei kuitenkaan täysin vastaamaan oikean järjestelmän kokonaisuutta.

Tarkoituksena oli luoda visualisaatio, josta mahdolliset asiakkaat voisivat saada näkemyksiä millaisia automaatiojärjestelmiä on mahdollisesti luotu, sekä todeta Orfer Oyn kyvykkyyden tuottaa niitä.

Työ aloitettiin selvittämällä millainen visualisaatio haluttiin, sekä selvittämällä kuinka valmis still-malli saadaan toiseen ohjelmaan. Visualisaation luominen aloitettiin siistimällä still-malli. Kun malli oli siisti luotiin komponenteille hierarkiat, colliderit, sekä liikkeet. Lopuksi mahdollistettiin virtuaalistodellisuuden käyttö.

Opinnäytetyön tekijällä oli aiempaa kokemusta simulaatioiden ja visualisointien luonnista Visual Components -ohjelmistolla opintojen puolesta, mutta Unity oli aivan uusi ympäristö. Opinnäytetyön aikana oli opeteltava Unityn käyttöä lähes alkeiden tasolta.

Opinnäytetyön lopputuloksena saatiin visualisaatio, joka täyttää sille projektin alussa asetetut vaatimukset. Visualisoituun malliin saatiin yhdistettyä virtuaalitodellisuus ja liikkuminen mallin sisällä SteamVR-paketin avulla, laitteistojen liikkeet pohjautuvat oikeaan järjestelmään kuitenkin hiukan yksinkertaistettuina ja lopullinen malli toimii ilman virhetiloja. Visualisaatiota tullaan käyttämään mahdollisille asiakkaille laitteistojen esittelyyn.

LÄHTEET

Deemec Oy. 2018. 3D-visualisointi [viitattu 7.2.2018]. Saatavissa: http://www.deemec.com/3d-visualisointi/

Design Reform Oy. 2018. Visualisointi [viitattu 7.2.2018]. Saatavissa: http://www.reform.fi/visualisointi/

Orfer Oy. 2018a. Historia 1970-luku [viitattu 30.1.2018]. Saatavissa: https://www.orfer.fi/suomeksi/WITHORFER/HISTORIA/1970LUKU/tabid/13088/language/en-US/Default.aspx

Orfer Oy. 2018b. Historia 1980-luku [viitattu 30.1.2018]. Saatavissa: https://www.orfer.fi/suomeksi/WITHORFER/HISTORIA/1980LUKU/tabid/13089/language/en-US/Default.aspx

Orfer Oy. 2018c. Historia 1990-luku [viitattu 30.1.2018]. Saatavissa: https://www.orfer.fi/suomeksi/WITHORFER/HISTORIA/1990LUKU/tabid/13090/language/en-US/Default.aspx

Orfer Oy. 2018d. Historia 2000-luku [viitattu 30.1.2018]. Saatavissa: https://www.orfer.fi/suomeksi/WITHORFER/HISTORIA/2000LUKU/tabid/13091/language/en-US/Default.aspx

Orfer Oy. 2018e. Historia 2010-luku [viitattu 30.1.2018]. Saatavissa: https://www.orfer.fi/suomeksi/WITHORFER/HISTORIA/2010LUKU/tabid/13092/language/en-US/Default.aspx

Orfer Oy. 2018f. Liiketoiminta [viitattu 30.1.2018]. Saatavissa: https://www.orfer.fi/suomeksi/WITHORFER/LIIKETOIMINTA/tabid/12874/language/en-US/Default.aspx

Reality Technologies. 2018. Virtual Reality [viitattu 14.2.2018]. Saatavissa: http://www.realitytechnologies.com/virtual-reality

Symetri. 2018. Visualisointi [viitattu 7.2.2018]. Saatavissa: http://www.symetri.fi/tuotteet-ja-ratkaisut/visualisointi/

Unity. 2018a. Company Facts [viitattu 6.2.2018]. Saatavissa: https://unity3d.com/public-relations

Unity. 2018b. Creating and Using Scripts [viitattu 6.2.2018]. Saatavissa: https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html

Unity. 2018c. Features [viitattu 6.2.2018]. Saatavissa: https://unity3d.com/unity

Unity. 2018d. Learning the interface [viitattu 12.2.2018]. Saatavissa: https://docs.unity3d.com/Manual/LearningtheInterface.html

Unity. 2018e. MonoDevelop [viitattu 6.2.2018]. Saatavissa: https://docs.unity3d.com/Manual/MonoDevelop.html

Unity. 2018f. Multiplatform [viitattu 6.2.2018]. Saatavissa: https://unity3d.com/unity/features/multiplatform

Unity. 2018g. The Game view [viitattu 12.2.2018]. Saatavissa: https://docs.unity3d.com/Manual/GameView.html

Unity. 2018h. Unity Store [viitattu 13.2.2018]. Saatavissa: https://store.unity.com/

Virtual Reality Society. 2018a. Applications of Virtual Reality [viitattu 14.2.2018]. Saatavissa: https://www.vrs.org.uk/virtual-reality-applications/

Virtual Reality Society. 2018b. What is Virtual Reality? [viitattu 14.2.2018]. Saatavissa: https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html