



3ds Max MAXScript -automaatio

Mikko-Oskari Laitinen

OPINNÄYTETYÖ
Elokuu 2022

Tietojenkäsittelyn tutkinto-ohjelma
Pelituotanto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma
Pelituotanto

LAITINEN, MIKKO-OSKARI:
3ds Max MAXScript -automaatio

Opinnäytetyö 21 sivua
Elokuu 2022

Opinnäytetyön tarkoituksena oli luoda erilaisia MAXScript-kielellä kirjoitettuja scriptejä, joiden tavoitteena oli auttaa nopeuttamaan työpaikan 3D-mallinnusprosessia. Tilaajana toimi POF Visuals, joka toteuttaa visualisointeja erilaisista rakennusprojekteista asiakkailleen käyttäen 3ds Max sovellusta.

Toteutustapana opinnäytetyössä oli koodin kirjoittaminen ja aikaisempien scripttien tutkiminen ja päivittäminen sekä toimeksiantajan tarpeiden kartoitus. Tutkimustapana käytettiin konstruktivistista tutkimustapaa, sillä tavoitteena oli tuottaa uusia toimintamalleja ja tehokkaita tapoja tehdä 3D-mallinnusprosessin toistuvia työvaiheita.

Opinnäytetyöprosessin alussa perehdyttiin itse MAXScript-kieleen ja siitä löytyvään kirjallisuuteen ja dokumentaatioon, sekä kirjoitettiin scriptit. Tuotoksia testattiin lähes koko kehityksen ajan, jolloin pystyttiin korjaamaan mahdolliset ongelmakohdat ja lisäämään uusia toiminnallisuuksia, joiden tarve ilmeni usein vasta käyttäjien testatessa scriptejä.

Työssä tuotettiin scriptejä, joita käytetään lähes päivittäin 3D-mallinnusprosessin nopeuttamiseksi. Opinnäytetyötä varten tuotettujen scripttien kirjoittaminen opetti tekijäänsä koodaamaan tehokkaammin, sekä haastoi ohjelmistokehityksen ongelmilla. Työtä tehdessä tekijä oppi myös monia hyödyllisiä asioita 3D-mallintamisesta ja 3d Max-ohjelman käytöstä.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Game Development

LAITINEN, MIKKO-OSKARI:
3ds Max MAXScript Automation

Bachelor's thesis 21 pages
August 2022

The goal of the thesis was to create scripts written in MAXScript programming language, which would help to speed up the workplace 3D-modeling processes. The thesis was commissioned by POF Visuals, a company that produces 3D visualizations of different building projects for their customers.

The process for creating the scripts started with mapping the needs of the users and exploring the code that had already been written. Next, the new code was written and some of the already existing one was updated.

The scripts were tested extensively during the development. This allowed for efficient fixing of any problems that arose, and the addition of new functionality to the scripts, the need for which only became apparent during testing. Writing the scripts proved to be time-consuming and most of them will require more development time.

In the end, the produced scripts are used at the workplace daily to make the work process better.

Key words: 3ds Max, MAXScript, automation, 3D visualization, script

SISÄLLYS

1	JOHDANTO	6
2	TAUSTAT	7
2.1	3ds Max ohjelman historiaa	7
2.2	Toteutustavan valinta	7
2.2.1	MAXScript-ohjelmointikieli	8
2.2.2	Python-ohjelmointikieli.....	8
2.2.3	Muut ohjelmointikielet ja 3ds Max SDK	9
3	TOTEUTUS	10
3.1	AloitUS	10
3.2	Scriptien toteutus	11
3.2.1	Hotspot scripti.....	12
3.2.2	Balcony Extruder scripti.....	15
3.2.3	XML scripti.....	16
3.2.4	Rotate 90 scripti.....	17
4	POHDINTA	18
	LÄHTEET.....	20

LYHENTEET JA TERMIT

.NET rajapinta	Microsoft Windowsin rajapinta
Autodesk	Autodesk on yritys, joka tuottaa monia 3D-mallintamiseen liittyviä ohjelmia maailmanlaajuisesti näihin kuuluu muun muassa opinnäytteessä käsitelty 3ds Max
Editori	Koodin kirjoittamiseen tarkoitettu tekstinkäsittelyohjelma
MAXScript	Autodeskin 3ds Max ohjelman mukana tuleva ohjelmointikieli, jolla voi helposti laajentaa ohjelman toiminnallisuuksia
Renderöintimoottori	Ohjelma, joka piirtää tietokoneen näytölle kuvan joko 2D tai 3D mallista
Scene	3ds Max ohjelman sen hetkinen maailma, eli tietokoneen näytöllä näkyvät asiat, jossa käyttäjä voi käsitellä objekteja
Scripti	Komentosarja, lyhyt tietokoneen ohjelma. Tämän opinnäytetyön kontekstissa Autodeskin 3ds Max ohjelman MAXScript kielellä kirjoitettu, ohjelman toimintaa laajentava lyhyt tietokoneen ohjelma
SDK	Software development kit, ohjelmistonkehityspaketti

1 JOHDANTO

3ds Max-ohjelman mukana tulee MAXScript-ohjelmointikieli, jolla voidaan muokata sovellusta ja lisätä siihen toiminnallisuuksia. Kielen syntaksi on helposti ymmärrettävä toisiin ohjelmointikieliin verrattuna. Kieli on jossain määrin rajoittunut verrattuna yleisemmin käytettyihin ohjelmointikieliin. Silti sillä on toteutettu monia 3D-mallinnus alalla toimivien 3ds Max käyttäjien käyttämiä scriptejä.

Opinnäytetyön tekijän harjoittelupaikassa oli ilmennyt tarve automatisoida mallinnusprosessin eri vaiheita tuottamalla uusia scriptejä kirjoittamalla uutta lähdekoodia ja lisäksi jatkojalostamalla ja laajentamalla jo toteutettuja työkaluja. Niinpä opinnäytetyön tekijä kehitti osana opinnäytetyötään tähän tarkoitukseen soveltuvia scriptejä 3ds Max -ohjelman MAXScript-kielellä.

Opinnäytetyön tarkoituksena oli toteuttaa toimeksiantajalle erilaisia 3ds Max ohjelman scriptejä, joilla nopeutettaisiin työpaikan prosesseja. Tämän lisäksi scriptit dokumentoitiin. Tämä sisälsi dokumentaation niin kirjoitetun lähdekoodin puolella, kuin käyttöohjeet tuleville scriptien käyttäjille.

2 TAUSTAT

2.1 3ds Max ohjelman historiaa

3ds Max ohjelma julkaistiin ensimmäisen kerran vuonna 1990 MS-DOSille nimellä 3D studio. Tällöin sitä kehitti Yost Group ja Autodesk toimi julkaisijana. Vuonna 1996 nimi muuttui 3D studio MAX 1.0:ksi kun siirryttiin Windows NT:lle. Samana vuonna julkaistun MAX 1.1 mukaan sisällytettiin 3ds Max SDK, jonka myötä kolmansien osapuolien oli helpompi kehittää uusia plugineja laajentamaan ohjelman toiminnallisuuksia. (Autodesk, 2015.)

Vuonna 1997 Autodesk osti koko tuotteen ja julkaisi uuden version 3D Studio MAX R2 nimellä. Mainittavaa tästä julkaisusta on, että sen yhteydessä MAXScript esiteltiin osana tuotetta ensimmäisen kerran. Vuonna 2000 nimi muuttui 3ds Max -muotoon, kun siirryttiin Windows 98, ME -ja 2000 aikaan. Vuonna 2007 uusi versio julkaistiin nimellä Autodesk 3ds Max 2008, joka vastaa nykyistä nimeämis-muotoa. Tämän julkaisun myötä MAXScript editoria päivitettiin vastaamaan nykyistä versiota editorista. Vuodesta 1997 MAXScript on ollut osana tuotetta, ja sillä on kirjoitettu paljon erilaisia scriptejä moniin tarkoituksiin. (Autodesk, 2015.)

2.2 Toteutustavan valinta

Ennen toteutuksen käytännön tekemisen aloittamista, oli tärkeää valita oikea toteutustapa opinnäytetyölle. Tässä tapauksessa tekijä keskittyi opinnäytetyön tekemisessä käytettävän ohjelmointikielen valintaan ja siitä seuraaviin käytännön implikaatioihin. Myös toteutuksien rakennetta, toimintalogiikkaa, sekä joihinkin toteutuksiin vaadittavia trigonometrian käsitteitä tutkittiin.

Lisäksi toteutusten aikataulu ja niiden tavoitteet selkenivät tekijälle tässä työn vaiheessa.

2.2.1 MAXScript-ohjelmointikieli

Autodeskin sivuilla MAXScriptiä kuvaillaan seuraavasti. MAXScript ohjelmointikieli on suunniteltu täydentämään 3ds Max -sovellusta. Kieli käyttää olio-ohjelmointia, ja se sisältää monia erikoisominaisuuksia ja konstruktoreita, jotka peilaavat korkean tason konsepteja 3ds Max ohjelman käyttöliittymässä. Näihin kuuluu muun muassa koordinaatisto systeemin konteksti, animointi automatisoiduilla keyframeilla ja pääsy 3ds Max scenen objekteihin käyttäen hierarkkisia polun nimiä, jotka vastaavat 3ds Max sovelluksen objekti hierarkiaa. Kielen syntaksi on helposti ymmärrettävä ihmisille ilman ohjelmointi taustaa, sillä se sisältää vain minimaalisesti erikoismerkkien käyttöä ja formatointiin liittyviä sääntöjä. (Autodesk, 2021.)

MAXScriptiä käytetään paljon työpaikkojen sisäisten prosessien automatisoinnin toteuttamiseen. Sillä pystyy kirjoittamaan hyvinkin monimutkaisia asioita toteutuvia scriptejä ja näin lisäämään toiminnallisuuksia Autodeskin 3ds Max -ohjelmaan. Monet näistä scripteistä ovat paljon käytettyjä 3D-mallinnus alalla edelleen, esimerkiksi Relinkbitmaps (Autodesk, 2022a) ja Floorgenerator (CGpress, 2016) sekä Copitor (CGpress, 2011).

2.2.2 Python-ohjelmointikieli

Vuonna 2016 Autodesk lisäsi 3ds Max ohjelmaan python-ohjelmointikielen tuen, joka on toteutettu hyvin samaan tapaan, kuin tuki Microsoftin .NET rajapinnan käyttöön. Tämän ohjelmointikielen käyttö 3ds Max ohjelmassa on kuitenkin huomattavasti huonommin dokumentoitu kuin 3ds Maxin oman MAXScriptin.

Tekijä kuitenkin huomasi, että suurin osa hyvästä opetusmateriaalista ja keskustelupalstoista on hyvin hajallaan ja vaikeasti löydettävissä erilaisilla aiheeseen liittyvillä foorumeilla. Python ohjelmointikieli itsessään on erittäin käytetty ja hyvin dokumentoitu, sekä paljon keskusteltu ohjelmointikieli 3ds Max -sovelluksesta irrallaan, mutta ei vaikuttanut tähän opinnäytetyöhön soveltuvalla ohjelmointikieltä.

2.2.3 Muut ohjelmointikielet ja 3ds Max SDK

Autodesk itse ja Autodeskin ulkoiset toimijat ovat kehittäneet ja julkaisseet monia, pienempiä MAXScript scriptejä isompia kokonaisuuksia, mutta ne ovat ennemminkin niin sanottuja plugineja eli ne on kirjoitettu yleisimmin C++ ohjelmointikielellä, tai C# ohjelmointikielellä käyttäen 3ds Max SDK:ta, joka tulee ohjelman lisenssin mukana. Toisin sanoen ne ovat ikään kuin omia erillisiä ohjelmiaan, jotka on kirjoitettu samalla kielellä kuin itse 3ds Max. Lisäksi ne täytyy ennen käyttöä kompiloida kääntäjällä konekieleksi, jonka tietokone voi suorittaa, kuten kaikki C++ ohjelmointikielellä kirjoitetut sovellukset. Niihin voi sisältyä usein myös jonkin verran MAXScriptiä. (Autodesk, 2007a.)

Nämä niin kutsutut pluginit ovat monesti maksullisia ja niissä on yleensä paljon kattavampi tuki kuin pienemmissä, usein ilmaisissa, puhtaasti MAXScriptiä käytävissä scripteissä. Näistä merkittävimpinä voidaan mainita esimerkiksi Chaos Gropin V-ray renderer, joka korvaa 3ds Max -ohjelman oman renderöintimoottorin. V-ray on 3D-visualisointi alalla paljon käytetty renderöintimoottori (Mottle, 2021).

3 TOTEUTUS

3.1 Aloitus

Aluksi kartoitettiin mitä toimeksiantaja tarvitsee, sekä millä aikataululla toteutukset pitäisi saada valmiiksi ja käyttöön. Tämän jälkeen päätettiin tapa ja ohjelmointikieli, jolla scriptit tullaan kirjoittamaan.

Kirjoittaja perehtyi samaan aikaan aiheesta löytyvään kirjallisuuteen ja moniin verkkomateriaaleihin, jotta voisi paremmin päättää toteutustavan ja käytettävän ohjelmointikielen.

Tekijä päätyi toteuttamaan scriptit 3ds Maxin omalla MAXScript-ohjelmointikielillä, sillä ensinnäkin se oli paremmin dokumentoitu kuin toisen sisäänrakennetun vaihtoehdon eli python-ohjelmointikielen käyttö 3ds Maxissa.

Toisekseen MAXScript-ohjelmointikieli oli helpompi käyttää ja oppia verrattuna ohjelman SDK:n ja alemman tason C++ ohjelmointikielen käytön opettelemiseen. Toteutukset olivat tarpeeksi pieniä, jotta tekijä ei nähnyt tarvetta käyttää edellä mainittuja raskaampiin ja monesti kaupallisempiin toteutuksiin soveltuvia työkaluja. (Autodesk, 2022b.)

Kolmas, ja ehkä painavin syy MAXScript-kielen valintaan oli se, että työpaikalla oli jo käytössä valmiiksi kirjoitettuja scriptejä, jotka oli toteutettu MAXScript-ohjelmointikieltä käyttäen. Yhteensopivuuden takaamiseksi vanhojen scriptien kanssa tekijä päätti toteuttaa uudetkin scriptit samalla ohjelmointikielillä.

Monia työpaikalla jo aiemmin kirjoitettuja MAXScript-scriptejä piti parannella ja niihin piti lisätä toiminnallisuutta, joten tekijä perehtyi aluksi tapaan ja käytäntöihin, jolla ne oli toteutettu, kommentoitu ja dokumentoitu. Näin tekijä oppi paljon kielen käytöstä käytännössä, ja pystyi hyödyntämään tätä tietoa omissa toteutuksissaan.

3.2 Scriptien toteutus

Tekijä totesi, että MAXScriptin opettelu kannattaa aloittaa käyttämällä hyväksyen Autodeskin omilta verkkosivuilta löytyvää MAXScript dokumentaatiota, jossa on dokumentoitu melko kattavasti MAXScript-ohjelmointikielen toiminnallisuuksia (Autodesk, 2007b).

MAXScript Listenerin Macro Recorder on hyödyllinen työkalu MAXScript lähdekoodin kirjoittamista aloittaessa, sillä se tulostaa MAXScript listeneriin mitä MAXScriptiä 3ds Max taustalla suorittaa, kun käyttäjä tekee sovelluksessa asioita. Esimerkiksi, kun käyttäjä luo objekteja tai valitsee ja muokkaa niitä. Näin käyttäjä voi nähdä mitä lähdekoodiin pitäisi kirjoittaa, jotta scripti tekisi samoja asioita kuin käyttäjä teki sovelluksessa. Listener tulostaa lähes kaiken mitä käyttäjä voi 3ds Maxissa tehdä. (Autodesk, 2007c.)

Kuten edellä mainittiin, MacroRecorder ei tulosta kaikkia 3ds Maxin toiminnallisuksia Listeneriin, joten tähän tapaan kirjottaa lähdekoodia ei kannata missään nimessä luottaa liikaa.

On lisäksi suositeltavaa käyttää hyödykseen MAXScriptin tehokkaampia, ohjelmointikielelle tyypillisiä ominaisuuksia. Varsinkin jos scriptejä tulee käyttää laajemmissa sceneissä, sillä näin scriptit ovat suorituskyvyltään nopeampia ja koodi on paljon luettavampaa.

MAXScriptin yleisimmin käytettyihin ominaisuuksiin sisältyvät muun muassa muuttujat joihin käyttäjä voi tallentaa dataa, johon voi näin päästä käsiksi myöhemmin (Autodesk, 2007d).

Myös Funktiot ovat erittäin tärkeä työkalu MAXScript scriptejä kirjoittaessa, sillä niiden avulla koodin jäsentely on helpompaa ja koodin luettavuus paranee (Autodesk, 2007e).

Lisäksi Arrayt ja objektien verteksien manipulointiin optimoidut BitArrayt ovat tehokas tapa suorittaa asioita isolle joukolle objekteja (Autodesk, 2007f).

Lisäksi MAXScriptissä paljon käytettyihin ominaisuuksiin kuuluvat if-lauseet ja else-lauseet (Autodesk, 2007g). Sekä lisäksi for-loopit ja while-loopit (Autodesk, 2007h). Nämä koodin jäsentelyä varten olevat työkalut ovat lähes välttämättömiä hyvin skaalautuvan ja ymmärrettävän koodin kirjoittamiseen.

MAXScript tukee olio-ohjelmointia ja näin ollen kirjoittaja voi käyttää hyväkseen myös itse luotuja luokkia ja konstruktoreita, sekä pääsee käsiksi jo valmiiksi luotujen 3dsMax ohjelman omien luokkien ja konstruktorien ominaisuuksiin (Autodesk, 2007i).

Seuraavissa alakappaleissa esitellään merkittävimmät tämän opinnäytetyön aikana toteutetuista scripteistä.

3.2.1 Hotspot scripti

Ensimmäinen projekti oli nimeltään Hotspot scripti. Toimeksiantajalla oli tarve saada dataa 3ds Max scenestä ulos tekstinä yhtä nappia painamalla. Tarkemmin sanottuna tarve oli saada scenessä olevien kameroiden eli kuvapisteen sijainnit toisiinsa nähden laskettua ja kirjoittaa ne teksti tiedostoon.

Huomioitavaa on, että heillä oli jo scripti tähän, mutta se ei toiminut. Tässä toteutuksessa kuitenkin pystyttiin hyödyntämään suurta osaa jo kirjoitetusta koodista. Ja jo kirjoitettu koodi toimi tekijälle ohjenuorana, miten asioita voi MAX-Scriptissä toteuttaa.

Toteutus vaati melko paljon aikaa ja iteraatioita. Suurimmaksi ongelmaksi tekijälle muodostui trigonometrian soveltaminen vektorien laskemisen taustalla.

Koko scripti on käytännössä yhden ison for-loopin sisällä, jossa se vertaa kaikkia scenen hotspot helper-objektien sijaintia toisiinsa ja kirjoittaa niitä sitä mukaa MaxScriptin filestream-tyyppiseen muuttujaan, jota käytetään, kun halutaan kirjoittaa tiedostoon, tässä tapauksessa lopulliseen json-tiedostoon. (Autodesk, 2006a).

Aluksi scripti piti saada laskemaan objektien sijainnit suhteessa toisiinsa oikein. Vaikutti siltä, että se suoritti vaadittavat laskutoimitukset välillä oikein ja välillä väärin. Lopulta todettiin, että laskeminen oli aina sujunut oikein, mutta objektien nimeämisestä aiheutui ongelma, jota ei ollut otettu huomioon. Tämä vaikutti järjestykseen, jossa MAXScript laittaa objekteja arrayhin. Tämän johdosta scripti vertaili joissain tapauksissa väärin objektien koordinaatteja keskenään käydesään arrayta läpi.

Toinen isompi muutos oli se, että scripti kirjoittaa luvut suoraan json-tiedostoon, eikä niitä tarvitse jälkeempään formatoida käsin oikeaan formaattiin niin kuin ennen oli tehty. Tämä vaati melko paljon muutoksia tapaan, jolla scripti kirjoittaa tekstiä tiedostoon. Toteutuksessa päädyttiin luomaan paljon merkkijono muuttujia, joihin tallennettiin json-formaatissa käytettyjä merkkejä. Näin välttyttiin kirjoittamasta kaikkea manuaalisesti, kuten kuvasta 1 näkyy. Tämän jälkeen teksti kirjoitettiin tiedostoon format-funktiota käyttäen (kuva 2).

```

curlyB1 = "{"
curlyB2 = "}"
hottispotti = "\"hotspots\":"
globesVal = "\"Globes\":"
globeVal = "\"Globe\":"
fromVal = "\"from\":"
toVal = "\"to\":"
xVal = "\"x\":"
yVal = "\"y\":"
zVal = "\"z\":"
brace1 = "["
brace2 = "]"
floorVal = "\"floor\":"
format "%\n" curlyB1 to:fs
format "%\n" hottispotti to:fs
format "%\n" brace1 to:fs

format "%\n" curlyB1 to:fs_globes
format "%\n" globesVal to:fs_globes
format "%\n" brace1 to:fs_globes

```

KUVA 1 string -eli merkkijono muuttujien luonti ja kirjoittaminen filestream muuttuun. Tekijä: Mikko-Oskari Laitinen

```

kokoNimi = obj.name
erotellut = FilterString kokoNimi ", "
realName1 = erotellut[1]
realName2 = erotellut[2]
--print realName1
--print realName2
if o != num then (
  format " {\n\t% \"%\",\n\t% \"%\",\n\t% %, \n\t% %, \n\t% %\n },\n" fromVal realName1 toVal realName2 xVal x yVal y zVal z to:fs
) else (
  format " {\n\t% \"%\",\n\t% \"%\",\n\t% %, \n\t% %, \n\t% %\n }\n" fromVal realName1 toVal realName2 xVal x yVal y zVal z to:fs
)

```

KUVA 2 format-funktio käyttää valmiiksi luotuja string-muuttujia. Tekijä: Mikko-Oskari Laitinen

Kolmantena uutena toiminnallisuutena scriptin piti ottaa huomioon kerros, jossa kamera sijaitsee, jotta se osaisi kirjoittaa kerroksen numeron json-tiedostoon.

Tämä toteutettiin niin, että 3ds Max scenessä kameran nimen eteen kirjoitetaan numero, joka indikoi kerrosta ja erikoisvälimerkki (¤), jota scripti käyttää kerroksen erotteluun muusta nimestä. Myöhemmin koodissa tarkastetaan, löytyykö välimerkkiä helper-objektin nimestä käyttäen MatchPattern-funktiota, näin otetaan huomioon tapaukset, joissa kerroksia on vain yksi, jolloin välimerkkiä ei tarvita nimessä ollenkaan ja kerrokset ovat oletuksella 0. Jos kerrosmerkki löytyy, nimi puretaan kahteen osaan käyttäen FilterString-funktiota, jolla string muuttuja halkaistaan tietyn merkin kohdalta ja osat palautetaan arrayna. Näin kerros on eritellyn arrayn ensimmäinen alkio ja objektin nimi toinen alkio. Alla oleva kuva 3 havainnollistaa edellä mainittua koodia.

```

kokoNimi = obj.name
--- onko nimessä kerrosindikaattori merkki
if (MatchPattern kokoNimi pattern:"*¤*") == true then (
  realGlobeName = (FilterString kokoNimi "¤")[2]
  kerros = (((FilterString kokoNimi "¤")[1]) as integer) - 1
) else (
  realGlobeName = kokoNimi
  kerros = 0
)

```

KUVA 3 kerroksen tarkastelu käyttäen FilterString ja MatchPattern funktioita käyttäen. Tekijä: Mikko-Oskari Laitinen

Neljännessä vaiheessa scripti tarvitsee yhden objektin, joka vastaa esim. talon rajoja. Tähän tarkoitukseen toimii esimerkiksi yksinkertainen box objekti, joka

luodaan niin että se vastaa talon rajoja. Sitten scripti hakee kyseisen box-objektin suurimman x y ja z koordinaatin ja kirjoittaa ne tekstinä ulos. Tämä tehdään siksi, jotta kuvapisteeet asettuisivat oikealle kohdalle myöhemmässä vaiheessa.

3.2.2 Balcony extruder scripti

Tämän scriptin tavoitteena oli tehdä parvekkeiden mallintamisesta helpompaa ja nopeampaa.

Scripti hakee olemassa olevasta scenestä valmiita spline-objekteja, jotka toimivat ikään kuin parvekkeen poikkileikkauksena ja käyttää MAXScriptin sisäänrakennettua merge toimintoa tuodakseen ne nykyiseen sceneen, kuten kuvasta 4 voidaan nähdä. (Autodesk & Roger Cusson. 2009a)

```
>mergeMaxFile "filepath" #select #MergeDups #useSceneMtlDups  
>--set active layer  
>(LayerManager.getLayerFromName "parveke_BE").current.=.true  
>parvekeSelection.=.selection.as.array
```

KUVA 4 Scripti tuo Merge-toimintoa käyttämällä toisesta scenestä objekteja nykyiseen sceneen. Tekijä: Mikko-Oskari Laitinen

Tämän jälkeen scripti tarkastaa valituista parvekkeen lattian reunan edgeistä eli polygonien reunoista mihin suuntaan parveke tulee osoittamaan ja kuinka leveä se on. Siten scripti osaa 3ds Max ohjelman extrude-toimintoa käyttäen venyttää parvekkeen oikealle leveydelle ja oikeaan suuntaan. (Autodesk & Roger Cusson. 2009b)

Myöhemmässä vaiheessa lisätään osia parvekkeeseen manuaalisesti luomalla ne box konstruktoria kutsumalla (Autodesk, 2007j). Toisin sanoen luodaan uusi laatikko objekti ja siirretään se oikealle paikalle, jonka jälkeen se vielä kloonataan instancena (Autodesk, 2007k), jonka jälkeen kloonit siirretään oikealle paikalleen. Valmiiksi mallinnettuja parvekkeen osia myös tuodaan jo olemassa olevasta scenestä käyttäen taas 3ds Max ohjelman edellä mainittua merge toimintoa.

Tästä scriptistä tehtiin kaksi eri versiota, joista toinen luo highpoly-objektin, jossa on paljon polygoneja, mikä tekee siitä aidomman näköisen. Toinen versio luo lowpoly-objektin, jossa polygonien määrä on huomattavasti pienempi, näin sitä voidaan käyttää reaaliaikaisissa visualisoinneissa (Autodesk, 2006b).

3.2.3 XML scripti

Tämän scriptin tavoitteena oli kirjoittaa 3ds Max-scenestä dataa xml-tiedostomuotoon hyvin samaan tapaan kuin aiemmin esitellyssä hotspot scriptissä. Tämän johdosta kirjoittaja pystyi käyttämään monia aikaisemmin oppimiaan asioita ja osaa vanhasta koodista hyödykseen.

Tässä toteutuksessa käytettiin Windowsin XML .NET rajapintaa, johon pääsee käsiksi MAXScriptillä (Autodesk, 20071). Tämä siitä syystä, että näin voidaan käsitellä dataa xml-tiedostona, jossa data on pilkottu nodeiksi kutsuttuihin palasiin, joiden hallintaan sen toiminta perustuu. Muuten jouduttaisiin kirjoittamaan string-eli merkkijonomuuttujat manuaalisesti ja lisäksi vielä formatoimaan ne oikein.

.NET rajapinnan toteutus MAXScriptissä oli vaikea käyttöinen, sillä sen toiminta oli melko huonosti dokumentoitua. Eniten tietoa sen toiminnasta löytyi kolmansien osapuolien kautta erilaisista verkkolähteistä (CGSociety.com, 2022).

Kirjoittaja käytti referenssinä ja ohjenuorana .NET rajapinnan xml-toiminnallisuuksien käyttöön suunniteltua MAXScript-konstruktoria, joka löytyi paljon erilaisia MAXScriptejä sisältävältä verkkosivulta. Tällä tavalla .NET luokkien kutsuminen suoraan MAXScriptistä helpottui huomattavasti. (Scriptspot.com, 2022.)

Tässä toteutuksessa käytettiin paljon vektoreita, sillä scriptin piti pystyä laskemaan kulmia niiden välillä. Tekijä loi funktion, joka laskee sille parametreinä annettujen kahden vektorin välisen kulman ja palauttaa tuloksen, käyttäen vektorien pistetuloa ja arkuskosinia (kuva 5).


```
fn.GetVectorsAngle.v1.v2.=
(
theAngle.=.acos(dot.(normalize.v1).(normalize.v2))
)|
```

KUVA 5 kahden vektorin välisen kulman laskeminen luodussa funktiossa. Tekijä: Mikko-Oskari Laitinen

3.2.4 Rotate 90 scripti

Tämä scripti kääntää objektia, groupia(lähde) eli ryhmää objekteja tai editable poly objektin elementtiä 90 astetta z akselilla eli horisontaalisti. Tämän jälkeen scriptin voi suorittaa esimerkiksi pikanäppäintä painamalla. Tämä nopeuttaa työkentelyä paljon, sillä tarve kääntää scenessä olevaa objektia tasan 90 astetta on paljon käytetty toiminto.

3ds Max käyttää editable poly-objektissa viittä eri tasoa, jolla sitä voi muokata. Ensimmäinen näistä on vertex- taso, jossa käyttäjä voi muokata editable poly-objektin verteksejä eli koordinaattipisteitä, jotka muodostavat objektin 3d tilassa. Toinen taso on edge -eli reuna taso, yksi edge muodostuu kahdesta verteksistä piirtäen näin viivan. Kolmas, border -eli rajataso on kokoelma edgejä jotka muodostavat rajan ne ovat usein looppeja jotka eivät ympäröi facea. Esimerkiksi jos luodaan sylinteri ja siitä poistetaan päistä facet niin molempiin päihin muodostuu raja. Neljäs taso on face -eli pintataso pinnat muodostuvat 3d sovelluksissa kolmesta tai enemmästä verteksistä. Viides taso on element -eli elementtitaso, joka on kokoelma pintoja. (Autodesk, 2006c.)

Scriptiä toteutettaessa huomattiin, että 3ds Max ohjelman elementtiä kääntääkseen piti ensin purkaa elementistä pinnat, jotka muodostivat sen. Tämän jälkeen purettiin vielä pinnoista verteksit, jotka muodostivat ne. Tämän jälkeen ne voidaan valita ja siirtää jokaista objektin verteksiä niin, että elementti pyörii akselinsa ympäri. Oikeastaan elementti ei siis scriptissä käänny rotate toiminnolla akselinsa ympäri. Elementtejä voi kääntää normaalisti oikein, jos niitä kääntää manuaalisesti sovelluksen vakio-toiminnallisuuksiin kuuluvalla rotate -eli käännä toiminnolla. (Autodesk & Cusson R. 2009c.)

4 POHDINTA

Opinnäytetyössä toteutettiin MAXScript kielellä kirjoitettuna kaksi laajempaa kokonaisuutta ja joitain pienempiä scriptejä. Myös mahdollista uusien scriptien tarvetta kartoitetaan.

Toimeksiantajan päätöksellä opinnäytetyön tekijä ei voi liittää opinnäytetyössä toteutettujen scriptien lähdekoodia opinnäytetyöhön kokonaisuudessaan. Tekijä kuitenkin pystyi käyttämään pieniä osia koodista kuvankaappauksen muodossa havainnollistamaan niiden toimintaa paemmin.

Scriptien toteutus vei melko paljon aikaa, sillä käytetty MAXScript-ohjelmointikieli oli kirjoittajalle ennestään tuntematon ja käytännön koodaamisesta oli kulunut melko paljon aikaa. Eniten ongelmia itse scripteissä aiheutti se, ettei kirjoittaja osannut ottaa huomioon inhimillisiä virheitä tarpeeksi hyvin. Esimerkiksi yksinkertaiset kirjoitusvirheet objektien nimissä aiheuttivat ongelmia monissa scripteissä, jotka toimivat kyllä oikein, mutta käyttäjän kirjoitusvirheen takia ne hajosivat. Jälkeenpäin ajatellen olisi ollut viisasta koodata asiat niin, että käyttäjän virheen mahdollisuus minimoituu tai otetaan vähintäänkin huomioon.

Monet scripteistä vaativat vielä jatkokehitystä, esimerkiksi edellä mainittua käyttäjän virheen minimointia ja mahdollisesti lisää toiminnallisuuksia. Myös koodin luettavuudessa ja dokumentaatiossa olisi jonkin verran parannettavaa mahdollista myöhempää scriptien kehittäjää varten.

Scriptien toteutuksissa olisi myös voitu käyttää myös python-ohjelmointikieltä, sillä se on nykyään tuettu 3ds Maxissa. Tekijä kuitenkin päätti tätä vastaan kattavan 3ds Max ohjelman python tuen toteutuksen dokumentoinnin puutteen johdosta. Myös se fakta, että työpaikalla jo aikaisemmin kirjoitetut scriptit oli toteutettu MAXScript ohjelmointikielellä vaikutti tekijän päätökseen käyttää sitä toteutuksissa muiden kielten sijaan. Toisaalta python ohjelmointikieli on muissa yhteyksissä hyvin yleisesti käytetty kieli, joten siitä olisi todennäköisesti löytynyt paljon enemmän tietoa kuin MAXScriptistä.

Jos tavoitteena olisi kehittää tässä opinnäytetyössä esiteltyjä toteutuksia laajempia, mahdollisesti kaupallisia kokonaisuuksia, olisi ne parempi tehdä ainakin osin käyttäen 3ds Maxin SDK:ta ja C++ kieltä tai muita alemman tason kieliä hyväksi käyttäen, sillä näin niistä saisi suorituskykyisempiä ja helpommin laajennettavia kokonaisuuksia.

Opinnäytetyön tuloksena syntyi MAXScript scriptejä, joita työnantaja pystyy hyödyntämään heti. Myös tuotettu lähdekoodi ja scriptien dokumentaatio on heillä, joten niiden avulla voi scriptien toiminnallisuutta parantaa tai muokata hyvin helposti.

LÄHTEET

Autodesk & Roger Cusson. 2009a Learning Autodesk 3ds Max Design 2010 Essentials. Focal Press. e-kirja. sivu 67. Merging. Luettu 29.6.2022

Autodesk & Roger Cusson. 2009b Learning Autodesk 3ds Max Design 2010 Essentials. Focal Press. e-kirja. sivu 347. Shape Modifiers: Extrude. Luettu 29.6.2022)

Autodesk & Roger Cusson. 2009c Learning Autodesk 3ds Max Design 2010 Essentials. Focal Press. e-kirja. sivut 95–96. Transforms. Luettu 29.6.2022

Autodesk. 2006a. 3ds Max 8 MAXScript Essentials. Focal Press, 2006. e-kirja. sivu 199. Writing to files. Luettu 21.3.2022

Autodesk. 2006b. 3ds Max 9 Essentials. Focal Press, 2006. e-kirja. sivu 190. Low-poly Modeling. Luettu 20.5.2022

Autodesk. 2006c. 3ds Max 9 Essentials. Focal Press, 2006. e-kirja. sivut 193–194. Sub-object levels. Luettu 20.5.2022

Autodesk. 2007a. 3ds Max Maxscript Essentials (Second edition). Focal Press, 2007. e-kirja. sivut 6–7. MAXScript vs. Plug-ins. Luettu 29.6.2022

Autodesk. 2007b. 3ds Max Maxscript Essentials (Second edition). Focal Press, 2007. e-kirja. sivu 1. Introduction. Luettu 29.6.2022

Autodesk. 2007c. 3ds Max Maxscript Essentials (Second edition). Focal Press, 2007. e-kirja. sivut 22–26. Macro Recorder. Luettu 29.6.2022

Autodesk. 2007d. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 12–13. Variables. Luettu 29.6.2022

Autodesk. 2007e. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 26–36. Functions. Luettu 29.6.2022

Autodesk. 2007f. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 47–54. Collections. Luettu 29.6.2022

Autodesk. 2007g. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 39–41. Conditional Statements. Luettu 29.6.2022

Autodesk. 2007h. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 42–46. Loop Structures. Luettu 29.6.2022

Autodesk. 2007i. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 106–123 Understanding Objects and Classes. Luettu 29.6.2022

Autodesk. 2007j. 3ds Max MAXScript Essentials (Second edition). Focal Press. 2007. e-kirja sivut 54–59. Structures. Luettu 29.6.2022

Autodesk. 2007k. 3ds Max Maxscript Essentials (Second edition). Focal Press, 2007. e-kirja. sivut 94–98. Copy, Instance, and Reference. Luettu 29.6.2022

Autodesk. 2007l. 3ds Max Maxscript Essentials (Second edition). Focal Press, 2007. e-kirja. sivut 235–285. ActiveX and .NET (dotNet). Luettu 21.3.2022

Autodesk. 2015. History of Autodesk 3ds Max. Arkistoitu Verkkosivu. Luettu 19.3.2022. <https://web.archive.org/web/20151024145611/http://area.autodesk.com/maxturns20/history>

Autodesk. 2021. About MAXScript. Verkkosivu. Luettu 19.3.2022. <https://knowledge.autodesk.com/support/3ds-max/getting-started/caas/CloudHelp/cloudhelp/2022/ENU/3DSMax-Getting-Started/files/GUID-FE40F021-5444-4709-BF08-DF1F1F0960C3-htm.html>

Autodesk. 2022a. Autodesk App Store. Verkkosivu. Luettu 19.3.2022. <https://apps.autodesk.com/3DSMAX/en/Detail/In-dex?id=2039551202535585769&appLang=en&os=Win64>

Autodesk. 2022b. 3ds Max SDK Help Overview: MaxScript or C++. Verkkosivu. Luettu 23.7.2022. https://help.autodesk.com/view/MAXDEV/2022/ENU/?guid=Max_Developer_Help_overview_overview_maxscript_or_c_html

CGPress. 2011. Copitor. Arkistoitu verkkosivu. Luettu 19.3.2022. <https://cgp-ress.org/archives/copitor.html>

CGPress. 2016. FloorGenerator 2.0 available. Arkistoitu verkkosivu. Luettu 19.3.2022. <https://cypress.org/archives/floorgenerator-2-0-available.html>

CGSociety.com. 2022 Verkkosivu. Luettu 29.8.2021 <https://forums.cgsociety.org/c/autodesk/3dsmax-sdk-and-maxscript>

David Mackenzie. Scriptspot.com. 2022. Verkkosivu. Luettu ja ladattu 29.8.2021 <http://www.scriptspot.com/3ds-max/scripts/simple-xml-library-for-3dsmax>

Mottle, J. 2021. Architectural Visualization Rendering Engine Survey Results. Luettu 13.6.2022. <https://www.cgarchitect.com/features/articles/a9976335-2020-architectural-visualization-rendering-engine-survey-results>