Tallinna Ülikool
Digitehnoloogiate Instituut

Blender 3D mudelite loomine Unity keskkonnas kasutamiseks

Seminaritöö

Autor: Vladislav Minajev
Juhendaja: Andrus Rinde

Tallinn 2017

Sisukord

Sissejuhatus

Unity on üks enimkasutuim tasuta mängude arendamise keskkondi maailmas, millega saab luua erinevaid 2D ja 3D mänge. Unity'l on palju sisseehitatud võimalusi mängu elementide ja 3D objektide loomiseks. Paljud inimesed aga otsustavad kasutavad mängu objektide loomiseks spetsiaalseid 3D programme, kus võimalusi ja töövahendeid on veelgi rohkem ja kus neid on mugavam kasutada.

Üks tuntuimaid vabavaralisi 3D programme, kus on hulgaliselt funktsioone, võimalusi ja töövahendeid on Blender. Lisaks on Blenderi kohta palju juhendmaterjali. Blenderi abil loodud 3D mudeleid saab importida Unity keskkonda ja kasutada loodavates mängudes. Loodud mudelite importimine Unity keskkonda on üldiselt lihtne kuid mõnikord esineb probleeme.

Käesoleva seminaritöö Eesmärgiks on anda soovitusi, kuidas Blender tarkvaraga loodud 3D mudeleid edukalt Unity keskkonda importida. Eesmärgi saavutamiseks annab autor lühiülevaate Unity ja Blender tarkvarast. Autor loob Blenderis 3D mudelid ja impordib need Unity keskkonda ning annab ülevaate, millised probleemid võivad esineda ja pakub välja võimalikke lahendusi, et tulemus oleks positiivne.

Käesolev töö on jagatud kaheks osaks. Esimene osa annab ülevaate käsitletavatest programmmidest. Töö teine osa kujutab endast praktilist ülesannet, mille tulemusena valmib 3D mudel Blender keskkonnas. Teises osas käsitletakse ka võimalike probleemide leidmise ja nende lahendamisega lugejatele järgitaval moel.

See töö on mõeldud inimestele, kes on varasemalt Blender ja Unity tarkvaradega kokku puutunud ja kellel on eelnevalt omandatud algteadmised ja baasoskused 3D modelleerimisest.

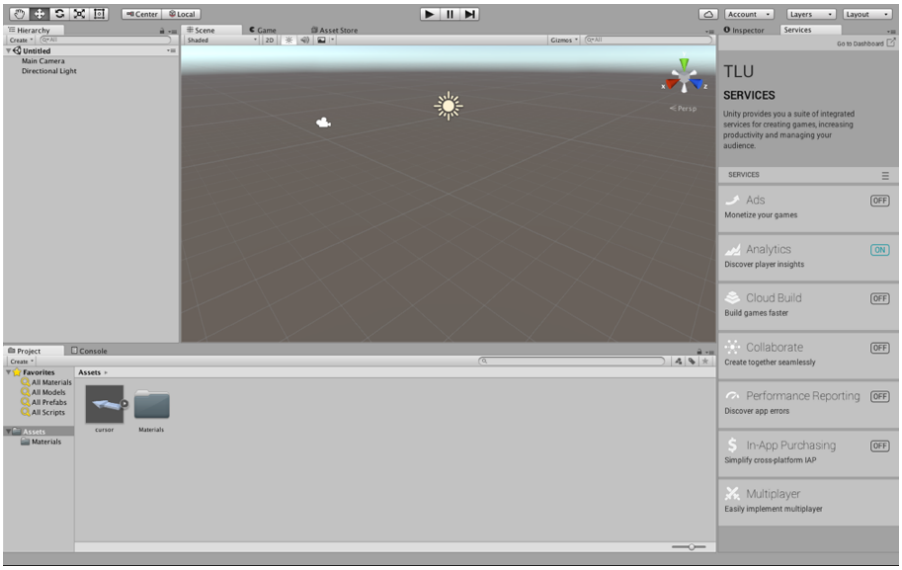
1. Tarkvara ülevaade

Kaasaegsed arvutimängud on enamuses ruumilised – kasutavad 3D mudeleid. Mudelite loomiseks on tänapäeval lugematu arv erinevaid tarkvarasid. Antud juhul me hakkame vaatlema kaht tarkvara, mida kasutatakse paljude kaasaegsete mängude arendamisel. On väga oluline, et need tarkvarad oleks üks teisega ühilduvad. Järgnevalt anname ülevaate mängude loomevahendist Unity ning sellele 3D mudelite loomiseks sobivat 3D modelleerimisprogrammi Blender.

1.1. Unity

Unity on loomevahend kahe- ja kolmemõõtmeliste videomängude või muusuguse interaktiivse sisu, näiteks animatsioonide või arhitektuurivisualisatsioonide loomiseks. Unity töötab operatsioonisüsteemidel Microsoft Windows ja Mac OS X ning loodud mängud töötavad Windows, OS X, Windows Phone, Android, Apple iOS, Linux, samuti mängukonsoolidel, Wii, Playstation, Xbox. Sellega saab luua ka veebimänge, mis kasutavad Unity veebimängija tarkvaramoodulit. (Haas, 2002)
Unity keskkonnas loodud rakendused toetavad DirectX ja OpenGL. Aktiivselt Unity kasutavad ka niisugused suuremad arendajad nagu Blizzard, EA, QuartSoft ja Ubisoft. Unity tarkvaral on lihtne kasutajaliides, mille on lihtne seadistada. See koosneb mitmest erinevast aknast, tänu sellele saab korrigeerida mängu otse kasutajaliidese kaudu. *Project Window* aknas saab näha kõike objekte, mis on võimalik kasutada jooksvad projektis. Lisaks seal või leida objekte materjale ja nende miniatüüre. *Scene View* võimaldab visuaalselt liikuda ja muuta oma stseeni. Stseenis saab näha 3D kui ka 2D projekte. Sõltub sellest, millise projektiga on tegu. *Hierarchy Window* kirjeldab tekstina iga objekti mis on stseenis. Igal objektil on olemas link stseeni koordinaatidele, sellega stseeni ja pingerida aknad on omavahel seotud. Paremal võib leida *Inspector Window.* See aken võimaldab muuta ja vaadata kõiki valitud objekti omadusi. Kuna igal objektil võival olla erinevad omadused ja kujundus, selle akna sisu kogu aeg erineb. Üleval asub *Toolbar* paneel. Selle sees võib leida põhilisi tööristu. Lisaks seal asuvad *play, pause* ja *step control* nuppud. Paremal on kasutaja menüü, gruppid(*layers)* ja aknade paigutuse variandid.
Mootor toetab kahte programmeerimiskeelt: C# ja JavaScript. Füüsika arvutused teeb füüsika mootor PhysX NVIDIA. (Haas, 2002)

3D mudelite loomiseks on Unity's primitiivsed vahendid. Luua saab kerasid(*sphere*), kuubikuid(*cube*) ja plaate(*plane*) ning neid saab oma soovi järgi suurendada ning pöörata. Kui kasutaja soovib luua näiteks mängutegelast, siis üldiselt Unity võimalustest jääb väheks, sest puudub keerukamate modelleerimise protsesside tegemise võimalus. Selle tõttu ei sobi Unity keerukamate 3D mudelite tegemiseks ja tarvidus on modelleerimisele keskendunud programmi järele. (Haas, 2002)



Joonis 1 – Unity programmi töökeskkond

1.2. Blender

Blender on tasuta ja avatud lähtekoodiga 3D arvutigraafika tarkvara mittetulundusühingult Blender Foundation. Seda kasutatakse animatsioonide, visuaalsete efektide, 3D mudelite, interaktiivsete 3D rakenduste ja videomängude loomiseks.

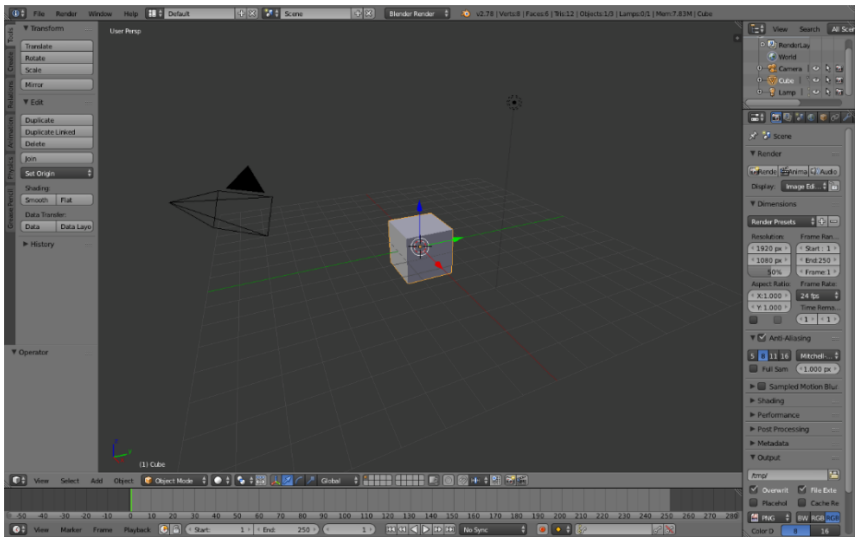
Tänaseks on Blenderi funktsionaalsus võrreldav konkureerivate kommertstarkvaradega, olles seejuures tasuta. See võimaldab väikestuudiotel oluliselt vähendada enda kulusid tarkvarale.

Blender'i funktsioonide hulka kuuluvad 3D-modelleerimine(*3d modelling*), tekstuurimine(*textures*), rastergraafika toimetamine(*raster graphics*), vedeliku ja suitsu simulatsioon(*liquid and smoke simulation*), osakeste simuleerimine(*particle simulation*), pehme keha simulatsioon(*soft body simulation)*, mängu liikuv kaamera positsioneerimine(*game moving camera positioning*), video monteerimine(*video mounting*). Samuti on Blender'is sisseehitatud mängumootor(*game engine*). (Gordon, 2009)

Blender'i kasutajaliides on algaja jaoks alguses ebamugav. Peaaegu igal funktsioonil on oma vastav klahvikombinatsioon. Esmalt oli Blender mõeldud firmasiseseks kasutamiseks. Pärast seda, kui Blender'i lähtekood avalikustati muutusid tööriistad kasutajasõbralikumaks ja paindlikumaks. Lisaks sellele tehti sõbralikkumaks kasutajaliides, lisati värvilahendused, objektide puu ja erinevad väikesed muudatused. Igal kasutajal on võimalik kasutajaliidest vastavalt oma soovidele ümber kohandada. (Gordon, 2009)

Blender toetab palju erinevaid failiformaate: Collada(.dae), Alembic(.abc), 3D Studio(.3ds), FBX(.fbx), Standford(.ply), Wavefront(.obj), X3D Extensible 3D(.x3d), Stl(.stl). Eelnevalt loendatud formaatidest on Unity jaoks kõige sobivam FBX(.fbx). (Gordon, 2009)

Selle seminaritöö raames kasutab autor Blender'i versiooni 2.8.



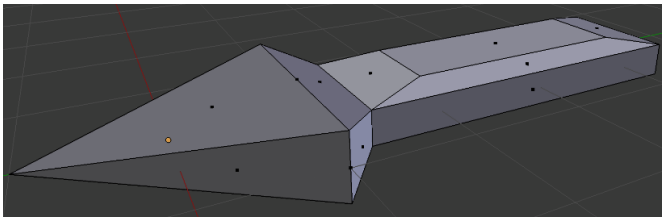
Joonis 2 – Blender programmi töökeskkond

2. Blender 3D mudeli loomine ja ümbertõstmine Unity keskkonda

Selles peatükis kirjeldab autor Blender'is loodud 3D mudelite importimist Unity keskonda. Autor vaatleb, millised probleemid võivad esile kerkida ning kuidas neid lahendada.

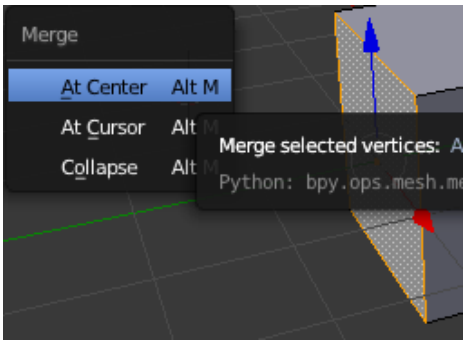
2.1. Mudeli loomine

Esmalt loob autor mudeli, mille kuju võimaldab aru saada, mis pidi mudel on. Tavaline kuup selleks näiteks ei sobi. Autor valis selleks mudeliks noole. Vaadates noolele on kohe näha kuhu ta suunatud on. Lisaks on tehtud noolele väike katus, et oleks võimalik vahet teha ülemisel ja alumisel mudeli poolel. Noole modelleerimist alustab autor primitiivsest kujundist – kuubist.



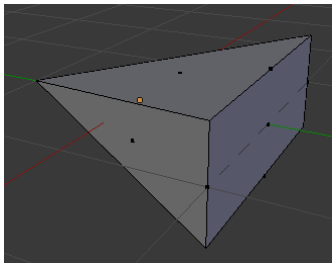
Joonis 3 – Blender – Valmis mudel

Kuubist teeb autor esmalt püramiidi kujulise objekti kasutades funktsioon *Merge At Center*. Tulemuseks tekib noole teravik.



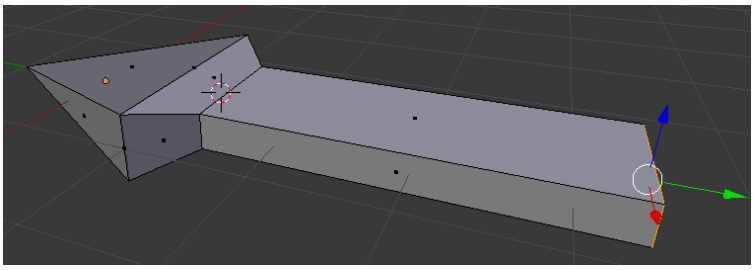
Joonis 4 – Blender – Merge menüü

Muudame oma loodud objekti suurust *scale* käsuga.



Joonis 5 – Blender – Scale tulemus

Pärast seda teeme noorevarre. Venitame ühe külje välja, teeme selle kõrguse väiksemaks ja seejärel lisame pikkust. Tulemus peaks olema järgnev.

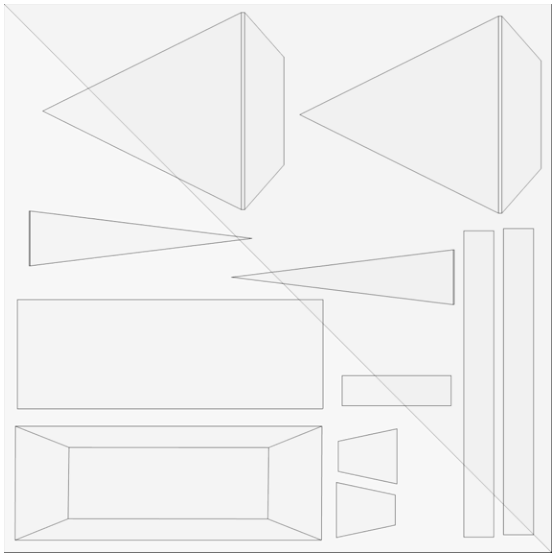


Joonis 6 – Blender – mudeli tulemus

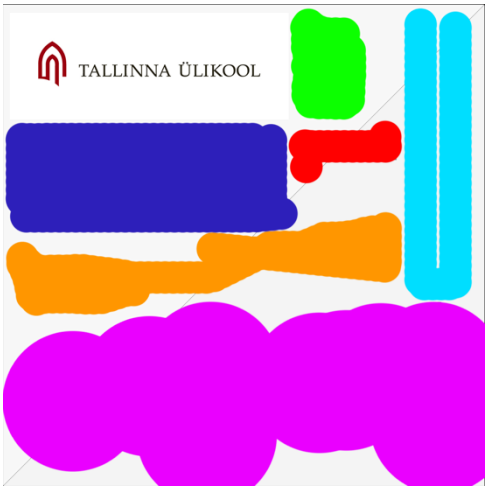
Noole pealmise külje eristamiseks valime noole varre pealmise tahu ja pikendame seda. Seejärel teeme selle tahu väiksemaks ja tõstame kõrgemale. Nüüd on mudelil võimalik eristada pealmine soovitud pealmine külg ja eesmine ots. (Hughes, 2016)

Lisame meie mudelile tekstuuri. Alguses lõikame meie nool, selleks et teha UV-mappingu ja selle abil panna meie tekstuur mudelile. Kui küljed on lõigatud, teeme UV-map ja paigutame meie pinnad nii, et oleks mugav neid värvida. Kui pinnad on paigutatud, ekspordime meie UV-map *.png* formaadis. (zay116, 2013)

Loodud *.png* faili saab värvimiseks avada mistahes pilditöötlusprogrammiga(näiteks Gimp, Photoshop vms).

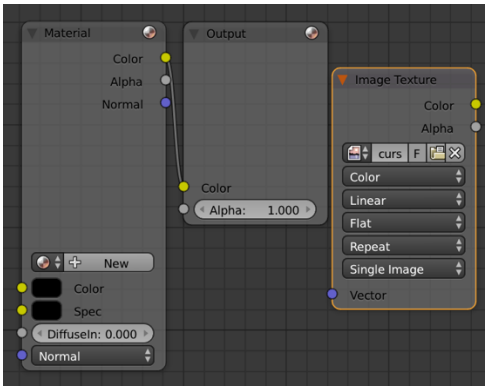


Joonis 7 - Eksporditud Uv-map



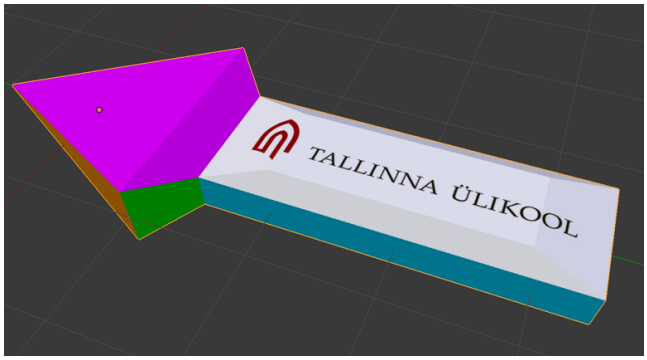
Joonis 8 - värvitud UV-map

Pärast seda meil tuleb võtta kasutusele meie uus ja värvitud UV-map. Selleks kasutame *Node Editor* akna.



Joonis 9 - Blender - Node Editor aken

Algusese tekitame meie mudelile uus materjal. Päras *Node Editor* aknas lisame uue *Image Texture* ala. Seal avame meie värvitud UV-map faili ja aknas, kus on meie mudel, lülitame sisse *Texture Viewport Shading.* Nagu on näha meie mudel nüüd omab uut tekstuuri.



Joonis 10 – Blender – Mudel koos tekstuuriga

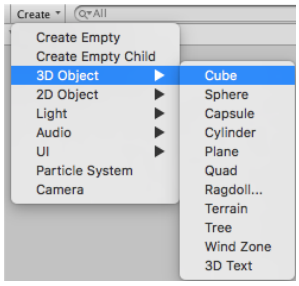
Mudel on valmis, nüüd tuleb see sobivas formaadis salvestada.
Selleks formaadiks me valime *FBX(.fbx).* See formaat on praeguse näite puhul sobilik seetõttu, et Unity toetab seda.

2.2. Probleemid 3D objektide ületoomisel Blender'ist Unity'sse

Blenderis loodud mudelite Unity keskkonda üle toomisel võib tekkida mitmeid probleeme. Objektide suurus võib muutuda, sest Unity ja Blender kasutavad vaikimisi erinevaid mõõtühikuid. Samuti on erinev objektide orientatsioon, sest kasutuses on erinevad koordinaatide süsteemid. Mudeli üle toomisel erines ka mudeli tippude arv.

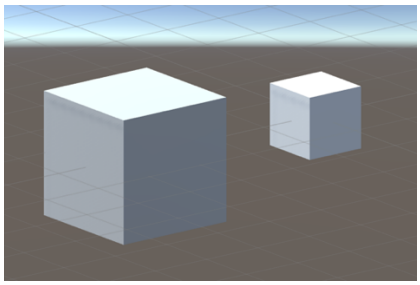
2.2.1. Suurus

Esimene probleem, mis autoril tööd kirjutades ette tuli oli probleem objekti suurustega. Blender programmis oli mudel ühe suurusega, aga Unity keskkonnas oli see teise suurusega.
Illustreerime seda probleemi luues mõlemas programmis kuubi, servapikkusega 1. Ekspordime Blender standardse kuubi *FBX* formaadis ja tõstame selle Unity'sse.



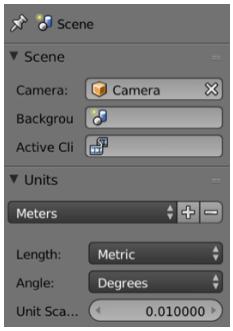
Joonis 11 – Unity – mudeli loomise menüü

Tulemus on järgnev – Blenderis loodud kuup on suurem kui Unity's loodud kuup.



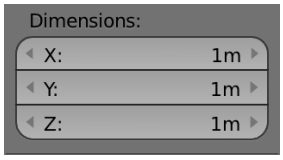
Joonis 12 – Unity – Ekspodi tulemus

Selleks et parandada meie kuupide suuruste vahe muudame mõõtühikute süsteem Blenderis. Vaikimisi Blender ja Unity programmides on pandud *Units* aga igal tarkvaral nad on omapäraselt tõlgitud. Selle pärast enne ekspordimist muudame valminud Blenderi mudel ja muudama mõõtühikuid *Metric* süsteemi. Paneme *Unit System->Metric* ja *Unit Scale: 0.01*. (Pavel, 2015)

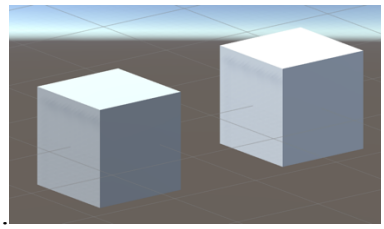


Joonis 13 – Blender – Sceeni seadistused

Tulemuseks me näeme, et standardne Blender'i kuubi külje pikkus on ainult 2 cm. Vajutades *N* klahvi kutsume objekti omaduste paneeli ekraanile. Muudame mudeli suurust 1-ks meetriks. Unity's vaikimisi 1 Unit võrdub 1 meetriga. (Pavel, 2015)



Joonis 14 – Blender – objekti suurusmääramise menüü

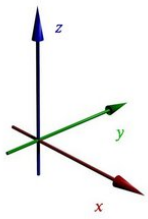
Pärast jälle salvestame meie mudeli *FBX* formaadis ja näeme et kuupide suurus on samasugune. 

Joonis 15 – Unity – lahendatud probleemi tulemus

Esimene tekkinud probleem sai lahenduse. Selleks, et ära hoida mudelite suuruste erinevused Blender'is ja Unity's tuleb enne modelleerimise alustamist seadistada Blender'is mudeli mõõtmed.

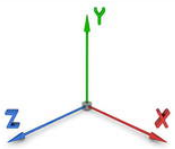
2.2.2. Mudelite orientatsioon

Võib ette tulla, et Blender'is loodud mudel (näiteks auto) on Unity'sse importides külili. See tekib sellepärast et nendes programmides on erinevad koordinaatide süsteemid. Blender programmis on parempoolne koordinaatide süsteem, kus Z-telg on suunatud üles. (Pavel, 2015)



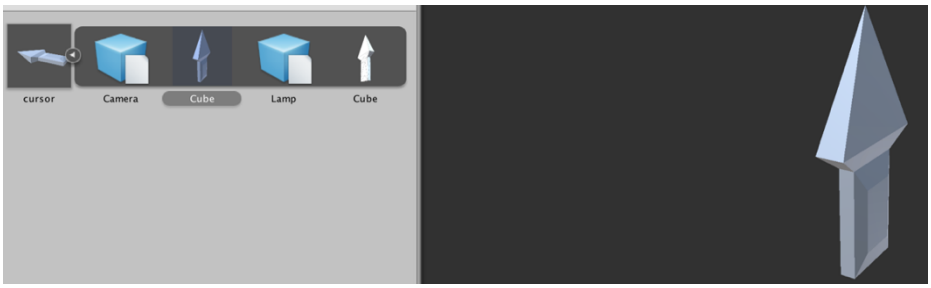
Joonis 16 – Blender – koordinaatide süsteem

Unity omakorda kasutab vasakpoolset koordinaatide süsteemi, kus Y on suunatud üles.



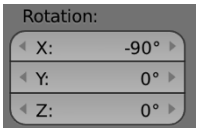
Joonis 17 – Unity – koordinaatide süsteem

Kasutame varasemalt loodud noole mudelit. Salvestame ja ekspordime selle Unity'sse. Objekti omaduste paneelis *Preview* aknas on näha, et valminud nool näitab üles. See tekkis selle pärast et nagu enne oli mainitud, nende programmide koordinaatide süsteemid on erinevad ja Blenderis meie mudel näitab Y-teljel. See sama telg Unity's on suunatud ülespoole. Selline pilt meile ei sobi ja tuleb parandada varem eksporditud mudel. (Pavel, 2015)



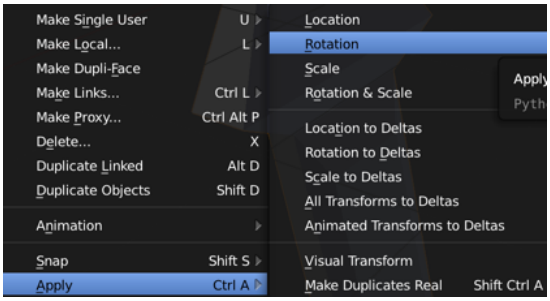
Joonis 18 – Unity – preview aken

Meie ülesandeks on paigutada meie mudel õiges suunas, et Unity's see näeks sama välja. Probleemi lahendamiseks tuleb mudel pöörata juba Blender'is ümber X-telje 90° kraadi.

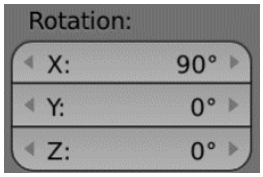


Joonis 19 – Blender – rotation menüü

Järgmise sammuna teeme meie *Object* menüüs *Apply-> Rotation* ja jälle pöörame meie mudelit 90° kraadi ümber X-telje.

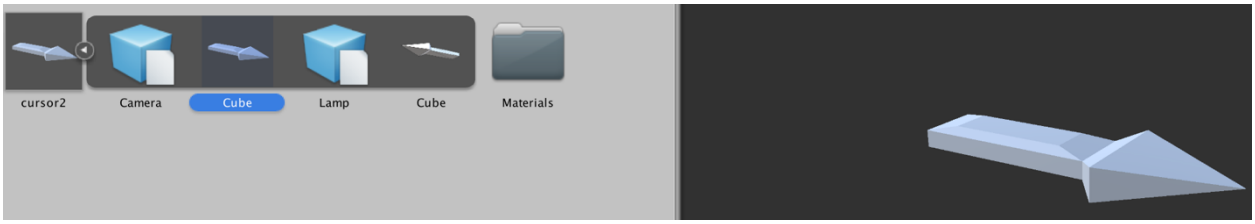


Joonis 20 – Blender – apply menüü



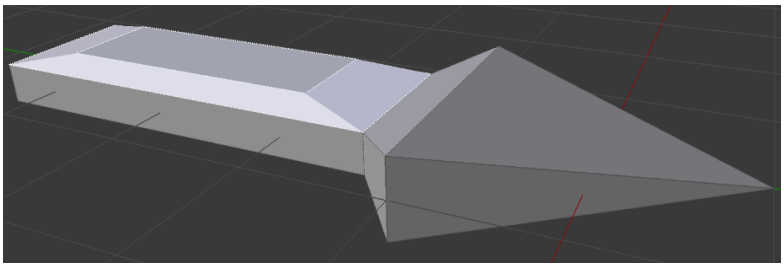
Joonis 21 - Blender – rotation menüü

Ekspordime meie mudeli uuesti *FBX* formaadis ja vaatame tulemust.



Joonis 22.1 – Unity – preview aken

Nüüd probleem on lahendatud, sest meie nool vaatab samas suunas nagu Blenderis.

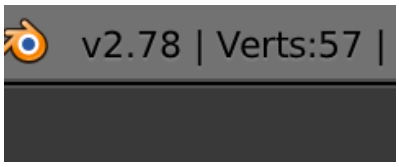


Joonis 23.2 – Blender - noole suund

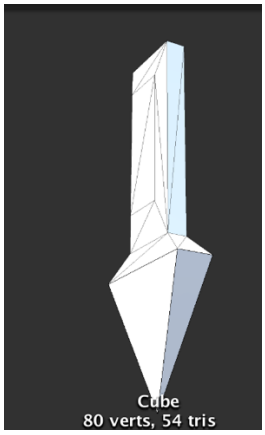
2.2.3. Tippude arv

Natuke uurides selgus, et Blender'is loodud mudelid omavad vähem tippe kui sama mudel eksporditud Unity's. See juhtub selle pärast et Unity lisab juba valmis mudelile ka tippe, mis tekkivad *UV Mapis*. Mudeli me lõikame selleks, et saaks selle mudeli värvida ja lisada sellele tekstuure. Tippude arvu suurendamisega kasvab koormus protsessorile. Kui neid mudeleid on palju, see võib kannatada mängu võimsust. Meie ülesanneks on selle arvu võimalikumalt väiksemaks teha. (Bunny83, 2011)

Näidiseks võtame meie alguses tehtud noole, salvestame ja toome üle Unity programmisse. Võrdleme tippude arvu.

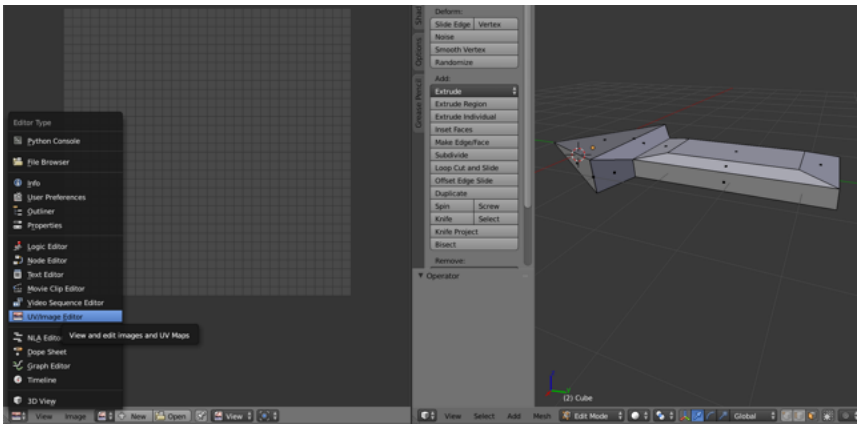


Joonis 24 – Blender – mudeli tippude arv



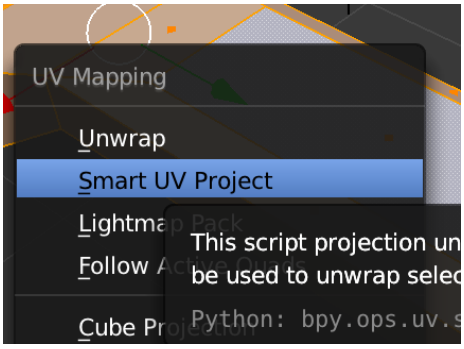
Joonis 25 – Unity – mudeli tippude arv

Nagu on näha, Blender programmis olev mudel näitab 57 tippu(*Verts)* aga Unity omal on juba 80 tippu. Vaatamata sellele, et see on üks ja sama mudel.
Tippude arvu võib mõjutada ka näiliselt mudeli keerukust mitte mõjutav värvide ja tekstuuride kasutamine. Mida keerulisem on meie mudel, seda rohkem on tippe Unity keskkonnas. Selleks et mudeli pinda erinevate tekstuuridega katta, tuleb luua *UV-map.*

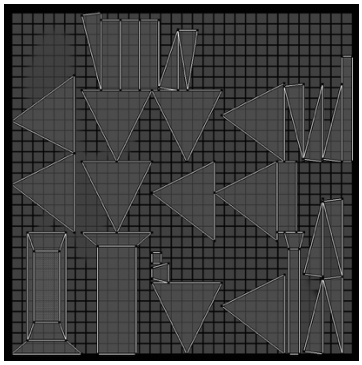


Joonis 26 – Blender – UV/Image Editor akna tekkitamine

Selleks et lõigata meie mudel, me kasutame *Smart UV Project* funktsiooni. Selleks valime meie mudel *Edit Mode* režiimis ja vajutame klahvi *U.* Tekkivas menüüs valime *Smart UV Project* ja kohe pärast seda näeme vasakus tööalas, et meil on tekkinud meie mudeli lõike.

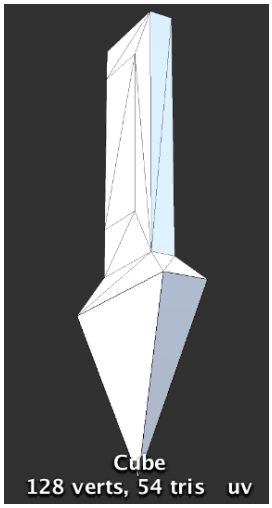


Joonis 27 – Blender – UV Mapping menüü



Joonis 28 – Blender – mudeli lõike

Nüüd veel kord ekspordime meie mudel *FBX* formaadis ja vaatame tippude arv mõlemates programmides. Blender nagu algusel näitab 57 tippu, aga Unity juba 128.



Joonis 29 – Unity – preview ake ja tippude arv

Sellest võib teha järelduse, et mida rohkem meil on servasid (*edge*) ja keerulisem on meie mudel, seda rohkem tekkib Unity keskkonnas tippe. Mida rohkem tippe, seda enam ressursse kulub graafikakaardil selle mudeli kuvamiseks. Lisaks kasutatakse ka palju protsessori võimsust.

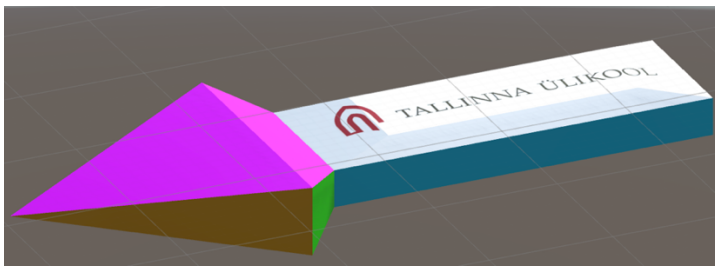
Kahjuks võimalust teha nii, et Blenderis ja Unitys tippude arv oleks sama ei ole. Ainuke lahendus on proovida selle arvu väiksemaks teha.

2.2.4. Tekstuur

Uurides igasugustes foorumites, ilmus probleem tekstuuridega. Paljudel kasutajatel tekkisid vead kui Blenderis tehtud mudel koos tekstuuriga eksportisid Unity keskkonda. Need Vaatame mis tuleb välja.

Varem eksporditud mudel avame Unity keskkonnas ja paname meie mudeli *Assets* kausta ja valminud materjaal sama kausta. Valime meie mudel ja inspektori aknas leiame *Mesh* ala. Pärast seda tassime meie materjal *Albedo* ruudule. Nüüd on mudel *Scene* akna paigutamiseks valmis.

Joonis 30 - Unity - materjaali paigutamine

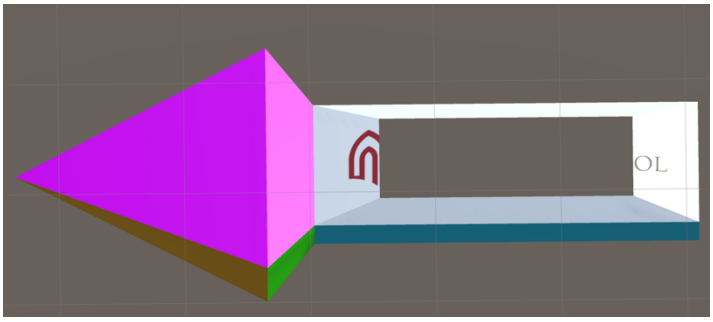


Joonis 31 - Unity - Mudel tekstuuriga

Nagu näeme probleeme tekstuuriga meil ei tekkinud. Siit võib teha järelduse, et see sõltub mudeli keerukusest ja tekstuuri keerukusest. Või on esinenud probleeme erinevate programmide versioonidega.

2.2.5. Pindade normaalid

Mõnikord võib tekkida selline olukord, et te olete teinud kõik sammud tekstuuri lisamisel õigesti, aga ikka ei näe enda tekstuuri pärast seda, kui mudel on eksporditud Unity'sse. See juhtub, sest mudeli loomisel on on normaalvektorid suunatud mudeli sisemusse. Blenderis seda ei ole hästi näha, pind muutub natukene tumedamaks, aga kui me ekspordime meie mudel koos materjaliga Unity'sse, siis tekkib auk.

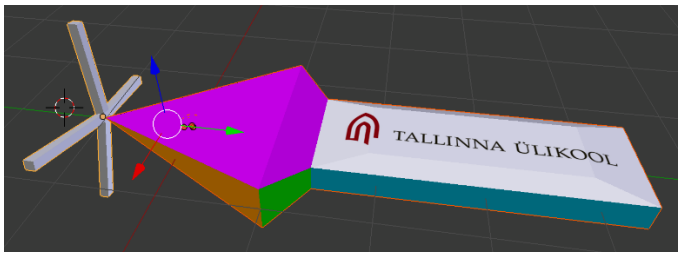


Joonis 32 - Unity - vale pinna suund

Selleks et lahendada see probleem tuleb Blenderis valida kõik meie mudeli pinnad ja nende normaalvektorid mudelist väljaspoole suunata. Selleks *Shading* menüüs valime *Normals:Recalculate.* Pärast seda tavaliselt kõik pinnad pöörduvad õigesse suunda.
Kui ei õnnestunud, siis võib valida eraldi üks pind ja vajutada *Flip Direction.* Pind peaks heledamaks muutma. (The_r0nin, 2010)

2.3. Animationsiooni loomine Blenderis, Unity keskkonnas kasutamiseks

Selleks et mäng oleks atraktiivsem peaavad paljud elemendid liikuma. Selleks tuleb neid animeerida. Järgnevalt kontrollib autor, kas on võimalik lisada mingeid tsüklilisi animatsioone mudelitele juba Blenderis selliselt, et need toimiks ka Unity keskkonnas. Seda võiks kasutada mingide stseenide objektide jaoks. Näiteks rohi, taevas, rattad, veskitiivad, mängutegelased jms.
Mudeliks me võtame taas meie varem valminud noole. Lisame meie noole otsa propeller ja paneme see pöörlema.
Kaadrite arvuks määras autor 100. Autor asetas valmis tehtud propeller mudeli otsa ja tegi 2 propelleri täispööret. Valminud animatsiooni tegi tsükliliseks ja ekspordis mudeli Unity keskkonnas kasutamiseks valmis. (Williamson, 2013)



Joonis 33 - Blender - Valmis mudel koos propelleriga

Avame meie mudel Unitys ja *Inspektor* aknas näeme et meie mudelil on olemas animatsioon. Vajutades *play* nuppu, me saame näha, kuidas meie mudel liigub.

Kokkuvõte

Käesolevas seminaritöös oli eesmärgiks anda ülevaade Blender ja Unity keskkonnast. Peamine eesmärk oli leida probleemid, mis võivad tekkida Blender mudelite kasutamisel Unity keskkonnas. Teha nende probleemide ülevaade ja proovida neid lahendada ja kirjutada kasutusjuhend nende lahendamiseks.
Töö käigus autor tegi ülevaade Blender tarkvarast, kus seda kasutavad, kes seda arendab ja tegi. Näitas Blender programmi töökeskkonna. Autor tegi ka Unity tarkvarast ülevaade, natuke ajalugu programmist ja kasutuskeskkonnad.
Järmiseks sammuks autor näitas kuidas luua lihtsam mudel Blender keskkonnas, kirjeldas iga samm ja näitas kuidas salvestada mudeli, selleks et seda tulevikus võiks kasutada Unity programmis.
Pärast seda oli valitud viis põhiprobleemi, mis võivad tekkida mudeli eksportimisel ja autor tõi enda töökäigus nende probleemide kirjeldus ja lahendus. Objektide suuruse probleem laheneb mõõtude süsteemi muutmisega Blenderis ja valmis mudeli suuruste korrutamine 0.01. Ainus probleem, mille lahendust ei leitud, on tippude arvu suurenemine Unity's. Autor tegi selgeks millisel olukorral tippude arv suureneb, ning tuleb võimalusel olla ettevaatlis objektidel keerukamate tekstuuride kasutamise, *UV-Mapping* abil.
Tektuuriga probleeme ei tekkinud, aga see võib sõltuda igasugustest faktoritest. Pindade normaalide probleem omab lihtne lahendus, lihtsalt tuleb jälgida, kuhu on suunatud pinnad.
Autor proovis ka teha valminud mudelile animatsioon ja vaadata, kas on võimalik ekspordida mudel koos Blenderis tehtud animatsiooniga Unity keskkonda.
Antud seminatiröö peaks aitama vältima vigu, nendel kes esimene kord proovib Blender keskkonnas valminud mudeleid kasutada Unity keskkonnas.

Kasutatud allikad

Bunny83. (29. aprill 2011. a.). *Vertex count 10 times higher in Unity* . Kasutamise kuupäev: november 2016. a., allikas Answers. Unity3d: http://answers.unity3d.com/questions/61597/vertex-count-10-times-higher-in-unity.html

Gordon, F. (2009). *Blender 3D Basics.* Albion, USA: Packt publishing.

Haas, J. (21. mai 2002. a.). *A History of the Unity Game Engine.* Kasutamise kuupäev: juuli 2016. a., allikas Worcester Polytechnic Institute: https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/unrestricted/Haas_IQP_Final.pdf

Hughes, K. (07. september 2016. a.). *38 brilliant Blender tutorials*. Kasutamise kuupäev: oktoober 2016. a., allikas Creative Bloq: http://www.creativebloq.com/3d-tips/blender-tutorials-1232739

Pavel. (5. aprill 2015. a.). *Секреты экспорта из Blender в Unity*. Kasutamise kuupäev: november 2016. a., allikas Хабрахабр: https://habrahabr.ru/post/254937/

The_r0nin. (18. november 2010. a.). *Blender Mesh Import issue* . Kasutamise kuupäev: detsember 2016. a., allikas Answers. Unity3d: http://answers.unity3d.com/questions/34572/blender-mesh-import-issue.html

Williamson, J. (29. mai 2013. a.). *How may I create a continuously looping animation?* Kasutamise kuupäev: detsemberw 2016. a., allikas Blender. Stack Exchange: http://blender.stackexchange.com/questions/440/how-may-i-create-a-continuously-looping-animation

zay116. (2013). *How to add a texture in Blender*. Kasutamise kuupäev: oktoober 2016. a., allikas instructables: http://www.instructables.com/id/How-to-add-a-texture-in-Blender/?ALLSTEPS