

Henri Forsström

# Korttikuvan kehitysprosessi mobiilipeliin

Tradenomi

Tietojenkäsittely

Kevät 2018



KAJAANIN  
AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

## Tiivistelmä

**Tekijä:** Forsström Henri

**Työn nimi:** Korttikuvan kehitysprosessi mobiilipeliin

**Tutkintonimike:** Tradenomi, tietojenkäsittely

**Asiasanat:** peligrafiikka, 3D-mallinnus, kehitysprosessi, mobiilipelit, videopelit

Opinnäytetyö perustui työtehtävään, jossa tarkoituksena oli luoda kehitysprosessi mobiilipelin Badland Brawl hahmojen korttikuvia varten. Tehtävällä oli suuri merkitys, sillä pelissä hahmot ja niiden korttikuvat olivat tärkeässä roolissa. Prosessin tavoitteena oli korvata vanha korttikuvien kehitysmenetelmä siten, että uusi prosessi olisi nopeampi ja resurssillisesti kevyempi. Tehtävään kuului myös korttikuvien uuden graafisen tyylin selvittely ja toteutus. Työssä keskityttiin vain yhteen pelin hahmon korttikuvaan.

Työn huolellisella suunnittelulla oli suuri merkitys toteutuksen ajankäytön ja sujuvuuden kannalta. Kehitysprosessin ja graafisen tyylin selvittelyä seurasi itse työn toteutus, joka koostui eri vaiheista: 3D-mallin valmistelu, scenen kehittäminen, materiaalien luominen ja lopulta kuvan renderointi peliä varten. Työssä käytettiin pääosin Blender-mallinnusohjelmaa.

Tehtävän lopputuloksena saavutettiin sopivampi korttikuvan kehitysprosessi sekä uusi graafinen tyyli useiden pienten iteraatioiden myötä. Prosessia käyttämällä uusien korttikuvien luominen on helppoa sekä nopeampaa, pitäen korttien graafisen tyylin yhtenäisenä.

## **Abstract**

**Author:** Forsström Henri

**Title of the Publication:** Development Process of a Card Image for a Mobile Game

**Degree Title:** Bachelor of Business Administration (UAS), Business Information Technology

**Keywords:** game graphics, 3D-modeling, development process, mobile games, video games

This thesis was based on a work assignment, in which the purpose was to create a development process for characters' card images for a mobile game Badland Brawl. The task was significant because the characters and their card images in the game played an important role. The objective of the process was to replace the old development process so that the new process would be quicker and less laborious resource-wise. The task also included researching and achieving a new graphical style for the card images. The process focused solely on one character's card image.

Thorough planning of the task was significant for the time usage and fluency of the process. The planning of the development process and graphical style was followed by the actual execution of the assignment, which consisted of different phases: preparing the 3D-model, developing the scene, creating the materials and eventually rendering the image for the game. Blender was used as the main software in the work process.

The task resulted in achieving a more suitable card image development process and a new graphical style through several small iterations. It is easy and faster to create new card images using the process while keeping the graphical style of cards coherent.

## Sisällys

1	Työn lähtökohdat.....	1
1.1	Frogmind Oy .....	1
1.2	Työn kohdepele Badland Brawl .....	1
1.3	Mobiilipeliteollisuus .....	2
1.4	Työtehtävä ja tavoitteet .....	3
1.5	Työn suunnittelu.....	4
1.6	Graafinen tyyli .....	6
2	Ohjelmisto.....	8
2.1	Blender työympäristö.....	8
2.2	Renderointimoottori.....	11
3	Mallin valmistelu.....	12
3.1	Modifier-muotoilutyökalut .....	12
3.1.1	Subdivision Surface .....	13
3.1.2	Bevel .....	14
3.1.3	Modifier-työkalujen käyttö hahmomallissa.....	15
3.2	Hahmon poseeraus.....	16
4	Scene.....	17
4.1	Valaistus .....	18
4.2	Valaistusmenetelmä .....	18
4.3	Kamera .....	20
5	Materiaalit .....	21
5.1	Tavoitteet .....	21
5.2	Materiaalit mallissa.....	22
5.3	Shader .....	22
5.4	Materiaalin kehittäminen .....	24
6	Renderointi.....	31
6.1	Laatiminen .....	31
6.2	Tulos ja jälkikäsittely .....	33
7	Loppusanat .....	35
	Lähteet.....	37

## Symboliluettelo

Highpoly	3D-malli, jonka kappaleen rakenteessa on paljon yksityiskohtia ja dataa, ja siksi sen piirtäminen on usein laitteelle raskaampaa.
Iterointi	Menetelmä, jossa samaa työvaihetta toistamalla voidaan saavuttaa sopivampi ratkaisu.
Lowpoly	3D-malli, jonka kappaleen rakenteessa on mahdollisimman vähän pintoja ja dataa, ja sen piirtäminen on usein laitteelle kevyempää.
Materiaali	Määrittää 3D-mallin pinnan visuaaliset ominaisuudet.
Modifier	Mallinnustyökalu, jolla voidaan muuttaa 3D-objektin muotoa.
Node	3D-ohjelman solmu, joita käyttäen ja yhdistäen voidaan muodostaa esimerkiksi materiaaleja.
Renderointi	3D-ohjelman prosessi, joka muuntaa scenen informaation kuvaksi.
Scene	3D-ympäristö, joka koostuu erilaisista objekteista, kuten 3D-objekteista, valaistuksesta ja kameroista.
Shader	Pienoisohjelma, joka määrittää 3D-objektin pinnan ominaisuudet renderointia varten.
Toon shading	3D-grafiikan renderointitekniikka, jolla saadaan aikaiseksi sarjakuvamaisesti tyylitelty sävytys.

## 1 Työn lähtökohdat

Opinnäytetyö perustui korttikuvan kehitysprosessiin mobiilipeliä Badland Brawl varten. Työtehtävän toimeksiantajana toimi Frogmind Oy, jossa työskentelin peligraafikkona opinnäytetyön aikana.

Opinnäytetyö on jaoteltu eri työprosessin vaiheisiin. Aloitin prosessin työtehtävän selvittämisellä sekä kehitysvaiheiden laatimisella. Työn suunnittelulla oli suuri merkitys kehityksen ajankäytön ja sujuvuuden kannalta, joten sen tarkka laadinta oli tärkeää. Opinnäytetyössä käyn ohjeistavasti läpi työn päävaiheet, aloittaen ohjelmistoon ja työympäristöön perehtymisellä.

### 1.1 Frogmind Oy

Frogmind on helsinkiläinen mobiilipelistudio. Yrityksen perusti kaksi pelikehittäjää vuonna 2012, ja on siitä lähtien julkaissut kaksi peliä: BADLAND (2013) ja BADLAND 2 (2015). Frogmind kehitti myös kolmatta mobiilipeliä Badland Brawl, mitä varten tehdystä työtehtävästä tämä opinnäytetyö perustuu.

BADLAND-sarjan pelit ovat aikaisemmin olleet premium-markkinointimallin pelejä, mutta nyt Frogmind on siirtynyt kehittämään ilmaisia pelejä, joissa pelaajilla on mahdollisuus tehdä mikromaksuja.

Syksyllä 2016 peliyhtiö Supercell Oy osti enemmistön Frogmindista liikekumppanuuden nimissä.

### 1.2 Työn kohdepele Badland Brawl

Badland Brawl on BADLAND-maailmaan sijoittuva toimintastrategiapeli, missä pelaajat taistelevat reaaliajassa toisiaan vastaan verkkoyhteyden välityksellä. Pelaaminen keskittyy lyhyisiin otteluihin kahden pelaajan kesken, joissa tarkoitus on päihittää vastustaja tuhoamalla sen torni sekä puolustaa omaa tornia tuhoutumiselta. Taisteluympäristö on fyysikkapainoinen ja hullunkurinen, mikä tekee pelistä energisen sekä omalaatuisen.

Ennen ottelua pelaaja valitsee oman joukkueensa, johon mahtuu kahdeksan erilaista hahmoa. Hahmoja kerätään pelin edetessä, ja hahmovalikoima on runsas. Korttikuvan työtehtävää tehtäessä hahmojen kokonaismäärä oli 30. Sekä pelin valikossa että ottelussa hahmoja kuvastetaan korteilla. Otteluissa kortteja käytetään kutsumaan korteissa kuvattuja hahmoja, jotka sitten ilmestyvät taistelukentälle 3D-hahmoina. (Kuva 1).



Kuva 1: Kuvakaappaus Badland Brawl -pelin korttivalikkoruudusta

Peli käyttää Frogmindin omaa pelimoottoria, jolloin pelin kehitys voidaan räätälöidä mahdollisimman sopivaksi itse peliä varten. Tämä mahdollistaa esimerkiksi luovan sekoituksen 2D- ja 3D- grafiikkaa.

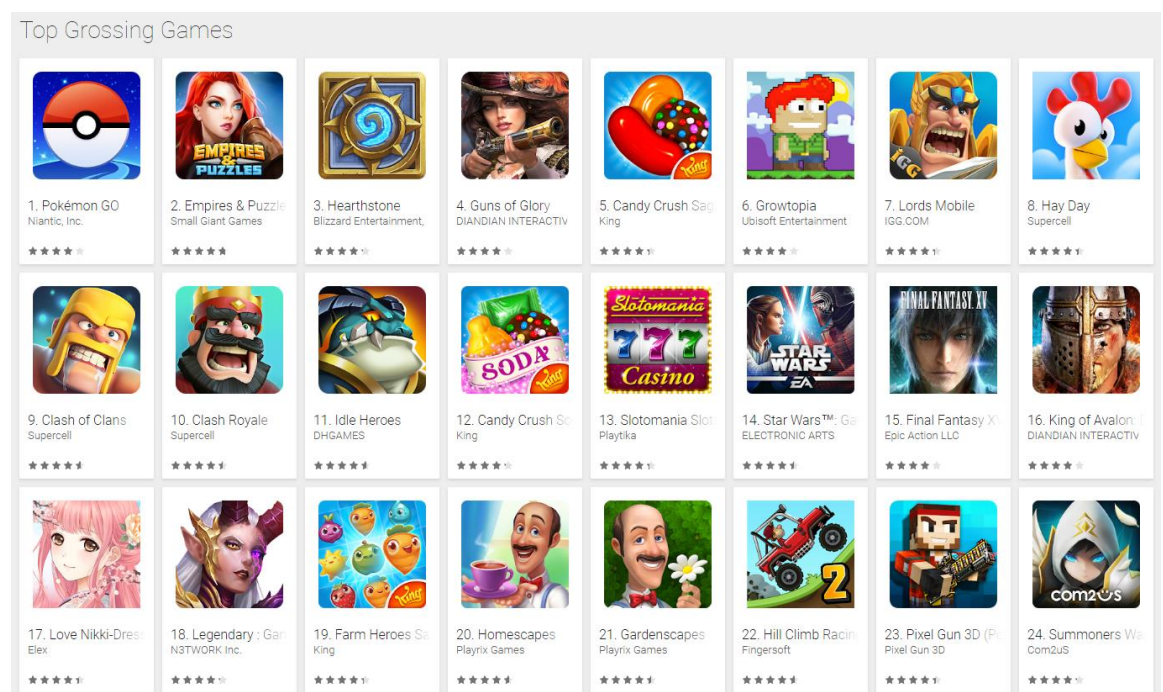
### 1.3 Mobiilipeliteollisuus

Massiiviseksi kasvaneet mobiilipelimarkkinat ovat haaste jokaiselle uudelle pelijulkaisulle. Pelien on saavutettava suosiota ja sen kautta vahva pelaajakanta, jotta peli menestyisi ja pelin kehittäjät voisivat jatkaa sen ylläpitoa. Menestyminen on lähtöisin pelin markkinoinnista sekä tuotteen laadusta. Pelaajakunta on kuitenkin se, mikä määrittää peliprojektien onnistumisen ja epäonnistumisen rajat.

Ilmaiseksi pelattavien (free-to-play markkinointimallin) mobiilipelien suosio perustuu ratkaisuihin poiketa massamarkkinoista. Tärkeimmät haasteet ovat poikkeaminen ulkoasullisesti ja pelattavuudellisesti. Erilainen mutta onnistunut sekä addiktoiva pelimekaniikka

pitää valtaosan pelaajista kiinni pelissä, mutta ulkoasu on se, mikä usein ensimmäiseksi herättää mahdollisten pelaajien mielenkiinnon, ja sen avulla pyritään saamaan pelaaja lataamaan tuotteen. Ulkoasun vaikuttavimpia tekijöitä ovat teema, graafinen tyyli sekä graafinen toteutus.

Markkinoiden suurimpia mobiilipelien kauppialustoja, iOS-laitteiden App Storea ja Android-laitteiden Play Kauppaa, tutkimalla voidaan todeta, että menestyksekkäimmät pelit ajasta riippumatta ovat graafisesti viimeistelyjä, laadukkaita sekä omalaatuisia. Graafiset tyylit ovat usein persoonallisia ja helposti lähestyttäviä, sekä cartoon-tyyppisesti tyyliteltyjä. Karkkimaiset värit sekä karikatyyrisesti kuvatut hahmot osoittautuvat suosituiksi tyylikelementeiksi valtaosassa markkinoiden menestyksekkäimpiä mobiilipelejä (kuva 2).



Kuva 2: Play Kaupan parhaiten myyneet (top grossing) mobiilipelit, 4.2018.

#### 1.4 Työtehtävä ja tavoitteet

Opinnäytetyön aiheena on määrätty työtehtävä, missä oli tavoitteena tuottaa resurssillisesti järkevä prosessi korttikuvien kehitykseen. Työtehtävään sisältyi myös korttikuvien graafisen tyylin selvitys ja toteutus.



Aiemmin korttikuvan tuottamisprosessi keskittyi digitaaliseen maalaamiseen, käyttäen pohjana pelissä käytettävää hahmon 3D-mallia edustavaan asentoon poseerattuna. Tämä prosessi oli ajallisesti raskas sekä tulos ei täysin sopinut pelin visuaaliseen tyyliin. (Kuva 3).



Kuva 3: Aikaisempi versio kortista, missä kulmikkaasta hahmomallista tehtiin korttikuva kuvankäsittelyohjelmalla päälle maalaamalla.

Työtehtävän lopputuloksena oli tarkoitus saada aikaiseksi pelin tyyliin paremmin sopiva korttikuva käyttäen vähemmän resursseja, kuten työvoimaa sekä työaika.

### 1.5 Työn suunnittelu

Projektit rajoitetulla työajalla voivat venähtää tai jäädä tulokseltaan keskeneräisiksi tai hutaistuuksi. Siksi projektien läpikotainen suunnittelu ja työvaiheiden selvitys ovat tärkeitä osia koko projektin sujuvuuden kannalta. Tehtävien tarjoamiin ongelmiin tulisi löytää vastauksia tai johtolankoja heti työn alkuvaiheilla.

Työtehtävän pääongelmat keskittyivät siihen, kuinka saada aikaiseksi helppokäyttöinen sekä nopea tapa tehdä peliin korttikuvia. Pelissä oli jo 30 hahmoa ja uusia hahmoja lisätään peliin kehityksen edetessä. Kaikille hahmoille tulisi teettää uudet korttikuvat käyttämällä tätä kehitysprosessia ja samalla standardisoida kuvien laatu sekä tyyli riippumatta kuvien kehittäjästä.

Opinnäytetyö keskittyy vain yhden hahmon korttikuvan kehitykseen. Hahmo valittiin työn aiheeksi työpaikalla siksi, koska sen design oli yksinkertainen mutta hyödynsi erilaisia materiaaleja.

Uuden prosessin tulisi löytää toimivampia ratkaisuja aikaisemman kehitysprosessin tuottamiin ongelmiin, jotka tulivat resurssien raskaudesta sekä tyylin yhteensopimattomuudesta varsinaiseen peliin verrattuna. Tutkin mahdollisuutta jättää prosessista pois 3D-mallin päälle tehty maalaus ja korvata se paremmalla renderoinnilla. Näin kuvan tuottamisesta voisi tulla paljon nopeampaa. Tällainen menetelmä koostuisi mallin valmistelusta, mallin tyyliä korostavan materiaalin luomisesta, korttikuvan asetelman rakentamisesta ja viimeiseksi renderoinnista. Tämä menetelmä mahdollistaisi pelissä käytetyn valmiin 3D-mallin käyttämistä korttikuvaa varten työn alusta loppuun, hyödyntäen 3D-grafiikan mallinsohjelmia. Pelissä näkyvät hahmot vastaisivat hyvin tarkasti korteissa kuvattuja hahmoja ja niiden muotoja, kun korttikuvat perustuisivat suoraan pelissä käytettäviin malleihin. (Kuva 4).

Työprosessin varsinainen suunnittelu alkoi visuaalisen tyylin laatimisella. Pelin pelattavassa osiossa grafiikka oli valtaosin 3D-painotteinen, kun taas valikko-osuus perustui 2D-grafiikkaan. Hahmojen ja kenttien 3D-mallit perustuivat lowpoly-geometriaan, jotta mobiililaitteiden resurssit riittäisivät grafiikan esittämiseen.



Kuva 4: Kuvakaappaus pelin Badland Brawl 3D-näkymästä.

## 1.6 Graafinen tyyli

Pelin grafiikan yleinen teema sekä hahmomallit oli tyyllitelty sopimaan lowpoly-rajoituksiin, keskittyen helposti ymmärrettäviin siluetteihin, karkkimaisiin väreihin sekä piirroshahmo- maiseen, karikatyyriseen luettavuuteen. (Kuva 5). Korttikuvien tyylin tuli vastata hahmojen ja pelimaailman teemaa mahdollisimman tarkasti. Visuaalisesti tyylin tuli olla myös oma- laatuinen ja muista vastaavista peleistä positiivisesti erottuva.



Kuva 5: Pelissä Clash Royale hahmojen korttikuvat kertovat paljon hahmon tarkoituksesta sekä persoonasta.

Tyylin hahmottamisen aloitin keskustelemalla pelin ulkonäöstä vastaavien kehittäjien kanssa. Vahvaksi referenssityylyksi tuli *toon shading*, mikä tarkoittaa 3D-grafiikan renderointitekniikkaa, jolla saadaan aikaiseksi sarjakuvamainen ja käsin piirretyltä animaatiolta vaikuttava grafiikkamaailma. (Hachigian, 2005). Referenssityyliä edustavaksi peliesimerkiksi ehdotettiin peliä The Legend of Zelda: Breath of the Wild (kuva 6), jossa pelin hahmot



korostuvat voimakkailla väreillä ja terävillä valaistuksen taitoskohdilla hahmojen pinnoissa. Muunlaisia esikuvia toon-tyyliä varten ovat esimerkiksi The Legend of Zelda: Wind Waker HD ja Punch Out!!, joissa hahmojen grafiikka korostaa pelin sympaattista vaikutelmaa. (Kuva 7).



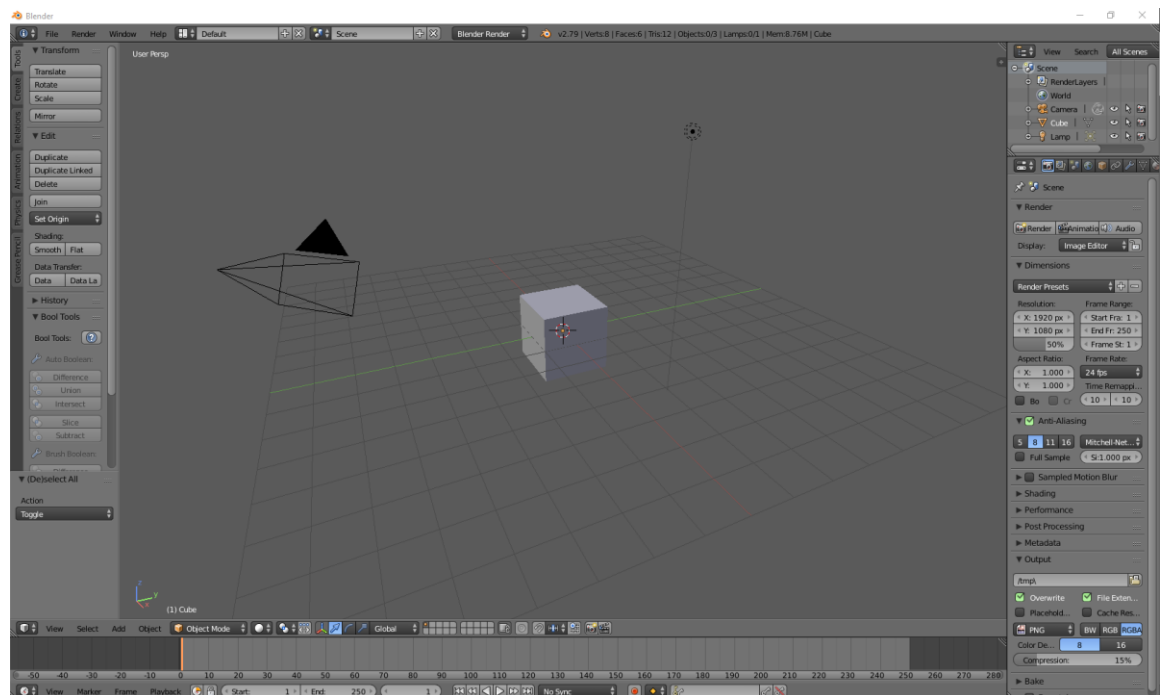
Kuva 6: The Legend of Zelda: Breath of the Wild -pelissä hahmot renderoidaan toon shading-tekniikalla.



Kuva 7: Referenssipelejä, joissa toon-renderointi perustuu kolmeen sävyyn: valaistu, valaisematon sekä reunan niukka kiilto.

## 2 Ohjelmisto

Työprosessissa käytettäväksi ohjelmaksi valittiin Blender, sillä se on tehokas ja ilmainen, avoimen lähdekoodin 3D-editointiohjelma, ja työpaikalla suuressa käytössä. Sitä levitetään GNU GPL -lisenssin alla, joka antaa kenelle tahansa oikeuden käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmia ja niiden lähdekoodia. Blenderillä voi mallintaa, teksturoida, animoida ja renderoida, sillä on myös luotu lähes kaikki peliprojektin 3D-sisältö, mukaan lukien hahmot. (Kuva 8).



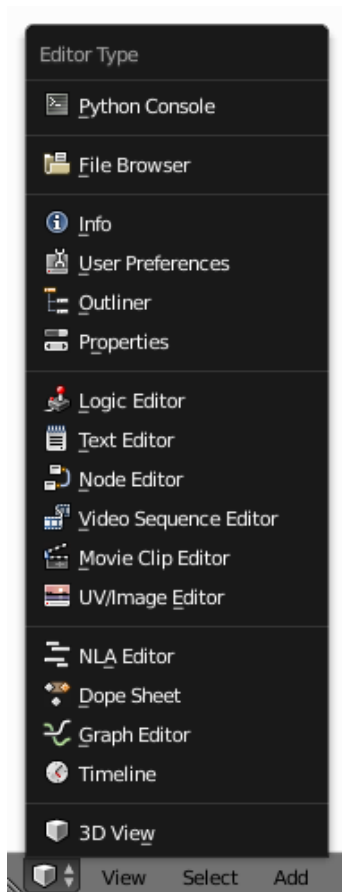
Kuva 8: Blender-ohjelman oletuskäyttöliittymä

Lopullinen renderoitu kuva oli tarkoitus viedä kuvankäsittelyohjelmaan Adobe Photoshop, jotta hahmokuvalle pystyttäisiin liittämään korttitausta.

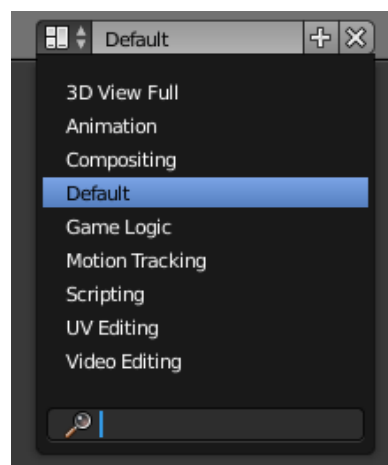
### 2.1 Blender työympäristö

Blender erottuu muista teollisuuden standardiohjelmista sovelluksen käytettävyydellä; ohjelman käyttöliittymän voi luontevasti muokata omanlaiseksi. Tämä onnistuu siksi, koska ohjelman ruutu on jaettu alueisiin (area). Jokaisella alueella on oma editoritila ja koko, joita voi helposti muuttaa. Blenderin käyttäjä voi valita ruutuunsa ne editoritilat, joita hän

työskentelyssään tarvitsee. Erilaisia editoritiloja ovat esimerkiksi 3D-näkymä, tiedostose-lain ja kuvaeditori. (Kuva 9). Kaikissa editoritiloissa on oma header-palkki, mistä löytyy asetukset editoritilaa varten. Ruudun alueet ja editoritilat voidaan myös tallentaa ruutu-asetelmiksi, joita vaihtamalla voi vaivatta muuttaa koko ruudun kokoonpanoa, kun sille on tarvetta, esimerkiksi vaihtaessa mallintamisesta animoimiseen. (Kuva 10).

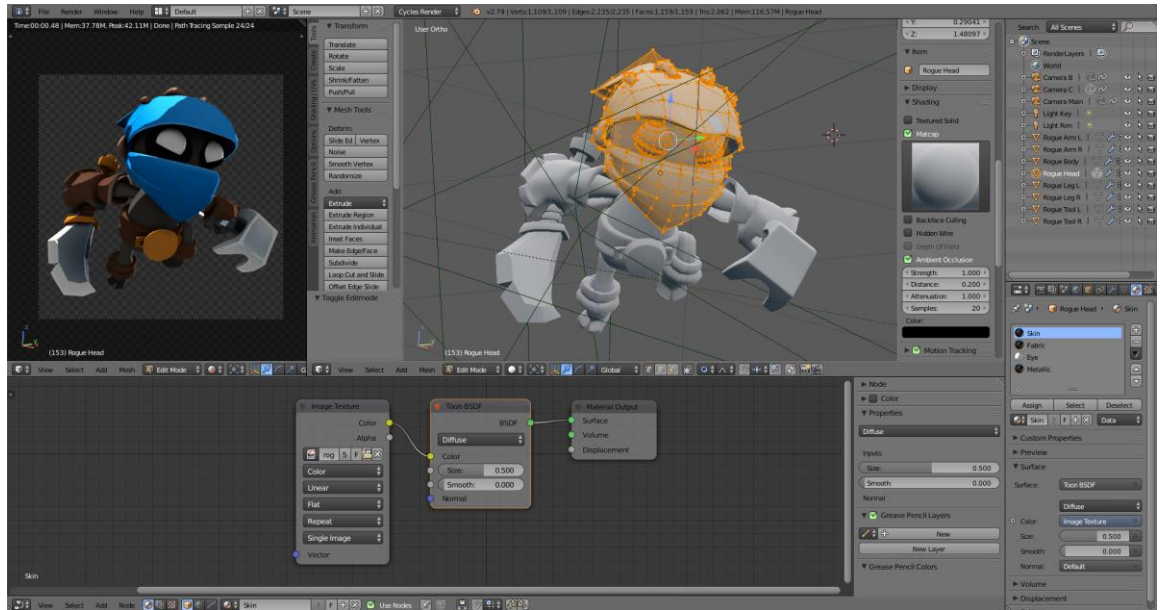


Kuva 9: Editoritilan valitsin.



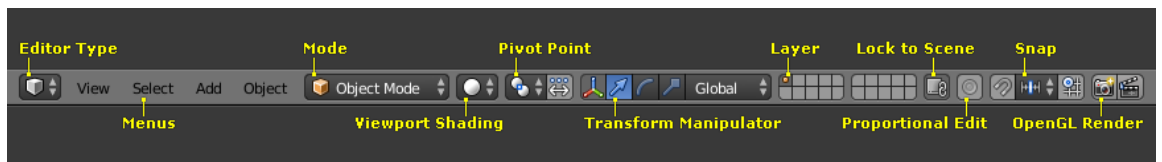
Kuva 10: Ruutuasetelman valitsin.

Työprosessissa työskentelyä nopeutti ruudun mukautus työvaiheen mukaiseksi. Pohjana käytin oletusasetelmaa, vaihtaen aikajana-editorin (timeline) node-editoriksi. Lisäksi tarvitsin kaksi 3D-editoritilaa, joista toinen näyttäisi kameraobjektin näkymän aina renderoi-tuna ja toinen soveltuisi vapaaseen mallin muokkaukseen ilman kameraa. (Kuva 11).



Kuva 11: Työprosessissa käytetty ruutuasetelma.

3D-editoritila on monipuolinen ja olennainen mallinnukseen ja objektien tarkasteluun soveltuva editoritila. (Kuva 12). Siinä on aina käytössä yksi työtila useasta vaihtoehdosta, joita ovat esimerkiksi objektitila, muokkaustila, ja tekstuurinmaalauk. Työprosessi keskittyy objekti- ja muokkaustiloissa työskentelyyn.



Kuva 12: 3D-näkymän header.

Scenen geometria koostuu yhdestä tai useammasta objektista. Tällaisia objekteja voivat olla scenessä olevat valaisimet, 2D- ja 3D-kappaleet, figuurirungot (armature) animoimaan niitä malleja, sekä kamerat, joilla otetaan kuvia tai videota kokonaisuudesta. (Blender, 2015.)

Työprojektin perusta koostuisi aluksi kamerasta sekä valaisimesta. Sceneen lisättäisiin hahmomalli ja sen teksturi. Mallille annettaisiin sopiva asento eli poseeraus. Seuraavaksi mallille luotaisiin materiaali käyttämällä mallin omaa tekstuuria ja valmistelemalla tarvittavat asetukset shaderille. Lopuksi kamera sijoiteltaisiin kuvaamista varten ja valaistuksen asetus olisi sovellettava sceneä varten. Viimeiseksi kokonaisuus olisi renderoitava valmiiksi korttikuvaksi.

## 2.2 Renderointimoottori

Renderoinnilla tarkoitetaan 3D-sovelluksessa tapahtuvaa prosessia, joka kääntää scenen 3D informaation 2D kuvaksi monimutkaisten matemaattisten laskelmien avulla. Proses- sissa koko scenen alue-, tekstuuri- ja valaistustiedot yhdistetään määrittämään väriarvo jokaiselle pikselille lopullisessa 2D-kuvassa. Renderointityyppjä on pääsääntöisesti kaksi: reaaliaikarenderointi (real-time rendering) sekä esirenderointi (pre-rendering). Reaaliaikarenderoinnissa kuvat laskelmoidaan 3D informaatiosta vauhdikkaasti, esimerkkinä videopelit, joissa kuvattava scene muuttuu pelaajan vaikutuksesta. Esirenderoinnissa nopeus korvataan prosessin ja lopputuloksen korkealla laadulla, esimerkkinä ani- maatioelokuvat, joissa video koostuu valmiiksi renderoiduista kuvista. (Slick, 2017.)

Renderointimoottorilla tarkoitetaan 3D-sovelluksen sisäistä ohjelmaa, joka suorittaa ren- derointiprosessin. Blenderissä renderointimoottoreita on kaksi erilaista: Blender Render ja Cycles Render. Blender Render on Blenderin alkuperäinen renderointimoottori, jonka lähdekoodi on lähtöisin varhaiselta 90-luvulta. Tämä moottori on tehokas ja erittäin nopea prosessiltaan, mutta se ei kykene suorittamaan fotorealista renderointia, toisin kuin Cy- cles. Cycles on vuonna 2011 julkaistu monipuolinen ja suosittu renderointimoottori, joka osaa suosittaa fyysisesti tarkkoja laskutoimituksia ja siten saavuttaa fotorealistisia tulok- sia. (Kuva 13). (Trammell, 2016.)



Kuva 13: Cycles on fyysikaalisuuteen perustuva renderointimoottori Blenderissä.



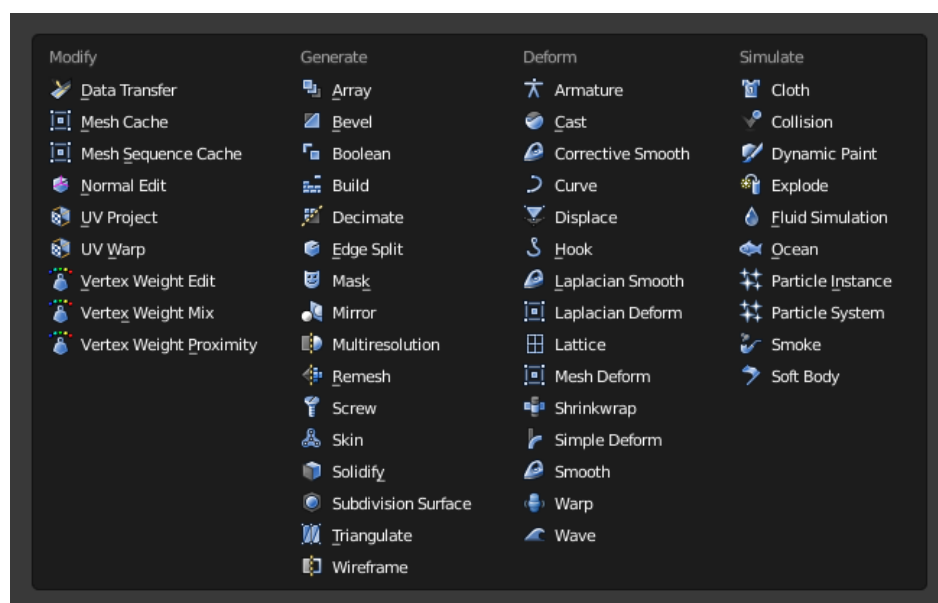
### 3 Mallin valmistelu

Korttikuvan renderoinnin keskeisin elementti oli hahmon 3D-malli, josta kuva muodostettiin. Peliprojektin hahmomallit käyttivät tiedostomuotoa FBX (filmbox), joka on alustasta riippumaton ja tarkka 3D-mallin tallennusmuoto, sekä tekstuuritiedostot muotoa PNG (portable network graphics), joka on häviötön bittikarttagrafiikan tallennusformaatti.

Hahmolle kuuluvat malli- ja tekstuuritiedostot tuotiin Blenderiin käsiteltäviksi. Hahmon mallin suurimmat kappaleet, kuten keho, raajat ja pää, irrotettiin toisistaan ja jaettiin omiin objekteihinsa. Jotta mallista saatiin yksityiskohtaisempi ja pinnoista pyöreämpiä lowpoly-rakenteen sijaan, mallin geometriaa tuli lisätä ja pintoja pehmentää, jolloin mallista tuli niin sanottu highpoly-malli. Tapoja saavuttaa haluttu rakenne oli useampia, mutta työprosessin proseduraalisuuden kannalta tähän käytettiin Blenderin modifier-työkaluja.

#### 3.1 Modifier-muotoilutyökalut

Mallintamisen helpottamiseksi ja nopeuttamiseksi Blenderissä on valikoima erilaisia modifier-työkaluja (suom. muotoilu), joita voidaan määrittää objekteille muokkaamaan tämän geometriaa. Nämä muotoilutyökalut ovat erityisen käteviä siitä, etteivät muotoilut muokkaa mallia ennen tämän hyväksymistä, ja tuloksia voi tarkastella ja testaila muuttamalla työkalujen parametreja. (Kuva 14).

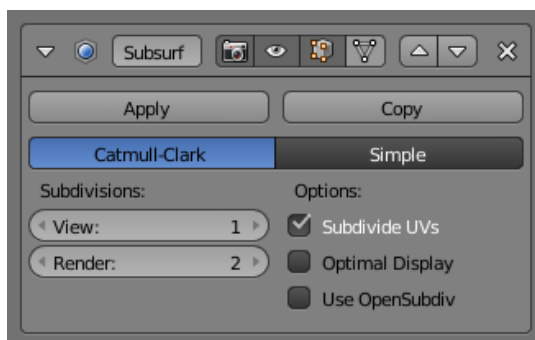


Kuva 14: Blenderin modifier-työkaluvalikko.

Työprosessissa päätin hyödyntää pääasiassa kahta muotoilutyökalua; Subdivision Surface (Subsurf) ja Bevel. Käyttämällä näitä työkaluja lowpoly-hahmomallista pyrittiin saamaan yksinkertaisilla vaiheilla orgaanisen ja laadukkaan näköinen lopputulos. Prosessia tulisi pystyä toistamaan kaikissa hahmomalleissa luontevasti siten, että visuaalinen iterointi ja tulevaisuuden muutokset hahmojen 3D-malleihin eivät johtaisi suureen työmäärään.

### 3.1.1 Subdivision Surface

Subsurf-muotoilutyökalua käytetään kaksinkertaistamaan pintojen (face) määrä, mikä samalla pyöristää kappaleen muotoa. Tämän toiminnon voi moninkertaistaa määrittämällä ”subdivisions”-kertoimen erikseen ohjelmanäkymälle (view) sekä renderointiin (render). (Kuva 15).

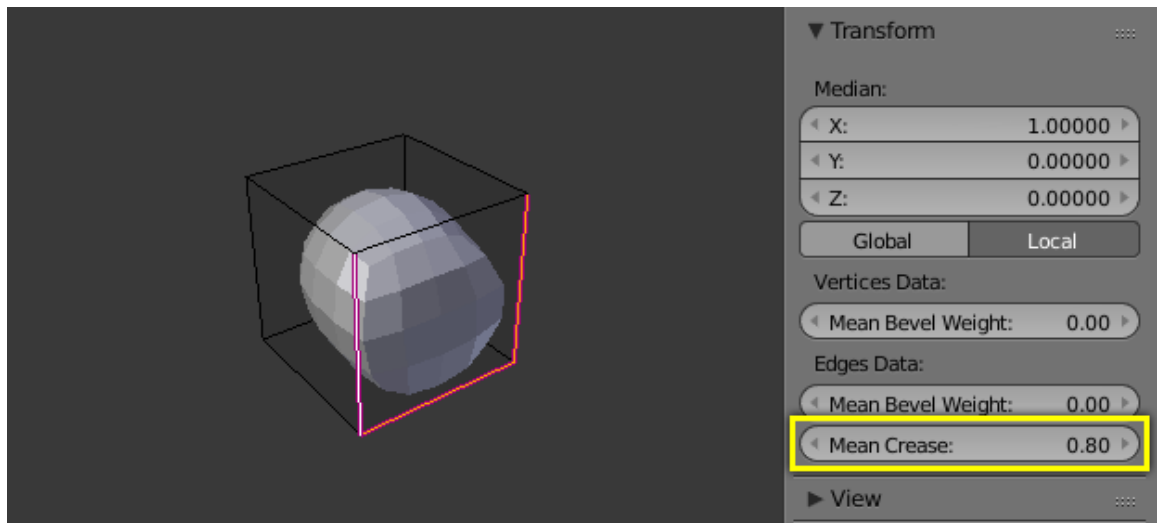


Kuva 15: Subsurf-muotoilutyökalun paneeli.



Kuva 16: Subsurf-muotoilutyökalun tasot nolasta kolmeen, pinnan pehmennyksellä ja ilman.

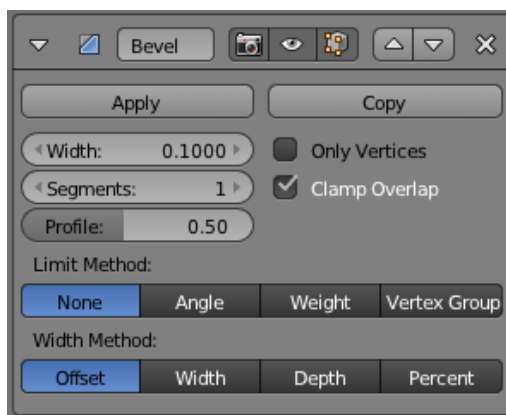
Subsurf-muotoilun voimakkuuteen geometrian eri kohdissa voi vaikuttaa asettamalla kappaleen reunaviivoille "mean crease"-arvoja. Nämä arvot määrittävät reunaviivoille vastustusta muotoilutyökalun pyöristämistä vastaan. (Kuva 17).



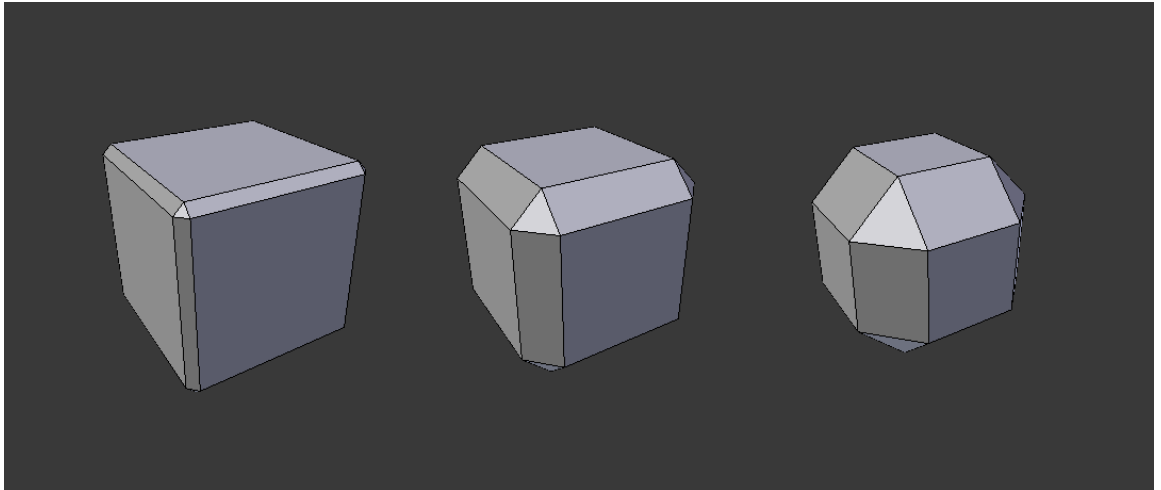
Kuva 17: Subsurf-muotoilulla pyöristetty kuutio minkä valituille reunaviivoille on määritetty Mean Crease -arvoja reunojen terävöitykseen.

### 3.1.2 Bevel

Bevel-muotoilutyökalulla voidaan määrittää kappaleen reunaviivoista viistoja. Työkalu tarjoaa laajan valikoiman asetuksille reunojen viistoamista varten, yleisimmin käytettyinä esimerkiksi kulman leveys (width), viistoamissegmenttien määrä (segments) sekä kulman rajoitusehdot (limit method). (Kuva 18). (Kuva 19).



Kuva 18: Bevel-muotoilutyökalun paneeli



Kuva 19: Reunaviivojen viistoaminen kuutiolla, viistokulman eri leveyksillä.

### 3.1.3 Modifier-työkalujen käyttö hahmomallissa

Subsurf-työkalua käyttäen koko lowpoly hahmomallista saatiin pyöreämpi ja luonnollisemman oloinen (kuva 20). Pyöreä pinta on tärkeää etenkin valaistuksessa, jotta valon taittuminen seuraisi kappaleen muotoa pehmeämmin.



Kuva 20: Vas. hahmomalli ilman muotoilutyökaluja, kesk. yhden tason subsurf-työkalulla, oik. kahden tason subsurf-työkalulla sekä bevel-työkalulla.

Bevel-työkalun käyttö hahmomallissa keskittyi subsurf-muotoilun soveltamiseen. Hahmomallissa oli jyrkkiä reunoja, kuten hahmon tikarin terä, jotka haluttiin pitää viistoina ja korkealaatuisena. Jotkin terävät kulmat mallin geometriassa voisivat vaikuttaa valaistuksen taittumiseen epäluonnollisesti, mutta sitä voi korjata bevel-muotoilun viistoamissegmenttien määrällä, pyöristäen kappaleen reunojen geometriaa.

### 3.2 Hahmon poseeraus

Kun jokaiselle objektille oli määritelty tarvittavat muotoilutyökalut sopivilla asetuksilla, oli mallinnusvaihe valmis, ja sitä seuraisi poseerausvaihe. Poseerauksessa hahmomallin objektit aseteltaisiin siten, että hahmomallissa olisi elävyyttä sekä lopputulos kuvastaisi hahmoa mahdollisimman suorasti. Työtehtävän hahmon rooli pelissä on toimia ketteränä taistelijana, minkä pitäisi kuvastua myös poseerauksessa.

Lopputuloksessa on tärkeää, että jokainen hahmo olisi helposti ymmärrettävä ja muista erottuva pelkällä vilkaisulla. Tämä pohjautuu hahmon suunnitteluvaiheeseen sen ulkonäöstä ja värien käytöstä. Voi olla tapauksia, missä eri hahmojen siluetit tai väripaletit saattaisivat muistuttaa toisiaan, joten vastaava ongelma voidaan ratkaista korttikuvassa hahmon poseerauksella.

Asetin scenen oletuskameraobjektin osoittamaan hahmomallia kohti ja säädin kameran asetukset lopputuloksen mukaisiksi. Nyt aktiivisesti renderoivasta 3D-editoritilasta tuli tärkeää, sillä sen näkymästä lopputulos renderoitiin. Asettelin eli poseerasin hahmomallin erilliset objektit, kuten pään, kehon ja raajat, siten, että hahmon korttikuva huokuisi toimintaa. Asennoksi valitsin pelin tapahtuman, missä hahmo loikkaa ilmaan hyökätäkseen, ase valmiina taisteluun. (Kuva 21).



Kuva 21: Poseerattu hahmomalli.

## 4 Scene

Korttikuvien kehitysprosessin tuli olla rakennettu siten, että jokaisen pelihahmon kuva saataisiin tehtyä samaa prosessia käyttäen. Sen sijaan että, jokainen hahmomalli olisi omana tiedostona erillään toisista, päädyin optimoidumpaan ratkaisuun, missä kaikille pelin hahmoille voitaisiin renderoida korttikuva saman Blender-projektin sisällä. Tärkeää oli pystyä palaamaan projektiin, jos jokin hahmo tarvitsi muutosta tai parannusta renderointiin. Samalla myös samoja materiaalipohjia, valaistusmenetelmiä sekä renderointiasetuksia voitaisiin käyttää eri hahmojen renderoinnissa, säästäten paljon ylimääräistä työtä.

Hahmomalli jaettiin suurimpien ruumiinosien mukaan eri objekteihin, jotta esimerkiksi raajoja olisi helppo asetella ja säätää yksi kerrallaan. Koska jokaiselle objektille voi määrittää yksilölliset muotoilutyökalunsa, voitiin joistakin kappaleista saada kulmikkaampia ja toisista pyöreämpiä, esimerkkinä hahmon teräväkärkinen miekka verrattuna hahmon pyöreäpintaisiin jalkoihin.

Koska hahmoja on lukuisia, on Blender-projekti pidettävä järjestyksessä. Samalle hahmolle kuuluvat hahmomallin objektit, valonlähteet sekä kameraobjekti asetettiin kaikki samaan ryhmään (group), hahmon mukaan nimettynä. Blenderissä ryhmiä voi käsitellä yksi kerrallaan, mikä tulisi helpottamaan työskentelyä, kun projekti täyttyy erilaisista objekteista. (Kuva 22).



Kuva 22: Hahmon eri objektien ryhmittely Blenderissä.

## 4.1 Valaistus

Jotta 3D-ympäristö heräisi eloon, on scenessä oltava digitaalisia valonlähteitä valaisemassa kappaleita, juuri kuten valaistus elokuvatuotannossa valaisisi näyttelijät. Valaistus on renderoinnin jälkeen toiseksi monimutkaisin ja teknillisin vaihe koko 3D-tuotannossa. (Slick, 2017.)

Varsinaisen valaistuksen tulee olla mahdollisimman realistinen ollakseen uskottava, mutta silti tarpeeksi dramaattinen voidakseen ilmaista tuottajan suunnittelemaa tunnelmaa. Valaistuksella voidaan myös saavuttaa scenen kuvan paletti, tunnelma sekä henki. (Slick, 2017.)

Blenderissä valaistus voidaan saavuttaa lampuilla sekä valaisevalla materiaalilla. Lampuja on viisi erityyppistä: piste (point), suunnattu, eli aurinko (sun), kohde (spot), puolipallo (hemi) ja alue (area). Kullakin lampulla on erilaisia ominaisuuksia ja vahvuuksia. Blenderin scenen ympäristölle voi myös määrittää valaisevia arvoja. Korttikuvan kehitysprosessissa käytin aurinkotyyppistä valonlähdettä yhdessä ympäristövalaisun kanssa.

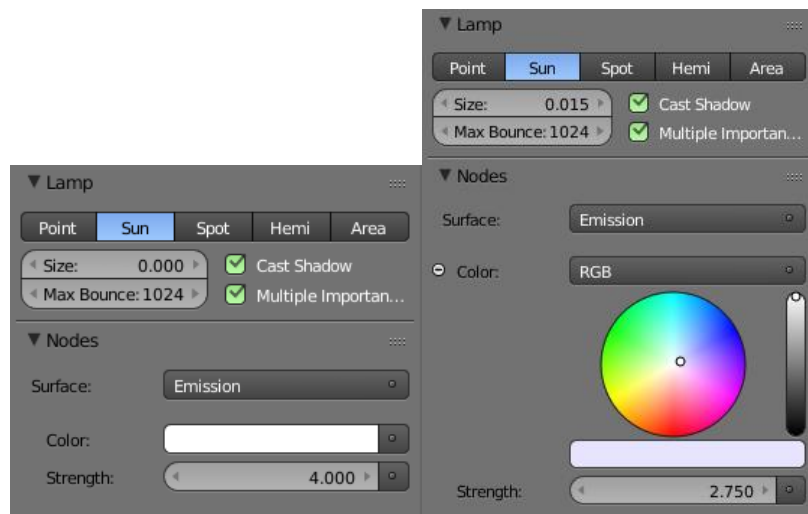
## 4.2 Valaistusmenetelmä

Valaistuksen asettelulla on suuri rooli lopputuloksen kannalta. Yleisin asettelu studiokuvauksessa perustuu kolmen pisteen valaistukseen. Nämä kolme valopistettä ovat päävalo (key light), tasoitusvalo (fill) ja takavalon (backlight).

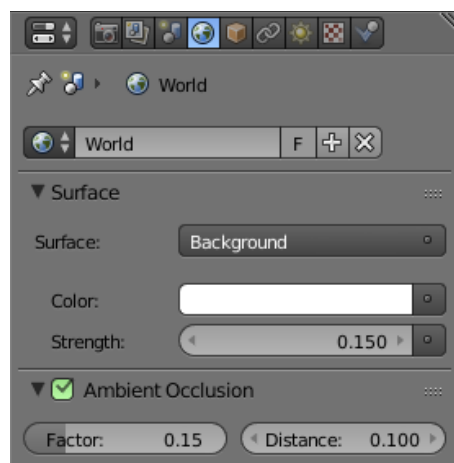
Päävalo on joko voimakas, auringonkaltainen valo tai kohdevalo loistaen kappaleen edessä kameran suhteen. Heikompi tasoitusvalo asetetaan päävalon vastakkaiselle puolelle kappaletta täyttämään varjoja. Takavalon sijainti on kappaleen takana ja osittain sivulla osoittamassa kappaletta sekä kameraa korostaakseen kappaleen reunaa, jotta tämä erottuisi taustasta. (South, 2015.)

Korttikuvan kehitysprosessin tuli toimia myös valaistuksen puolesta kaikissa muissa korttikuvissa käyttäen samaa tekniikkaa, jotta lopulliset kuvat olisivat valaistuksen puolesta samankaltaisia. Päädyin tyypilliseen studiovalaistusasetelmaan, missä scenen päävalo ja tasoitusvalo toimivat aurinkotyyppisinä valopisteinä. Lampun koko (size) määrittää, kuinka pehmeä valon tuottaman varjon reuna on. Päävalon tuli käyttää mahdollisimman terävää varjon reunaa eli kokoa 0, korostaakseen toivottua toon-tyyliä. Tasoitusvalo sai

olla hennosti pehmeämpi, arvolla 0.015. Päävalon värin tuli olla 100 % valkoinen, voimakkuudella 4.0, kun taas luonnollisemman tasoituksen sai aikaiseksi antamalla tasoitusvalolle lievästi sinistä sävyä, mutta pienemmällä voimakkuudella, 2.75. (Kuva 23). Takavalona käytin ympäristövaloa, World > Surface > Background, asetuksilla 100 % valkoinen väri, 15 % voimakkuudella (0.15). Asetin Cycles-renderointimoottoriin kuuluvan ympäristöshaderin (ambient occlusion) käyttöön alhaisilla asetuksilla; vaikutus 15 %, vuorovaikuttavien kappaleiden etäisyys 0.1 Blenderin mittayksikköä. (Kuva 24). Tämä valaistustapa ei ole fyysisesti täsmällinen, mutta on silti hyödyksi varsinkin kappaleiden muotojen korostuksessa muiden valojen kanssa.



Kuva 23: Asetukset päävalolle (vas.), ja tasoitusvalolle (oik.)

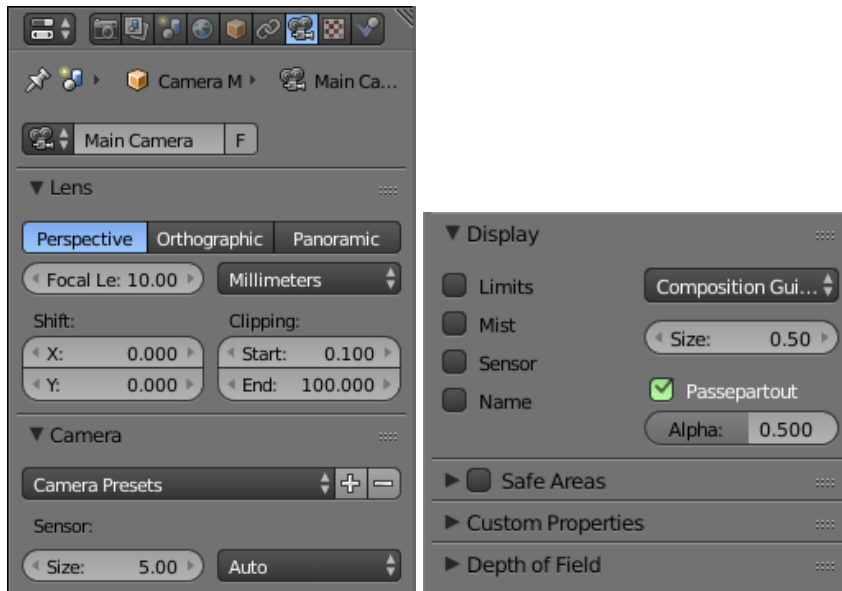


Kuva 24: Asetukset takavalolle (surface) sekä ympäristöshaderille.



### 4.3 Kamera

Blenderin kameraobjekti jäljittelee oikean kamerasäätöpaneelin käyttäytymistä. Kameran linssi-asetuksilla voidaan saavuttaa erilaisia tapoja nähdä 3D-kappaleet scenessä. Blenderin kameralla on kolme linssityyppiä: perspektiivinen, ortografinen ja panoraamainen. (Kuva 25).



Kuva 25: Kameraobjektin asetukset.

Kameran asetelmalla oli merkitystä hahmon korttikuvaan. Yritin löytää sopivan kompromissin asetusten suhteen siten, että hahmon siluettia olisi vaivatonta lukea mutta silti lopputulos pysyisi energisenä. Perspektiivilinssiä käyttäen, eri kamerasyvyysasetuksia kokeilemalla päädyin arvoihin 10 (focal length) ja 5 (sensor size), korostaen hahmon liikettä ja dynaamisuutta mutta pitämällä syvyysvääristymän alhaisena. (Kuva 26). Lähtökohtaisena tavoitteena oli myös prosessin kannalta pitää kameran asetukset standardina kaikkia eri hahmojen korttikuvia varten, mutta silti pystyä vaihtelevaan kameran sijaintia ja kulmaa hahmokohtaisesti.



Kuva 26: Testailuja eri syvyysasetuksilla kameralle (focal length / sensor size).

## 5 Materiaalit

Materiaali määrittää sen, mistä aineesta kappale on tehty. Yksinkertaisimmillaan materiaaleja voidaan käyttää kertomaan kappaleen aineesta, tai maalaamaan kappale eri väreillä. Useimmiten ainetta kuvataan sen pinnan ominaisuuksilla, kuten värillä, kiiltävyydellä ja heijastuskyvyllä. Tyypillisimpiä materiaaleja voivat olla esimerkiksi messinki, iho, lasi tai kangas. (Blender, 2015.)

Blenderissä objekteille voidaan määrittää materiaaleja, jotka näkyvät renderoidussa näkymätilassa, kuten opinnäytetyön työympäristön renderoivassa 3D-editoritilassa ja tietenkin lopputuloksen renderoidussa kuvassa. Objektilla voi olla useampia materiaaleja, joiden näkyvyyttä objektin pinnalla voidaan muuttaa määrittämällä mitä materiaaleja eri pinnat (face) käyttävät.

### 5.1 Tavoitteet

Vaikka hahmomalli itsessään olikin nyt pääosin valmis ja poseerattu, lopputuloksen laatuun vaikuttaisi eniten materiaalien ja valaistuksen käyttö. Korttikuvaan oli tavoitteena saavuttaa aikaisemmin suunniteltu graafinen tyyli. Koska korttikuvaa ei ollut tarkoitus muokata kuvankäsittelyohjelmassa lopullisen renderoinnin jälkeen, oli materiaalien ja niiden shaderien aikaansaama jälki vastattava haluttua tulosta. Hahmojen korttikuvilla oli suuri rooli pelissä, joten kuvista oli tehtävä laadukkaita ja persoonallisia, että ne erottuvat muista peleistä, kuten Clash Royalesta. Korttikuvat tulevat näkymään ruudulla pienessä koossa, joten niiden tulee olla helposti luettavia sekä selkeästi keskenään erilaisia. (Kuva 27).



Kuva 27: Kuvakaappaus pelin korttivalintaruudusta, missä näkyy erilaisia kortteja hahmoista.

## 5.2 Materiaalit mallissa

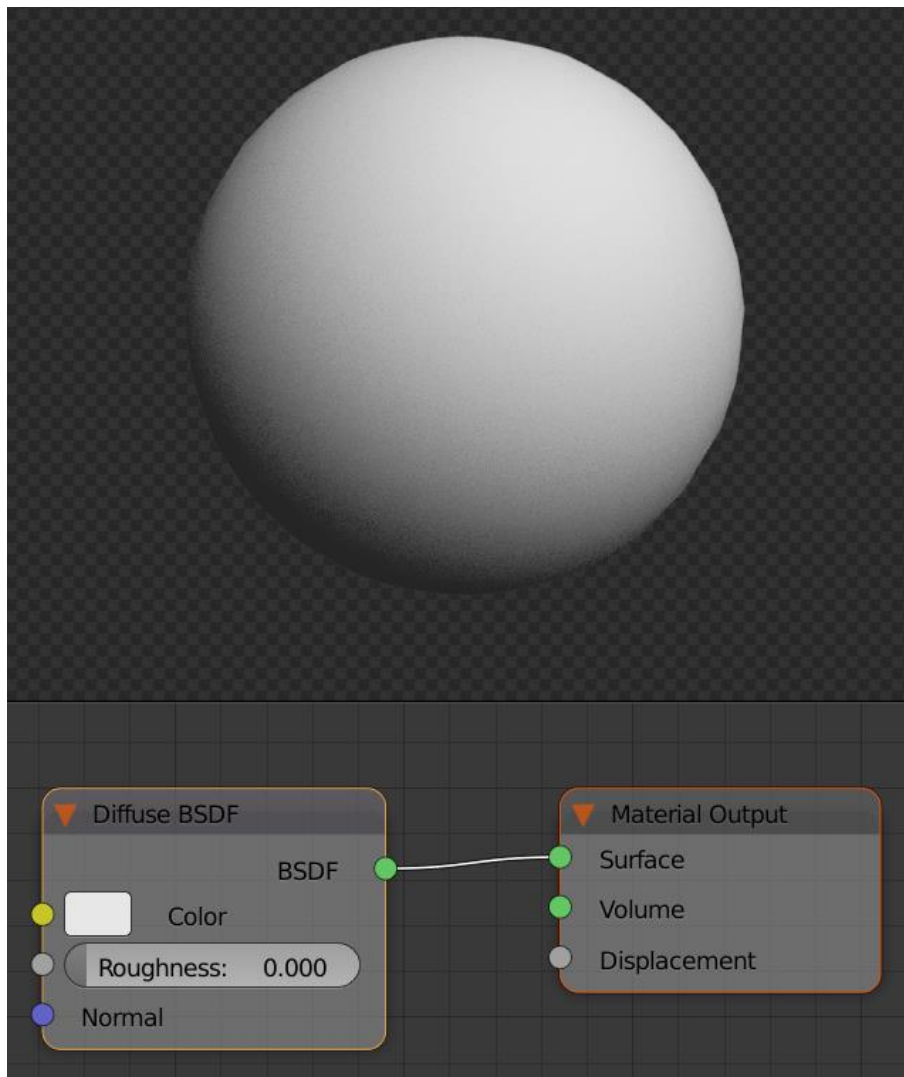
Blender-työympäristössä jokaista hahmoa varten tehtäisiin omat materiaalit, ja ne nimettäisiin selkeästi. Ohjelmisto järjestää materiaalit valikossa nimijärjestykseen, eli jokainen materiaali nimettäisiin aloittaen hahmon kutsumanimellä, jota seuraisi materiaalin pääominaisuus. Työtehtävän hahmolla suunniteltiin olevan erilaisia pintoja, jotka jaettiin aineiden mukaan: turkki, kangas, silmä ja metalli. Hahmolle tuli yhteensä neljä erilaista materiaalia, joilla useampia erilaisia shadereita. Jokaiselle hahmon objektille, kuten keholle, raajoille sekä tikarille, oli määriteltävä (assign), millä pinnoilla kukin materiaali esiintyisi.

## 5.3 Shader

Blenderissä materiaali määrittää kappaleiden ulkonäön. Materiaali koostuu kolmesta shaderista: surface (pinta), volume (volyymi) ja displacement (siirtymä). Surface shader määrittää valon käyttäytymisen kappaleen pinnalla, esimerkiksi valon taittumisen ja pinnan värin. Jos pinnan shader ei heijasta tai absorboi valoa, se siirtyy kappaleen volyymiin, jolloin volume shader määrittää, kuinka valo läpäisee kappaleen aineen. Volume shaderia voidaan käyttää esimerkiksi läpinäkyvissä materiaaleissa, kuten jäässä tai vedessä. Jos volume shaderia ei käytetä, valo kulkeutuu suoraan kappaleesta läpi ilman taitosta.

Displacement shaderilla voidaan vaikuttaa kappaleen pinnan ja volyymin muotoon, korostaen yksityiskohtia kappaleessa esimerkiksi tekstuuria käyttäen.

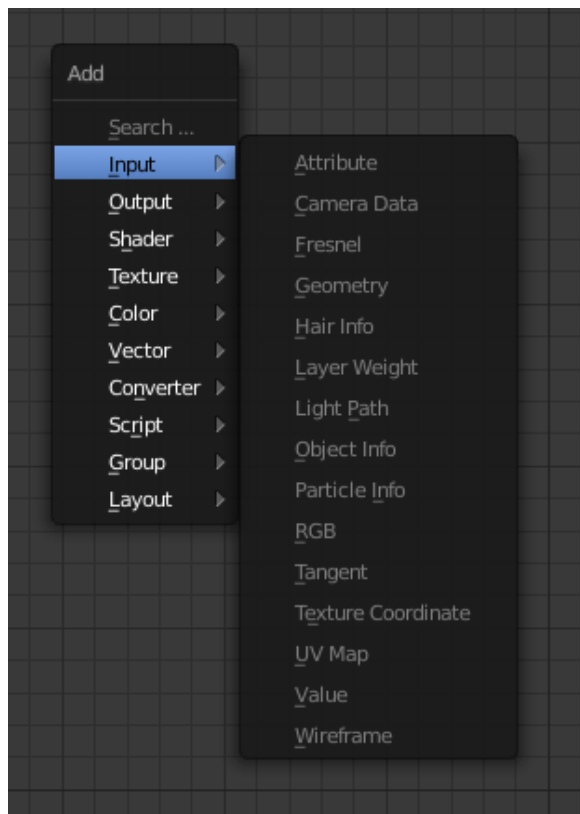
Blenderin Cyclesissä shaderien rakennusprosessi on erilainen kuin muissa markkinoiden renderintimoottoreissa. Huolimatta siitä, että ohjelmisto tarjoaa käyttäjälle valmiiksi luotuja shadereita muutamilla yksinkertaisilla säädöillä, se mahdollistaa myös shaderien luomisen alusta alkaen halutulla tavalla, oli kyseessä realistinen tai fyysisesti epärealistinen materiaali. (Kuva 28).



Kuva 28: Esimerkki yksinkertaisesta materiaalista pallossa käyttäen tavallista Diffuse-shaderia.

## 5.4 Materiaalin kehittäminen

Blenderin Cyclesissä materiaaleja voidaan rakentaa nodeilla, node editorissa. Erilaisia nodeja on lukuisia, ja niiden samantyyppisiä arvoja, kuten muuttujia ja värejä, voi kuljettaa muihin nodeihin yhdistämällä kaksi nodea janaalla. Nodet jaetaan kymmeneen pääryhmään: input, output, shader, texture, color, vector, converter, script, group ja layout. (Kuva 29). Shader-tyyppiset nodet määrittävät materiaalista oleellimmän osan, mitä muun tyyppisillä nodeilla voidaan soveltaa ja muuttaa.

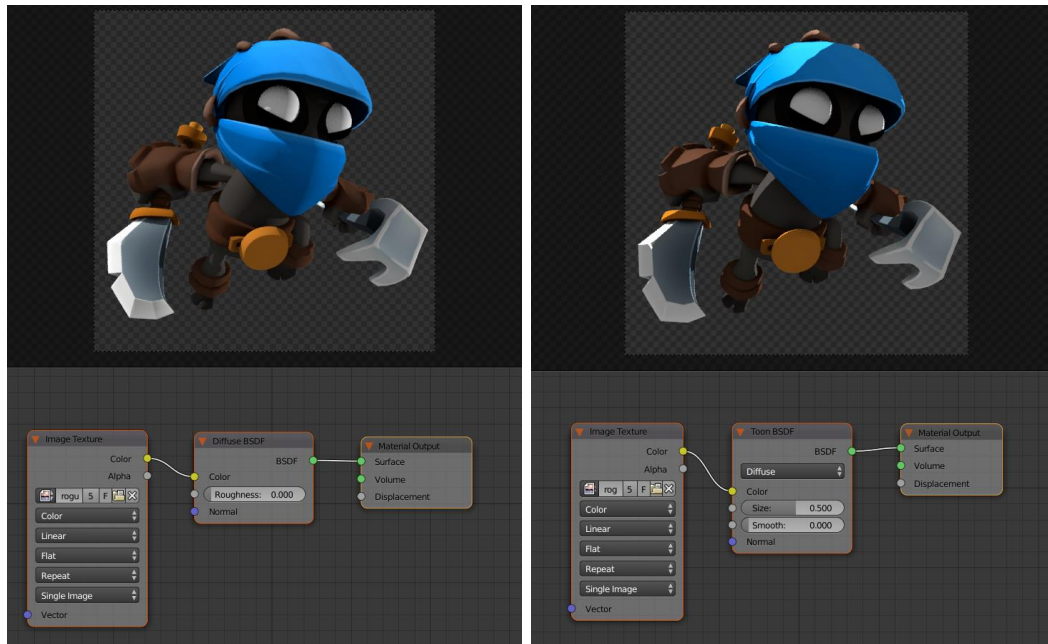


Kuva 29: Blender Cyclesin nodevalitsin.

Tutkin, miten saisin rakennettua suunniteltuun graafiseen tyyliin sopivan materiaalin. Referenssityyliä tutkimalla löytyi muutama tärkeä ominaisuus, mille olisi löydettävä toteutusratkaisu; toon-tyylissä keskeisimpinä elementteinä ovat terävä valaistuksen taitos, rajoitettu värisävyjen käyttö, karkkimaisen puhtaat pinnat sekä vahvat mutta ohuet kiillot kapaleiden valaistuissa reunoissa.

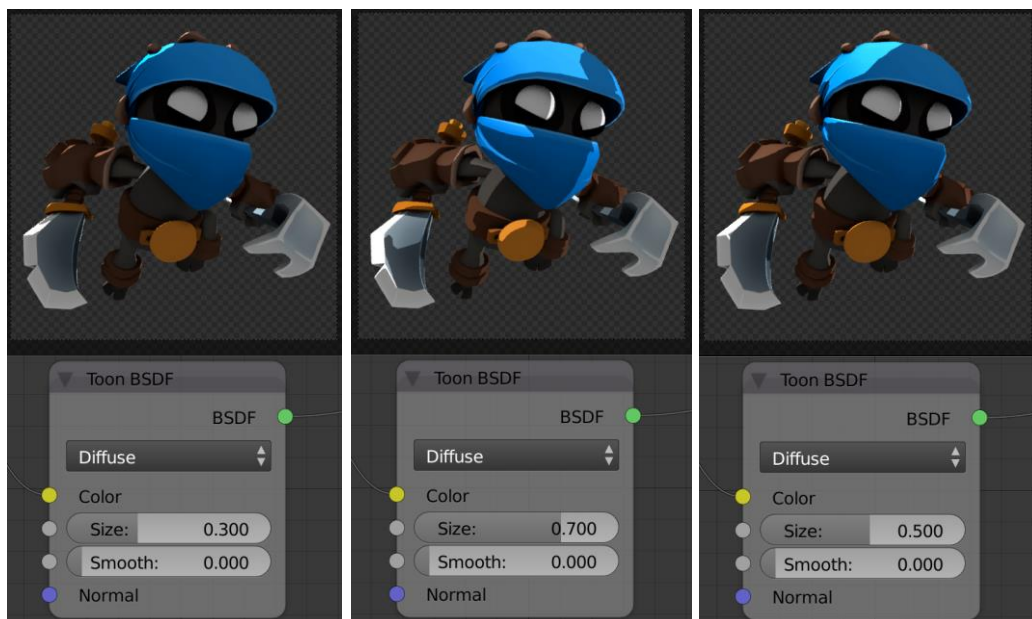
Perusta materiaalille löytyi Toon BSDF -shader nodesta, mitä käytetään terävöittämään materiaalissa ilmenevä valon ja varjon raja, sen sijaan että valaistuksen taittuma olisi pehmeä, kuten Diffuse BSDF-shaderilla. (Kuva 30). Asetin tämän noden käyttämään väriä,

mikä saatiin sille kuuluvasta tekstuuriedostosta Image Texture -nodella. Viimeiseksi shaderin data kulkeutui Material Output -noden surface-attribuuttiin, muuttaen kokonaisuuden näkyväksi materiaaliksi.



Kuva 30: Hahmomalli tavallisella diffuse-tyyppisellä shaderilla, sekä terävällä Toon BSDF-shaderilla.

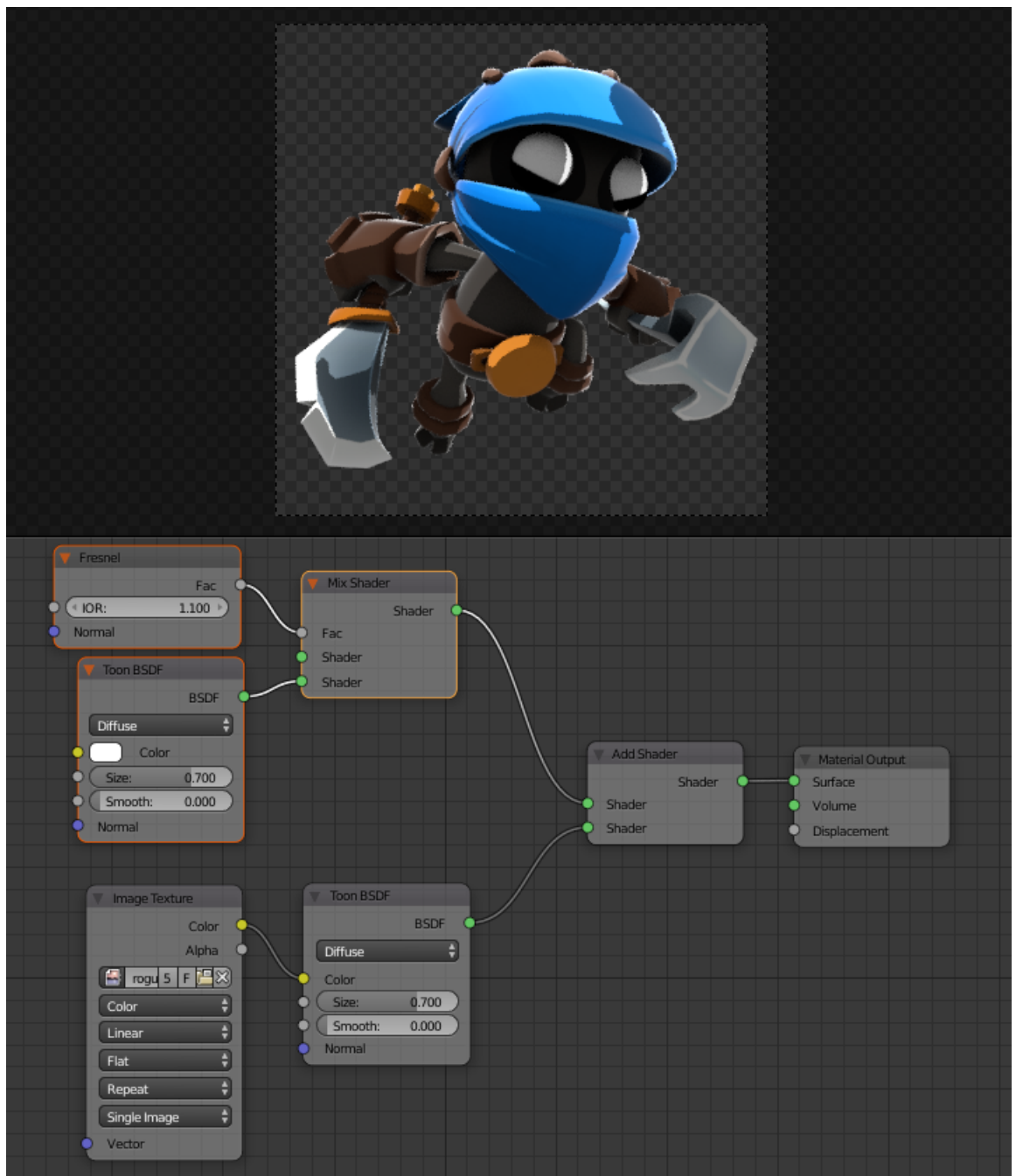
Kokeilin eri arvoja Toon BSDF-shaderin valon suuruuteen (size), ja päädyin käyttämään arvoa 0.700. (Kuva 31).



Kuva 31: Toon BSDF-shaderin valon suuruuden kokeilua arvoilla 0,3, 0,5 ja 0,7.

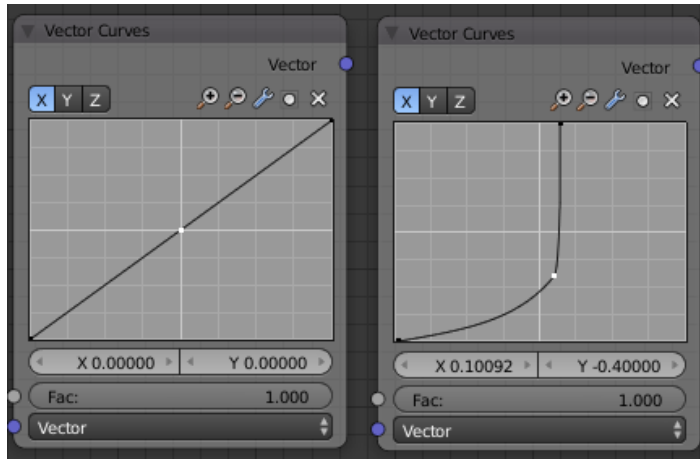


Seuraavaksi oli saavutettava kuvaan referensseissä ilmenevä ohut kiilto kappaleen valaistussa reunassa. Tähän hyödynsin Blenderin fresnel-nodea, mikä laskelmoi kuinka paljon valoa heijastuu tasolta, missä loput taittuvat tason läpi. Tuloksen arvoa voidaan käyttää esimerkiksi lisäämään fresnelin tuloksen alueelle toinen shader. Kokeilin tätä tapausta hahmomallin kanssa; fresnelillä on IOR (index of refraction) arvona 1.100, jotta toinen Toon BSDF-shader piirtyisi kappaleiden ulkoreunojen päälle. Tämän shaderin väriksi asetin puhtaan valkoisen, mukailemaan kiillon efektiä. Fresnel-noden shader lisättiin valmiiksi olevan Toon BSDF-shaderin päälle käyttämällä Add Shader-nodea. (Kuva 32).

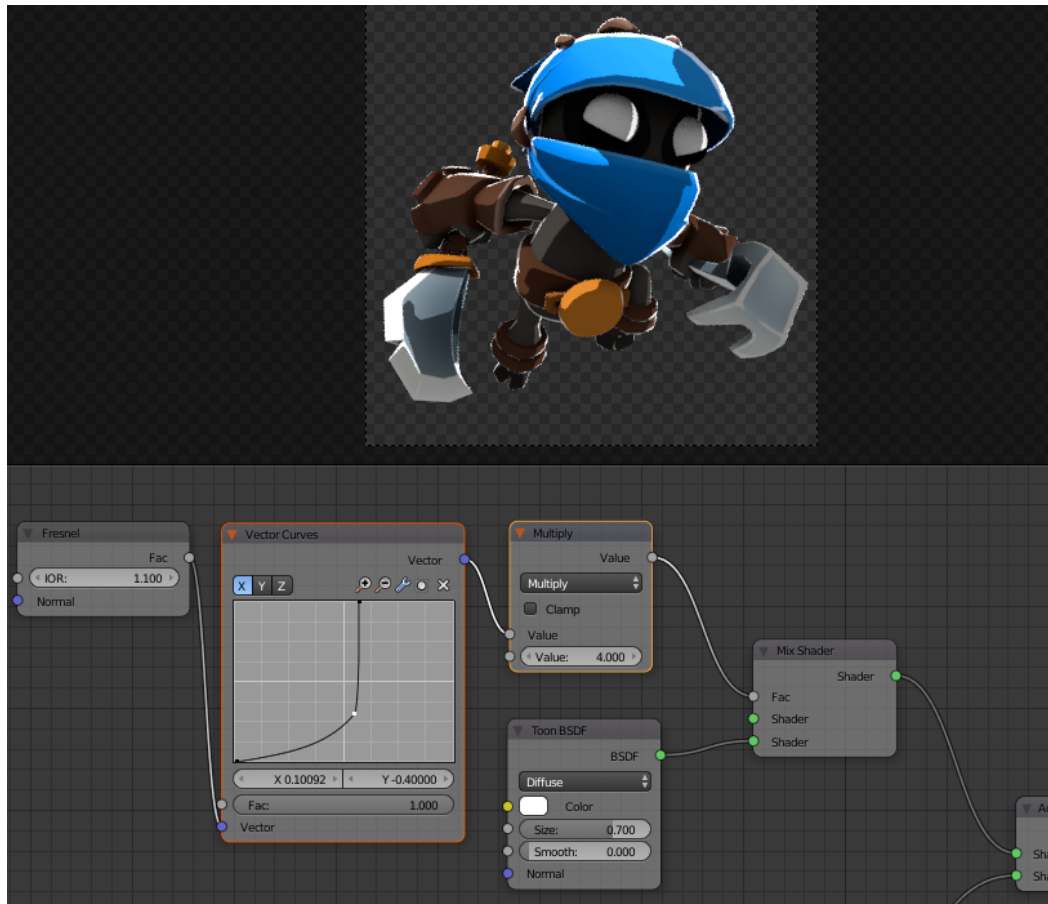


Kuva 32: Hahmomallin materiaali Fresnel-nodella saavutetun reunakiillon kanssa.

Tulos vaikutti hyvältä alulta. Halusin reunan kiillosta terävän pehmeän sijaan, joten fresnel-nodesta saatu data piti terävöittää ja moninkertaistaa. Tähän käytin Vector Curves -nodea (kuva 33), millä vähensin kaikkea muuta paitsi kaikista valkoista kiiltoa, ja Multiply-matematiikkanodea, millä Vector Curves-noden data kerrottiin neljällä. Tuloksena oli terävä ja valkoinen reunakiilto. (Kuva 34).



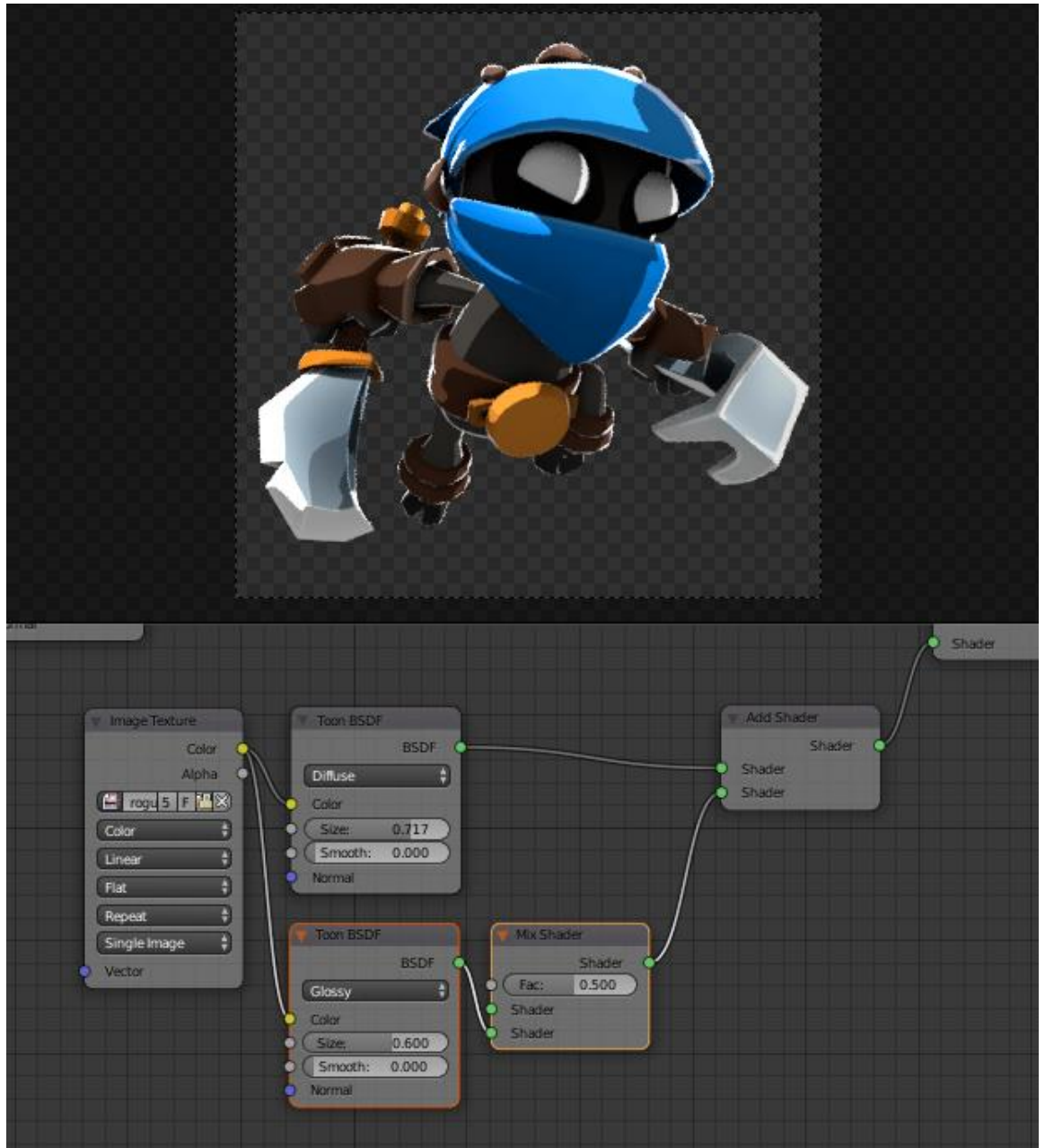
Kuva 33: Vector Curves -nodella pehmeästä fresnelin vaikutuksesta tehtiin tasainen.



Kuva 34: Hahmomallin materiaali terävän ja tasaisen reunakiillon kanssa.

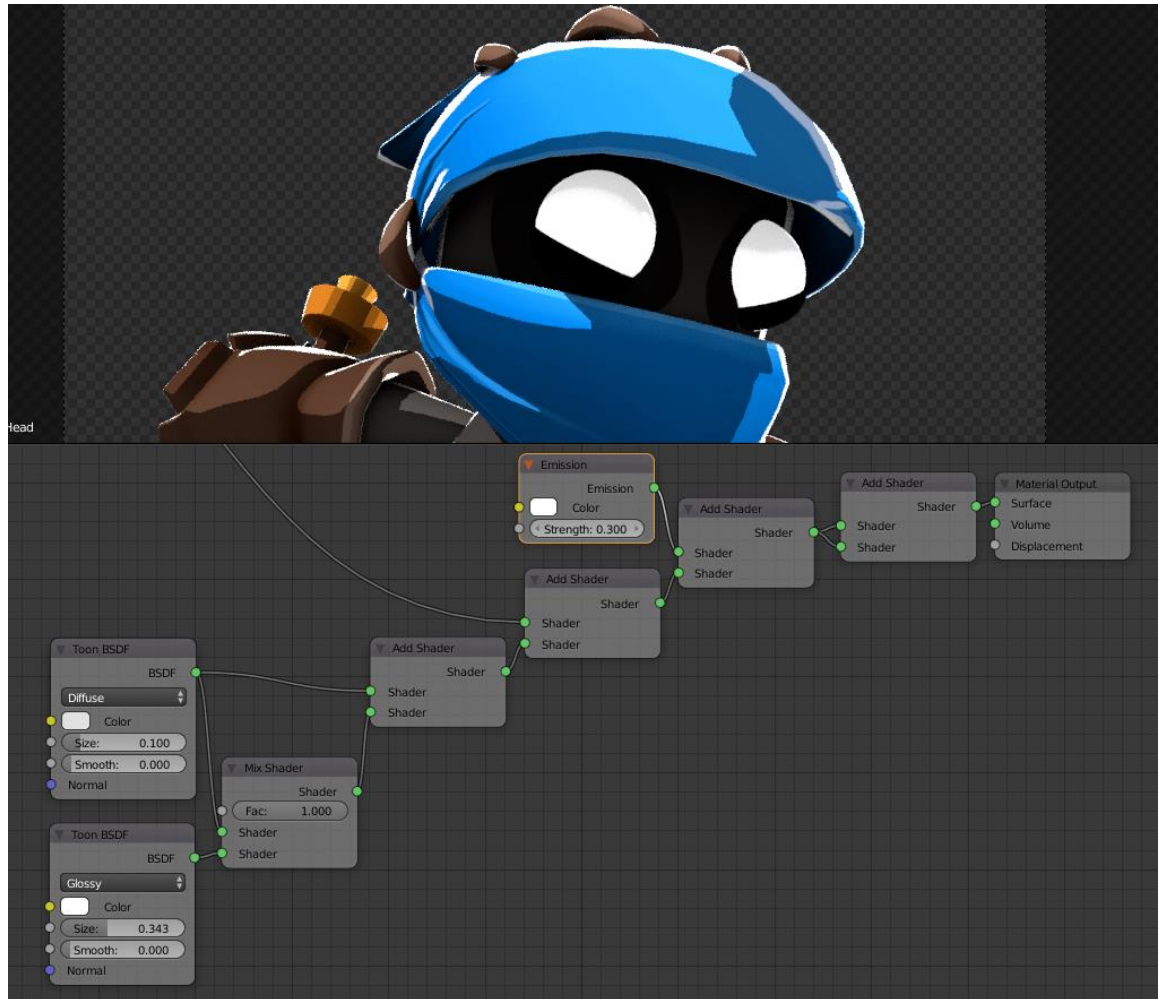


Seuraavana oli metallikappaleiden materiaali, missä ominaisuutena olisi pinnan kiilto. Tähän mennessä metallimateriaali oli seurannut muiden materiaalien rakennetta, minkä rinnalle oli lisättävä uusi Toon BSDF-shader kiiltävänä (glossy) tavallisen diffuusin (diffuse) sijaan. Löysin glossy shaderin valon suuruudelle sopivaksi arvoksi 0.600, ja kasvatin diffuse shaderin valon suuruuden arvosta 0.700 arvoon 0.717. Uuden shaderin vaikutus oli liian voimakas, joten jatkoin sen nodea Mix Shader -nodella, arvolla 0.5, puolittaen glossy shaderin vaikutuksen materiaalissa. (Kuva 35).



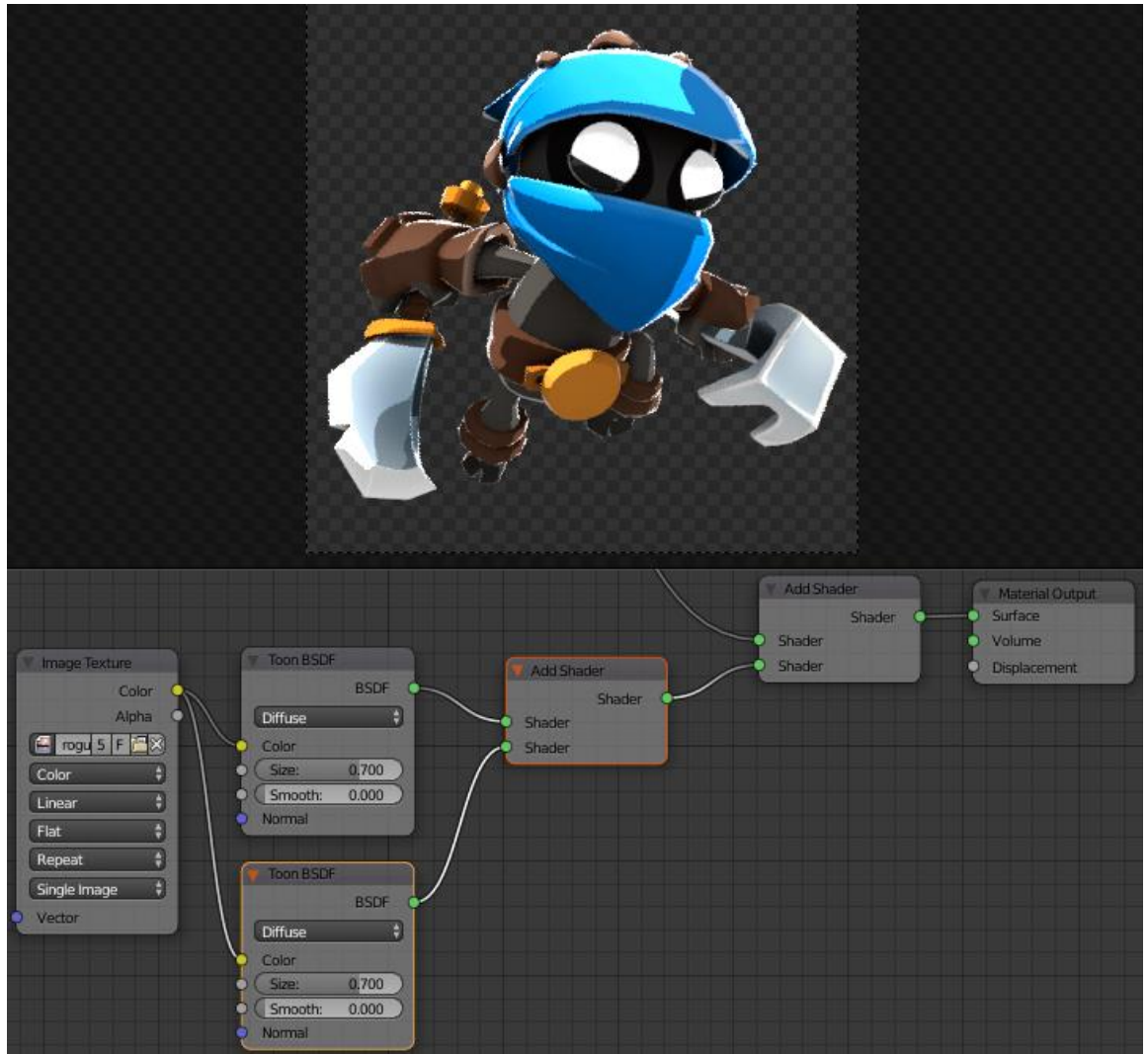
Kuva 35: Metallimateriaalissa diffuse shaderiin lisättiin myös kiiltävyyttä.

Jäljellä oli enää silmän materiaali. Hahmon silmät oli suunniteltu näyttämään lasimaisilta palloilta, joissa on himmeä hehku. Tätä varten päätin hyödyntää aikaisempaa metallimateriaalia ja tehdä siitä silmälle sopiva materiaali. Vähensin glossy shaderin valon suuruuden arvoksi 0.343 ja diffuse shaderin valon suuruuden arvoksi 0.100, jotta silmäpaloissa olisi posliinimainen valon taittuminen. Lisäsin materiaalille myös Emission-shaderin valkoisena, ja voimakkuudella 0.300, antaen silmille pienen valoemission. Koko materiaali vielä tuplattiin Add Shader -nodella, ja tulos näytti sopivalta. (Kuva 36).



Kuva 36: Silmien materiaali mukailee metallimateriaalin shaderien käyttöä.

Pääosilta kaikki materiaalit olivat nyt kunnossa, mutta pyrin viemään materiaalia vielä lähemmäksi haluttua tyyliä. Hahmo oli hiukan liian pimeä, joten karkkimaisuus tavoitteena päätin tuplata hahmon kangas-, iho- ja metallimateriaalien väriä käyttävän Toon BSDF-shaderin yhdistämällä kaksi kyseistä shaderia Add Shader -nodella. (Kuva 37).



Kuva 37: Hahmon kangas-, iho- ja metallimateriaalien Toon BSDF -shaderien vaikutus tuplattiin luomalla toinen samanlainen shader ja lisäämällä se alkuperäisen päälle Add Shader -nodella.

Korttikuvan kokonaisuus oli nyt valmiina renderoitavaksi.

## 6 Renderointi

Blenderissä renderointi on automaattinen prosessi, missä kameran kuvaama 3D-ympäristö käännetään 2D-kuvaksi. Renderointiin voi vaikuttaa lukuisilla asetuksilla. Tarkoituksena oli löytää korttikuvia varten sopivat asetukset mitkä pysyisivät vakioina kaikkia pelin renderoitavia korttikuvia varten.

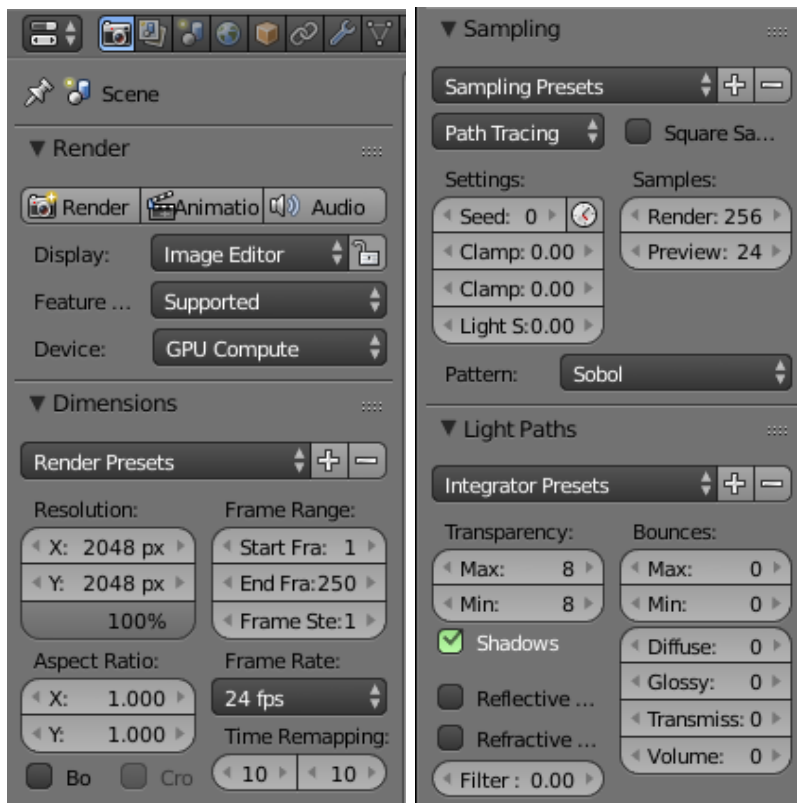
### 6.1 Laatiminen

Renderointiasetukset riippuvat aina renderoitavasta scenestä sekä halutusta lopputuloksesta, joten täydellisiä asetuksia ei ole olemassa. Blender tarjoaa mahdollisuuden muuttaa kaikkia mahdollisia asetuksia mitkä vaikuttavat renderoinnin laskelmointiin, mikä on erinomainen ominaisuus ohjelmassa, mutta myöskin monimutkaistaa menetelmää. Oletusasetukset on määritelty sellaisiksi, että käyttäjä voisi suorittaa renderoinnin mahdollisimman vaivattomasti. Silti parhaiten sopivat asetukset löytyvät usein yrityksen ja erehdyksen kautta.

Asetukset löytyvät Blenderin Properties-tilasta, Scene-välilehdestä, mikä on oletuksena esillä käyttöliittymän oikeanpuolisella sivulla. Erilaiset asetukset on jaoteltu useampaan ryhmään. Kehitysprosessiin oli muutettava seuraavia asetusrhymiä: Render, Dimensions, Sampling sekä Light Paths. (Kuva 38).

Ensimmäiseksi asetin renderoinnin käyttämään näytönohjainta (GPU) laskelmointilaitteena suorittimen (CPU) sijaan, sillä se oli käyttämässäni järjestelmässä tehokkaampi ratkaisu.

Seuraavaksi asetusrhmästä dimensions (mittasuhteet) lopullisen renderoidun kuvan resoluutio oli vaihdettava sopimaan kuvan jatkokäyttöä varten. Asetin resoluutioksi 2048 x 2048 pikseliä, mikä on suhteellisen suuri resoluutio pienelle korttikuvalle. Kuvaa oli käsiteltävä erikseen renderoinnin jälkeen, missä lopullinen kuva tallennettaisiin kokoon 440 x 480 pikseliä. Yleisesti graafikkona projektin tulevaisuuden kannalta on hyvä pitää työvaiheessa käytetty grafiikka suuremmassa kuvasuhteessa ja tarkkuudessa kuin mitä ajattelisi tarvitsevan. Korttikuvia esimerkiksi voidaan tarvita myöhemmin materiaalina markkinointiin sekä muihin kuviin, jolloin materiaalin tarvitsee usein olla laadukasta.



Kuva 38: Kehitysprosessiin vaikuttavat asetusrhyhmät arvoineen.

Renderoinnin laskelmointiprosessiin vaikuttavista asetuksista keskityin ryhmiin Sampling ja Light Paths. Samples vastaa renderoinnissa suoritettuja otoksia, joilla pyritään saavuttamaan haluttua vastaava lopputulos mahdollisimman vähäisellä kohinalla (noise). Kohina johtuu monimutkaisesta valaistuksen laskelmoinnista, ja sitä voi vähentää asettamalla suuremman arvon renderoinnin sampleille. Käytin lopullisessa renderoinnissa 256 samplea (render), mutta 3D-editoritilan aktiivisessa renderoinnissa vain 24 samplea (preview), koska käytännössä samplejen määrä on suoraan verrannollinen renderoinnin raskauteen. Muut asetukset sampling-ryhmästä jätin oletusarvoihin, sillä ne eivät olleet relevantteja tuloksen kannalta.

Light Paths -ryhmän asetukset vastaavat valosäteiden kimpoamisesta erilaisilla pinnoilla. Näillä asetuksilla voidaan määrittää scenessä ilmenevän valon leviämistä. Toon-tyyliä seuraten tätä ei kuitenkaan tarvittu, joten asetin kaikki kimpoamisarvot nolnaan.

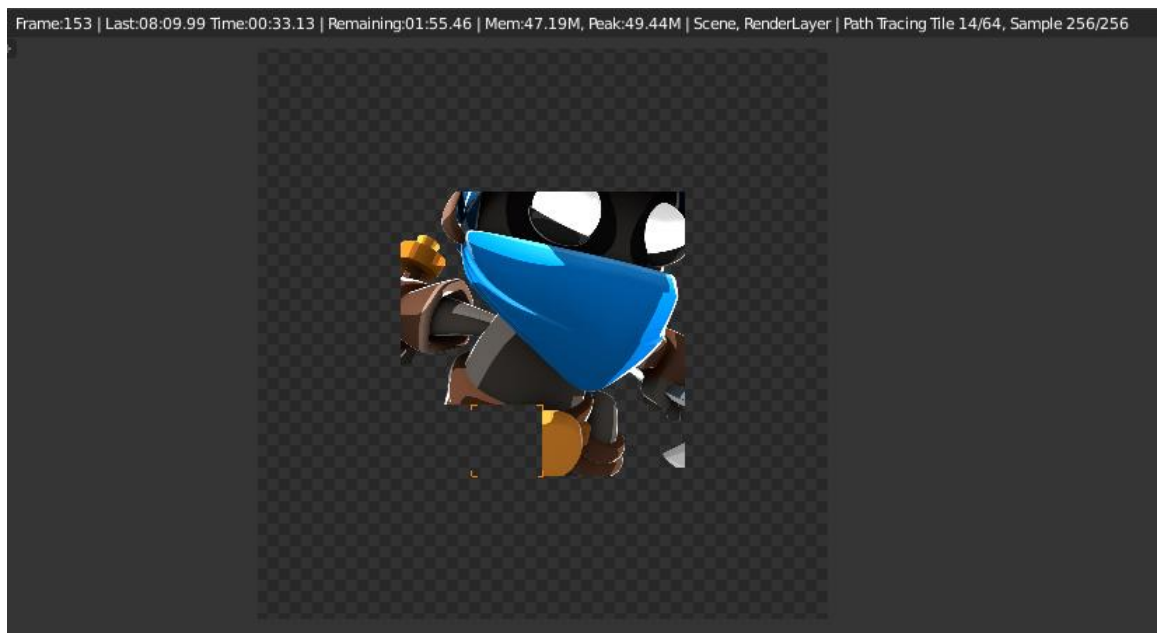
Viimeiseksi hahmon valmis kuva oli renderoitava läpinäkyvälle taustalle, jotta sen hyödyntäminen jälkikäsitteilyssä olisi helppoa. Tämä myös vähentää renderoinnin työtä, sillä vain hahmoon kuuluvat pikselit on renderoitava. Taustan läpinäkyväksi muuttaminen onnistui asettamalla Film-asetusrhyhmän Transparent-asetuksen päälle. (Kuva 39).



Kuva 39: Renderoidun kuvan käyttämätön tausta muutettiin läpinäkyväksi Film-asetusryhmän Transparent-asetuksella.

## 6.2 Tulos ja jälkikäsittely

Kuvan lopullinen renderointi oli hidasta, suuren kuvan resoluution sekä korkean sample-arvon takia. Prosessi itsessään kesti 81.95 sekuntia. (Kuva 40).



Kuva 40: Renderointi käy läpi jokaisen pikselin ruudukon muodossa.

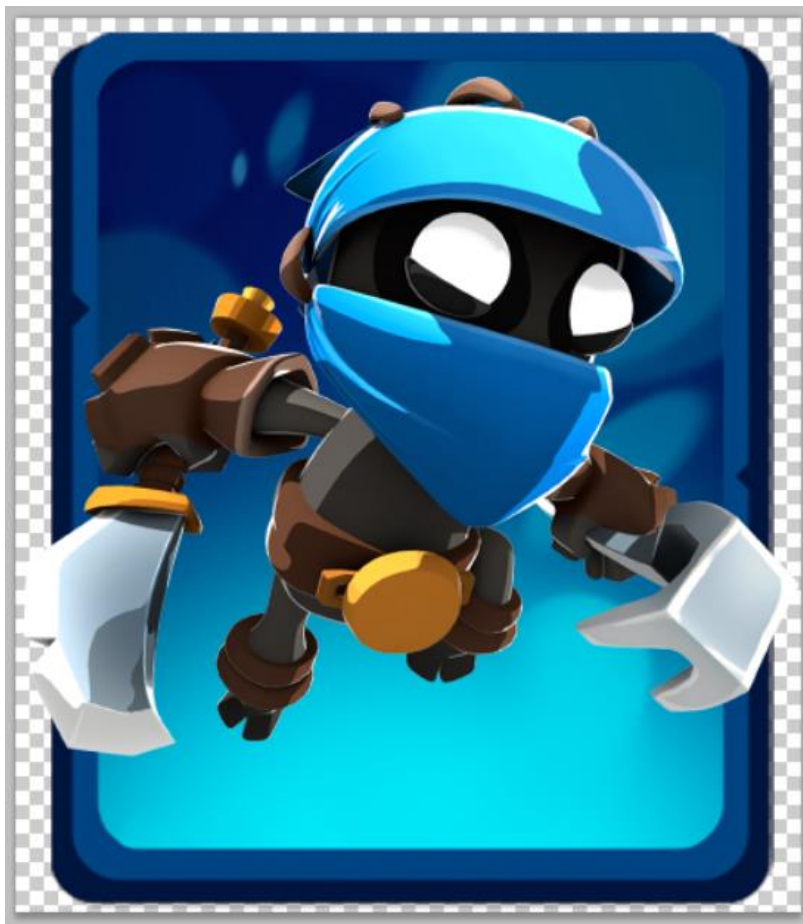
Lopullinen kuva tallennettiin PNG muotoon, mikä säilyttää kuvan läpinäkyvyyden. Seuraavaksi kuva oli vietävä kuvanmuokkausohjelmaan Adobe Photoshop, missä renderoidusta kuvasta oli tehtävä pelissä käytettävä korttikuva.

Peliprojektin korttikuville oli jo kehitetty kortin taustagrafiikka, joten avasin kuvankäsittelyohjelmaan pohjaksi valmiin korttigrfiikan ja toin toiselle kuvakerrokselle (layer) renderoidun hahmon. (Kuva 41). Lopulta korttikuva oli valmis peliä varten. (Kuva 42).





Kuva 41: Renderoitu kuva hahmosta lisättiin valmiin korttitaustan päälle.



Kuva 42: Valmis korttikuva kuvankäsittelyohjelmassa.

## 7 Loppusanat

Peliprojektin graafisesta ilmeestä vastaavat olivat mukana tarkastelemassa kehitysprosessin eri vaiheita. Näin lopputuloksen onnistuminen ilman suurempia muutoksia oli varmempaa. Korttikuva otettiin positiivisesti vastaan, joten kaikille muille hahmoille teetettiin myös korttikuvat tätä kehitysprosessia noudattaen. Lopputulos on suuri osa hahmojen sekä itse pelin identiteettiä. (Kuva 43).



Kuva 43: Kuvakaappaus Badland Brawl -pelin korttivalikkoruudusta.

Aiemmin korttikuva keskittyi digitaaliseen maalaamiseen, missä piirretyn tyylin lisäksi oli helppo lisätä ja muokata yksityiskohtia, mutta koko maalaamisprosessi oli aikaa vievä. Uudella korttikuvan kehitysprosessilla saavutettiin peliin sopivampi tyyli. Prosessi ja sen iterointi olivat nopeampia, varsinkin kun kuvia tuli tulevaisuudessa tehdä kymmenille eri hahmoille. (Kuva 44).





Kuva 44: Aikaisempi korttikuva (vas.) verrattuna uudempaan korttikuvaan (oik.).

## Lähteet

### Verkkolähteet:

Hachigian, Jennifer. (2005). Celshader FAQ. Web-dokumentti. Viitattu 12.5.2018. <http://www.celshader.com/FAQ.html>

Slick, Justin. (2017). What is Rendering – Finalizing the 3D Image. Web-dokumentti. Viitattu 14.4.2018. <https://www.lifewire.com/what-is-rendering-1954>

Slick, Justin. (2017). Guide to 3D Lighting Techniques for Digital Animation. Web-dokumentti. Viitattu 28.4.2018. <https://www.lifewire.com/3d-lighting-techniques-2090>

Trammel, Kent (2016). Big Idea: Blender Render VS Cycles. Web-dokumentti. Viitattu 1.5.2018. <https://cgcookie.com/articles/big-idea-blender-render-vs-cycles>

South, Phil. (2015). Mastering 3D Lighting in Blender. Web-dokumentti. Viitattu 5.5.2018. <https://www.maketecheasier.com/mastering-3d-lighting-blender/>

Blender Foundation. Manual – 3D View Introduction. (2015). Web-dokumentti. Viitattu 12.5.2018. <https://docs.blender.org/manual/en/dev/editors/3dview/object/introduction.html>

Blender Foundation. Manual – Materials Introduction. (2015). Web-dokumentti. Viitattu 12.5.2018. [https://docs.blender.org/manual/en/dev/render/blender\\_render/materials/introduction.html](https://docs.blender.org/manual/en/dev/render/blender_render/materials/introduction.html)

### Kuvalähteet:

Kuva 5. <https://itunes.apple.com/us/app/clash-royale/id1053012308>

Kuva 6. <http://nintendotoday.com/zelda-breath-of-the-wild-gameplay/>

Kuva 13. <https://docs.blender.org/manual/en/dev/render/cycles/introduction.html>

Kuva 15. <https://docs.blender.org/manual/en/dev/modeling/modifiers/generate/subsurf.html>

Kuva 16. <https://docs.blender.org/manual/en/dev/modeling/modifiers/generate/subsurf.html>