



Karelia-ammattikorkeakoulu
Medianomi (AMK)

Valaistuksen toteuttaminen 2D- peliympäristössä

Miikka Hämäläinen

Opinnäytetyö, Kesäkuu 2025

www.karelia.fi



OPINNÄYTETYÖ
Kesäkuu 2025
Media-alan koulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä(t)
Hämäläinen Miikka

Nimeke
Valaistuksen toteuttaminen 2D-peliympäristössä

Tiivistelmä

Opinnäytetyö käsittelee valaistuksen merkitystä 2D-peliympäristön suunnittelussa. Peliympäristöjä suunnitellessa valaistus kantaa useita rooleja niin esteettisinä kohteina kuin osana pelimekaniikkaa. Opinnäytetyössä tutkitaan näitä valaistuksen käyttökohteita ja kuinka valaistus voidaan toteuttaa peliympäristössä. Lisäksi opinnäytetyössä tutustutaan toiminnallisessa osassa käytettävän Godot-pelimoottorin valaistus- ja tekstuuriominaisuuksiin.

Opinnäytetyön toiminnallisessa osassa luotiin pelidemo, jossa 2D-peliympäristön valaistus toteutettiin hyödyntämällä työssä käytetyn Godot-pelimoottorin tarjoamia valaistus- ja tekstuuriominaisuuksia. Demoon koostettiin peliympäristö, jossa näitä ominaisuuksia voidaan demonstroida. Toteutusta suunnitellessa hyödynnettiin aiemmin tietopohjaa tutkiessa opittuja menetelmiä ja periaatteita. Demoa varten tarvittavien tekstuurien luontiprosessin suunnitelma, tuotanto ja koostaminen pelimoottoriin raportoitiin opinnäytetyön toiminnallisessa osassa.

Toiminnallisen osuuden aikana toteutetun demon pohjalta opinnäytetyössä pohditaan toteutuksen lopputulosta ja tuotannossa käytettyjen menetelmien ja ohjelmien mahdollista hyödyllisyyttä peliprojekteissa. Lisäksi opinnäytetyössä pohditaan, kuinka demon suunnittelu, tuotanto ja koostaminen onnistui ja missä asioissa on vielä parantamisen varaa. Lisäksi tarkastellaan henkilökohtaisia saavutuksia projektin aikana.

Kieli
suomi

Sivuja 42
Liitteet 2
Liitesivumäärä 4

Asiasanat
peliala, kenttäsuunnittelu, peligrafiikka



THESIS
June 2025
Degree Programme in Media

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600 (switchboard)

Author (s)
Hämäläinen Miikka

Title
Implementation of Lighting in a 2D Game Environment

Abstract

The topic of this thesis deals with the relevance of lighting while designing 2D game-environments. Lighting can carry several roles, from aesthetic elements to parts of the game mechanics. In this thesis these use cases are explored, as well as ways to implement the lighting into the game environment. In addition, the thesis also studies Godot's, the game engine used during in the project portion of the thesis, lighting and texturing properties.

During the practical part of the thesis, the goal was to create a game demo, in which the lighting of a 2D game environment was implemented using the lighting and texturing properties of the Godot game engine. In the game demo, an environment was created, in which the properties can be demonstrated. While planning the demo, the use cases and roles of lighting learned during the theoretical part of the thesis were taken into consideration. In addition, the process of planning, asset creation and implementation into the game engine were reported in the practical part of the thesis.

In this thesis, the result as well as the usefulness of both the production methods and the tools used in the project are analysed. In addition, the achievements and learning during the planning, production and implementation parts of the project as well as the improvement suggestions are also discussed in the conclusion. Finally, the personal achievements during the thesis are also covered in the conclusion.

Language
Finnish

Pages 42
Appendices 2
Pages of Appendices 4

Keywords
game development, level design, game graphics

Sisältö

Sanasto

1	Johdanto	7
1.1	Peliympäristön valaistus	7
1.2	Opinnäytetyön tavoitteet	7
2	Videopelien graafinen- ja kenttäsuunnittelu	8
2.1	Pelimekaniikka ja estetiikka ovat yhteen sitoutuneita	8
2.2	Pelien graafinen suunnittelu seuraa kuvatuotannon perinteitä	12
2.3	Kenttäsuunnittelulla rakennetaan vaikuttavia ympäristöjä	17
2.4	Kenttäsuunnittelun periaatteet	17
2.5	Godot -pelimoottori	21
3	Peliympäristön valaistuksen menetelmät	24
3.1	Valaistuksen merkitys peleissä	24
3.2	Valaistuksen elementit	25
3.3	Valaistuksen rakentamisen työkalut	25
4	Peliympäristön valaistuksen toteutus	26
4.1	Tuotanto alkaa suunnitelmasta	26
4.2	Ympäristön grafiikan tuottaminen	27
4.3	Demon koostaminen pelimoottoriin	29
5	Toteutuksen tulokset	32
5.1	Demon pohjatutkimuksen tulokset	32
5.2	Tuotantomenetelmien hyödyt	33
5.3	Henkilökohtaiset saavutukset	33
6	Pohdinta	34
6.1	Pelimoottorin valinta	34
6.2	Tuloksen pohdintaa	35
	Lähteet	37

Liitteet

Liite 1 GDD - Pelituotantosuunnitelma: Valaistus Demo

Liite 2 Kuvasarja valmiista demosta

Sanasto

2D	(eng. two-dimensional) Kaksiulotteinen. Ympäristö, jossa elementit liikkuvat kahdella, yleensä x ja y akselilla (Egenfeldt-Nielsen, Smith & Tosca 2008, 112–115).
3D	(eng. three-dimensional) Kolmiulotteinen. Ympäristö, jossa elementit liikkuvat kolmella, yleensä x, y ja z akseleilla (Egenfeldt-Nielsen, Smith & Tosca 2008, 112–115).
Kiiltokartta	(eng. Specularmap) Tekstuurien materiaali, joka määrittää millä tavoin valo heijastuu tekstuurista säätämällä pikseli kerrallaan niiden väriä ja kirkkautta (Puhakka, 237–239).
HUD	Heads-Up Display, huomautusnäyttö. Käyttöliittymän taso, joka esittää kriittisiä tietoja, kuten terveyden tai ammusten määrän, pelaajalle. On yleensä aina päällimmäisenä näytöllä (Linietsky, Manzur & the Godot community -yhteisö 2023).
Normaalikartta	(eng. Normalmap) Tekstuurien materiaali, joka määrittää mihin suuntaan valo heijastuu tekstuurista (Puhakka, 248–254).
Näkymä	(eng. Scene) Pelimoottoriin koottu ympäristö tai pelikenttä, joka voidaan ladata näytölle koodiin kirjattujen ohjeiden mukaisesti (Linietsky, Manzur & the Godot community -yhteisö 2023).
Näyttö	Pelaajan näkemän alueen äärikoko. Sisältää pelattavan ympäristön lisäksi yleensä myös HUD:in tai muita UI-elementtejä (Linietsky, Manzur & the Godot community -yhteisö 2023).
Parallaksitausta	Tausta, joka vierii eri tahtiin kuin pelaajan vierittämä pelikenttä (Linietsky, Manzur & the Godot community -yhteisö 2023).

Partikkeli	Visuaalisia hiukkasmaisia elementtejä, joiden avulla voidaan luoda luonnollisia ja monimutkaisia visuaalisia efektejä, kuten liekkejä tai kipinöitä. Elementit näytölle piirtää yleensä erikoistunut pelimoottorin moduuli, joka luo, animoi ja liikuttaa näitä partikkeleita annettujen ohjeiden mukaisesti. (Linietsky, Manzur & the Godot community -yhteisö 2023.)
Piirtoalue	Alue, jolle pelimoottori renderöi visuaaliset elementit. Voi olla suurempi kuin ruutualue, jonka pelaaja näkee (Egenfeldt-Nielsen, Smith & Tosca 2008, 115–119).
Ruutu	(eng. frame) Kuvat, joista animaatio koostuu. Ne ovat joko erillisiä kuvia tai koottuna spritesheet-tiedostoon. (Linietsky, Manzur & the Godot community -yhteisö 2023).
Sprite	Kaksiulotteisia tekstuureja, joita voidaan sijoittaa suurempaan näkymään ylikirjoittamatta taustaa täysin. Käytetään usein kaksiulotteisissa peleissä (Linietsky, Manzur & the Godot community -yhteisö 2023).
Sprite-animaatio	Animaatio, jolla sprite-elementit animoidaan toistamalla ruutu toisensa jälkeen (Linietsky, Manzur & the Godot community -yhteisö 2023.)
Spritesheet-tiedosto	Kuvatiedosto, jossa on visuaalisen elementin animaatio ruudut tai useampi erillinen visuaalinen elementti koottuna ruudukkoon. (Linietsky, Manzur & the Godot community -yhteisö 2023).
Tekseli	Kaksiulotteisen tekstuurikuvan pikseleitä. Sijoittuvat koordinaatistoon, josta voidaan etsiä vastaavat tekstuurikoordinaatit rasteripinnan muodostamiseksi (Puhakka 2008, 206–208).
Valaistus	Pikseleiden kirkkauden intensiivisyys graafisissa elementeissä (Puhakka 2008, 231–232).

1 Johdanto

1.1 Peliympäristön valaistus

Tämän opinnäytetyön aiheena on valaistuksen hyödyntäminen peleissä ja sen merkitys kenttäsuunnittelussa, etenkin 2D-peliympäristöjä suunniteltaessa. Opinnäytetyö käsittelee pelisuunnittelun lisäksi pelien estetiikan, kenttäsuunnittelun ja grafiikan tuotannon aiheita. Lisäksi opinnäytetyö uppoutuu myöhemmin toiminnallisessa osassa käytettävien teknologioiden, menetelmien ja ohjelmistojen ominaisuuksiin.

Kareliassa opinnäytetöitä pelikehitykseen on kirjoitettu aiemmin kuten myös digitaalisen grafiikan tuotantoon liittyen, mutta erityisesti 2D-ympäristöä, sen suunnittelun prosessia ja valaistuksen merkitystä peleissä ei olla aiemmin tutkittu opinnäytetöissä. Kareliassa aiemmin tuotettuja opinnäytetöitä, jotka käsittelevät samankaltaisia aiheita, ovat mm. Tanja Pekkarisen (2021) *PBR-materiaalin luonti proseduraalisesti ja valokuvan avulla 3D-malleille*, Annika Moision (2019) *Videopeleissä käytettävän pikseligrafiikan jatkuva suosio*, ja Petteri Holmin ja Pinja Ignatiuksen (2020) *Pelisuunnitteludokumentin luominen konseptitaiteen avulla*.

1.2 Opinnäytetyön tavoitteet

Opinnäytetyön tavoitteena on tutkia ja toteuttaa valaistusta 2D-peliympäristössä. Peleissä hyödynnettävien valaistusmenetelmien kehitystä, merkitystä ja toteutusta tutkitaan eri peliympäristöissä. Näitä seikkoja pohjustetaan myös tutkimalla kuvataiteen perinteitä, ja niihin sitoutuneita symboleja ja merkityksiä, joita voidaan hyödyntää myös digitaalisen grafiikan tuotannossa.

Toiminnallisen osuuden tavoitteena on luoda demo Godot-pelimoottoriin, jossa tutkittuja seikkoja voidaan soveltaa käytäntöön, ja keskeisiä valaistuksen menetelmiä voidaan demonstroida. Tavoitteena on tuottaa toiminnallinen demo,

jossa 2D-ympäristön valaistusta ja sen merkitystä kenttäsuunnittelussa voidaan havainnollistaa. Työn aikana tuotetussa demossa tullaan hyödyntämään 2D-peligrafiikan menetelmiä, Godot-pelimoottorin ominaisuuksia, kuin myös yleisesti 3D-grafiikan laskennassa hyödynnettäviä algoritmeja valaistuksen toteuttamiseksi. Työn lopputuloksen perusteella voidaan pohtia menetelmien tuotantotehokkuutta, osaamisvaatimuksia ja ajankäyttöä. Näistä seikoista voidaan lisäksi pohtia, onko menetelmistä hyötyä käytännössä, ja olisiko niistä hyötyä pelien kehittäjille peliprojekteissa.

2 Videopelien graafinen- ja kenttäsuunnittelu

2.1 Pelimekaniikka ja estetiikka ovat yhteen sitoutuneita

Egenfeldt-Nielsen, Heide Smith ja Pajares Tosca (2008) määrittelevät kirjassaan *Understanding Videogames – an essential introduction* kaikkien pelien, niin analogisten kuin digitaalisten, sisältävän sarjoja sääntöjä. Pelejä ei voida kuitenkaan määrittää ainoastaan sarjaksi sääntöjä. Jos pelit määriteltäisiin ainoastaan sarjaksi sääntöjä, tämä kapea määritelmä tarkoittaisi, että mikä tahansa peli voitaisiin siirtää mihin tahansa mediatyyppiin, joka ei ole mahdollista todellisuudessa. Tähän kirjassa mainitaan pelisuunnittelijoiden Eric Zimmermanin ja Katie Salenin näkemys sääntöjen merkityksestä pelin kokonaisuuteen, jossa media-analyysissä pelit tulisi jakaa kahteen analyttiseen tasoon, muodolliseen (formal) ja fenomenologiseen (phenomenological) tasoon. Muodollisella tasolla pelit käsitellään edellä mainittuina sarjoina sääntöjä. Tämä taso ei kuitenkaan ole riittävä kuvailemaan peliä kokonaisuutena. Tästä syystä on tärkeää myös tutkia pelejä fenomenologisella tasolla, johon sisältyvät pelien perusestetiikan seikat. Pelimekaniikka ja pelien estetiikka ovat siis toisiaan sitovia ja täydentäviä elementtejä pelejä luodessa. (Egenfeldt-Nielsen, Smith & Tosca 2008, 97–107.)

Pelin perspektiivi määrittää mistä tai kenen näkökulmasta peliä pelataan. Kyseessä ei kuitenkaan ole ainoastaan narratiivinen seikka vaan pelien perspektiivi vaikuttaa paljon niin pelimekaanisiin kuin esteettisiin valintoihin peliä

kehittäessä. Ainoina poikkeuksina ovat täysin tekstipohjaiset pelit ja abstraktit pulmapelit, joissa pelaajan perspektiivi on joko tilaton tai mahdotonta ilmaista visuaalisesti. Pääsääntöisesti perspektiivi voidaan jakaa kahteen pääosa-alueeseen, ensimmäiseen ja kolmanteen persoonaan. (Egenfeldt-Nielsen, Smith & Tosca 2008, 107–111.)

Näistä kahdesta useampia genrejä kattaa kolmannen persoonan perspektiivi, jossa pelaaja tarkkailee pelin kulkua pelielementtien ulkopuolelta, esimerkiksi hahmo, ajoneuvo tai kokonainen armeija. Pelaaja itsessään ei toimi suoraan hahmon kengissä, vaan käyttää hahmoa itsensä jatkeena manipuloidakseen pelimaailmaa. Käyttökohteiden laajuutensa vuoksi voidaan kolmannen persoonan perspektiivi jakaa myös useaan visuaaliseen perspektiiviin. Näihin kuuluu esimerkiksi isometrinen- (isometric), kavaljeeri- (bird's eye), sivuvieritys- (side-scrolling), takaa-ajo- (chase) ja olanyliperspektiivi (shoulder). Laajan näkökulman ansiosta pelaajalla on yleensä myös suuri määrä informaatiota, ellei toinen pelimekaniikka ole asetettu rajoittamaan tiedon määrää. Esimerkiksi pelaajahahmo itse ei voi nähdä aarretta seinän sisällä heidän näkökulmastaan, mutta pelaaja voi nähdä aarteen ja sitä myös vartioivan hirviön, ja täten ohjata pelihahmoa aarretta kohti vaaraa välttäen. (Egenfeldt-Nielsen, Smith & Tosca 2008, 107–111.)

Toisin kuin kolmannen persoonan perspektiivin pelit, ensimmäisen persoonan perspektiivi laittaa pelaajan suoraan pelihahmon kenkiin. Perspektiivi tarjoaa hektisen ja spontaanisen jopa intiimin näkökulman pelimaailmaan. Pääasiassa ensimmäisen persoonan pelit rajoittuvat kolmiulotteiseen peliympäristöön. Tämän lisäksi puhtaasti ensimmäisen persoonan perspektiiviin sijoittuvat pelit ovat yleensä rajoittuneita tiettyihin genreihin. Ensimmäisen persoonan perspektiivi myös tarjoaa vähemmän informaatiota pelaajalle kuin kolmannen persoonan perspektiivi. Edellisen perspektiivi esimerkin mukaisesti pelaaja ei voi tietää aarteista tai vaaroista ilman edeltävää tietoa, sillä perspektiivi on rajoittunut pelaajahahmon näkökulmaan. (Egenfeldt-Nielsen, Smith & Tosca 2008, 107–111.)

Pelien kehittyessä eri perspektiivien tärkeyden huomioiminen on noussut osaksi pelimekaanista ja -esteettistä suunnitteluprosessia. Esimerkiksi monissa strategiapeleissä havaintopiste pysyy yleensä pelaajan joukkojen yllä kolmannen persoonan perspektiivissä. (Egenfeldt-Nielsen, Smith & Tosca 2008, 107–111.)

Ajan myötä, ja teknologian kehittyessä, tiukka sitoutuminen yhteen perspektiiviin on muuttunut suurilta osin pelisuunnittelun kysymykseksi. Se kuinka monta perspektiiviä pelaajalle on sallittu ja missä tilanteissa riippuu paljon siitä, mihin pelin kehittäjät haluavat sijoittaa pelaajan havaintopisteen. Esimerkiksi monet kolmannen persoonan kolmiulotteiset tasohyppelypelit sisältävät ominaisuuden siirtyä ensimmäiseen persoonaan, jotta pelaaja voi tarkastella ympäristöään tarkemmin, ja monet modernit roolipelit, kuten Bethestan kehittämät Fallout- ja Elder Scrolls -pelit, tarjoavat ensimmäisen ja kolmannen persoonan perspektiivin pelaajalle, joiden välillä pelaaja voi helposti vaihtaa toisesta toiseen pelitilanteen mukaan. Havaintopisteen sijoittamiseen vaikuttavat pelimekaaniset tekijät, kuten kuinka perspektiivi vaikuttaa pelaajien peliympäristön ymmärtämiseen ja kuinka paljon informaatiota perspektiivillä annetaan pelaajille. (Egenfeldt-Nielsen, Smith & Tosca 2008, 107–111.)

Pelien perspektiivillä on myös suuri vaikutus pelien estetiikkaan. Se mistä suunnasta pelimaailmaa ja sen sisältämiä elementtejä tarkastellaan, ja määrittää, millaista grafiikkaa peliin tarvitaan. Perspektiivi vaikuttaa myös siihen, kuinka lähelle pelihahmoja tai -elementtejä sekä niiden toimintaa pelaaja pääsee. (Egenfeldt-Nielsen, Smith & Tosca 2008, 107–111.)

Perspektiivien lomassa nousee kysymys ulottuvuudesta. Pelit jakautuvat kahteen eri ulottuvuus tyyppiin, kaksi- ja kolmiulotteisiin (2D/3D). Ulottuvuuksien välinen ero peleissä esiintyy tuotantomenetelmissä, käytetyissä grafiikkaohjelmissa ja kuinka grafiikka renderöidään pelimooottorissa. Eroa kaksi- ja kolmiulotteisen välille voi tosin olla ajoittain vaikea määrittellä erilaisten esteettisten ratkaisujen ansiosta. (Egenfeldt-Nielsen, Smith & Tosca 2008, 112–115.)

Kaksiulotteinen grafiikka on verrattavissa perinteisimpiin kuvatuotannon teoksiin, kuten taidemaalauksiin tai julisteisiin. Kaksiulotteinen grafiikka renderöidään, skaalataan ja muunnetaan x- ja y-akseleilla. Ne voivat olla joko pikseleistä koostuvaa rasteroitua grafiikkaa eli bittikarttoja, tai matemaattisesti määriteltyä vektorigrafiikkaa.

Erot näiden tiedostomuotojen välillä ovat niiden muunneltavuus, skaalattavuus, tiedostokoko ja renderöintiin vaadittava teho. Rasteroitu grafiikka hyödyntää yleensä samoja bittikarttatiedostomuotoja kuin monet tietokoneilla olevat kuvat, esimerkiksi png- tai jpg-tiedostomuodot (Mozilla 2024). Bittikarttojen vahvuutena on niiden suhteellisen pieni tiedostokoko ja laitevaatimukset niiden renderöimiseksi. Vektorigrafiikka taas hyödyntää omia tiedostomuotojaan kuten eps-, pdf- ja svg-muotoja (FileInfo.com, 2024). Niitä käytetään yleensä taulukoiden ja datan visualisoinnin tuottamiseen.

Kolmiulotteinen grafiikka sisältää kolmannen akselin, jolla grafiikkaa voidaan renderöidä, skaalata ja muuntaa. Verrattuna kaksiulotteiseen grafiikkaan, kolmiulotteinen grafiikka yleensä vaatii korkeampia laitevaatimuksia.

Kolmiulotteista grafiikkaa voidaan tosin skaalata myös vähemmän tehokkaille järjestelmille. (Egenfeldt-Nielsen, Smith & Tosca 2008, 112–113.)

Kolmiulotteiset graafiset objektit tallennetaan omiin tiedostomuotoihinsa kuten obj- ja abc- muodot (Blender Foundation 2024).

Verrattuna suoraan renderöityyn kolmiulotteiseen grafiikkaan, kaksiulotteinen grafiikka, etenkin rasteroitu grafiikka, ei vaadi yhtä paljon suoritustehoa sitä renderöivältä laitteistolta. Tästä, ja joskus esteettisestä, syystä kolmiulotteinen vaikutelma voidaan saavuttaa myös kaksiulotteisessa peliympäristössä.

Vaikutelman saavuttamiseksi voidaan käyttää eri perspektiivejä kuten isometristä perspektiiviä, ja parallaksivieritystä, jossa pelaajaa lähempänä olevat kohteet liikkuvat nopeammin kuin pelaajahahmon takana olevat kauemmat kohteet. (Egenfeldt-Nielsen, Smith & Tosca 2008, 112–115.)

2.2 Pelien graafinen suunnittelu seuraa kuvatuotannon perinteitä

Peligrafiikan perustana toimii vuosikymmenten, ellei -satojen, aikana kehittynyt kuvataiteiden perinne. Teoksessa *Game Art Complete* (Gahan 2008) kerrotaan kuvataiteen luomasta perustasta. Luke Ahearnin (2008) mukaan kuvataiteiden perinteiden ja peruseriaatteiden ymmärtäminen on keskeinen osa peligrafiikan tuottamisen oppimisprosessia, ja hyödyllinen peligrafiikan tuottajien välisessä kommunikaatiossa. Artikkelissaan Ahearn keskittyy viiteen peruseriaatteeseen: perspektiivi, valo ja varjo, muoto, tekstuuri ja väri. (Gahan 2008.)

Visuaalisten elementtien perustana ovat muodot. Kun katsot mitä tahansa kohdetta, olivat ne tosimaailmassa tai digitaalisia, voidaan ne rikkoa erilaisiin perusmuotoihin. Tämä helpottaa kohteiden kokonaisuuden hahmottamista. Muotoja voi olla kaksi- ja kolmiulotteisia, ja grafiikkaa voidaan muokata molemmissa ulottuvuuksissa ohjelmiston mukaan. Näistä yleisempi on kuitenkin kaksiulotteisten visuaalisten elementtien ja materiaalien luonti, sillä kolmiulotteista grafiikkaa luodessa luodaan mallille myös kaksiulotteista grafiikkaa hyödyntävät pintamateriaalit. (Gahan 2008.)

Visuaalisia elementtejä tuottaessa valon ja varjon hyödyntäminen on yhtä tärkeää kuin perinteisessä kuvataiteessa. Valoa ja varjoa hyödyntämällä voidaan eri graafisiin elementteihin luoda syvyyttä ja ne toimivat yhtenä osana määrittämässä elementtien tekstuuria. Niiden avulla voidaan myös hahmottaa kohteiden muotoa niin kaksi- kuin kolmiulotteisessa ympäristössä, ja tätä voidaan hyödyntää erilaisten ulottuvuus efektien luonnissa, kuten kolmiulotteisen vaikutelman luonti kaksiulotteiseen grafiikkaan. Toisin kuin perinteisessä kuvataiteessa, peligrafiikan valaisu ja varjot ovat dynaamisempia erilaisten pintamateriaalien, kuten kiilto-, normaali- ja korkeuskartat, ja eri varjostimien ansiosta. (Gahan 2008.)

Peligrafiikassa tekstuurilla tarkoitetaan visuaalisia elementtejä sisältäviä tiedostoja, kuten materiaali- tai spritesheet-tiedostot. Tekstuurilla voidaan kuitenkin tarkoittaa, etenkin perinteisessä kuvataiteessa, myös luodun

elementin pinnan tuntua. Perinteisesti taiteen tekstuuri on jaettu kahteen tyyppiin: visuaaliseen ja taktiiliseen tekstuuriin. (Gahan 2008.)

Taktiilisella tekstuurilla tarkoitetaan taiteen tulkitsemista fyysisen kontaktin kautta. Taktiiliseen tekstuuriin sisältyy esimerkiksi kohteesta tunnettu kovuus, sileyks ja lämpö. Pelit eivät vielä tosin hyödynnä tätä tekstuurin aspektia kovin laajasti, ainakaan tavalla, jossa pelaaja itse pääsisi tuntemaan tekstuureja pelin sisällä. Monissa peleissä pelaajan ohjaama kohde, kuten pelihahmo, voi tosin kokea esimerkiksi pinnan tahmaisuuden, kovuuden tai lämmön, jos tämä on ohjelmoitu reagoimaan eri tavoin erilaisiin materiaaleihin. (Gahan 2008.)

Visuaalista tekstuuria taas tulkitaan täysin visuaalisten aistimusten ja materiaalmielikuvien kautta. Visuaalista tekstuuria tarkastellessa syntyy psykologinen ilmiö, jossa mieli luo vaikutelman pinnan tunnusta myös ilman taktiilista kontaktia. Marmoritekstuurin pinta voi esimerkiksi tuntua kovalta ja sileältä, kun taas kankainen tekstuuri voi tuntua pehmeältä ja karhealta. Pelien visuaalisten elementtien ja grafiikan tuottajien osalta tämä on merkittävämpi tekstuurin aspekti. Elementtien visuaalisen tekstuuriin liittyvän tuntuman vaikutus merkitsee paljon elementtien laatuun. (Gahan 2008.)

Eri pintojen visuaalista tekstuuria voidaan tehostaa valaistuksen avulla. Kovilla varjoilla voidaan luoda tuntemusta karheasta materiaalista, kun taas pehmeämmät varjot luovat tuntemusta tasaisemmasta materiaalista. Valaistus voidaan toteuttaa käyttämällä elementin pinnan tekstuuria, tai muita lisätekstuureja. Antti Puhakan teoksen *3D-grafiikka* (2008) mukaan tekstuuri koostuu tekseleistä. Tekselit ovat tekstuurikuvan pikseleitä sijoitettuna rasteripinnan koordinaatistoon. Tämä tarkoittaa, että pinnan koordinaatistoon sijoitetaan tekstuurikuvan vastaavat tekstuurikoordinaatit. Näille tekstuurikoordinaateille voidaan määrittää väriarvo, joka taas siirtyy rasteripinnalle. (Puhakka 2008, 206–207.)

Rasteripinta itsenäään toimii diffuusissa eli siihen osuvan valon oletetaan hajaantuvan pinnasta satunnaisesti suuntiin, joten valo heijastuu katsojaan yhtä voimakkaasti suunnasta riippumatta, ja tekstuurin jokainen teksteli reagoi

valaistuksen muutokseen samalla tavoin. Tämä ei kuitenkaan ole välttämättä haluttu tulos. Pinnan materiaalin reaktioon valaistuksen muutokseen voidaan vaikuttaa lisätekstuurien avulla. Näitä ovat esimerkiksi spekulaaari-, normaali- ja korkeuskartat. Niillä voidaan määrittää, millä tavoin pinnan materiaalin tekselit reagoivat valaistukseen. (Puhakka 2008, 231–250)

Yleisesti grafiikan tuotannossa, niin kaupallisessa, viihteellisessä kuin taiteellisessa tuotannoissa, väri on ehkä tärkein aspekti tuotetusta grafiikassa. Värien vuorovaikutusta ja psykologiaa on tutkinut sveitsiläinen taitelija ja väritutkija Johannes Itten 1920-luvulta vuosisadan puoliväliin saakka. Itten oli kiinnostunut värien vuorovaikutuksesta, ihmismielen luomista assosiaatioista väreihin liittyen, ja siitä mitä eri värit saavat ihmiset tuntemaan. (Itten 1998, 5–6.)

Esineiden pinnat todellisessa elämässä imevät ja heijastavat niihin osuvaa valoa. Tähän ominaisuuteen perustuu vähentävä eli subtraktiivinen värinmuodostus järjestelmä, jossa niin kohteiden pintojen molekyyliden tai maaliaineen pigmenttien imetty ja heijastama valo muodostaa kohteen havaittavan värin. (Itten 1998, 16) Pelituotannossa subtraktiivista järjestelmää hyödynnetään yleisimmin käsin piirretyissä konseptitaiteessa, ja vain harvoin itse pelissä käytettävän grafiikan luonnissa. Esimerkiksi pelin *Cuphead* (2017) kehittäjät juhlivat käsin piirretyn animaatiotaiteen kulta-ajan perinteitä tuotantomenetelmiin saakka (StudioMDHR Entertainment, 2017.), ja seikkailupelin *Neverhood* (1996) graafiset elementit olivat alun perin muovailtu oikeasta muovailusavesta, jotka peliä kehittänyt tiimi animoi stopmotion -menetelmin ja muunsi digitaaliseen muotoon (The Neverhood Inc, 1996.)

Lisäävässä eli additiivisessa järjestelmässä värit taas muodostuvat valosta. Jos valkoista valoa, kuten päivänvaloa, johdetaan kolmikulmaisen prisman läpi, se jakaantuu värispektriin eli kirjon väreihin. Sama toimii myös päinvastoin, jos värispektri kootaan takaisin yhteen, esimerkiksi linssin avulla, värispektri tulee muodostamaan jälleen valkoista valoa. Mustaa ja tummempia sävyjä saadaan vähentämällä valon kirkkautta tai suodattamalla värejä pois valosta. (Itten 1998, 15–16.) Näytöt muodostavat eri värejä kolmea pääväriä sisältävien

valopisteiden, pikseleiden, avulla ja tästä syystä digitaalisen grafiikan tuotannossa käytetään lisäävää järjestelmää. (Ahearn 2008.)

Additiiviseen järjestelmään perustuvan RGB-, eli puna-viher-sini-mallin lisäksi digitaalisen grafiikan tuotannoissa värien sekoittamiseen hyödynnetään HSB-mallia. Mallin nimi juontuu englannin kielen sanoista sävy (hue), kylläisyys (saturation) ja kirkkaus (brightness). Malleja hyödynnetään grafiikan tuotantohjelmien värivalinta työkaluissa eri värien sekoittamiseksi. Sävyllä määritetään värisävyn sekoitus. Tätä sekoitusta voidaan hienosäätää päävärien arvoilla RGB-mallin säätimien avulla, jos työkalu sisältää sellaiset. Kylläisyydellä määritetään värin sekoitusta valkoiseen väriin, jonka avulla värisävyä voidaan vaalentaa. Kirkkaudella taas säädetään värin sekoitusta mustaan väriin, jonka avulla värisävyä voidaan tummentaa. Kylläisyys ja kirkkaus toimivat yhdessä, ja sekoittamalla molempia yhteen saadaan harmahtavia värisävyjä. (Gahan 2008.)

Väriin sitoutuu paljon edeltä määrittyneitä merkityksiä ja vaikutelmia. Väreillä voidaan vaikuttaa esimerkiksi siihen mihin katsoja, tai pelaaja, kiinnittää ensimmäisenä huomionsa. Ihmissilmä yleensä kiinnittää huomiota suurimpaan nähtyyn kohteeseen. Tätä huomiota voidaan tosin ohjata väriä käyttämällä. Värikkäät kohteet korvaavat suurien kohteiden vievän huomion usein niiden koosta huolimatta. Tätä tekniikkaa hyödynnetään paljon pelisuunnittelussa. Kirkas kimallus maassa on esine, räjähtävä tynnyri on kirkkaan punainen, ja vihollisen heikko kohta hohtaa kirkkaasti. Tarkoituksena näille on herättää pelaajan huomio. Tämä myös tarkoittaa, että loput ympäristöstä tulisi olla huomion kohteisiin verrattuna hieman hillitympi kylläisyydessään ja kirkkaudessaan, jotta halutut kohteet nousevat huomion keskipisteeksi ja etteivät ne huku muiden kohteiden sekaan. Tosin päinvastainenkin on myös totta, muuten kirkkaassa ympäristössä tummat kohteet, kuten luolan suu tai äkkinäinen pudotus tuntemattomaan, voivat kiinnittää pelaajan huomion. (Gahan 2008.)

Sveitsiläisen taiteilija ja väriopin teoreetikko Johannes Itten saa maininnan myös Ahearnin (2008) teoksessa. Itten tunnetaan kylmä-lämmin-kontrastin kehittäjänä, ja näitä värejä ympäröivän symbolismin tutkijana. Kylmä-lämmin-

kontrastin avulla voidaan kuvailla värien symbolista lämpötilaa. Punaiset, oranssit ja keltaiset sävyt jakaantuvat lämpimiin väreihin, sillä niistä nousevat mielikuvat liittyvät tuleen, valoon ja lämpöön. Siniset, vihreät ja violetit värit taas jakautuvat kylmiin väreihin, sillä niistä nousevat mielikuvat liittyvät viileyteen, pimeyteen ja talveen. (Itten 1998, 45–48.)

Värien kylmä-lämmin-kontrasti on vain yksi tapa tulkita värien ilmaisevia merkityksiä. Yksittäisiin väreihin on liittynyt ajan saatossa erilaisia merkityksiä, oletuksia ja symbolismia. Esimerkiksi purppura on kuninkaallinen ja tumman sininen on surullinen väri. Edeltä määrittyneistä merkityksistä huolimatta värit ovat täysin kontekstisidonnaisia. Vesi yleensä esitetään sinisenä, mutta entä jos vesi onkin vihertävää? Mieleen saattaa nousta, että vesi on likaista tai myrkyllistä. Tällä tavoin pelissä voidaan värin avulla erotella esimerkiksi, mitkä veden lähteet ovat turvallisia uida läpi tai juoda. (Gahan 2008.)

Viimeisenä on perspektiivi. Perspektiivillä määritetään, mistä kulmasta grafiikkaa katsotaan, ja minne grafiikan katoamispisteet asettuvat ja kuinka ne vaikuttavat graafisten elementtien skaalaan. Esimerkiksi mitä lähempänä kohde on katoamispistettä, sitä pienempi kohde on. Perspektiivi voidaan toteuttaa yhdellä tai useammalla, tosin harvoin enemmän kuin kolmella, katoamispisteellä. Katoamispisteiden määrällä voidaan toteuttaa erilaisia efektejä. Yhdellä katoamispisteellä voidaan esimerkiksi luoda vaikutelma horisontista, kun taas kolmella katoamispisteellä voidaan luoda näkymään syvyyttä. (Gahan 2008.)

Viiden kuvataiteen aspektin lisäksi esille nousee kuvataiteelle hyvin tärkeä seikka, ympäristön rakenteellinen tutkiminen. Rikkomalla eri kohteet helpommin käsiteltäviin osiin ja tutkimalla kuinka eri pinnat ja muodot rakentuvat, on mahdollista hakea näkökulmaa ja inspiraatiota visuaalisten elementtien luontiin. Tämän prosessin aikana voidaan pohtia, kuinka kohde voitaisiin toteuttaa virtuaalisessa ympäristössä niin teknisesti kuin taiteellisesti. Havaintojen pohjalta voidaan laajentaa omaa käsitystä eri pintojen, materiaalien ja muotojen käyttäytymisestä luonnossa, ja harjoitella visuaalisten elementtien luontia konkreettisten harjoitusten kautta. (Gahan 2008.)

2.3 Kenttäsuunnittelulla rakennetaan vaikuttavia ympäristöjä

Kenttäsuunnittelun esteettisiin valintoihin voivat vaikuttaa arkkitehtuurista kumpuavat menetelmät ja periaatteet, samalla tavoin kuin grafiikan tuotantoon voivat vaikuttaa kuvataiteen perinteet, periaatteet ja menetelmät. Tavat, joilla ihmiset reagoivat eri tiloihin ja kuinka arkkitehti voi hyödyntää näitä seikkoja suunnitellessaan tiloja, ovat sovellettavissa myös kenttäsuunnittelijan tarpeisiin peliympäristöä suunniteltaessa. Näiden kahden eri alan myötäilevää suhdetta tarkastelee Christopher W. Tottenin (2014) teos *An Architectural Approach to Level Design*. Teoksessa tarkastellaan arkkitehdin työkalupakkia kenttäsuunnittelijan näkökulmasta, ja tarjotaan neuvoja, kuinka tätä arsenaalia voidaan hyödyntää kenttäsuunnittelussa parhaiten. (Totten 2014.)

Ympäristön valaisusta Totten käsittelee teoksen viidennessä luvussa, jonka pääteesi liittyy tunteikkaaseen kenttäsuunnittelun ja selviytymisvaistojen väliseen yhteyteen. Arkkitehtuurissa valo ei ole välttämättä ainoastaan tarpeenmukainen tilan ominaisuus. Valaistusta voidaan hyödyntää myös tosielämän arkkitehtuurissa johdattelevana tai tilan tunnelmaa säätävänä voimana. Valonlähteiden ympäristöön muodostamalla kuvioilla voidaan johdatella tilassa olijaa haluttuun suuntaan. Samoja analogioita voidaan hyödyntää myös peliympäristöissä. (Totten 2014.)

2.4 Kenttäsuunnittelun periaatteet

Michael Salmond (2021) käsittelee teoksessaan *Video game level design: How to create videogames with emotion, interaction and engagement* videopelien kenttäsuunnittelua niin osana pelinkehitys prosessia kuin omana taiteenalanaan, ja tutkii pelien kenttäsuunnittelussa käytettyjä menetelmiä ja työkaluja. Teoksen tavoitteena on täydentää alaa käsittelevää kirjallisuutta, jossa kenttäsuunnittelu on jäänyt muiden pelikehityksen sisältämien aiheiden alle. (Salmond 2021.)

Kenttäsuunnittelussa yhdistyvät taiteet ja teknologia muodostamaan elävän ympäristön tutkia, taistella vihollisia tai etsiä salaisuuksia. Kenttäsuunnittelu on yksi pelisuunnittelun sovelletuista muodoista. Se jakaa elementtejä muiden osaluokkien kanssa. Kenttiä suunniteltaessa pelin rakentavat elementit liittyvät yhteen. Pelimekaniikasta grafiikkaan ja äänimaailmasta narratiiviin, kenttäsuunnittelijan tehtävänä on kutoa nämä elementit yhteen mukaansatempaavaan ja viihdyttävään muotoon. (Salmond 2021, VIII–XIII.)

Kenttäsuunnittelu aloitetaan jo pelinkehityksen varhaisessa vaiheessa. Suunnittelua varten kenttäsuunnittelijat työskentelevät peli- ja graafistensuunnittelijoiden kanssa projektia alustavan tutkimuksen kanssa. Suunnittelijat tutkivat ja kokoavat yhteen inspiraatiota viitekuvien ja esimerkkien muodossa. Näiden esimerkkien muodostama viitekirjasto ei ainoastaan anna suuntaa projektin visuaaliseen ilmeeseen vaan sen avulla voidaan määrittää myös konkreettisempia ominaisuuksia kuten pelimaailman skaalaa tai kohtauksen kompositiota. (Salmond 2021, 1–10.)

Kun pelimaailmalle on löydetty suuntaa ja pelin mekaniikat hahmoteltuina, kenttäsuunnittelijat voivat siirtyä tutustumaan projektissa käytettävään pelimoottoriin tai kenttäeditoreihin. Kenttäsuunnittelun työnkulku riippuu suuresti, siitä millaisia työkaluja suunnittelijoilla on käytettävissä. Esimerkiksi pelimoottorit, jotka ovat pääosin suunniteltu 3D-ympäristöjä varten tulee vaatimaan erilaisen työnkulun kuin 2D-ympäristöä varten suunnitellut. Kyseessä on myös kokemus kysymys. Työnkulku määrittyy paljolti siitä, kuinka kokeneita kenttäsuunnittelijat ovat käyttämään annettuja työkaluja ja joutuvatko he oppia käyttämään uusia ohjelmistoja. (Salmond 2021, 6–9.)

Tutkimuksen ja työkaluihin perehtymisen jälkeen varsinainen kenttäsuunnittelu voidaan aloittaa luomalla yksinkertaisia prototyyppisiä mahdollisista kentistä. Tätä kutsutaan yleensä nimellä gray boxing eli harmaalaatikointi, joka johtaa nimensä tuotantomenetelmän hyödyntävien yksinkertaisista, teksturoimattomista geometrisistä muodoista, joiden avulla alustavaa kentän muotoa aloitetaan hahmottamaan. Harmaalaatikoimalla voidaan testata pelin

suunniteltua skaalaa ja tehdä jo alkuvaiheessa tarvittavat hienosäädöt varsinaisessa pelimoottorissa tai -editorissa ennen ensimmäisien tekstuurien ja muiden visuaalisten ja pelimekaanisten elementtien luontia. (Salmond 2021, 8–9.)

Esituotannon aikana laaditut tuotantosuunnitelmat ja ohjeistukset määrittävät, millaista pelimaailmaa kenttäsuunnittelijat alkavat rakentamaan. Muutokset tuotannon aikana voivat aiheuttaa laajoja viivästyksiä, joten huolellinen suunnittelu esituotannossa voi säästää paljon päänvaivaa tuotannon aikana. Tuotantovaiheeseen siirtyessä kenttäsuunnittelijoiden työksi muodostuu suunnitteluvaiheessa syntyneiden prototyyppien ja konseptien jalostaminen toimivaksi pelimaailmaksi, sekä muiden pelikehitystiimin tuottajien luomien elementtien sijoittaminen pelimaailmaan. (Salmond 2021, 10.)

Salmond (2021) esittelee teoksessa kymmenen tuotantovaiheen kenttäsuunnittelussa käytettävää tuotantomenetelmää. Niin kuin muissa taidetuotannon aloissa, kenttäsuunnitteluun ei sisälly täysin kiveen hakattuja, absoluuttisia sääntöjä, vaan suuntaa antavia periaatteita ja menetelmiä, joiden avulla teoksen tekijät voivat parantaa teoksensa laatua, ja paremmin saavuttaa ja viestiä teoksen tavoitteet. Näiden kymmenen periaatteen tarkoituksena ovat siis antaa suuntaa kenttäsuunnittelijoille siihen, kuinka luoda pelimaailma, joka saman aikaisesti saavuttaa pelin mekaaniset ja taiteelliset aspektit, ja kykenee viestimään pelin narratiivia, mekaniikkaa ja audiovisuaalista suuntaa. Valaistuksen toteuttamiseen keskeisimmät periaatteet ovat ympäristön graafiseen ilmeeseen ja sen vaikutuksiin liittyvät periaatteet, kuten myös pelimekaniikan vaikutus niin ympäristön esteettisiin valintoihin kuin kenttäsuunnittelun ottamaan suuntaan. (Salmond 2021, 18–19.)

Pelikenttien esteettisillä valinnoilla voidaan vaikuttaa ympäristön tunnelmaan ja herättää tunteita pelaajassa. Värivalinnat, pintojen materiaalit, sekä tilan koko ja muoto ovat vaikuttavia tekijöitä, joita kenttäsuunnittelija voi vaikuttaa ympäristön välittämään tunnelmaan. Onnistuneet esteettiset valinnat ympäristön luonnissa kykenevät välittämään tahdotun viestin pelaajalle, olipa kyseessä ympäristön tunnelma tai haluttujen tunteiden herättäminen. (Salmond 2021, 30–31.)

Tunteiden herättämisen ja tunnelma ylläpitämisen lisäksi esteettisten valintojen tulisi myötäillä pelin mekaanisia tarpeita. Esteettisillä valinnoilla voidaan viestiä, mitä pelin mekaniikkaa pelaaja voi käyttää edetäkseen. Esimerkiksi kiivettävän seinämän tulisi erota muusta ympäristöstä, jos pelissä on kiipeämismekaniikka. Esteettiset valinnat ovat tältä kannalta onnistuneita silloin, kun peliympäristön navigointi on hauskaa, ja ympäristöön sidotut viestit ovat yhtenäisiä ja loogisia pelin kontekstissa. (Salmond 2021, 19–21.)

Samoin kuin muut osat pelikenttien estetiikkaa, myös valaistuksella voidaan viestiä niin halutusta ympäristön tunnelmasta kuin myös konkreettisimmista pelimekaanisista seikoista. Pelien valaistus noudattaa usein todellisen maailman analogeja esteettisessä mielessä. Esimerkiksi punaisella valaistuksella voidaan viestiä uhkaavaa tunnelmaa tai estetettyä kulkusuuntaa. (Salmond 2021, 65.)

Peleissä ympäristöä hyödyntävä tarinankerronta käyttää hyväkseen peliä varten luotuja elementtejä ympäristössä, kuten tekstuureja, malleja tai muita maailmaan sijoitettavia kohteita, laajentaakseen pelin narratiivia. Jo yleinen miljöö tai arkkitehtuurinen suunta osana pelin maailmaa ovat tehokkaita laajentamaan pelin narratiivia. Esimerkiksi maailmanlopun jälkeisen maailman ottama muoto voi vihjata millainen entinen maailma oli ja mitä sille tapahtui kertomatta siitä sanoin tai tekstin kautta suoraan pelaajalle tai näyttämällä välikohtauksia tapahtumasta. (Salmond 2021, 20–21)

Ympäristön esteettinen suunta ja hyödyntäminen narratiivisena työkaluna auttaa myös pelaajan syventymistä pelinmaailmaa. Syventyessään peliin pelaaja tuntee olonsa osaksi pelimaailmaa ja kykenee sen herättämien tunteiden perusteella melkein kuin aistimaan pelin lähettämää tunnelmaa. Tunteita herättämällä voidaan esimerkiksi pelaajalle viestiä uhista kulman takana luomalla alistavaa tai pelottavaa tunnelmaa tiukoilla ja klaustrofobisilla käytävillä, tummilla ja vähemmän kylläisillä sävyillä, ja himmeällä valaistuksella. Samalla tavoin avarampi ympäristö voi luoda ihmetyksen ja mielenkiinnon tunteita, ja rohkaista täten pelaajaa tutkia ympäristöään lisää, etenkin, jos

valaistusta ja graafisia elementtejä hyödynnetään kiinnostavalla tavalla. (Salmond 2021, 30–32)

Pelimaailmaa suunnitellessa pelin mekaniikka tulisi olla ajavana voimana estetiikan ja narratiivin rinnalla. Pelien kentät voidaan tulkita pelimekaniikkojen sommitelmina. Kentän ottama visuaalinen muoto on myös paljolti pelin sisältämän mekaniikan määrittämä. Esimerkiksi, jos pelaajalle annetaan vapaus kulkea minne tahansa kentässä, pelkkien esisivujen käyttäminen rakennuksissa ei ole suositeltavaa, kun taas jos kenttä on lineaarinen ja pelaaja näkee katseltavan kohteen ainoastaan tietyistä kulmista, esimerkiksi rakennuksen esisivu voi olla riittävä. (Salmond 2021, 32)

Valaistus kantaa mekaniikan osalta usein navigointia avustavia tehtäviä. Ne voidaan jakaa neljään konkreettiseen käyttökohteeseen kenttäsuunnittelun näkökulmasta. Valaistuksella pelaajalle voidaan ilmaista kentän sisään- ja uloskäynnit, valaista suoria tai haluttu reitti kentän läpi, paljastaa salaisuuksia, tai kiinnittää pelaajan huomio halutulla tavalla. Onnistuneen kenttäsuunnittelun kannalta näiden valaisuelementtien tulisi sulautua kenttään ja tuntua luonnollisilta pelaajalle kentän kontekstissa. Kenttäsuunnittelijan kannattaa harkita ympäristön valaistusta myös pelikokemuksen ja helppokäyttöisyyden kannalta. Esimerkiksi tärkeät tai interaktiiviset kohteet voidaan valaista huomiota herättävällä tavalla, jotta ne ovat helpommin erotettavissa muusta ympäristöstä tai ei-interaktiivisista kohteista. (Salmond 2021, 65.)

2.5 Godot -pelimoottori

Pelikehityksen aikana projektin jäseniltä suunnittelijoista ohjelmoijiin vaaditaan laajasti alan teknistä tietoa ja -taitoa. Monissa tilanteissa on parempi olla keksimättä pyörää uudelleen ja sen sijasta käyttää jo valmiiksi olemassa olevia raameja ja kokonaisuuksia kuten koodikirjastoja. Tällaisista kokonaisuuksista kattavimmat ovat pelimoottorit. Pelimoottorilla tarkoitetaan ohjelmistokokonaisuutta, joka tarjoaa pelinkehittäjille sarjan työkaluja pelin rakentamiseen. Pelimoottorit yleensä sisältävät yhtenäisen käyttöliittymän

työnkulun optimoimiseksi, valmiit järjestelmät renderöintiin ja fysiikkojen simulointiin niin kaksi- kuin kolmiulotteisessa peliympäristössä, ja tuen yhteen tai useampaan ohjelmointikieleen, virheenkorjaukseen ja verkko-ominaisuuksiin. Usein pelimoottorit sisältävät myös ominaisuuden tuoda muissa sovelluksissa tuotettuja elementtejä, kuten 3D-malleja, animaatioita tai muita audiovisuaalisia elementtejä, suoraan ohjelmaan, ja julkaista valmis projekti usealla eri alustalla, kuten konsoleille, tietokoneille ja mobiiliin. Pelimoottoria käyttämällä pelinkehittäjät voivat keskittyä itse projektiin ja säästää vaivaa projektia alustavien järjestelmien rakentamisessa. (Bradfield 2023.)

Godot 4 (2023) on avoimen lähdekoodin pelimoottori. Ohjelmisto tarjoaa laajan valikoiman yleisiä työkaluja ja useiden suosittujen alustojen välisen tuen niin kaksi- kuin kolmiulotteisille peliprojekteille. Avoimen lähdekoodin pelimoottorina, ohjelmisto on täysin ilmainen, ja kehittäjäyhteisön kehittämä ja ylläpitämä. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

Godot-pelimoottorissa peli kootaan käyttäen ohjelmaan valmiiksi integroituja tai käyttäjän itse ohjelmoimia Node-moduuleja ja koodia. Pelimoottorin valmiilla moduuleilla voidaan suorittaa pelinkehityksessä käytettäviä yleisiä funktioita ilman ohjelmointia. Moduulit jakavat keskenään ominaisuuden nimeämistä varten, moduulille ominaisia ominaisuuksia, joita voidaan muokata Inspector-ikkunassa ja jatkaa tarvittaessa ohjelmoimalla lisäominaisuuksia itse, kyvyn kutsua moduulia päivittymään joka ruutupäivityksen aikana, ja kyvyn sisältää yhden tai useamman muun alimoduulin hierarkiassa. Moduuleilla voidaan täten muodostaa hierakkisia rakenteita pelielementtejä luodessa. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

Ympäristön ja sen valaistuksen luontia varten Godot 4 tarjoaa käytettäväksi pelimoottoriin valmiiksi ohjelmoituja moduuleja, joiden avulla kehittäjät voivat toteuttaa haluamansa peliympäristön haluamallaan grafiikoilla. Kaksiulotteisen peliympäristön luomiseen keskeisiä moduuleja ovat Sprite2D-, AnimatedSprite2D-, Tilemap- ja elementtien järjestystä ohjaavat moduulit. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

Pelimoottorin kaksi Sprite-moduulia renderöivät valitun, tyypillisesti bittikartta, grafiikan ruudulle. Eroa näiden kahden moduulin välillä on millaista grafiikkaa ne renderöivät. Nimensä mukaisesti AnimatedSprite2D-moduuli on suunniteltu animoidun grafiikan renderöintiin, kun taas Sprite2D-moduuli on monipuolisempi käyttökohteidensa suhteen, mutta yleisimmin sitä käytetään ei-animoitujen elementtien renderöintiin. Tilemap-moduulin avulla voidaan moduulin luomalle ristikolle sijoittaa niin renderöitäviä visuaalisia elementtejä kuin kokonaisia pelimekaanisia kokonaisuuksia näkymien muodossa. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

Jokainen Node-moduuli kaksiulotteisessa peliympäristössä sisältää itseensä aina vähintään kolme eri ominaisuuskategoriaa. Nämä ominaisuudet määrittävät moduulin funktion asetukset, kuinka sen ominaisuudet periytyvät alimoduuleihin, metadatan kuten editorissa näkyvä kuvaus, moduulin muuntumista näkymässä, ja moduulin renderöinti ja fysiikka prioriteetteja muihin näkymän moduuleihin verrattuna. Jos renderöitäviin ja pelimekaanisiin elementteihin tarvitaan lisäfunctioita, voidaan niitä lisätä luomalla alimoduuleja. Myös elementtien välistä interaktiota ja näkymän muotoilua voidaan kontrolloida alimoduuleja käyttämällä. Esimerkiksi, jos moduuli sisältää näkymässä renderöitävää tekstiä, teksti voidaan muotoilla käyttämällä alimoduulia. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

Kaksiulotteisen peliympäristön valaistusta varten Godot sisältää PointLight2D-, DirectionalLight2D-, LightOccluder2D- ja CanvasModulate-moduulit. CanvasModulate-moduulin voidaan hallita näkymän yleistä sävyä ja kirkkautta, ja sillä voidaan tarvittaessa himmentää värien kirkkautta. Varsinaiset valaisevat moduulit ovat PointLight2D- ja DirectionalLight2D-moduulit, joista PointLight2D-moduuli valaisee alueen pisteen ympärillä säädettävällä alalla, ja DirectionalLight2D-moduuli valaisee koko näkymän määritetystä suunnasta. LightOccluder2D-moduuli taas toimii valoa tukkivana esteenä, jonka muoto voidaan määrittää moduulin asetuksissa halutuksi. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

3 Peliympäristön valaistuksen menetelmät

3.1 Valaistuksen merkitys peleissä

Visuaalisen median lajista huolimatta, valaistuksella on suuri merkitys siihen, miltä valmis teos näyttää. ”Ilman valaistusta jäljelle jää vain musta ruutu”, oli foorumikeskustelun vastaus, joka sai videokuvaaja ja kirjailija John Jackmanin kiinnostumaan valaistuksesta visuaalisessa mediassa. Teoksessa *Lighting for Digital Video and Television, 3rd Edition* Jackman avaa näitä valaistuksen periaatteita ja merkitystä visuaalisessa mediassa. (Jackman 2012.)

Valaistuksella esiintyy peleissä samanaikaisesti niin esteettinen kuin pelimekaaninen merkitys. Valaistuksella voidaan vahvistaa pelin miljöön tunnelmaa tai korostaa grafiikan värimaailmaa, ja samaan aikaan ohjata pelaajan huomiota haluttuun suuntaan tai auttaa pelaajaa piiloutumaan varjoihin. (Salmond 2021, 65). Esimerkiksi *Thief: The Dark Project* (1998) hyödyntää valaistusta edellä mainituin tavoin. Pelin keskiaikaistyylisen steampunk-vivahteisen miljöön tunnelmaa vahvistaa hämärit kivikujat ja kaasuvalojen himmeästi valaisemat kadut. Samalla valaistus pelaa suuressa roolissa pelin päämekaniikan, hiiviskelyn, kanssa. Pelaaja voi piiloutua kaupungin varjoihin, hyökätä varjoista vihollisvartijoiden kimppuun muiden vihollisten näkemättä ja sitten piilottaa tyrmätyn vihollisen piiloon varjoihin.

Valaistuksella on merkitys myös teknologian ja pelien historian kannalta. Erilaisten teknologioiden ja laitteistojen kehittyessä edistyvät myös pelien hyödyntämät tekniikat, grafiikan tekniikat mukaan lukien. Valaistuksen osalta yksi suurimmista harppauksista lienee id Softwaren *Quake II* (1997), joka on ensimmäinen tietokoneelle ja konsoleille kehitetty peli, joka hyödyntää reaaliaikaista dynaamista valaistusta (Innovecsgames 2023). Tällä tarkoitetaan valaistusta, jonka lähteen ominaisuuksia voidaan muuttaa lennossa ajon aikana, ja joka kykenee efekteihin kuten värin vaihtamiseen ja liikkeeseen reagointiin (Pixcap 2024). Id Softwaren *Doom 3* (2004) taas on yksi ensimmäisistä peleistä hyödyntämään dynaamista varjostusta (Kosak 2004, 2).

3.2 Valaistuksen elementit

Eri materiaalien reaktioon valaistuksen muutokseen voidaan vaikuttaa erilaisten tekstuurien avulla. Näihin tekstuureihin kuuluvat kiilto- (specularmap), normaali- (normalmap) ja korkeuskartat (heightmap). Nämä tekstuurit määrittävät ominaisuutensa per pikseli eli ne liittävät muutoksen pintamateriaalin tekstuuriin määritettyjen asetusten perusteella. (Puhakka 2008, 231–250)

Kiiltokartalla voidaan määrittää pintamateriaalin kiiltävyyttä. Niiden luomat pinnat toimivat samalla tavoin kuin esimerkiksi monet metallit tai lasi reagoivat valoon todellisessa elämässä. Kartalle sijoitettavat alueet määrittävät, kuinka kiiltäviä tekstuurin alueet ovat, mustat alueet ovat matta- ja valkoiset kiiltäviä-alueita. (Puhakka 2008, 237–239.)

Korkeuskartoilla voidaan määrittää syvyyttä pintamateriaalille. Esimerkiksi tiiliseinän tekstuuriin voidaan määrittää korkeuseroa tiilen ja laastisauman välille. Korkeuskartat muodostetaan samalla tavoin kuin kiiltokartat, mustat alueet edustavat syvää ja valkoiset alueet korkeaa aluetta. (Puhakka 2008, 248–258.)

Normaalikartalla pintamateriaalille voidaan määrittää hienovaraisia valoa taittavia yksityiskohtia, esimerkiksi kuhmuja, säröjä tai viiltoja. Tekstuurin avulla pelimoottori määrittää, mihin suuntaan ja missä kulmassa simuloitu valo heijastuu pelaajan näkökulmaan. Normaalikartat muodostuvat hieman eri tavoin kuin spekulaaari- tai korkeuskartat, sillä niissä hyödynnetään useita värisävyjä heijastuvan valon suunnan määrittämiseksi. (Puhakka 2008, 250–251.)

3.3 Valaistuksen rakentamisen työkalut

Valaistukseen vaikuttavien tekstuurien tuottamiseen käytetään niin pelituotantoon erikoistuneita kuin yleisesti grafiikan tuotannossa hyödynnettäviä ohjelmistoja. Tekstuurin rasteripinnan luontiin käytetään yleensä piirto- tai

kuvamanipulaatio ohjelmia, kuten Adoben tarjoamat Photoshop (2025) ja Illustrator (2025) tai ilmaiset ohjelmistot GIMP (2025) ja Krita (2025). Lisätekstuurien luontiin manuaalisesti voidaan myös hyödyntää piirto- tai kuvamanipulaatio-ohjelmia, mutta niiden luontiin löytyy myös erikoistuneita ohjelmia, joiden avulla halutut lisätekstuurit voidaan generoida automaattisesti. Näitä ovat esimerkiksi maksulliset Adobe Substance 3D Painter (2025) ja Zbrush (2022), tai ilmainen Laigter (2023).

Graafiset elementit eivät itsenään riitä toteuttamaan ympäristön valaistusta, vaan ne tarvitsevat simuloitua valoa heijastettavaksi. Simuloidun valon tuottamiseksi tarvitaan pelimoottorin tarjoamia ominaisuuksia. Godot-pelimoottorissa voidaan luoda valopisteitä ja suunnattuja valoja. Pelimoottorin tavallinen valaistus toimii kolmiulotteisessa peliympäristössä, mutta pelimoottorista löytyy myös kaksiulotteiseen pelitilaan suunniteltuja valoja. Kaksiulotteisten valojen lisäksi pelimoottorilla voidaan määrittää, mitkä elementit estävät valon pääsyä läpi käyttämällä LightOccluder2D-moduulia. (Linietsky, Manzur & the Godot community -yhteisö 2023.)

4 Peliympäristön valaistuksen toteutus

4.1 Tuotanto alkaa suunnitelmasta

Peliprojektin tuotantoprosessi alkaa esituotannossa, jonka aikana voidaan laatia pelituotannon suunnitelma ja käsikirjoitus, jotka yhdessä muodostavat dokumentin, jota kutsutaan GDD:ksi (Game Design Document). Dokumentin tarkoituksena on olla mahdollisimman kattava kokoelma suunnitelmiksi työstettyjä ideoita ja kuvata pelituotannon keskeiset tavoitteet. GDD sisältää konsepteja pelimekaniikoille, ohjeet pelin audio-visuaaliseen suunnitteluun, konseptitaidetta, ja pelin tarinan.

Tämän opinnäytetyön toiminnallisen osan demoa varten kirjoitettuun pelituotantosuunnitelmaan (Liite 1) on ensin selvitetty projektin keskeiset tavoitteet. Projektin tavoitteeksi on määritelty hyödyntää Godot -pelimoottorin

ominaisuuksia kaksiulotteisen peliympäristön valaistuksen luonnissa. Dokumentissa käsitellään projektin pelimekaniikkaa, tarkennetaan projektin tavoitteita ja listataan ideoita tavoitteiden saavuttamiseksi sekä määritellään demon genreä. Listatuista ideoista on demoon valittu tasohyppelygenre.

Dokumentin viimeisessä kappaleessa pohditaan graafisten elementtien tuottamisen tavoitteita ja ideoita. Tavoitteissa määritetään grafiikan tavoitteeksi olla immersiivisenä voimana valaistuksen toteutuksessa. Tavoitteissa määritetään myös luotavat grafiikan elementit ja niiden tuottamiseen käytettävät ohjelmat. Dokumentissa luetellaan visuaalisen ulkoasun genrejä, miljöitä ja ideoita elementtien tuotantoon. Näistä ideoista projektin tyyliin valittiin urbaanifantasia ja luolasto.

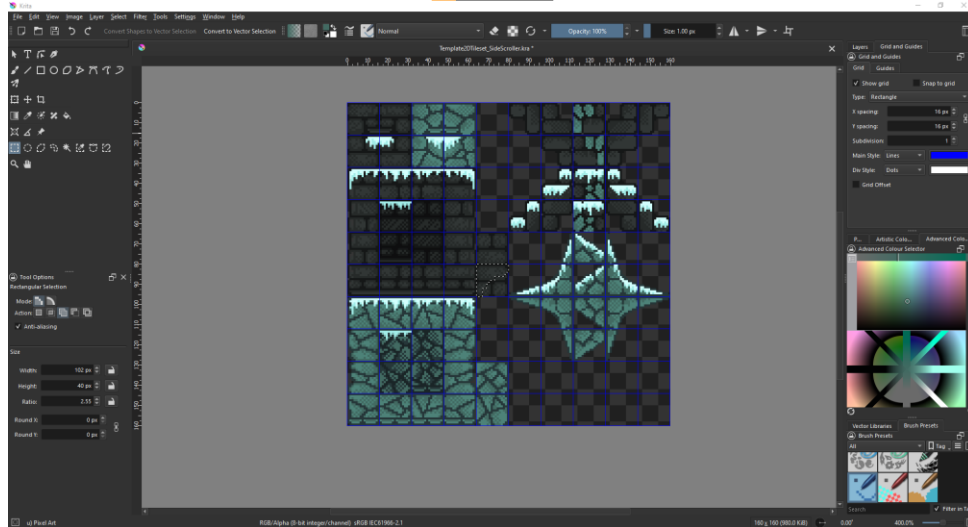
4.2 Ympäristön grafiikan tuottaminen

Grafiikan tuotanto projektissa alkoi päättämällä mihin muotoon graafiset elementit luodaan. Ympäristön grafiikkaa varten luodaan tekstuurien kokoelma eli tekstuuriatlas. Spritesheet-tiedostojen tavoin tekstuuriatlakset jakautuvat myös ruudukoksi. Jokainen ruutu on samankokoinen, mutta niiden sisältämät tekstuurit voivat jatkua ruutujen välillä.

Projektia varten ruutujen kooksi valittiin 16x16 pikseliä. Tällä koolla luodut tekstuurit ovat hyvin pienikokoisia ja tyyllisesti pikselitaidetta. Tämä tyyli on projektissa tarkkaan harkittu, sillä sen avulla voidaan demonstroida per pikseli valaistusta. Koska kentän elementit sijoittuvat eri tasoille, tekstuurit tallennetaan erillisiin tekstuuriatlaksiin. Ympäristön geometria ja taustan tekstuurit ovat erillisissä tiedostoissaan, jotka ovat jakaantuvat 16x16 ruutuihin. Projektiin luodut objektit tallennetaan erilliseen atlaksiin samalla tavoin, mutta objektit ovat 32x32 pikselin kokoisia.

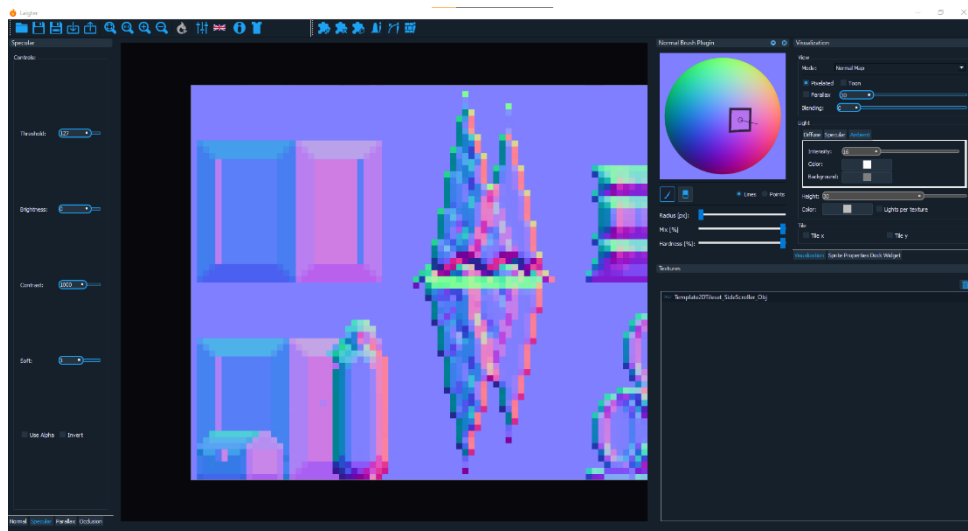
Projektin diffuusitekstuurit luotiin Krita-piirrosohjelmalla. Ohjelman tarjoamat työkalut sopivat projektin tarpeisiin, kuten valmiit siveltimet pikseligrafiikan tuotantoon, säädettävä ruudukko ja tarvittavat työkalut myös normaalikarttojen manuaaliseen luontiin. Ympäristön tekstuureja varten projektissa hyödynnettiin

mallia tarvittavista ympäristön muodoista, joka osoittaa mihin suunnitellut törmäyspinnat asettuvat.



Kuva 1. Ympäristön geometrian diffuusitekstuurit Krita-ohjelmassa.

Projektia varten valittiin luolastomiljö. Perusteluna valinnalle on ympäristön luonnollinen pimeys, jossa pistevaloilla voidaan leikitellä helpommin. Lisäksi ympäristön taustan luominen helpottui suljetussa tilassa, sillä tausta pysyy staattisena ja pääasiassa samassa syvyydessä, joka helpottaa tekstuurien luontiprosessia. Demon grafiikkaan luotiin myös kiilto- ja normaalikartat. Niiden luomiseen demoon projektissa hyödynnetään Laigter-ohjelmaa. Ohjelma luo kiilto- ja normaalikarttojen lisäksi myös parallaksi- ja okluusiokarttoja automaattisesti.



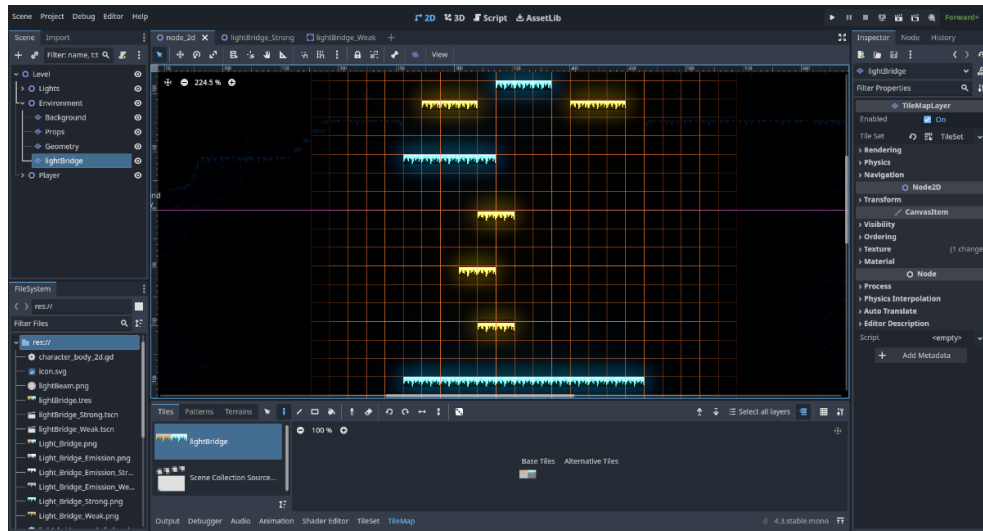
Kuva 2. Taustaobjektien normaalikartta Laigter-ohjelmassa.

Ohjelman tulkinta syötetystä tekstuurista ei välillä tuota haluttua tulosta, joten tarvittaessa tekstuureja voidaan muokata myös manuaalisesti ohjelman sisällä. Laigter -ohjelman avulla voidaan myös esikatsella tuotettuja tekstuureja reaaliaikaista valoa käyttämällä. Ohjelman tuottamat tekstuurit viedään kuvien muodossa erillisinä kuvatiedostoina.

4.3 Demon koostaminen pelimoottoriin

Demon koostaminen Godot-pelimoottoriin alkaa luomalla näkymä eli scene. Godot-pelimoottorissa näkymät voivat sisältää niin valmiiksi koottuja pelielementtejä kuin kokonaisia peliympäristöjä. Projektiin lisätyistä näkymistä on valittava yksi päänäkymä, johon yleensä varsinainen projektin pää rakenne kootaan. Näkymät voivat sisältää näkymiä lisäämällä niitä toisiinsa objekteina. Demossa päänäkymän lisäksi lisänäkymiä hyödynnettiin pelimekaanisia elementtejä toteuttaessa.

Demon valosillat (Kuva 3), niin pelaajan kosketusta kestävä vahvat kuin pelaajan kosketuksesta hajoavat heikot sillat, toteutettiin käyttämällä lisänäkymiä. Näkymien avulla niistä kyettiin tekemään toistettavia elementtejä, joita voidaan sijoittaa ruudukkomoduuleihin ilman lisätyötä tai demon rakenteen liikaa monimutkaista. Lisäksi elementtien hienosäätö helpottui, sillä kun muutoksia tehdään näkymään perustuvaan objektiin, muutokset vaikuttavat jokaiseen objektin instanssiin demossa.



Kuva 3. Demon valosilta elementit Godot-pelimoottorin editorissa.

Demon kokoaminen jatkuu suunnittelemalla perusrakenne päänäkymään objekteja varten. Godot käyttää tasohierarkiaa, joka määrittää kuinka visuaaliset objektit järjestyvät näytölle ja kuinka ne reagoivat toisiinsa. Oletuksena, mitä alempana hierarkiassa elementti on, sitä ylempänä se piirtyy näytölle. Jokainen objekti voi myös sisältää alaobjekteja. Tässä objektien välisessä hierarkiassa ylempi objekti määrittää tietyt ominaisuudet, kuten sijainnin tai kierron, sen sisältäville objekteille. Tämän hierarkian avulla voidaan järjestellä elementit omiin kategorioihinsa käyttämällä tyhjiä objekteja. Nämä objektit eivät sisällä muita funktiota kuin objektien perusfunktiot, kuten muunnos-, sijainti- ja metadatan.

Kun perusrakenne oli koostettu, demon rakenteeseen lisättiin alaobjekteja, jotka itsessään sisältävät moduuleja. Moduulien avulla voidaan hyödyntää Godot-pelimoottorin sisäänrakennettuja ominaisuuksia, kuten ruudukkoja ja dynaamisen valaistuksen ominaisuuksia. Ruudukkoja hyödyntämällä voidaan koota ympäristö modulaarisesti. Salmond (2021) mainitsee modulaarisuuden olevan edullista kenttäsuunnittelussa ympäristön muokattavuuden, suorituskyvyn, yhdenmukaisuuden ja pelaajien luoman sisällön osalta (Salmond 2021, 110–111). Demon osalta ympäristön modulaarisuuden helpottama muokattavuus on tärkeä ominaisuus. Ruudukko tosin voi jättää oman jälkensä lopputulokseen tekemällä ympäristöstä kulmikkaan, mutta tätä vaikutusta

voidaan välttää lisäämällä ympäristön elementtejä sisältävään tekstuuriatlakseseen kulmia pyöristäviä muotoja.

Ympäristön eri elementit jaetaan demossa eri tasoille. Tasot rakennetaan lisäämällä ruudukkomoduulilla varustettu objekti ympäristön kokoamiseen varatun objektin alle. Tasojen järjestys editorin ikkunassa määrittää, missä järjestyksessä tasot piirtyvät näytölle. Demossa käytettävä ympäristön geometria ja pelimekaaniset elementit piirretään päällimmäisinä, ja taustalle sijoitettavat elementit, kuten taustatekstuurit ja ympäristön koristeet, piirretään alempana.

Ruudukolle sijoitettavien elementtien tekstuurit määritetään laattasetteihin eli tileset-tiedostoihin. Laattasetit toimivat kuin palettina, josta halutut tekstuurit voidaan sijoittaa ruudukolle. Godot-pelimoottorin laattasetit voivat hyödyntää diffuusitekstuuriin lisäksi normaali- ja kiiltokarttoja, jos laattasetin atlaksen tekstuuriksi määritetään canvas-tekstuuri. Tekstuurin lisäksi laattasetteihin voidaan määrittää muutakin informaatiota, kuten törmäysalueiden määrittäykset ja piirtojärjestyksen. Demossa tätä ominaisuutta hyödynnettiin ympäristön geometriassa määrittämään törmäysalueet, jotka määrittävät missä pelaajan hahmo koskettaa tekstuurin pintaa.

Dynaamisen valaistuksen saavuttamiseksi demoon tarvitaan valaistuista ja väritystä ohjaavia moduuleja. Nämä moduulit asetetaan omaan kategoriaansa. Kategorian kolme moduulia ovat DirectionalLight2D-, CanvasModulate- ja WorldEnvironment -moduulit. DirectionalLight2D-moduuli on määritetyn alueen kattava suunnattu valo. Se simuloi päivänpaisteen kaltaista koko ympäristöä kattavaa valokohdetta. CanvasModulate-moduulilla hallitaan, kuinka näytölle piirretyt kangasobjektit värittyvät. Tällä moduulilla voidaan hallita myös värien kirkkautta ja kylläisyyttä, ja luoda täten himmeämmän valaistuksen vaikutelman. WorldEnvironment-moduuli säätelee useita peliympäristön asetuksia. Moduulia tarvitaan silta elementin hohtoeftien tuottamiseksi.

5 Toteutuksen tulokset

5.1 Demon pohjatutkimuksen tulokset

Demon tuotantoa varten tarvittiin tausta-aineistoa kuvaamaan valaistuksen merkitystä niin peleissä yleensä kuin myös pelien suunnittelun, etenkin kenttäsuunnittelun kannalta. Aiheen tutkinnan pohjalta valaistuksen merkityksen voisi tulkita kasvaneen teknologian kehittyessä ainakin esteettiseltä kannalta. Valaistus niin peleissä kuin muissakin digitaalisessa, tietokoneella tuotetussa grafiikassa seuraa yhä samoja sääntöjä ja merkityksiä kuin muukin kuvataide.

Erilaisten tekniikoiden ja teknologioiden kehittyessä yhä sivistyneemmät valaistusmenetelmät ovat tulleet mahdollisiksi ja toteuttamiskelpoisiksi. Pelattuani vanhempia 1990-luvun loppupuolen ja 2000-luvun puolivälissä julkaistuja pelejä olen huomannut, että erilaisten valaistuksien toteutukseen hyödynnettiin monenlaisia menetelmiä, usein myös monia menetelmiä yhdistellen. Monet näiden kaltaisista valon renderöintitavoista olen myös huomannut nykyaikaisissa pelimoottoreissa olevan usein helposti lisättävissä näpinpainalluksella.

Nämä pohjatutkimuksen tulokset olivat hyödyllisiä ja auttoivat demon kokoamista. Ne antoivat suuntaa demoon tarvittaville elementeille ja sille, kuinka hyödyntää näitä elementtejä valmiissa tuloksessa. Taustatutkimuksen ansiosta oli helpompaa päättää, mitä ja millaisia elementtejä, kuten tekstuureja, demossa tulisi olla haluttujen ominaisuuksien demonstraatiota varten. Taustatutkimus auttoi myös pelimekaniikan muotoutumista antamalla suuntaa sille, kuinka haluttuja ominaisuuksia voidaan demonstroida. Demon kokoamista helpotti myös taustatutkimus Godot-pelimoottoriin ja sen tarjoamiin ominaisuuksiin, joka nopeutti uuden ohjelman käyttämisen oppimista. Yleisellä tasolla uuden ohjelmiston käyttöön ottaminen helpottui tutkimalla ohjelmistoihin liittyviä ohjeita ja tietokantoja, ja suunnittelemalla kuinka hyödyntää ohjelmien tarjoamia ominaisuuksia.

5.2 Tuotantomenetelmien hyödyt

Demon kokoamisprosessissa oli havaittavissa se, että käytetyillä ominaisuuksilla ja tekniikoilla voitiin lisätä syvyyttä valaistuihin kohteisiin suhteellisen pienellä lisätyöllä. Godot-pelimoottorin tarjoamat ominaisuudet ovat sisäänrakennettuja, ja eivät vaadi ylimääräistä ohjelmointia toimiakseen. Demon kokoamisessa käytetyt Laigter ja muut samankaltaiset materiaalien luonti ohjelmat voivat myös helpottaa tuotantoprosessia automatisoimalla jokseenkin pikkutarkan prosessin. Grafiikkaan tultaisiin tekemään jonkinlaisia esteettisiä valintoja valaistukseen liittyen jo ilman, että pelimoottorin sisäisiä valaistusominaisuuksia hyödynnettäisiin.

Pelimoottorin valaistusjärjestelmä tosin lisää ylimääräisen tason projektin kehitysprosessiin, jossa ongelmia voidaan kohdata. Lisäksi, vaikka tarvittavat tekstuurit voidaan generoida automaattisesti erillisillä ohjelmilla, tämä ei välttämättä takaa korkeaa laatua luoduille tekstuureille. Parhaimman tuloksen saavutin hyödyntämällä Laigter-ohjelmaa luomaan laajemmat tekstuurien kokonaisuudet ja parannella sitten yksityiskohdat manuaalisesti. Uskoisin silti, että kaikista parhaimman tuloksen niin normaali- kuin kiiltokarttoihin saadaan luomalla ne manuaalisesti.

5.3 Henkilökohtaiset saavutukset

Projektin tietoperustaa tutkiessani opin uutta valaistuksesta kuvataiteessa, videopelien estetiikasta ja sen vuorovaikutuksesta pelien mekaniikkoihin. Kehityin kenttäsuunnittelussa ja digitaalisen grafiikan tuotannossa. Tutkiessani valaistusta kuvataiteessa opin värien hyödyntämisestä kuvataiteessa, valaistuksen ja värien merkityksistä ja symbolisesta tulkinnasta, ja kuvataiteen tuottamisen menetelmistä. Näistä kuvataiteen kautta saaduista opeista opin lisää myös pelituotannosta soveltaessani opittua käytäntöön. Opin pelien estetiikasta, estetiikan sidonnaisuudesta pelien mekaaniseen puoleen ja siitä, kuinka esteettiset valinnat vaikuttavat pelimekaniikkaan ja päinvastoin.

Kenttäsuunnittelusta taas opin sen työvaiheista, merkityksestä pelituotantoon ja siinä käytetyistä menetelmistä.

Tärkeimpänä henkilökohtaisena saavutuksena opinnäytetyössä on sen aikana tuotettu pelidemo. Tuotannon aikana edistin omia graafisen tuotannon taitojani, kuten piirtämistä, ja opin käyttämään Godot-pelimoottoria. Tuotannossa halusin haastaa itseni tuottamalla mahdollisemman paljon itse, tekstuureista koodiin. Tekstuureita piirtäessäni kehitin niin ympäristögrafiikan kuin tekstuurien piirtotaitoja. Piirtäessäni graafisia elementtejä sain myös harjoitusta värien käyttöön käytännössä. Godot-pelimoottorin käytön opettelu kävi kädenkäänteessä tuotannon aikana ohjelmiston kattavan dokumentaation ansiosta. Uuteen pelimoottoriin siirtyminen sujui odotettua paremmin. Odotin enemmän vaikeuksia uuteen ohjelmistoon siirryessä, mutta Godotin ja Unityn välinen käyttöliittymien samankaltaisuus ja laaja valikoima ohjeita tietokannasta ja muualta internetistä teki siirtymän paljon helpommaksi.

6 Pohdinta

6.1 Pelimoottorin valinta

Opinnäytetyö oli alun perin suunniteltu toteutettavaksi Unity -pelimoottorissa hyödyntäen sen ominaisuuksia. Olin aiemmin tutustunut Unity -pelimoottoriin opinnoissani, minkä vuoksi pelimoottori oli alkuperäinen valintani.

Kirjallisuuskatsausta kirjoittaessani, ohjelmistoa kehittävä yhtiö muutti pelimoottorin käyttöehtoja kehittäjille epäedullisiksi ja epäselviksi, joten päädyin vaihtamaan toiseen pelimoottoriin.

Siirtyminen Godot-pelimoottoriin sujui yllättävän mutkattomasti. Molempien moottorien editorit ovat rakenteeltaan hyvin samankaltaisia ja molemmat hyödyntävät C#-ohjelmointikieltä. Godot-pelimoottorin ominaisuuksien kattava dokumentaatio ja selkeä käyttöliittymä helpottivat siirtymää uuteen ohjelmistoon.

6.2 Tuloksen pohdintaa

Loppujen lopuksi olen tyytyväinen opinnäytetyön kirjallisen osuuden laajuuteen ja toiminnallisen osuuden projektin aikana tuotetun demon lopputulokseen. Tukevaa materiaalia löytyi niin kirjoina kuin digitaalisina aineistoina verkosta. Tosin kirjallisuutta suoraan aiheesta 2D-ympäristön kenttäsuunnittelusta tai 2D-valaistuksesta peleissä ei painettuna juuri ole julkaistu, joten jouduin soveltamaan peli- ja kenttäsuunnittelun yleisteoksia. Joissakin tapauksissa jouduin hyödyntämään ja soveltamaan 3D-grafiikan tuotantoon liittyvää kirjallisuutta. Kirjallisuuden lisäksi keskeisimmiksi tiedonlähteiksi osoittautuivat verkkotietokannat, etenkin Godot-pelimoottorin virallinen verkkotietokanta pelimoottoriin liittyen, Blender-ohjelmiston tietokanta tekstuureihin ja grafiikan tuotantoon liittyen, ja Microsoftin viralliset dotNET-tietokannat C# -ohjelmointikieleen liittyvään tietoon.

Opinnäytetyön toiminnallisen osan suoritukseen ryhdyin heti, kun sain kirjallisen osuuden mahdollisimman valmiiksi. Projekti eteni muuten hyvin, mutta kovin hitaasti syksyn 2024 aikana. Tämän lisäksi jouduin leikkaamaan sisältöä alkuperäisestä suunnitelmasta. Alun perin olisin vertaillut kahta erilaista toteutusta, mutta päädyin lopulta tekemään näistä vain yhden. Syynä päätökseen oli valitsemieni menetelmien samankaltaisuus ja opinnäytetyön laajuus. Jo yhden demon tekeminen oli tarpeeksi laaja haluamieni elementtien demonstraatioon.

Demon visuaalisuuteen olen tyytyväinen ja sanoisin, että kykenin luomaan tarpeeksi visuaalisia elementtejä demonstroimaan valaistuksen esteettistä vaikutusta ja merkitystä kenttäsuunnittelulle. Tästä huolimatta jotkin osa-alueet jäivät kumminkin mielestäni hieman vajaiksi. Valaistuksen pelimekaaninen puoli tuntui jäävän hieman heikommaksi demossa. Pelimekaanisessa käytössä valaistusta demossa hyödynsivät ympäristöä valaisevat soihdut, pelaajan ohjaaman hahmon loisto ja valo hohtavat tasot. Oli hankalaa keksiä valaistusta hyödyntäviä mekaniikkoja puhumattakaan sellaisia, joilla dynaamista valaistusta voitaisi demonstroida hyvin. Keksimäni katoavat tasot tuntuvat jälkikäteen

ajatellen edustavan enemmän demon esteettistä näkökulmaa kuin pelimekaanisia ominaisuuksia.

Lähteet

- Blender Foundation. 2023. Blender.org. <https://www.blender.org/>. 11.10.2023.
- Bradfield, C. 2023. Godot 4 Game Development Projects - Second Edition. Birmingham, Iso-Britannia: Packt Publishing Ltd.
- Cuphead. 2017. StudioMDHR Entertainment.
- Egenfeldt-Nielsen, S. & Smith, J.H. & Tosca, S.P. 2008. Understanding video games: The essential introduction. New York, NY: Routledge.
- FileInfo.com. 2024. Vector Image Files. https://fileinfo.com/filetypes/vector_image. 30.5.2024.
- Gahan, A. 2009. Chapter 7 A Basic (Game) Art Education. Teoksessa Ahearn, L. (toim.) Game Art Complete. Burlington, MA: Focal Press/ Elsevier.
- Holm, P; Ignatius, P. 2020. Pelisuunnitteludokumentin luominen konseptitaiteen avulla. Karelia Ammattikorkeakoulu. Media-ala. Opinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2020121628711>. 27.03.2025.
- Innovecsgames. 2023. Next-Gen Realities: The Evolution of 2D and 3D Gaming Experiences. <https://www.innovecsgames.com/blog/next-gen-realities-the-evolution-of-2d-and-3d-gaming-experiences/>. 04.12.2024.
- Itten, J. 1998. Värit taiteessa. Helsinki: Taide.
- Jackman, J. 2012. Lighting for Digital Video and Television, 3rd Edition. Burlington, MA: Focal Press.
- Kosak, D. 2004. John Carmack: The Technology of DOOM 3 and What's Next. GameSpy. <http://pc.gamespy.com/pc/doom-3/539049p1.html>. 04.12.2024.
- Linietsky, J. & Manzur, A. & the Godot community -yhteisö. Godot. 2023. Godot Engine 4.1 documentation in English. <https://docs.godotengine.org/en/stable/index.html>. 13.10.2023.
- Microsoft. 2023. C# | Modern, open-source programming language for .NET. <https://dotnet.microsoft.com/en-us/languages/csharp>. 11.10.2023.
- Moisio, A. 2019. Videopeleissä käytettävän pikseligrafiikan jatkuva suosio. Karelia Ammattikorkeakoulu. Media-ala. Opinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2019052211255>. 27.03.2025.

Mozilla. 2024. Image file type and format guide. https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types. 24.5.2024

Neverhood. 1996. Neverhood Inc.

Pekkarinen, T. 2021. PBR-materiaalin luonti proseduraalisesti ja valokuvan avulla 3D-mallille. Karelia Ammattikorkeakoulu. Media-ala. Oppinnäytetyö. <https://urn.fi/URN:NBN:fi:amk-2021111820606>. 27.03.2025.

Pixcap. 2024. Dynamic lighting. <https://pixcap.com/glossary/dynamic-lighting>. 04.12.2024.

Puhakka, A. 2008. 3D-grafiikka. Helsinki: Talentum.

Salmond, M. 2021. Video game level design: How to create video games with emotion, interaction and engagement. Lontoo, Iso-Britannia: Bloomsbury.

Schell, J. 2020. The Art of Game Design, 3rd Edition. Boca Raton, FL: CRC Press/Taylor & Francis Group LLC.

Totten, C. 2014. An Architectural Approach to Level Design. Boca Raton, Florida: CRC Press; Taylor & Francis Group, LLC.

Unity Technologies. 2023. Unity. <https://unity.com/>. 11.10.2023.

Käytetyt ohjelmat:

Krita 5.0.2 (2022)

Godot 4.3 (2024)

Laigter (2023)

GDD - Pelituotantosuunnitelma: Valaistus Demo

Projektin tavoite

Projektin tavoitteena on demota kaksiulotteisen peliympäristön valaistuksen toteutustapaa Godot -pelimoottorissa. Projektissa tullaan hyödyntämään pelimoottorin sisäänrakennettuja ominaisuuksia valaistuksen toteuttamiseksi. Valmis teos tulee sisältämään elementtejä, joiden avulla valaistuksen ominaisuuksia voidaan demonstroida.

Pelimekaniikka

Tavoitteet

Pelimekaniikan kannalta, teos tulisi olla mahdollisimman yksinkertainen. Tärkeimpänä ominaisuutena on luoda järjestelmä valaistukselle pelimoottorin Node-moduuleja hyödyntämällä. Demo koostuu yhdestä kaksiulotteisesta pelikentästä. Pelikenttään tulee sijoittaa visuaalisia elementtejä, jotka joko reagoivat valaistuksen muutokseen tai tuottavat valoa. Pelaajan tulisi myös olla kykenevä hallitsemaan jonkinlaista elementtiä teoksessa, jotta pelaajalla olisi jonkinlaista kontrollia valaistuksen muuttamiseen.

Pelin varsinainen pelattavuus on toissijaista projektin toteutukseen, mutta demo voi sisältää myös pelattavia elementtejä. Jos mahdollista, pelattavat elementit tulisi sisältää valaistuksen ominaisuuksia, kuten kohteen valaistukseen vaikuttavia tekstuureja, valopisteitä tai valon kulkua estäviä elementtejä.

Ideoita

Pelin genre ideoita:

- Ei spesifiä genreä / Walkingsim
- Shoot-Em-Up
- Tasohyppely / Metroidvania
- Point-N'-Click

Pelimekaniikan ideoita:

- Pelin mekaniikka korostaa valaistusta

Grafiikka

Tavoitteet

Grafiikan päätavoite on valaistuksen toteutus. Grafiikan tuottamiseksi käytetään Krita -piirrosohjelmaa ja tarvittaessa GIMP-kuvamanipulaatio ohjelmaa. Valaistuksen luomiseksi ympäristön grafiikalle tulee luoda pintatekstuurin lisäksi kiilto- ja normaalikartat. Näiden lisätekstuurien luomiseksi projektissa käytetään Laigter-ohjelmaa.

Ideoita

Pelin esteettisen genren ideoita:

- Futuristinen Sci-Fi
- Retrofuturistinen Sci-Fi
- High-Fantasia
- urbaani-Fantasia
- urbaani

Miljöo ideoita:

- Alus (Avaruus, laiva...)
- Luolasto
- Kaupunki
- Huoneisto

Visuaalisten elementtien ideoita:

- Muotoja, jotka korostavat valaistusta
- Valonlähteitä elementeissä
- Himmeä yleinen valaistus, korostaa valaistuja kohteita
- Grafiikka piirretään spritesheet ja tilemap muotoihin

Viitekuvia

Tummat/ei-valaistut kohteet



#418E8F

#207273

#005758

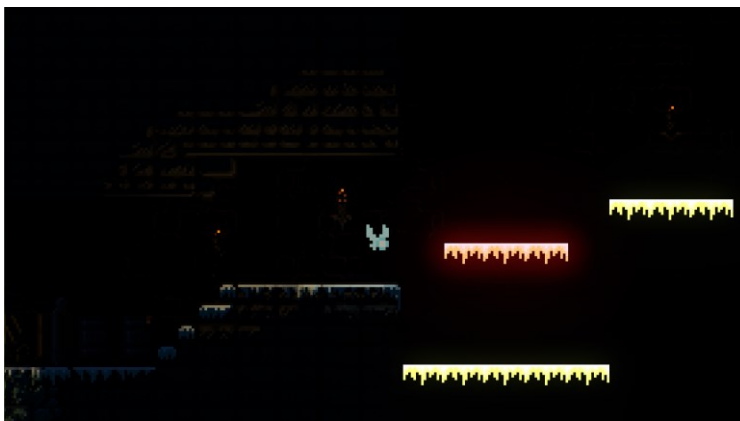
#003D3F

#002527

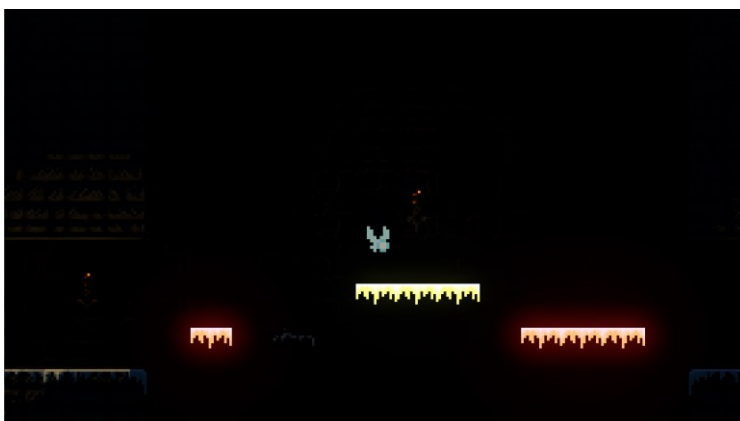
Kuvasarja valmiista demosta



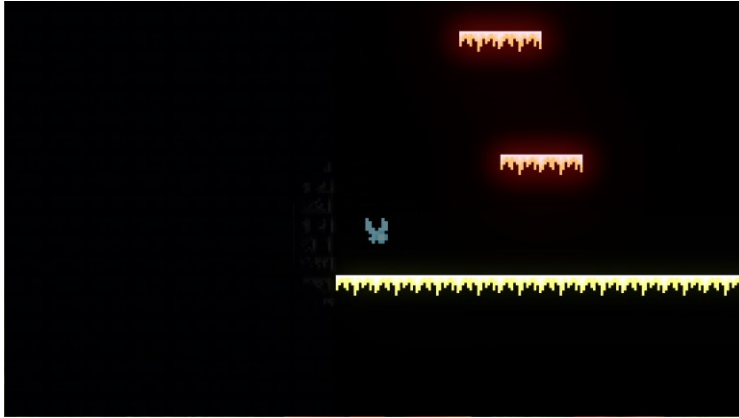
Kuva 1. Kentän alku. Pelaaja ja soihdut valaisevat ympäristöä. Soihdut etenevät kentän kulkusuuntaan.



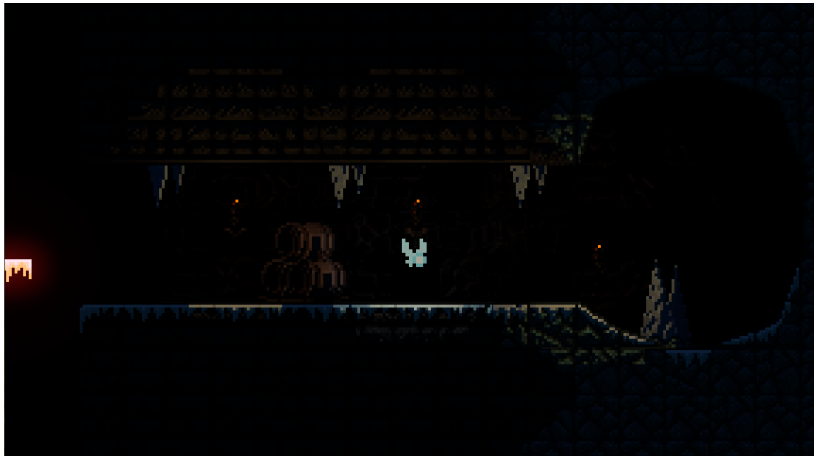
Kuva 2. Kuilu, jossa on hohtavat tasot. Punertavat tasot katoavat pelaajan hypättyä niiden päälle.



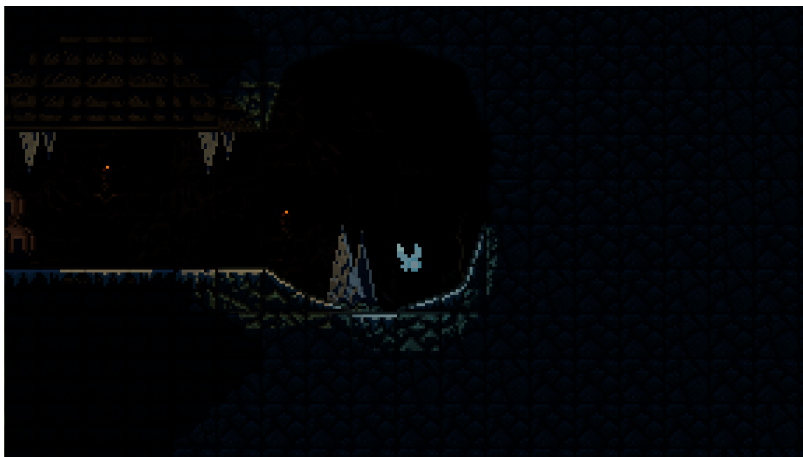
Kuva 3. Kuilun keskellä on kahden heikomman tason ympäröimä vahvempi taso. Tasojen vahvuutta edustavat soihdut, heikompien tasojen soihdut ovat sammuneet ja vahvemman tason soihtu on yhä tullessa.



Kuva 4. Jos pelaaja putoaa kuiluun, pohjalla on vahvempia tasoja pysäyttämässä pudotuksen. Pelaaja voi palata kuiluosion alkuun hyppäämällä ylös katoavia tasoja pitkin.



Kuva 5. Kuilu päihitetty. Soihdut ohjaavat kohti kentän loppua.



Kuva 6. Kentän loppu. Huomaa, kuinka Kuva 5:ssä näkyvän jääpuikon valaistus muuttuu pelaajan hohtavassa valossa.