

samk



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

MIRO SILMU
JERE ARONSALO

PELIN LUOMINEN MODERNEILLA KEHITYSTYÖKALUILLA

TIETOJENKÄSITTELYN TUTKINTO-OHJELMA
2024

TIIVISTELMÄ

Aronsalo Jere, Silmu Miro: Pelin luominen moderneilla kehitystyökaluilla
Opinnäytetyö, AMK
Tietojenkäsittelyn koulutusohjelma
Huhtikuu 2024
Sivumäärä: 35

Opinnäytetyö tehtiin omien intohimojen ja ideoiden innostamana, jonka vuoksi sillä ei ole toimeksiantajaa.

Tavoitteenamme oli luoda peli uusimmalla Unreal Engine versiolla, jotta opinnäytetyö toisi lukijalle realistisen kuvan pelikehityksen helppoudesta.

Pelissä käytettiin ohjelmointikielenä Unreal Enginen omaa Blueprint -kieltä, sillä se mahdollistaa erittäin paljon helpomman ja sulavamman koodaamisen.

Peli ei tullut opinnäytetyön aikana täysin valmiiksi, sillä opinnäytetyön pääpaino oli osoittaa, kuinka helppoa pelikehitys on nykyisillä kehitystyökaluilla. Peli valmistui kuitenkin pelattavaksi versioksi ja sitä on tarkoitus jatkokehittää opinnäytetyön jälkeen.

Avainsanat: Unreal Engine, Pelikehitys, Ohjelmointi

ABSTRACT

Aronsalo Jere, Silmu Miro: Creating a game with modern development tools
Bachelor's thesis
Business Information Systems
April 2024
Number of pages: 35

The thesis was made of our own passions and ideas hence it does not have commissioning party.

Our goal was to create a game using the latest version of Unreal Engine, so that the thesis provides a realistic portrayal to the reader of the ease of game development.

The game utilized Unreal Engine's own Blueprint language as the programming language, as it enables much easier and smoother coding.

The game was not fully completed during the thesis period, as the main focus of the thesis was to demonstrate how easy game development is with current development tools. The game, however, was completed into a playable version and we have plans on developing it further after the thesis.

Keywords: Unreal Engine, Game Development, Programming

SISÄLLYS

1 JOHDANTO	7
2 UNREAL ENGINE.....	8
2.1 Tuki eri alustoille.....	8
2.2 Uudet ominaisuudet.....	9
2.2.1 Nanite	9
2.2.2 Lumen.....	9
2.2.3 MetaHuman	10
3 PELI-IDEA.....	11
3.1 Tarina	12
3.2 Teema	12
4 SUUNNITTELU.....	13
5 TOTEUTUS.....	14
5.1 Kehitysympäristön perustaminen	14
5.2 Maailman luonti	15
5.2.1 Maailman värjäys.....	17
5.2.2 Maailman yksityiskohtien optimointi.....	19
5.3 Pelihahmon luonti.....	19
5.3.1 Animointi.....	20
5.4 Cut Scenet.....	21
5.5 Pelimekaniikat	22
5.6 Äänet.....	23
5.7 Ajoneuvot	25
5.7.1 Ajoneuvon ajettavuus	25
5.7.2 Ajoneuvon sisään- ja ulostulo	26
5.8 Pelin käyttöliittymä.....	27
5.9 Päätehtäväketju.....	30
6 JATKOKEHITYS.....	31
7 LOPUKSI	33
LÄHTEET	34

SYMBOLI- JA LYHENNELUETTELO

NPC – Non-player character, eli sivuhahmoja, jota pelaaja ei itse ohjaa, vaan pelinkehittäjät ovat koodanneet hahmon liikkeet ja vuorovaikutukset.

Repository – Paikka, jossa säilytetään ja hallitaan tiedostoja. Repository voi olla fyysinen, kuten palvelin, tai virtuaalinen, kuten pilvipalvelu.

Renderöinti – Prosessi, jossa tietokone luo visuaalisen kuvan tai animaation 3D-mallista tai tietokonegrafiikasta.

Asset – Digitaalinen resurssi, jota voidaan käyttää osana projektia. Voi olla mitä tahansa resurssia, esimerkiksi kuvia, ääniä, videoita tai 3D-malleja.

Tekstuuri – Kuvio tai kuviointi, joka lisätään pinnalle antamaan visuaalisia elementtejä ja yksityiskohtia.

Cut Scene – Välianimaatiokohtaus, joka esitetään pelaajalle pelin edetessä. Näiden kohtausten tarkoitus on usein edistää ja selventää tarinaa, sekä esitellä uusia hahmoja ja käännteitä.

Open World – Pelityyppi, jossa pelaajalla on vapaus tutkia maailmaa vapaasti ilman rajoja, tai selkeää lineaarista tehtäväketjua.

Optimointi – Tarkoittaa prosessia tai toimenpidettä, jolla pyritään parantamaan järjestelmän tehokkuutta ja suorituskykyä.

Gitignore – tiedosto, tai ominaisuus, jota käytetään versionhallintajärjestelmä Gitissä määrittämään, mitä tiedostoja tai kansioita ei tule seurata tai tallentaa repositorioon.

Mannequin-hahmo – valmis hahmo, joka toimii referenssinä ja resurssina pelinkehityksessä Unreal Engineissä. Mannequin-hahmo on tarkoitettu käytettäväksi animaatioiden ja liikkumisen testaamiseen.

Plug & Play – Ilmaisee ohjelmistoja tai laitteita, joiden yhdistäminen ei vaadi fyysistä laitekonfigurointia tai käyttäjän toimia resurssiristiriitojen ratkaisemisessa.

Vsync – Vertical synchronization, eli tekniikka, joka kertoo näytönohjaimelle, milloin näyttöpuskuriin voi piirtää, jotta piirtäminen ei näy ruudulla.

Skeletal Mesh – Joukko polygoneja, jotka on tehty muodostamaan verkkoja hierarkkiseen luurankoon, jotka voidaan animoida verkon muuttamiseksi.

Physics Asset – Fyysinen ominaisuus, jota käytetään määrittämään skeletal mesh:n fysiikka ja törmäys.

Hitbox – pelaajalle näkymätön alue, jota käytetään merkitsemään pelissä olevia alueita, joiden kanssa pelaaja voi vuorovaikuttaa.

1 JOHDANTO

Tässä opinnäytetyössä kerrotaan yhdessä suunnitellun peli-idean toteuttamisesta uusimmalla Unreal Engine 5 versiolla. Molemmat opinnäytetyöntekijät ovat suorittaneet opintojen aikana Let's Make a Game –kurssin, jossa pelinkehitys tapahtui vanhemmalla Unreal Engine 4.7 versiolla. Kyseinen kurssi toi molemmille erittäin tärkeitä taitoja toteuttaa tämä opinnäytetyö yhdessä, kursilla opittujen asioiden pohjalta.

Tämän opinnäytetyön tavoitteena oli rakentaa peli sellaiseen vaiheeseen, jossa Unreal Enginen mahdollisuuksia olisi saatu mahdollisimman laajasti hyödynnettyä ja peli olisi pelattavassa versiossa.

Opinnäytetyössä kerrotaan aluksi itse Unreal Engine pelimoottorista ja sen uusimman version tuomista lisäominaisuuksista, jonka jälkeen sisältö painottuu eri kehitysvaiheisiin matkan aikana. Tämä opinnäytetyö ei kumminkaan ole suunnattu opettavaisena aiheena, vaan enemmänkin toteutuksen ja Unreal Enginen mahdollisuuksien kerrontana. Opinnäytetyössä tullaan käyttämään englanninkielistä termistöä, sillä Suomen kielellä ei löydy vastaavia termejä. Termistö löytyy selitettynä lyhenneluettelossa.

2 UNREAL ENGINE

Unreal Engine on Epic Games -ohjelmistokehitysyhtiön julkaisema videopelien kehitystyökalu. Unreal Enginen avulla kehittäjät voivat rakentaa pelejä, luoda simulaatioita, editoida videoita, sekä animoida. Kehittäjät ovat käyttäneet kyseistä työkalua rakentaessaan suurimpia markkinoilla olevia pelejä, kuten Fortnite. (Moss, N.d.)

Unreal Engine tarjoaa monia merkittäviä etuja, jotka tekevät siitä erinomaisen valinnan sekä aloitteleville että kokeneille pelikehittäjille. Yksi sen suurimmista vahvuuksista on sen erinomainen käyttäjäystävällisyys, joka tekee siitä helposti lähestyttävän työkalun jopa niille, joilla ei ole aiempaa kokemusta pelinkehityksestä tai ohjelmoinnista.

Unreal Engine on ilmainen käyttäjälle siihen asti, kunnes projekti tuottaa tekijälleen yli miljoona dollaria (Saarenoja, 2022).

2.1 Tuki eri alustoille

Yksi syy, jonka vuoksi valitsimme Unreal Enginen kehitystyökaluksi, oli sen tuki kaikille kuuluisimmille alustoille, silmällä pitäen jatkokehitystä ja julkaisemista.

Unreal Enginen avulla voi kehittää sisältöä laajalle valikoimalle tietokone-, konsoli- ja mobiilialustoja, kuten Windows, MacOS, Playstation 4 ja 5, Xbox one ja Series X, Nintendo Switch sekä iOS- ja Android-mobiililaitteet. Unreal tarjoaa laadukkaan ratkaisun augmented reality (AR), virtual reality (VR), ja mixed reality (MR) pelien luomiseen. Unreal Engine tarjoaa myös tuen 4K UHD-videon ja ääniin, sekä täyden tuen useiden eri videosyötteiden ja signaalikäsittelylaitteiden välillä. (Unrealengine.com, 2021)

2.2 Uudet ominaisuudet

Unreal Enginen viides sukupolvi julkistettiin 13.5.2020 YouTubessa Playstation 5:n moottorin trailerilla. Unreal Enginen alkuperäinen luoja Tim Sweeney kertoi, että Unreal Enginen viidennessä sukupolvessa on panostettu kehittäjien helppompaan ja mutkattomampaan kehittämiseen, sillä upeiden pelien teko vaatii erittäin paljon aikaa ja rahaa. (Statt, 2020.)

Unreal Enginen viidennen sukupolven isoimpia muutoksia ja parannuksia ovat Nanite, joka on Unreal Enginen uusi geometriasysteemi, sekä Lumen, joka on uudelle sukupolvelle päivitetty valoitussysteemi.

2.2.1 Nanite

Nanite on virtualisoitu geometriajärjestelmä, joka mahdollistaa kehittäjille uskomattoman yksityiskohtaisten isojen maailmojen luomisen tehokkaasti. Nanite on teknologia, joka pystyy käsittelemään valtavia yksityiskohtamääriä reaaliajassa ilman suorituskyvyn heikkenemistä. Se on suunniteltu poistamaan perinteisten polygonipohjaisten renderöintijärjestelmien asettamat rajoitukset. Perinteisissä renderöintitavoissa monimutkaiset 3D-mallit jaetaan yksinkertaisempiin monikulmihin, jotka sitten renderöidään näytölle. (Premawardhana, 2023).

Lyhykäisyydessään Nanite tarjoaa moottorin käyttäjille rajattoman geometrian, joka mahdollistaa miljoonien pienten pikseleiden ansioista erittäin tarkat tekstuurit pienellä suorituskyvyllä. (Barrier, 2022.)

2.2.2 Lumen

Lumen on täysin dynaaminen valaistus- ja heijastusjärjestelmä. Lumen käyttää epäsuoraa valaistusputkistoa, eli kohtauksen geometria, materiaali ja valo-ominaisuudet voivat muuttua milloin tahansa. Dynaamisen epäsuoran

valaistusputkiston avulla valaistuspäivitykset ovat välittömiä ja se poistaa rakennusaikaa lopullisen valaistuksen laadun saavuttamiseen. Tämän lisäksi Lumen-valaistut kohtaukset eivät tarvitse heijastuskuutiokarttoja, koska Lumen korvaa täysin muut menetelmät ja pystyy tuottamaan geometrisesti tarkkoja heijastuksia. (Souza, N.d.)

Uuden sukupolven konsoleille suunnattu järjestelmä tarjoaa erittäin realistiset valaistukset ja heijastukset suurissa ja yksityiskohtaisissa ympäristöissä. Edeltävässä sukupolvessa näiden kanssa oli joitakin ongelmia, kuten vääriä heijastumia yksityiskohtaisissa paikoissa pelimaailmassa. (Starloopstudios, N.d.)

2.2.3 MetaHuman

Unreal Engine 5 toi mukanaan täysin ilmaisen MetaHuman –nimisen ominaisuuden, jonka avulla kehittäjät pystyvät tekemään hahmoja peliinsä erittäin helposti. Peliin tehtiin päähahmo, sekä sivuhahmot, eli NPC:t, käyttämällä kyseistä ominaisuutta. MetaHuman Creator tarjoaa vakiona useita valmiita hahmoja, joita voi itse muokata haluamallaan tavalla.

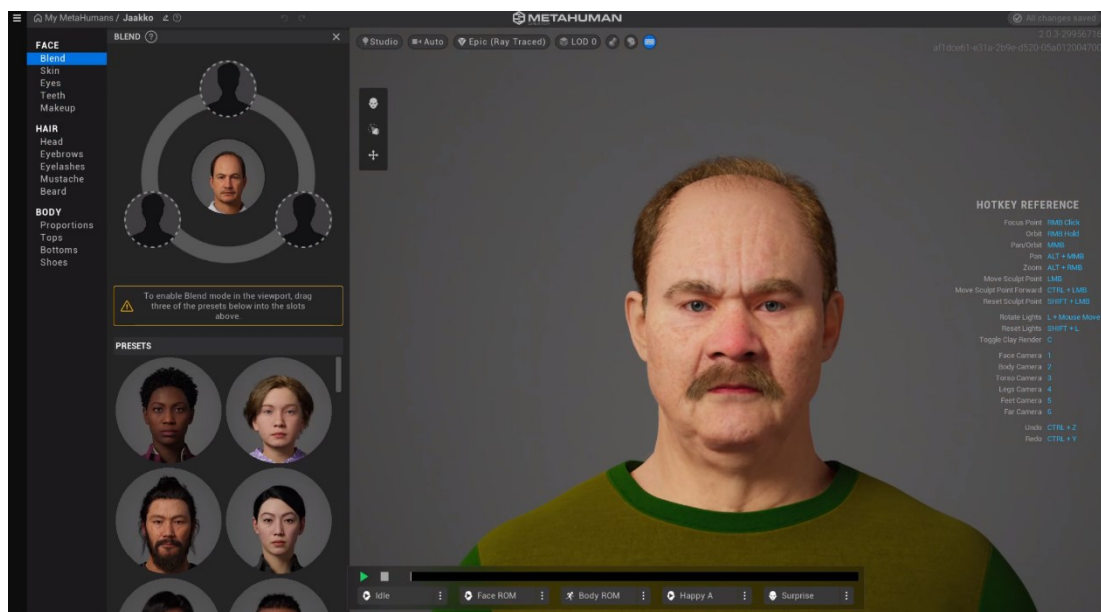
Mesh To Metahuman -ominaisuuden avulla käyttäjät voivat ottaa itse tehdyt kasvomallinsa, jotka on luotu skannaamalla, muotoilemalla tai perinteisillä mallinnustyökaluilla, ja muuntaa ne MetaHumaniksi. Lisäksi niitä pystyy hienosäätämään lisää MetaHuman Creatorissa.

MetaHuman Creatorissa on useita ominaisuuksia, mukaan lukien yli kymmenen kasvojen animaatiota, kuusi vartalotyyppiä, useita vaatetusvaihtoehtoja, monia hiustyylejä, sekä yli kymmenen parta-, viiksi-, kulmakarva- ja ripsityyliä. (Miller, 2022)

Tarkoituksena oli alun perin mallintaa oikean ihmisen kasvot pelihahmon naamaksi, mutta tämä osa projektia jäi toteuttamatta rajallisen ajan vuoksi.

MetaHuman Creatorin valmiit hahmot ja niiden muokattavuus olivat enemmän, kun tarpeeksi pähahmon valinnalle.

Luodun MetaHumanin pystyy lataamaan kätevästi suoraan projektiinsa Quixel Bridge-sovelluksen kautta, joka on integroituna Unreal Engineen. MetaHumanin negatiivinen puoli on hahmojen suuri tilan ja resurssien vaativuus yksityiskohtaisuutensa vuoksi. Tähän asiaan viitattu luvussa 5.1.



Kuva 1. MetaHuman Creator ja pelihahmo Jaakko.

3 PELI-IDEA

Ideana oli luoda Open World -tyyppinen Suomisimulaattori –peli, jossa pelaaja pääsee kokemaan suomalaista elämää keski-ikäisen miehen näkökulmasta. Kesämökin kalustaminen toimi tehtävien päämääränä ja tavoitteena tässä peliprojektissa. Peliin oli suunnitelmassa luoda tehtävämekanismi, jossa pelaaja

suorittaa erilaisia tehtäviä ansaitakseen rahaa huonekaluihin, joilla kalustaa vanhemmilta perinnöksi jäänyt kesämökki.

Päätarinan ohella peliin oli suunnitelmassa tehdä suomalaisille tuttuja sivutehtäviä, kuten marjastus, puiden hakkaus, saunan sytyttäminen ja saunominen. Valitsimme tähän peliin kolmannen persoonan näkökulman, koska koimme sen kyseiselle projektille parhaaksi vaihtoehdoksi näyttämään Unreal Enginen kykyjä ja mahdollisuuksia.

3.1 Tarina

Pelin tarina kertoo 50-vuotiaasta Suomalaisesta Jaakosta, joka vanhempien menehtymisen myötä saa perinnöksi heidän vanhan mökkinsä maaseudulta. Vanhempien suurten velkojen vuoksi melkein kaikki mökin huonekalut ovat ulosotettu, joten Jaakon tehtävänä on ansaita rahaa mökin uudelleen kalustamiseen tekemällä erilaisia palveluksia ja tehtäviä maaseudun muille asukkaille. Jaakon matkassa tapahtuu kuitenkin erilaisia ongelmia, sillä Jaakko ei ole nuoruusvuosien riehuminen vuoksi kylällä erityisen pidetty henkilö.

Jaakon tehtäviin lukeutuu esimerkiksi erilaisten tavaroiden kускаaminen, siivoaminen ja naapureiden auttaminen. Seikkailuun kuuluu myös erilaisia pieniä sivutehtäviä, joita pelaaja voi suorittaa milloin tahansa tarinankulun aikana.

3.2 Teema

Teemaksi valitsimme rauhallisen Suomen luontomaiseman, joka muistuttaa lomaa mökillä. Teema valittiin jo ajoissa, koska sitä pystyttiin hyödyntämään pelin myöhemmissä kehitysvaiheissa helposti, eikä tarvinnut miettiä sopiiko keksitty ominaisuus pelin teemaan vai ei.

Kun pelin teema on päätetty, pelintekijät voivat käyttää sitä linssinä, jonka läpi voidaan tarkastella mitä tahansa ominaisuutta, kohtausta, hahmoa, tekstiä tai jopa mekaniikkaa, joka on luotu peliin. (Thomas, 2020.) Teeman avulla

esimerkiksi pelin kenttien, hahmojen ja äänien valitseminen on kaikille pelinkehittäjille helpompaa.

4 SUUNNITTELU

Pelin suunnittelu alkoi omasta ajatuksesta. Olisi hienoa tutustua uuteen Unreal Engine 5 pelimoottoriin, ja vielä tehdä opinnäytetyö aiheesta. Pelin suunnittelu on keskeinen asia pelin kehityksessä. Alkaen yksinkertaisesta ajatuksesta, pelin konseptoinnista ja suunnittelusta aina valmiin pelin toteuttamiseen asti. Jokainen vaihe vaatii omistautumista, luovuutta ja teknistä osaamista. Pelin kehittäminen vaatii hurjasti työtä ja loputtomasti työtunteja. Onneksemme meitä oli kaksi innostunutta kehittämässä peliä, joten peli valmistui pelattavaan kuntoon.

Tässä opinnäytetyössä suunnitelma ei kuitenkaan toteutunut loppuun asti rajallisen ajan vuoksi. Onneksi meillä on kuitenkin mielenkiintoa ja mahdollisuus jatkaa pelin kehittämistä opinnäytetyöprojektin jälkeen. Jatkotyöskentely avaa myös ovia uusiin mahdollisuuksiin, kuten pelin julkaisemiseen ja markkinoimiseen, joskus kaukana tulevaisuudessa.

Peliin haluttiin luoda mökkiaktiviteetteja, kuten uiminen, saunominen, juominen ja autolla kulkeminen. Halusimme peliin paikoiksi oman mökin, naapurin talon ja pienen kaupan. Näiden lisäksi halusimme luoda ääniefektit ja maiseimat, jotka sopivat pelin teemaan ja antavat pelaajalle realistisen kuvan maaisympäristöstä.

5 TOTEUTUS

Tässä kappaleessa käymme läpi pelinkehityksen eri vaiheet ja ominaisuudet, joita opinnäytetyön aikana toteutimme. Matkan varrella kohtasimme monia haasteita ja epäonnistumisia, mutta kuitenkin myös onnistumisia.

Kaikkea suunniteltua työtä, jotka loppujen lopuksi haluamme saada valmiiksi, emme kuitenkaan saaneet toteutettua opinnäytetyön aikana, ja tätä avaamme lisää opinnäytetyön luvuissa 6. ja 7.

5.1 Kehitysympäristön perustaminen

Kehitysympäristönä käytimme GitHub Desktop -sovellusta, jolla pystyimme kehittämään kätevästi yhdessä projektia. Tällä tavalla pelinkehitys ei onnistunut samanaikaisesti, mutta emme kokeneet tätä opinnäytetyön aikana ongelmaksi.

GitHub Desktop on avoimen lähdekoodin graafinen käyttöliittymä, versionhallintajärjestelmä Gitille, sen sijaan, että käytettäisiin komentoriviä tai verkkoselainta. GitHub Desktop mahdollistaa kehittäjille komentojen, kuten repositorioiden luomisen vain yhdellä napsautuksella. Tämä kätevyys lisää joustavuutta työskentelyyn Gitin parissa ja yhteistyöhön muiden kehittäjien kanssa. (Terra, 2023)

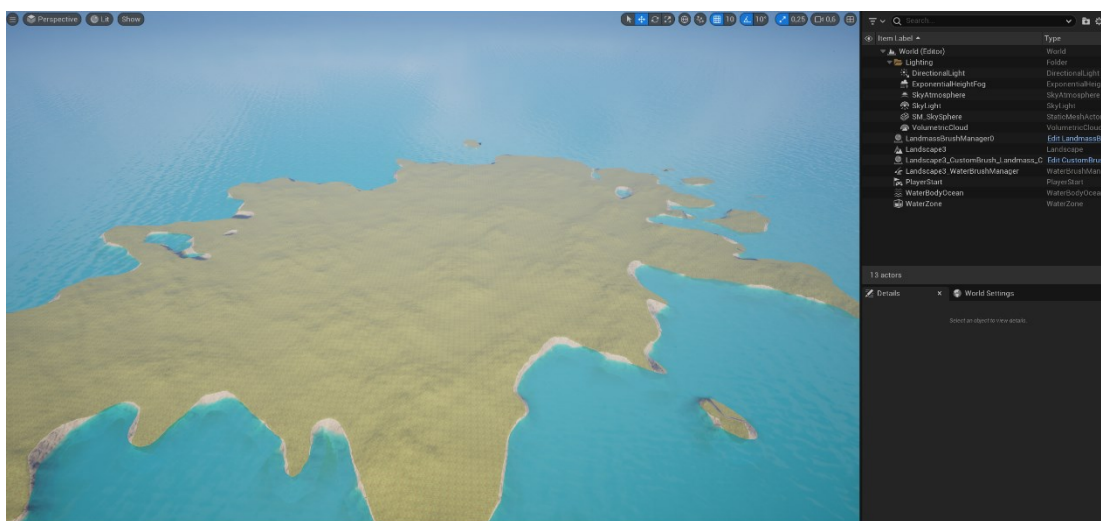
Loimme yhden repositorion, johon kutsuimme molemmat opinnäytetyön tekijät. GitHub Desktop -sovelluksella voi kätevästi vetää uusimmat päivitykset omaan paikalliseen repositorioonsa yhdellä napin painalluksella, jonka jälkeen oma kehittäminen voi alkaa. Joka kerta kun uusia päivityksiä tekee, tulee kehittäjän puskea päivityksensä yhteiseen kansioon, jotta toinen kehittäjä saa kopioitua uusimmat tiedostot ja muutokset itselleen jatkaessaan projektia. Kun Unreal Engineissä tekee muutoksia ja tallentaa, havaitsee Github Desktop

automaattisesti muutokset, jotka voidaan itse sovelluksesta puskea yhteiseen repositorioon.

GitHub Desktopilla ei pysty puskemaan yhteiseen repositorioon yli 100 megatavun tiedostoja, jonka koimme jokseenkin ongelmalliseksi, sillä esimerkiksi pelihahmojen päät olivat kooltaan yli 100 megatavua yksityiskohtaisuutensa vuoksi. Tämä asia ratkesi jakamalla kyseiset tiedostot toisillemme Sharepointin avulla, jotta ne saatiin ladattua itsellemme paikallisesti tietokoneelle. Tämän jälkeen käytimme Gitignore-ominaisuutta. Kun tiedosto lisätään gitignoreen, se ei puskiessa kopioidu yhteiseen repositorioon, eikä tällöin 100 megatavun raja ylitä. GitHub Desktop kehitysympäristön perustamisen jälkeen loimme Unreal Engine-projektin itse pelimoottorissa, jonka jälkeen pääsimme aloittamaan pelin kehittämisen.

5.2 Maailman luonti

Maailman luonti alkoi maaston tekemisestä, joka on havainnointu kuvassa 2. Maailmaan rakennettiin kaupan lisäksi muutamia mökkejä, jotka toimivat maailman muiden NPC-hahmojen koteina.



Kuva 2. Maailman luonti alkoi tästä.

Unreal Engine tarjoaa Landscape –tilassa erinomaiset työkalut saaren tekemiseen, josta syntyi tämän pelin kartta. Saari toimi hyvin rajoittamaan pelialuetta, jotta se ei paisu liian suureksi tämän kaltaiselle pelille, jonka tarina on erittäin rajallinen. Saari on ympäröitynä vedellä, ja näkymättömällä seinällä, jotta pelaaja ei pääse pelimaailman rajojen ylitse. Saaren luomiseen käytettiin Unreal Enginestä löytyvää pohjaa tyhjälle maailmalle, jotta maailman sai itse muokata alusta asti omanlaiseksi. Tämän jälkeen maastoon lisättiin topografista vaihtelua luomalla kuoppia ja pieniä kumpareita, jotta maailma ei olisi tasainen vaan monimuotoisempi.

Unreal Enginestä löytyy myös Ocean Plugin, joka on kokeilullinen kehittäjien käytössä oleva vesilisäosa. Tämä lisäosa muodosti luonnonmukaisia lammi-koita ja järviä, elävöiden ympäristöä ja tuoden siihen vesielementin. Vesi oli oletuksena kirkkaan sinistä, mutta se värjättiin lopuksi muistuttamaan suomen järviä, eli sinivihreäksi.

Pelimaailman yksityiskohtiin käytettiin Quixel Bridge–sovelluksen assetteja, eli esimerkiksi 3D-mallinnettuja objekteja, kuten puita, puskia ja nurmikkoa. Näitä käyttämällä pelimaailma muuttui mukaansa tempaavammaksi ja eloisammaksi. Quixel Bridge on sovellus, joka on Unreal Enginen 5:ssä versiossa integroituna, ja josta kehittäjät voivat hakea ilmaiseksi assetteja peliinsä. Yksityiskohdissa panostettiin erityisesti Suomessa esiintyviin kasveihin ja puihin, esimerkiksi kuusiin ja mäntyihin. Pelihahmon kesämökki haettiin Unreal Enginen virallisesta kaupasta, sen ollessa kuukauden ilmaistarjous.

Quixel Bridge –sovelluksesta haimme myös materiaaleja, esimerkiksi lehdikköä, puuta, ja asfalttia. Näitä käyttämällä maailmaa pystyy värjäämään haluamakseen, käyttämällä Unreal Enginen Landscape -toiminnon maalaus työkalua, josta kerrotaan lisää luvussa 5.2.1.

Maailmaa luodessa ideana oli toteuttaa mahdollisimman realistinen Suomen näköinen maalaismaisema. Maailmaan luotiin esimerkiksi kauppa nimeltä L-Market, joka saattaa etäisesti muistuttaa Suomessa esiintyvää kauppa

logoltaan. Kuvassa 3. on havainnollistettu kuva L-Marketista. L-Market toimii yhtenä kulmakivenä pelin mekaniikoissa ja tarinassa, sillä muutamiin tehtäviin lukeutuu kaupassa asiointi erilaisten aktiviteettien vuoksi.



Kuva 3. L-Market, jossa pelaaja tulee asioimaan tehtäviin liittyvissä asioissa.

5.2.1 Maailman värjäys

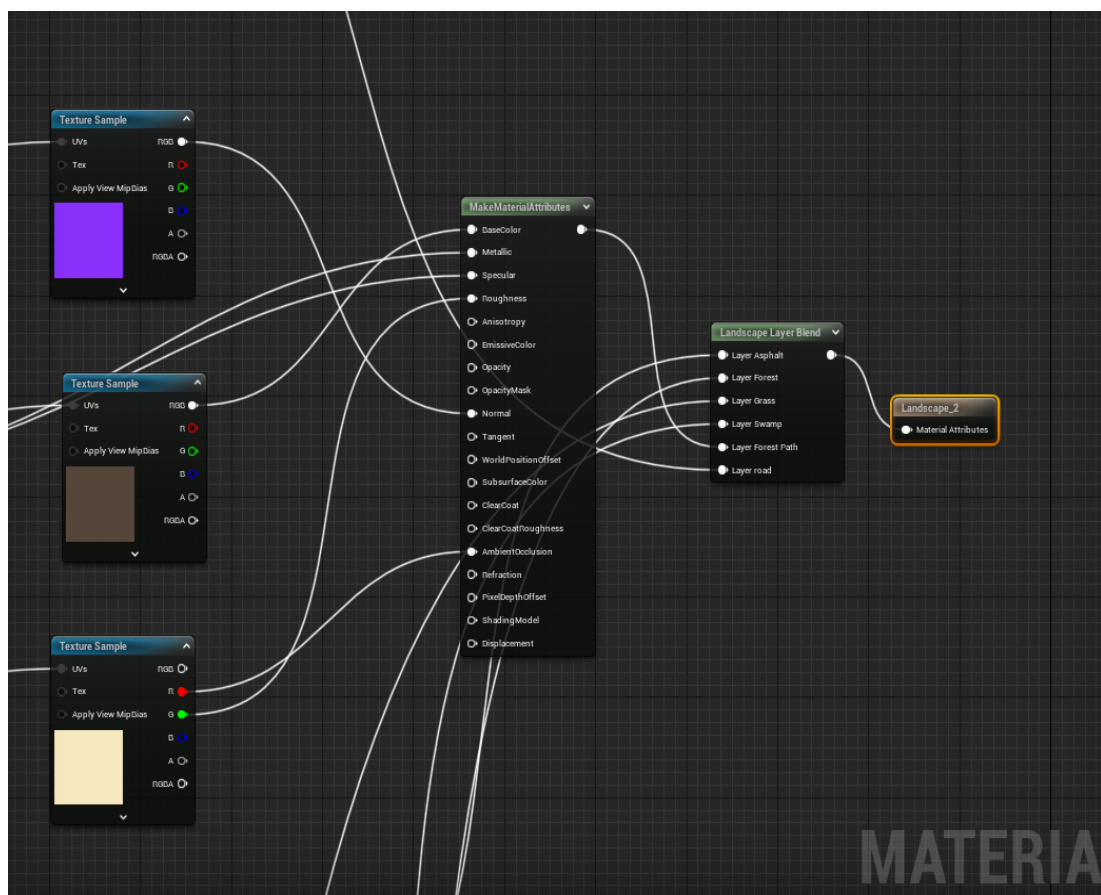
Landscape-toiminnon maalaustyökalulla pystyy kätevästi värjäämään myös maailman haluamansa näköiseksi. Tässä projektissa värjäsimme maaston Bridge sovelluksesta haetulla ruohikkotekstuurilla, joka toimi pohjana kaikelle muulle värjäykselle. Pohjatekstuurin päälle maalasimme myös esimerkiksi asfalttia ja mutaa. Pohjatekstuuri on havainnoituna kuvassa 3.

Maiseman maalaaminen alkaa luomalla uusi Landscape materiaali. Tämän luodun materiaalin tapahtumakaaviossa voimme määrittää mitä tekstuureita tai värejä haluamme käyttää maiseman maalaamiseen. Materiaalille luodaan tapahtumakaavioon kerroksia, jotka vastaavat tekstuureita, joita aiomme maalaamiseen käyttää, esimerkiksi ruohikko, asfaltti, tai kivetys (Premawardhana, 2021).

Kun lataamme Quixel Bridgestä uuden tekstuurin, tuo se mukanaan 3 erilaista tiedostoa, jotka ovat Normal, Masks ja Albedo.

- Albedo-tiedosto sisältää tekstuurin perusvärit. Se määrittää siis kohteen värit ja visuaalisen ulkonäön.
- Normal-tiedosto sisältää tietoja pinnan korkeusmuodoista. Se vaikuttaa siihen, miten valo heijastuu pinnasta ja auttaa luomaan vaikutelman korkeuseroista, vaikka pinta olisi tasainen. Se lisää yksityiskohtia tekstuurin pintaan.
- Masks-tiedosto sisältää tietoja esimerkiksi karkeudesta, kiillosta ja metallisuudesta. (Ashley. C, VNTANA.com)

Nämä kaikki yhdistettynä syntyvät tekstuuri, jota voidaan käyttää värjäämään maasto realistisen näköiseksi, tämä on havainnointu kuvassa 4.



Kuva 4. Tekstuurin luonti tapahtumakaaviossa.

5.2.2 Maailman yksityiskohtien optimointi

Kun luodaan yksityiskohtaista ja eloisaa pelimaailmaa, tulee eteen erilaisia ongelmia. Yksi näistä on suorituskyvyn heikentyminen silloin, kun useita objekteja renderöidään samanaikaisesti. Esimerkiksi yksi tapa käsitellä tätä ongelmaa on asettaa rajoituksia sille, kuinka kauas ja kuinka yksityiskohtaisesti pelihahmon näkökenttä piirretään. Tämä tarkoittaa käytännössä sitä, että vain tietyllä etäisyydellä pelaajasta olevat objektit ja yksityiskohdat piirretään täydellä tarkkuudella, kun taas kauempana olevat kohteet voidaan piirtää yksinkertaisemmalla tavalla, tai ne voidaan jättää piirtämättä kokonaan.

Quixel Bridge tarjoaa ilmaiseksi animoituja asetteja, kuten puita ja ruohikkoa, jotka liikkuvat tuulen mukana. Animoidut objektit ovat resursseja vaativia tekijöitä.

Optimointi alkoi tekemällä kopio olemassa olevasta liikkuvasta puu -asetista, joka oli haettu Quixel Bridgestä. Asetista otettiin pois animointi ja tätä asetta käytettiin liikkuvien puiden takana. Liikkuvat puut metsässä tuovat eloisan tunteen peliin, ja niiden taakse on helppo piilottaa liikkumattomia puita, jotka vievät vähemmän resursseja. Liikkumattomia puita lisättiin myös pelaajan tavoittamattomiin paikkoihin kaukaisuuteen, tuoden eloisuutta peliin, viemättä turhaan pelaajan resursseja. Myös puskia ja nurmikkoa lisättiin maailmaan ilman animaatioita, sillä niiden liikkeet eivät olleet yhtä näkyviä, kun puiden liikkeet. Näin saimme säästettyä hiukan resursseja suorituskyvystä.

5.3 Pelihahmon luonti

Hahmo, jolla pelaaja tulee pelin pelaamaan, on tärkeä osa peliä. Varsinkin pelissä, jonka kuvakulma on kolmannesta persoonasta. Tällöin pelaaja näkee hahmonsa jatkuvasti. Kolmannen persoonan kuvakulmalla tarkoitetaan

näkökulmaa, jossa kamera seuraa pelaajaa jatkuvasti takaapäin. Tällöin pelaaja näkee ympäristönsä laajemmin, sekä hahmonsä liikkeet tarkemmin.

Yksi avainsyistä, jonka vuoksi loimme pelin kolmannen persoonan näkökulmasta tässä opinnäytetyöprojektissa, oli Metahuman-ominaisuuden hyödyksi käyttö. Ajatuksena oli tehdä hahmo, joka muistuttaa keski-ikäistä suomalaista miestä. Tämä onnistui erittäin helposti Metahumana hyödyntämällä. Kuvassa 1. näkyy peliin lisätty hahmo, joka nimettiin Jaakoksi. Pelihahmoa saatiin muokattua Metahuman Creatorissa vanhemmaksi lisäämällä ihoon ryppyjä ja kulumia. Hiuksina käytimme yhtä ominaisuudesta löytyvää valmista hiustyylä, jossa hiusraja oli hieman karannut. Pelihahmon hampaat muokattiin kellertäväksi ja huonokuntoiseksi.

5.3.1 Animointi

Pelihahmon perusanimaatioina hyödynsimme Unreal Enginestä löytyviä valmiita animaatioita, kuten hyppäämistä ja kävelyä. Lisäsimme peliin animaatioita, kuten käden liikkeen oven avaamista varten, juoksemisen ja uimisen. Hankimme kaksi uintanimaatiota Mixamo-nimiseltä sivustolta, koska tarvitsimme erilliset animaatiot uinnille sekä paikallaan uivalle pelaajalle. Animaatioiden integrointiin käytimme Unreal Engine tarjoamaa BlendSpace1D:tä, jonka avulla eri animaatiot voidaan yhdistää saumattomasti. Tämän avulla voidaan määrittää pelaajan liikkeiden nopeus ja suunta animaatiota suorittaessa. Lisäksi BlendSpace1D mahdollistaa näkymän animaatioiden välisestä siirtymästä, kuten paikallaan uimisen ja eteenpäin uimisen yhdistämisen.

Luomamme uimisanimaatio piti toteuttaa koodaamalla Unreal Enginen Mannequin-hahmolle. Ensimmäisenä vaiheena muutimme pelissä olevan veden törmäysmekaniikkoja, jotta se olisi uimakelpoista. Tämän jälkeen kehitimme mekaniikan uimisen aloittamiselle ja lopettamiselle, jotta peli tunnistaa, milloin hahmon tulee uida ja milloin kävellä. Uuden syötön liikkeelle saamiseksi uimista varten piti lisätä peliin erillinen mekaniikka, koska Unreal Engine 5:ssä liikkeiden syöttö on normaalisti tarkoitettu vain kävelyyn ja hyppimiseen.

Tämän jälkeen toteutimme animaation Mannequin-hahmolle käyttäen Unreal Enginen animaatio Blueprint-menetelmää, jossa määrittelimme uimisanimaation aktivoituvan, mikäli hahmo oli vedessä tarpeeksi syvällä uimista varten, ja vastaavasti kävelyanimaation, jos hahmo oli tarpeeksi lähellä rantaa kävelyä varten.

5.4 Cut Scenet

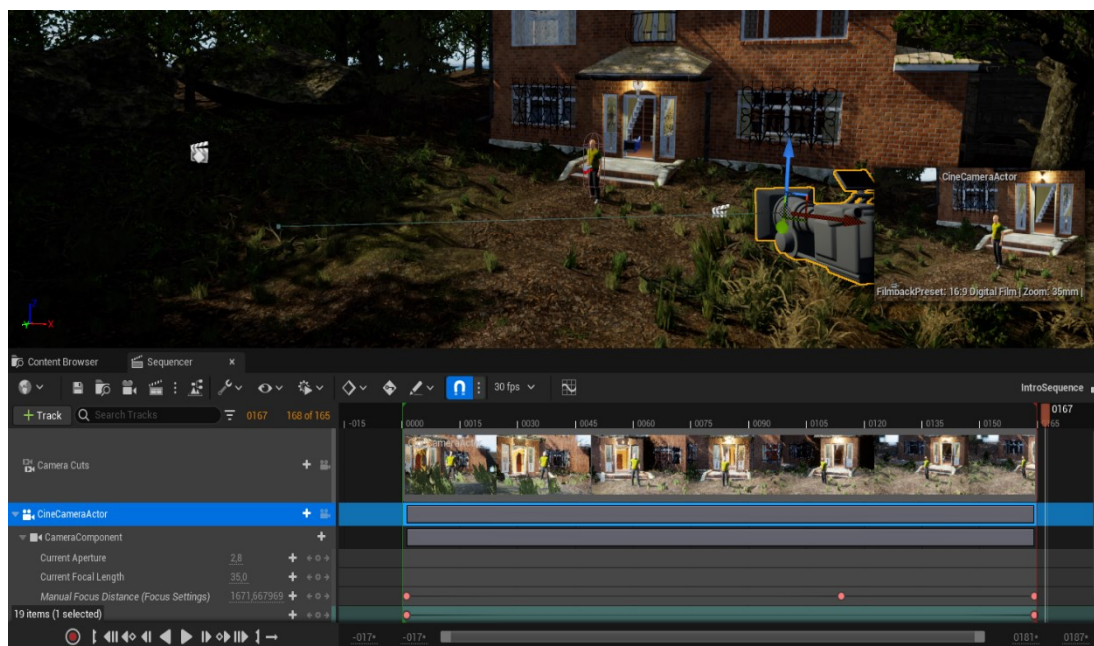
Unreal Enginellä peliin pystyy luomaan pieniä animoituja kohtauksia eli Cut Scenejä. Se voi sisältää tarinan kannalta tärkeitä käänteitä ja tarinan selvennyksiä. Cut scenejä voi luoda eri tarinankulun vaiheisiin. Esimerkiksi kun pelaaja saapuu uudelle tehtävälle, voi tehtävän aloittaa cut scenen saattelemana, jolloin tehtävän tavoite tulee pelaajalle heti tehtävän alkaessa selville.

Unreal Engineen on luotu tätä varten Sequencer –niminen ominaisuus. Sequencer on tehokas työkalu, joka mahdollistaa animoitujen kohtausten luomisen suoraan pelimoottorissa. Se tarjoaa työkaluja kameran liikkeiden, ääniraitojen ja visuaalisten efektien hallintaan.

Opettelumielessä teimme pienimuotoisen pelin alkamisen cut scenen, jossa pelihahmo venyttelee ja kamera nousee kuvatessa pelihahmoa. Cut scenen luonti alkoi Unreal Enginen kohtauksen, eli Sequencen luomisella. Tämän jälkeen asetettiin kamera maailmaan. Sequencer- työkalulla pystyttiin määrittämään kameran aloitus- ja loppukuvakulma laittamalla aikajanaan merkinnät kameran alku- ja loppupisteistä. Tämän seurauksena Unreal Engine luo kameralle automaattisesti liikkumisradan näiden kahden pisteen välillä. Kuvassa 5. on havaittu punaisilla pisteillä kameran liikkuminen ja tarkennus vasemmalta oikealle kulkevassa aikajanassa. Sekä tarkennus että kameran kuvakulma muuttuvat progressiivisesti pisteestä pisteeseen kohtauksen edetessä.

Loimme lisäksi päätarinaketjun ensimmäiseen tehtävään lyhyen cut scenen, jossa kamera kuvaa kauppiasta puhuessaan pelihahmolle. Kyseiselle cut scenelle piti määrittää myös triggerbox, eli animaatiokohtauksen laukaisukohta.

Pelimaailmaan asetettiin kyseinen näkymätön laatikon muotoinen alue ja kun pelaaja astuu tämän laatikon sisään, alkaa animaatiokohtausta välittömästi.



Kuva 5. Sequencer työkalu. Punaiset pallot aikajanalla esittävät merkintöjä, jossa on määriteltä kameran liikerata ja tarkennus kohtausten aikana.

5.5 Pelimekaniikat

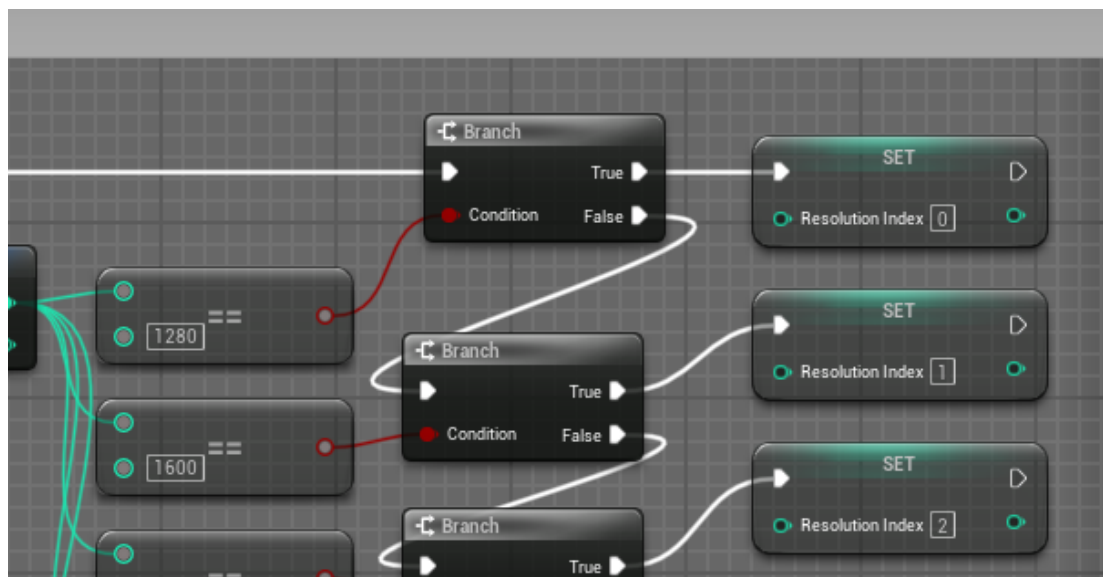
Pelimekaniikat kuvastavat pelaajan suorittamia vuorovaikutuksia peliympäristön ja esineiden kanssa, jotka vaikuttavat pelitilaan. Nämä mekaniikat toteutetaan pelimoottorin avulla, ja niiden avulla määritellään säännöt ja metodit pelikokemukseen, mahdollistaen pelaajalle esimerkiksi voiton tai tappion pelissä.

Unreal Engine 5 hyödyntää peliprojektissa käyttämäämme Blueprint-menetelmää, joka on visuaalinen koodaustapa. Tämä poistaa tarpeen oppia Unreal Enginen C++ ohjelmointikieltä pelin kehittämiseksi, sillä voit hyödyntää aloittelijaystävällistä Blueprint-menetelmää, joka on tavallaan kuin Plug & Play. Kuvassa 6. esitetään esimerkki pelissä käytetystä Unreal Enginen Blueprint-koodauksesta. Korostimme kommenttien käyttöä mahdollisimman paljon, jotta kaikki koodia käyttävät ymmärtävät koodin toiminnan. Kommentointi helpottaa

jatkokehitystäkin, kun mahdolliset uudet kehittäjät pääsevät helpommin sisälle koodin logiikkaan.

Vaikka Unreal Engine 5 sisältääkin valmiina perusmekaniikkoja, kuten kävelmisen ja hyppimisen, peliin kehitettiin lisäksi useita mekaniikkoja, kuten ovien avaaminen, juokseminen ja uiminen. Esimerkiksi uimismekaniikan sisällyttäminen peliin oli tietoinen päätös, sillä halusimme tarjota pelaajille mahdollisuuden kokea uimisen suomalaisessa mökkiympäristössä. Emme kuitenkaan halunneet sisällyttää peliin liiallista määrää mekaniikkoja, sillä tavoitteenamme oli ohjata pelaajan mielenkiinto enemmän pelin tarinaan, kuin uusien mekaniikojen testailuun.

Uudet mekaniikat voivat silti tuoda uniikkia ja mielenkiintoista pelikokemusta, mutta uusien mekaniikojen valtavan määrän tuonti peliin ei ole paras tapa tuoda syvyyttä ja monimutkaisuutta. (Lockhart, 2016.)



Kuva 6. Asetetaan pelaajan valitsema resoluutio asetustalvikosta koodaamalla tapahtumakaavioon.

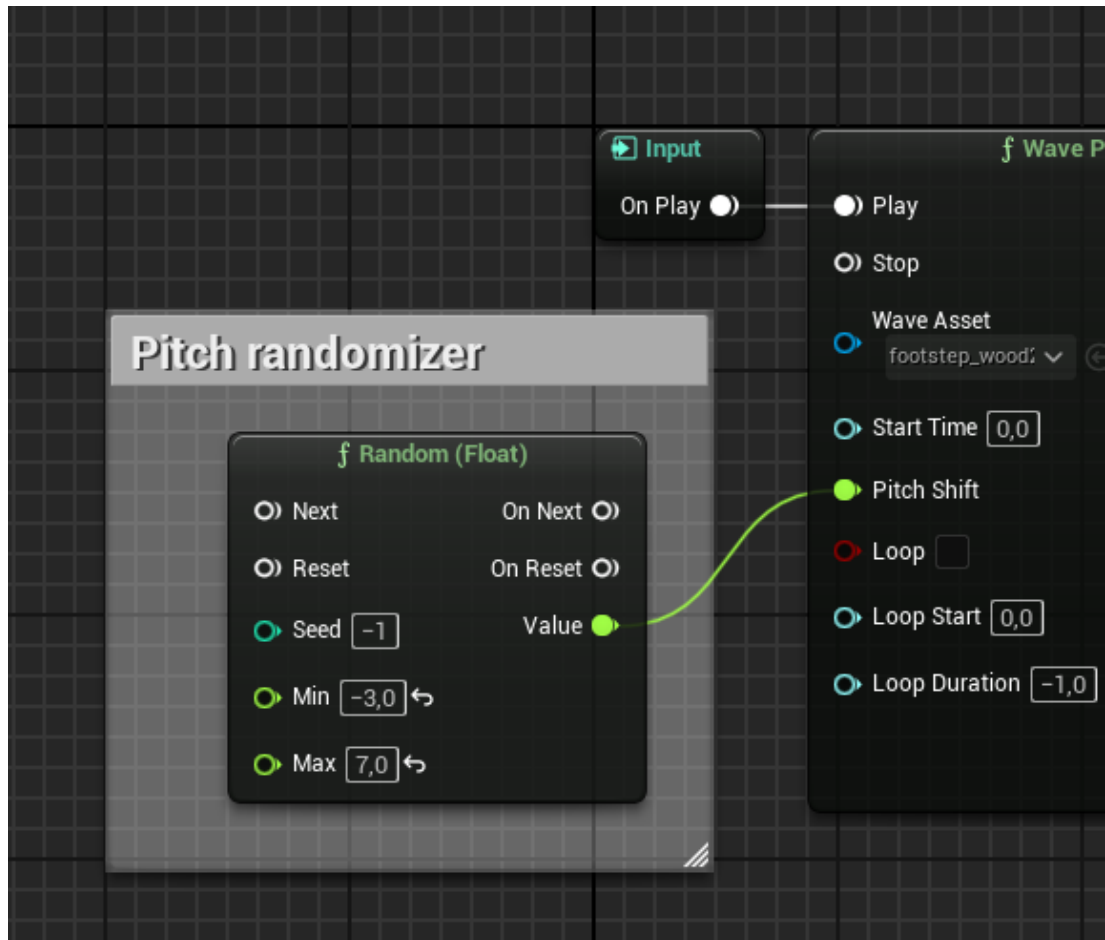
5.6 Äänet

Oikein hyödynnetyillä äänitehosteilla voidaan luoda peliin eloisuutta ja lisätä vuorovaikutteisuutta. Esimerkiksi kun pelaaja putoaa korkealta ja kuulee

äänimerkin jalan murtumisesta, hän tiedostaa ottaneensa vahinkoa ja pyrkii välttämään vastaavaa tilannetta tulevaisuudessa. Kun suunnittelee ääniä, täytyy muistaa, että halutaan luoda joukko muunnelmia, erityisesti toistuville äänille, kuten askeleille. (Lyon, n.d.)

Peliin integroitiin askeläänet eri alustoille, kuten puulle, kalliolle ja irtokiville. Askeleet piti asettaa manuaalisesti sekä kävely- että juoksuanimaatioille, jotta ääni käynnistyy aina kun hahmo astuu. Tämä lisää peliin vuorovaikutteisuutta, antaen pelaajalle vihjeitä siitä, millaisella alustalla hahmo liikkuu. Lisäksi jokaiselle askeleelle muutettiin äänen sävelkorkeutta satunnaisesti, jotta äänen-toisto olisi monipuolisempaa ja toistumista voitaisiin välttää. Kuvassa 7. esitetään, kuinka sävelkorkeuksia muutetaan satunnaisiksi. Lisäksi taustääniksi lisättiin esimerkiksi rauhallisia lintujen ääniä, jotka tuovat pelin teemaa esiin, sekä luovat halutunlaista tunnelmaa peliin.

Unreal Engineissä ympäristön äänien luonti on vaivatonta. Äänitiedoston voi raahata kansioista pelimaailmaan, jonka jälkeen pelimoottori tekee automaattisesti sille alueen, jossa se kuuluu. Aluetta pystyy myös itse laajentamaan isontamalla äänen kuuluvuuden aluetta. Metsästä kuuluviin ääniin lisättiin toistuvuus, jotta äänitehosteet alkavat aina uudelleen niiden ääniraidan loputtua.



Kuva 7. Sävelkorkeudet askeläänille asetetaan satunnaisesti -3 ja 7 välillä.

5.7 Ajoneuvot

Lisäsimme peliin ajoneuvoja, joilla pelaaja voi kulkea paikasta toiseen. Ajoneuvon tuominen peliin on vaivatonta, sillä löysimme ilmaisen assetin Epic Games Marketplace:sta, jonka pystyi tuomaan pelimoottorin kautta peliin. Tämä sisälsi muutaman ajoneuvon luurangot, physics assetit ja skeletal meshit.

5.7.1 Ajoneuvon ajettavuus

Ajoneuvon lisäämisen jälkeen voi käyttää paketin mukana tulleita skeletal mesh:jä ja lisätä ne peliin, jotta autot saadaan näkyviin pelissä. Halusimme kuitenkin ajoneuvot ajettaviksi pelissä, joten ensimmäisenä pelin laajennuksista piti laittaa kokeellinen ChaosVehiclePlugin päälle. ChaosVehiclePlugin on Unreal Enginen kevyt järjestelmä, jolla suoritetaan ajoneuvojen fysiikoiden

simulaatioita. Tämän jälkeen ajoneuvolle tehdään oma blueprint, jonka kuuluu olla mallia WheeledVehiclePawn. Tämän blueprintin avulla voimme asettaa haluamamme pelaajan kameran kulman ja etäisyyden autoon, kun pelaaja astuu autoon sisään. Tämän lisäksi voimme asettaa auton väännön, eli sen, kuinka nopeasti auto kiihtyy. Lisäksi projektin asetuksista piti lisätä näppäimet ajoneuvon kaasulle, jarrulle ja ohjaukselle. Kaasulle määritimme näppäimistön W-napin ja jarrulle S-napin, sillä koimme ne luonnolliseksi normaalin liikkumisen kannalta, jossa liikkuminen tapahtuu näppäimistön W, A, S ja D nappuloita painamalla.

Tämän jälkeen teimme kaksi uutta blueprintiä auton etu- ja takarenkaille, joiden kuuluu olla mallia ChaosVehicleWheel. Eturenkaiden blueprintin sisältä muutettiin eturenkaiden olevan vuorovaikuttaisia auton jarrujen, moottorin, ohjauksen ja mahdollisen käsijarrun kanssa. Takarenkaat ovat vuorovaikuttaisia ohjausta vaille samojen asioiden kanssa, sillä emme halua takarenkaiden vaihtavan suuntaa käännettäessä autoa. Tämän jälkeen ajoneuvon blueprintissä voidaan asettaa renkaat käyttämään kehittämiämme ChaosVehicleWheel:n asetuksia.

Ajoneuvo oli tässä kohtaa ajovalmis, mutta se täytyi vielä koodata ajamista varten. Ajoneuvon tapahtumakaaviossa pystyy nyt tehdä toiminnot ajamiselle, jarruttamiselle, käsijarrulle ja ohjaukselle, sillä olimme aiemmin jo asettaneet syötöille näppäimet projektin asetuksissa.

5.7.2 Ajoneuvon sisään- ja ulostulo

Nyt auto on ajettava, mutta tarvitsemme vielä toiminnon, jolla pelaaja pääsee autoon sisälle. Tarvitsimme näppäimen päästääksemme pelaajan auton sisään ja ulos. Tarvitsimme myös hitboxin, jonka sisällä pelaaja pääsee autoon sisälle, jotta autoon ei pääse, ellei ole auton vierellä. Napin painalluksen aikana auto niin sanotusti vallataan, eli pelaaja jättää pelihahmon ja alkaa pelaamaan autolla. Tämän takia pelihahmo piti asettaa siksi aikaa näkymättömäksi, kun pelaaja pelaa auton ja asettaa taas näkyviin, kun pelataan

pelihahmona. Kun pelaaja lopettaa autoilun ja hahmo astuu autosta pois, halutaan hahmo saada karttaan auton vierelle, eikä sinne mistä autolla lähdettiin ajamaan ja jossa hahmo asetettiin näkymättömäksi. Tämä tehtiin auton tapahtumakaavion sisällä, jossa visuaalisesti asetetaan paikka, kuinka pitkälle autosta hahmo halutaan syntyvän ja koodilla kerrotaan pelin etsivän se paikka kartalta. Kuvassa 8. havainnollistetaan, kuinka hahmon paikka asetettiin.



Kuva 8. Asetetaan paikka, johon hahmo syntyy, kun pelaaja astuu ulos autosta. Lisäksi kuvassa näkyvä suorakulmio auton ympärillä on pelaajalle näkymätön hitbox, jonka alueella autoon voi E-näppäintä painamalla päästä sisään.

5.8 Pelin käyttöliittymä

Pelin käyttöliittymä tarjoaa pelaajalle tarvittavat tiedot ja valikot, kuten asetukset, päävalikot, pelin elämät, ajat ja pisteet. Käyttöliittymän päätehtävänä on avustaa pelaajaa suorittamaan toimintoja pelissä, kuten esimerkiksi painamaan oikeaa nappia, jolla saadaan ovi auki. Kuvassa 9. esimerkki pelissä käytetystä käyttöliittymästä.

Käyttöliittymä on pelin aikana näkymätön kerros kuvan päällä, joista voi lukea tarvittavat tiedot. Hyvin suunniteltu käyttöliittymä parantaa pelikokemusta tarjoamalla selkeyttä ja helppoa navigointia, mikä puolestaan antaa pelaajille mahdollisuuden keskittyä pelaamiseen sen sijaan, että he joutuisivat hämmentäviin käyttöliittymiin jumiin. (Gupta, 2023.)



Kuva 9. Esimerkki pelin käyttöliittymästä, kun pelaaja lähestyy ovea.

Sisällytimme peliin esimerkiksi päävalikon, mistä pääsee aloittamaan pelin, lopettamaan pelin ja muokkaamaan pelin asetuksia. Pelin asetuksista voi säätää pelin grafiikka-asetuksia, resoluutiota, näytön tilaa ja Vsyncin tilaa. Grafiikka-asetuksista voi valita vähemmän resursseja vaativat tai graafisesti paremmat ja resursseja vaativammat asetukset, jotta peli pyörii sulavasti ja tietokoneen komponentit eivät kärsi. Tämä tuo myös mahdollisuuden pelata peliä halvemmilla tietokoneilla.

Resoluution valinta määrittelee näytön pikselitiheyden, eli suurempi resoluutio tarjoaa yksityiskohtaisempaa kuvaa. Näytön tila voi olla joko kokonäyttötila, ikkunatila, tai kokonäytöllinen ikkunatila.

- Kokonäyttötilalla (Fullscreen mode) peli kattaa koko ruudun, minimoi viiveen ja piilottaa kaikki muut käynnissä olevat sovellukset.

- Ikkunatilassa (Windowed mode) peli jakaa näyttötilan kaikkien käynnissä olevien sovellusten kanssa, joten viive lisääntyy, mutta voit käyttää muita sovelluksia pelin ohella.

- Kokonäytöllinen ikkunatila (Fullscreen window mode) on näiden kahden välinen hybridi, eli peli on koko näytön kokoinen ilman reunoja, mutta se jakaa silti näyttötilaa muiden sovellusten kesken.

Vsync on tekniikka, jolla pyritään estämään näytön repeytymistä, eli Screen Tearingia. Näytön repeämistä tapahtuu, kun näytön päivitys ei ole synkronoitu näytön virkistystaajuuden kanssa. Näiden asetusten luominen Unreal Enginen avulla oli vaivatonta, sillä grafiikka-asetukset, näytön tila ja Vsync olivat valmiiksi sisäänrakennettu pelimoottoriin, ja ne kutsuttiin Blueprint-metodilla. Resoluutio oli ainoa, jonka joutui manuaalisesti koodaamaan peliin. Lisäksi peliin lisättiin näkymätön kerros, joka tulee näkyviin pelaajan lähestyessä vuorovai-kutteista objektia.

Kun peli aloitetaan ja päätehtäväketju alkaa, ilmestyy pelaajalle ruudun vasempaan ylälaitaan ohjeet tehtävän jatkamiseksi. Opinnäytetyön aikana ohjeet olivat ainoastaan teksti, joka muuttuu sitä myötä, kun pelaaja saavuttaa seuraavan etapin tehtävässä. Tämä tullaan tyyllittelemään opinnäytetyön jälkeen ja siihen lisätään häivytysoanimaatio.

5.9 Päätehtäväketju

Päätehtäväketjulla tarkoitetaan pelin tarinankulkua pelatessa. Kun peliä aloitetaan pelaamaan, on siinä jokin selkeä päätavoite, joka suoritetaan askel askeleelta. Päätehtävän rinnalla maailmassa on myös sivutehtäviä, joita pelaaja voi suorittaa. Sivutehtävät eivät vaikuta päätarinan kulkuun millään lailla. Nämä tehtävät tuovat peliin lisää pelattavaa ja luovat syvyyttä pelimaailmaan.

Peli ja sen tarina alkoi opinnäytetyöprojektin aikana pelihahmo Jaakon kesämökiltä. Päätehtävä ei opinnäytetyöaikana valmistunut haluamaksemme, sillä cut scene -animaatioiden ja sivuhahmojen ääniraitojen teko koitui liian työlääksi prosessiksi. Ensimmäiseen päätehtäväketjun tehtävään kuului löytää L-Market kauppa, jossa pelaaja juttelee kauppiaan kanssa. Tämän jälkeen Jaakko menee vaihtamaan kuulumiset naapurilleen, joka löytyy muutaman sadan metrin päästä kaupasta.

Ensimmäiseen tehtävään rakennettiin myös pienimuotoinen cut scene, joka laukeaa sillä hetkellä, kun pelaaja saapuu kauppaan sisään. Tämä on havainnoitu kappaleessa 5.4. Tähän cut sceneen on tarkoituksena lisätä ensimmäiset pelin hahmojen dialogit, eli ääniraidat.

6 JATKOKEHITYS

Peli jäi opinnäytetyön jälkeen tilanteeseen, jossa se on toimiva, mutta alun perin suunniteltuja asioita jäi paljon tekemättä. Projektisuunnitelman olisi voinut laatia realistisemmin.

Pelin kehitys kaksin on hidasta, sillä aiempaa kokemusta kummallakaan ei ollut runsaasti. Tästä huolimatta halu saada peli valmiiksi on pysynyt vahvana, joten olemme ajatelleet jatkaa pelin kehitystä opinnäytetyöprojektin jälkeen. Pelin kehitys jatkuu harrastemielessä useamman henkilön voimin.

Peliin tullaan jatkamaan itse päätarinaa, sekä tekemään lisää erilaisia sivutehtäviä tuodakseen lisää sisältöä pelaajalle. Marjastaminen ja saunominen tullaan toteuttamaan ensimmäisinä sivutehtävinä peliin. Myös kalastus tullaan lisäämään tulevaisuudessa yhdeksi sivutehtäväksi. Peliin tehtiin valmiiksi jo vene tätä varten pelihahmon rannalle. Pelihahmon kesämökki tullaan rakentamaan alusta alkaen uudelleen, sillä opinnäytetyön aikana käyttämä valmis rakennus ei muistuttanut suomalaista kesämökkiä tarpeeksi.

Pelin päätehtäväketjuun tullaan myös rakentamaan pelaajan valintojen mukainen tehtäväketju, eli pelaajan valinnat vaikuttavat sen kulkuun ja lopputulokseen.

Peliä tullaan optimoimaan entistä sulavammaksi ja toimivammaksi. Erityisesti liikkumisen tönkköys ja pelin suorituskyky havaittiin kehityskohteeksi.

Peliin tullaan lisäämään taustamusiikkia ja ääniä, kuten esimerkiksi äänet uimiselle, saunan syttymiselle, juoman juomiselle ja niin edelleen.

Peliin tullaan luomaan lisää Cut Scenejä tuomaan tarinaan yksityiskohtia. Peli tulee alkamaan pienen animaatiokohtauksen saattelemana, jossa pelihahmo saapuu kesämökille ja jonka jälkeen päätarinaketju alkaa.

Päätehtäväketjun käyttöliittymään tullaan lisäämään nuoli, johon suuntaan pelaajan tarvitsee mennä tavoittaakseen seuraavan etapin nykyisessä

tehtävässä. Opinnäytetyön aikana käyttämä teksti ei tuo tarpeeksi yksityiskoh-
taisia ohjeita pelaajalle. Maailman kartta tullaan myös rakentamaan tulevai-
suudessa, jotta voimme auttaa pelaajaa hahmottamaan paremmin missä seu-
raava tehtävä sijaitsee, sekä havainnoimaan missä pelaaja sillä hetkellä liik-
kuu.

Peliprojektille tullaan myös luomaan esittelyvideo markkinointimielessä, sitten
kun peli on edustavassa vaiheessa. Tähänkin Unreal Engine tarjoaa mainiot
työkalut Sequencer ominaisuuden avulla.

Peliin tullaan lisäämään hahmoille kasvojen eleitä, kuten juttelemisen aikana
liikkuvat suut. NPC –hahmoille tullaan lisäämään liikkeitä ja animaatioita, joita
ne tekevät silloin, kun pelaaja ei ole heidän kanssaan kanssakäynnissä. Opin-
näytetyön aikana NPC-hahmot seisoivat vain paikoillaan.

Opinnäytetyöprojektin aikana pelihahmoilla ei ollut ääniraitoja lainkaan. Näi-
den äänittäminen koitui työlääksi prosessiksi, joten se jätettiin projektin myö-
häisemmäksi vaiheeksi, emmekä kokeneet sitä tarvittavana ominaisuutena
näytettäväksi opinnäytetyön aikana.

7 LOPUKSI

Opinnäytetyöprojekti oli meille molemmille erittäin opettava työ. Opinnäytetyön aikana opimme paljon uusia asioita, jo aiemman Let's Make a Game -kurssin asioiden ohelle. Edellä mainitusta kurssista oli meille suuri hyöty tässä opinnäytetyöprojektissa, sillä Unreal Engine pelimoottoria ei tarvinnut opetella käyttämään alusta asti. Tämän ansiosta pystyimme keskittymään syvällisemmin uusien asioiden oppimiseen ja kehittämään peliämme tehokkaammin.

Parannettavia asioita voisi luetella monia, kuten alkuperäinen suunnitelma opinnäytetyön kulusta ja kehitettävistä asioista suunnitellulla aikataululla. Opinnäytetyön ajaksi olisi pitänyt sopia myös jokaviikkoiset palaverit opinnäytetyön tiimoilta. Kommunikointi tapahtui enemmänkin satunnaisten viestien lähettelyinä.

Opinnäytetyöprojekti sai meidät myös innostumaan enemmän pelinkehityksestä alana ja harrastuksena. Tämä projekti vahvisti kiinnostustamme koodaukseen, pelisuunnitteluun ja muihin pelinkehityksen osa-alueisiin. Olemme tämän projektin myötä innokkaita jatkamaan oppimista ja kehittymistä tulevaisuudessa kyseisen aiheen tiimoilta.

Opinnäytetyöprojektin aikana kohtaamistamme epäonnistumisista ja vaikeuksista aiotaan jatkokehityksessä ottaa oppia ja jakaa kokemuksellista tietoa niistä uusille kehittäjille, jotka tulevat työskentelemään projektin parissa.

LÄHTEET

Ashley, C. (N.d) What Are Texture Maps and Why Do They Matter For 3D Fashion? Haettu 16.03.2024 osoitteesta <https://www.vntana.com/blog/what-are-texture-maps-and-why-do-they-matter-for-3d-fashion/>

Barrier, R. (17.05.2022). What Unreal Engine 5 Mean for the game industry? <https://nordic.ign.com/hold-the-matrix-awakens-an-unreal-engine-5-experience/56403/feature/what-unreal-engine-5-means-for-the-games-industry>

Gupta, A. (19.12.2023) Game UI Design: Improving Video Game Experience. <https://blog.searchmyexpert.com/game-ui-design/>

Lockhart, B. (28.11.2016). How Many Mechanics Should a Game Have? <https://www.gamedeveloper.com/design/how-many-mechanics-should-a-game-have->

Lyon, J. (N.d.). 10 Top Tips for Video Game Sound Effect Design. Haettu 24.04.2023 osoitteesta <https://sound.krotosaudio.com/video-game-sound-effects-tips/>

Miller, M. (13.06.2022). Epic Games Brings Metahumans to Unreal Engine 5 <https://www.broadcastnow.co.uk/tech/epic-games-brings-metahumans-to-unreal-engine-5/5171614.article>

Moss, E. (N.d). What is Unreal Engine? Haettu 2.3.2024 osoitteesta <https://www.bairesdev.com/blog/what-is-unreal-engine/>

Premawardhana, M. (2021). Unreal Engine Landscape Paint Layers <https://madarapremawardana.medium.com/unreal-engine-landscape-paint-layers-f4b4b7000546>

Premawardhana, M. (2023). A long due article: Unreal Engine Nanite for Dummies. <https://madarapremawardana.medium.com/a-long-due-article-unreal-engine-nanite-for-dummies-eb25ca28af03>

Saarenoja, S. (5.4.2022). Uuden sukupolven pelimoottori Unreal Engine 5 julkaistiin virallisesti – Microsoft-studio paljasti uuden teknologianäytteensä. <https://pelaaja.fi/uutiset/uuden-sukupolven-pelimoottori-unreal-engine-5-julkaistiin-virallisesti-microsoft-studio>

Souza, P. (N.d). Lumen in UE5: Let there be light! Haettu 08.04.2024 osoitteesta <https://www.unrealengine.com/en-US/blog/lumen-in-ue5-let-there-be-light>

Statt, N (13.05.2020). Epic Games announces Unreal Engine 5 with stunning PlayStation 5 demo. <https://www.theverge.com/2020/5/13/21256079/epic-unreal-engine-5-playstation-5-demo-next-gen-graphics-release-date>

Thomas, I (04.12.2020) Theme and Game Story <https://wildwinter.medium.com/theme-and-game-story-f5dece050291>

Terra, J (30.8.2023) How to Use GitHub Desktop: A GitHub Desktop Tutorial <https://www.simplilearn.com/how-to-use-github-desktop-tutorial-article>

Unreal Engine ominaisuudet (2022) <https://www.unrealengine.com/en-US/features>