Karelia-ammattikorkeakoulu
Tietojenkäsittely

Audiovisuaalisen maailman luonti kauhupeliin Unreal Enginellä

Peetja Litja

Opinnäytetyö, Marraskuu 2023

www.karelia.fi

Karelia AMMATTIKORKEAKOULU

OPINNÄYTETYÖ
Marraskuu 2023
Tietojenkäsittelyn koulutus

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

Tekijä
Peetja Litja

Nimeke
Audiovisuaalisen maailman luonti kauhupeliin Unreal Enginellä

Tiivistelmä

Tässä opinnäytetyössä on tarkoituksena luoda audiovisuaalinen maailma kauhupeliin Unreal Engine 5 -pelimoottorilla. Opinnäytetyössä käydään läpi muista peleistä löytyviä audiovisuaalisia mekaniikkoja ja niiden pohjalta luodaan omat toiminnallisuudet kauhupeliin.

Opinnäytetyön toteutukset tehtiin Unreal Engine 5:llä hyödyntäen siitä löytyvää blueprint-ohjelmointia. Peliin luotiin useita erilaisia audiovisuaalisia mekaniikkoja, joita löytyy muistakin kauhupeleistä.

Hyödyntämällä muista kauhupeleistä löytyviä audiovisuaalisia mekaniikkoja, saatiin peliin luotua toimivia ja kauhua ylläpitäviä audiovisuaalisia mekaniikkoja. Opinnäytetyön jatkokehitykseen kuuluu tekemättä jääneiden audiovisuaalisten mekaniikkojen luonti samalla hyödyntäen muita Unreal Engine 5:n toiminnallisuuksia.

Kieli
suomi

Sivuja 44
Liitteet 0
Liitesivumäärä 0

Asiasanat
peligrafiikka, peliohjelmointi, videopelit

Karelia
UNIVERSITY OF APPLIED SCIENCES

THESIS
November 2023
Degree Programme in Business Information Technology
Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author
Peetja Litja

Title
Creating an Audiovisual World for a Horror Game with Unreal Engine

Abstract

The aim in this thesis is to develop an audiovisual world for a horror game with Unreal Engine 5 game engine. In this thesis, audiovisual world mechanics found in other games are reviewed, and based on them, own functionalities are created for the horror game.

The implementations in this thesis are done with Unreal engine 5, utilizing its blueprint programming. Several different audiovisual mechanics were created to the game, which can be found in other horror games as well.

By utilizing audiovisual mechanics found in other horror games, audiovisual mechanics that work and maintain horror have been created in the game. The further development based on this thesis includes the creation of other audiovisual mechanics that were not created while utilizing other functionalities of Unreal Engine 5.

Language
Finnish

Pages 44
Appendices 0
Pages of Appendices 0

Keywords
game graphics, game programming, videogames

Sisältö

Sanasto

Assetti Mikä tahansa sisältö Unreal Enginen projektissa (Unreal Engine 2023a).

Blueprint Unreal Enginen visuaalinen ohjelmointimalli (Unreal Engine 2023b).

Jumpscare Äkillinen tapahtuma, mikä säikäyttää pelaajan tai katsojan

Lumen Unreal Engine 5:n dynaaminen valaistus ja heijastusjärjestelmä (Unreal Engine 2023c).

Nanite Virtualisoitu geometriajärjestelmä (Unreal Engine 2023d).

PFS Procedural Foliage Spawner, Unreal Engine 5:n järjestelmä useiden Foliage tyyppisten objektien synnyttämisen (Unreal Engine 2023e).

SMF Static Mesh Foliage, Foliage tyyppi, joka käyttää mesh-instanssia. (Unreal Engine 2023e).

Sound Cue Ääniresurssi, joka kapseloi monimutkaisia äänisuunnittelutehtäviä solmukaavioon. (Unreal Engine 2023g).

TriggerBox Laatikkomuotoinen liipaisin, joka aiheuttaa tapahtuman vuorovaikutuksen syntyessä. (Unreal Engine 2023h).

1 Johdanto

Tämän opinnäytetyön aiheena on audiovisuaalisten kauhupelimekaniikkojen luominen Unreal Engine 5:llä. Projektissa käyn läpi, miten erilaisia kauhupeleihin liittyviä mekaniikkoja toteutetaan Unreal Engine 5:llä. Olen aiemmin opintojeni kautta päässyt työskentelemään Unreal Enginen aimman version kanssa ja tässä opinnäytetyössä pääsenkin paremmin käsiksi Unreal Enginen uudempaan versioon. Olen kiinnostunut peliohjelmoinnista, mistä syystä opinnäytetyössäni käytävät asiat ovat mielestäni mielenkiintoisia ja tärkeitä peliohjelmoinnin kannalta. Kauhupelin audiovisuaalisten mekaniikkojen luonnista Unreal Enginellä ei myöskään ole aiemmin tehty opinnäytetöitä, mistä syystä kyseisestä aiheesta on mielenkiintoista tehdä opinnäytetyö. Kyseisestä aiheesta tehty opinnäytetyö on peliohjelmoinnista kiinnostuneelle lukijalle tärkeä, sillä siitä voi olla hyötyä omissa projekteissa pelikehityksen parissa.

Opinnäytetyössä kartoitan erilaisia audiovisuaalisia mekaniikkoja kauhupeleistä ja opettelen luomaan niitä Unreal Engine 5 -pelimoottorissa. Tarkoituksena on selittää tarkasti, kuinka toteutukset on tehty ja minkä takia kyseiset toteutukset on valittu opinnäytetyötä varten. Tavoitteenani on oppia lisää Unreal Engine 5-pelimoottorin toiminnasta, kauhupelimekaniikoista sekä niiden luomisesta.

Opinnäytetyössä käydään alkuun läpi videopelien grafiikkojen ja äänien historiaa ja kuinka ne ovat kehittyneet. Tämän jälkeen käyn läpi projektiin käytettäviä työvälineitä ja projektin suunnittelua. Näiden jälkeen kerron minkälaisia audiovisuaalisia mekaanikkoja kauhupeleistä saattaa löytyä ja kuinka niitä voidaan tehdä Unreal Engine 5:llä. Lopuksi kerron omat tulokseni ja johtopäätökseni aiheesta sekä opinnäytetyön toteutuksesta.

Alun perin tarkoituksenani oli käydä läpi immersiivistä kokemusta kauhupeleissä, mutta muiden kehittäjien mielenkiinnon loputtua projektiin vaihdoin aiheeni keskittymään audiovisuaalisten mekaniikkojen luomiseen. Tämä aihe oli sopivan lähellä alkuperäistäni ja tämän avulla pääsen oppimaan erilaisten mekaniikkojen luomista Unreal Engine 5:llä, josta voi olla hyötyä tulevaisuudessa pelikehityksessä.

2 Audiovisuaalisten keinojen merkitys pelisuunnittelussa

2.1 Immersio

Immersiivisissä peleissä pelaaja uppoutuu pelimaailmaa, missä erilaisten kokemusten avulla pelaaja tuntee olevansa osa pelihahmoa ja peliä. Tähän vaikuttavia tekijöitä ovat pelin flow, pelissä tapahtuva yhtenäinen ja selkeä edistyminen sekä realistiset grafiikat. Koska pelimaailma kehittyy koko ajan, on tärkeää ylläpitää peleissä olevia stimuloivia kokemuksia. Tämän takia peleihin lisätäänkin aistinvaraisia kokemuksia, joihin sisältyy esimerkiksi lisättyä todellisuutta tai virtuaalitodellisuutta. (Arm.com 2023.)

Nykypäivänä pelikehitys on kasvanut huomattavaa vauhtia, minkä takia pelinkehittäjien tuleekin miettiä vahvasti, millaisia pelejä kannattaa tehdä. Wirtz (2023a) kertoo artikkelissaan immersiivisien pelien saavan pelaajan imeytymään pelimaailmaan niin hyvin, että tunnit tuntuvat minuuteilta. Immersiivisten keinojen avulla pelaajan pelikokemus kasvaa, eikä kiinnostus peliin lopu hetkessä.

Immersiivistä peliä luodessa tulee ottaa huomioon erilaisia immersioon vaikuttavia tekijöitä. Näitä ovat graafinen immersio, audioimmersio, tarinankerronta ja näytteleminen. (Gameace 2021.)

Immersiosta puhuttaessa monet viittaavat pelien grafiikkaan, sillä monesti pelin visuaalinen puoli vetää pelaajaa puoleensa aiheuttaen immersiivisen kokemuksen. Pelissä ei tarvitse olla erittäin realistiset grafiikat immersion luomista varten, vaikka siitä voikin olla hyötyä. Pelien grafiikka voi olla hyvinkin tyyliteltyä, mikä luo pelaajalle täysin uuden maailman, joka vetää pelaajan täysin puoleensa. (Capaldi 2020.)

Virtuaali- ja lisätyn todellisuuden avulla saadaan tuotua peliin graafisia ominaisuuksia minkä avulla pelaaja pystyy tuntemaan olevansa osa pelimaailmaa. Lisätyssä todellisuudessa peliin pystytään skannaamaan kameroiden avulla lähiympäristöä ja sen avulla tuomaan peliin erilaisia esineitä ja asioita. Tämän avulla pelaajalle luodaan tuttu tunne ympäristöstä ja saadaan vahvistettua immersiota. Virtuaalitodellisuudessa pelaaja käyttää hyödyksi VR-

laseja, jotka tuovat pelaajan suoraan virtuaalimaailmaan. Tämän avulla pelaaja näkee vain ympärillään virtuaalimaailman ja on osa sitä. (Gameace 2021.)

Audio- eli ääni-immersiossa käytetään hyödyksi erilaisia ääniä ja musiikkia, jotka luovat pelistä realistisemman ja mielenkiintoisemman. Musiikin avulla luodaan pelille tunnelmaa ja herätetään pelaajan mielenkiitoa ja ajattelutapaa pelin tilannetta kohtaan. Muiden äänien avulla voidaan luoda pelaajalle erilaisia reaktioita pelitilanteesta riippuen. Esimerkiksi äänien voimakkuuksien ja suunnan avulla saadaan pelaaja miettimään, tapahtuuko asia lähellä vai kaukaa ja mistä päin ääni kuuluu. (Gameace 2021.) Tätä voidaan hyödyntää kauhupeleissä erilaisten reaktioiden luomiseen. Kauhupeleissä pelaaja yleensä pakenee tai piilottelee viholliselta ja näiden äänien avulla pystytään aiheuttamaan pelaajalle kauhun tunnetta esimerkiksi jahtaustilanteissa.

Tarinankerronnan ja näyttelemisen avulla pystytään vahvistamaan tai kompensoimaan pelin eri näkökohtia, jotta pelaaja saa pelistä mahdollisimman hyvän kokemuksen. Hyvän tarinan avulla pelaajaa saadaan vedettyä mukaan pelimaailmaan. Aidon ääninäyttelyn kanssa peliin saadaan mukaan tunteita ja tarkoitusta, mikä lisää pelin aitoutta ja immersiota. (GameAce 2021.)

2.2 Videopelien grafiikat ja niiden kehitys

2.2.1 2D-grafiikat

2D-Grafiikat ovat yksinkertaisuudessaan taidetta, joka ei ole 3D-taidetta. Jos grafiikkaa ei ole piirretty tavalla, missä yritetään simuloida 3D-perspektiiviä, se on 2D-taidetta. Tähän kuuluvat esimerkiksi sarjakuvat, taidemaalaukset sekä monet videopelit. Jos maalaukseen tai kuvaan lisätään syvyyttä, se muuttuu kaksiulotteisesta grafiikasta kolmiulotteiseksi. Esimerkiksi pikselitaide on tehty monelta osin 2D:nä. (Wirtz 2023b)

2.2.2 Ascii

ASCII-taide on erilaisia kuvia, jotka ovat tehty käyttämällä kirjaimia, numeroita ja merkkejä, jotka on määritetty American Standard Code for Information

Interchange (ASCII) -tietokonemerkistössä. ASCII grafiikkoja löytyy jo varhaisista peleistä kuten Star Trek (1971) ja Rogue (1980). (Cassel 2018.)

ASCII-grafiikat tulivat käyttöön, kun tietotekniset ja graafiset ominaisuudet olivat rajallisia. Käyttämällä erilaisia kirjaimia, merkkejä ja numeroita pystyttiin luomaan erilaisia visuaalisia elementtejä peliin tekstipohjaisessa muodossa. ASCII-grafiikoilla pystytään luomaan peliin esimerkiksi käyttöliittymiä ja niiden elementtejä, visuaalisia efektejä, latausruutuja tai muuta taidetta. ASCII-grafiikkoilla luotiin myös kokonaisia pelejä. (Sharpcoderblog 2018.)

2.2.3 Pikseligrafiikka

Pikseligrafiikat tulivat käyttöön videopelien alkuaikoina 1970-luvulla. Alkuun pikseligrafiikka oli hyvinkin yksinkertaista ja karkeaa teknisten rajoitusten vuoksi ja sen takia pelaajan mielikuvituksen tulikin täyttää pelissä olevia tyhjiä kohtia. (Griffiths 2018.)

Pikseligrafiikassa esineet ja hahmot esitettiin pieninä neliöinä, joita yhdistämällä saatiin kokonaiskuva kyseisestä hahmosta tai esineestä. Vaikka pikseligrafiikat olivatkin yksinkertaisia, niin nämä pelit mullistavia ja hyvinkin pidettyjä aikoinaan (Anastasia 2023.)

1970-luvulla julkaistiin Pong, missä värit olivat vain mustaa ja valkoista. Pari vuotta myöhemmin vuonna 1972 julkaistiin kuitenkin ensimmäinen pelikonsoli Magnavox Odyssey, jonka mukana peleihin tuli myös rajattu määrä värejä. (Mishcel 2023.)

1980 luvulle mentäessä tuli käyttöön 8-bittiset grafiikat, jotka mahdollistivat hieman suuremman, mutta rajatun määrän värejä pikselöityjen pelien grafiikkoihin. Esimerkiksi Pac-Man, Donkey Kong ja Super Mario Bros olivat tähän aikaan suosittuja pelejä, jotka käyttivät hyödykseen 8-bittisitä grafiikkaa. (Gamestate 2023.)

1980-luvun loppupuolella tuli käyttöön 16-bittiset grafiikat, joita hyödynnettiin pelikonsoleissa. Nämä antoivat mahdollisuuden suurempaan valikoimaan värejä, selkeämpiä animaatioita ja yksityiskohtaisempia pelitaustoja. (Gamestate 2023.) Jotkut pelit yrittivät jopa sekoittaa pikseligrafiikkoja 3D-pelin grafiikkoihin eri tehokkuusalueilla (Griffiths 2018).

2.2.4 2.5D

2.5D-grafiikat ovat nimensä mukaisesti 2D ja 3D-grafiikkojen välimälli. 2.5D-grafiikoissa käytetään 2-ulotteisia elementtejä 3-ulotteisessa maailmassa. 2.5D-grafiikkojen avulla pelaajalle saadaan luotua illuusio 3D-pelistä, vaikka pelissä olevat objektit ja elementit ovat luotu 2D-mallien avulla. Näissä peleissä hyödynnetään eri perspektiivejä illuusioiden luomiseksi. Tämä voi myös toimia toisten päin käyttämällä pelin taustalla 3D-malleja, mutta peli toteutetaan sivulta päin katsottuna eli 2D:nä, jonka avulla luodaan illuusio pelin syvyydestä. (Knight 2021.) 2.5D-peleissä siis käytetään hyödyksi pelien kuvakulmia ja malleja 3D-illuusion luomiseksi.

2.5D-pelejä on ollut lähes arcade-pelien alusta alkaen. Taito nimisen yrityksen kehittämä peli Interceptor oli yksi ensimmäisistä videopeleistä, jotka käyttivät hyödykseen 2.5D-maailmaa. Myöhemmin julkaistuissa peleissä kuten Doom ja Wolfenstein käytettiin hyödykseen 2D-malleja 3D-pelimaailmassa luoden niistä 2.5D-pelejä. (Knight 2021.)

Knight (2021) kertoo artikkelissaan, että 2.5D-pelit luotiin alun perin laitteistorajoituksien takia. Esimerkiksi NES konsolille tehty Legend of Zelda käytti hyödykseen 2D-malleja, mutta siinä pääsi liikkumaan useampaan kuin kahteen suuntaan. 3D-pelien luominen kyseiselle konsolille ei olisi onnistunut.

2.2.5 3D

3D-grafiikat käyttävät hyödykseen kolmiulotteisesti mallinnettuja esineitä, jotka ovat tuotu videopeliin luoden kolmiulotteisen maailman. 3D-grafiikoissa mallit luodaan jokaisesta suunnasta kuvattuna toisin kuin 2D-malleissa kuva luodaan yhdestä suunnasta kuvattuna. (Wiesen 2023.)
3D-grafiikat alkoivat yleistymään videopeleissä 1990 luvun puolivälissä, kun näytönohjaimet alkoivat kehittymään ja Nintendo 64:n sekä Sony Playstationin pelikehittäjät alkoivat luomaan 3D-pelejä. Nintendo 64:lle ensimmäisiä luotuja 3D-pelejä olivat esimerkiksi Super Mario 64, sekä The Legend of Zelda: Ocarina of Time. Playstationille luotiin Gran Turismo niminen autopeli, mikä käytti hyödykseen kokonaan 3D-maailmaa. (Balasubramanian 2023.)

2.3 Videopelien äänien kehitys

Äänillä on ollut suuri vaikutus videopeleihin niiden kehityksen alusta asti. Ensimmäiset äänet videopeleissä olivat yksinkertaisia, vain muutaman erilaisen soinnun avulla tehtyjä. Vaikka äänet eivät olleet monimutkaisia tai ihmeellisiä, ne toivat peliin uuden ulottuvuuden ja tekivät siitä käyttäjälle paljon mielenkiintoisemman. Klassikkovideopeli Pong oli ensimmäinen tietokoneelle tehty videopeli, joka sisälsi ääniä. Pong on tennistyylinen videopeli ja siinä äänet kuuluivat, kun pallo osui pelaajan mailaan, seinään tai kun toinen pelaaja sai pisteen. Vain kolmella erilaisella piippausmaisella äänellä pelille saatiin luotua syvyyttä ja mielenkiintoa. (Scarrat 2018.)

1980-luvulla peleissä alkoi löytymään jo ääniä sekä musiikkia, mutta ne olivat toisiinsa nähden hyvin samankaltaisia. Konsoleiden muistin koon pienuuden takia äänet luotiin tietokonemaisilla soinnuilla, eikä niitä ollut hyvinkään monta erilaista. Näiden ikonisten äänien pohjalta on kuitenkin luotu paljon nykypäivänkin videopelien ääniä. (Scarratt 2018.)

Ensimmäisissä ääniä ja musiikkia sisältävissä peleissä, kuten Space Invaders alettiin hyödyntämään äänikortteja musiikin ja ääniefektien toistamiseen. Space Invaders hyödynsi Texas Instruments SN76477 nimistä äänikorttia, jonka avulla pelille pystyttiin luomaan melua, sekä erilaisia ääniä. Tätä äänikorttia käytettiin myös hyväksi muissa arcade-peleissä. (Askew 2022.)

Ensimmäiset ääniä sisältävät pelit olivat monoäänisiä ja toistuvia. Niissä äänillä ei ollut vielä syvyyttä, mutta ne lisäsivät kuitenkin pelaajalle pelistä saamaa nautintoa. Videopelikonsoli NES, sekä yllä mainittu Space Invaders olivat monoäänisiä. Sega Genesis toimi äänien kanssa lähes samalla tavalla kuin NES, mutta se tuki kuitenkin stereoääntä. Myös Game Boy tuki stereoääntä, mutta se toimi kuulokkeiden avulla. (Weaver 2019.)

1990-luvulla videopelien äänet alkoivat jo kehittyä huomattavasti. Teknologian kehityksen ansiosta peleihin saatiin tehtyä musiikkia oikeiden soittimien avulla.

Foley-äänien ansiosta pelien äänet eivät jääneet vain soittimien ja elektronisten äänien pariin, vaan peleihin saatiin lisättyä kaikenlaisia ääniä oikeasta maailmasta. (Scarrat 2018.)

CD-ROM:lla oli merkittävä vaikutus pelien musiikkeihin ja ääniin. CD-levyille saatiin tallennettua huomattavasti isompi määrä dataa kuin aiempiin muistikortteihin, jonka ansiosta saatiin luotua paljon vaikuttavampia ääniä paremmalla laadulla. Esimerkiksi Playstation käytti CD-levyjä hyödykseen. (Weaver 2019.)

Nykypäivän videopelien äänet ovat jo erittäin laajat. Peleihin lisätyt äänet ovat joka tilanteessa erilaiset ja musiikki voi vaihtua hyvinkin nopeasti tilanteen mukaan, vaikka vain menemällä luolaan sisään tai astumalla toiseen huoneeseen. Monessa pelissä musiikkiin on panostettu niin paljon, että niitä saatetaan kuunnella pelinkin ulkopuolella. Esimerkiksi Undertale (2015) on tunnettu erityisesti musiikistaan.

2.4 Videopelien erilaiset äänet ja niiden merkitys

2.4.1 Foley-äänet

Foley-ääniä löytyy televisio-ohjelmista, elokuvista sekä videopeleistä. Ne ovat jälkikäteen nauhoitettuja ääniä, jotka luodaan tiettyjä hetkiä varten. Foley-äänet eivät ole digitaalisesti tehtyjä ääniä, jotka löytyisivät äänikirjastoista, vaan ne suunnitellaan yksilöllisesti oikeassa maailmassa erilaisten esineiden ja asioiden avulla, jokaista kohtausta varten. Näitä ääniä ovat esimerkiksi kävely, oven avaaminen, vetoketjun avaaminen ja aika lailla kaikki mitä oikeiden asioiden avulla tehdään. (Adobe 2022.)

Foley-ääniä hyödynnetään pääasiassa paljon elokuvissa, mutta niitä voidaan hyödyntää helposti myös videopelien cutsceneissä. Cutscenet ovat kohtauksia videopeleissä, joissa kerrotaan videopelin tarinaa eteenpäin ilman, että pelaaja joutuu tekemään mitään. Nämä cutscenet yleensä täydentävät pelaajalle tarinasta puuttuvia osia, joita ei tule ilmi peliä pelatessa eteenpäin.

Never Alone -pelin tekijä Hogan (2014) kertoo blogissaan, kuinka he loivat erilaisia ääniä peliinsä. He loivat esimerkiksi ketun lumiset jalanäänet hanskan ja paperiliittimien avulla, jotka narisevat lumen jäisen lumen pinnalla. (Hogan 2022.)

2.4.2 SFX

SFX eli äänitehosteet ovat keinotekoisia ääniä, joiden avulla parannetaan pelissä olevaa läsnäolon tunnetta. Näiden avulla luodaan pelissä oleva äänimaailma. Näitä ääniä voivat olla esimerkiksi keskeisessä roolissa olevat toimintaäänet, kuten ampumisäänet ja räjähdykset tai taustaäänet kuten eläinten äänet tai muut luonnonäänet. (Keane 2021.)

Äänitehosteita luodessa ja miettiessä tulee ottaa monia eri tekijöitä huomioon. Keane (2021) mainitsee, että pelkästään miettiessä ääniä veden valumiselle hanasta ulos tulee ottaa huomioon, kuinka korkealta vesi putoaa, millaisella paineella se tulee, mihin vesi osuu sekä kuinka kaukana hana on kuvassa. Äänitehosteet ovat erittäin merkittävässä roolissa videopeleissä ja jos pelaajan mielestä ne eivät ole tarpeeksi uskottavia, ei niistä olla saatu pelille hyötyä. Äänitehosteiden tärkeimpänä merkityksenä on olla pelaajalle tai katsojalle uskottavia ja parantaa pelikokemusta.

2.4.3 MIDI

MIDI tulee sanoista Musical Instrument Digital Interface, joka tarkoittaa soittimien digitaalista käyttöliittymää. Se kehitettiin vuonna 1983, kun ei ollut vielä yleistä standardisoitua keinoa yhdistää erilaisia elektronisia soittimia toisiinsa. MIDI:n avulla ei lähetetä suoraan äänisignaalia, vaan se toimii binaarien avulla, joihin sisällytetään tarpeellinen tieto soittimesta ja sen käytöstä, kuten mikä nuotti on ollut kyseessä, kuinka pitkään sitä on painettu sekä muita sointuun liittyviä tietoja. MIDI tukee 128 nuottia, eli pitkälti kaikkia mahdollisia nuotteja. (Wregleswort 2022.) MIDI:n avulla pystytään siis lähettämään soittimesta digitaalista tietoa toiselle laitteelle, esimerkiksi tietokoneelle ja soittamaan kyseiset soinnut siellä.

Kaikki soittimesta saatu tieto voidaan tallentaa MIDI-tiedostoon (.mid). Suurin osa pelimoottoreista tukee MIDI-tiedostoja, jonka ansiosta musiikin lisääminen ja käyttäminen kyseisissä peleissä on helppoa. Pelimoottorit hyödyntävät MIDI-tiedostosta saatavaa dataa, jonka perusteella pelin tai pelimoottorin oma äänigeneraattori toistaa musiikkia. Koska MIDI-tiedostot ovat pieniä, niitä voidaan hyödyntää paljon peleissä ilman, että pelin koko muuttuu suureksi. Näin pelistä ei tule turhan isoa ladattavaa tiedostoa, eikä se käy raskaaksi laitteistolle. (de Arteaga 2018.)

Esimerkiksi Rust (2013) videopelissä on pelin sisällä paljon erilaisia soittimia, joita pelaaja pystyy soittamaan hieman samalla tavalla kuin oikeita soittimia, kuitenkin näppäimistön avulla. Käyttämällä MIDI-tiedostoja pelaaja pystyy kuitenkin toistamaan oikeita kappaleita ilman, että pelaajan tarvitsee soittaa soitinta näppäimistöllä. Peli toistaa halutun kappaleen automaattisesti, kunhan se on vain pelin kautta kytketty päälle ja pelaaja voi nauttia musiikista. Myös jotkin klassikkopelit, kuten Final Fantasy ja Legend of Zelda käyttivät musiikeissaan hyödyksi MIDI-tiedostoja (de Arteaga 2018).

2.4.4 Monoääni

Monoääni käyttää äänen toistamisessa vain yhtä äänikanavaa. Tämä tarkoittaa sitä, että jokainen kaiutin saa saman äänisignaalin, jolloin ne toistavat samaa ääntä. Monoääntä on hyödyllistä käyttää silloin, kun käytössä on paljon kaiuttimia isolla alueella, jotka suuntaavat eri suuntiin. Tällöin jokainen kaiutin toistaa samaa äänisignaalia eikä se aiheuta äänentoiston ongelmia. (Haroosh 2022.)

Videopeleissä monoääntä voidaan käyttää esimerkiksi silloin, kun ei tarvitse miettiä syvyyksiä tai mistä suunnasta äänet kuuluvat. Nämä ovat pääasiassa vanhoja 2D-videopelejä. Nykypäivänä suurin osa peleistä käyttää stereoääntä tai jopa tilaääntä.

2.4.5 Stereoääni

Stereoääni tarkoittaa kahden kanavan kautta tulevaa ääntä. Stereoäänen avulla kaksi eri kaiutinta toistavan äänen sen perusteella tuleeko ääni vasemmalta vai oikealta. Stereoäänen avulla pystytään luomaan suuntaavuuden tunnetta ja simuloimaan oikean tilan äänimaailmaa. Lähes kaikki nykypäivän laitteista tukevat stereoääntä. (Haroosh 2022)

3D-peleissä stereoääni on varsinkin hyödyllinen. Sen avulla pelaaja pystyy kuulemaan mistä suunnasta mikäkin ääni tulee. Esimerkiksi ensimmäisen persoonan räiskintäpeleissä on erittäin tarpeellista tietää mistä suunnasta vihollisen askeleet kuuluvat. Ilman stereoääntä näiden pelien pelattavuus olisi paljon huonompaa, eikä vihollisen ääniä pystyttäisi hyödyntämään millään tavalla. Esimerkiksi Counter-Strike: Global Offensive (2012) ja Fortnite (2017) käyttävät stereoääntä.

2.4.6 Tilaääni (Surround sound)

Tilaääni tarkoittaa ääntä, joka tulee jokaisesta suunnasta. Tilaääneen sisältyy myös äänen monimuotoisuus, eli vahva basso samaan aikaan kun toisesta suunnasta kuuluu muita ääniefektejä. Tilaääni käyttää hyödykseen vähintään 5 kaiutinta sekä subwooferia eli alaäänikaiutinta. Tämä on nimitykseltään 5.1 tilaääni. Tilaäänilaitteistossa jokaisella kaiuttimella on oma merkityksensä ja ne tulee sijoittaa oikein. Tilaäänilaitteistoon kuuluu keskikaiutin, satelliittikaiuttimet eli oikealla ja vasemmalla puolella olevat kaiuttimet sekä subwoofer. (Casey 2021.)

Tilaäänen voi olla erittäin hyödyllinen esimerkiksi elokuvien katseluun, jolloin katsoja saadaan immersoitua vahvasti elokuvaan. Videopeleissä tilaääni ei ole vielä niin hyödyllinen kuin se voisi olla. Suurin osa kuulokkeista ovat stereokuulokkeita, mutta jotkin uusimmista kuulokkeista tukevat myös 5.1 tai 7.1 tilaääntä. Coltorti (2022) kertoo julkaisussaan, että tilaäänestä voi olla joskus pelaajille kilpailullista haittaa. Vaikka tilaäänen tulisikin kertoa tarkasti mistä suunnasta mikäkin ääni tulee, kaikki pelit eivät kuitenkaan tue tätä ominaisuutta kunnolla. Monet pelit ovat suunniteltuja vielä stereokuulokkeille, jolloin

tilaäänikuulokkeiden käyttö voi johtaa valheellisen tiedon saamiseen. (Coltorti 2022.)

Kilpailullisissa peleissä, kuten aiemmin mainitussa Counter-Strike: Global Offensive:ssa äänet ovat erittäin merkityksellisiä. Jos tilaääni antaa kyseisessä pelissä valheellista tietoa, se voi helposti johtaa kierroksen tai jopa koko pelin häviämiseen. Tästä syystä kaikissa peleissä ei vielä ole hyödyllistä käyttää tilaääntä. Tilaäänen käyttämiseenkin tarvitaan oikeanlaiset kuulokkeet tai kaiuttimet, mitä ei ole vielä kaikilla käytössä.

2.4.7 3D-äänet

3D-äänet ovat vielä yksi askel eteenpäin tilaäänistä. Kun tilaäänet toistavat ääntä oikean ja vasemman lisäksi myös edestä ja takaa niin 3D-äänet lisäävät tähän vielä korkeuden, eli äänet kuuluvat myös ylhäältä sekä alhaalta. 3D-äänien avulla pystytään luomaan illuusio äänien tarkasta sijainnista kolmiulotteisessa tilassa. Tämän ansiosta pelaajalle pystytään luomaan täysi immersio peliin. (Dolby Games 2021.)

3D-äänet eivät ole vielä kovinkaan yleisiä, mutta esimerkiksi Sonyn videopelikonsoli Playstation 5 on ottanut 3D-äänet käyttöön. Sony on luonut AMD:n kanssa yhteistyössä Tempest Enginen, jonka avulla PS5 käsittelee kaikki 3D-äänet. Sonyn mukaan jopa yksittäiset sadepisarat pystyy paikantamaan heidän 3D-äänien avulla. Heidän äänimoottorinsa ansiosta käyttäjän ei myöskään tarvitse turvautua muiden äänilaitteiden äänentoistoon, sillä Tempest Engine prosessoi äänet suoraan. (Raymond 2022.)

Vaikka uudemmat videopelikonsolit tukevatkin jo 3D-ääniä, eivät ne silti ole jokapäiväistä. Esimerkiksi kotiteatterin avulla 3D-äänien simuloimiseen tarvitaan 7.1 tilaäänikaiutinjärjestelmä sekä 4 virtuaalista kaiutinta ylä- ja alapuolelle, jotta saadaan simuloitua täysi 3D-ympäristö (Dolby Games 2021). 3D-ääniä voidaan ajatella vielä osittain tulevaisuuden ääninä. 3D-äänet tulevat kuitenkin olemaan erittäin iso osa immersion lisäämisessä ja varsinkin yhdistettynä VR-laitteistoon, pystytään pelaajalle luomaan täydellinen immersio.

3 Kauhupelin suunnittelu

3.1 Projektin tausta

Projekti oli alun perin tarkoitus tehdä toisen kehittäjän kanssa (Tomi Kiiski), jotta pelistä olisi tehty läpipelattava lyhyt kauhupeli. Tomin kanssa suunnittelimme pelikentän pohjaa ja sen perusteella aloin luomaan pelikenttää ja sen graafista puolta. Toisen kehittäjän luovuttua projektista, jouduin jatkamaan sitä yksilökehityksenä hieman erilaisesta näkökulmasta.

Projektissa on tarkoitus käydä läpi millaisia ääniä ja graafisia elementtejä käytetään kauhupeleissä ja kokeilla tehdä niitä omaan projektiin Unreal Engine 5:ssä. Projektissa ei siis luoda valmista videopeliä, vaan opetellaan luomaan näitä yleisiä kauhupelien audiovisuaalisia elementtejä. Lopullisena tavoitteena on oppia luomaan Unreal Engine 5:llä erilaisia audiovisuaalisia elementtejä, joita voidaan hyödyntää tulevaisuudessa pelikehityksessä.

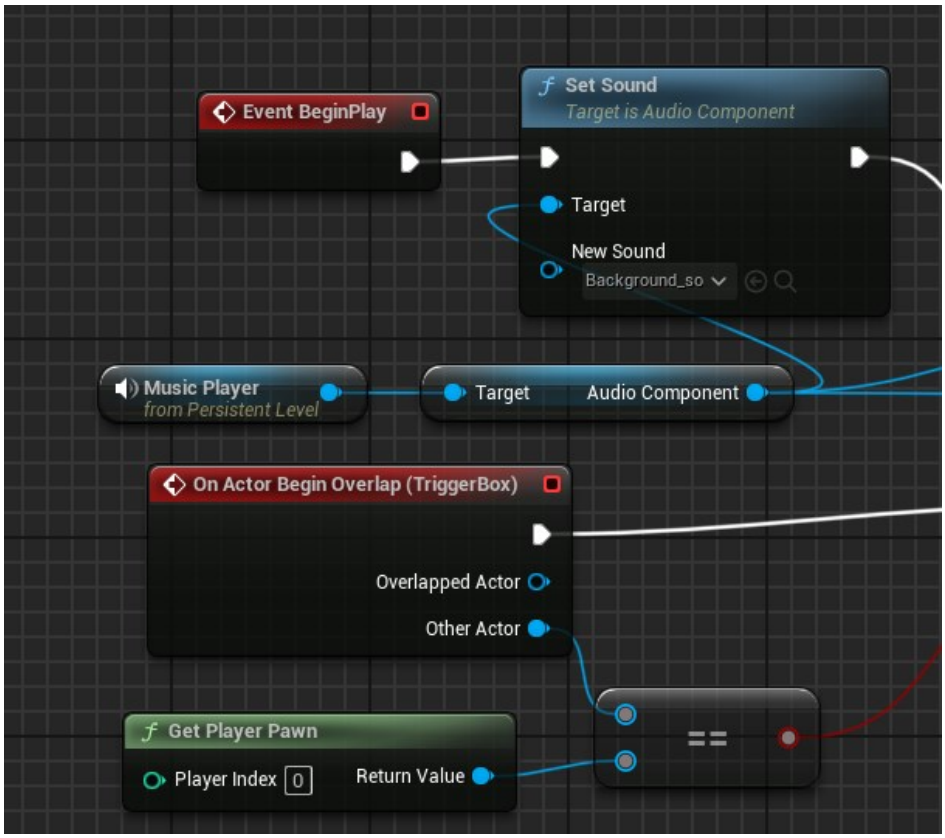
3.2 Pelikentän suunnittelu

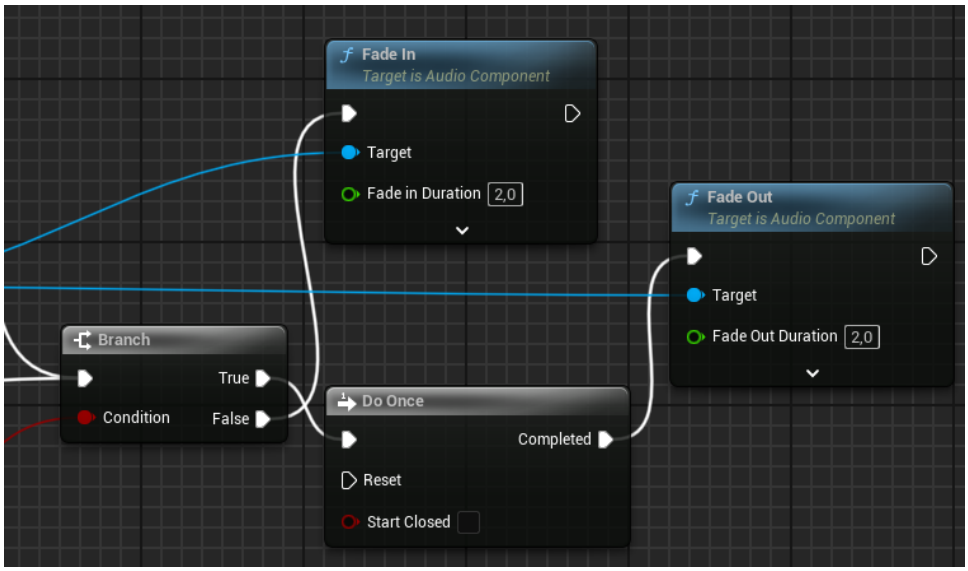
3.2.1 Työvälineet

Tämän opinnäytetyön projekti suunniteltiin ja luotiin Ureal Engine 5 -pelimoottorissa. Unreal Engine 5 on uusin versio Unreal Engine -pelimoottorista, jonka avulla pystytään luomaan suhteellisen aidon näköisiä ja tuntuisia pelejä. Suurina eroina aiempiin versioihin ovat pelimoottoriin päivitetty valaistus Lumen, mikropolygonigeometrijärjestelmä Nanite sekä pelimoottoriin integroitu Quixel Bridge, josta pystyy etsimään ja lataamaan projektiin ilmaisia assetteja. (Unreal Engine 2023i.)

Unreal Engine 5 ei kuitenkaan ole ainut pelimoottori mitä projektia varten olisi voinut käyttää. Unreal Enginen lisäksi löytyy muitakin pelimoottoreita, kuten Unity ja Godot. Unity on oman kokemukseni perusteella hieman helpompi käyttää, mutta kokemukseni Unitylla perustuu vain 2D-pelien tekemiseen. Unreal Enginellä sen sijaan olen työskennellyt pääasiassa 3D-pelien parissa. Godot-pelimoottoria en ole käyttänyt ollenkaan, mistä syystä en ottanut sitä huomioon pelimoottoria valittaessa.

Unreal Enginen ja Unityn välillä on paljon eroja. Unreal Engine käyttää hyödyksi C++ ohjelmointikieltä ja Unity käyttää C# ohjelmointikieltä. Yleisesti ottaen C# on helpompi kieli opetella, mikä on uusille käyttäjille parempi, sillä kieli on hieman helpompi opetella kuin C++. Unity on myös käyttäjäystävällisempi ja sitä on helppo ja selkeä käyttää. Unreal Enginestä löytyy ohjelmointikielen lisäksi Blueprint-mallinen visuaalinen tapa luoda peliin ominaisuuksia, jolloin ohjelmointikieltä ei aina tarvita. (Kuva 1.) (Kaur Arora 2023.)





Kuva 1. Esimerkki Blueprint-ohjelmoinnista 2 osassa. Taustaääni ja sen pysäyttäminen, kun pelaaja astuu tietylle alueelle.

Pelimoottoriksi valitsin Unreal Engine 5:n, sillä olen käyttänyt Unreal Enginen aiempia versiota Unitya enemmän. Unreal Enginellä pystyy luomaan graafisesti hieman aidompia ja näyttävämpiä pelejä, mikä sopii tähän projektiin paremmin. Projektin teko on erittäin aikaa vievää työtä, mistä syystä en halunnut opetella uutta pelimoottoria projektin luontia varten. Tähän olisin joutunut käyttämään huomattavasti enemmän aikaa, jolloin projektin toteutus olisi voinut kärsiä huomattavasti, jonka takia Godot jäi vaihtoehdoista kokonaan pois. Unity jäi pois kokemuksen vähäisyydestä johtuen.

Peliprojekti on ensimmäisen persoonan kauhupeli, jonka visuaalinen ympäristö keskittyy pimeään metsäiseen maisemaan. Projekti tallennetaan GitHubiin, jotta projektin edistymistä on helppo seurata, sekä tarvittaessa palauttaa aiempiin versioihin. GitHubia käytettäessä tulee ottaa huomioon projektissa olevien assettien koot, sillä tarpeeksi isoja tiedostoja ei pysty siirtämään GitHubiin. GitHubin eri versioissa on myös rajallinen määrä tilaa jokaiselle projektille. Projekti tehdään ilmaisia assetteja käyttämällä, joten myös GitHubista käytetään ilmaisversiota, jonka maksimikoko on 2 GB. (GitHub Docs 2023.)

GitHubin lisäksi versionhallinnan vaihtoehtoina olisi voinut olla GitLab tai Azure DevOps. Päädyin kuitenkin käyttämään GitHubia, sillä olen käyttänyt sitä aikaisemmissa projekteissa ja se toimii käsi kädessä GitHub Desktop

sovelluksen kanssa, mikä mahdollistaa helpon versionhallinnan käytön projektille.

3.2.2 Unreal Engine 5

Unreal Engine 5 on Epic Gamesin kehittämän pelimoottorin päivitetty versio. Unreal Engine on täysin ilmainen pelimoottori omia projekteja varten ja sen käytöstä joutuu maksamaan vain, jos projektilla tienaa yli miljoona dollaria. Unreal Engine 5 mahdollistaa reaaliaikaisten 3D-pelien luomisen vapaammin, tarkemmin ja joustavammin kuin koskaan ennen. (Unreal Engine 2023j.)

Unreal Engine 5 keskeisimpinä avainominaisuuksina ovat esimerkiksi Nanite ja Lumen, joiden avulla saadaan luotua yksityiskohtaisempia assetteja ja aidompaa valaistusta. Yhdistämällä näitä uusia uusia ominaisuuksia pystytään luomaan aidompia, immersiivisempiä ja visuaalisesti näyttävämpiä pelejä kuin aiemmilla versioilla. (Unreal Engine 2023j.)

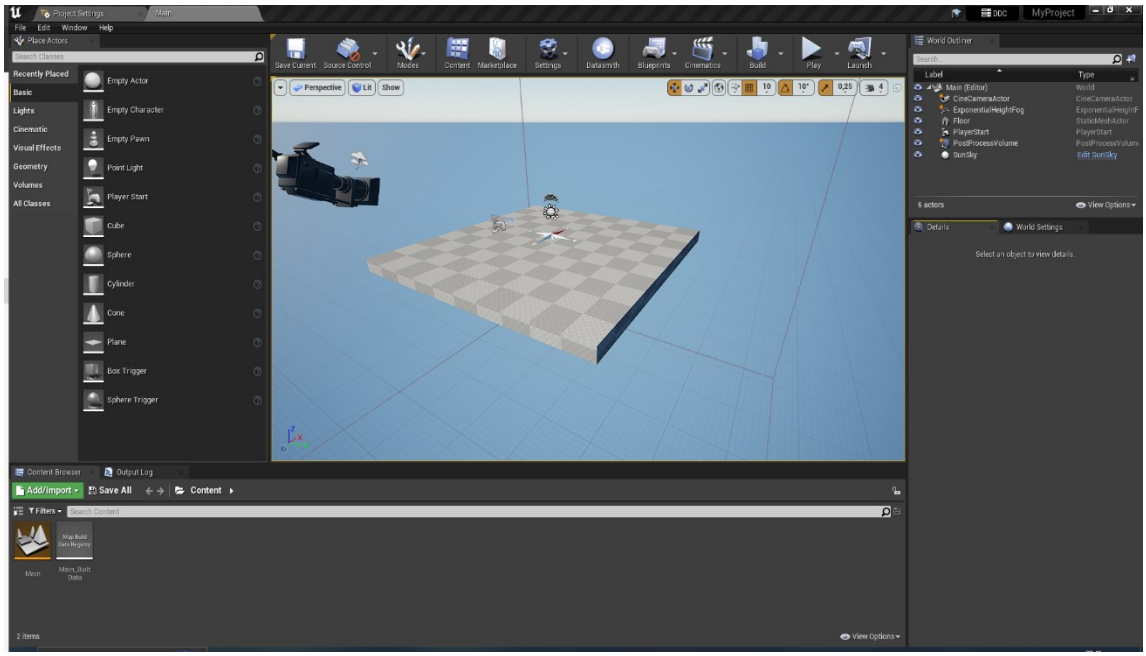
Naniten avulla pystytään luomaan visuaalisesti näyttävämpiä, aidompia ja syvempiä graafisia esineitä ja asioita pelikenttään. Nanite on yksinkertaisuudessaan staattinen verkko, mille on lisätty Nanite ominaisuus. Nanite siis koostuu kolmioista, johon sovelletaan paljon yksityiskohtia ja se pakataan datallisesti tiiviimmin. Naniten avulla esineisiin luodaan paljon yksityiskohtia, mikä lisää näiden esineiden kolmioiden ja verteksien määrää. Tämän voisi olettaa vievän paljon enemmän tilaa, mutta kun verrataan suurien yksityiskohtaisesti tehtyjen esineiden kokoa, niin Nanite avulla näiden datallinen koko huomattavasti pienempi. Esimerkiksi Unreal Enginen sivulla vertaillaan High Poly Static Mesh kiveä 4K kartassa Nanite Mesh kiveen 4K kartassa, joissa molemmissa on käytettä yhtä paljon kolmioita ja verteksejä, niin Nanitella luotu kivi on 7.6 kertaa pienempi (19.64MB), kuin ilman Nanitea luotu kivi (148.95MB). (Unreal Engine 2023d.) Nanite on siis erittäin hyödyllinen luomaan visuaalisesti näyttäviä ja aitoja yksityiskohtia peliin samalla pienentämällä pelin kokoa.

Lumen-valaistuksen avulla saadaan luotua entistä realistisempaa ja muutoksiin reagoivaa valaistusta pelikenttään. Kaikki pelissä tapahtuvat muutokset vaikuttavat reaaliaikaisesti epäsuoraan valaistukseen ja heijastuksiin. Tätä

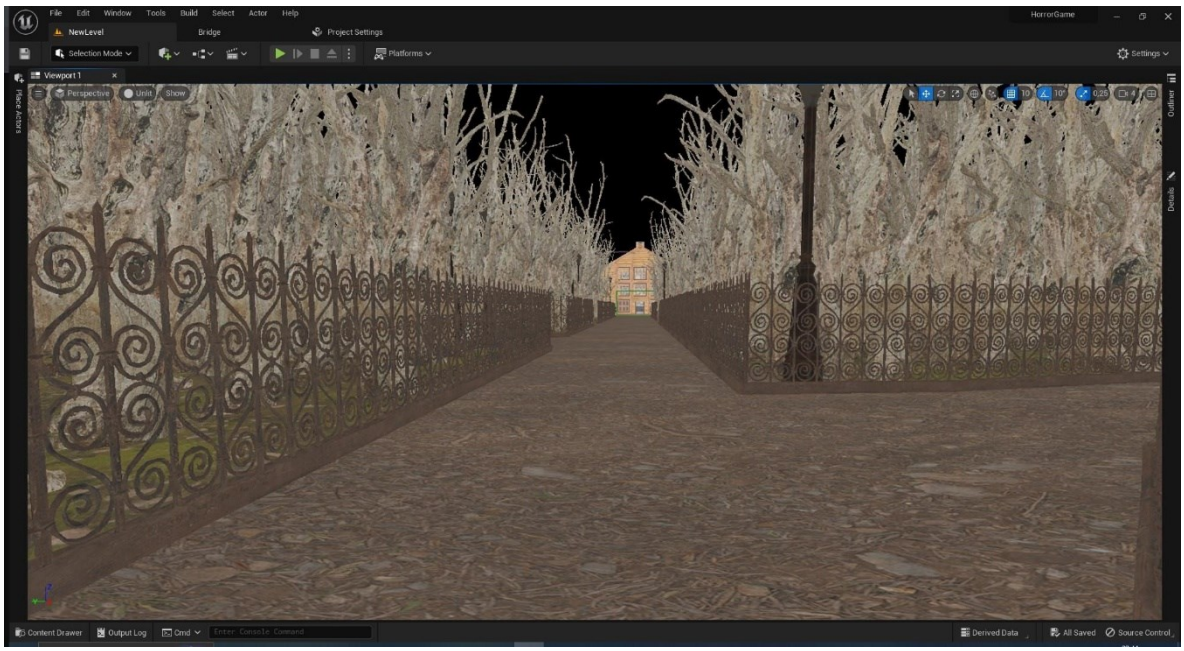
voivat olla esimerkiksi taskulampun käyttö, räjähdys ja oven avaaminen. Lumenin avulla saadaan korostettua pienimpiäkin yksityiskohtia tarkoissa ja monimutkaisissa ympäristöissä. (Unreal Engine 2023c.)

Lumen avulla on tässä projektissa yrittää luoda mahdollisimman realistinen valaistus hämärään ja pimeään pelikenttään, jotta pelaajalle saadaan immersiivinen visuaalinen kokemus. Pelikentässä on paljon valaistusta estäviä puita, aitoja sekä muita esineitä ja rakennuksia, minkä takia Lumen mahdollistaa paremman immersiivisen kokemuksen.

Unreal Enginen uudessa versiossa on myös päivitetty käyttöliittymä mikä on aiempaa verrattuna mielestäni huomattavasti selkeämpi ja silmälle miellyttävämpi. Käyttöliittymän valikot pystytään kiinnittämään Unreal Enginen sivuille ja alas niin, että ne eivät ole koko ajan näkyvillä viemässä tilaa. Käyttöliittymän uusien ominaisuuksien avulla on helpompi navigoida oikeiden työkalujen pariin, silloin kun niitä tarvitaan. Käyttöliittymästä pääsee myös suoraan käsiksi Unreal Engineen integroituun Quixel Bridgeen, josta löytyy paljon ilmaisia assetteja projekteja varten. Tämän avulla on helppo lisätä ja etsiä juuri oikeita assetteja sopivaan kohtaan projektia ilman, että tarvitsee aukaista esimerkiksi Unreal Enginen assetti kauppaa erikseen. (Unreal Engine 2023j.) Alla vielä erot Unreal Engine 5:n ja Unreal Engine 4:n käyttöliittymien välillä. (Kuva 2.) (Kuva 3.)



Kuva 2.Unreal Engine 4:n käyttöliittymä



Kuva 3. Unreal Engine 5:n käyttöliittymä

Unreal Enginestä löytyy myös erittäin hyödyllinen Procedural Foliage Tool, jonka avulla pelikenttään pystytään luomaan esimerkiksi metsiä ilman, että jokaista puuta tai tarvitsisi lisätä manuaalisesti. Tämä ominaisuus täytyy ensiksi laittaa päällä projektin asetuksista, jotta sitä voi käyttää. Kun se on saatu otettua käyttöön, pystytään luomaan Procedural Foliage Spawner, johon liitetään jokin tietty assetti, kuten puu. Tämän jälkeen sille annetaan haluamat arvot ja asetukset ja PFS voidaan lisätä kenttään. Kenttään lisäämisen voidaan päättää

mille alueelle assetteja synnytetään ja näin peliin saadaan luotua esimerkiksi metsä. (Unreal Engine 2023e.)

3.2.3 Assettien etsiminen ja käyttäminen

Suuri osa projektin ja pelikentän suunnittelusta on erilaisten assettien etsimistä tai vaihtoehtoisesti niiden luomista. Tässä projektissa käytetään pelkästään kaikille jaossa olevia ilmaisia assetteja joita löytyy Unreal Enginen 5 omasta kaupasta, Quixelistä, Sketchfabista sekä freesound.org sivulta.

Esimerkiksi pelihahmon taskulamppu, varis sekä joitakin rakennuksia on ladattu Sketchfabista ja otettu käyttöön Unreal Engine 5:ssä. Unreal Engine 5:een integroidusta Quixel Bridgestä olen ottanut käyttöön esimerkiksi puita, puskia ja aitoja. Quixel Bridgestä ladatut assetit ovat siitä hyviä, että ne ovat lähes valmiita käyttöön otettavia assetteja. Ulkoisilta sivuilta ladattaessa assetteja tarvitsee hieman muokata ennen kuin niitä voidaan käyttää pelikentässä. Niille tarvitsee yleensä luoda blueprintit erikseen ja yhdistää materiaalit toisiinsa, jotta ne näyttävät pelikentässä oikealta.

Ääniä etsiessä tulee ottaa huomioon, sopiiko ääni kyseiseen projektiin vai ei. Koska pelissä on tarkoitus luoda kauhupelille sopiva maailma, tulee äänet valita tarkasti. Mikäli äänet eivät ylläpidä kauhupeleissä olevaa pelon tunnetta tai kauhumaista äänimaisemaa, ei niistä ole pelissä hyötyä. Tämän takia yritän löytää esimerkiksi mahdollisimman aidolta kuulostavia eläinten ääniä, pelottavia huutoja sekä aavemaisia taustaääniä. Myös muiden ympäristön äänien tulee kuulostaa aidolta, jotta kauhupelille ominainen äänimaisema ei rikkoudu.

3.2.4 Projektiin valmistautuminen

Projektia varten joudun katsomaan paljon erilasia ohjevideoita Youtubesta, sekä katsomaan Unreal Enginen dokumentaatiota, sillä Unreal Engine 5-pelimoottoriin on päivittynyt paljon aiemmasta versiosta. Suurena apuna on Youtubesta löytyvä Gorka Games Youtube-kanava. Gorka Gamesin tutoriaalien avulla luon peliin esimerkiksi, Procedural Foliage Spawnerin sekä 3D-äänet. (Gorka Games 2023.)

Gorka Games on Youtubesta löytyvä Unreal Engine 5:llä pelikehitykseen keskittyvä Youtube-kanava ja sitä ylläpitää Gorka Aranzabal niminen Indie videopelikehittäjä. Azanbal on Gorka Games Studion ylläpitäjä ja hän on mukana kehittämässä Bromeliad nimistä videopeliä. Azanbal tekee Youtube kanavalleen Unreal Engine 5:n liittyviä ohjevideoita ja hän on luonut myös GameDev.tv:n kanssa kurssin Unreal Engine 5:n liittyen. (Azanbal, G. 2023.)

Kyseiseltä kanavalta löytyy hyvin paljon erilaisia ohjevideoita, joissa käydään läpi Unreal Engine 5:n ominaisuuksia ja toiminnallisuuksia, sekä kuinka niitä pystytään hyödyntämään omissa projekteissa. Tältä kanavalta löytyvät videot ovat selkeitä ja hyvin toteutettuja, mistä syystä käytän näitä videoita projektissani hyödyksi.

Projektia varten joudun opettelemaan paljon uutta, sillä suuri osa projektiin toteutettavista asioista ovat minulle ennestään tuntemattomia, varsinkin uuden Unreal Engine version myötä. Vaikka aikaisempaa kokemusta Unreal Enginen aiemmista versioista onkin, niin joudun silti tutkimaan paljon uuden Unreal Enginen toimintaa, sillä tämän tyylistä projektia en vielä ole tehnyt. Onneksi uudessa Unreal Enginen versiossa on kuitenkin uudet ominaisuudet tehty selkeäksi ymmärtää, mikä nopeuttaa niiden käyttöönottoa ja hyödyntämistä.

3.2.5 Äänimaailma

Äänimaailma tarkoittaa kaikkea sitä ääntä mitä pelikentästä löytyy. Näitä ovat esimerkiksi ambienssiäänet, äänitehosteet, musiikki ja puhe. Ambienssiäänien avulla luodaan ympäristön tunne peliin. Tähän kuuluu esimerkiksi eläinten ja luonnon äänet. Ambienssiäänien avulla yritetään luoda peliin syvyyttä ja realismia, jotta saataisiin synnytettyä mahdollisimman immersiivinen pelikokemus. (Sonic Minds. 2023.)

Projektissa on tarkoitus luoda kauhupelimäinen maailma ja tehdä teemaan sopiva äänimaailma. Tähän kuuluu jännitystä ylläpitävä taustamelu, erilaiset metsässä tapahtuvat äänet, kuten eläinten äänet tai liikehdintä, jumpscaret sekä pelissä olevien hahmojen äänet.

3.2.6 Visuaalinen maailma

Visuaalinen maailma koostuu kaikesta siitä, mitä pelaaja näkee pelissä. Oli se sitten pelihahmon kädet, pelimaailman ympäristö tai käyttöliittymä. Ilman vakuuttavaa visuaalista maailmaa ei pelille pystytä luomaan toimivaa immersiota, eikä pelille tule tarpeeksi kauhua synnyttävää tunnelmaa. Visuaalisen maailman tulee olla näyttävä, pelin genreen ja tyyliin sopiva sekä yhtenäinen. tähän jonkin lähde yms

Tähän peliin on tarkoitus luoda mahdollisimman vakuuttava maisema käyttäen ilmaisia assetteja, koska en osaa kunnolla tehdä graafisesti vakuuttavia assetteja.

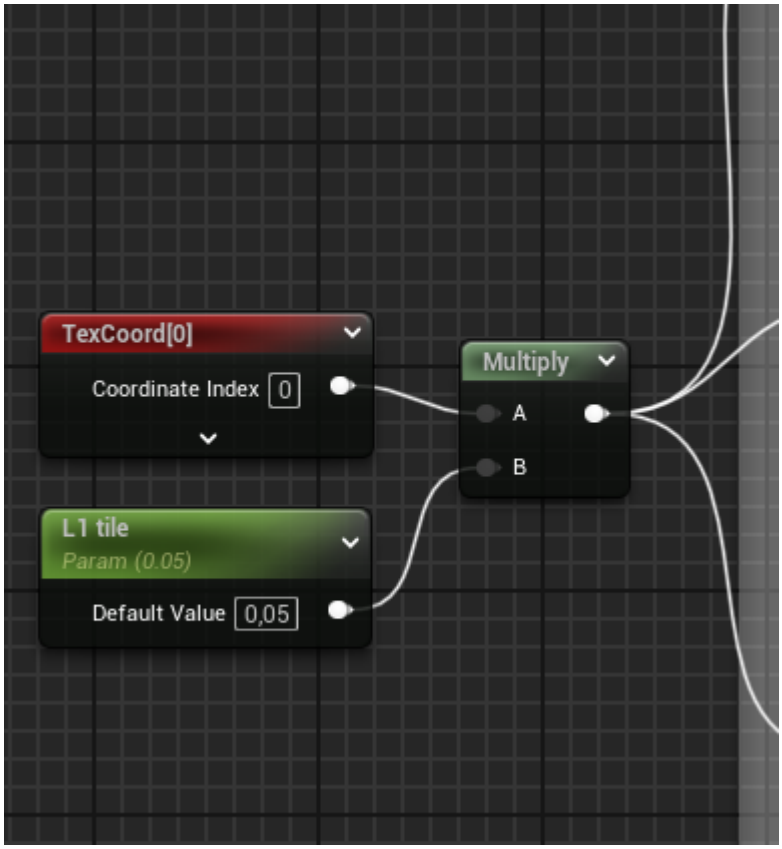
4 Kauhupeliprojektin toteutus

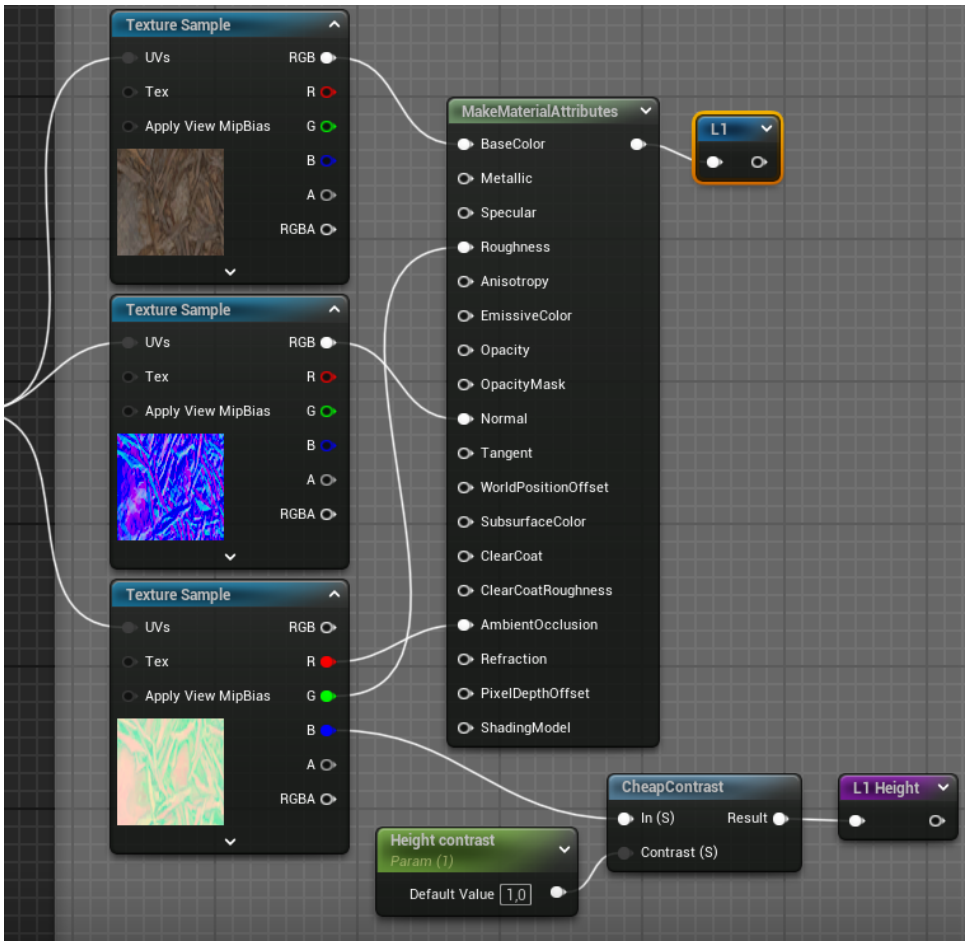
4.1 Pelikentän mallintaminen ja toteutus

Pelikenttä aloitettiin luomalla Unreal Enginen uusi projekti ja tekemällä sinne tyhjä kenttä. Tyhjän kenttään lisättiin pohja maailmalle (Landscape), taivas (SkyAtmosphere) sekä valaistus taivaalta (SkyLight). Tämän jälkeen kenttää pystyttiin alkamaan muokkaamaan halutun laiseksi.

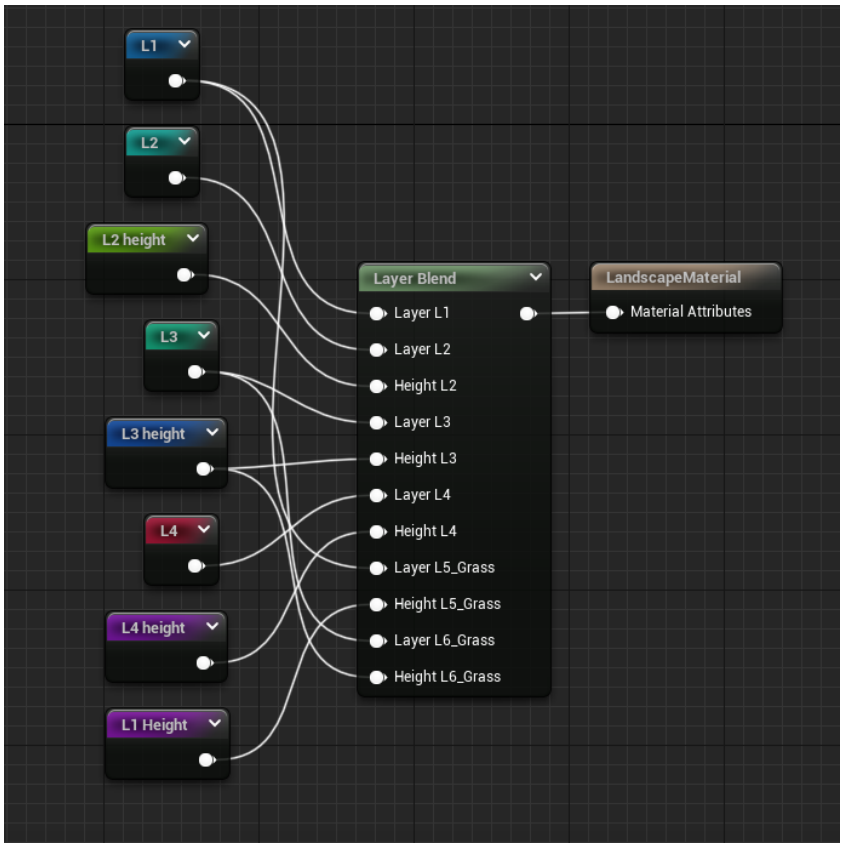
Peliprojektiin tuotiin useita eri assetteja ja materiaaleja Quixel Bridgestä, jotta sitä voidaan alkaa muokkaamaan paremmin. Latasin esimerkiksi useita erilaisia maaperiä, jotta pelimaailma ei näyttäisi kaikkialta samalta. Käyttämällä Unreal Enginen Landscape Modesta löytyvää Paint-ominaisuutta pystyin muokkaamaan kentän eri osia erilaisilla maaperillä, jotta kaikki ei näyttäisi samalta. Tätä varten loin 6 erilaista kerrosta, joiden avulla pelikenttään pystyy maalaamaan omia alueita erilaisilla materiaaleilla. (Kuva 4.) Näistä neljä on tavallisia materiaaleja, mutta kahteen niistä lisäsin pientä kasvillisuutta, jotta maaperä ei näyttäisi niin tasaiselta ja yksinkertaiselta. Nämä kerrokset yhdistin LandscapeMateriaaliin (kuva 5), jonka jälkeen niitä pystytään käyttämään Landscape Modessa (Kuva 6). Kasvillisuutta vaativia kerroksia varten jouduin luomaan myös niille erilliset Landscape Grass Type assetit, joihin lisäsin

ruohoa. Tämän jälkeen kentän maaperän muokkaaminen on eri kerroksilla maalaamista ja suunnittelua, kunnes siitä tulee halutun näköinen.

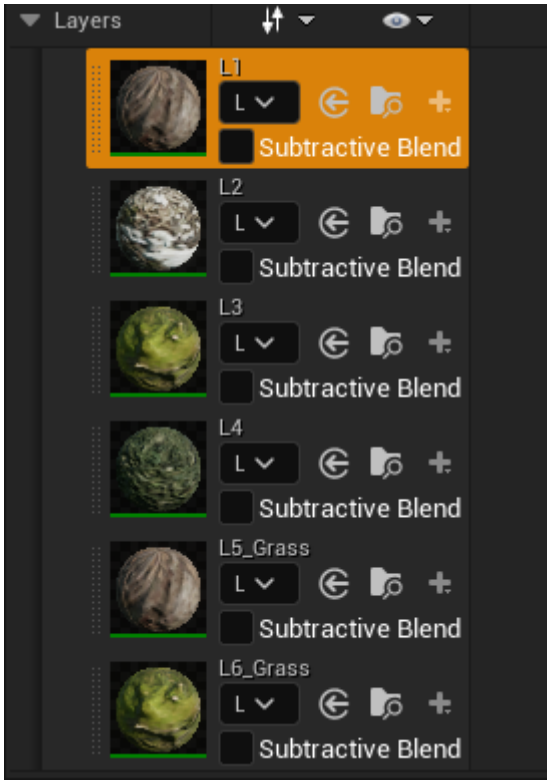




Kuva 4. Esimerkki Landscape-materiaalin kerroksen luonnista 2 osassa.

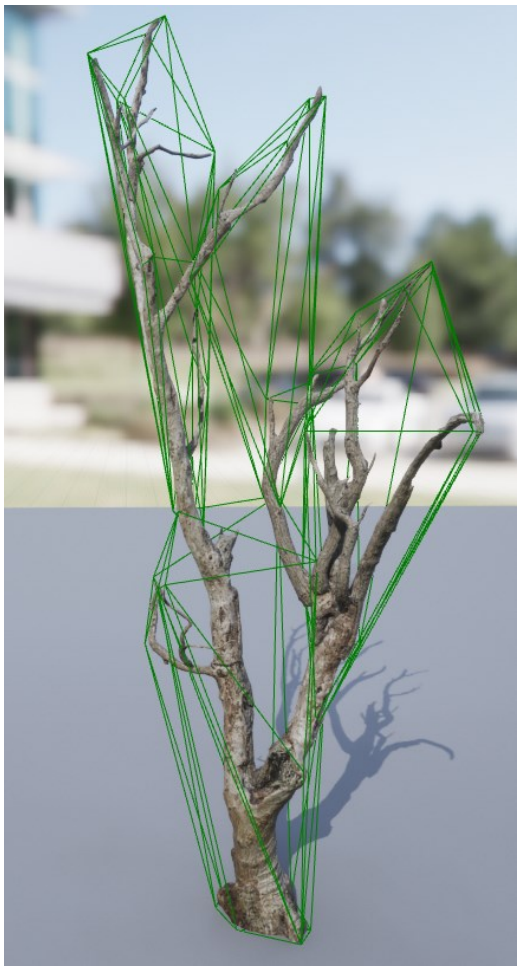


Kuva 5. Kerrosten yhdistäminen Landscape Materiaaliin.

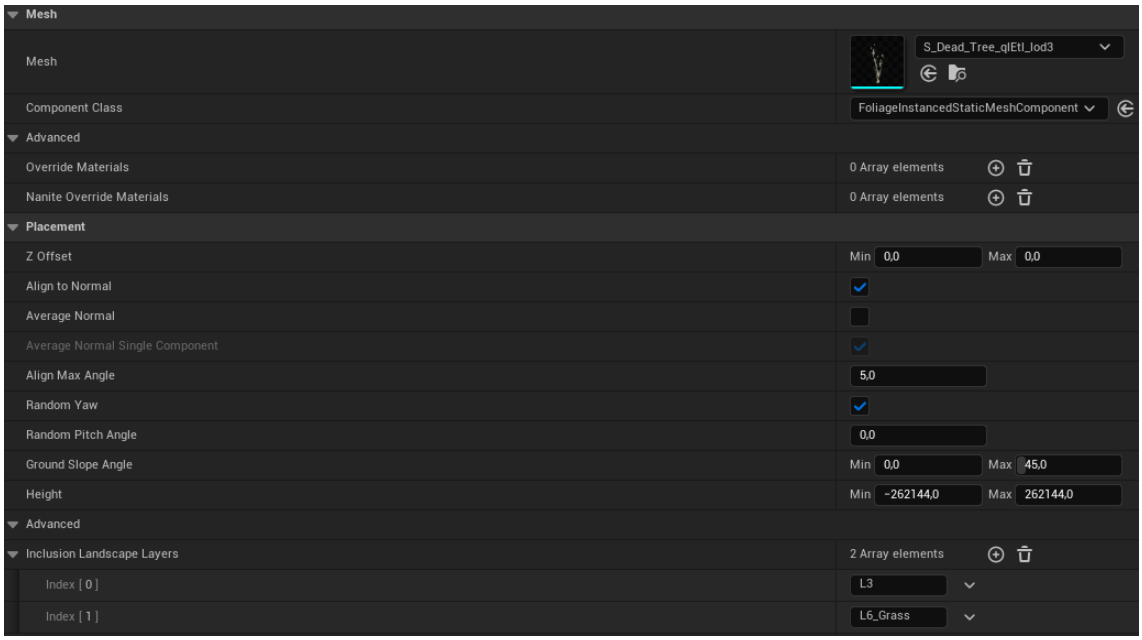


Kuva 6. Eri kerrokset Landscape Mode-editorissa.

Kenttää varten loin myös metsän käyttämällä hyödyksi Unreal Engine 5:n Procedural Foliage työkalua. Tätä varten hankin ensiksi assetin kuollutta puuta varten Quixel Bridgestä ja loin sille tarvittavat colliderit eli fyysiset ominaisuudet, johon pelaaja voi törmätä (Kuva 7). Tämän jälkeen lisäsin peliin Static Mesh Foliagen (SMF), johon liitin kyseisen puun (Kuva 8). Tämän avulla peliin pystyttiin luomaan Procedural Foliage Spawner (PFS), jolle annettiin halutut arvot metsän tiheyttä ja kokoa varten (Kuva 9). PFS:ään tuli myös liittää aiemmin luotu SMF, jotta PFS osaa luoda metsän kenttään määritetylle alueelle. SMF:än asetuksia säätämällä päätin mille maaperille kyseistä puuta synnytetään ja pystyykö puut kasvamaan epätasaisessakin maastossa suorassa. Tämän jälkeen PFS lisättiin suoraan kenttään ja sen koko määritettiin halutuksi, tässä tapauksessa koko kentän kokoiseksi ja generoitiin pelikenttään metsä (Kuva 10). Lisäsin pelikenttään myös pari rakennusta, aitoja, lamppuja sekä hautausmaan.



Kuva 7. Kuollut puu ja sen colliderit.



Kuva 8. SMF:n asetukset

Kuva 9. PFS:n asetukset

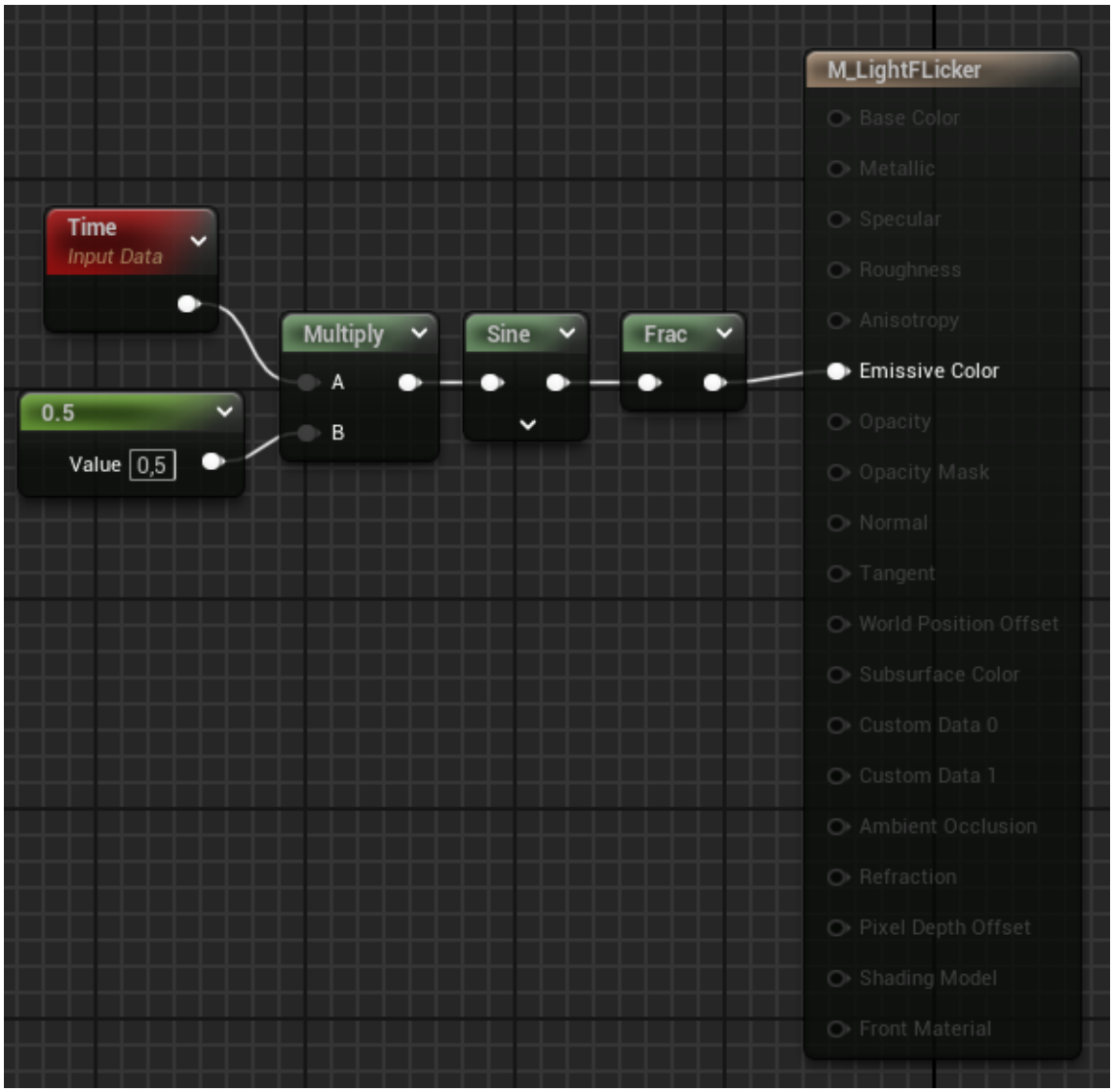
Kuva 10. Pelikenttä ja siihen luotu metsä

4.2 Audiovisuaalisten mekaaniikkojen luonti

4.2.1 Lamppuun liittyvät mekaaniikat

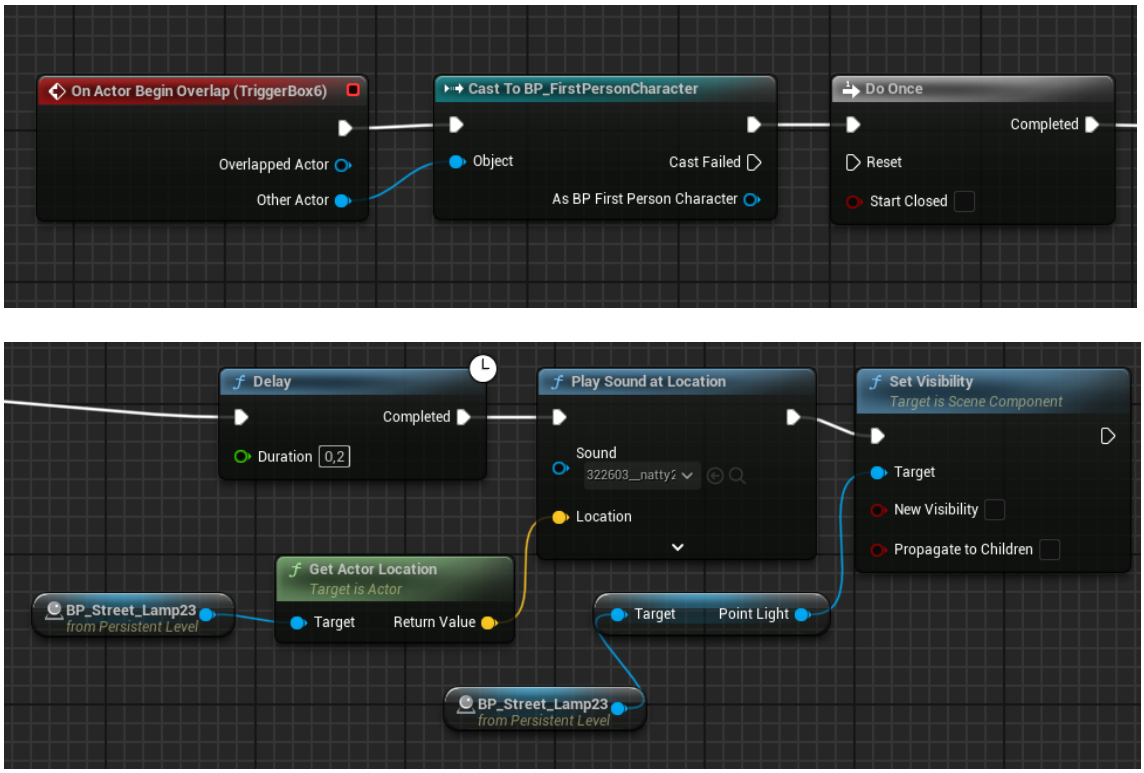
Kauhupeleissä on tyypillisesti pimeä ympäristö ja siellä olevat valot eivät aina tuo paljoakaan valoa. Projektiin on lisätty useita lamppuja, jotka tuovat kenttään vähän valoa, mutta tein osalle niistä myös muutoksia, jotta ne hajoavat tai muuten vain välkkyvät, ikään kuin olisivat viallisia.

Välkkyvän valon luonti oli yksinkertaista. Sitä varten loin uuden materiaalin ja vaihdoin sen materiaalialueen valofunktioksi. Blueprinttien avulla annoin materiaalille ajallisen arvon, kerroin sen vakiolla, joka oli tässä tapauksessa 0.5:llä ja yhdistin sen sine ja frac solmuun. Tämä yhdistettiin materiaalissa olevaan emissiiviseen väriin ja materiaali on valmis (Kuva 11). Vakiota muokkaamalla pystyn säätämään, kuinka nopeasti valo välkkyy ja tässä tilanteessa 0.5 oli sille juuri sopiva arvo. Tämän jälkeen materiaali piti yhdistää pelissä olevaan valoon ja se alkoi välkkymään.



Kuva 11. Välkkyvän valon materiaalin Blueprint.

Peliin tein myös valon, joka hajoaa, kun pelaaja astuu tietylle alueelle. Tämän tein TriggerBoxin avulla. Blueprinttien avulla Triggerbox aktivoituu vain, kun pelaaja astuu sen alueelle. Tämä yhdistetään Do Once solmuun, jonka ansiosta mekaniikka tapahtuu ainoastaan kerran, eikä sitä voi toistaa, vaikka alueelle astuisi uudelleen. Tämän jälkeen tulee pieni viive, lampun valo katoaa ja kuuluu lampun lasin särkymisen ääni. Tämä on yhdistetty pelikentässä olevaan yhteen lamppuun, mikä näkyy Blueprinteissä nimellä BP_Street_Lamp23. (Kuva 12.) Näin saatiin luotua katulamppu, mikä särkyy.



Kuva 12. Hajoavan lampun Blueprint 2 osassa

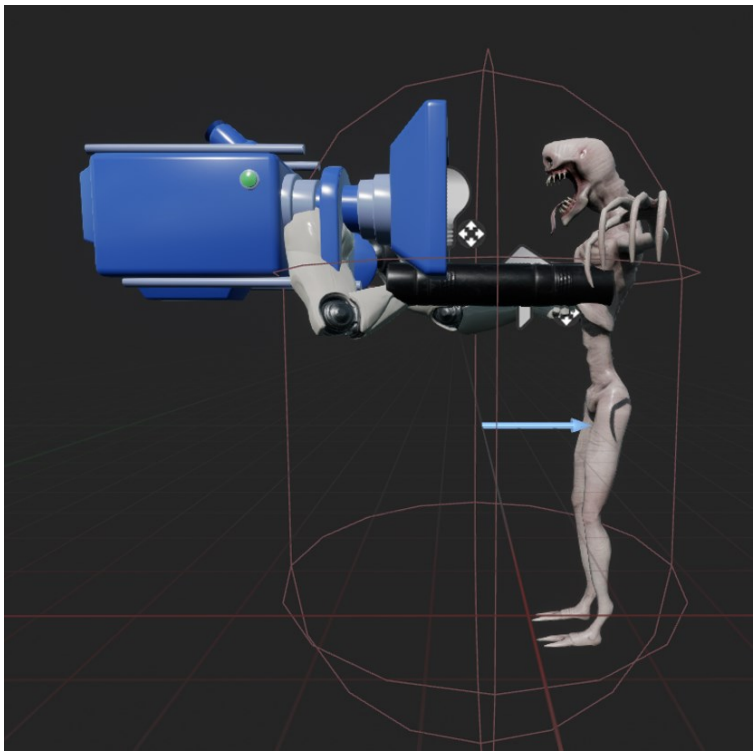
4.2.2 Jumpscaren luonti

Jumpscareja löytyy lähes jokaisesta kauhupelistä tai kauhuelokuvasta. Ne ovat tyypillisiä ja niiden avulla saadaan pelaaja tai katsoja säikähtämään hyvinkin helposti. Padilla (2023.) mainitseekin artikkelissaan tämän olevan kauhugenressä yksi vanhimmista taktiikoista säikäyttää pelaaja tai katsoja. Tästä syystä tein myös tähän projektiin jumpscaren, jossa näytölle ilmestyy hahmo, eikä peliä pysty enää sen jälkeen jatkamaan.

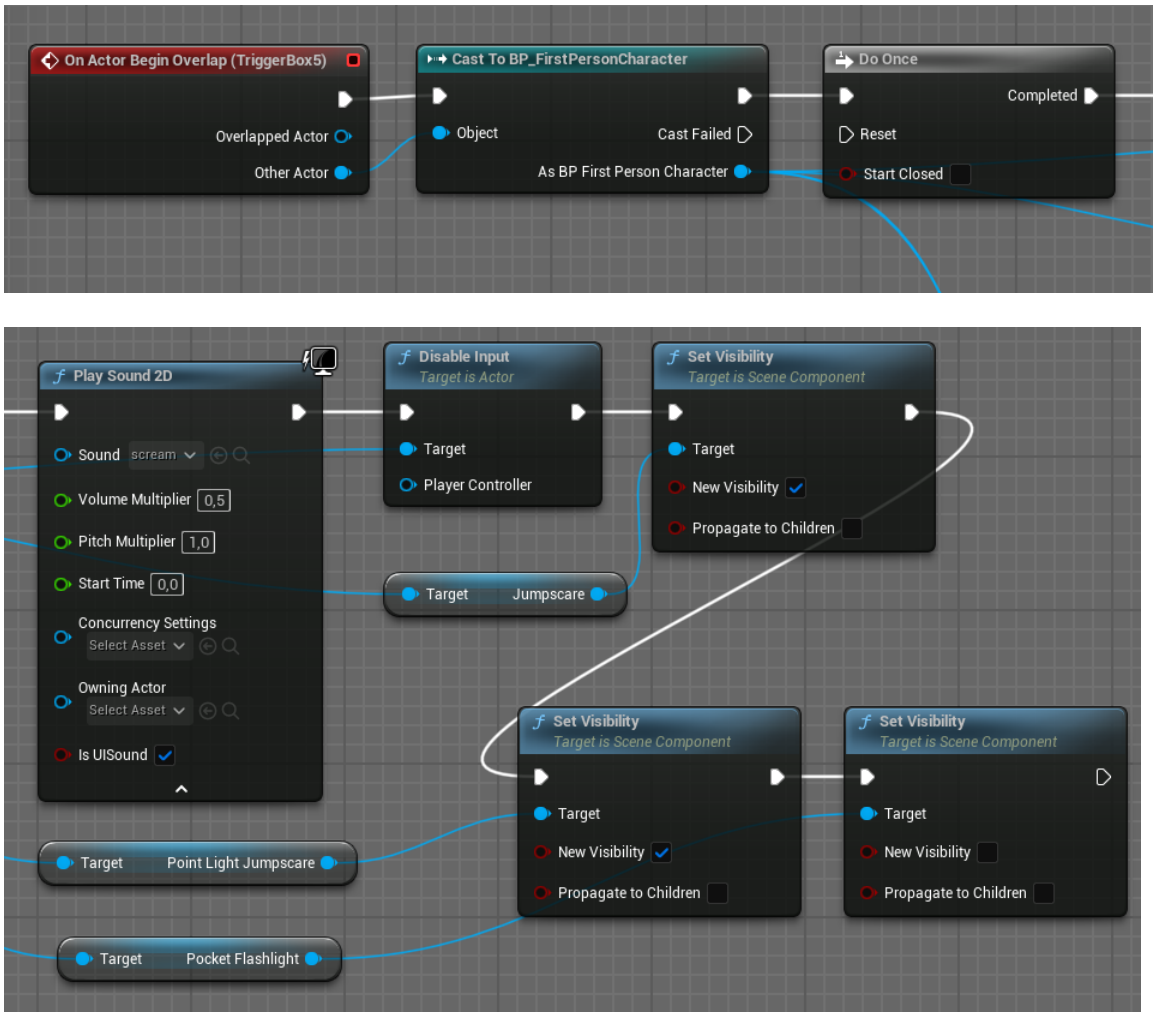
Jumpscare tapahtuu, kun pelaaja astuu pelikentässä tietylle alueelle, mikä on etukäteen määritelty. Tämä alue on lähellä pelikentässä olevaa vihollista ja siinä pelaajan näytölle ilmestyy vihollinen ja samalla kuuluu todella kovaääninen huuto. Pelaaja menettää tässä hahmon hallinnan ja peli loppuu. Pelihahmon blueprintissä lisäsin pelihahmon kameran eteen vihollisen ja tein siitä oletuksena näkymättömän. Alla olevassa kuvassa näkyy, miten vihollinen on asetettu kameran eteen. Kyseisessä kuvassa vihollinen on näkyvissä, mutta tavallisesti se on näkymätön (Kuva 13). Tämä hahmo on siis koko ajan kameran

edessä, mutta se tulee näkyviin vain, kun pelaaja astuu kentällä olevaan Triggerboxiin.

Toiminnallisuus tälle jumpscarelle tehtiin pelikentän Blueprintissä. Se tapahtuu, kun pelihahmo astuu TriggerBoxiin. Do Once:n avulla tämä tapahtuu vain kerran, eikä jää toistamaan esimerkiksi ääntä useaan otteeseen. Disable Inputin avulla pelihahmoa ei voi enää liikuttaa, peli siis loppuu jumpscaren tapahtuessa. Set Visibilityn avulla vihollinen saadaan näkyviin valoisana ja myös samanlaisten silmujen avulla voidaan myös ottaa pois päältä esimerkiksi taskulamppu, jotta ylimääräistä valoa ei tule tähän tapahtumaan. (Kuva 14.) Jumpscaren tapahtuessa pelaajan näytölle ilmestyy vihollishahmo ja peli loppuu (Kuva 15).



Kuva 13. Jumpscare hahmo lisättynä pelihahmon Blueprinttiin viewportissa.



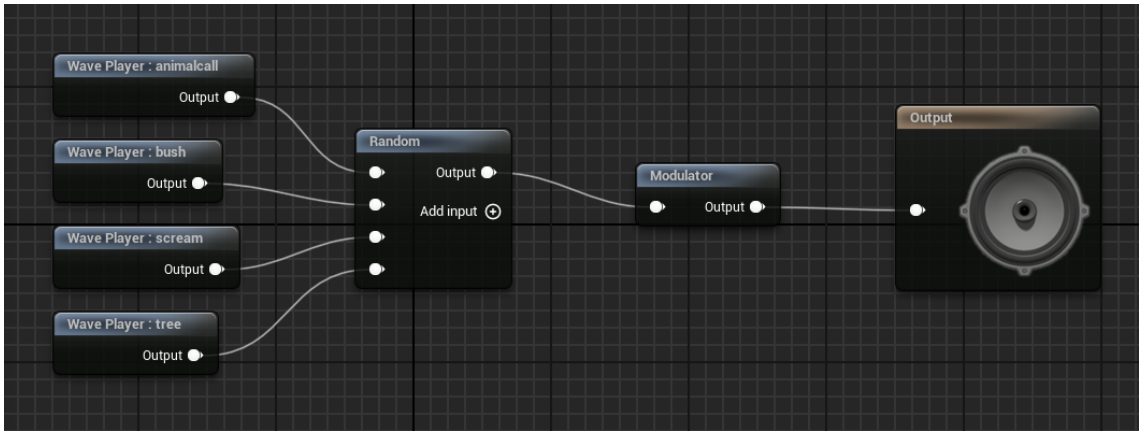
Kuva 14. Jumpscaren toiminnallisuuden Blueprint 2 osassa.

Kuva 15. Jumpscare hahmo näytöllä jumpscaren tapahtuessa.

4.2.3 Audiovisuaalisten mekaaniikkojen yhdistäminen pelikenttään

Pelikenttään on tuotu paljon erilaisia ääniä, jotta kauhupelille saataisiin synnytettyä kauhumaista tunnelmaa. Tähän kuuluu esimerkiksi taustamusiikki tai taustamelu ja satunnaiset äänet ympäri kenttää. Kaikki peliin tuodut äänet ovat hankittu freesound.org verkkosivulta ja niissä on Creative Commons lisenssi, jotta niitä on lupa käyttää pelissä vapaasti. Kaikki äänet ovat sen jälkeen tuotu Unreal Engineen ja niistä on tarvittaessa tehty Sound Cue tai Sound Attenuation assetit, jotta niitä voidaan käyttää erilaisilla tavoilla pelissä.

Taustamusiikki alkaa pelin käynnistyessä ja se soi pelissä koko ajan, ellei pelaaja astu tietylle alueelle ja taustamusiikki loppuu (Kuva 1). Pelikenttään on lisätty myös useita kohtia, missä soi satunnaisesti erilaisia ääniä. Nämä äänet ovat joko puun kaatuminen, puskan suhina, huuto tai eläimen ääni. Tätä toiminnallisuutta varten loin RandomCue nimisen SoundCue assetin, johon yhdistin nämä 4 ääntä. (Kuva 16.)



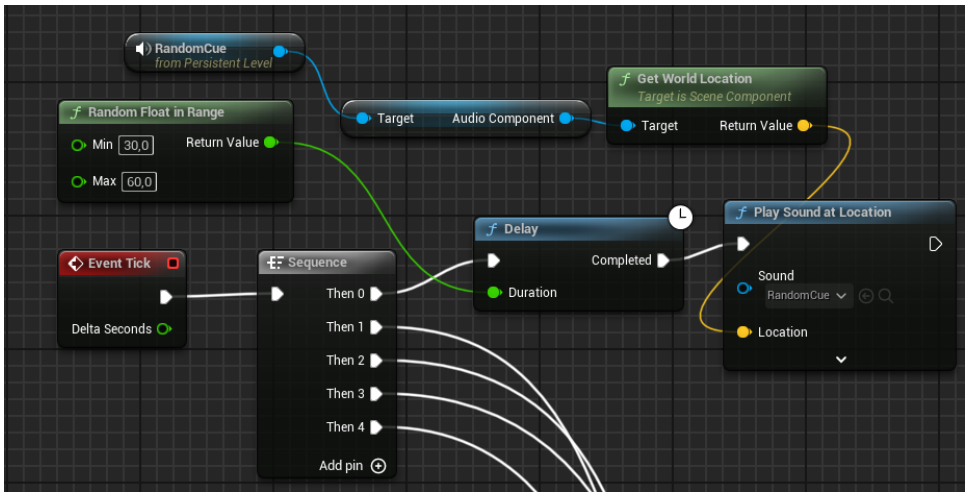
Kuva 16. RandomCue ääniassetti.

Tämän jälkeen lisäsin RandomCue assetteja eri puolille pelikenttää, jotta ääniä voi kuulua sieltä täältä. Äänet ovat tarkoituksellisesti lisätty keskelle metsää, jotta pelaaja ei näe mitä metsässä tapahtuu, mutta kuulee sieltä erilaisia ääniä. (Kuva 17.)

RandomCue assetti tarvitsi myös toiminnallisuuden, jonka tein pelikentän Blueprinteissä. Jokaista pelikentässä olevaa RandomCue assettia varten, jouduin tekemään Sequence nimisestä solmusta lähtevän yksinkertaisen

toiminnallisuuden. Jokaiseen RandomCueen annetaan viive, joka on 30 ja 60 sekunnin välillä ja se soitetaan pelikentässä olevan tietyn RandomCuen sijainnissa. (Kuva 18.) Tämä toiminnallisuus on muuten jokaiselle näistä sama, mutta sijainti on vain eri. Sijaintikin saadaan suoraan haettua viittaamalla kyseiseen RandomCueen.

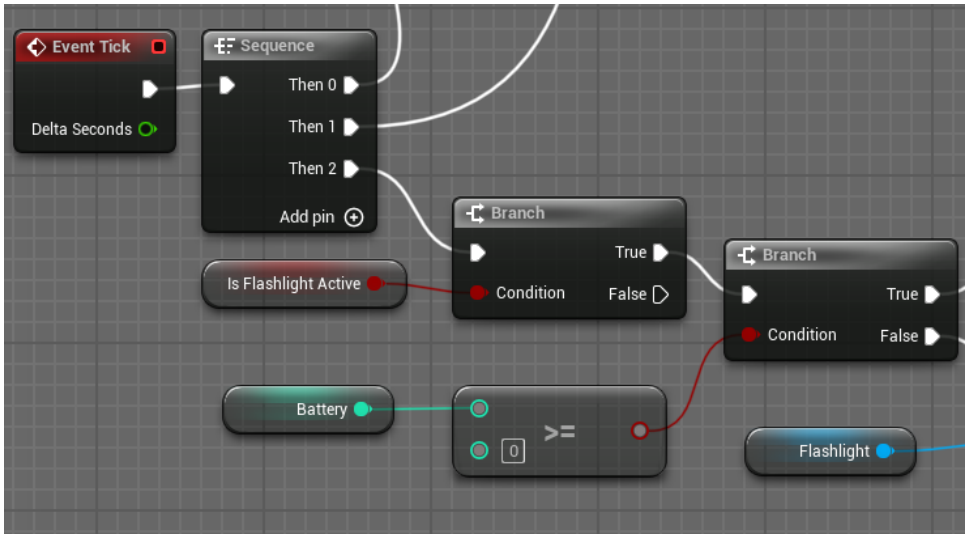
Kuva 17. RandomCue assettien sijoittelua ympäri pelikenttää.

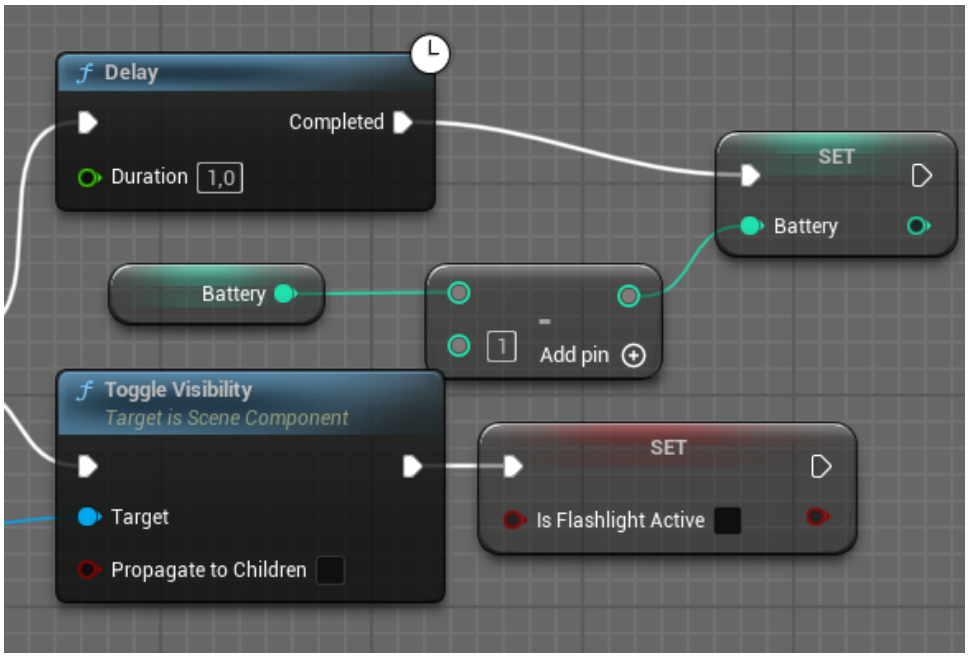


Kuva 18. Yhden RandomCuen toiminnallisuus.

Monissa kauhupeleissä ja tiivistunnelmaisissa peleissä valon kanssa tulee olla säästeliäs. Monesti pelaajalla on jokin valonlähde, joka täytyy laittaa uudelleen päälle, kun se sammuu. (Padilla, C. 2023.) Tämä on monesti joko tulitikku, sytkäri tai taskulamppu. Esimerkiksi The Forest (2014) nimisessä selviytymispelissä pelaajalla on sytkäri, joka sammuu välillä hetkeksi ja syttyy hetken päästä uudelleen. Pelissä on myös mahdollista löytää taskulamppu, jonka kanssa tulee myös olla säästeliäs, sillä se käyttää pattereita hyödykseen.

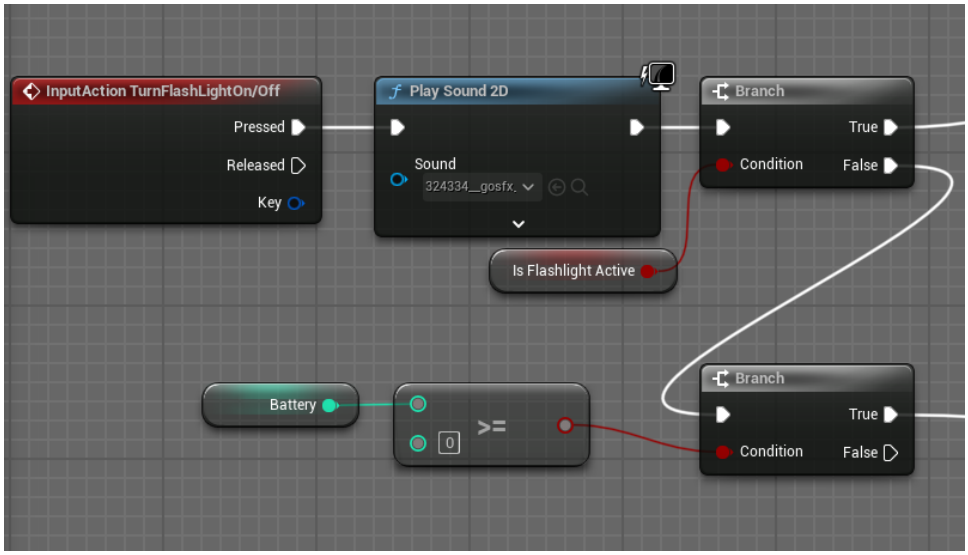
Tässä projektissa luotiinkin pelaajalle taskulamppu, mikä toimii paristoilla ja sammuu, kun sitä on käyttänyt tarpeeksi kauan. Tämä tehtiin pelihahmon Blueprinteissä, sillä pelihahmoon on suoraan yhdistetty taskulamppu. Blueprinteissä katsotaan, onko taskulamppu aktiivinen ja mikäli se on, niin taskulamppu kuluttaa paristoja sekunnin välein. Mikäli taskulampusta on paristot loppu, niin taskulamppu sammuu. (Kuva 19.)

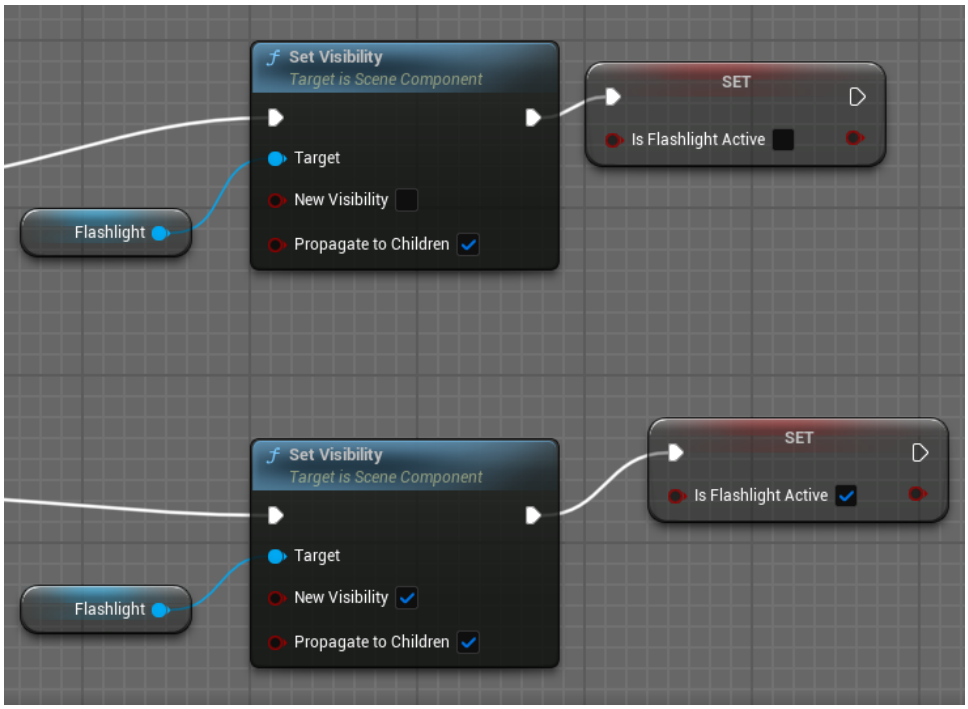




Kuva 19. Taskulampun paristojen kulutus 2 osassa.

Taskulamppu on asetettu syttymään painamalla F-painiketta ja tämä on asetettu suoraan projektin asetuksista tekemällä siitä InputAction. Blueprinteissä katsotaan, onko taskulampun paristoissa tarpeeksi virtaa sen päälle laittamiseksi ja mikäli on, niin taskulamppu saadaan sytytettyä F-painikkeella. Taskulamppu myös sammutetaan saman painikkeen avulla. SetVisibility on asetettu taskulampusta tulevaan SpotLight-komponenttiin, mikä on Blueprinteissä nimellä Flashlight. Tämän avulla valo on joko näkyvä tai ei. (Kuva 20.)



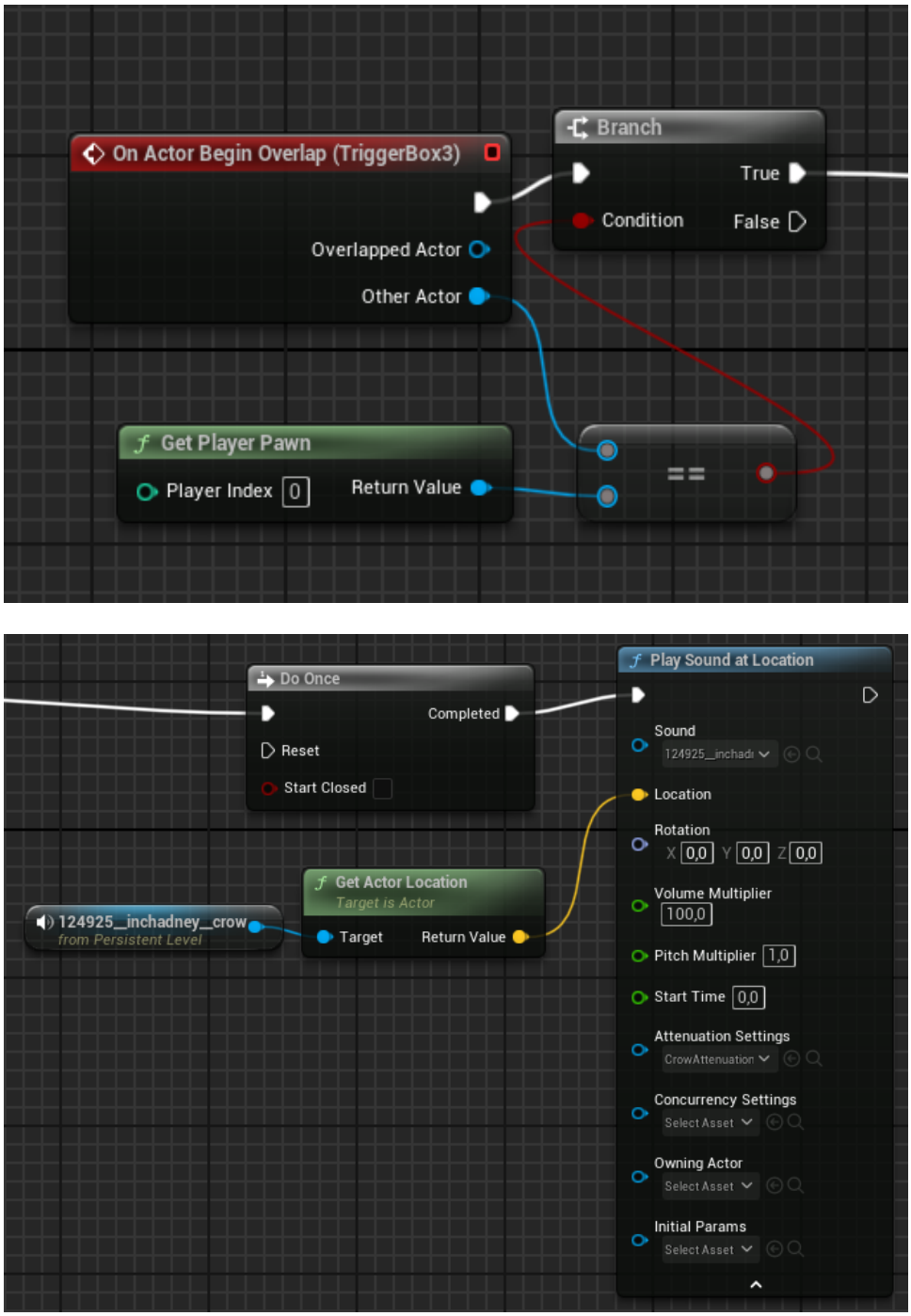


Kuva 20. Taskulampun sytytys ja sammutus 2 osassa

Taskulampun toiminnallisuuden loi alun perin projektissa ollut toinen kehittäjä Tomi Kiiski. Olisin luonut taskulampun toiminnallisuuden lähes samalla tavalla, joten suuria muutoksia sen toiminnallisuuteen ei tarvinnut tehdä. Suurimpina muutoksina taskulamppuun oli sen tehokkuuden muokkaaminen, sekä se että taskulamppu ei katoa pelaajan kädestä, kun sen sammuttaa vaan taskulampusta lähtee valo pois päältä.

Pelikenttään on myös tehty pari kohtaa, missä kuuluu tietyltä alueelta ääni, kun pelaaja astuu pelikentässä tietylle alueelle. Esimerkiksi kun pelaaja lähestyy kentässä olevaa taloa, pelaajan astuessa polulla olevien kahden lampun väliin, kuuluu talon katolta variksen huuto. Tämäkin on toteutettu TriggerBoxin avulla ja se kuuluu pelissä vain kerran, kun pelaaja osuu TriggerBoxin alueelle. (Kuva 21)

Pelikenttään on tehty samalla tyylillä myös hirviönhuuto, mikä tapahtuu, kun pelaaja kävelee hautausmaalla tarpeeksi lähelle hautakiviä. Tämän toteutus on tehty samalla tavalla kuin variksen huuto, mutta siinä on vain vaihdettu variksen ääni hirviön ääneksi.



Kuva 21. Variksen ääni 2 osassa

4.2.4 Haasteet

Suurimpina haasteina projektissa oli oikeanlaisten assettien löytäminen ja käyttäminen. Vaikka Unreal Enginen kaupasta löytyykin paljon assetteja mitä projektissa olisi voinut hyödyntää, niin ne olivat kuitenkin niin suuria, että niiden siirtäminen GitHubiin ei olisi onnistunut sen rajoitteiden takia.

Toiminnallista puolta varten jouduin katsomaan erilaisia ohjevideoita, kuten Gorka Gamesin tekemiä videoita, jotka veivät hieman aikaa. Toteutukset kuitenkin onnistuivat ilman suuria ongelmia, jonka ansiota niiden korjaamiseen ei tarvinnut käyttää paljoakaan aikaa. Suurimpana ongelmana oli pelikentässä olevan pohjan toteutus, sillä maaperän muokkaaminen ei aluksi toiminut tarkoituksen mukaisesti. Tästä syystä jouduin luomaan pelikentän pohjan kokonaan uudelleen, jonka jälkeen ongelmia siinä ei enää ilmennyt.

5 Pohdinta

Olen tehnyt opintojeni pohjalta pelikehitystä Unreal Enginellä jo ennen opinnäytetyötäni, mutta kokemukseni perustuu aiempaan versioon pelimoottorista. Unreal Engine 5 toiminnallisuus oli siis minulle täysin uutta projektin alusta lähtien, jonka takia jouduin käyttämään paljon aikaa sen opetteluun. Käytin hyödykseni paljon Unreal Enginen dokumentaatiota sekä erilaisia Youtubesta löytyviä ohjevideoita. Tarkoituksenani oli luoda enemmän erilaisia audiovisuaalisia mekaniikkoja kuin mitä opinnäytetyöstäni löytyy, mutta kokemuksen puutteen ja aikarajoitusten takia en pystynyt luomaan kaikkia haluamiani mekaniikkoja. Opinnäytetyöhöni olisin halunnut tehdä esimerkiksi vihollisen, joka jahtaa pelaajaa hiljalleen silloin kun sitä ei katso. Tähän olisin halunnut yhdistää myös äänet jahtausta varten. Toinen mekaniikka, minkä olisin halunnut tehdä olisi ollut satunnainen välähtävä valoefekti kentässä kuten salama. Näitä en kuitenkaan kerennyt luoda opinnäytetyötäni varten, vaan ne jäävät jatkokehitykseen.

Pelikentässä käytin hyödykseni vain ilmaisia kaikille jaollisia assetteja, mutta haluaisin myös oppimaan tekemään näitä itse. Jouduin aikarajoitusten sekä opinnäytetyön aiheen muutosten takia jättämään hahmojen mallintamisen sekä äänien luomisen tästä syystä jatkokehitykseen. Mikäli aiheeni olisi jatkunut alkuperäisenä, missä keskityin enemmän äänimaailman luontiin ja sen immersioon, niin olisin päässyt tekemään pelikenttään erilaisia ääniä.

Tavoitteenani oli luoda erilaisia audiovisuaalisia mekaniikkoja Unreal Enginellä ja pääsin tärkeimpään tavoitteeseen hyvin. Sain peliin luotua useita eri

mekaniikkoja, kuten jumpscaren, kentässä soivat satunnaiset äänet, rikkoutuvan lampun sekä särisevän lampun. Toiminnallisuudet näille luotiin Unreal Enginen blueprint-ohjelmointia hyödyntämällä, mistä minulla oli jo aiempaa kokemusta. En kuitenkaan osaa blueprint-ohjelmointia vielä täydellisesti ja opinnäytetyön ansiosta opinkin siitä paljon lisää.

Minulle jäi vielä paljon opittavaa ja jatkokehitettävää mekaniikkoihin liittyen, sillä en saanut kaikkea haluamaani tehtyä. Vaikka opinnäytetyön aikana opinkin paljon Unreal Engine 5:n toiminnallisuudesta, niin minulta jäi silti monia sen ominaisuuksia käyttämättä. Unreal Engine 5:ssä on vielä paljon opittavaa, eikä sen toiminnallisuutta kerkeä oppimaan kokonaan lyhyessä ajassa. Haluaisin oppia vielä paremmin käyttämään Lumen-valaistusjärjestelmää, jotta peleistä saadaan luotua aidomman näköisiä. Myös Nanite-järjestelmän oppiminen olisi hyödyllistä, sillä sen avulla pystyisin luomaan yksityiskohtaisia graafisia assetteja peliin ilman, että tietotekniset rajoitukset pelin pelaamista varten kasvaisivat huomattavasti.

Olen kuitenkin tyytyväinen opinnäytetyössäni luoduista mekaniikoista sekä siitä, mitä kaikkea opin Unreal Engine 5:n toiminnallisuudesta. Projektin aikana huomasin kuinka mukavaa Unreal Engine 5:llä on tehdä pelikehitystä ja aion jatkossakin käyttää kyseistä pelimoottoria vapaa-ajan projekteihini. Mikäli kiinnostus tätä projektia varten vielä jatkuu, niin aion jatkaa projektin kehittämistä ja luoda siitä lyhyen valmiin kauhupelin.

Lähteet

Adobe. 2022. Discover the hidden World of Foley Sound Effects https://www.adobe.com/creativecloud/video/discover/foley-sound-effects.html 13.5.2023

Anastasia. 2023. The Evolution of Video Game Art Styles https://ejaw.net/the-evolution-of-video-game-art-styles/ 21.10.2023

Arm.com, 2023. Immersive Gaming. https://www.arm.com/glossary/immersive-gaming 30.9.2023.

Askew, E. 2022. Level Up: The Evolution of Video Game music https://www.sweetwater.com/insync/level-up-the-evolution-of-video-game-music/ 11.5.2023

Azanbal, G. 2023. LinkedIn https://www.linkedin.com/in/gorka-aranzabal/ 4.1.2023

Balasubramanian, K. 2022. The Evolution of 3D Graphics in Video Games (The Hunt for Photorealism) https://www.gameopedia.com/the-evolution-of-3d-graphics-in-video-games-the-hunt-for-photorealism/ 28.10.2023

Capaldi, V. 2020. What Makes a Video Game Immersive? https://glassyeyewear.com/blogs/article/what-makes-a-video-game-immersive 30.9.2023

Casey, M. 2021. What Is Surround Sound Audio? https://www.lifewire.com/what-is-surround-sound-audio-2640440 19.5.2023

Cassel, D. 2018. The Surprisingly Rich History of ASCII Art https://thenewstack.io/surprisingly-rich-history-ascii-art/ 20.10.2023

Coltorti, G. 2022. Is Surround Sound Good For Gaming? https://switchandclick.com/is-surround-sound-good-for-gaming-or-a-gimmick/ 19.5.2023

Counter-Strike: Global Offensive. 2012. Valve Corporation

de Arteaga, P. 2018. MIDI format in video games https://patrickdearteaga.com/midi-format-in-video-games/ 15.5.2023

Dolby Games. 2021. Deeply Immersive Game Audio with Spatial Sound https://games.dolby.com/news/deeply-immersive-game-audio-with-spatial-sound/ 20.5.2023

Fortnite. 2017. Epic Games

Gameace Creative Studio. 2021. What is The Key to Immersive Game Design? https://game-ace.com/blog/immersive-game-design/ 30.9.2023

Gamestate. 2023. The evolution of video game graphics: from 8-bit to HD and VR https://gamestate.com/blogs/news/the-evolution-of-video-game-graphics-from-8-bit-to-hd-and-vr 21.10.2023

GitHub Docs. 2023. About Git Large File Storage https://docs.github.com/en/repositories/working-with-files/managing-large-files/about-git-large-file-storage 2.10.2023

Gorka Games. 2023. https://www.youtube.com/@GorkaGames 7.10.2023

Griffiths, D. 2018. The History of Pixel Art https://www.thefactorytimes.com/factory-times/2018/9/27/the-history-of-pixel-art 21.10.2023

Haroosh, I. 2022. Mono VS Stereo – Which One Should You Use – And Why! https://wealthysound.com/posts/the-difference-between-mono-stereo 19.5.2023

Hogan, B. 2014. How the Sound and Music Came Together http://neveralonegame.com/sound-music-came-together/ 13.5.2023

Kaur Arora, S. 2023. Which game Engine Should You Choose? https://hackr.io/blog/unity-vs-unreal-engine 2.10.2023

Keane, P. 2021. What Are SFX And How Can They Make Your Video Better? https://taketones.com/blog/what-are-sfx-and-how-can-they-make-your-videos-better 15.5.2023

Knight, S. 2021. What Are 2.5D Games? How They Differ From 2D and 3D Games https://www.makeuseof.com/what-are-2-5d-games-2d-3d/ 23.10.2023

Mischel, S. 2023. From Pong To The Last Of Us: The Evolution of Video Game Graphics https://www.cutscenesleep.com/blogs/news/from-pong-to-the-last-of-us-the-evolution-of-video-game-graphics 21.10.2023

Padilla, C. 2023. 13 Most Intense Horror Game Mechanics https://www.thegamer.com/most-intense-horror-game-mechanics/ 18.10.2023

Raymond, S. 2022. PS5 3D Audio – Everything You Need to Know https://thegamingsetup.com/ps5/buying-guides/3d-audio 20.5.2023

Rust. 2013. Facepunch Studios

Scarrat, D. 2018. The evolution of audio in videogames. https://www.acmi.net.au/stories-and-ideas/evolution-audio-videogames/. 11.5.2023

Sharpcoderblog. 2023. ASCII Art in Game Development https://www.sharpcoderblog.com/blog/ascii-art-in-game-development 20.10.2023

Sonic Minds. 2023. The Importance of Sound Design in Video Games https://sonicminds.dk/sectors/videogames/ 8.10.2023

Undertale. 2015. Toby Fox

Unreal Engine. 2023a. Assets and Content Packs https://docs.unrealengine.com/5.0/en-US/assets-and-content-packs-in-unreal-engine/ 13.11.2023.

Unreal Engine. 2023b. Blueprints Visual Scripting https://docs.unrealengine.com/5.0/en-US/blueprints-visual-scripting-in-unreal-engine/ 13.11.2023.

Unreal Engine. 2023c. Lumen Global Illumination and Reflections https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/ 13.11.2023.

Unreal Engine. 2023d. Nanite Virtualized Geometry https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/ 13.11.2023.

Unreal Engine. 2023e. Procedural Foliage Tool https://docs.unrealengine.com/5.0/en-US/procedural-foliage-tool-in-unreal-engine/ 13.11.2023.

Unreal Engine. 2023g. Sound Cue Reference https://docs.unrealengine.com/5.0/en-US/sound-cue-reference-for-unreal-engine/ 13.11.2023.

Unreal Engine. 2023h. Trigger Volume Actors https://docs.unrealengine.com/5.3/en-US/trigger-volume-actors-in-unreal-engine/ 13.11.2023

Unreal Engine. 2023i. Unreal Engine 5.0 Release Notes
https://docs.unrealengine.com/5.0/en-US/unreal-engine-5.0-release-notes/
13.11.2023

Unreal Engine. 2023j. Unreal Engine 5
https://www.unrealengine.com/en-US/unreal-engine-5 13.11.2023

Weaver, D. 2019. The History Of Audio And Music In Video Games
https://abbeyroadinstitute.com.au/blog/history-audio-music-video-games/
11.5.2023

Wiesen, G. 2023. What is a 3D Computer Graphic?
https://www.easytechjunkie.com/what-is-a-3d-computer-graphic.htm 28.10.2023

Wirtz, B. 2023a. The Power of Experience: The Wonders of Video Game Immersion
https://www.gamedesigning.org/learn/game-immersion/ 30.9.2023.

Wirtz, B. 2023b. 2D Video Game Art: Styles and History
https://www.gamedesigning.org/learn/2d-game-art/ 20.10.2023

Wreglesworth, R. 2022. A Beginner's Guide To MIDI: What Is It? How Does It Work?
https://musicianshq.com/a-beginners-guide-to-midi/ 15.5.2023