

VIRTUAALIYMPÄRISTÖN ÄÄNIMAAILMAN SUUNNITTELU JA TOTEUTUS UNITY- PELIMOOTTORILLA

Case: Revulon

Tiivistelmä

Tekijä(t) Miettinen, Mirko	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 53	Valmistumisaika Syksy 2019
Työn nimi Virtuaaliympäristön äänimaailman suunnittelu ja toteutus Unity-pelimoottorilla Case: Revulon		
Tutkinto Insinööri (AMK)		
Tiivistelmä <p>Opinnäytetyön aiheena on virtuaaliympäristön äänimaailman suunnittelu ja toteutus Unity-pelimoottorilla. Toimeksiantajana työssä toimi lahtelainen Revulon-niminen IT-alan yritys. Työn tarkoituksena oli selvittää, minkälaisia keinoja äänisuunnittelijalla, -ohjelmoijalla ja -teknikolla on käytettävissä 3D-ympäristön äänimaailman toteuttamista varten.</p> <p>Projektin työstäminen aloitettiin 3D-ympäristön suunnittelusta. Kun visuaalista puolta koskeva suunnitelma oli valmis, tehtiin myös äänisuunnitelma. Siihen kirjattiin lista tarvittavista äänistä, joita kyseisessä ympäristössä olisi mahdollisesti havaittavissa. Ennen äänityssessioita tutustuttiin eri äänitystapojen ja mikrofonien käyttötarkoituksiin. Myös muutamaa 3D-ympäristön luomisen kannalta oleellisinta signaaliprosessoria opeteltiin käyttämään. Lopulta äänitetyt ääniraidat asetettiin virtuaaliseen 3D-ympäristöön vertailun vuoksi kolmella eri äänityökalulla: Unityn omalla äänimoottorilla sekä FMOD- ja Steam Audio-nimisillä äänityökaluilla.</p> <p>Lopputulena todettiin Steam Audion tarjoamien ominaisuuksien puoltavan sen käyttöönottoa niin helppokäyttöisyyden, kuin realististen lopputuloksienkin kannalta. Varsinainen sovellus tehtiin Steam Audiota hyödyntäen, ja se luovutettiin Revulonille demonstraatiokäyttöön.</p>		
Asiasanat äänisuunnittelu, virtuaalituodellisuus, äänittäminen, ääniohjelmointi		

Abstract

Author(s) Miettinen, Mirko	Type of publication Bachelor's thesis	Published Autumn of 2019
	Number of pages 53	
Title of publication Designing and creating the world of sound in Unity game engine Case: Revulon		
Name of Degree Bachelor of engineering		
Abstract <p>The subject of this Bachelor's thesis is designing and creating the world of sound in Unity game engine. The work was done for Revulon, an IT company based in Lahti, Finland. The main goal was to find out what kind of tools a sound designer, programmer and audio technician have for implementing a world of sound for a 3D environment.</p> <p>The project was started with the basic design of the 3D environment. Visual and sonic plans were made before recording. The sonic design included a list of audio tracks which could be heard in such a location. Different recording methods and purposes of microphones were examined before the recording sessions. The most important signal processors for creating a 3D environment were also learned to use. Finally, recorded audio tracks were placed in the 3D environment with three different audio tools for comparison. Tools that were used were Unity's own audio engine, FMOD and Steam Audio plugin.</p> <p>As a result, the features offered by Steam Audio supported its deployment due to the ease of use and realistic results it provided. Steam Audio was used in the final version of the application, which was released to Revulon for demonstrational use.</p>		
Keywords sound design, virtual reality, audio recording, audio programming		

SISÄLLYS

1	JOHDANTO.....	1
2	ÄÄNISUUNNITTELU	3
2.1	Työnjako.....	3
2.1.1	Sovelluskehityksen johtajat.....	3
2.1.2	Äänitiimin toimijat.....	4
2.2	Kuunteleminen.....	4
2.3	Äänen moniulotteisuus	5
2.4	Interaktiivisen ääniraidan suunnittelu	6
2.4.1	Yksinkertainen interaktiivisuus.....	7
2.4.2	Monimutkainen interaktiivisuus	8
3	ÄÄNITUOTANTO.....	10
3.1	Äänitystilanteet	10
3.1.1	Äänitysympäristön valinta	10
3.1.2	Akustiikka	11
3.2	Äänitystavat ja mikrofonit	12
3.2.1	Perinteinen äänitysmenetelmä.....	12
3.2.2	Kondensaattorimikrofoni	13
3.2.3	Lähimikrofonitekniikka	15
3.2.4	Dynaaminen mikrofoni.....	16
3.2.5	Ympäroivän äänen tallennus ja nelikanavainen ääni.....	16
3.3	Synteettinen ääni.....	17
3.4	Signaaliprosessorit	18
3.4.1	Taajuuskorjain ja leikkurit.....	19
3.4.2	Kompressori	21
3.4.3	Kaiku	23
4	OHJELMOINTI.....	27
4.1	Unity-pelimoottori.....	27
4.1.1	Peruskomponentit.....	28
4.1.2	Unityn 3D-äänityökalu.....	30
4.1.3	Mikseri.....	31
4.2	Äänen laukaisijat ja dynaaminen ääniraita	33
4.3	FMOD.....	35
4.3.1	Signaaliprosessorit	36

4.3.2	Äänen satunnaistaminen	37
4.4	Steam Audio	37
4.4.1	Materiaalit, heijastukset ja tukkeuma	38
4.4.2	Tekniset ominaisuudet.....	38
4.5	Projektin toteutus	39
5	CASE.....	40
5.1	Suunnittelu	40
5.2	Kohtauksen luonti	40
5.3	Ensimmäinen äänityssessio.....	41
5.3.1	Toinen äänityssessio	43
5.3.2	Synteettiset äänet.....	43
5.4	Äänimaailman peruskomponentit.....	44
5.4.1	Tilan tunnun luominen	45
5.4.2	Oletusasetuksien muokkaaminen	46
5.5	FMOD.....	46
5.5.1	Ääniraitojen luonti	47
5.5.2	Ongelma Doppler-ilmiön kanssa	48
5.6	Steam Audio	48
5.6.1	Steam Audio äänilähde.....	50
5.6.2	Edistyneempi etäisyyden tarkastelu.....	52
6	YHTEENVETO	53
	LÄHTEET	54

1 JOHDANTO

Kautta aikojen ihmiset ovat nauttineet erilaisista audiovisuaalisista elämyksistä. Aluksi näitä elämyksiä tarjosivat ainoastaan teatterit, myöhemmin radio ja elokuvateatterit. Lopulta oli mahdollista ostaa kotiin televisio. Nykypäivänä saatavilla on pelikonsoleita, älypuhelimia, tabletteja, tietokoneita ja jopa virtuaalitodellisuusjärjestelmiä. Tällaisien laitteiden myynti käy vilkkaana niin viihde-, kuin hyötykäyttöön ja niitä ostavat niin yksityishenkilöt kuin yrityksetkin.

Viimeisimpien audiovisuaalisten laitteiden yhdistävä teema on selkeästikin ollut päästä niin sanotusti käyttäjän ympärille. Televisiolähetysten ja laitteiden mahdollistaman kuvanlaadun parantuessa myös televisioiden koot ovat kasvaneet huomattavasti. Myös projektorin käyttäminen television korvaajana on yleistä. Viimeisimpänä edistysaskeleena televisiotekniikkaan on tullut näytön kaarevuus. Janne Luotola kirjoittaa tivi.fi-verkkosivulla artikkelissaan *”Onko kaarevassa tv:ssä mitään järkeä? Asiantuntijat paljastavat sen mitä kuluttaja ei hoksaa”*, että kaarevuudella pyritään tarjoamaan katsojalle mahdollisimman realistinen katseluelämys ilman vääristymiä ja heijastuksia.

Myös äänentoistojärjestelmien kehityksessä on havaittavissa samanlaisia seikkoja. Aluksi televisioissa oli vain yksi kaiutin, myöhemmin kaksi, eli oikea ja vasen. Seuraavaksi markkinoille saapuivat kotiteatterijärjestelmät, joiden käyttämä surround-tekniikka tukee esimerkiksi 5.1- tai 7.1-äänijärjestelmiä. Näiden toiminta perustuu siihen, että kaiuttimia sijoitellaan niin kuulijan eteen, kuin taaksekin. Tällä tekniikalla pyritään luomaan kuulijalle sellainen tunne, että hän olisi katselemaisensa ohjelman tai elokuvan tapahtumien keskellä.

Myös elokuvateatterit ovat hyötäneet niin ääni-, kuin videotekniikankin kehityksestä. Moderneissa elokuvateattereissa käytetään vähintäänkin 7.1-äänijärjestelmiä, mutta esimerkiksi Finnkino Omenan äänijärjestelmä tukee yhteensä 62.2-kanavaista äänijärjestelmää. (Tervola 2017.) Lisäksi nykyaikaisissa elokuvateattereissa on mahdollisuus 3D-elämykseen, joka onnistuu tiettyjen elokuvien kohdalla 3D-laseja käyttämällä. Tällöin voidaan alkaa puhua immersioista, joka on yksi tämän opinnäytetyön pääteemoista.

Opinnäytetyössä tarkastellaan immersiota ja sen luomista kuitenkin vielä hieman kehittyneemmän tekniikan kautta. Tarkoituksena on luoda äänimaailma virtuaalitodellisuusympäristölle, niin että se vastaisi mahdollisimman hyvin reaali maailman ääniavaruutta. Virtuaalitodellisuudella tarkoitetaan digitaalisesti mallinnettua ympäristöä, jonka sisällä pystytään liikkumaan, suorittamaan tietynlaisia interaktioita ja aistimaan ympäröiviä ärsykeitä. Työn

käytännön osuus suoritetaan lahtelaiselle Revulon-nimiselle yritykselle, mutta valmis sovellus jää odottamaan yrityksen asiakkailleen järjestämiä virtuaalitodellisuussovelluksien esittelytilaisuuksia.

2 ÄÄNISUUNNITTELU

2.1 Työnjako

Kun äänimaailmaa aletaan luoda, aloitetaan kaikki suunnittelusta. Suunnitteluvaiheessa laaditaan eräänlainen punainen lanka koko tuotannolle. Jos projektin parissa työskentelee useampi kuin yksi henkilö, suoritetaan myös työnjako, ellei sellaista ole jo tehty. Äänisuunnitteluun ryhdyttäessä sovelluksesta täytyy olla jonkinlainen visio tai jopa raakaversio, jotta tiedetään, minkälaista tunnelmaa halutaan luoda mihinkin eri vaiheeseen ja minkälaisia äänitehosteita sovellukseen tulisi sisällyttää.

Pienien yritysten sisällä koko äänituotannosta saattaa vastata yksi henkilö, mutta suurimmassa produktioissa on yleensä mukana kokonainen tiimi. Äänitiimi on usein jaettu eri toimijoihin, joista tosin jotkut saattavat hoitaa useita eri tehtäviä. Yrityksen sisäisiin toimijoihin kuuluu tyypillisesti muun muassa tuottajia, suunnittelijoita ja teknikkoja sekä ääniohjaaja. Usein yrityksen ulkoisena toimijoina näkyvimpänä osapuolena on sovelluksen julkaisija, jonka avulla sovellukselle saadaan näkyvyyttä ja kysyntää.

2.1.1 Sovelluskehityksen johtajat

Karen Collins (2008, 87) teoksessaan *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design* listaa tyypillisen äänitiimin eri toimijat. Hän aloittaa tuottajasta, jonka toimenkuvaan kuuluu koko projektin kehitysvaiheen yleinen johtaminen. Tuottaja valvoo ja johtaa useita prosessiin liittyviä eri aspekteja. Näitä ovat esimerkiksi sovelluksen kehitysprosessi, laillisuus, markkinointi sekä sopimusten solmiminen esimerkiksi ulkopuolisten toimijoiden kanssa. Nämä eivät siis suoranaisesti liity pelistä äänimaailman luomiseen, mutta ovat kuitenkin kytköksissä siihen.

Ylimpänä seuraavista tahoista on sovelluksen julkaisija, joka johtaa sovelluskehitystä ja valvoo valmiin tuotteen laatua. Usein julkaisija on ulkopuolinen taho, mutta toki julkaiseminen voidaan toteuttaa myös pelin kehittäneen yrityksen sisäisin voimin. Julkaisijan suurin intressi on, että valmiista sovelluksesta tulee markkinointikelpoinen sekä kiinnostava.

Koska julkaisija tarjoaa sovelluskehitykseen suuren taloudellisen tuen, on sillä myös suuri sananvalta sovelluksen suunnittelussa. (Collins 2008, 85.) Usein julkaisijat järjestävätkin esimerkiksi tapahtumia, joissa esitellään julkaistavaa sovellusta. Julkaisijan on siis hoidettava sovelluksen markkinointi niin, että se tavoittaa mahdollisimman laajan yleisön. Viisi suurinta julkaisijaa pelimaailmassa vuonna 2018 oli Capcom, Sega, Electronic Arts, Nintendo sekä Ubisoft. (Dietz 2019.)

2.1.2 Äänitiimin toimijat

Ääniohjaaja on äänitiimin johtaja. Hän vastaa yleisesti äänivision sekä suunnitelman noudattamisesta ja ohjaa äänimaailman luovaa sekä teknistä toteutusta. Ohjaajan on laadittava äänisuunnitelma, aikataulu ja budjetti, rekryoitava tuotantotiimi ja vastattava tekniikan toimivuudesta. Ohjaajan vastuulla on myös ulkopuolisten toimijoiden johtaminen. Näitä toimijoita voi olla esimerkiksi ääninäyttelijät, dialogien ohjaajat sekä ääniteknikot. Ohjaaja voi olla vastuussa myös lopullisesta äänimiksauksesta. (Collins 2008, 86.)

Ääniteknikolla voidaan tarkoittaa foley-artistia tai ohjelmoijaa. Foley-artisti on henkilö, joka tuottaa sovellukseen äänitehosteet luomalla sekä äänittämällä ne keinotekoisesti uudelleen studio-olosuhteissa. Foley-nimitykseen törmää usein elokuvamaailmassa, mutta se tarkoittaa samaa asiaa myös peli- ja sovelluskehityksessä. Foley merkitys korostuu nimenomaan pelimaailmassa, sillä pelit eivät juuri koskaan sisällä kuvattua materiaalia, vaan kaikki luodaan digitaalisesti. Tällöin myöskään ääntä ei synny prosessin aikana. Foley-artisti voi toki olla vastuussa myös dialogin sekä taustamusiikin äänittämisestä ja muusta toteuttamisesta, mutta usein varsinkin musiikista vastaa kuitenkin ulkoinen toimija. (Collins 2008, 86.) Ääniohjelmoija on sen sijaan henkilö, joka toimii jo työstettyjen ääniraitojen kanssa. Hänen tehtävänä on ohjelmoida oikeat ääniraidat alkamaan oikeisiin aikoihin ja oikeiden tapahtumien yhteydessä. Esimerkkinä tästä lasin särkyminen, kun tiputettu pullo osuu maahan tai oven narina sillä hetkellä, kun ovi avataan.

Koska äänisuunnittelu on yksi tämän opinnäytetyön pääaiheista, keskitytään myös äänisuunnittelijan ominaisuuksiin ja tehtäviin tarkemmin. Englannin kielen nimityksellä ”sound designer”, eli äänisuunnittelija, tarkoitetaan usein ketä tahansa äänitiimin jäsentä, eli henkilöä, joka osallistuu äänien tuottamiseen, niiden jälkikäsitteilyyn tai ohjelmoimiseen. Sanan varsinaisessa merkityksessä äänisuunnittelija on kuitenkin äänitiimin luova johtaja, joka ohjeistaa äänitiimin muita jäseniä. Kuten kaikilla äänitiimin jäsenillä, eritoten äänisuunnittelijalla täytyy olla hyvin harjaantunut kuulo, jotta hän pystyy kuuntelemaan eri ääniä useiden eri aspektien kautta. Lisäksi äänisuunnittelijan ja tiimin jäsenien täytyy tietää, miten vaikuttaa siihen, mitä yleisön halutaan havaitsevan. (Alten 2011, 278.)

2.2 Kuunteleminen

Stanley R. Alten (2011, 278.) kirjoittaa kuuntelemisesta teoksessaan *Audio in Media*. Hänen mukaansa taidon kuunnella ympäröiviä ääniä äänisuunnittelijan lailla voi harjoitella. Se vie aikaa, mutta on välttämätön edellytys äänitiimissä työskentelylle. Ympäröivien äänien havainnointi edellyttää sitä, että pystytään sulkemaan muut äänet ja keskittymään vain siihen ääneen, jota halutaan kuunnella. Kovat tai jopa ärsyttävät äänet, kuten koiran

haukkuminen, hälytysajoneuvon sireeni tai rikkoutuva lasi, kiinnittävät usein huomiomme. Tämä johtuu niiden voimakkuuden lisäksi lähinnä siitä, että ne ovat usein yllättäviä eivätkä tapahdu läheskään joka kerta, kun ollaan samassa tilanteessa. Monia ympäriltä tulevia ääniä sen sijaan pidetään itsestäänselvyytenä. Taustääänet, kuten ilmastoinnin tai tuulen humina ja tietokoneen surina, ovat joko niin hiljaisia tai yleisiä, ettei niihin välttämättä kiinnitetä juurikaan huomiota. Juuri tällaisia huomaamattomia ääniä äänisuunnittelijan täytyy pystyä havainnoimaan, jotta hän pystyy realistisesti toteuttamaan ne uudelleen. (Alten 2011, 278.)

Alten kirjoittaa myös kriittisestä kuuntelutavasta. Hänen mukaansa kriittinen kuunteleminen liittyy etenkin äänen tekniseen kokonaisuuteen. Kun kuunnellaan kriittisesti, pyritään äänestä havaitsemaan joko sen teknisiä ominaispiirteitä tai epäkohtia. Tällaisia saattaa olla muun muassa äänen tekninen laatu tai sen heikkous, yksityiskohtien selkeys, jonkun taajuusalueen korostuminen tai eri instrumenttien äänenvoimakkuuden epätasapaino. Harjaantuneemman korvan omaava henkilö voi esimerkiksi havaita äänitteessä korkean bassorummun äänenvoimakkuuden, joka saa aikaan miksausuksen tummuuden. Tällaiset seikat liittyvät usein nimenomaan äänitteen kuunteluun, mutta myös musiikkiesityksiä voi kuunnella kriittisesti. Tällöin aiempien esimerkkien lisäksi saattaa esiintyä inhimillisyydestä johtuvia epäkohtia tai ominaisuuksia. Laulaja saattaa vahingossa aloittaa laulamisen hie-man liian aikaisin tai jokin uniikki äänitehoste voi herättää kuulijan huomion. Kriittisesti voi kuunnella myös televisiolähetystyksiä, joista voidaan havaita esimerkiksi yleisön aplodien liiallisen äänenvoimakkuuden tai tapahtumapaikan akustiikan vaikutuksen lähetysten äänimaailmaan. (Alten 2011, 279.)

2.3 Äänen moniulotteisuus

Tunnelman luomisen kannalta ääni on siitä moniulotteinen työkalu, että sillä ei ole niin sanottua rajattua aluetta. Videon täytyy olla ruudun sisällä ja noudattaa tiettyä kuvasuhdetta, mutta videoon liitetyt äänitehosteet voivat liittyä tapahtumaan, joka ei ruudulla näy. Tällaisilla ruudun ulkopuolelta tulevilla äänillä onkin usein valtava vaikutus kohtauksen tunnelmaan, ympäristöön ja toimintaan. Tässä yhteydessä korostuu johdannossakin mainittu kuvassa 1 nähtävillä oleva surround-tekniikka, jonka avulla ääniä voidaan toistaa kuulijan takaa. Sama pätee virtuaalitodellisuussovelluksen luomiseen. Koska käyttäjällä on vapaus valita mihin suuntaan hän katsoo, on myös äänien elettävä sen mukaan. Äänisuunnittelijan vastuulla onkin äänimaailman suunnittelu siten, että se kattaa 360 astetta niin vaaka-, kuin pystysuunnassakin. Tämä tarkoittaa sitä, että ääntä on tultava myös suunnasta, johon sovelluksen käyttäjä ei juuri sillä hetkellä katso.



Kuva 1. Surround-tekniikkaa hyödyntävä kaiutinjärjestelmä 5.1-kokoonpanolla (Stanton 2017)

Randy Thom (1999) kirjoittaa ääniraitaa priorisoivan elokuvan suunnittelutyylistä artikkelissaan *Designing for Sound*. Artikkelin on julkaistu vuonna 1999, eli se on melko vanha, mutta pätee edelleen. Thom korostaa, että toimiakseen ruudun ulkopuolelta kantautuvat äänet tarvitsevat jonkinlaisen visuaalisen vastineen tuekseen. Kun tämä visuaalinen asia on kerran näytetty äänen kanssa yhtä aikaa, katsoja yhdistää ne jatkossa itse, eikä visuaalista puolta välttämättä enää tarvita. Kohtaus voidaan esitellä siten, että siinä on jokin tunnusomainen taustääni. Myöhemmin kuva voi vaihtua esimerkiksi kahden ihmisen vuoropuheluksi, mutta ruudun ulkopuolelta kantautuvan äänen jatkuessa katsoja tietää, että tilanne kohtauksessa on edelleen sama. Tämä tekniikka on yleistä esimerkiksi kauhu elokuvissa. Olento, jota päähahmo pakenee, on ruudun ulkopuolella ”piilossa”, mutta koska sen pitämä ääni on edelleen kuultavissa, tiedetään, että se on edelleen läsnä ja vaara on yhä olemassa. Thom korostaa tällaisten äänitehosteiden tärkeyttä kohtauksen tunnelman ja kiinnostavuuden luojina.

2.4 Interaktiivisen ääniraidan suunnittelu

Tähän asti on käsitelty lähinnä lineaarisen median äänisuunnittelussa toimivia tekniikoita. Linearisella mediallyä tarkoitetaan mitä tahansa mediaa, jolla ei ole interaktiivisuutta. Tällaisia ovat esimerkiksi elokuvat ja videot. Elokuvien ääniraita voidaan ennalta tuottaa

alusta loppuun asti. Jokaiselle äänitehosteelle, musiikille ja vuorosanalle on oma aikaleimansa, ja ne ajoitetaan tukemaan elokuvan visuaalista puolta. Lineaarisen median ääniraita voidaan miksata niin, että sen dynamiikalla, taajuuskorjauksella ja muulla signaaliprosessoinnilla saavutetaan mahdollisimman selkeä ja realistinen ääni. Äänisuunnittelija saa apuja visuaalisesta puolesta luovien ratkaisujen tekemiseen. Jos kuvassa esimerkiksi ovi suljetaan hitaasti, se saattaa pitää pitkän narisevan äänen, mutta jos se taas paiskataan kiinni, oven aiheuttamasta ilmavirrasta kuuluu suhahdus, jonka jälkeen voimakas paukahdus sen osuessa oven karmeihin. (Alten 2011, 336.)

Interaktiiviseen sovellukseen ääniä suunniteltaessa täytyy ottaa huomioon useita eri seikkoja. Ensinnäkin sovelluksen ajon aikana sen etenemisestä voidaan ennustaa suhteellisen vähän. Interaktiivinen media perustuu siihen, että käyttäjä voi sovelluksessa tehdä mitä haluaa, miten haluaa ja milloin haluaa. Hyvänä esimerkkinä tästä on videopelit, joiden etenemistapaan ja -nopeuteen käyttäjällä on hyvinkin paljon mahdollisuuksia vaikuttaa. Tämä tarkoittaakin sitä, että sovelluksen äänitehosteet eivät voi olla lineaariseen toistoon suunniteltuja, sillä sovellus ei koskaan etene samalla tavalla. Tämän lisäksi tietokonesovelluksen ääniraidan pituutta on vaikea määrittää, sillä jotkut pelit saattavat olla jopa loputtomia. Äänisuunnittelijan sekä taustamusiikin säveltäjän täytyykin tuottaa tilanteiden mukaan elävä, loputtomalla silmukalla saumattomasti soiva ääniraita, jotta sovelluksen taustamusiikki ei lopu kesken.

2.4.1 Yksinkertainen interaktiivisuus

Ääniraidan interaktiivisuus voidaan jakaa kahteen eri osa-alueeseen. Peruseräkkäisyyden ja monimutkaisen interaktiivisuuden eron voi huomata etenkin silloin, kun äänitehosteen laukaiseva toiminto tehdään useita kertoja peräkkäin. (Alten 2011, 343 - 345). Esimerkkinä tästä voidaan käyttää aiemmin mainittua oven sulkemista. Joskus oven liike on ohjelmoitu tapahtumaan sovelluksessa aina samalla tavalla, eli toisin sanottuna se on animoitu. Tällöin voidaan turvautua peruseräkkäisyyteen, eli voidaan käyttää äänikirjastosta löytyvää yhtä ja samaa oven sulkemisääntä joka kerta, kun ovi suljetaan. Mikäli toiminto suoritetaan useita kertoja peräkkäin, toistuu myös sen laukaisema äänitehoste aina uudestaan ja uudestaan. Tehosteesta tulee helposti toistuvan kuuloinen, mikä todennäköisesti heikentää sovelluksen todellisuuden tuntua. Aina, kun sovelluksen käyttäjällä on mahdollisuus vaikuttaa, kuinka sovelluksen fyysisiä interaktioita tarkalleen ottaen tehdään, täytyy äänisuunnittelijan ja -ohjelmoijan keksiä keino, jolla niitä vastaavat ääniraidat saadaan kuulostamaan mahdollisimman realistisilta ja tapauskohtaisilta.

Tietokonesovelluksien taustamusiikit tai niiden lyhyet pätkät voivat olla peruseräinteraktiivisia. Lisäksi äänitehosteet, jotka toistuvat vain kerran sovelluksen ajon aikana, tai ovat reaali-maailmassakin aina samanlaisia, on järkevää toteuttaa niin, että ne noudattavat vain peruseräinteraktiivisuutta. Tällaisia ovat esimerkiksi ovikello, aseiden laukaus tai auton äänitorven päästämä ääni. Myös käyttöliittymään ja valikkoihin liittyvät äänitehosteet kuten kilaus, joka kuuluu, kun hyväksytään jokin valinta, tai fanfaari, kun saavutetaan virstanpylväs, ovat lähes poikkeuksetta peruseräinteraktiivisia äänitehosteita. Näin sovelluksen käyttäjä saadaan mielessään yhdistämään äänitehoste ja interaktio toisiinsa. Monimuuttokainen interaktiivisuus

2.4.2 Monimuuttokainen interaktiivisuus

Usein virtuaalituodellisuudessa oven avaamis- ja sulkemistapoja on äärettömän määrä, sillä oven kahvaan voi ”tarttua” ja oven voi avata ja sulkea joko hitaasti tai nopeasti. Joku voi paistata oven kiinni, toinen voi ensin raottaa sitä ja sitten vasta avata sen kokonaan. Joku saattaa jättää oven puoliksi auki ja keskittyä sitten johonkin muuhun asiaan. Tällöin kysymykseen tulee monimuuttokaisen interaktiivisuuden suunnittelu ja toteuttaminen.

Monimuuttokaista interaktiivisuutta voidaan lähestyä kahdella eri tavalla riippuen siitä, mitä äänitetään. Ensimmäinen ja selkeämpi tapa on äänittää kaikille äänitehosteille useita eri variaatioita. Tämä tulee kyseeseen, kun äänitetään esimerkiksi miekkataistelusta kantautuvia terän iskun ääniä. Nämä äänit toistuvat useita kertoja yhden kohtaamisen aikana, ja yksikään niistä ei tosielämässä ole samanlainen kuin toinen. Kun miekaniskut koostuvat useista eri ääniraidoista, on käyttäjänkin kokemus sovelluksesta realistisempi, eikä äänitehosteiden toistuvuus pääse heikentämään sitä. Kuitenkaan tätä tapaa ei kannata soveltaa tilan tunnun luomiseen varsinkaan silloin, kun sovelluksen käyttäjä saa täysin vapaasti valita sijaintinsa sovellukseen luodussa 3D-maailmassa. Tällöin on syytä ottaa käyttöön käytössä olevan pelinkehitysohjelman sisäiset tai liitännäissovelluksesta löytyvät signaalinprosessointityökalut.

Moneen äänen käsittelyyn tarkoitettuun ohjelmaan on sisällytetty perustyökalut, joilla äänisignaalia voidaan prosessoida. Tällaisia ovat muun muassa kaiku, kompressori ja erilaiset suodattimet. Virtuaalituodellisuusympäristöön ääniä luotaessa tarvitaan todennäköisimmin juurikin kaikua sekä ala- ja ylätaajuussuodattimia. Näillä pystytään muokkaamaan yhtä äänitehostetta sovelluksen ajon aikana niin, että se säilyttää realismin silloinkin, kun pelaajan sijainti 3D-ympäristössä muuttuu. Esimerkiksi auton äänitorven soidessa kadulla, ääniraitaan lisätään mahdollisesti kaikua. Jos äänilähde taas on kauempana, siitä poistetaan ylätaajuuksia, sillä nämä taajuudet vaimenevat äänen kantautuessa kauempaa.

(Case 2007, 127.) Näitä asioita tarkastellaan lisää seuraavissa pääkappaleissa, jotka käsittelevät itse äänittämistä ja siihen liittyviä jälkikäsittelemahdollisuuksia sekä äänien ohjelmointia valitussa pelinkehitysympäristössä.

3 ÄÄNITUOTANTO

3.1 Äänitystilanteet

Kun äänisuunnittelu on tehty, voidaan käynnistää äänitysprosessi. Lyhyesti selitettynä äänittämisellä tarkoitetaan minkä tahansa äänen tallentamista siten, että se voidaan myöhemmin toistaa uudelleen. Koska tässä opinnäytetyössä äänitetyjä raitoja käytetään tarkoitukseen, johon ne harvemmin kelpaavat sellaisenaan, käydään läpi myös muutamia äänen käsittelyyn tarkoitettuja komponentteja. Äänittämään ryhdyttäessä oletuksena on, että tiedetään tarkalleen ottaen, mitä äänitetään ja mihin tarkoitukseen tuotokset menevät. Näiden kysymyksien pohjalta valitaan käytettävät mikrofonit ja muut työkalut sekä pääteään, minkälaisia tekniikoita aiotaan käyttää. Lisäksi äänittäjän tulee tietää, mitä äänitehosteita ja efektejä on mahdollista simuloida jälkeempään studio-olosuhteissa. Hyvän äänisuunnitelman avulla voidaan minimoida ylimääräisen työn määrä, mutta toisaalta saada kaikki tarvittava materiaali kerralla äänitettyä.

Koska elokuvaan, peleihin tai tämän opinnäytetyön tähtäämään virtuaalitodellisuusympäristöön tullaan todennäköisesti tarvitsemaan laaja valikoima erilaisia äänitehosteita eri tiloissa, joudutaan asiaa pohtimaan äänityssessiota suunniteltaessa. Äänityssessiolla tarkoitetaan yhtä sellaista kertaa, kun äänityskalusto asetetaan äänityskuntoon ja äänitetään kaikki sellainen materiaali, mitä kyseisellä kokoonpanolla on mahdollista äänittää. Elokuvia ja muita videoita tehtäessä tarve on usein niin sanotulle live-äänittämiselle. Live-äänittämisellä tarkoitetaan sitä, että dialogi ja muut tapahtumat taltioidaan kuvauksien aikana. Tällöin ääniraita vastaa täydellisesti näyttelijöiden huulien liikettä ja heidän tekemisiään, ja luo täten realistisemmän kuvan, kuin esimerkiksi dubattuna. Toki tällaisiin lineaarisiin medioihin ääniä luodaan jälkikäteenkin, mutta ne rajoittuvat lähinnä taustamusiikkiin ja äänitehosteisiin. Sen sijaan virtuaalitodellisuutta varten ei juurikaan pystytä live-äänittämistä toteuttamaan, sillä kaikki sovelluksen sisältö luodaan tietokoneella. Tällöin kyseeseen tulee sovelluksen koko äänimaailman tuottaminen sovellusta vastaavassa ympäristössä tai äänitysstudioissa.

3.1.1 Äänitysympäristön valinta

Kun tuotetaan äänimaailmaa virtuaalitodellisuusympäristöön, joudutaan todennäköisesti äänittämään myös ulkotiloissa. Tällöin akustiikan muokkaaminen saattaa osoittautua hankalaksi tai mahdottomaksi. Toisaalta tämä antaa äänittäjälle mahdollisuuden etsiä virtuaaliympäristöä akustisilta ominaisuuksiltaan vastaavan paikan ja tehdä äänitykset kyseisessä ympäristössä. Tämä onnistuu varsinkin silloin, kun virtuaaliympäristönä on tosimaailmaa.

ilmaan pohjautuva miljö. Tällöin äänitteiden sisältöä ei välttämättä tarvitse muokata vastaamaan mallinnetun ympäristön oletettuja akustisia ominaisuuksia. Mikäli nämä ominaisuudet, kuten kaiku tai poikkeuksellinen taajuusvaste, päätetään luoda vasta äänien jälkikäsittelyssä, voidaan äänitykset toteuttaa äänitysstudioissa.

Yksi studioäänittämisen suurimmista hyödyistä on se, että tallennettu äänite sisältää vain sen materiaalin, jota valmiiseen tuotokseen tarvitaan. Hyvä studio on suunniteltu ja rakennettu siten, että ulkoa tulevat äänet eivät juurikaan kantaudu sisätiloihin. Ylitse lentävät lentokoneet, katuporat tai ukkonen voivat kuulua normaalissa huoneistossa hyvinkin voimakkaana, mutta kun äänitetään, ei tällaisia ääniä saa äänitteelle tallentua. Lisäksi häiriöäänet, kuten tietokoneen hurina tai loisteputkista kuuluva sirinä, on minimoitu tai eliminoitu kokonaan. Ihmisistä kantautuvat esimerkiksi verenkiertoon liittyvät äänet voivat myös tallentua suurella herkkyydellä äänitetylle äänitteelle, mutta niitä ei tietenkään pystytä täysin vaimentamaan. Kaikkia tällaisia ylimääräisiä ja ei-toivottuja ääniä kutsutaan yleisesti meluksi. (Alten 2011, 26.) Melun poistamisen tärkeys korostuu, kun äänitetään kaikista hiljaisimpia ääniä. Vaikuttaakseen realistiselta asuntoon sijoittuva virtuaaliympäristö saattaa vaatia ääniraidaksi ilmastoinnin huminan tai ulkotiloihin sijoittuva ympäristö tuulenvireen, mutta taustamelun ollessa liian voimakas ei kyseisiä ääniä voida taltioida. Jos melun lähteitä ei voida hiljentää, tulee kyseeseen äänieristyksen parantaminen ja akustiikan muokkaaminen.

3.1.2 Akustiikka

Usein sanalla *akustiikka* tarkoitetaan tilan akustisia ominaisuuksia eli niin sanottua huonesoundia. Sanan varsinaisessa merkityksessä akustiikka tarkoittaa kuitenkin koko äänioppia, jonka yksi osa-alue liittyy tilan akustisiin ominaisuuksiin. Monesti tilan akustiikan vaikutusta äänitteen laatuun vähätellään tai se unohdetaan, vaikka tosiasia on, että sillä on keskeinen rooli koko äänijärjestelmässä. (Blomberg & Leppäluoto 1991, 14.) Äänitysstudioissa vallitsevaa akustiikkaa muokataan usein kalliilla komponenteilla ja isoilla ja aikaakin vievillä toimenpiteillä, jotta saavutetaan haluttu lopputulos. Tällaisia komponentteja voi olla esimerkiksi akustiikkalevyt, joista on esimerkki kuvassa 2. Äänityshuoneessa soittimien täytyy kuulostaa mahdollisimman hyvältä, jotta ne kuulostaisivat hyvältä myös äänitettynä. Lisäksi studion tarkkaamon akustiikan täytyy olla sellainen, että äänittäjä, miksaaja ja muut läsnäolijat saavat äänitteestä mahdollisimman realistisen kuvan ilman äänihäviötä tai ylimääräisiä taajuuksia. (Mäkelä 2002, 76.)



Kuva 2. Akustiikkaelementtejä (Thomann 2019)

Vaikka vallitsevan tilan akustisia ominaisuuksia voidaan mitata mittarein ja käyttää apuna akustiikkaan liittyvää teoriaa, on lopputulos kuitenkin mitattava aina kuuntelemalla. Äänittäjän sekä äänitysjärjestelmän ja äänitystilan suunnittelijan on koko äänitysprosessin alusta loppuun asti asetettava kuulijan asemaan. Se, miten ääniärsyke havaitaan ja minkälaisia tuntemuksia tai reaktioita se herättää, riippuu kuulijasta. Tällaista ääniopin inhimillistä ja fysiologista osa-aluetta kutsutaan psykoakustiikaksi. Kiteytettynä psykoakustiikkaan liittyy äänilähteen sijainnin havainnointi sekä vaikutelma siitä, minkälaisessa tilassa ollaan. Näihin aspekteihin pystytään vaikuttamaan usealla eri tavalla, joita voidaan simuloida sovelluskehitysohjelmassa. Aiheeseen paneudutaan tarkemmin ohjelmointia käsittelevässä luvussa.

3.2 Äänitystavat ja mikrofonit

Kun tilan akustiset ominaisuudet on saatu kuntoon, on seuraavaksi tehtävä muutamia itse äänittämiseen liittyviä valintoja. Ensinnäkin täytyy valita käytettävä äänitystapa ja seuraavaksi mikrofoni. Nämä kaksi seikkaa ovat hyvinkin läheisesti kytköksissä toisiinsa. Tietynlaisia mikrofoneja käytettäessä voidaan käyttää vain tietynlaisia äänitystekniikoita ja vastaavasti mitä tahansa tekniikoita ei voida toteuttaa millä tahansa mikrofonilla. Valinnat perustuvat ennen kaikkea äänitettävän kohteen ominaisuuksiin sekä ympäröiviin olosuhteisiin.

3.2.1 Perinteinen äänitysmenetelmä

Virtuaaliympäristöä varten äänitettäessä yleisimmäksi äänitystavaksi nousee perinteinen äänitysmenetelmä. Virtuaaliympäristöjä on toki monia erilaisia, ja ne saattavat vaatia edistyneempiäkin äänitystapoja. Perinteisellä menetelmällä pystytään kuitenkin äänittämään

suurin osa tarvittavasta äänimateriaalista, sen ollessa yleispätevä metodi kaikenlaiseen äänittämiseen. Äänittäjän olisikin hyvä syventää osaamistaan vähintäänkin juuri tämän äänitystekniikan osalta.

Perinteinen äänitysmenetelmä on toimintaperiaatteiltaan hyvin yksinkertainen: Kun äänitään jonkin instrumentin soittoa, asetellaan mikrofoni soittajaan nähden sellaiselle etäisyydelle, jossa soiton kuuntelijan voisi kuvitella istuvan. Tällöin on tärkeää, että tilan akustiikka on hyvä, sillä äänitteelle tallentuu myös huonesoundi. (Mäkelä 2002, 103.) Sama pätee äänitehosteisiin. Teoriassa laadukas mikrofoni tallentaa kaiken sen, mitä myös terve ihmiskorva kuulee. Tämä tarkoittaaakin sitä, että virtuaaliympäristöä varten äänitetty raita saattaa kuulostaa hyvinkin realistiselta jo sellaisenaan, mikäli äänitystila vastaa virtuaalista ympäristöä akustisilta ominaisuuksiltaan.

3.2.2 Kondensaattorimikrofoni

Perinteisellä menetelmällä äänitettäessä kaikista sopivin valinta mikrofoni on kondensaattorimikrofoni. Tähänkään tapaukseen ei ole yhtä ja ainoa oikea mikrofonia, sillä äänitystilanteita on ääretön määrä, ja lopullinen ratkaisu on edelleen tehtävä kuuleman perusteella. J. Pekka Mäkelä (2002, 90) kirjoittaa kirjassaan *Kotistudio* kondensaattorimikrofonien korkeista hinnoista. Teos on kuitenkin yli 15 vuotta vanha, ja ajan saatossa näiden mikrofonien hinnat ovatkin laskeneet huomattavasti. Internetin verkkokaupoissa, ja miksei musiikkiliikkeissäkin, on runsas valikoima edullisia kondensaattorimikrofoneja, jotka eivät laadultaan häviä merkittävästi huippuluokan mikrofoneille. Toki saatavilla on edelleen eri mikrofoni valmistajien klassikkomalleja, joiden hinnat ovat edelleen korkealla. Näitä mikrofoneja ostavat lähinnä ammatikseen äänitystöitä tekevät ihmiset.

Kondensaattorimikrofonit voidaan jo ulkonäkönsä mukaan jakaa kahteen ryhmään: suurikalvoisiin ja pienikalvoisiin mikrofoneihin. Kyse ei kuitenkaan ole pelkästään estetiikasta, vaan erilaisia kondensaattorimikrofoneja käytetään eri tarkoituksiin. Pienikalvoisilla voidaan äänittää soittimia sekä luoda tilan tuntua. Käytännössä tämä tarkoittaa sitä, että pienikalvoista kondensaattorimikrofonia käytetään yhtä aikaa luvussa 3.2.3 esiteltävän lähimikrofonin kanssa, ikään kuin sen parina. (Mäkelä 2002, 91 - 92.) Toki pienikalvoisella kondensaattorimikrofonilla voidaan äänittää myös pelkästään, mikäli näin saadaan aikaisiksi hyväkuuloisia tuotoksia.

Suurikalvoiset kondensaattorimikrofonit sen sijaan ovat omiaan puhetta tai laulua taltioitaessa, mutta toki niitä voi käyttää myös instrumenttien ja äänitehosteiden äänittämiseen. Tällaisille mikrofoneille ominaista on niiden erottelevaisuus ja täyteläisyys. Ne myös poimivat voimakkaasti matalia taajuuksia sekä teräviä ääniä, kuten konsonanteja ja erilaisia

naksahduksia, joten äänitettäessä on syytä käyttää pop-filtteriä. (Mäkelä 2002, 92.) Kuvassa 3 on nähtävissä jatkojohdon kytkimen naksahduksen äänitystilanne suurikalvoisella kondensaattorimikrofonilla.



Kuva 3. Äänittäminen suurikalvoisella kondensaattorimikrofonilla

Molemmille kondensaattorimikrofonityypille ominaisia tunnuspiirteitä on niiden kyky poimia ääni hyvinkin yksityiskohtaisesti. Näin tehosteita äänitettäessä saadaan aikaan selkeä ja kirkas äänite. Toisaalta, koska pienetkin yksityiskohdat tallentuvat äänitteelle, täytyy tilan akustiikan ja äänieristyksen sekä itse äänilähteeseen liittyvien seikkojen olla kunnossa. Kondensaattorimikrofonilla kuoroa äänitettäessä yksittäisenkin ihmisen ässävika kuuluu äänitteellä. (Mäkelä 2002, 91.) Sama pätee tilanteeseen, jossa äänitetään esimerkiksi oven sulkeutumisaäntä. Saranan ei-toivottu narina sekä tuulenvireestä tai äänenpaineesta johtuva juomalasien kilinä on eliminoitava ennen kuin voidaan äänittää.

Toinen yhdistävä tekijä kaikenlaisille kondensaattorimikrofoneille on niiden lisävirran tarve. Oletuksena mikseristä, ulkoisesta äänikortista tai tallentimesta voidaan kytkeä normaalisti 48 voltia oleva phantom-virta päälle, mikä mahdollistaa kondensaattorimikrofonin käytön. Mikäli tällaista mahdollisuutta ei ole, voidaan käyttää niin sanottua keinokuormaa, joka liitetään signaaliketjuun mikrofonin ja tallentimen väliin. Tietyissä kondensaattorimikrofoneissa on tila myös paristolle. Tällöin mikrofoni saa luonnollisestikin virtansa siitä.

3.2.3 Lähimikrofonitekniikka

Lähimikrofonitekniikka, tai arkikielessä *lähimikitys*, on nimensä mukaan tekniikka, jossa mikrofoni asetetaan äänilähteen lähelle. Kyseinen tapa on suosittu esimerkiksi silloin, kun vahvistetaan puheen pitäjän ääntä. Studioissa lähimikityksellä haetaan juurikin läsnäolon tunnetta ja yritetään välttyä huonesoundin liialliselta korostumiselta ja muilta ympäröiviltä ääniltä. Kuvassa 4 äänitetään jatkojohdon kytkimen painamisesta kuuluvaa naksahdusta.



Kuva 4. Lähimikrofonitekniikka ja dynaaminen mikrofoni

Lähimikitystekniikkaa käytettäessä ympäriltä mikrofoniin kantautuva melu on paljon vähäisempää, kuin perinteisellä menetelmällä äänitettäessä. Tämä tarkoittaakin sitä, että lähimikitettyjen instrumenttien äänenvoimakkuuksia on helpompi kontrolloida ja sen takia tämä tekniikka onkin toimiva ja suosittu ratkaisu myös live-tilanteissa, joissa ympäröivä melu on saatava minimoitua. (Alten 2011, 188.) Virtuaalitodellisuussovellusta varten äänitettäessä voidaan ajatella, että lähimikitys tulee kyseeseen aina silloin, kun ympäröiviä ääniä ei voida eliminoida ja halutaan rajata äänitteen sisältämä materiaali johonkin tiettyyn ääneen. Esimerkkinä tästä auton äänitorven äänen tallentaminen ilman, että tuulen humina tai ohi kulkevasta liikenteestä kantautuva ääni tallentuu äänitteelle. Toisaalta lähimikitykseen voidaan päätyä jo pelkästään sen takia, että se kuulostaa kyseisessä yhteydessä paremmalta.

3.2.4 Dynaaminen mikrofoni

Lähimikrofonitekniikkaa toteuttamaan ryhdyttäessä oivallinen ratkaisu on käyttää dynaamista mikrofonia. Dynaamisen mikrofonin toiminta perustuu siihen, että äänen paineesta värähtelevä kalvo muodostaa mikrofonin kelaan sähköjännitteen muutoksia. Kela taas on kytketty johdoilla mikrofonin esivahvistimeen, joka vahvistaa signaalin. Dynaamiset mikrofonit ovat pääsääntöisesti kestävämpiä ja paremmin painetta sietäviä kuin kondensaattorimikrofonit. Toisaalta niillä ei saada aikaan yhtä erottelevaa ja selkeää tallennetta äänestä, sillä usein ne korostavat tiettyä taajuusaluetta, tai hävittävät joitakin taajuuksia tai yksityiskohtia. Dynaamisia mikrofoneja käytetään kuitenkin karkeasti arvioituna yhtä paljon live- kuin studiotilanteissakin. (Mäkelä 2002, 89 - 90.)

Lisäksi on monia muitakin niin toimintaperiaatteeltaan, kuin käyttötarkoitukseltaankin erilaisia mikrofoneja, mutta ne eivät kuitenkaan ole tämän opinnäytetyön aiheen kannalta välttämättömiä. Kondensaattorimikrofoni sekä dynaaminen mikrofoni ovat tämän päivän kaksi yleisintä mikrofonityyppiä, ja niillä pystytäänkin toteuttamaan virtuaalitodellisuussovelluksen äänimaailma varsin menestyksekkäästi. Myöskään mikrofonien toimintaan ei syvennytä tarkemmin, sillä se ei ole oleellista aiheen kannalta.

3.2.5 Ympäröivän äänen tallennus ja nelikanavainen ääni

Jotta virtuaalitodellisuussovellus saadaan tuntumaan realistiselta, on siihen lisättävä ympäristössä vallitseva ääni. Ympäröivä ääni on virtuaalitodellisuussovellukselle tärkeä, sillä mikään tila ei reaali maailmassakaan ole täysin äänetön. Ympäröivä ääni voidaan äänittää jälkikäteen, jolloin esimerkiksi ulkotiloihin sijoittuvaa kohtausta varten äänitetään tuulen huminaa. Tämä ääniraita lisätään muiden äänitehosteiden taustalle. Tällaiset äänitehosteet on prosessoitava jälkikäteen, sillä reaali maailmassa ne sisältävät tilalle ominaisen taajuusvasteen ja kaiun. Toki äänitehosteeseen voidaan luoda ympäristön tuntua myös niin, että se toistetaan tarpeeksi korkealaatuisesta kaiuttimesta ja äänitetään uudelleen, jolloin äänitteelle tallentuu tilan huonesoundi. (Alten 2011, 484.) Tämä tapa ei kuitenkaan sovi virtuaalitodellisuuden äänituotantoon, sillä valmiissa sovelluksessa käyttäjällä on valittavissa ääretön määrä mahdollisia sijainteja, joiden mukaan akustisten ominaisuuksien, kuten kaiun ja taajuusvasteen, on elettävä. Jos pelaaja menee esimerkiksi tunneliin, on hänen korviinsa kantautuvissa äänissä oltava kaikua. Jos äänilähde on kaukana, on siitä leikattava ylätaajuuksia pois.

Työkaluja neliraitaisen äänen taltioimiseen on vielä tämän opinnäytetyön kirjoitusvaiheessa melko vähän tarjolla. Euroopan suurimman musiikkiliikkeen, Thomannin, verkkokaupasta löytyy kuitenkin muutamia vaihtoehtoja, joilla tällaista ääniraitaa on mahdollista

taltioida. Yksinkertaisimmalta vaikuttava tapa olisi hankkia esimerkiksi *Ambisonic microphone* -nimikkeellä myytävä Røden NT-SF1-merkkinen mikrofoni. Sanalle *ambisonic* ei toistaiseksi ole vakiintunut suomennosta. Kyseinen malli sisältää neljä mikrofoniakapselia, joiden taltioima signaali ohjataan esivahvistimen kautta nelihaaraiseen XLR-johtoon. Tallennettu äänite sisältää siis joka tapauksessa neljä yksittäistä monoäänikanavaa. Kuvassa 5 on havaittavissa mikrofoniakapselien tetraedrin muotoinen asettelu, jonka ansiosta ne pystyvät ainakin Røden oman verkkosivun mukaan poimimaan ääntä tasapuolisesti joka suunnasta. Kyseisen mikrofoniin kapselit on toteutettu kondensaattorimikrofonitekniikalla. (Røde Microphones 2019.)



Kuva 5. Røde NT-SF1 (Røde Microphones 2019)

3.3 Synteettinen ääni

Välttämättä kaikkia ääniä ei kuitenkaan kannata ryhtyä äänittämään. Jos jokin ääni on helpompi luoda synteettisesti, kuin äänittämällä, kannattaa tätä vaihtoehtoa harkita jo ajan säästämisenkin vuoksi. Loppujen lopuksi pääprioriteetti äänimaailman tuottamisessa on

äänien realistisuudella. Tässä tapauksessa äänitekniikan täytyykin harkita, onko järkevää pystyttää äänityskalustoa esimerkiksi tuulen huminan äänittämistä varten vai saadaanko synteettisin menetelmin aikaan yhtä realistisia tuloksia.

Yksi vanhimmista synteettisen äänen lähteistä on nimeltään oskillaattori. Kyseinen laite kuuluu laboratoriovarusteisiin, mutta äänen tuottamiseen sitä on käytetty jo kauan ennen kuin kuluttajalla oli resursseja hankkia oskillaattoria hyväksi käytettävää syntetisaattoria. Alkeellisimmat oskillaattorit tuottavat vain siniaaltoja, mutta tänä päivänä niillä voidaan saada aikaan useampia eri aaltomuotoja. Koska oskillaattori on alun perin suunniteltu äänen testaamiseen, oli sen käyttäminen musiikin tuottamisessa hankalaa muun muassa sävelkorkeuden epävakaisuuden ja häiriöalttiuden vuoksi. Oskillaattoriperiaatteella toimiva syntetisaattori tarjoaa kuitenkin nopean ja vaivattoman tavan tuottaa ääntä, mikäli nämä ongelmat eivät nouse esteiksi. (Russ 2009, 99, 256.)

Digitaaliset syntetisaattorit sen sijaan eivät kärsi analogisille syntetisaattoreille tyypillisistä epäpuhtauksista. Ne ovat usein hankalampia käyttää, mutta niillä voidaan saavuttaa monipuolisempia lopputuloksia. Digitaaliset syntetisaattorit korvaavat analogisen signaalin numeerisella muodolla ja käyttävät tietokonetta prosessoimaan tätä dataa. Virtuaalituodellisuuden äänisuunnittelijan kannattaneen kuitenkin syntetisaattorin sijaan kääntyä ääntä generoivan tietokonesovelluksen puoleen. Tällaiset sovellukset tuottavat halutunlaista ääniaaltoja, jota voidaan yhdistää toisenlaiseen aaltoon tai ajaa signaaliprosessorin läpi, jolloin voidaan saavuttaa laaja kirjo erilaisia lopputuloksia. (Russ 2009, 11.)

3.4 Signaaliprosessorit

Kronologisesti seuraava vaihe on prosessoida äänitetty tai muutoin luotu ääniraita, ellei niin tehty jo äänitysvaiheessa. Signaaliprosessoreita on niin fyysisiä laitteita, kuin äänitysohjelmaan liitettäviä sovelluslisäosiaakin. Fyysisiä prosessoreita, kuten alataajuusleikkuria, käytetään usein jo äänitysvaiheessa, mutta sovelluslisäosilla signaali prosessoidaan äänityssovelluksessa jälkikäteen. (Case 2007, 55, 122.) Suuri hyöty virtuaalisien signaaliprosessorien käytössä on se, että niiden asetuksia voidaan muuttaa, samalla kun äänitettyä raitaa toistetaan. Tällöin tiedetään tarkalleen, miten tehty muutos vaikuttaa äänitteen sisältöön, sillä se kuullaan reaaliajassa. Virtuaalituodellisuussovelluksessa signaaliprosessorit luovat tilan tuntua, tekevät äänitehosteista selkeämpiä tai luovat niihin mielenkiintoisia vaikutelmia.

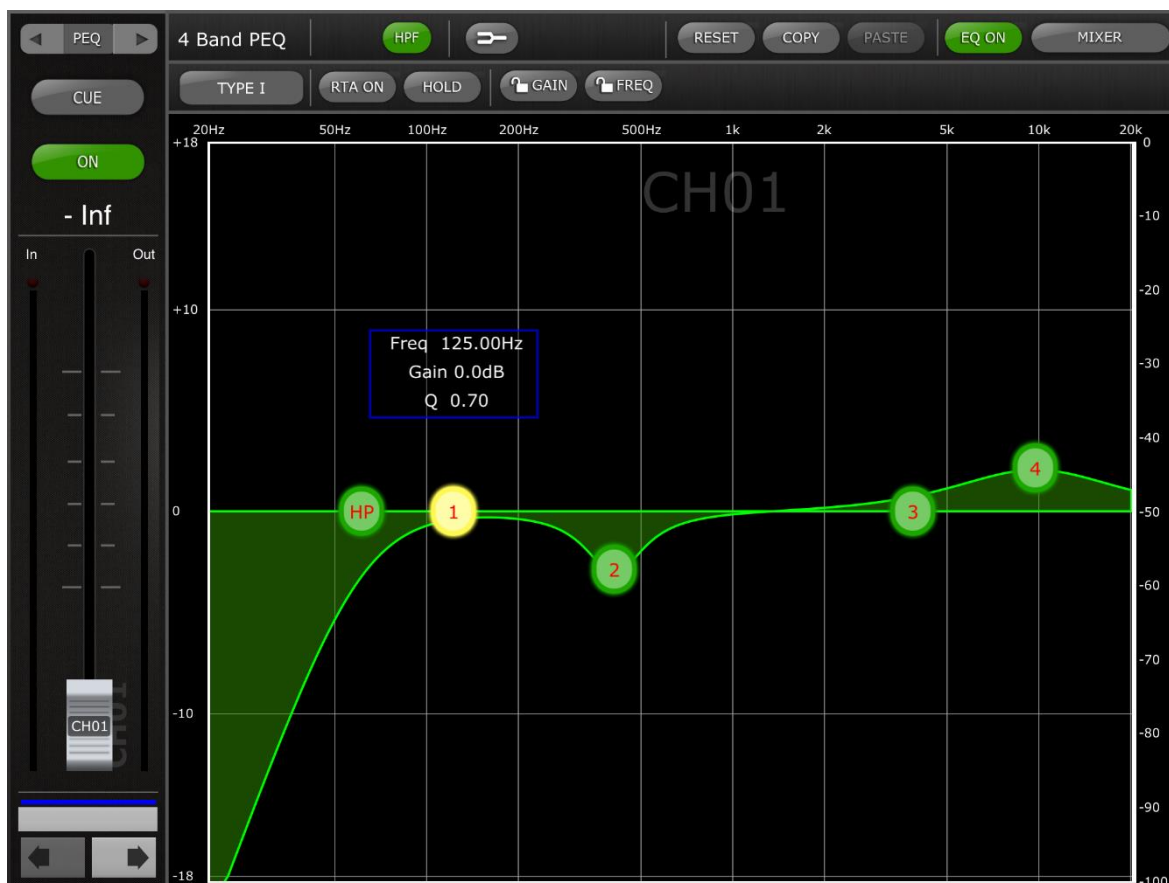
Puhekielessä signaaliprosessoreista käytetään usein nimitystä efekti, mutta koska tämä nimitys on helposti sotkettavissa äänitehosteisiin, käytetään kaikista äänisignaalia muokkaavista komponenteista – niin fyysisistä laitteista, kuin sovelluslisäosistakin – nimitystä

signaaliprosessori. Signaaliprosessorin aikaansaamaa vaikutusta nimitetään kuitenkin efektiksi. Äänitehosteella taas tarkoitetaan lyhyehköä ääniraitaa, jonka on tarkoitus korostaa materiaalin visuaalista puolta.

3.4.1 Taajuuskorjain ja leikkurit

Seuraavista signaaliprosessoreista yleisimmin ihmisten tiedossa lienee taajuuskorjain. Tämä työkalu löytyy äänityssovelluksien ja mikserien lisäksi tavallisista kotiteatterijärjestelmistä, vahvistimista ja miksei jopa televisioistakin. Nimitys ”taajuuskorjain” on kuitenkin hieman harhaanjohtava. Aikoinaan taajuuskorjaimia käytettiin juurikin taajuusvasteen vääristymien korjaamiseen, eli signaalin eri taajuuksien korostumisen hillintään tai voimakkuudeltaan heikkojen taajuuksien korostamiseen. Nykyään taajuuskorjaimilla pyritään myös elävöittämään äänisignaalia tai muokkaamaan siitä muuten mielenkiintoinen. (Case 2007, 13.)

Kaikista monipuolisista taajuuskorjaintyyppiä on nimeltään parametrinen taajuuskorjain. Siinä on kolme eri parametriä, joiden avulla äänisignaalin sisältämiä taajuuksia voidaan säätää. Ensimmäinen näistä on taajuuden valinta. Tällä parametrilla taajuuskorjauspisteelle valitaan taajuus, jota halutaan säätää. Seuraavana valitaan taajuuden äänenvoimakkuus. Jos miksaajasta tuntuu, että ääniraidassa on liikaa bassoa, valitsee hän jonkin matalan taajuuden, ja vähentää sitä. Toisaalta jos ääni vaikuttaa kimeältä, voisi tulla kyseeseen ylätaajuuksien voimakkuuden vähentäminen. Viimeisimpänä säätimenä on korjauspisteen kaistanleveys. Kaistanleveys määrittää, kuinka paljon vaimennus tai korostus vaikuttaa valitun taajuuden vieressä oleviin taajuuksiin. Toisin sanottuna, kuinka laajan taajuusalueen ope- roitavana oleva taajuuskorjauspiste sisältää. Kaistanleveyttä säätämällä voidaan valita, vaikutetaanko koko bassoalueeseen, vai vaan tiettyyn bassotaajuuteen, joka aiheuttaa esimerkiksi epätoivottua resonointia. (Case 2007, 106 - 107.) Kuvassa 6 on havainnollistava esimerkki parametrisesta taajuuskorjaimesta sovelluslisäosana eräässä digitaalisessa mikserissä.



Kuva 6. Parametrinen taajuuskorjain

Hieman yksinkertaisempi taajuuskorjain on nimeltään graafinen taajuuskorjain, jossa korjauspisteitä on yleensä parametristä taajuuskorjainta enemmän. Käyttäjä ei kuitenkaan voi määrittää niille taajuuksia eikä kaistanleveyttä. Lisäksi kaikista matalin ja korkein taajuus on usein toteutettu hieman eri tavalla. Jos esimerkiksi 80 hertsin kohdalla olevaa painiketta painaa, häviävät kaikki sen alla olevat taajuudet. Ylätaajuuksien puolelta löytyvä painike sen sijaan toimii päinvastoin. Näitä säätimiä kutsutaankin toimintojensa mukaan leikkureiksi tai yli- ja alipäästösuotimiksi. (Case 2007, 111 - 112.) Esimerkki graafisesta taajuuskorjaimesta on havaittavissa kuvassa 7.



Kuva 7. Graafinen taajuuskorjain

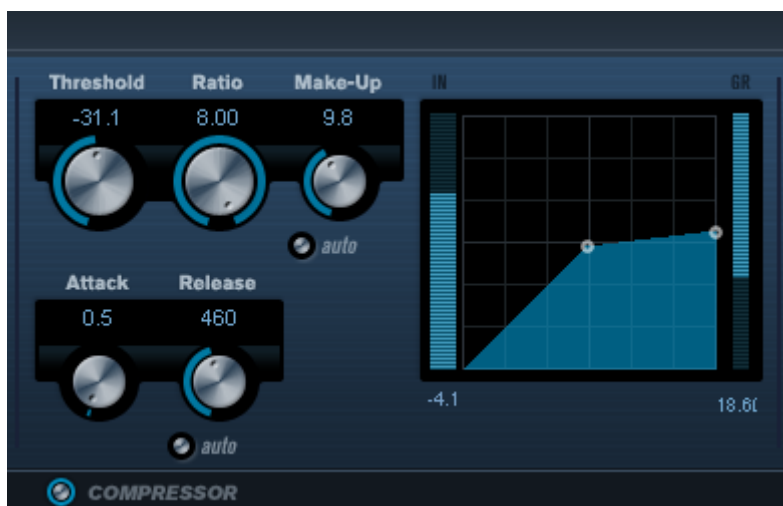
3.4.2 Kompessori

Äänenvoimakkuuden säätö lienee yleisin virtuaalitodellisuussovelluksen äänimaailman muokkaamiseen liittyvistä osa-alueista. Jos jokin ääni ei kuulu muiden taustalta, voidaan sen voimakkuutta nostaa, ja päinvastoin. Joskus saatetaan kuitenkin tulla tilanteeseen, jossa itse ääniraidan dynaaminen alue on liian laaja. Kiteytettynä tämä tarkoittaa sitä, että hiljaisimmat äänet ovat niin hiljaisia, etteivät ne kuulu taustamusiikin alta. Toisaalta jos ääniraitaa vahvistaa, sen voimakkaimmat äänet korostuvat liikaa. Ihmisen puhetta äänitettäessä äänenpaino saattaa aiheuttaa suuremmalla voimakkuudella sanottujen tavujen liiallista korostumista tai hiljempaa sanottujen tavujen hukkumista muiden taustaäänien alle. Tällaisia liiallisia äänisignaalin dynaamisia eroja voidaan tasoittaa kompressorilla. Yksinkertaisuudessaan kompressorin on siis vain äänikanavan voimakkuuden säätöön suunniteltu automaattinen komponentti. Se, miten voimakkaasti, nopeasti tai herkästi kompressorin toimii, on ääniteknikon päätettävissä. (Case 2007, 132.)

Kuten taajuuskorjaimessa, on myös kompressorissa muutamia säätimiä. Ensimmäisenä näistä säätimistä kannattaneen säätää kompressorin kynnyksen (*engl. Threshold*). Tällä säätimellä valitaan desibelimäärä, joka toimii raja-arvona kompressorin aktivoitumiselle. Jos äänisignaali menee kynnyksen yli, sitä vaimennetaan. Vaimennus lakkaa vasta kun signaali palaa takaisin kynnyksen alle. Se, miten paljon signaalia vaimennetaan, on säädettävissä "suhde"-säätimellä (*engl. Ratio*). Tämä suhde vaikuttaa ainoastaan kynnyksen yli menevään ääneen. Jos desibelit pysyvät kynnyksen alapuolella, ei suhdeluvullakaan ole mitään vaikutusta äänisignaaliin. Kompressoinnista voidaan puhua, kun suhdeluku on yli

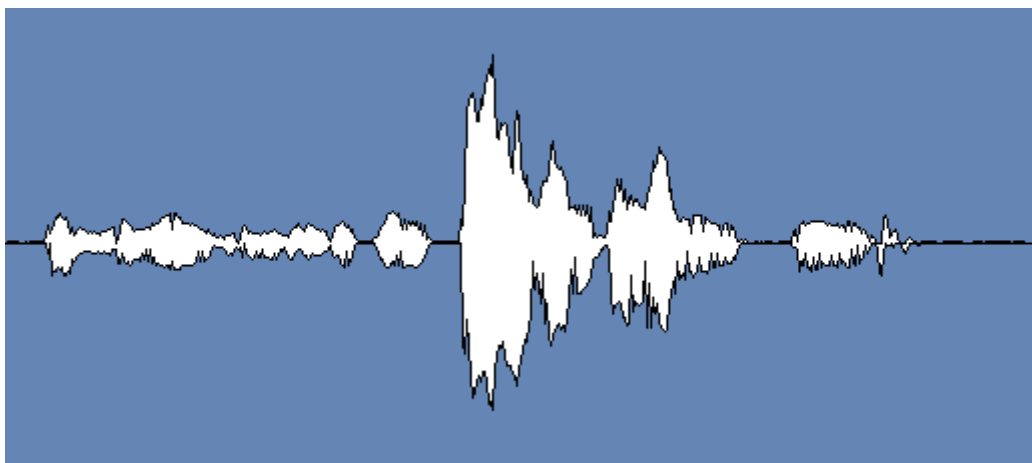
1:1, mutta alle 10:1. Suurempiin suhdelukuihin mentäessä, on kyse ennemminkin rajoittimesta (*engl. Limiter*). Tämän vuoksi moni kompressorivalmistaja myykin tuotteitaan *Compressor/Limiter* -nimikkeellä. (Case 2007, 133.)

Seuraavaksi kompressorista tai rajoittimesta säädetään sen toimintanopeus (*engl. Attack*). Tällä säätimellä säädetään aika, joka kuluu signaalin hiljentymiseen asetetun suhteen määräämälle tasolle sen noustua kynnyksen yli. Kun signaalin voimakkuus laskee takaisin kynnyksen alle, lakkaa kompressorin toiminta. Tällekin toiminnolle voidaan määrittää viive (*engl. Release*). Lopuksi valitaan kompressorin lisäämä äänenvoimakkuuden vahvistus (*engl. Out gain/Make-up gain*). Vahvistuksella pyritään palauttamaan signaalin yleisäänvoimakkuus sille tasolle, jolla se oli ennen kompressointia. (Case 2007, 134.) Kuvassa 8 on havaittavissa edellä mainitut säätimet sisällytettynä virtuaaliseen kompressoriiin.

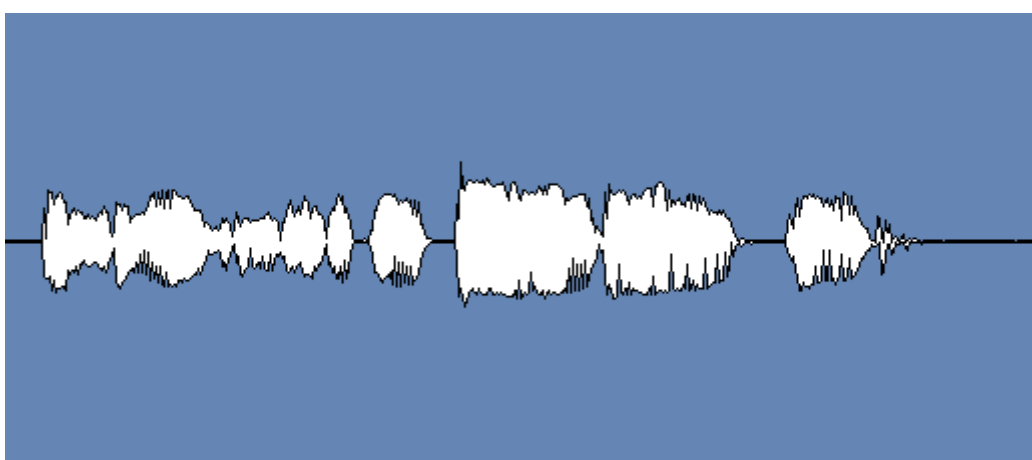


Kuva 8. Virtuaalinen kompressorii

Harjaantuneemmalla korvalla kompressorin vaikutus on helppo kuulla, vaikka sen asetukset olisivatkin säädetty maltillisesti, mutta esimerkiksi musiikkikappaleissa sen aikaansaama vaikutusta ei kuitenkaan välttämättä aina edes havaita. Seuraavissa kuvissa on esitettyä aaltomuoto, jossa signaalin voimakkuus on esitetty ajan funktiona. Äänite sisältää puhutun lauseen, jonka dynamiikka-alueita puhuja on tarkoituksella laajentanut sanomalla keskellä lausetta olevan sanan suuremmalla voimakkuudella. Kuvassa 9 oleva signaali ei sisällä minkäänlaista prosessointia, mutta kuvassa 10 signaali on kompressoitu voimakkaasti. Aaltomuodon korkeinta huippua on rajoitettu vahvalla kompressoinnilla ja yleistä äänenvoimakkuustaloa on sen jälkeen vahvistettu 5 desibelin verran. (Case 2007, 134.)



Kuva 9. Puhutun lauseen dynamiikka ilman kompressointia



Kuva 10. Sama ääniraita kompressoituna

3.4.3 Kaiku

Tilan tunnun luominen on yksi tämän opinnäytetyön tärkeimmistä päämääristä. Tilan tuntu voidaan saada aikaisiksi pelkästään taajuuskorjaimella, mutta usein tallenne vaatii niin sanotusti väljyyttä, jotta se vastaisi virtuaaliympäristön oletettuja akustisia ominaisuuksia. Väljyyden tunteen voi saada aikaan keinotekoisestikin. Saatavilla on lukuisia erilaisia kaikulaitteita ja -sovelluslisäosia. Koska kaiku itsessään on yleisesti ymmärretty ilmiö, ei sitä sen tarkemmin käydä läpi. Kaikulaitteen tuottama kaiku on muokattavissa useiden eri säätimien avulla, joten seuraavassa tekstikappaleessa käydään läpi kaikulaitteisiin tai kaikua simuloiviin sovelluslisäosiin liittyvät oleelliset parametrit.

Näistä parametreista helpoiten ymmärrettävä lienee kaikuma-aika. Kaikuma-aika on tarkemmalta nimeltään RT_{60} , jolla tarkoitetaan sitä aikaa, joka kuluu kaiun vaimenemiseen äänilähteen voimakkuuden ollessa 60 desibeliä. 5 sekunnin RT_{60} -arvolla tarkoitetaan käy-

tännössä siis sitä, että esimerkiksi muuten täysin hiljaisessa kirkossa 60 desibeliä voimakkuudeltaan olevan äänen jälkikaiunta-aika olisi 5 sekuntia, jonka jälkeen on taas täysin hiljaista. (Case 2007, 265.)

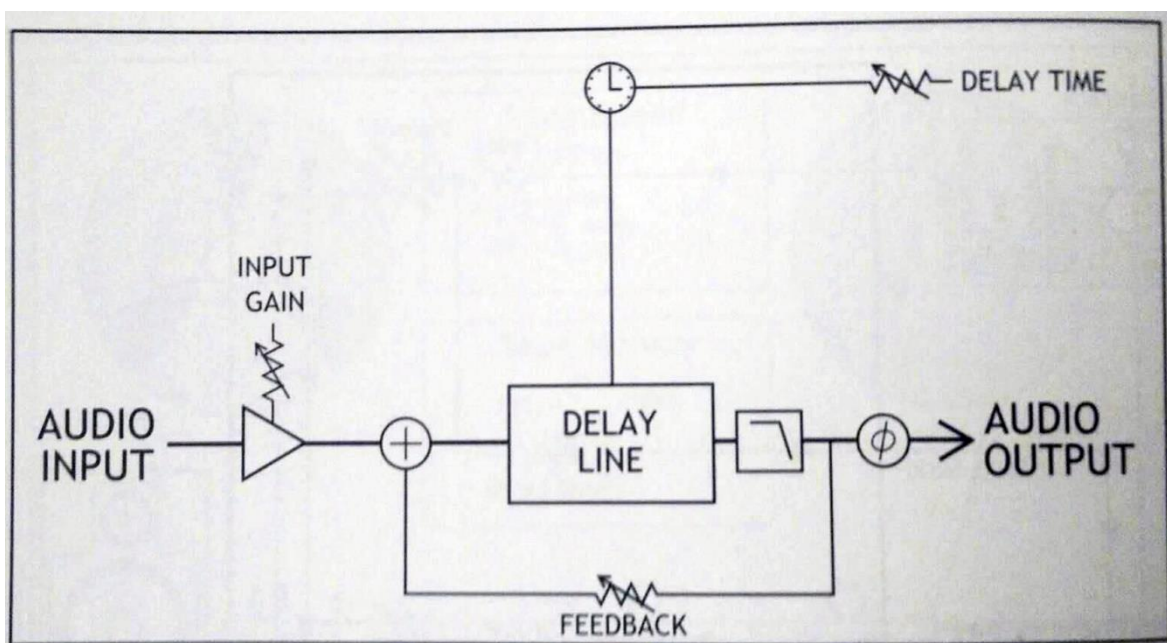
Myös jälkikaiunnan äänellinen sisältö on ääniteknikon säädettävissä. Oletuksena kaiussa kuullaan ne kaikki taajuudet, jotka musiikin katkeamishetkellä ovat aktiivisia. Tämä tarkoittaa sitä, että jos kaiunnan lähteenä on pelkkä bassokitara, sisältää se pääosin matalia taajuuksia. Toisaalta jos kaiunnan aiheuttaa naiskuoro, on myös kaiku sisällöltään korkea ja heleä. Kaiun pituutta eri taajuusalueilla mitataan oktaaveissa. Esimerkiksi arvo RT_{1000} kertoo, kuinka pitkään kaiku jatkuu oktaavilla, jonka keskikohta on 1000 hertsin kohdalla. Samoin tarkastellaan 500, 250 ja 125 hertsin taajuusalueita. Bassotaajuuksien suhdeluvuksi (*engl. Bass ratio*) kutsutaan lukemaa, joka saadaan RT_{125} :n ja RT_{250} :n summasta jaettuna RT_{500} :n ja RT_{1000} :n summalla. Tämä johtaa siihen, että matalien taajuuksien kaikuessa pidempää, bassotaajuuksien suhdeluku on suurempi kuin yksi, ja päinvastoin. (Case 2007, 266 - 267.)

Kun kaikuisassa tilassa lyödään kädet yhteen, kuullaan ensin käsistä suoraan lähtevä ääni, jota seuraa kaiku. Koska ääni on fyysinen ilmiö, kestää sillä aikaa myös etenemiseen. Tämä fysiikan laista johtuva ilmiö on lähes poikkeuksetta sisällytetty myös kaiuntalaitteisiin ja -sovelluksiin, eli kaiulle pystyy määrittämään viiveajan. Tämä parametri kuvastaa sitä aikaa, joka kuluu, ennen kuin äänienergia kulkee kaiuntapintaan, eli esimerkiksi seinään, ja siitä takaisin kuulijan korvaan. Yksinkertaisuudessaan se on siis tyhjä väli alkuaänen ja siitä johtuvan kaiun välissä. Kaiun viivettä säädetään yksinkertaisesti ajan funktiona, joten käyttäjän on helppo ymmärtää se. Isommille tiloille on asetettava suurempi viiveaika, pienemmille lyhyempi. Ääniteknikon täytyy kuitenkin muistaa, että myös katto voi toimia kaiuntapintana, eli jos tila on matala, on myös kaiunnan viive asetettava lyhyeksi, vaikka tilan pinta-ala olisi laaja. (Case 2007, 267.) Kuvassa 11 on esimerkki virtuaalisesta kaikulaitteesta.

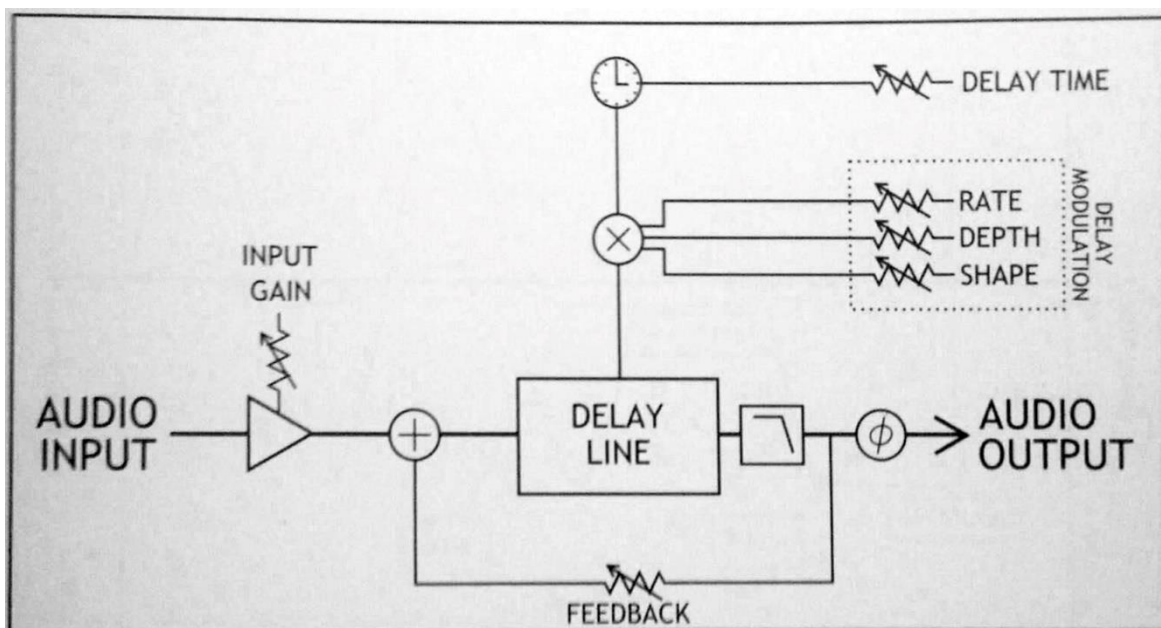


Kuva 11. Esimerkki virtuaalisesta kaikulaitteesta

Hieman samankaltaisella periaatteella toimivaa signaaliprosessia, viivettä, voidaan käyttää myös tilan tunnustamiseen. Yksinkertaisuudessaan viive kopioi signaalin ja toistaa sen uudelleen käyttäjän määrittämien parametrien mukaan. Käyttäjän päätettävissä on, kuinka suurella voimakkuudella signaali syötetään laitteeseen, kuinka pitkän ajan päästä se toistetaan uudelleen, kuinka paljon siitä hävitetään joka toiston aikana ja kuinka suurella voimakkuudella se ajetaan ulos laitteesta. Edistyneemmillä viivelaitteilla pystytään moduloimaan signaalia, jolloin sen toistama signaali saadaan kuulostaan erilaiselta, kuin lähtösignaali. Kuvissa 12 ja 13 on nähtävillä esimerkit kahdesta erilaisen viivelaitteen toiminnasta.



Kuva 12. Viivelaitteen toimintakaavio (Case 2007)



Kuva 13. Viivelaite, jossa myös ääntä moduloivia ominaisuuksia. (Case 2007)

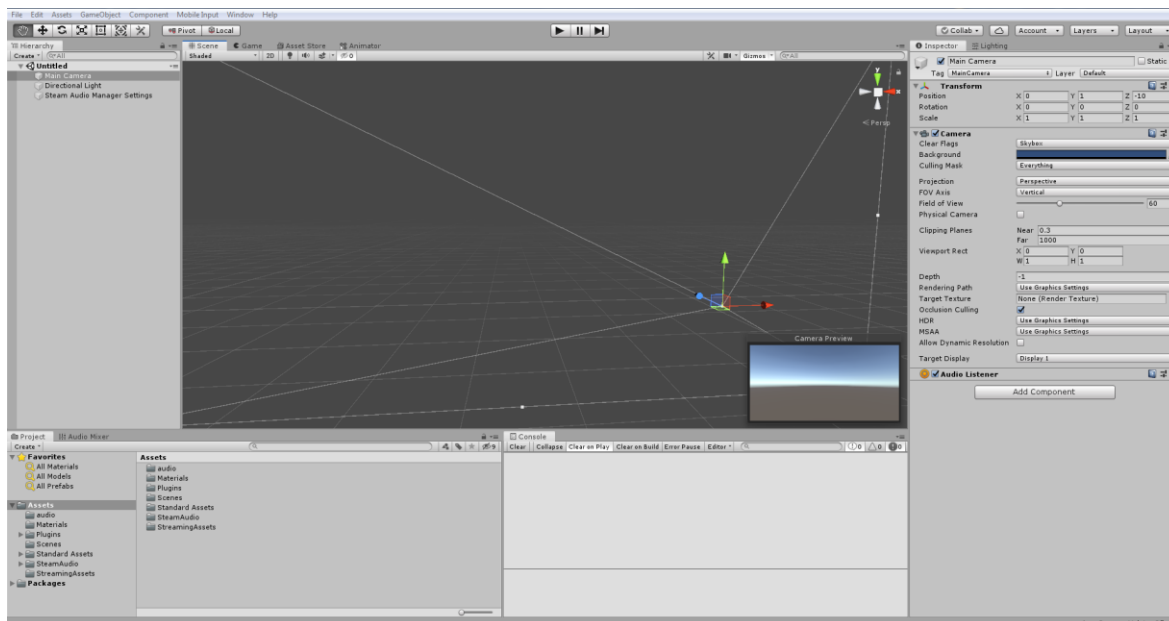
Signaaliprosessorit eivät suinkaan rajoitu vain edellä mainittuihin laitteisiin. Etenkin soittimien tuottamaa signaalia muokataan mitä erilaisimmilla laitteilla. Saatavilla on niin äänisignaalin vaihetta muokkaavia, viivästäviä ja toistoon perustuvia, kuin sen tarkoituksellisesti yliohtaavia ja särkeviäkin signaaliprosessoreja. Reaalimaailmasta tällaiset vaikutukset ovat kuitenkin vähemmän tunnettuja ja siksi näihin prosessoreihin ei paneuduta sen tarkemmin, sillä työn tarkoituksena on luoda mahdollisimman realistinen virtuaaliympäristö autenttisine äänineen. Alexander U. Case (2007) käy läpi laajan valikoiman erilaisia signaalia muokkaavia laitteita kirjassaan *Sound FX – Unlocking the Creative Potential of Recording Studio Effects*, johon aiheesta kiinnostuneen kannattanee tutustua.

4 OHJELMOINTI

4.1 Unity-pelimoottori

Viimeisessä työvaiheessa on aika ottaa äänitetyt raidat käyttöön, eli istuttaa ne virtuaaliympäristöön. Tässä vaiheessa äänitiimin ohjelmoija ottaa työn vastuulleen. Hänen tehtävänsä on saada äänet kuulostamaan realistisilta myös mallinnetussa ympäristössä. Tämä tarkoittaa käytännössä sitä, että äänien on vastattava niiden mallinnettuja lähteitä sekä vallitsevia olosuhteita, kuten etäisyys ja vallitseva ympäristö. Kuten jo opinnäytetyön otsikossa mainitaan, käytetään pelimoottorina Unitya, jonka peruskäyttöliittymä on esitettyä kuvassa 14. Opinnäytetyössä pyritään pitäytymään ominaisuuksissa, jotka ovat toteutettavissa graafisen käyttöliittymän avulla. Tämä tarkoittaa sitä, että perinteiseen ohjelmointiin turvaututaan vasta siinä vaiheessa, kun muuta vaihtoehtoa ei ole. Sovellus sisältää ohjelmointia helpottavan *MonoBehaviour*-nimisen luokan, johon on rakennettu muutamia komentoja, joita voidaan käyttää toiminnallisuuksia luotaessa. Ennen Unityn käyttöönottoa olisi kuitenkin hyvä hallita esimerkiksi C#-ohjelmoinnin perusteet.

Kun äänimaailmaa aletaan luoda Unityllä, on kyseisestä sovelluksesta tiedettävä muutamia perusasioita. Äänituotannon kannalta oleellisinta on ymmärtää, että koko peliprojekti jaetaan kohtauksiin. Kohtauksessa olevat aistittavat asiat ovat nimeltään objekteja. Näihin objekteihin liitetään komponentteja, jotka määrittävät objektien ominaisuudet ja toiminnallisuudet. Tässä tapauksessa kohtausta saattaisi sijoittua esimerkiksi metsään. Tällöin kasvillisuus ja eläimet olisivat objekteja. Kasvillisuudella komponenttina voisi olla väri ja eläimillä animaatiot ja liikkuminen. Unityssa äänimaailmaan liittyvät pakolliset komponentit ovat äänilähde ja kuuntelija. Myös mikseri on hyvä ottaa käyttöön ääniraitojen prosessointia varten. Unityn virtuaalisiin prosessoreihin kuuluu muun muassa äänituotantoa käsittelevässä luvussa mainitut kaiku ja taajuuskorjain. (Unity Technologies 2019a.)



Kuva 14. Unityn peruskäyttöliittymä

4.1.1 Peruskomponentit

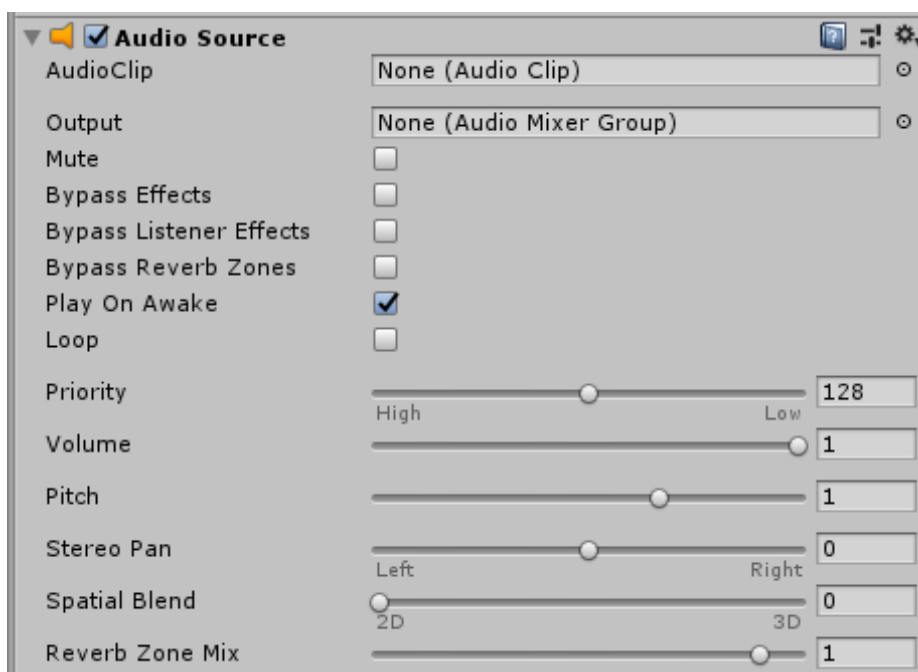
Kuuntelijakomponentti vastaa toimintaperiaatteiltaan korvia. Nimensäkin mukaisesti se kuuntelee sovellukseen liitettyjä ääniä. Yhdessä kohtauksessa voi olla vain yksi kuuntelija, jotta kohtaaminen toimisi oikein. Kuuntelija liitetään oletuksena pelaajakomponentissa olevaan kameraan, joka vastaa reaali maailman silmiä. Näin kuuntelija saadaan seuraamaan myös sovelluksen käyttäjän pään liikettä. Komponentissa ei ole säädettäviä ominaisuuksia, vaan kaikki säädöt, jotka muokkaavat kuuntelijan aistimia ääniä, tehdään muille komponenteille. (Unity Technologies 2019a.)

Mikäli kuuntelijaan kantautuvia ääniä halutaan muokata, on otettava käyttöön erilaisia signaaliprosessoreja. Nämä prosessorit ovat nimenomaan komponentteja, joten ne voidaan liittää suoraan objektiin. Esimerkiksi kaiun saa helpoiten aikaan siten, että lisää *Audio echo filter*-nimisen komponentin objektiin, joka toistaa äänen. Samoin voi tehdä kaikilla signaaliprosessoreilla, mutta mikäli kohtaaminen sisältää paljon objekteja, voi tämä toimintatapa käydä hitaaksi toteuttaa. Lisäksi se saattaa liiaksi kuormittaa tietokoneen suorituskykyä. Unityn äänityökalut ovat kuitenkin sen verran edistyneitä, että äänien toteutukseen on järkevämpiäkin tapoja. (Unity Technologies 2019b.)

Äänilähde on komponentti, joka nimensäkin mukaisesti toistaa sovellukseen sijoitetun äänen. Tämä komponentti lisätään pelin sisällä olevaan mallinnettuun tai tyhjiin objektiin, jonka halutaan pitävän ääntä. Komponentti sisältää muutamia valintoja, joita säätämällä äänestä saadaan halutunlainen. Ensinnäkin äänilähteelle valitaan ulostulo. Ääni voidaan

toistaa suoraan kuuntelijalle tai vaihtoehtoisesti ajaa mikserin kautta. Seuraavana valitaan, mitä äänestä toistetaan. Ääniraita voidaan mykistää kokonaan, siitä voidaan mykistää vain kyseessä olevaan objektiin lisätyt efektit, tai toistaa niiden kera. (Unity Technologies 2019c.) Komponentin efekteihin liittyvät säädöt eivät vaikuta mikserin puolella lisättyihin efekteihin.

Seuraavana tehdään muutamia valintoja liittyen äänilähteen toistotapaan. Äänilähteen voi määrittää toistumaan joka kerta, kun sovelluksen kohtaaminen käynnistetään. Tällöin oletuksena ääniraita soi vain kerran läpi. Toisaalta äänen voi asettaa toistumaan myös loputtomalla silmukalla, mikä on toimiva tapa usein esimerkiksi taustamusiikin toistamiseen. Äänelle määritettävällä tärkeysarvolla (*engl. Priority*) taas pyritään varmistamaan se, ettei ääniraita katkea missään vaiheessa. Esimerkiksi taustamusiikille on hyvä asettaa korkea tärkeysluokitus. Huomattavaa tärkeyden säätämisessä on se, että nimenomaan matala arvo tarkoittaa korkeaa tärkeysluokitusta, kuten kuvasta 15 on havaittavissa. Lopuksi säädetään vielä ääniraidan voimakkuus, sävelkorkeus, panorointi sekä etäisyyteen ja sijaintiin liittyvät tekijät. Näitä ovat mm. kappaleessa 4.2 esiteltävä Dopplerin ilmiötä simuloiva efekti, äänen leviäminen 3D-maailmassa sekä äänenvoimakkuuden muuttuminen kuuntelijan sijainnin mukaan. Jokaista ääntä ei välttämättä tarvitse istuttaa 3D-maailmaan, vaan ääniohjelmoijan on harkittava, olisiko järkevämpää toistaa ääni 2D-muodossa. Esimerkiksi ympäristöissä, joissa ääni tulee tasaisesti joka puolelta, tällainen valinta saattaisi toimia. Äänen ulottuvuus säädetään *spatial blend* säätimellä ja lopputulos voikin olla sekoitus molempia. (Unity Technologies 2019c.)



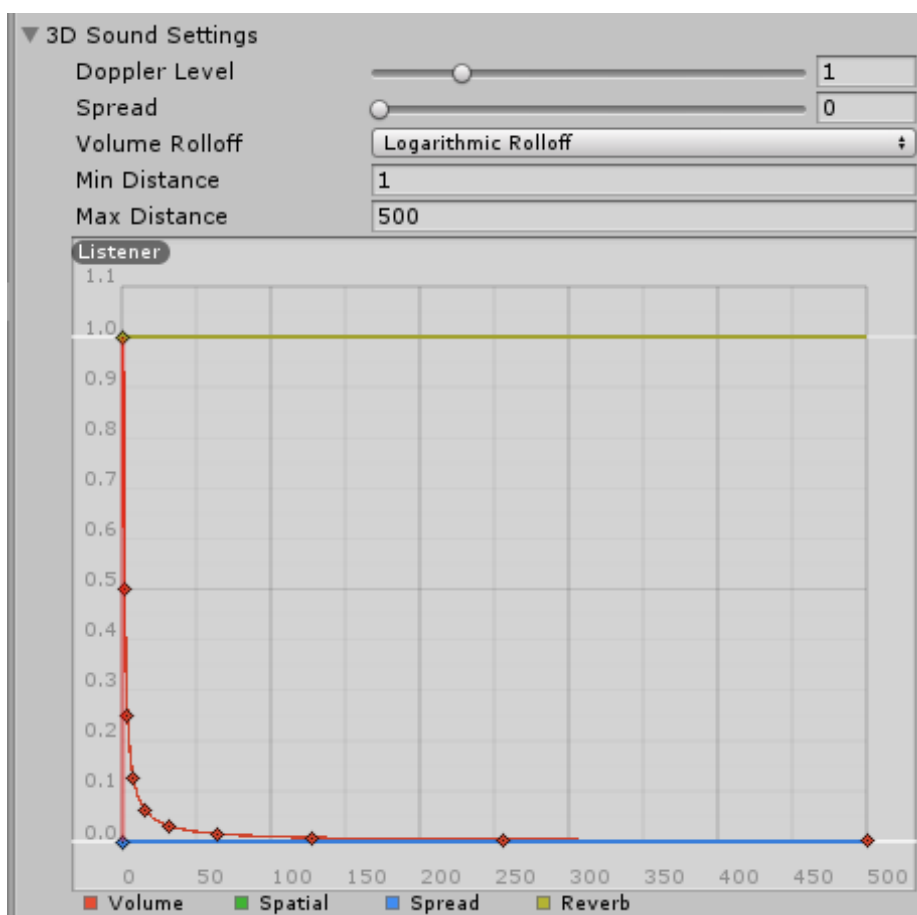
Kuva 15. Äänilähde ja siihen liittyvät säätimet

4.1.2 Unityn 3D-äänityökalu

Virtuaalitodellisuuskonseptin esittelyn jälkeen pelimoottoreille on luotu useita erilaisia lisäosia, joista on apua äänimaailman realistisuuden luomisessa. Lisäksi lisäosien avulla äänimaailman luominen on vaivattomampaa, eikä niitä käyttäessä välttämättä vaadi edes ohjelmointia. Lisäksi kaikille näiden lisäosien tarjoamille mahdollisuuksille ei Unityn omassa äänimoottorissa vastinetta edes ole. Ohjelmoijan on kuitenkin hyvä tietää, miten Unityn omalla äänimoottorilla voidaan luoda esimerkiksi käyttäjän sijainnin mukaan vaihtuva taustamusiikki tai 3D-ääni, joka seuraa virtuaalitodellisuussovelluksen käyttäjän toimintoja, kuten pään liikkeitä ja etäisyyttä äänilähteeseen.

Micheal Lanham (2018, 42 - 43) kirjoittaa kirjassaan *Game Audio Development With Unity 5.X* Unityn oman äänimoottorin toiminnasta. Hänen mukaansa ensimmäinen askel 3D-äänimaailman luomisessa on tutustua 2D- ja 3D-äänen eroihin ja niiden sekoittamiseen. 2D-äänellä tarkoitetaan kaksikanavaista, eli stereoääntä, joka toistetaan sellaisenaan sovelluksen käyttäjälle. 3D-ääni sen sijaan on yksikanavaista, ja se pyritäänkin istuttamaan tiettyyn sijaintiin 3D-ympäristössä. Tämä tarkoittaa käytännössä sitä, että jos ääni tulee kaukaa oikealta, on sen vaikutusta vasempaan korvaan vaimennettava sekä viivästettävä hieman. Tämä aiheuttaa sellaisen psykoakustisen ilmiön, että käyttäjä voi määritellä äänilähteen sijainnin hyvinkin tarkasti, vaikka hänellä olisi vain stereokuulokkeet korvilla. Tätä ilmiötä, jonka keinotekoisesti luotua versiota hyväksikäyttäen saadaan vaikutelma äänen tulosuunnasta, kutsutaan Haas-efektiksi.

Toisaalta taas sellaiset äänet, joiden täytyy kuulua joka puolelle virtuaaliympäristöä, mutta joilla on selkeä lähde, voidaan toteuttaa sekoittamalla 2D- ja 3D-ääntä äänilähdekomponentin *spatial blend* säätimellä. Unityssa äänitehosteen voimakkuustason muuttumista suhteessa kuuntelijan ja äänilähteen etäisyyteen säädetään äänilähdekomponentissa olevalla graafilla. Ensin komponenttiin määritetään suurin etäisyys, jossa ääni halutaan olevan vielä kuultavissa. Lisäksi äänen vaimenemista voidaan säätää graafin muokkauspiSTEillä, jotka Unity oletuksena asettaa logaritmiseen muotoon, kuten kuvasta 16 on havaittavissa. Graafin oikean jyrkkyyden ja muodon löytäminen kyseessä olevaa kohtausta varten lienee kuitenkin helpointa määrittää kokeilemalla. (Lanham 2018, 42 - 45.) Toisessa pääluvussa mainittiin korkeiden taajuuksien häviämisestä äänen tullessa kauempaa, ja ohjelmoijan onkin syytä miettiä, onko pelkkä äänenvoimakkuustason vaimeneminen riittävä efekti etäisyyden simulointiin.



Kuva 16. Äänilähde-komponentin 3D-osio

4.1.3 Mikseri

Kaikki sovelluksen sisältämät äänet voidaan toistaa myös mikserin kautta. Perinteisesti mikseriä käytetään etenkin kohtauksessa olevien ääniryhmien käsittelyyn. Ohjelmoija voi luoda omat ryhmät esimerkiksi äänitehosteille, taustamusiikille ja puheelle. Näihin ryhmiin hän reitittää niiden otsikoita vastaavat ääniraidat. Mikserin äänenvoimakkuuksien säätö on yksinkertaisesti visualisoitu, sillä tehosteiden voimakkuuden oikean tasapainon löytäminen on mikserin yksi merkittävimmistä käyttötarkoituksista. Käyttäjän luomien ryhmien lisäksi mikserissä on aina myös master-kanava, jonka kautta kaikki toistettava ääni kulkee. Näiden lisäksi siitä löytyy myös solo-, mykistys- ja ääniryhmäkohtaisien efektien ohituspainikkeet. (Unity Technologies 2019d.) Kuvassa 17 on esitettynä Unityn mikserin perusnäkyä oletusasetuksilla. Käyttäjä on luonut siihen kolme ääniryhmää, jotka ovat äänitehosteet (*sound FX*), taustääänet (*background*) ja dialogi (*dialogue*).

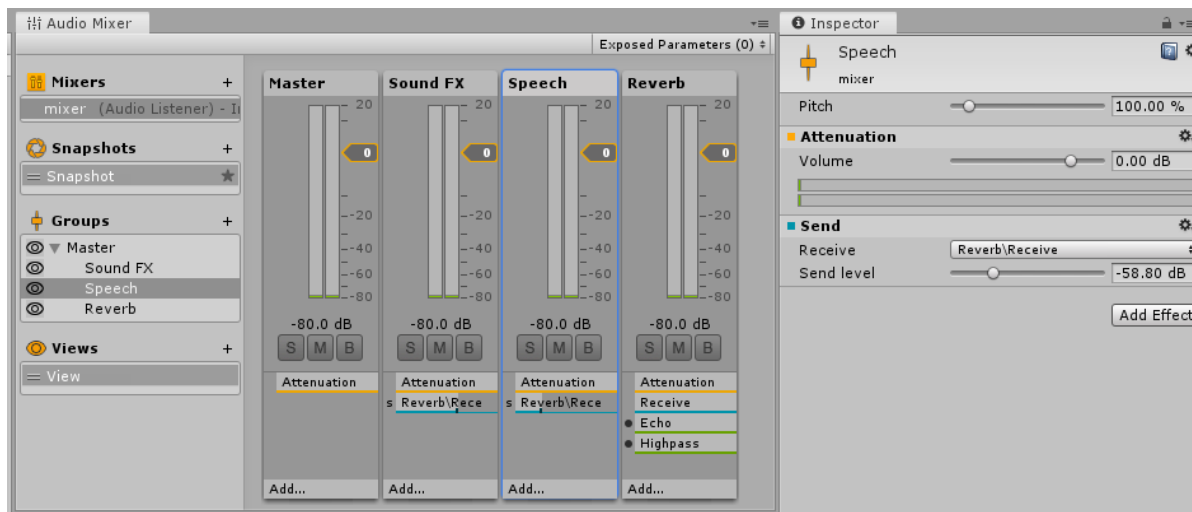


Kuva 17. Unityn mikseri

Toinen mikserin käyttötarkoituksista on signaaliprosessorien lisääminen kaikille ryhmän radoille kerrallaan. Tiettyyn ryhmään lisätty signaaliprosessori vaikuttaa kaikkiin tämän ryhmän sisältämiin ääniraitoihin. Myös master-kanavan signaali voidaan prosessoida, joten jos halutaan vaikuttaa koko kohtauksen kaikkiin ääniin, on virtuaalinen prosessori syytä lisätä master-kanavaan. Tietyissä tapauksissa tätä keinoa kutsutaan myös masteroinniksi.

Signaaliprosessorien lisäämiseen mikserin avulla on kaksi eri tapaa. Yksinkertaisempi tapa näistä on lisätä prosessori suoraan ääniryhmään, jolloin ryhmän sisältö ajetaan signaaliprosessorin kautta sellaisenaan. Tätä keinoa kannattanee käyttää taajuuskorjaimen ja kompressorien käyttöön, sillä näissä tapauksissa prosessoimatonta signaalia ei todennäköisesti haluta käyttää ollenkaan. Hieman edistyneempi ja monipuolisempi tapa efektiivien toistamiseen on luoda paluukanava niille. Tällöin mikseriin luodaan tyhjä kanava, jossa ei ole muuta kuin yksi tai useampi signaaliprosessori. Siihen syötetään haluttuja ryhmiä halutulla äänenvoimakkuudella, ja ne toistetaan kanavan kautta sen sisältämine efekteineen. Tämä tapa on käytännöllisempi, kun halutaan käyttää yhtä signaaliprosessoria useampaan ääniryhmään tai prosessoida signaali siten, ettei alkuperäistä, eli raakaa signaalia menetetä. (Unity Technologies 2019d.) Tätä tapaa taas on syytä käyttää erilaisien kaikujen toistamiseen. Kuvassa 18 vasemmanpuoleisen master-kanavan lisäksi on luotu ryhmät äänitehosteille (*Sound FX*) ja puheelle (*Speech*). Kaiku ja ylipäästösuodin on lisätty omaan paluukanavaan (*Reverb*), johon esimerkiksi puhetta syötetään -58.8 desibelin verran. Tämä tarkoittaa sitä, että ääniryhmän signaalista vähennetään 58.8 desibeliä, ja

jäljelle jäävä signaali syötetään paluukanavaan, jossa siihen lisätään kaiku sekä siitä leikataan ylimpiä taajuuksia pois. Alkuperäisen puhekanavan äänenvoimakkuustaso voidaan yhä säätää vaikuttamatta prosessoidun kanavan voimakkuuteen. Tällaisilla asetuksilla kaiun vaikutus puheet sisältävään ääniryhmään olisi melko mieta.



Kuva 18. Efektien lisääminen paluukanavia käyttäen

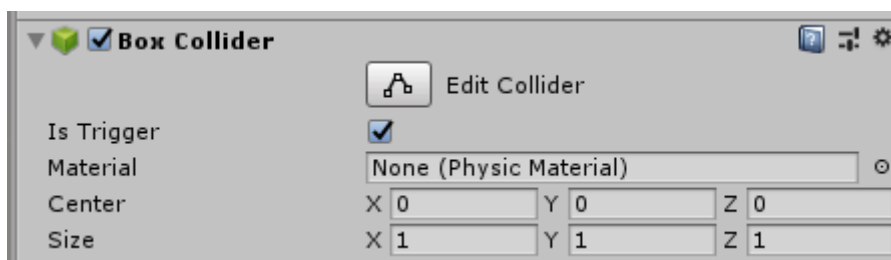
Unityn mikseriin voidaan tallentaa myös mikserin eri tiloja. Sovellus käyttää näistä tiloista englannin kielistä nimitystä ”*Snapshot*”, ja niitä käytetään efektien hallitsemiseen ohjelman ajon aikana. Yksi tila voisi olla esimerkiksi kaikuisaan tunneliin tarkoitettu ja toinen avaralla pellolla, jossa kaikua ei juurikaan ole. (Lanham 2017, 127.) Aiheeseen palataan myöhemmin, kun tutustutaan äänen laukaisevaan komponenttiin sekä tilan tunnun ohjelmointiin.

4.2 Äänen laukaisijat ja dynaaminen ääniraita

Mikäli kyse ei ole kohtausten käynnistyessä laukeavasta ääniraidasta, on ohjelmoijan seuraavaksi valittava keino, joka laukaisee ääniraidan. Tapahtuma toteutetaan yleensä jonkinlaisella ehtolauseella. Kiteytettynä ehtolauseella tarkoitetaan sitä, että pelimoottori odottaa tietyn ehdon täyttymistä ja havaitessaan tapahtuman, suorittaa sille määritetyn toiminnon. Tässä tapauksessa ehto voisi olla esimerkiksi kappaleen törmäminen seinään, mistä aiheutuva toiminto olisi siitä kuuluva ääni. Näin laukeava ääni on Unitylla työskenneltäessä järkevintä toteuttaa törmäystarkastelujen avulla. Komponentin nimi Unityssä on *collider*, ja sen ensisijainen tehtävä onkin simuloida fysiikkaa törmäyksien osalta. Komponentin voi kuitenkin määrittää myös laukaisijaksi, eli triggeriksi, kuvan 19 osoittamalla tavalla. Tällöin sen tehtävä ei ole enää toteuttaa törmäystä vaan aistia objektin tunkeutuminen toiseen objektiin sekä poistuminen siihen liitetyn triggerin vaikutusalueelta. (Unity technologies 2019e.) Näiden viestien avulla voidaan laukaista ääni juuri sillä hetkellä, kun

kappale törmää seinään tai esimerkiksi mikserin tilojen avulla mykistää kaiku, kun auto poistuu tunnelista. Alla esimerkki tämänkaltaisesta ratkaisusta.

```
private void OnTriggerEnter(Collider other)
{
    AudioSource.Play();
}
```



Kuva 19. Laatikon törmäystarkastelu asetettuna triggeriksi

Toinen merkittävä äänen laukaisutapa liittyy käyttäjän syötteeseen. Kyse on edelleen ehtolauseesta, mutta tähän toimintoon ei välttämättä tarvitse liittää mitään visuaalista interaktiota. Ääni voidaan asettaa laukeamaan myös painikkeen painalluksesta alla esitetyn skriptin avulla. Tämä tapa on oivallinen, kun ohjelmoidaan esimerkiksi aseesta lähtevää ääntä. Yksinkertaisuudessaan ampumiseen varatun painikkeen toimintoihin lisätään myös kertaalleen laukeava ase-ääni.

```
if (Input.GetButtonDown("Fire1"))
{
    gunShotAudio.Play();
}
```

Kuten jo aiemmin mainittiin, on Unityn äänimoottori hieman rajoittunut, varsinkin kun siirrytään virtuaalitodellisuuskehitykseen. Tämä ei kuitenkaan tarkoita sitä, että sillä olisi mahdollonta tehdä mukautuvia äänimaailmoja. Esimerkiksi mukautuvan taustamusiikin toteuttamiseen Unityn tarjoamat työkalut ovat täysin riittäviä. (Lanham 2017, 161.) Mukautuvalla taustamusiikilla tarkoitetaan sitä, että musiikki vaihtuu 3D-ympäristössä vallitsevien tilanteiden mukaan. Yksi yleisimmistä tekijöistä tässä tapauksessa on käyttäjän sijainti. Ympäristö voi olla esimerkiksi rauhallinen, hektinen, uhkaava tai neutraali ja musiikin vaihtuminen niiden mukaan on yleisesti käytetty keino korostaa näitä tunnelmia. Eri tunnelman

omaavat ympäristöt voidaan erotella triggereillä, jonka jälkeen niille määritetään eri taustamusiikit. Kun triggeri havaitsee pelaajan liikkuvan sen vaikutusalueen sisään, vaihtuu myös taustamusiikki sen mukaiseksi. (Lanham 2017, 167 - 168.)

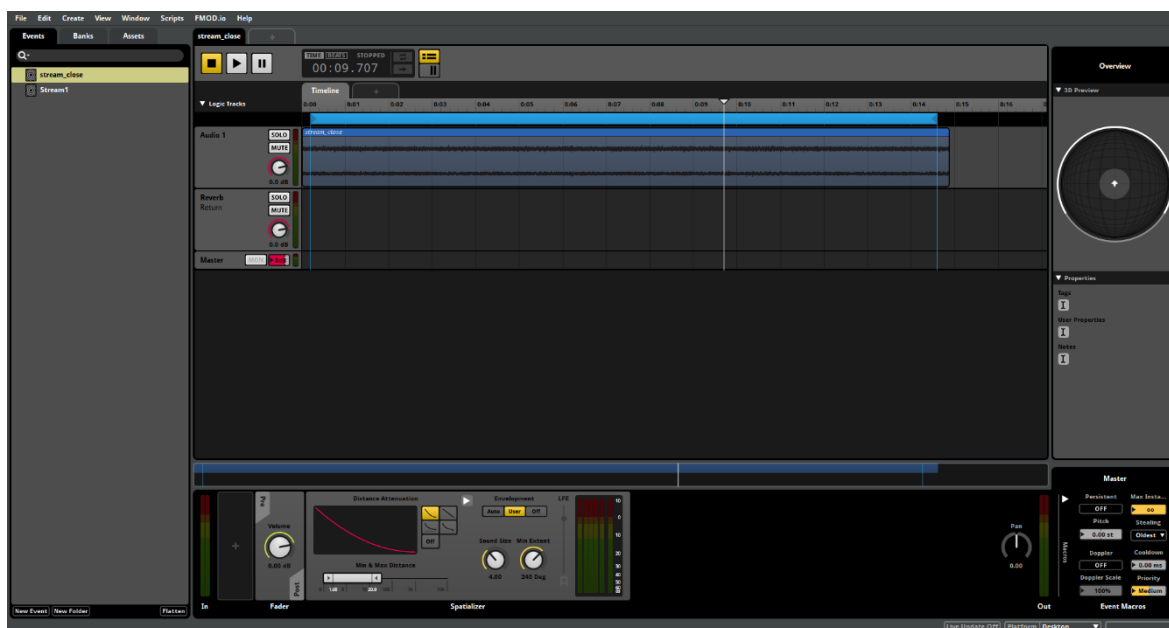
Liikkuva ääni on myös asia, johon ohjelmoijan täytyy kiinnittää erityistä huomiota. 3D-ääninä ohjelmoitaessa äänet voidaan istuttaa 3D-ympäristöön, minkä jälkeen ne saattavat kuulostaa realistiselta jopa sellaisenaan. Liikkuvaan ääneen liittyy kuitenkin enemmän ongelmakohtia. Aiemmin mainittu Doppler-ilmiö, on syytä toteuttaa myös virtuaaliodellisuusympäristöön, mikäli sovelluksesta halutaan mahdollisimman realistinen. Kiteytettynä se on ilmiö, joka havaitaan äänenkorkeuden muuttumisena lähteen ollessa liikkuva. Kun äänilähde liikkuu, on sen lähettämät ääniaallot sen etupuolella tiheämpiä ja takapuolella harvempia. Tämä johtuu siitä, että ääniaallon alkupiste ei pysy samassa paikassa suhteutettuna edelliseen ääniaaltoon. Yleinen käytännön esimerkki ilmiöstä on ohi ajava ambulanssi. Kun ambulanssi lähestyy kuulijaa, on äänenkorkeus suurempi ja ohitushetkellä se laskee huomattavasti. Unityn omassa äänimoottorissa on Dopplerin ilmiötä simuloiva työkalu ja sen toimintaa kokeillaan case-osuudessa. (Korpinen 2006.)

4.3 FMOD

Jos Unityn tarjoamat työkalut äänimaailman toteuttamiseen eivät riitä, kannattanee ääniohjelmoijan tutustua FMOD-nimiseen liitännäiseen. Kyseisen liitännäisen voi ladata verkosta ilmaiseksi ja se toimii saumattomasti Unityn kanssa. Ohjelma ei ole kovin yksinkertainen käyttää, mutta sen haltuun ottamisen seurauksena työskentely nopeutuu sekä tuotoksista saadaan realistisempia, kuin Unityn omalla äänimoottorilla tehdessä. FMOD:ia on käytetty monen kaupallisen ja maailmanlaajuisesti suosittu pelin äänimaailman luomiseen. (Lanham 2017, 193.)

Käyttöliittymältään FMOD näyttää hyvin samankaltaiselta kuin useat digitaaliset äänityöskentelyasemat, kuten kuvasta 20 on havaittavissa. Siinä on esitettynä eri raidat päällekkäin ja jokaisella näistä raidoista on aikajana, jossa näkyy kyseisen raidan ääniaalto. Sovelluksen perusnäkyvässä ruudun vasemmassa reunassa on havaittavissa ääniraidat, joista sovellus käyttää nimeä *asset*. Näistä raidoista luodaan tapahtumia (*engl. Event*), joita voidaan toistaa Unityn puolella liitännäisen avulla. Unityn puolella tarvittavia toimenpiteitä on kuuntelijakomponentin vaihtaminen FMOD:n kuuntelijaksi sekä äänilähteiden vaihtaminen FMOD:n omiksi äänilähteiksi. Lisäksi raidoille voi asettaa signaaliprosessoreja, kuten kaikuja ja suodattimia. Selkeimpänä eroavaisuutena normaaliin äänityöskentelyasemaan FMOD:lla on 3D-tapahtuman luonnin jälkeen sovelluksen perusnäkyvän oike-

aan reunaan ilmestystä panorointityökalu, jolla voidaan testata myöhemmin 3D-ympäristöön asetettavaa ääntä eri suunnista. FMOD:n käyttöliittymään ja mahdollisuuksiin tutustutaan tarkemmin käytännönvaiheessa.



Kuva 20. FMOD:n peruskäyttöliittymä

4.3.1 Signaaliprosessorit

Äänituotantoa käsitelleessä luvussa esitellyt parametrit ja säätimet löytyvät miltei samantapaisina FMOD:sta, ja niitä säätäessä voidaan käyttää apuna täysin samoja teorioita. Itse efektin lisääminen tapahtuu FMOD:ssa suoraan raitaan lisäämällä tai paluukanavan avulla. Kyseessä on siis täysin samankaltaiset toimenpiteet, kuin Unityn mikserillä, vaikka käyttöliittymä onkin erinäköinen. Kaikkia FMOD:n sisältämiä signaaliprosessointimenetelmiä ei tämän opinnäytetyön puitteissa pystytä käymään läpi niiden monipuolisten käyttötapojen vuoksi. Perusidea signaaliprosessorien käytössä on kuitenkin sama, kuin missä tahansa äänityöskentelysovelluksessa, ja ohjelmoija voi käyttää niitä parhaaksi kokemallaan tavalla. (Lanham 2017, 212 - 214.)

Myös FMOD:n puolella voidaan luoda Unityn mikseristä tuttuja tiloja. FMOD käyttää tiloista samaa englannin kielistä nimitystä *snapshot* ja näitä tiloja käytetään täysin samaan tarkoitukseen, kuin Unityn äänimoottorilla työskenneltäessä. FMOD:n kanssa tilojen käyttöönotto ei vaadi lainkaan ohjelmointia. FMOD:iin voidaan asettaa esimerkiksi kahden objektin etäisyyden mukaan muuttuva parametri, joka säätelee tietyn ääniraidan voimakkuustasoa tai alipäästösuotimen raja-arvoa. Kuuntelijan ja äänilähteen etäisyyden toisistaan 3D-ympäristössä FMOD havaitsee automaattisesti. Tämän ansiosta tilojen välinen

sekoittuminen on FMOD:n puolella yksinkertaisempaa toteuttaa. (Lanham 2017, 218 - 220.)

4.3.2 Äänen satunnaistaminen

Kappaleessa 2.4 tutustuttiin hieman äänen monipuolistamiseen sovelluksen ajon aikana. Mikäli tietty ääniefekti toistetaan useita kertoja peräkkäin täysin samanlaisena, se saattaa myös tuntua toistuvulta, mikä todennäköisesti rikkoo immersion. Toki tiettyyn tapahtumaan voidaan liittää useita eri ääniraitoja, joista ohjelma toistaa satunnaisesti joka kerralla jonkun. Tämä toimintatapa saattaa kuitenkin kasvattaa valmiin sovelluksen kokoa liiaksi. Ohjelmoijan on siis keksittävä ratkaisu, jolla saavutetaan äänen vaihtuvuus, vaikka itse ääniraita olisi joka toistokerralla sama.

Unityn omalla äänimootorilla äänitehosteiden muuttumista sovelluksen ajon aikana voidaan toteuttaa vain ohjelmoimalla. Sen sijaan FMOD:lla tämänkin toiminnallisuuden luominen on mahdollista ilman ohjelmointia. Ääniraidoille voidaan lisätä modulaatioita, sävelkorkeuden säätimiä ja muita signaaliprosessoreja, joiden säätimet voidaan asettaa satunnaisesti liikkuviksi. Näillä keinoilla äänitehosteista saadaan monipuolisia ja vaihtelevia. Ohjelmointiin tarvitaan kuitenkin tilanteessa, jossa halutaan jonkin automatisoidun parametrin vaikuttavan toiseen parametriin. Esimerkki tällaisesta tapauksesta on automatisoidun tuulen nopeuden vaikutus tuulen äänitehosteeseen ja sen voimakkuuden vaihteluun. (Lanham 2017, 233 - 237.)

4.4 Steam Audio

Tähän asti esiteltyjä mukautuvan ääniraidan toteuttamiseen käytettäviä työkaluja voidaan kutsua pääosin manuaalisiksi. Ohjelmoija pystyy näillä työkaluilla itse määrittämään eri tilojen akustiset ominaisuudet ja esimerkiksi FMOD:lla säätämään ääniraidan taajuusvas-teen sekä muun prosessoinnin sen kaltaiseksi. Monelle kokeneemmalle ohjelmoijalle tämä on mieluisin tapa toimia, mutta esimerkiksi itsenäisille, tai aloitteleville ohjelmoijille on myös vaihtoehto. Peliyhtiö Steam on vuonna 2017 julkaissut äänilisäosan, jolla mukautuvaa ääntä voidaan simuloida automaattisesti 3D-ympäristön geometrian avulla. Sovelluksen Unity-liitännäisen voi ladata ilmaiseksi, ja sitä voi käyttää haluamansa äänimootorin kanssa, ja jopa kaupallisten sovelluksien tuottamiseen. Seuraavissa kappaleissa esitellään sovelluksen tarjoamia mahdollisuuksia, joita ei välttämättä Unityn omalla äänimootorilla tai FMOD:lla pysty yhtä realistisesti toteuttamaan.

4.4.1 Materiaalit, heijastukset ja tukkeuma

Unitylla 3D-ympäristöä luotaessa kohtaukseen määritetään objekteille materiaalit, törmäystarkastelut ja pinnan ominaisuudet. Nämä eivät kuitenkaan vaikuta äänen käyttäytymiseen, vaan akustiset ympäristöt on määritettävä ikään kuin manuaalisesti ohjelmoinnin ja triggereiden avulla kappaleessa 4.2 esitellyllä tavalla. Steam Audio tarjoaa tähän ongelmaan ratkaisun, joka perustuu nimenomaan objektien materiaaleihin. Materiaalien vaihtuvuutta ja pintojen läheisyyttä Steam Audio simuloi niin paikallaan olevien, kuin liikkuvienkin komponenttien yhteydessä. (Valve Corporation 2018.) Ominaisuudesta löytyy hyvin vähän tietoa Steamin verkkosivulta, mutta tätä työkalua on helppo testata, ja sitä tullaankin käyttämään opinnäytetyön case-osiossa projektin edistyessä.

Toinen Steam Audion tärkeimmistä hyödyistä on äänien tulosuuntaan liittyvät ominaisuudet. Unityn omalla äänimoottorilla tai FMOD:illa ei lähtökohtaisesti pystytä tarkastelemaan onko äänilähteen ja pelaajan välissä esimerkiksi ääntä vaimentava este. Toki tällaisen ominaisuuden voi toteuttaa skriptin tai triggerin avulla, mutta Steam Audio tarjoaa ongelmaan helpomman ratkaisun. Liitännäisessä on työkalu, joka tarkastelee kaiken aikaa pelaajan suhdetta äänilähteeseen, sekä havaitsee välissä olevat esteet. Se lähettää eräänlaisen säteen pelaajaksi määritettyyn objektiin. Jos säde ei pääse perille, tekee komponentti toimenpiteitä säteen estävän objektiin määritetyn materiaalin mukaisesti. Se saattaa esimerkiksi vaimentaa ääntä, tai lisätä siihen alipäästösuotimen simuloidakseen tukkeumaksi kutsuttua efektiä. Lisäksi Steam Audio toteuttaa pään liikkeisiin ja kuuntelijakomponentin sijaintiin liittyvää simulaatiota, joten jos käyttäjä menee esimerkiksi kulman taakse, voidaan ääni havaita heijastuvan kulman takaa. Tämänkin ominaisuuden voi ottaa käyttöön graafisen käyttöliittymän puolella, ja sen toiminta mukautuu automaattisesti 3D-ympäristön mukaisesti. (Valve Corporation 2018.)

4.4.2 Tekniset ominaisuudet

Teknisiltä ominaisuuksiltaan Steam Audio lupaa myös paljon. Sen verkkosivuilla taataan liitännäisen varaavan suhteellisen vähän tietokoneen suoritustehoa, jotta yleensä muuten tehoja vaativa virtuaalitodellisuusprojekti ei ylikuormittaisi tietokoneen suorituskykyä. Verkkosivun mukaan liitännäinen on sopiva jopa mobiilikäyttöön. Projektin vaatimaa suoritustehoa pystytään vielä optimoimaan toimenpiteellä, jota Steam Audio kutsuu englannin kielisellä nimityksellä *build*, eli ikään kuin kaikkien äänitehosteiden kypsytämiseksi. Termi saattaa kuulostaa omituiselta, mutta koska sille ei tässä yhteydessä ole vakiintunut suomenkielistä käännöstä, käytetään yhtä sen alkuperäisistä käännöksistä. Unityn oma äänimoottori ei itse asiassa edes tarjoa äänille tätä vaihtoehtoa, vaan idea tulee ennemminkin

virtuaalisien valojen ja materiaalin yhteydestä. Kypsyttämällä tarkoitetaan sitä, että toimenpiteen jälkeen äänet eivät käyttäydykään dynaamisesti tai reaaliaikaisesti, vaan ne ikään kuin istutetaan paikalleen. Tämä vähentää vaadittavaa laskentatehoa sovelluksen ajon aikana, ja seurauksena saadaan sovelluksen sulavampi toimiminen. Toimenpidettä ei tietenkään voida tehdä ääntä lähettävien liikkuvien objektien kohdalla, mutta mikäli tila on staattinen, on sitä syytä harkita. (Unity Technologies 2019f.)

4.5 Projektin toteutus

Seuraavaksi päästään työskentelemään itse projektin pariin. Koska moni asia toteutuksessa riippuu ohjelmoijan mielikuvituksesta ja mieltymyksistä, tullaan case-osuudessa todennäköisesti kokeilemaan erilaisia vaihtoehtoja eikä niinkään nojata pelkästään teoriaan. Toki tietyt lainalaisuudet pätevät tiettyihin toimintatapoihin, mutta tämänkaltaisessa projektissa on hyvä olla utelias ja kokeilla mitä erilaisimpiakin toimintatapoja.

5 CASE

5.1 Suunnittelu

Koska käytännön osuudesta valmistuvan sovelluksen käyttöönotosta ei ole varmuutta, ei sille myöskään saatu rahoitusta. Tästä syystä projektiin ei voida käyttää niin paljon tunteja, kuin ehkä olisi syytä. Projektista tulee pienimuotoinen, mutta äänimaailmaltaan vaikuttava varsinkin silloin, kun sitä testataan asiaan kuuluvalla kalustolla, eli virtuaalimallitodellisuuskalusteilla ja kuulokkeilla. Kohtauksen ympäristö perustuu Mäntyharjussa sijaitsevaan Pyhäkosken laavun ympäristöön. Kyseisessä paikassa on pienehkö koski, jonka ylitse kulkee melko vilkas maantie. Laavu on muutaman metrin päässä koskesta ja maantielle matkaa siitä on noin 15-20 metriä. Maantielle mallinnetaan ja animoidaan ohitse ajavia autoja, joiden moottorin ääneen testataan Unityn omaa Doppler-ilmiötä simuloivaa työkalua. Valtaosa äänistä tullaan asettamaan 3D-ääniksi.

Projektin punaisena lankana on selvittää, kuinka lähelle realistista äänimaailmaa päästään Unityn omilla työkaluilla. Koska tässä vaiheessa tiedetään jo, että täysin realistista lopputulosta ei näin tulla saavuttamaan, tehdään sovellukseen kohtaus myös FMOD:ia hyödyntäen. Lopuksi valitun äänimoottorin sisältämään projektiin lisätään Steam Audio-liitännäinen. Projektin valmistumisen jälkeen tiedetään, minkälaisia ääniä ja mitä ilmiöitä kannattaa toteuttaa milläkin lisäosalla, sekä onko jokin näistä lisäosista jopa turha. Näiden tietojen pohjalta tehdään valmis kohtaus. Sovelluksen koko kasvanee varsinkin virtuaalimallitodellisuuteen liittyvien lisäosien tuomisen jälkeen, minkä takia turhista lisäosista on syytä hankkiutua eroon.

5.2 Kohtauksen luonti

Itse kohtauksesta päätettiin tehdä visuaalisesti melko pelkistetty. Kohtaus sisältää maaston, metsän muodostavat puut, maantien siltoineen ja kaiteineen sekä laavun. Tielle animoitiin ohitse ajava auto, joka on melko alkeellisesti mallinnettu. Maaston luomiseen ja sen muotojen muokkaamiseen Unity tarjoaa valmiin, melko yksinkertaisen työkalun. Kyseisellä työkalulla voidaan luoda halutun kokoinen taso, jonka profiilia pystytään muokkaamaan sitä korottamalla, laskevalla tai tiettyyn korkeuteen asettavalla siveltimellä. Tässä tapauksessa maaston lähtökorkeudeksi asetettiin 5 metriä. Maastonmuotoihin sovellettiin erimuotoisia siveltimiä sekä sattumanvaraisia korkeuksia. Ympäristön reuna oli tässä vaiheessa vielä silmin havaittava, joten yhteen reunoista mallinnettiin kallio. Kallioon mallinnettiin tunneli, johon maantie johti. Muut reunat piilotettiin näkyvistä puuston sekä maaston muotojen muokkaamisen avulla. Maastoon ”kaivettiin” ura, johon asetettiin virtaava

koski. Veden mallintaminen olisi vienyt liikaa aikaa, joten se päätettiin ladata valmiina Unityn omasta kaupasta. Myös puustoon päätettiin käyttää Unityn valmiita malleja, sillä puiden mallintaminen olisi vienyt liian paljon aikaa. Toisaalta yhden puun mallintaminen ja kopioiminen olisi toistuvuuden takia näyttänyt liian alkeelliselta. Puuston luontiin on maastotyökaluun sisällytettynä myös sivellin, jolla puut ikään kuin piirretään haluttuun sijaintiin. Työkalun valinnoissa voidaan vaikuttaa puuston tiheyteen, puiden korkeuteen sekä muihin puihin liittyviin perusasioihin.

Materiaalit ladattiin rojaltivapailta verkkosivuilta. Haluttuun lopputulokseen ei aivan päästy, sillä esimerkiksi metsän maastomateriaali ei täysin näytä siltä, mitä se Suomessa todennäköisesti olisi. Laavuun ja autoihin asetettiin yksinkertainen väri, jota muokattiin erilaisilla heijastuksiin liittyvillä säädöillä. Unityn tarjoama puumateriaali rungon sekä lehtien suhteen sen sijaan vaikuttivat hyvinkin realistiselta. Asfaltin ja sillan kaiteen materiaalit kelpasivat tähän tarkoitukseen, vaikka nekin todennäköisesti menevät vaihtoon, jos visuaaliseen puoleen aletaan myöhemmässä vaiheessa panostamaan.

5.3 Ensimmäinen äänityssessio

Kun visuaalista puolta koskevat perusasiat oli saatu kuntoon, ryhdyttiin suunnittelemaan äänimaailmaa. Tiedettiin, että projekti vaatisi vähintään muutaman erilaisen kosken kohinan, metsän ympäröivän äänen ja auton moottorin hurinan. Jo ennen äänityssessio aloittamista tiedettiin, että signaaliprosessointi tullaan pääosin tekemään Unityn puolella, joten kaikki äänitehosteet äänitettiin mahdollisimman raakana.

Ensimmäisenä taltioitavaksi ääneksi valikoitui ikään kuin kohtauksen keskipiste, eli kosken kohina. Koska mitään kalustoa ei kyseisessä sijainnissa voitu kytkeä verkkovirtaan, oli keksittävä jokin langaton äänitysmenetelmä. Kosteaa sijainnin takia kannettavaa tietokoneetakaan ei pidetty vaihtoehtona. Lahden ammattikorkeakoulun kalustosta löytyi kuitenkin Zoom H6-merkkinen kannettava tallennin, jolla työ saataisiin tehtyä. Kyseisessä tallentimessa on kuusi mikrofonia, joista kahdessa on valmiina stereoperiaatteella toimivat pienikalvoiset kondensaattorimikrofonit. Neljään muuhun kanavaan käyttäjä voi kytkeä minkä tahansa mikrofoniin. Laite mahdollistaa myös phantom-virtasyötön, joten ulkoiset kondensaattorimikrofonitkaan eivät ole poissuljettuja vaihtoehtoja. Kosken äänittämiseen riitti kuitenkin laitteessa jo oleva stereomikrofonipari. Koska käytössä oli lainakalustoa, jouduttiin äänityssessio pitämään heti kun laitteen lainaaminen oli mahdollista. Tämän takia näitä ääniä ei ehditty Mäntyharjuun asti äänittämään, vaan äänittäminen tapahtui Orimattilassa sijaitsevan kosken läheisyydessä. Kyseinen koski on virtaukseltaan huomattavasti voimakkaampi, mutta tämän ei annettu haitata projektin edistymistä, sillä ympäristön

ei ollut alun perinkään tarkoitus olla tarkka mallinnus mistään oikeasta sijainnista. Kuvassa 21 on meneillään kosken äänitystilanne.



Kuva 21. Zoom H6 kosken äänityksessä

Ensimmäiseksi äänitettäviksi kohteiksi valikoitui erilaisia kosken voimakkaita kohinoita. Parhaat tällaiset kohinat löytyivät kohdista, joissa koski virtasi kivikon tai muun esteen lävitse. Myös koskesta haarautuvia puroja äänitettiin muutama ääniraita. Näin kosken äänikokonaisuuteen saatiin rikkautta ja tietynlaista dynamiikka. Suunnitelman mukaan käyttäjän liikkuesssa kosken suuntaisesti, havaitsee hän erilaisia ääniä eri kohdissa riippuen siitä, minkälainen maastonmuoto kyseisessä sijainnissa on. Kohinoita äänitettiin eri etäisyyksiltä, sillä niitä suunniteltiin käytettävän vaihtuvan taustamusiikin lailla ympäristön tunnelmaa luomaan. Suunniteltiin, että kohinaan yhdisteltäisiin läheltä sekä noin 20 metriä

etäämmältä äänitettyä kohinaa. Sovelluksen käyttäjän liikkuessa kauemmaksi, korostetaan kauempaa äänitettyä raitaa, ja toisinpäin.

5.3.1 Toinen äänityssessio

Toiseen sessioon kuului askelien äänittäminen eri maastoissa. Äänityskalusto pystytettiin omakotitalon pihapiiriin, jossa maastovaihtoehtoina olivat sepeli, kuorikate sekä nurmikko. Kuorikatetta mallinnetusta 3D-ympäristöstä ei löydy, vaan sen päällä äänitettyä askeltamista suunniteltiin käytettäväksi silloin, kun käyttäjä kävelee metsässä esimerkiksi sammalen päällä. Sepelillä tehtyä äänitettä tultaisiin soveltamaan rantakivikkoon. Metsään sijoittuvaa askeltamista ei äänitetty metsässä, sillä huomattiin, että askeltamisen voimakkuutta oli hyvä korostaa hieman, jotta se saatiin vaikuttavammaksi. Tämä tuntui helpommalta toteuttaa kuorikatteen päällä, sillä olosuhteet olivat niin sanotusti hallitummat.

Saman session aikana äänitettiin myös ohi ajavien autojen moottorin ääni. Äänitys tehtiin paikallaan olevaa autoa äänilähteenä käyttäen. Arvioitiin, että kyseisellä maantiesuudella autojen nopeudet olisivat n. 80-100 km/h. Äänitettävässä autossa oli melko pieni moottori, joten kierrokset päätettiin nostaa noin 4000 kierrokseen minuutissa. Tämän lisäksi äänitettiin suuremmalla moottorilla varustettua autoa, jolloin kierrokset nostettiin hieman maltillisempaan lukemaan. Tällä menetelmällä äänitettäessä renkaiden rullausääntä eikä vauhdista johtuvaa tuulen huminaa kuitenkaan saada taltioitua. Kyseisten äänitehosteiden tallentamiseen ei keksitty järkevää ja toteutettavissa olevaa menetelmää. Tämän takia näihin tehosteisiin päätettiin kokeilla synteettisiä äänenluontimenetelmiä.

5.3.2 Synteettiset äänet

Auton aiheuttaman tuulenvireen äänitehostetta varten otettiin käyttöön FL Studio-niminen musiikin tuottamiseen tarkoitettu sovellus. Sovellus sisältää erilaisia äänigeneraattoreita ja virtuaalisia instrumentteja, kuten esimerkiksi digitaalisia syntetisaattoreita ja pianoja, rum-pukoneita sekä lisäksi lukuisia yksittäisiä ääniä. Käyttöön valittiin kuvassa 22 oleva 3xOsc-niminen lisäosa, joka jo nimensäkin mukaisesti perustuu erilaisien äänen luomiseen kolmen eri oskillaattorin avulla. Lisäksi työkalussa on mahdollisuus muokata sen luomaa signaalia esimerkiksi leikkaamalla tiettyjä taajuuksia pois tai korostamalla niitä, muuttamalla signaalin sävelkorkeutta tai lisäämällä edellä mainittuihin parametreihin automaatioita.

Apuna tuulta kuvaavan äänitehosteen luomisessa käytettiin internetin videopalvelu Youtubesta löytynyttä opetusvideota ”*Creating Wind (White Noise/On Any Synth!)*”. Parhaat lähökohdat saatiin asettamalla kaikki kolme oskillaattoria saha-aaltomuotoiseksi. Näin ollen

signaalissa ei enää ollut havaittavissa eri säveliä, vaan se sisälsi pelkkää kohinaa. Tämän jälkeen signaaliin lisättiin leikkuri, jolla vähennettiin kohinan ylätaajuuksia. Tässä vaiheessa ääni muistutti kosken kohinaa, joten myös nämä säädöt päätettiin tallentaa äänen myöhempää käyttöä varten. Resonanssi-säätimen nostamisella saatiin aikaiseksi tuulen viheltävyys. Lopuksi signaaliin lisättiin taajuuskorjain sekä laajan stereokuvan omaava viive-efekti. Etenkin viiveen lisääminen loi tehosteeseen etäisyyden tuntua lisäten samalla sen realistisuutta. (DarkEco 2016.) Näitä asetuksia muokkaamalla saatiin aikaisiksi myös uskottava auton renkaiden pitämä ääni.



Kuva 22. 3xOsc-liitännäinen

5.4 Äänimaailman peruskomponentit

Kohtauksen äänimaailman työstäminen aloitettiin istuttamalla voimakas kohina kosken kuohuvimpaan kohtaan. Erilaisia kohinoita asetettiin myös pitkin matkaa, jotta ääni kantautuisi joka puolelle tasaisesti. Lisäksi näin saatiin kosken kokonaisäänimaailmasta rikkaampi. Koskesta haarautuvaan puroon asetettiin ääni, joka on selkeästi lähempänä solinaa, kuin kohinaa. Samoin ohi ajavaan autoon asetettiin moottorin hurina, renkaiden ääni sekä auton aiheuttama tuulenvire. Tuulenvireen äänitehosteen toiminta toteutettiin triggerillä, joka havaitsee sovelluksen käyttäjän hänen ollessa tarpeeksi lähellä autoa. Pelaajan liikkeitä tarkastelevan skriptin avulla saatiin käyttöön myös askeltamisen ääni. Skriptin herkkyyttä pienennettiin hieman, jottei askellus kuuluisi kaikista pienimmistä liikkeistä, eli

sellaisista, joissa tosimaailmassa askelta ei välttämättä edes otettaisi. Samaan skriptiin ohjelmoitiin vielä toiminnallisuus askellusäänitehosteen vaihtumiselle maaston mukaan. Näiden lisäksi 2D-ääneksi asetettiin vielä etäältä äänitetty kosken ääni, johon äänitystilanteessa saatiin tallennettua linnun laulua sekä ohitse lentävän hyönteisen ääni. Tämä äänitehoste on hyvin huomaamaton, mutta sillä saatiin kohtaukseen huomattavasti lisää realismia tuntua.

Oletuksena Unity asettaa äänet aina 2D-muotoon, joten tämä asetusta oli vaihdettava joka kerta, kun uusi 3D-ääni asetettiin ympäristöön. Toinen asetusta, johon oli puututtava ennen varsinaisten 3D-säätöjen tekemistä, on äänen toistotapa, jonka Unity asettaa oletuksena kertaluontoiseksi. Kosken äänitehosteen on toistuttava loputtamalla silmukalla, joten *play looped* asetusta oli kytkettävä päälle. Askelluksesta tämä asetusta sen sijaan kytkettiin pois. Kolmas toistoon liittyvä valinta on äänen laukaisumenetelmä, joka oletuksena on asetettu aina sovelluksen käynnistymisen yhteyteen. Kosken kohinaa ja auton ääntä luodessa tätä asetusta ei siis tarvinnut muuttaa, mutta esimerkiksi askeltamisen äänitehosteen täytyi käynnistyä vasta silloin, kun käyttäjä liikkui ympäristössä.

5.4.1 Tilan tunnun luominen

Kosken ylittävän sillan alle määritettiin kaiunta-alue. Realistinen kaiku löytyi Unityn omista esiasetuksista. Alue suurennettiin sen kokoiseksi, että se on voimakkaammillaan sillan alla, mutta vaikutuksen voi havaita hieman sen ulkopuolellakin. Etenkin kosken kohinasta saatiin tällä menetelmällä hyvin realistisen kuuloinen. Toinen merkittävä kaiunta-alue määritettiin tunneliin. Sen akustiikka säädettiin hyvin samankaltaiseksi, kuin sillan alla, mutta efekti asetettiin voimakkaammaksi ja kaiunta-aika pidemmäksi. Myös laavun ja ympäröivän metsän akustiikkaa muokattiin. Molemmille löytyi Unityn omista esiasetuksista sopivat kaiut, joiden vaikutus oli melko mieto. Toisaalta näissä ympäristöissä ei tosielämässäkään olisi havaittavissa huomattavaa kaiuntaa.

Pelaajan ja äänilähteen etäisyyden vaikutusta simuloiva työkalu tuli käyttöön, kun ääni asetettiin 3D-muotoon. Oletuksena tässä komponentissa on logaritminen käyrä, eli mitä lähemmäs ääntä mennään, sitä selkeämmin ääni voimistuu. Kosken äänitehosteen yhteydessä vaikutus tuntui hieman liian jyrkältä, ja siksi komponenttiin asetettiin muokattu käyrä, jonka muoto kuitenkin pidettiin logaritmisena. Lisäksi enimmäisetäisyyttä, jossa kohinan voi kuulla, laskettiin 50 metriin.

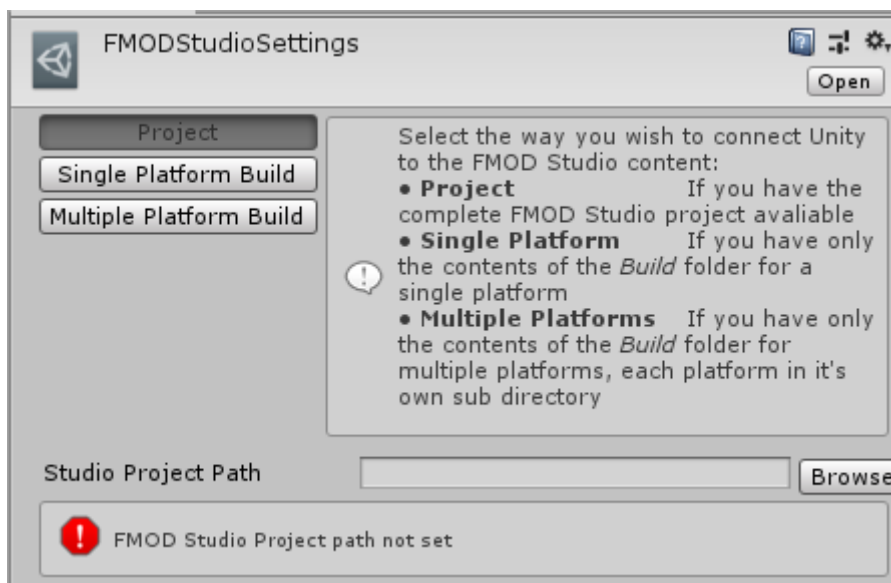
5.4.2 Oletusasetuksien muokkaaminen

Unityn oma kolmiulotteisuutta simuloiva ominaisuus tuntui aluksi toimivan riittävän realistisesti jo oletusasetuksilla. Muutaman minuutin testaamisen jälkeen siinä kuitenkin havaittiin pieni epäkohta. Äänen panorointi oli liian selkeä tapahtuakseen ulkotilassa. Kun tehosteita kuunteli selkeästi esimerkiksi vasemmalla korvalla, ei oikeaan korvaan kantautunut lainkaan ääntä. Tämä oli helppo korjata *spread* säätimellä joka ikään kuin loiventaa äänen tulosuunnan vaikutusta päästään ääntä myös äänilähteestä poispäin olevaan korvaan. Jokaisen äänilähteen *spread*-säätimen oikea arvo säädettiin kokeilemalla, eikä yhtäkään näistä jätetty ääriasentoon, jossa se oli oletuksena.

Pelkkä äänenvoimakkuuden muuttuminen loi vaikutelman etäisyyden muuttumisesta, mutta ilmiötä päätettiin korostaa lisäämällä siihen alipäästösuodin. Unityn omilla työkaluilla alipäästösuotimen raja-arvoa ei automaattisesti pysty muuttamaan, joten sitä varten täytyi luoda skripti. Yksinkertaisuudessaan tämä skripti havaitsee äänilähteen sekä pelaajan, laskee niiden etäisyyden ja muuttaa alipäästösuotimen raja-arvoa sille asetetun kertoimen mukaisesti. Tämänkin arvo oli asetettava kokeilemalla. Testatessa huomattiin, että vaikutus oli hyvä jättää melko lieväksi, sillä ilmiö ei välttämättä herätä erityistä huomiota tosielämässäkään.

5.5 FMOD

FMOD-liitännäinen ladattiin ilmaiseksi sovelluksen omilta verkkosivuilta ja sen liittäminen pelinkehitysovellukseen oli hyvin yksinkertaista. Unityyn tuodaan ulkopuolisia liitännäisiä *Asset*-valikon kautta valitsemalla *Import package*, jonka jälkeen *Custom package*. Tämän jälkeen ohjelma tuo liitännäisen ja siihen liittyvät komponentit Unityyn minkä jälkeen niitä voidaan käyttää samalla tavalla, kuin Unityn omia komponentteja. Kuvan 23 osoittamalla tavalla oli Unityn puolelta löytyvissä FMOD:n asetuksissa määritettävä projektikansioksi sama kansio, johon FMOD:n puolella tehtävä projekti tallennettiin. Sitä ennen FMOD:n puolella projekti täytyi kuitenkin ensin tallentaa ja kääntää (*engl. build*), jotta sitä voitiin käyttää ulkoisessa sovelluksessa. Tästä eteenpäin kaikki äänenmuokkaustoimenpiteet tehtiin siis nimenomaan FMOD:n puolella, eikä Unityssa. Lisäksi huomioitavaa FMOD:n käyttöön otossa on se, että niin kuuntelija-, kuin äänilähdekomentitkin on vaihdettava FMOD:n omiin komponentteihin. Mikäli FMOD ja sen komponentit otetaan käyttöön heti projektin alussa, ei se periaatteessa vaadi minkäänlaisia lisätoimenpiteitä verrattuna Unityn oman äänimoottorin käyttöön.



Kuva 23. FMOD-projektikansion valinta

5.5.1 Ääniraitojen luonti

Sovellusta varten luotiin tapahtumat koskelle, autolle, askeltamiselle ja ympäröivälle äänelle. Koskea varten tehtiin muutama eri tapahtuma ja jokaiselle annettiin eri ääniraita. Syynä usean eri ääniraidan käytölle oli sama, kuin aiemmassa vaiheessa, eli rikkauden ja dynamiikan luominen. Tapahtumaa luotaessa valitaan joko 2D- tai 3D-asetus, minkä jälkeen käyttäjälle avautuu samoja säätimiä, kuin Unityn äänityökaluissa. 3D-asetuksista tärkein lienee tapahtumaan vaikuttava etäisyyden säädin. Etäisyyden muutosta kuvaava käyrä asetettiin kokeilun perusteella oikeanlaiseksi. Tässäkin tapauksessa päädyttiin pääsääntöisesti loivaan logaritmiseen käyrrään. Oikeat arvot myös maksimi- ja minimietäisyydelle annettiin kokeilemalla. Tällä kertaa realistisimmaksi maksimiarvoksi koskelle muodostui 25 metriä. Arvo on puolet pienempi, kuin Unityn äänityökalua käytettäessä, vaikka yksikkö on sama, eli metri. Syytä tälle ei keksitty, eikä asiaan paneuduttu sen tarkemmin, koska sillä ei oikeastaan ollut merkitystä äänimaailman realismisuuden kannalta.

Koska FMOD on kuitenkin vain äänisovellus, eikä se sisällä geometriaa käsitteleviä ominaisuuksia, oli kunkin sijainnin akustiikka edelleen luotava manuaalisesti. Helpompaa FMOD:n kanssa työskentelystä teki se, ettei ohjelmointia tarvittu lainkaan. Siinä, missä Unityn oma äänilähdekomponentti toistaa vain ääniraidan, on FMOD:n komponentti kykeneväinen toistamaan myös mikseriin tallennetut tilat. Se, milloin kyseinen äänilähde aktivoituu, on käyttäjän valittavissa esimerkiksi etäisyyssäätimen avulla. Komponentissa on myös triggeriominaisuus, jolloin äänilähdekomponentti liitetään triggeriksi asetettuun läpinäkyvään laatikkoon. Sillan alle, tunneliin ja laavun sisään luotiin niille ominaiset kaiut,

jotka asetettiin aktivoitumaan triggerin avulla. Muuten ympäristössä vallitsee kaikkialla samanlainen akustiikka, joka sisältää hyvin miedon kaiun.

FMOD mahdollistaa myös eri ääniraitojen sekoittamisen saumattomasti. Tällä tarkoitetaan sitä, että ääniraitojen tai niiden efektien vaikutusalueen raja on häilyvä. FMOD:ssa tämä toimii siten, että esimerkiksi äänilähteen vaikutusalueen säteeksi voidaan asettaa 20 metriä, jolloin käyttäjän ollessa sen keskellä, ääni kuuluu täydellä voimakkuudella. Kun käyttäjä loittonee äänilähteestä alkaa sen voimakkuus pienenemään, ja 20 metrissä se katoaa kokonaan. Tämä toimintatapa on melko samanlainen, kuin normaalia 3D-työkalua käytettäessä, mutta se antaa käyttäjälle enemmän vaihtoehtoja vaikuttaa muiden parametrien mukaan muuttuvan äänen tai efektin käyttäytymiseen. Lisäksi näin toimittaessa itse ääniraitaan vaikuttavaksi parametriksi voidaan asettaa muitakin tekijöitä, kuin etäisyys.

5.5.2 Ongelma Doppler-ilmiön kanssa

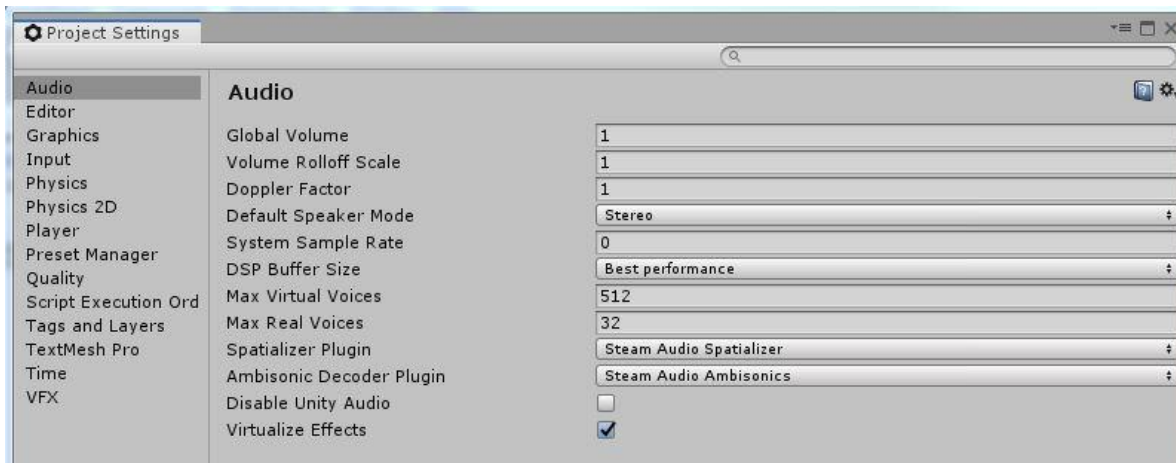
Ensimmäisenä vastaan tullut ongelma FMOD:n käytössä liittyi ohi ajavan auton äänitehosteeseen. Tehosteen realistisuuden saavuttamiseksi siihen haluttiin lisätä Doppler-ilmiötä simuloiva efekti, aivan kuten Unityn omalla äänityökalulla. FMOD tarjoaa työkalun, jossa tämän efektin käyttöönotto vaatii vain sen kytkemisen päälle, ja oikean voimakkuuden hakemisen. Näillä toimenpiteillä auton ääni ei kuitenkaan sen ohitse ajaessa muuttunut. Vastaus ongelmaan löytyi FMOD:n verkkosivun tukiosastolta. Sivusto toimii siten, että sovelluksen käyttäjät voivat kirjoittaa kysymyksiä, joihin FMOD:n henkilökunta vastaa. Jotta Doppler-ilmiö olisi saatu toimimaan, olisi auton liike pitänyt toteuttaa jollain muulla tavalla, kuin animoimalla. (FMOD Forums 2016.) Kyseinen aihealue ei kuitenkaan suoranaisesti äänimaailman luomisen liity, joten siihen ei tämän opinnäytetyön puitteissa paneuduttu tarkemmin.

5.6 Steam Audio

Aivan kuten FMOD:n, pystyi Steam Audionkin lataamaan ilmaiseksi internetistä. Lisäosan tuominen Unityyn oli samanlainen prosessi, kuin mitä tahansa muuta lisäosaa tuodessa *Import package* ja *Custom package* valikoiden kautta. Näkymään ilmestyi Steam Audion sisältämät materiaalit. Lisäksi ohjelma loi myös uuden, *Steam Audio Manager Settings*-nimisen objektin ja siihen liittyvän valikon kohtaukseen. Kyseisen objektin asetuksista pystyi muokkaamaan äänimaailmaan liittyviä seikkoja.

Ohjeet Steam Audion käyttöönottoon löytyi ladatun kansion mukana tulleesta *phonon_unity*-nimisestä HTML-tiedostosta. Lisäosan tuomisen jälkeen ensimmäisenä

muutettava asetus oli Unityn puolelta löytyvä 3D-ääniä käsittelevien lisäosien valinta. Kyseinen valikko löytyi *Edit* -valikosta *Project Settings*:n takaa *Audio*-välilehdestä. Unityn 3D-äänien käsittelyyn liittyvät lisäosat oli korvattava Steam Audion tarjoamilla vaihtoehdoilla. Kuvassa 24 on säädettyä oikeanlaiset asetukset Steam Audion toimimiseksi.



Kuva 24. Unityn ääniasetukset

Seuraavana oli muokattava itse Steam Audioon liittyviä asetuksia. Ensimmäisenä valittiin, minkä äänimoottorin kanssa sitä tullaan käyttämään. Tässä vaiheessa FMOD:n sijasta päädyttiin käyttämään Unityn omaa äänimoottoria. Näin tehtiin siksi, että Unityn äänityökalut tuntuivat tutummilta. Lisäksi ennustettiin, että yksinkertaisemman äänimoottorin kanssa Steam Audion mahdollisuudet voitaisiin havaita selkeämmin. Seuraavana asetuksissa pystyi valitsemaan lisäosan HRTF-ominaisuudet. Kyseinen lyhenne tulee sanoista *Head Related Transfer Function* ja lyhyesti selitettynä se tarkoittaa pään asennon vaikutusta kuultaviin ääniin. Tätä tarkoitusta varten ohjelmaan voi ladata SOFA-tiedoston. SOFA taas on lyhenne *Spatially Oriented Format for Acoustics*. Projektissa päätettiin käyttää oletuksena olevaa tiedostoa. *Steam Audio Manager*-komponentin valinnat on nähtävissä kuvassa 25.



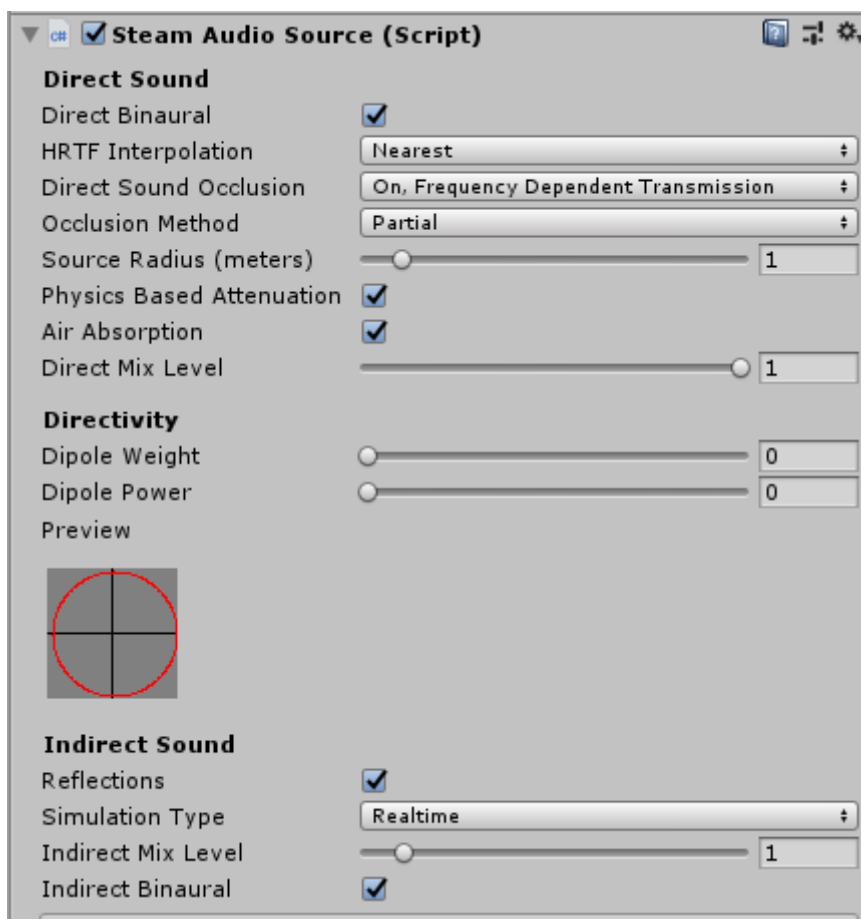
Kuva 25. Steam Audion hallintatyökalu

5.6.1 Steam Audio äänilähde

Jotta Steam Audion tarjoamat mahdollisuudet saatiin käyttöön, oli myös ääntä toistaviin objekteihin lisättävä Steam Audion oma äänilähdekomponentti, joka on esitettyinä kuvassa 26. Toisin kuin FMOD:n kanssa, ei tällä kertaa Unityn omaa äänilähdekomponenttia saanut poistaa. Ensimmäisenä käyttöön otettava valinta oli *Direct Binaural*-asetus, jonka päälle kytkennän seurauksena HRTF-tekniikkaan perustuvat 3D-äännet tulivat käyttöön. HRTF interpolointivalinnalla voitiin vaikuttaa siihen, kuinka monta eri äänen lähtösuuntaa vaikuttaa lopulta kuultavaan äänen.

Direct Sound Occlusion valikolla vaikutettiin äänen tukkeutumiseen esteen ollessa äänilähteen ja pelaajan välissä. Kyseisen ilmiön vaikutukselle oli neljä eri vaihtoehtoa. Ensimmäisenä näistä tukkeuman pystyi kytkemään kokonaan pois päältä, eli valitsemalla *Off*. Seuraavana vaihtoehtona oli *On, No Transmission*, jonka valittua tukkeuman vaikutus oli joko täydellinen tai olematon. Tällöin vähänkään tukkeutuneita ääniä ei siis kuultu laisinkaan, edes heijastuksen kautta. Kyseinen valinta voisi toimia hyvin äänieristettyä huonetta mallinnettaessa. Kolmas valinta *On, Frequency Independent Transmission* oli lähempänä realistisempaa tosielämän vaikutusta, sillä se vaikutti jo äänen sisältöön. Tällä valinnalla

saatiin aikaiseksi äänen sisältämien taajuuksien katoaminen ympäröivien ja äänen kulkeutumista estävien objektien mukaisesti. Kyseinen vaihtoehto on hieman alkeellisempi seuraavaan verrattuna, joka laski jokaisen taajuusalueen vaimenemisen erikseen. *On, Frequency Dependent Transmission*, eli neljäs vaihtoehto loi realistisimman vaikutuksen suuremmillakin etäisyyksillä. Tukkeumametodiksi, eli *Occlusion Method*-kohtaan voitiin valita osittainen tai yksittäiseen säteeseen perustuva tukkeuma. *Partial*, eli osittainen perustuu äänilähteestä lähtevään useaan säteeseen ja niistä laskettuun vaimennukseen, jonka sovelluksen käyttäjä kuulee. Vaihtoehtona oleva *Raycast* lähettää kuulijalle vain yhden säteen, ja suorittaa sille tukkeutumaa simuloivia toimenpiteitä etäisyyden, välissä olevien esteiden ja ympäröivien objektien mukaisesti. Suurimpaan osaan objekteista käytettiin osittaista tukkeutumaa, sillä sen ajateltiin olevan realistisempi joka puolelle säteilevän äänen yhteydessä.



Kuva 26. Steam Audion äänilähde-komponentti

5.6.2 Edistyneempi etäisyyden tarkastelu

Seuraavana oli valittava, otetaanko ilman ääniaaltoja imevä vaikutus käyttöön. *Air Absorption* valinnalla saatiin aikaan sama efekti, joka toteutettiin kappaleessa 5.4.2 ohjelmoimalla. Valinnan seurauksena sen lisäksi, että äänilähteestä kauemmas liikuttaessa äänen voimakkuus laski, katosi siitä myös korkeimpia taajuuksia. *Direct Mix Level* säätimellä sen sijaan voitiin vaikuttaa niiden äänien voimakkuuteen, jotka kantautuvat sovelluksen käyttäjän kuultaviksi suoraan äänilähteestä. Myös äänien suuntaavuuteen voitiin vaikuttaa *Directivity*-kohdan säätimillä. *Dipole Weight*-säätimellä äänestä voitiin tehdä joko yhteen suuntaan kantautuva, kaksisuuntainen tai jotain siltä väliltä. *Dipole Power* taas vaikutti siihen, kuinka kapea äänen kuuluvuusalue on.

Lopuksi kytkettiin päälle heijastukset. Suurimpaan osaan äänilähteistä käytettiin reaaliaikaista toistotapaa, mutta myös heijastuksien kypsyttämistä kokeiltiin. Sovellus ei vaadi tietokoneelta kovin suurta suoritustehoa, eikä tällä toimenpiteellä havaittu olevan minkäänlaista vaikutusta suorituskyvyn tai äänenlaadun kannalta. Näiden toimenpiteiden jälkeen havaittiin, että sovellus oli tässä vaiheessa äänimaailmaltaan aivan toisella tasolla, kuin aiemmat versiot.

6 YHTEENVETO

Opinnäytetyön aihealue osoittautui huomattavasti laajemmaksi, kuin mitä projektin alkuvaiheessa osattiin ennustaa. Mikäli yhdelle ihmiselle annetaan tehtäväksi toteuttaa virtuaalitodellisuussovelluksen äänimaailma yksin, on hänen syytä tutustua aiheeseen läpikotaisin sekä tehdä tarkka suunnitelma siitä, mitä sovellus tulee pitämään sisällään. Myös aikaa on syytä varata paljon, sillä kaikinensa tällainen prosessi voi muotoutua hyvinkin pitkäkestoiseksi. Äänisuunnittelusta, äänittämisestä sekä äänien ohjelmoinnista olisi voinut kirjoittaa kaikista omat opinnäytetyönsä, mutta silloin tutkimuksessa olisi todennäköisesti pitänyt mennä hieman syvemmälle. Esimerkiksi tämän opinnäytetyön äänittämistä käsittelevässä luvussa esiteltävät asiat toki mahdollistavat onnistuneen äänityssession, mutta alan todellinen hallitseminen vaatii toimenpiteiden toistamisen useaan kertaan.

Äänitettyjen raitojen sovellukseen istuttamisen jälkeen huomattiin, että raidat olivat itse asiassa todella hyvin onnistuneita. Kun niille annettiin visuaaliset vastineet, heräsivät ne ikään kuin eloon. Koko äänitysprosessi eteni hyvin jouhevasti, eikä suurempia ongelmia tullut vastaan. Toki koskea olisi voitu äänittää alun perin suunnitellussa ympäristössä, mutta onneksi sille löydettiin korvaaja.

Ohjelmointivaiheessa erillinen visuaalisen puolen ohjelmoija olisi nopeuttanut prosessia. Vajaa viikko ehti kulua kohtauksen luomiseen ennen kuin yhtään ääntä oli saatu asetettua sovellukseen. Äänityökaluista helppokäyttöisimmältä tuntui Steam Audio, ja kyseisellä työkalulla saatiin myös realistisin lopputulos. Steam Audion käyttöönottoa puoltaa myös se, että se on ilmainen ja vapaasti käytettävissä. Yksityisien pelinkehittäjien, peliharrastajien ja miksei ammattilaistenkin kannattaa tutustua kyseiseen lisäosaan, sillä sen avulla voidaan saavuttaa realistisia tuloksia pienelläkin vaivalla. FMOD on kuitenkin yhä äänimootorien ja -työkalujen standardi, vaikka Steam Audiolla koko prosessi on huomattavasti helpompi ja nopeampi toteuttaa.

LÄHTEET

Alten, S. R. 2011. Audio in Media. Cengage Learning

Blomber, E. & Lepoluoto, A. 1991. Audiokirja. Helsinki.

Case, A. U. 2007. SoundFX, Unlocking the Creative Potential of Recording Studio Effects. Taylor & Francis.

Collins, K. 2002. Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design. ProQuest Ebook Central. MIT Press.

DarkEco. 2016. Creating Wind (White Noise/On Any Synth!) [viitattu 7.10.2019] Youtube. Saatavissa: <https://youtu.be/1oe4S907-oM>

Dietz, J. 2019. Metacritic Features Editor (2019): Metacritic's 9th annual game publisher rankings [viitattu 29.4.2019]. Metacritic. Saatavissa: <https://www.metacritic.com/feature/game-publisher-rankings-for-2018-releases>

FMOD Forums. 2006. Doppler effect FMODUnity integration not working [viitattu 9.10.2019]. Saatavissa: <https://qa.fmod.com/t/doppler-effect-fmodunity-integration-not-working/12836>

Karisto, H, Kenttämies, J, Koivumäki, A & Korpinen, P. 2006. Analoginen ääni [viitattu 12.6.2019]. Saatavissa: http://www.aanipaa.tamk.fi/analog_1.htm

Korpinen, P. 2006. Äänipää: Arkielämän ääni-ilmiöitä, [viitattu 23.8.2019] Saatavissa: http://www.aanipaa.tamk.fi/arki_1.htm

Lanham, M. 2017. Game Audio Development with Unity 5.X. Birmingham: Packt Publishing Ltd.

Luotola, J. 2014. Onko kaarevassa tv:ssä mitään järkeä? Asiantuntijat paljastavat sen mitä kuluttaja ei hoksaa [viitattu 25.4.2019]. tivi.fi Saatavissa: <https://www.tivi.fi/uutiset/onko-kaarevassa-tvssa-mitaan-jarkea-asiantuntijat-paljastavat-sen-mita-kuluttaja-ei-hoksa/28789623-2e34-3a64-a002-fd5b11c6a0a9>

Mäkelä, J. P. 2002. Kotistudio. Helsinki: Like kustannus Oy.

Røde Microphones. 2019. Røde NT-SF1 [viitattu 17.7.2019]. Saatavissa: <https://www.rote.com/ntsf1>

Russ, M. 2009. Sound synthesis and sampling. IMW Curriculum Manager. Oxford: Elsevier Ltd.

- Stanton, B. W. 2016. Best Surround Sound Speaker Deals Black Friday [viitattu 16.4.2019]. Gazettereview. Saatavissa: <https://gazettereview.com/wp-content/uploads/2016/11/Sourround-Sound-Featured.jpg>
- Tervola, J. 2017. Näin suunnitellaan nykyaikainen elokuvasali – akustiikan simuloijasta selkeä arvio: ”Hän on nero” [viitattu 25.4.2019]. Tekniikka & Talous. Saatavissa: <https://www.tekniikkatalous.fi/ttpaiva/nain-suunnitellaan-nykyaikainen-elokuvasali-akustiikan-simuloijasta-selkea-arvio-han-on-nero-6637205>
- Thom, R. 1999. Designing A Movie For Sound [viitattu 3.6.2019]. Filmsound.org. Saatavissa: http://filmsound.org/articles/designing_for_sound.htm
- Thomann. 2019. the t.akustik Wedges 30 [viitattu 15.10.2019] Saatavissa: https://www.thomann.de/fi/the_t.akustik_wedges_30.htm?sid=61b4c9e080661e3c65804dea82cf1a97
- Unity technologies. 2019a. GameObjects [viitattu 29.7.2019]. Unity Documentation. Saatavissa: <https://docs.unity3d.com/Manual/GameObjects.html>
- Unity technologies. 2019b. AudioListener [viitattu 12.8.2019]. Unity Documentation. Saatavissa: <https://docs.unity3d.com/Manual/class-AudioListener.html>
- Unity technologies. 2019c. AudioSource [viitattu 12.8.2019]. Unity Documentation. Saatavissa: <https://docs.unity3d.com/Manual/class-AudioSource.html>
- Unity technologies. 2019d. AudioMixer [viitattu 12.8.2019]. Unity Documentation. Saatavissa: <https://docs.unity3d.com/Manual/class-AudioMixer.html>
- Unity technologies, 2019e. EventTrigger [viitattu 12.8.2019].Unity Documentation. Saatavissa: <https://docs.unity3d.com/Manual/script-EventTrigger.html>
- Unity Technologies, 2019f. Baked Lighting [viitattu 12.8.2019]. Unity Documentation. Saatavissa: <https://docs.unity3d.com/Manual/LightMode-Baked.html>
- Valve Corporation. 2017. Steam Audio [viitattu 20.8.2019]. GitHub. Saatavissa: <https://valvesoftware.github.io/steam-audio/>