

Lapin AMK, pLab

# 360 ÄÄNI 101

Tee-se-itse-äänituotanto virtuaalitodellisuuden sovelluksiin Unityssa

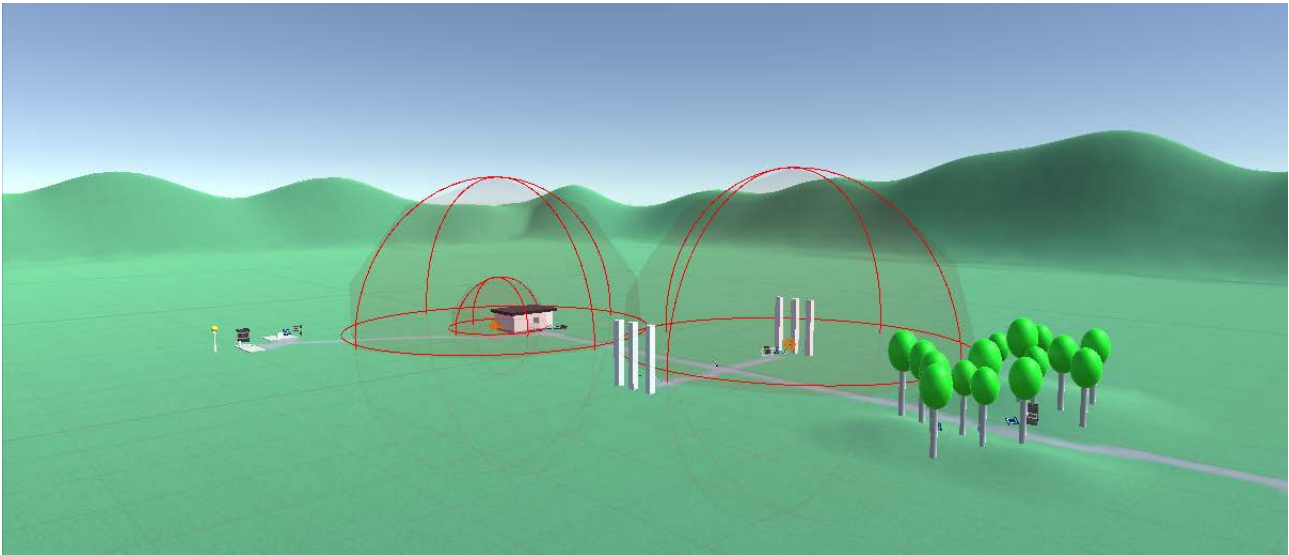
Heikki Tikkanen  
2-13-2019

## Sisällysluettelo

Demoprojektista valmiit työkalut ja esimerkit .....	1
Spatialisointilisäosat Unityyn.....	1
VR:n eri muodoista .....	2
Spatialisointi ja tilamallinnus .....	3
Äänisuunnittelu = kirjoita lista.....	4
Tallenna ambienssiäni oikeasta kohdasta .....	5
Äänitä pisteäänit sopivalta etäisyydeltä.....	5
Tallennuslaitteista ja -asetuksista.....	6
Joku muu on ehkä äänittänyt sen jo.....	6
Editointi pähkinäkuoressa .....	7
Editointiohjelmia .....	9
Saumaton luuppi editoimalla .....	9
Saumaton luuppi koodilla.....	11
Pitkien tiedostojen tuontiasetukset .....	12
Ambisonic-tiedostojen tuontiasetukset .....	12
3D-tilaan sijoitettu ambisonic-ääni Unityssa.....	12
VR-projektin äänten toteutus Unityssa vaihe vaiheelta.....	13
Oculuksen spatialisointilisäosan asentaminen Unityyn .....	13
Lisäosan käyttöönotto .....	14
Äänilähteiden lisääminen ja spatialisointi .....	15
Äänen vastaanotin – Audio Listener.....	15
Äänilähde – Audio Source.....	16
Äänilähteen spatialisointi Oculus Audio SDK:lla.....	17
Akustisen geometrian lisääminen Oculus Audio SDK:lla .....	18

Tämä opas on johdatus virtuaalitodellisuuden äänimaisemien toteuttamiseen. Erilaisten VR-sisältöjen ja niissä käytettävän äänen perusteiden jälkeen käydään läpi, kuinka Unity-pelimoottorilla luodaan eri tarkoituksiin sopivia äänitehosteita Oculus Rift –virtuaalilaseilla toimivaan VR-sovellukseen.

## Demoprojektista valmiit työkalut ja esimerkit



pLab on toteuttanut tässä oppaassa kuvatuilla menetelmillä Unityssa toimivan, riisutun esimerkkiprojektin. Projektissa voit kokeilla VR:ssä eri tavoilla luotujen äänten eroja. Esimerkkien ohessa selitetään tekstinä niihin liittyviä hyviä käytäntöjä. Projektin sisältää [Oculus Integration](#) ja [VRTK](#) –pluginit omine esimerkkeineen, ja sitä voi käyttää lähtökohtana omien projektien toteutuksessa.

Löydät esimerkkiprojektin GitHubista Lapin Elämysteknologiat 2025 –hankkeen repositoriosta.

Jos projektin kaikki toiminnallisuudet eivät vaikuta toimivan kuten pitäisi, valitse Unityn valikosta Assets – Reimport All.

## Spatialisointilisäosat Unityyn

Tässä oppaassa käsitellään [Oculus Audio SDK](#):n käyttöä.

Unityssa voi käyttää myös muita spatialisointilisäosia. Merkittävimmät niistä ovat tällä hetkellä:

- [Steam Audio Plugin](#)
- [Resonance Audio](#) (entinen GoogleVR Audio)
- [Microsoft Audio Spatializer](#) (Tazman Audion plugin)

## VR:n eri muodoista

VR:n perusajatus on, että käyttäjä kokee olevansa kokonaan toisessa, hänet ympäröivässä todellisuudessa. Koettavaa on edessä, takana, sivuilla, päällä ja alla. Puhutaan 360-sisällöstä tai –kokemuksesta. Käyttäjää joka puolelta ympäröivä toinen todellisuus on kaikenlaisille VR-sisällöille yhteinen tekijä.

Kokemus voi kuitenkin vaihdella paljonkin sen suhteen, mitä käyttäjä voi tehdä tuossa ympäröivässä todellisuudessa sen lisäksi, että vain uppoutuu siihen ja kokee olevansa jossain muualla. VR-sisältöä voi tehdä ja kokea erilaisilla periaatteilla samoin kuin valokuvaa, videota tai videopelejä. Näille kaikilla sisältömuodoilla on omat lainalaisuutensa äänen ja muidenkin kokemukseen vaikuttavien osien suunnittelussa.

Valokuva on esikuva 360-kuvalle. Se otetaan kameralla, joka näkee joka suuntaan yhtä aikaa. Sitä voi katsella VR-laseilla. Kääntämällä päätään käyttäjä näkee eri osan itseään ympäröivästä kuvasta. 360-videossa on sama periaate, mutta siinä on lisäksi ajallinen ulottuvuus. Kohteet saattavat liikkua käyttäjää ympäröivän ”videokuplan” sisäpinnalla, tai videossa näkyvän maiseman valaistus saattaa muuttua ajan myötä. 360-kuvia ja –videoita voi nähdä esimerkiksi YouTuben (Jump) ja Facebookin kaltaisissa palveluissa.

360-kuvassa tai –videossa äänikin voi seurata käyttäjän pään liikettä yhdessä kuvan kanssa. Tässä kohtaa puhutaan spatiaalisesta äänestä, 3D-äänestä tai 360-äänestä, jotka tarkoittavat samaa asiaa. Spatiaalisen äänen lisäksi perinteinen stereoääni on VR:ssä yhä käyttökelpoista, ja toisinaan näitä kahta muotoa yhdistellään VR-sovelluksen sisällöstä ja käyttötarkoituksesta riippuen.

Pelien tapaan toimivat VR-sovellukset voivat sisältää edellä mainittuja sisältömuotoja, mutta niiden idea on tarjota käyttäjälle katsojan ja kuulijan roolin lisäksi enemmän tekemistä. VR-pelien pelaaminen yleistyy hitaasti mutta varmasti kuluttajien keskuudessa. Pelimäisen, vuorovaikutteisemmän VR:n tuottaminen vaatii enemmän teknologiaa ja ohjelmointia kuin VR-kuva- ja video. Kuvan ja äänen tallentamisen ja editoimisen lisäksi ne täytyy tuoda erilliseen ohjelmistoon, pelimoottoriin, jolla rakennetaan se virtuaalinen todellisuus, johon käyttäjä lopulta uppoutuu. Pelimaailmat pohjautuvat usein 3D-geometriaan ja tilaan, jossa käyttäjä voi liikkua ja manipuloida asioita, vaikkapa siirrellä, rikkoa tai koota esineitä. Pelillisen VR-sovelluksen vuorovaikutteisuus täytyy lähes aina ohjelmoida räätälöidysti juuri sen sovelluksen käyttäjäkokemusta varten. VR-peleissä käytetään lasien lisäksi myös käsissä pidettäviä ohjaimia, joilla mm. pelimaailman manipulointi ja käyttäjän avatarin liikuttelu tapahtuu.

Pelillisissä VR-sovelluksissa ääntä koskevat samat asiat kuin kuvaa ja videota. Ääni voidaan rakentaa spatiaaliseksi, eli seuraamaan käyttäjän pään suuntaa ja kuulumaan tietystä kohtaa suhteessa kuvaan. Toisaalta voidaan käyttää edelleen myös perinteistä stereoääntä, tai molempia muotoja tarkoituksen mukaan. Työtä kuitenkin mutkistaa se, että nyt käyttäjä voi lisäksi liikkua eri paikkaan virtuaalitodellisuudessa. Äänimaiseman on muututtava myös käyttäjän sijainnin mukaan. Lisäksi käyttäjä voi manipuloida asioita, joiden tulisi luonnolliselta tuntuakseen pitää ääntä ”käskystä”. Pelimoottorit ovat onneksi mainioita työkaluja näiden asioiden aikaansaamiseksi.

## Spatialisointi ja tilamallinnus

Spatialisointi tarkoittaa äänen luonnollista käyttäytymistä mallintavaa tekniikkaa, jolla siitä tehdään käyttäjälle todentuntuisempaa ja tavallaan kolmiulotteisempaa. Käytännössä spatialisoinnilla luodaan monimutkaisempi miksaus, joka korvaa pelimoottorin normaalisti tuottaman vasen-oikea-stereomiksauksen. Normaalisti pelimoottori miksa äänen voimakkuutta ja suuntaa vain vertaamalla äänilähteen sijaintia äänen vastaanottajan sijaintiin. Nämä ovat vain pisteitä avaruudessa, eikä miksausessa oteta huomioon esimerkiksi suuntaa, johon molemmat osoittavat. Myöskään niiden sijaintia pystyakselilla ei huomioida, ja miksaus mallintaa niiden suhteellista sijaintia ja etäisyyttä vain vaakasuunnassa, vasen-oikea-akselilla.

Spatialisointi muuttaa miksaus tuottavaa geometristä laskentaa. Merkittävin spatialisoinnissa tapahtuva laskenta on se, että huomioidaan, missä asennossa äänen vastaanotin on, eli mihin käyttäjän virtuaalisen korvat sijoittuvat 3D-avaruudessa suhteessa soivaan äänilähteeseen. Luonnossa korvat kuulevat saman äänen hieman eri aikaan ja eri voimakkuudella riippuen pään asennosta. Tämän mallintaminen miksausessa vaikuttaa valtavasti siihen, miten äänen suunta hahmotetaan. Tavanomaisen vasen-oikea-miksauksen sijaan spatialisoitu ääni miksataan tavalla, jossa erottuvat myös eteen-taakse ja ylös-alas – suunnat. Äänestä tulee siis kolmiulotteisessa tilassa helpommin ja realistisemmin havaittavaa.

Spatialisoinnissa molemman virtuaalisen korvan kuulema informaatio lasketaan erikseen HRTF-funktiolla ([Head-Related Transfer Function](#)), ja tästä muodostuu pelimoottorien tavallista miksausta hienovaraisempi miksaus. Oculuksen käyttämä HRTF on tilastollinen yleistys ihmiskorvan tavasta kuulla. Todellisuudessa jokaisella meistä on omat kehon, pään ja korvan rakenteesta riippuvat HRTF-parametrit. Yksilöllinen HRTF on kuin sormenjälki, jonka tallentamiseen on olemassa jo standardi tiedostomuoto ([SOFA](#)). Sillä voidaan personoida esim. spatialisointialgoritmeja ja kuulokkeiden taajuusvasteita. Jotkut yritykset tarjoavat palvelua yksilöllisen HRTF:n tallennukseen.

Tässä oppaassa mainitut spatialisointiratkaisut tarjoavat muitakin ominaisuuksia, jotka eivät tarkalleen ottaen ole spatialisointia edellä kuvatulla tavalla, mutta vaikuttavat äänen tilalliseen hahmottamiseen erittäin paljon. Niitä voi kutsua yhdellä termillä esimerkiksi tilamallinnukseksi tai akustiseksi geometriaksi.

Tilamallinnus spatialisoinnin lisäksi tehtävää laskentaa, jolla voidaan entisestään rikastaa ja tarkentaa vaikutelmaa äänilähteen sijainnista, äänilähdettä ympäröivästä tilasta sekä tehdä äänestä todentuntuisempi. Tilamallinnuksella voi määritellä helposti varsinkin 3D-geometrialla rakennetuille tiloille ja objekteille, miten ne heijastavat, vaimentavat ja välittävät eri äänentaajuuksia. Nämä ominaisuudet ovat varsinkin sisätiloihin sijoittuvissa VR-kokemuksissa erittäin hyödyllisiä. Esimerkiksi kuution mallinen huone kuulostaa erilaiselta betoniseinillä kuin puisilla seinillä, ja vaikkapa keittiön kaikuminen vähenee, kun sinne lisätään verhot ja matto. Imurin ääni kuulostaa eriltä kuullessaan toisesta huoneesta suljetun oven takaa kuin aivan vierestä samassa huoneessa.

Pelimoottorien normaalit tavat mallintaa äänen käyttäytymistä eivät yllä lähellekään tällaista toiminnallisuutta yhtä vähällä vaivalla. Niissäkin on valmiina esimerkiksi kaikuja tilan tunnun luomista varten, mutta niiden saaminen kuulostamaan hyvältä vaatii melko paljon käsin säätämistä, eikä lopputulos ole yhtä hyvä.

Spatialisointi ja tilamallinnus ovat ainakin vielä melko raskaita suoritusteholle, joten kaiken äänen spatialisoiminen ja mallintaminen akustisesti ei ole läheskään aina järkevää. Toisinaan riittää pelkkä spatialisointi, jonka pitäisi olla tilamallinnusta jonkin verran kevyempää laitteistolla.

Ulkotiloissa tilamallinnus ei useinkaan ole tarpeen, mutta spatialisoinnista on niissäkin hyötyä. Jokaiselle projektille paras ratkaisu on aina omanlaisensa sekoitus, jossa on todennäköisesti kaikenlaista ääntä: spatialisoimatonta, spatialisoitua sekä spatialisoitua ja tilamallinnettua. Aivan kuten grafiikassa suurinta huomiota saavat kohteet usein tuotetaan tarkimmin ja huolellisimmin, äänessäkin tarkin mallinnus kannattaa keskittää käyttäjälle oleellisimpiin ääniin, ja taustalle jäävät äänet voidaan hoitaa kevyemmin.

## Äänisuunnittelu = kirjoita lista

Ennen kuin edes tallennetaan ääntä, kannattaa punnita, mikä on koko VR-tuotannon pointti, mitä käyttäjän pitäisi kuulla, ja miltä sen pitäisi tuntua. Tämän määrittelyn pohjalta on hyvä suunnitella mahdollisimman pitkälle, mitä ääniä äänitetään, kuinka paljon, miltä etäisyyksiltä ja mistä suunnista, miten ääni toistetaan sovelluksessa, jne. Nämä asiat kannattaa kirjata ylös ennen kuin menee yhtään pidemmälle. Lista ei välttämättä ole pitkä – itse asiassa sitä parempi mitä lyhyempi – mutta se on ainakin hyvin mietitty, ja turhan työn mahdollisuus pienenee heti kättelyssä.

VR:llä kuten millä tahansa muulla sovelluksella tai sisällöllä pyritään saamaan aikaan aina tietynlainen kokemus, ja äänen tulisi tukea tuota kokemusta. Onko käyttäjän tarpeellista pystyä liikkumaan ja/tai manipuloimaan asioita, vai riittääkö pelkkä pään kääntely ja paikallaan olo ilman vuorovaikutusta ympäristön kanssa, kuvan ja videon tapaan? Miltä koko kokemuksen pitäisi tuntua? Onko ympäristön äänineen tarkoitus vaikuttaa mahdollisimman realistiselta, vai ehkä jopa surrealistiselta? Todellisen maailman toisintaminen ei aina ole järkevää, ja joskus joitain luonnossa kuuluvia asioita kannattaa jättää kokonaan pois, jotta käyttäjän kokemukselle tärkeimmät äänet saavat ansaitsemansa huomion.

VR-ääni suunnitellaan aina kuulokkeilla koettavaksi, ellei kyseessä ole jokin harvinaisempi toteutustapa, kuten esimerkiksi surround-järjestelmässä toistettava installaatio tms. jossa VR-lasien käyttäjän lisäksi kokemuksessa on mukana muitakin ihmisiä, joiden tulisi kuulla VR-todellisuuden äänet. Oculus Rift – laseissa on mukana kiinteät stereokuulokkeet. Oletuksena Unity-pelimoottori toistaa äänen stereokuulokkeisiin sopivalla tavalla, mutta toiston voi vaihtaa projektin asetuksista esim. 5.1-surround-yhteensopivaksi.

Unity huolehtii automaattisesti monesta äänen toistoon liittyvästä asiasta puolestasi. Tehtäväksesi jää äänitiedostojen tallentaminen ja editoiminen, niiden tuonti pelimoottoriin, sijoittelu 3D-avaruuteen ja äänilähteiden hienosäätöjen tekeminen. Nämä vaiheet käydään läpi seuraavaksi.

## Tallenna ambienssiäni oikeasta kohdasta

VR-kokemuksissa on äänen kannalta tyypillisesti tärkeää se, että sillä tuetaan ympäröivään todellisuuteen uppoutumista. Yksittäiset pistemäiset äänet eivät välttämättä ole niin tärkeitä kuin virtuaalisen tilan yleinen äänimaisema, ambienssi. Näin on usein varsinkin sovelluksissa, joissa ei liikuta.

Ambienssiänen lähdemateriaalin tallennukseen on periaatteessa kaksi tapaa. Tavanomainen vaihtoehto on tallentaa tavallista mono- tai stereoääntä useista kohdista ja sijoitella nämä äänet sovelluksessa omiin pisteisiinsä eri puolille käyttäjää. Näin luodaan vaikutelma, että pelaaja on taustäänen ympäröimä.

Toinen, tällä hetkellä yleistynyt vaihtoehto on taltioida ambienssiäni nauhurilla, joka tallentaa sen valmiiksi 360-muotoon. Yleinen formaatti ja termi tällaiselle tallentamiselle on ambisonic, tai B-formaatti. Unity ja Oculusen äänirajapinta osaavat toistaa ambisonic-formaattia sellaisenaan, joten helpoimmillaan VR-äänen saa aikaan äänittämällä ambisonic-tiedoston, tuomalla sen Unityyn oikeilla asetuksilla ja sijoittamalla äänilähteen käyttäjän pään kohdalle. Äänittäessä tällaista materiaalia myös tallennin kannattaa sijoittaa kohtaan, joka vastaa käyttäjän pään sijaintia lopullisessa sovelluksessa. Esim. maan tasolta äänitetty ambienssi ei tunnu luonnolliselta käyttäjän korvissa, jos sovelluksessa kamera on seisovan ihmisen silmien korkeudella.

## Äänitä pisteänet sopivalta etäisyydeltä

Mahdollisia pisteääniä kannattaa pyrkiä tallentamaan niin, että ne saa eristettyä ympäröivästä taustamelusta. Tässä kannattaa käyttää mikkiä tai tallenninta, joka poimii ääntä tiukasti tietystä suunnasta, eikä joka puolelta 360-tallentimen tapaan. Mitä lähempää äänität, sitä vähemmän mukaan tulee taustamelua. Mahdollisimman läheltä ei silti ole aina paras, sillä äänen tulisi tuntua luonnolliselta sillä etäisyydellä jolla käyttäjä katselee pisteäänen lähdeä sovelluksessa. Jos näet auton ajavan 50 metrin päässä, mutta kuulet sen äänen kuin olisit aivan sen vieressä, kokemus tuntuu luonnottomalta.

Unityssa äänilähteen säädöillä ja efekteillä voi saada läheltäkin äänitetyn äänen kuulostamaan luonnolliselta kaukaa, mutta oikeiden säätöjen löytäminen voi olla työlästä. Tosin esim. Oculusen omat spatialisointiominaisuudet auttavat tässä, koska ne kaiuttavat ja vaimentavat ääntä sen lähdeä ympäröivän 3D-geometrian perusteella. Näitä ominaisuuksia käsitellään tarkemmin tuonnempana.

Voit myös äänittää samaa pisteäänen lähdeä useilta etäisyyksiltä, ja editissä miksata näitä ääniä päällekkäin eri suhteissa saadaksesi parhaan tuloksen. Tähän vaaditaan joko useita yhtä aikaa nauhoitettavia mikkejä tai tallentimia, tai se, että ääni toistuu tarpeeksi samanlaisena koko ajan tai joka kerta, jotta sen voi äänittää samalla tallentimella ja miksata äänet päällekkäin ilman että se kuulostaa kahdelta eri äänitapahtumalta. Kannattaa punnita jälleen, mikä on kunkin äänen tarkoitus ja kuinka luonnolliselta niiden pitää kuulostaa, ja valita sopivin tallennustapa sen mukaan.

## Tallennuslaitteista ja -asetuksista

Markkinoilla on paljon kohtuuhintaisia tallentimia, jotka soveltuvat hyvin pisteäänien tallennukseen. Osaan voi liittää ulkoisia mikrofoneja ja tallentaa useita raitoja yhtä aikaa. Verkossa on paljon [oppaita ja vertailuja](#) sopivan nauhurin valintaan. pLabissa käytetään yleensä [Zoom H6](#)-tallenninta.

Ambisonic-/360-nauhureita on vähemmän, mutta niitäkin saa jo kohtuullisella hinnalla. Niillä voi tallentaa vaivattomasti kolmiulotteisin ambienssiäänien, joka toimii sellaisenaan Unityssa. Kannattaa tarkistaa tallentimen tuottama formaatti ennen ostoa. Unityn ambisonic-toistoa varten sen tulisi olla B-format tai "first order ambisonics". Esimerkiksi Zoom valmistaa huokean hintaista mutta monipuolista 360-nauhuria: [Zoom H3-VR](#)

Käytä nauhurissa hyvänlaatuista tallennusformaattia. 16-bittinen WAV-tiedosto 44,1 kHz näytteenottotaajuudella on aivan riittävä. Korkeammalla taajuudella sinulla on editissä mahdollisuuksia tehdä tarkempia leikkauksia jne. Tarkkuus tulee nauhurin tallennustilan ja editointikoneen prosessointitehon kustannuksella. Älä ikinä nauhoita MP3-muodossa mitään, ellei ole aivan pakko.

## Joku muu on ehkä äänittänyt sen jo

Valmiiden äänipankkimateriaalien käyttö on joskus hyvä vaihtoehto itse tallentamiselle. Joskus se on ainoa vaihtoehto. Heinäsirkan sivitystä tai vaikkapa haarapääskyn laulua on mahdotonta äänittää talvella. Tällöin äänipankit ovat avuksi. Internet on pullollaan monenlaista erittäin laadukastakin sisältöä kohtuuhinnalla tai jopa ilmaiseksi. [Freesound.org](#) lienee tunnetuin sivusto, jolta voi ladata ilmaisia ääniä, kunhan sitoutuu noudattamaan niiden lisensointiehtoja. Esimerkiksi [Sonniss.com](#) ja [Sounddogs.com](#) tarjoavat maksullisia äänikirjastoja. Sonniss.com jakaa ilmaiseksi verkossa vuosittain ison pelikehittäjille suunnatun äänipaketin.

Äänipankeissa on kuitenkin tiettyjä ongelmia. Jos tähdätään tiettyyn paikkaan tai alueeseen perustuvaan realistiseen tai autenttiseen kokemukseen, paras ratkaisu olisi äänittää materiaali itse juuri tuosta paikasta. Keskieurooppalaiset kaupungit tai metsät kuulostavat erilaiselta kuin lappilaiset, eikä lapissa äänitettyä materiaalia ole tarjolla paljon. Haarapääskyn äänikin on melko harvinainen löytö äänipankeista, ja suomalainen heinäsiirikka kuulostaa eriltä kuin amerikkalainen. Äänipankkimateriaalia on sekä ilmaista että maksullista. Maksullista voi käyttää kaupallisiin tarkoituksiin, mutta ilmaisten äänten lisenssien kanssa täytyy olla tarkkana.

Ambisonic-ääntä on tarjolla äänipankeissa vielä melko vähän, mutta määrä on lisääntymässä nopeasti. Sitä on vaikea löytää ilmaiseksi, ja maksullisena sen hinnat ovat korkeammat kuin tavallisemmilla äänitteillä. Esimerkiksi [ASoundEffect](#) ja [ProSoundEffects](#) tarjoavat melko kattavasti erilaisia ambisonic-äänitteitä, mutta hinnat ovat joissakin niistä melko kovia.

Oculus tarjoaa ilmaiseksi muutamia ambisonics-malliääninä: [Oculus Ambisonics Starter Pack](#) ja [Oculus Audio Pack 1](#).

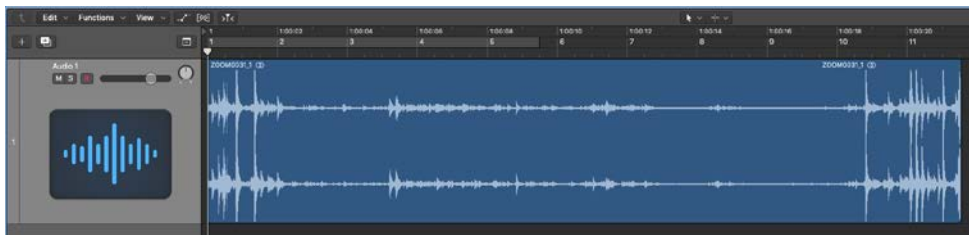


## Editointi pähkinäkuoressa

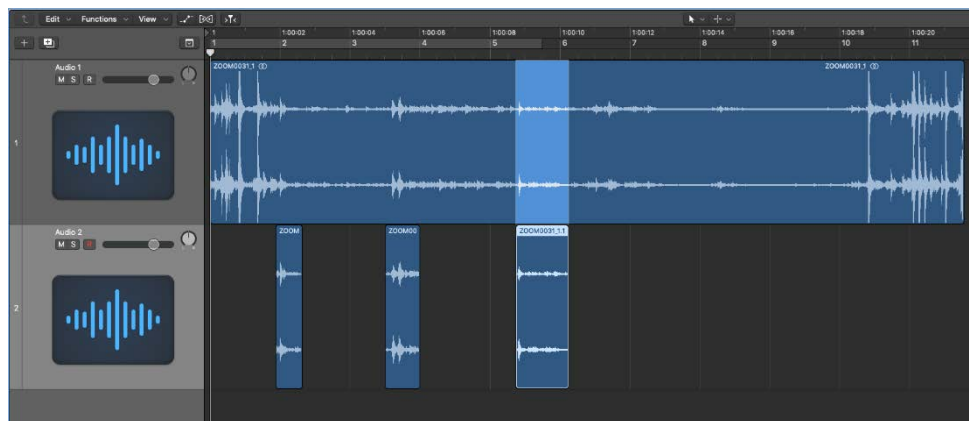
Kun äänitöt, äänitö kerralla paljon. Ota samasta äänestä useita otoksia. Näin sinulla on editissä mahdollisuuksia valita parhaat palat isosta määrästä materiaalia. Kun käytät äänittämisen työjärjestyksenä suunnitteluvaiheessa luomaasi listaa, turhaa materiaalia ei pitäisi juuri syntyä.

Tässä oppaassa ei mennä äänieditoinnin yksityiskohtiin, sillä erilaisia editointiohjelmistoja on valtavat määrät (ks. linkkejä alla), ja tietyllä ohjelmalla annettavat esimerkit eivät välttämättä päde muissa. Perusasiat ja työnkulku ovat kuitenkin aina jotakuinkin seuraavat. Esimerkin ohjelma on [Logic Pro X](#).

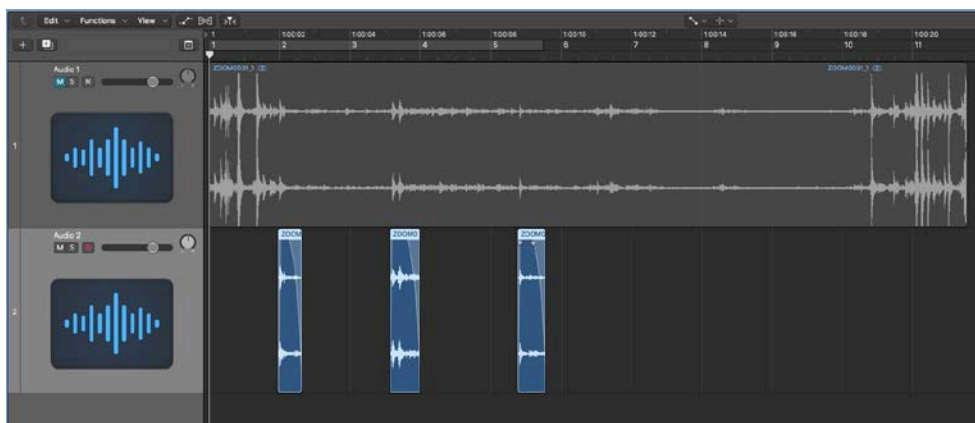
1. Tuo lähdemateriaali editointiohjelmaan kuunneltavaksi yhdelle raidalle.



2. Kuuntele materiaali läpi. Leikkaa matkan varrella käytettävät otokset irti lähdemateriaalista omiksi klippeiksi omalle raidalleen.



3. Voit poistaa lähdemateriaalin projektista, kun olet varma, että kaikki käytettävät klipit on irrotettu. Jos haluat pitää sen projektissa, voit esimerkiksi mykistää kyseisen raidan.



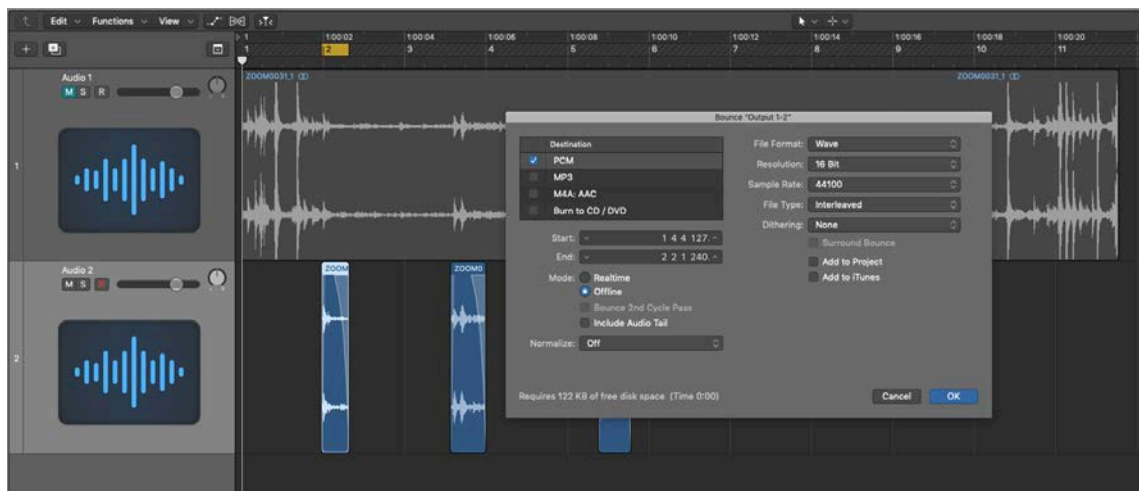
4. Siivoa valitsemasi klipit

- a. Leikkaa hiljaisuus klipin päistä pois, varsinkin jos ääni laukaistaan koodissa ja sen täytyy kuulua heti. Älä silti leikkaa liian tiukasti etenkin loppupäästä.
- b. Tee klipin alkuun fade in ja loppuun fade out –efektit. Tähän voi tarvita pienen pätkän hiljaisuutta, jotta fadet eivät vaimenna varsinaista ääntä. Jos heti klipin alussa tai aivan lopussa on voimakas signaali, se saattaa napsahdella tai poksahdella toistettaessa. Tämän välttämiseksi signaalin voimakkuus klipin päissä pitää olla nolla. Yleensä näin on, kun klipin aaltomuotoa esittävä viiva on klipin keskikohdassa näkyvän vaakaviivan tasalla. Fadella pakotetaan signaali nolleen.



- c. Siivoamiseen saattaa kuulua muitakin vaihteita, kuten häiritsevän taajuuden poistaminen ekvalisaattorilla tai häiriöäänen pois leikkaaminen klipin keskivaiheilta.

5. Taiteile! Tee tarpeen ja osaamisesi mukaan siivottuista äänistä alussa tekemääsi suunnitelmaa tukevia. Muuta ohjelmasta löytyvillä efekteillä äänen väriä, dynamiikkaa, korkeutta tms. Google ja YouTube kertovat lisää eri ohjelmista ja plugineista.
6. Vie klipit. Tallenna kukin siivottu klippi omaksi tiedostokseen. Unitylle tiedostomuodon täytyy olla 16-bittinen WAV. Näytteenottotaajuudeksi kannattaa laittaa CD-laatu eli 44,1 kHz. Tiedostot voi tallentaa suoraan Unity-projektin Assets-kansioon. Unity suorittaa uuden tiedoston havaittuaan automaattisesti oman tuontiprosessinsa. Unityn äänen tuontiasetukset käydään läpi tuonnempana.



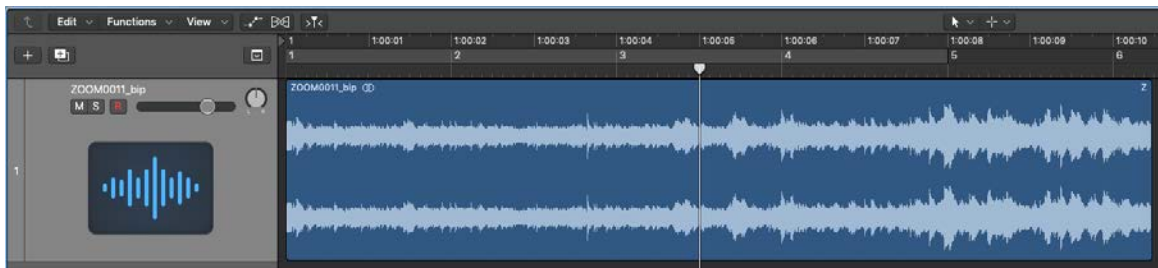
## Editointiohjelmiä

- [Lista maksuttomista DAW- ja äänieditointiohjelmissä](#)
- [Maksullisia DAW-ohjelmia vertailussa](#)

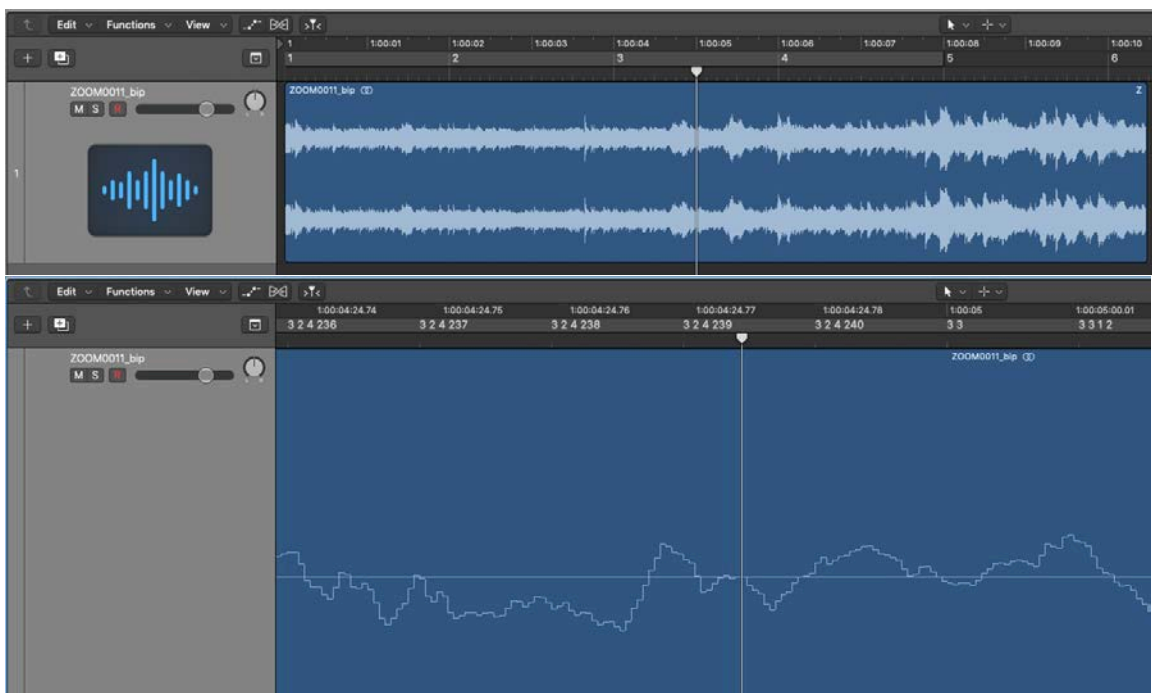
## Saumaton luuppi editoimalla

Joskus ääni vaatii luonnolliselta tuntuaakseen, että käyttäjä ei havaitse sen alkavan tai loppuvan. Tällöin siitä kannattaa tehdä saumaton luuppi. Periaate on yksinkertainen: tehdään äänen keskikohtasta sen alku ja loppu, ja ristiin häivyttämällä vanhoista lopusta ja alusta saadaan uusi keskikohta.

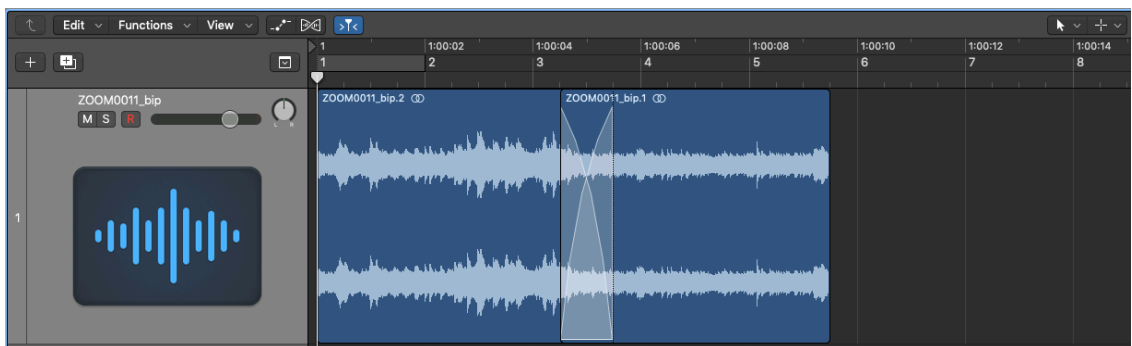
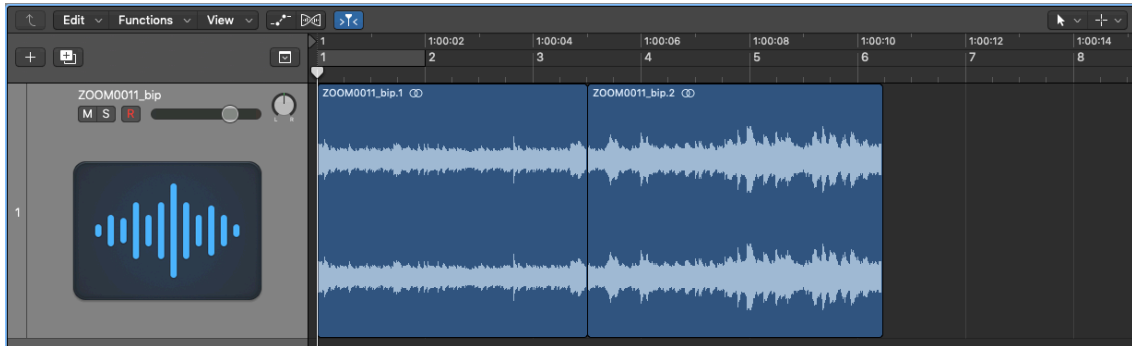
1. Valitse riittävän pitkä klippi. Pituus vaihtelee käyttötarkoituksesta riippuen. Yleensä liian lyhyt klippi paljastaa olevansa toistuva luuppi ja alkaa ärsyttää kuulijaa. 30 sekuntia on siedettävä, 2 minuuttia jo melko hyvä pituus. Muista että pidempi klippi tekee sovelluksesi tiedostokoosta suuremman.



2. Leikkaa valitsemasi klippi keskeltä poikki. Leikkaa täsmälleen sellaisesta kohdasta, jossa signaalin voimakkuus on nolla välttääksesi pokahtelua toistettaessa.



3. Vaihda klipin puolikkaiden järjestys: loppuosa alkuun ja alkuosa loppuun.
4. Vie puolikkaat niihin lähelle toisiaan, että ne risteävät. Monet ääniohjelmat tekevät risteyskohtaan automaattisesti ristiinhäivytyksen eli crossfaden. Muussa tapauksessa lisää päihin fadet itse.



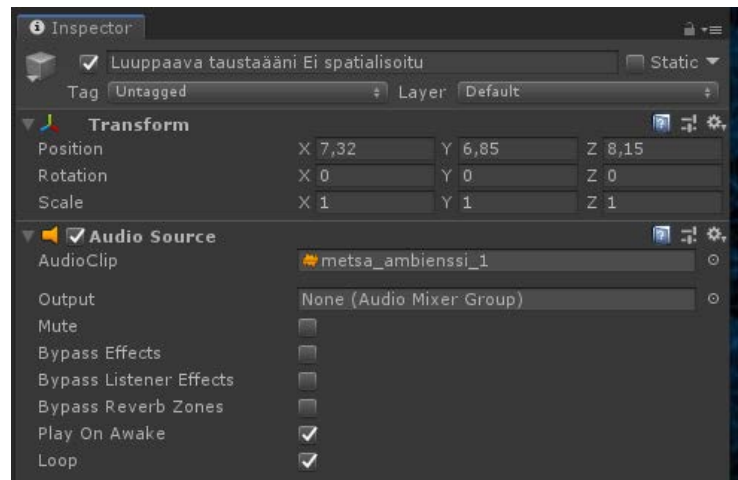
Tarkista että klippi ei kuulostaa omituiselta ristiinhäivytyksen kohdalta ja luupattessaan lopusta alkuun. Jos häivytyiskohta kuulostaa huonolta, säädä häivytyksen pituutta tai sijaintia. Jos luupaus lopusta alkuun napsahtaa, kokeile tehdä päihin mahdollisimman lyhyet häivytykset. Jos se ei korjaa ongelmaa, tee koko luuppi uudelleen ja leikkaa luuppauskohta eri kohdasta. Signaalin on oltava nollassa leikkauskohdasta.

5. Lopuksi vie näin syntynyt klippi uudeksi tiedostoksi Unity-projektiin.



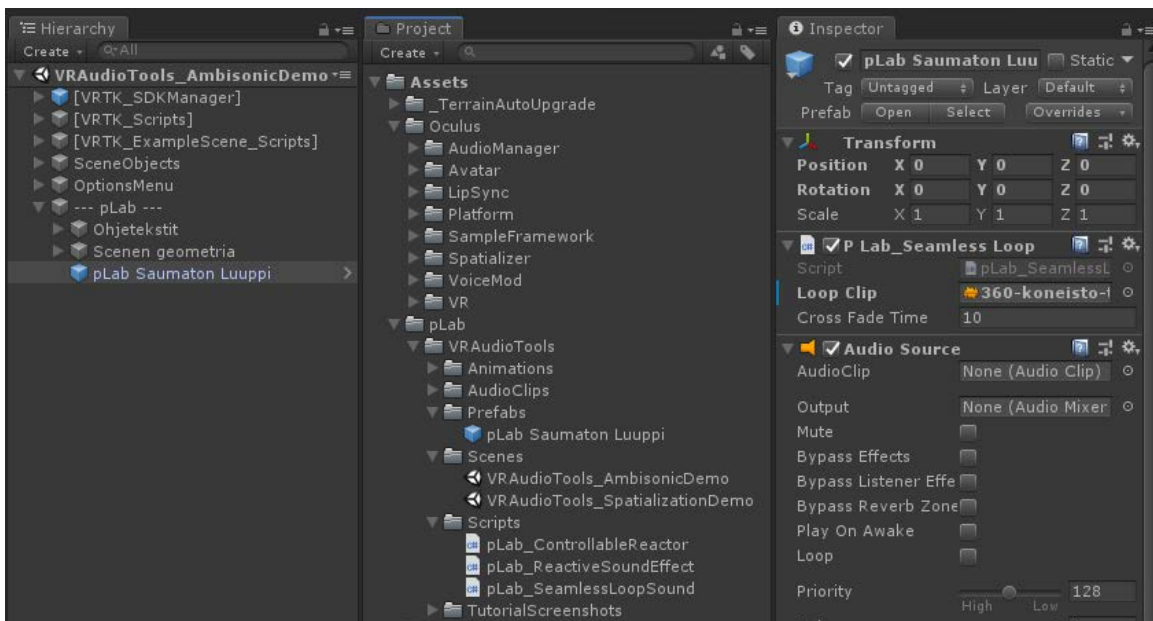
Unityssa editoimalla saumattomaksi tehdyn luupin käyttö on helppoa:

1. Tuo klippi Unityyn raahaamalla äänitiedosto Unityn Project-näkymään, haluamaasi kansioon.
2. Avaa haluamasi scene, luo siihen uusi GameObject, ja lisää siihen AudioSource-komponentti.
3. Raahaa tekemäsi klippi AudioSourcen Clip-kenttään.
4. Aseta AudioSourcen Play On Awake ja Loop –valinnat päälle.



## Saumaton luuppi koodilla

Ambisonic-/360-formaatin luoppaaminen editoimalla on paljon mutkikkaampi prosessi tavalliseen mono- tai stereoklippiin verrattuna. Ambisonic-tiedoston eri raidat on tuotava erillisinä tiedostoina omille raidoilleen, muutettava yhtä pitkiksi luepeiksi edellä kuvatulla tavalla, ja tallennettava takaisin ambisonic-tiedostoon. Prosessissa täytyy pitää huoli, että raidat päätyvät ambisonic-tiedostossa samaan järjestykseen missä ne olivatkin. Eri ohjelmien tapa käsitellä ambisonic-tiedostoja tekee yleispätevän ohjeen laatimisesta mahdotonta. Ambisonic-tiedostojen saumaton luoppaus editoimalla on sen verran mutkikasta, että se kannattaa tehdä mieluummin koodilla Unityssa. pLab on tehnyt tätä varten koodin, joka toimii ambisonic-äänten lisäksi myös tavallisen stereoäänen luoppaukseen.



pLabVRAudioTools -projektissa on valmiit tiedostot pLabSeamlessLoopSound ja prefab pLab Saumaton Luuppi, sekä VRAudioTools\_AmbisonicDemo –scene, jossa ne ovat toiminnassa.

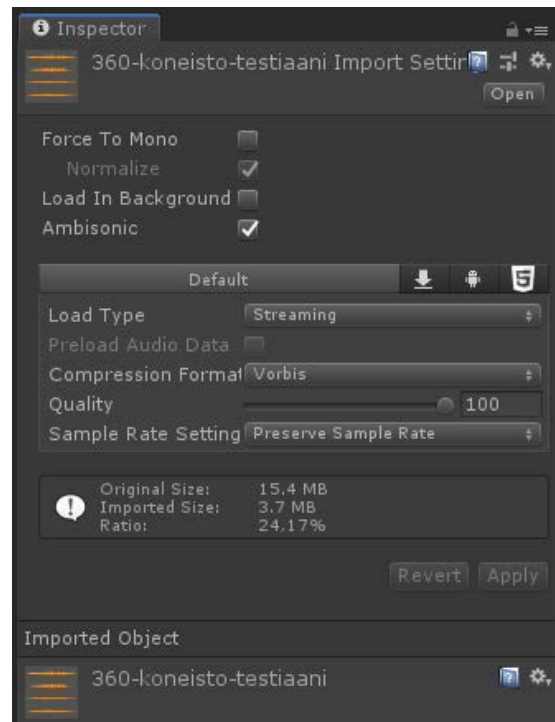
## Pitkien tiedostojen tuontiasetukset

Määrittele jokaiselle käyttämällesi klipille oikeat tuontiasetukset Unityssa. Äänitiedostojen Import settings – näkymä näkyy Unityn Inspectorissa kun valitset äänitiedoston klikkaamalla. Jos klippi on erityisen pitkä (yli 30 sekuntia), aseta sen Load Typeksi Streaming. Muuten kannattaa käyttää valintaa Compressed in Memory.

## Ambisonic-tiedostojen tuontiasetukset

Kun tuot ambisonic-/360-äänitiedostoja Unityyn, valitse niiden tuontiasetuksista Ambisonic-kohta aktiiviseksi. Tällaista klippiä toistavan AudioSource:n Spatialize-valinta pitää olla pois päältä. Ambisonic-toistoa varten myös projektin Project Settings – Audio –näkyssä täytyy olla valittuna jokin Ambisonic Decoder Plugin. Kun projektissa muutenkin käytetään Oculuksen lisäosaa, dekooderiksi valitaan luonnollisesti OculusSpatializer.

Unity tukee ”first order ambisonics” –muotoa, tai toiselta nimitykseltään B-formaattia. Jos käyttämäsi äänitiedoston muoto on jotain muuta, 360-vaikutelma ei todennäköisesti toistu kuten pitäisi.



## 3D-tilaan sijoitettu ambisonic-ääni Unityssa

Ambisonic-formaatti on erinomainen vaihtoehto silloin, kun käyttäjälle halutaan antaa 360-kokemus, jossa pysytään yhdessä pisteessä. Unitylla ambisonic-ääntä voi kuitenkin sijoittaa myös 3D-maailmoihin, joissa voi liikkua.

Ambisonic-ääni toimii Unityssa samoin kuin mikä tahansa ääni tullessaan toistetuksi AudioSourcessa. Samat säädöt vaikuttavat samalla tavalla normaaliin stereo- tai monoääneen ja ambisonic-formaattiin.

Jos asetat ambisonic-ääntä toistavan AudioSource:n Spatial Blend –säätimen arvoon 0, sen tilallinen miksaus on täysin 2D. Tämä ilmenee niin, että kyseinen ääni seuraa käyttäjää kaikkialle, mutta sen stereokuva muuttuu sen mukaan, miten käyttäjä kääntää päätään. Jos säätöarvo on 1, eli täysi 3D, käyttäjä kuulee äänen vain tietyllä alueella AudioSource:n 3D Sound Settings -säätöjen mukaisesti, ja stereokuva muuttuu edelleen sen mukaan, miten käyttäjä kääntää päätään.

AudioSource:n isäntäobjektia voi pyörittää 3D-avaruudessa objektin Transform-komponentin Rotation-säädöillä. Tällä voi vaikuttaa ambisonic-äänen suuntaukseen, riippumatta siitä, onko sitä toistava AudioSource 2D vai 3D.

## VR-projektin äänten toteutus Unityssa vaihe vaiheelta

Miten sitten toteutetaan kokonainen VR-projekti, jossa on toimiva tilallinen ääni? Unityssa se on varsin suoraviivaista. Seuraavissa ohjeissa ei oteta kantaa mihinkään lisäosiin tai materiaaleihin, mitä projektisi muut ominaisuudet äänen lisäksi mahdollisesti tarvitsevat toimiakseen. Näillä ohjeilla saat Unitysta ääntä ulos niin, että se toimii oikein Oculus Riftillä.

pLabin laatimissa esimerkkiprojekteissa VR-toiminnallisuudet kuten liikkuminen ja interaktiot ympäristön kanssa on toteutettu [VRTK](#)-lisäosalla, jonka saa Unity Asset Storesta.

pLabin esimerkkiprojekteja voi käyttää omien projektien pohjana sellaisenaan.

## Oculuksen spatialisointilisäosan asentaminen Unityyn

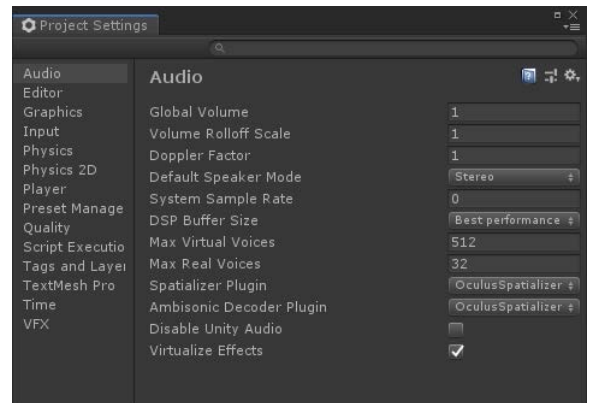
Jotta Unity-projektisi toimii Oculus Riftillä, tarvitset siihen [Oculuksen integraatiopakettin](#). Asset store –paketteja on helpoin lisätä projektiin suoraan Unitysta:

1. Avaa Unitylla projekti, johon haluat lisätä Oculus-integraation.
2. Kun projekti on auki, avaa Asset Store –ikkuna Window valikosta (tai Ctrl+9).
3. Etsi hakukentän avulla ”oculus integration”.
4. Klikkaa listaan ilmestyvää Oculus Integration –otsikkoa, jolloin paketin sivu aukeaa.
5. Klikkaa pinkkiä Download-painiketta.
6. Latauksen jälkeen painikkeessa lukee Import. Klikkaa sitä.
7. Ruutuun aukeaa lista paketin tiedostoista, ja listan lopussa uusi Import-painike. Klikkaa sitä. Jos tässä ikkunassa lukee ”Nothing to import!”, projektissa on jo kyseinen paketti, eikä sitä tarvitse tuoda uudelleen. Paketin tiedostopolku on Assets-Oculus-Spatializer.

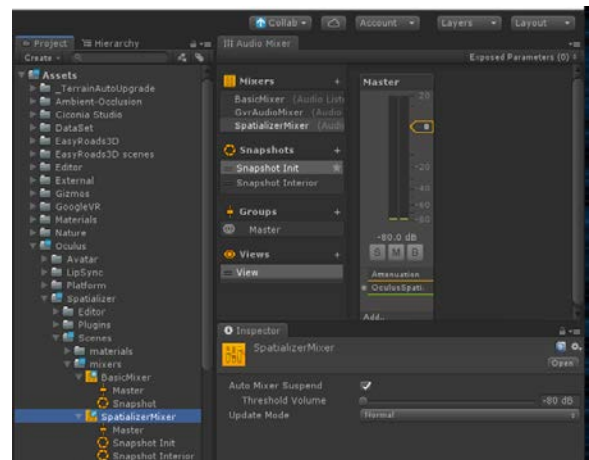


## Lisäosan käyttöönotto

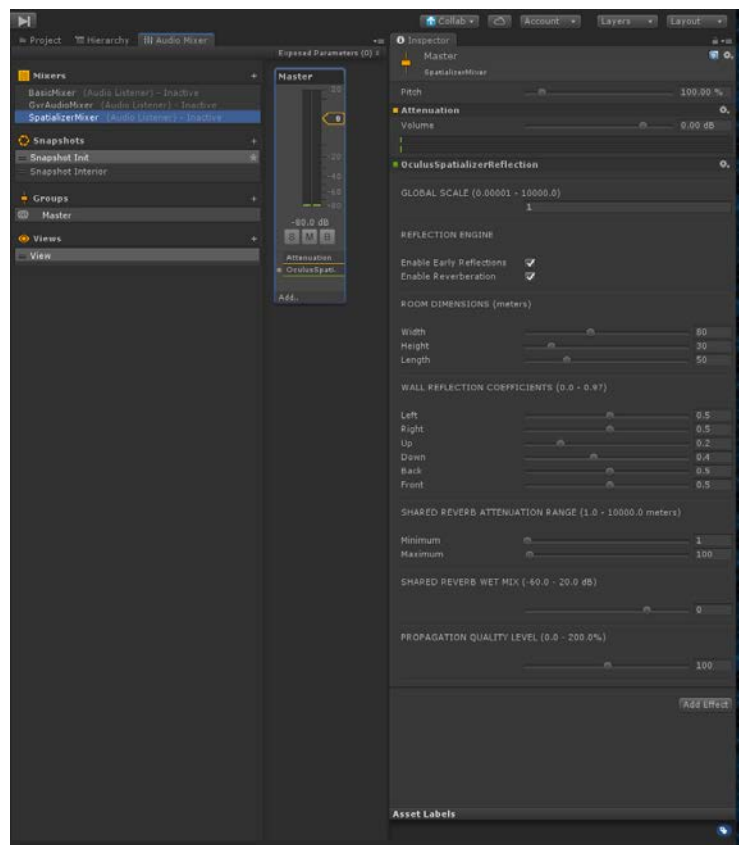
1. Avaa Unityssa File-Project settings –valikko.
2. Audio-kohdassa valitse Spatializer Pluginiksi ja Ambisonic Decoder Pluginiksi OculusSpatializer.



Oculusen lisäosan mukana tulee SpatializerMixer –niminen AudioMixer-tiedosto. (Project-näkymässä, kansiossa Assets-Oculus-Spatializer-Scenes-mixers) Tuplaklikkaamalla tiedostoa Unityyn aukeaa AudioMixer-ikkuna/välilehti. Sen Mixers-otsikon alla näkyy edellä mainittu SpatializerMixer –mikseri. Valitse se klikkaamalla.



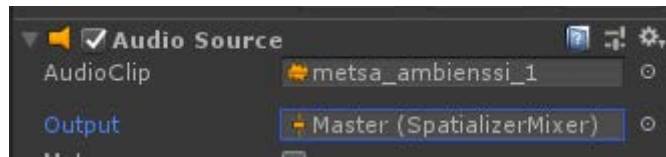
Kun valitset mikserin, AudioMixer-ikkunaan tulee näkyviin mikserin kanavat (AudioMixerGroupController). Tässä mikserissä on vain yksi kanava. Enempää ei tarvitakaan. Kanavan alaosa näkee, että kanavassa on valmiina efekti nimeltä "OculusSpatializerReflection". Tämä efekti vastaa siitä, että OculusSpatializer –pluginin tekemät laskelmat eri äänilähteiden kaudiista ym. todella toistetaan pelaajan korvissa Unityn normaalisti toistaman äänen sijasta. Jatkossa kun projektiin lisätään äänilähteitä, jotka halutaan spatialisoida, niistä lähtevä signaali reititetään aina tähän mikserikanavaan. Tämä tehdään asettamalla kunkin spatialisoitavan AudioSource:n Output –kenttään kyseinen mikserikanava. Efektin säätöjä voi muuttaa tarpeen mukaan, mutta yleensä niihin ei tarvitse koskea, jos äänilähteet ja kentän akustinen geometria on määritelty oikein.





Jos kanavassa ei ole kyseistä efektiä voit lisätä sen Add.. -painikkeesta kanavan alaosaan. Jos taas koko projektista ei löydy yhtään mikseriä, voit luoda sellaisen Unityn valikosta kohdasta Assets>Create-AudioMixer. Lisää uuden mikserin ainoaan kanavaan edellä mainittu efekti.

Jatkossa aseta luomiesi AudioSourcejen Output-kenttään kyseinen mikserikanava, jos haluat, että niiden ääni spatialisoidaan.



## Äänilähteiden lisääminen ja spatialisointi

Ääntä spatialisoivat lisäosat perustuvat Unityn oman 3D-äänijärjestelmän päälle rakennettuihin ominaisuuksiin. Unityn äänijärjestelmän kivijalka koostuu kahdesta erilaisesta komponentista, äänen vastaanottimesta (AudioListener) ja äänilähteestä (AudioSource).

### Äänen vastaanotin – Audio Listener

VR-sovelluksissa äänen vastaanotin simuloi käyttäjän korvia. Siksi se on yleensä järkevintä kiinnittää siihen objektiin (GameObject), joka esittää pelaajan päätä. Yleensä samassa objektissa on kiinni myös pelaajan silmiä vastaava kamera. Oletuksena Unityn editori lisääkin AudioListenerin automaattisesti aina samaan objektiin, johon lisätään kamera.

Kun käytetään Oculuksen spatialisointia, Unityn tavallinen AudioListener ei vaadi kaverikseen muita ääneen liittyviä komponentteja. Esim. Steam audiota käytettäessä samaan GameObjectiin pitäisi lisätä myös Steam Audio Listener –komponentti.

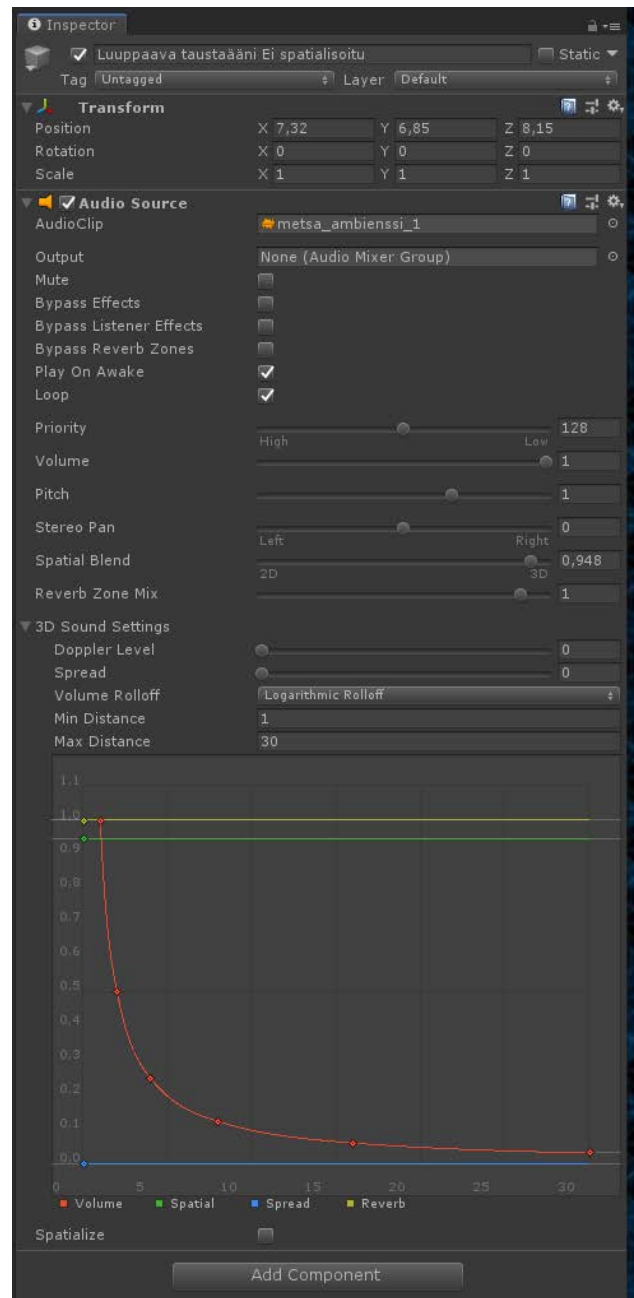


## Äänilähde – Audio Source

Unityssa AudioSource-komponentti hoitaa äänen varsinaisen toistamisen. Sen ikoni on keltainen kaiutin. Kun komponentille kerrotaan, mitä ääntä sen tulee toistaa (AudioClip), ja kuinka kauas äänen pitää kuulua (Spatial Blend ja 3D Sound Settings), ollaan jo pitkällä.

Äänilähteen soittamiseen käskystä tietyllä hetkellä tarvitaan jonkin verran ohjelmointia, mutta tilan, ympäristön ja läsnäolon tuntua luovat ambienssiäänet eivät vaadi sitä. AudioSourcen äänen voi asettaa luuppaamaan, laittaa sen toistumaan heti kun sovellus käynnistyy. Tämä neuvotaan luuppien tekemistä käsittelevässä kohdassa.

Usein Unityn tavallisten AudioSource-komponenttien käyttö ilman spatialisointiakin riittää vaikuttavien äänimaisemien luomiseen, mutta varsinkin VR-sovelluksissa niiden melko yksinkertainen tapa simuloida äänen suuntaa ja kaikuja on joskus parempi korvata spatialisoidulla äänellä. Tätäkään ei tosin tarvitse tehdä kaikelle äänelle projektissa, eikä usein kannatakaan. Äänen spatialisointi voi viedä paljon prosessointitehoa, joten siinä kannattaa keskittyä käyttäjän kannalta vain oleellisimpien äänten tarkkaan mallinnukseen.



## Äänilähteen spatialisointi Oculus Audio SDK:lla

1. Lisää Unityssä avoimeen sceneen uusi GameObject –objekti.
2. Lisää objektiin AudioSource-komponentti.
3. Määrittele AudioSourcen ominaisuudet: klippi, output-kanava, 3D settings, spatialisointi, jne.
4. Lisää objektiin ONSPAAudioSource –komponentti, ja aseta sille halutut säädöt

a. **Spatialization Enabled:** kun päällä, kyseinen ääni miksataan HRTF-mallilla käyttäjän korviin

b. **Reflections Enabled:** kun päällä, kyseisen äänen miksauksessa huomioidaan mahdollinen akustinen geometria (ks. alempana)

c. **Gain:** Oculuksen laskenta tekee äänistä tavallista hiljaisempia. Nosta tätä arvoa lisätäkseen äänenvoimakkuutta.

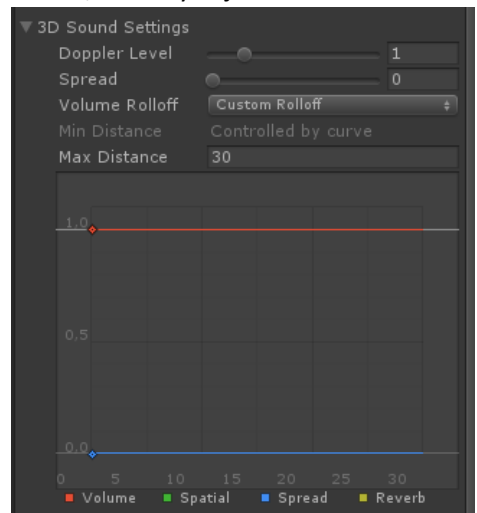
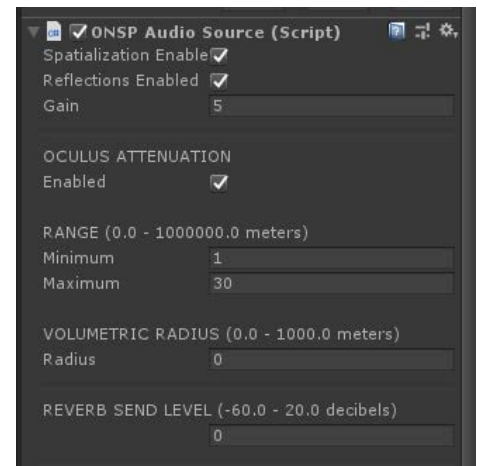
d. **Oculus attenuation:** Unityn tavallisten AudioSourcen 3D Settings –säätöjen ohella toimivat asetukset sille, miten äänen voimakkuus muuttuu, kun käyttäjä liikkuu sitä kohti tai siitä pois. Jos käytät äänilähteessä Oculuksen attenuaatiota, on suositeltavaa ”nollata” siitä Unityn oma 3D Settings – Volume Rolloff –asetus. Tee tämä poistamalla punaisesta käyrästä kaikki pisteet niin, että se muuttuu suoraksi viivaksi, joka alkaa arvosta 1 (kuva oikealla).

i. Minimi- ja maksimietäisyydet toimivat kuten Unityn omissa säädöissä. Minimissä ääni kuuluu täysillä, maksimin ylitettäessä se häviää kuuluvista.

ii. Volumetric radius –säädöllä äänen lähteen voi määritellä suuremmaksi kuin tavallinen yksi piste 3D-avaruudessa. Näin ääni voi esim. kuulua osittain akustiseksi esteeksi määritellyn objektin takaa, vaikka sen lähteeseen ei ole suoraa linjaa kuulijasta.

e. Reverb send level: tällä määrittelet, kuinka paljon aiemmin käsitelty SpatializerMixerin kanavassa oleva OculusSpatializerReflection –efekti kaiuttaa kyseistä ääntä.

5. Lisää mahdolliset muut komponentit, joilla ohjaat tämän GameObjectin ääniä. Esimerkiksi saumattomasti luoppaavaa ambisonic-ääni kannattaa tehdä koodilla editoinnin sijaan. pLab on tehnyt pLabVRAudioTools-projektiin tätä varten käyttövalmiin prefabin ”pLab Saumaton Luuppi”.



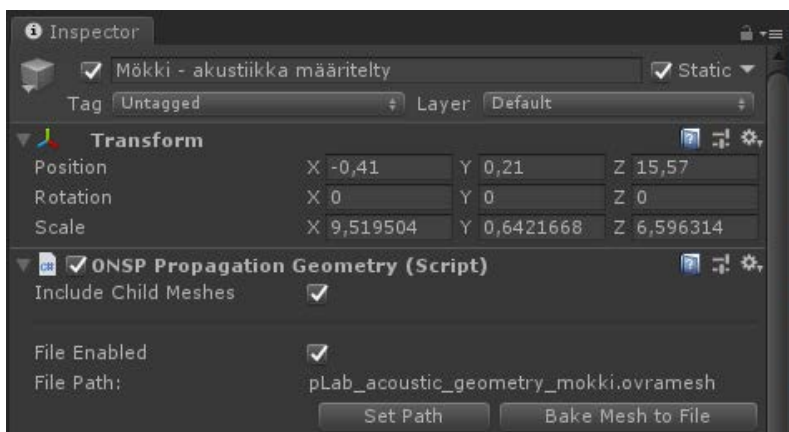
## Akustisen geometrian lisääminen Oculus Audio SDK:lla

Unity-sovelluksessa, jossa on 3D-geometriaa, Oculuksen äänilisäosalla on helppo antaa geometrialle akustisia ominaisuuksia, jotka otetaan huomioon spatialisoitujen äänten miksauksessa. Tämä tapahtuu lisäämällä haluttuihin 3D-objekteihin kaksi komponenttia, jotka löytyvät projektista Assets-Oculus-Spatializer-Scripts –kansioista:

- ONSPPropagationGeometry
- ONSPPropagationMaterial

Komponentit voi lisätä objektille raahaamalla kyseiset tiedostot objektin Inspector-näkymässä, tai klikkaamalla objektin Inspector-näkymässä Add Component –painiketta ja etsimällä oikean tiedoston.

**ONSPPropagationGeometry**-komponentti tavallaan tallentaa objektin Oculuksen äänisysteemin ymmärtämään muotoon, jotta sen 3D-geometria osataan huomioida, kun lasketaan, miten spatialisoitavat äänet kaikuivat tilassa tai miten ne vaimenevat, kun objekti estää äänen suoraan etenemisen äänilähteestä käyttäjän kohdalle. Nämä asiat lasketaan vai sellaisille äänilähteille, joissa on ONSPAAudioSource-komponentti. Komponentin säädöistä riippuu, mitä kaikkea kyseisen lähteen äänistä lasketaan (ks. edellä).



Jos haluat antaa akustiikkaominaisuuksia staattisille objekteille (Inspectorin oikean ylänurkan Static-valinta on aktiivinen), niiden geometria täytyy tallentaa tiedostoon. Tätä varten aseta File Enabled –valinta aktiiviseksi, anna tiedostonimi Set Path –painikkeesta, ja lopuksi tallenna geometria tiedostoon klikkaamalla Bake Mesh to File. Jos myöhemmin liikutat tällaisia objekteja Unityn Scene-näkymässä, tämä prosessi täytyy toistaa, jotta objektien akustiikka päivittyy ajan tasalle.

Ei-staattisten objektien akustiikkageometria lasketaan lennosta, kun sovellus ajetaan. Ei-staattisten objektien liikkeessa niiden akustiset ominaisuudet päivitetään automaattisesti ajon aikana.

**ONSPPropagationMaterial**-komponentti kertoo Oculuksen äänijärjestelmälle, miten kyseinen 3D-objekti vaikuttaa äänen eri taajuuksien hajoamiseen, vaimentumiseen ja eteenpäin välittymiseen. Lähtökohtana on kattava lista erilaisia fyysisiä materiaaleja, kuten puuta, lasia tai kasvillisuutta, mallintavia asetuksia. Näitä voi lisäksi hienosäätää itse.

Nämä komponentit voi lisätä yksittäisille objekteille, tai useille kerrallaan hyödyntämällä hierarkiaa. Voit antaa ONSPPropagationGeometry- ja -Material -komponentit sellaiselle objektille, joka sisältää Unityn Hierarchy-näkymässä muita objekteja. Tällöin akustinen geometria luodaan koko objektien joukolle kerralla, ja asettamasi materiaali koskee kaikkia objekteja. Voit ohittaa tämän yhteisen materiaaliasetuksen antamalla haluamallasi objektille oman ONSPPropagationMaterial-komponentin.

