

Arttu Aalto

Adaptiivisen musiikin perusteet mobiilipeleissä

Metropolia Ammattikorkeakoulu

Muusikko (AMK)

Musiikin tutkinto

Opinnäytetyö

18.11.2015

Tekijä(t) Otsikko Sivumäärä Aika	Arttu Aalto Adaptiivisen musiikin perusteet mobiilipeleissä 37 sivua 18.11.2015
Tutkinto	Muusikko (AMK)
Tutkinto-ohjelma	Musiikin tutkinto
Suuntautumisvaihtoehto	Tuottaja-teknologi
Ohjaaja(t)	Lehtori Jukka Väisänen, Lehtori Esa Onttonen
<p>Tässä opinnäytetyössä tutkitaan adaptiivisen mobiilipelimusiikin erityispiirteitä ja lainalaisuuksia sekä sen tuottamiseen liittyviä tyypillisiä työtapoja ja tekniikoita. Työn tarkoitus on antaa musiikintekijälle hyvä pohja toimia mobiilipelialalla, joka on viime vuosi- na kasvattanut suosiotaan nopeasti ja tarjoaa jatkuvasti enemmän työmahdollisuuksia myös musiikin parissa työskenteleville. Työ pohjaa kirjallisiin lähteisiin sekä kirjoittajan omaan kokemukseen.</p> <p>Ensin työssä määritellään ominaisuudet, jotka erottavat videopelimusiikin perinteisemmistä musiikkimuodoista, esimerkiksi äänilevystä. Nämä ovat epälineaarisuus ja adaptiivisuus, jotka määrittävät vahvasti videopelejä ylipäänsä. Niiden vaikutus musiikin tuottamiseen on työn keskiössä.</p> <p>Seuraavaksi esitellään lyhyesti videopelimusiikin historiaa, joka on ennen kaikkea tarina teknisten rajoitteiden voittamisesta ja jatkuvasta muutoksesta. Mobiilialustojen myötä historia myös ikään kuin toistaa itseään rajoitteiden suhteen. Haasteita aiheuttaa muun muassa rajallinen suorituskyky, taajuusvaste ja stereokuva. Työ käy läpi näitä rajoitteita ja esittää niiden parissa työskentelyyn toimintatapoja ja ajatusmalleja. Lisäksi tutkitaan hie- man mobiilipelien liiketoimintamalleja ja esitellään nykyaikaisen pelinkehityksen ydin, peli- moottori.</p> <p>Työssä tutkitaan vielä pelialalla äänen kanssa työskentelyn perustyökaluja, ääniväliohjel- mistoja, jotka ovat monella tapaa tehneet pelimusiikin tuottamisen helpommaksi lähestyä myös ilman ohjelmointiosaamista. Lopuksi esitellään yleisimpiä tekniikoita adaptiivisen musiikin tuottamiseksi ja eritellään niiden etuja ja ongelmia. Näitä tekniikoita ovat muun muassa horisontaalinen resekvensointi ja vertikaalinen reorkestraatio. Pohdintaosuudessa arvioidaan vielä alan tulevaisuudennäkymiä ja koulutusta Suomessa.</p>	
Avainsanat	pelimusiikki, ääniväliohjelmistot, videopelit, mobiilipelit

Author Title	Arttu Aalto Basics of Adaptive Music in Mobile Games
Number of Pages Date	37 pages 18 November 2015
Degree	Bachelor of Music
Degree Programme	Music
Specialisation Option	Music Production and Engineering
Supervisors	Jukka Väisänen, MMus, Esa Onttonen, MMus
<p>This thesis studies the characteristics, principles and workflows related to adaptive music in mobile games. The main purpose is to provide a good basis for music makers to work in the mobile game industry. Due to its fast growth during recent years, the industry provides more working opportunities also for music makers. The study is based on literature and the experiences of the author.</p> <p>The thesis first defines that the main differences between video game music and more traditional music forms such as a record are nonlinearity and adaptivity. The study mainly focuses on these two qualities that define video games in general and their impact on producing music.</p> <p>The next part briefly introduces the history of video game music, which comprises effort to overcome technical limitations and perpetual change. Along with the popularity of mobile platforms, the history seems to repeat itself in terms of limitations. The challenges include limited performance, frequency response and stereo image. The thesis outlines these limitations and provides paradigms and procedures for dealing with them. In addition, the business models of mobile games are explained and the study provides an overview of the core of modern game development, the game engine.</p> <p>Furthermore, the thesis studies audio middleware, the basic tools of game audio that have made game music more approachable also without programming knowledge. Finally, the thesis examines most common adaptive music creation techniques and analyzes their pros and cons. These techniques include horizontal resequencing and vertical reorchestration. In conclusion, the thesis speculates on the future of video game music and evaluates the relevant education in Finland.</p>	
Keywords	game music, video games, mobile games, audio middleware

Sisällys

1	Johdanto	1
2	Videopelimusiikin erityispiirteet	2
2.1	Epälineaarisuus	2
2.2	Adaptiivisuus	3
3	Videopelimusiikin historia lyhyesti	5
3.1	Ohjelmoitavat äänigeneraattorit	5
3.2	Tekniikan kehitys	7
3.3	Nykyinen tilanne	7
4	Mobiilialusta	8
4.1	Suosituimmat mobiililaitteet	8
4.2	Rajallinen suorituskyky	10
4.3	Äänen pakkaaminen	11
4.4	Taajuusvaste ja stereokuva	12
4.5	Free-to-play vs. Premium	13
4.6	Pelaako kukaan edes äänet päällä?	15
5	Pelimoottorit	15
6	Ääniväliohjelmistot	17
6.1	Äänisuunnittelijasta implementoijaksi	17
6.2	Toimintaperiaate	18
6.2.1	Tapahtumat ja parametrit	18
6.2.2	Miksaus ja efektit	19
6.3	FMOD Studio	20
6.4	Muita vaihtoehtoja	21
7	Tekniikoita adaptiivisuuteen	23
7.1	Horisontaalinen resekvensointi	23
7.1.1	Siirtymät liukuen tai tempossa	24
7.1.2	Fraasit	25
7.1.3	Stingerit	26
7.2	Vertikaalinen reorkestraatio	27
7.2.1	Työskentely stemmojen kanssa	27

7.2.2	Additiivisuus ja vaihtoehtoisuus	28
7.3	Sattumanvaraisuus	29
7.4	Muita tekniikoita	30
7.4.1	Generatiivinen musiikki	30
8	Pohdinta	31
	Lähteet	35

1 Johdanto

Video- ja tietokonepelejä pidettiin ennen vain lasten juttuna. Nykypäivänä nuo lapset ovat aikuisia ja pelaavat yhä, ja niin pelaavat heidän lapsensakin. Pelaamisesta on tullut tärkeä vapaa-ajan aktiviteetti ja siten myös iso osa viihdealaa. (Moormann 2013, 191.)

Tutkin opinnäytetyössäni adaptiiviseen (ks. luku 2.2) videopelimusiikkiin liittyviä käsitteitä ja työtapoja. Esittelen tyypillisiä keinoja rakentaa adaptiivisuutta ja käyn läpi niiden hyviä ja huonoja puolia. Pohjustukseksi esittelen videopelimusiikin historiaa, pelimoottoreita ja pelialalla äänisuunnitteluun useasti käytettyjä ääniväliohjelmistoja, jotka ovat monella tapaa tehneet pelimusiikista helpommin lähestyttävän työmaan kenelle tahansa omaa musiikkia tuottavalle.

Työni on tarkoitus antaa hyvä pohja videopelimusiikin tekemiseen sen erityispiirteisiin aikaisemmin perehtymättömälle lukijalle. Pyrin tiivistämään tarvittavan tietomäärän pelinkehityksestä ja pelimusiikin tuottamisesta mahdollisimman kompaktiin pakettiin. Lähestyn aihetta ensisijaisesti tekniseltä kannalta, enkä käsittele juurikaan adaptiivisuuteen liittyviä taiteellisia pulmia. Teknisen puolen mahdollisuuksien ja rajoitusten ymmärtäminen on mielestäni ensiarvoisen tärkeää, jotta adaptiivisen musiikin onnistunut suunnitteleminen on ylipäänsä mahdollista. Toisaalta tekninen puoli on useasti säveltäjien mielestä epäkiinnostavaa tai vaikeaselkoista, ja siksi yritän käsitellä vain sen kaikista tärkeimpiä ja olennaisimpia osa-alueita sekä käytän paljon esimerkkejä.

Keskityn erityisesti mobiilipeleihin, koska ne ovat nopeasti kasvattaneet suosiotaan ja ovat tämän työn kirjoittamisen aikaan ohittamassa liikevaihdollaan jo konsolipelit (Stuart, 2015). Mobiilipelialaa voi siis pitää tulevaisuuden alana ja työllistäjänä – etenkin Suomessa, joka on menestynyt poikkeuksellisen hyvin mobiilipelialalla viime vuosina (Piispanen, Vihikainen 2014, 17). Toisaalta mobiilipelit asettavat myös tiettyjä rajoituksia äänisuunnittelulle, ja pidän tärkeänä käydä niitäkin läpi. Suurinta osaa työstä voidaan kuitenkin soveltaa mille tahansa alustalle¹ tarkoitettun pelimusiikin

¹ Engl. *platform* eli alusta tarkoittaa laitetyyppiä tai -mallia, jolla sovellusta käytetään eli tässä yhteydessä peliä pelataan.

suunnitteluun, koska olen keskittänyt mobiilialustoihin liittyvän pohdinnan pääosin omaan lukuunsa ja toisaalta koska mobiilipeleihin liittyvien teknisten rajoitusten noudattaminen voi monesti olla hyödyksi muillekin alustoille pelimusiikkia suunniteltaessa.

Olen nauttinut videopeleistä ja niiden äänimaailmasta aina suuresti, minkä takia olen myös ollut kiinnostunut niiden äänisuunnittelusta ja musiikin tekemisestä niihin. Olen työskennellyt huhtikuusta 2015 lähtien Next Games -yhtiössä, joka kehittää mobiilipelejä. Työtehtäviini kuuluu äänisuunnittelu, musiikin tuottaminen ja ääniohjelmointi. Työssäni olen oppinut paljon mobiilipelien äänisuunnittelun ja pelialan lainalaisuuksista ja sovellan kokemuksiani myös tähän opinnäytetyöhön. Lisäksi käytän kirjallisia lähteitä, joita reflektoin oman kokemukseni pohjalta. Toivon oleellista tietoa muille tiivistäessäni syventäväni myös omaa osaamistani ja toisaalta löytäväni kirjallisuudesta vahvistuksia omille työelämässä muotoutuneille käsityksilleni.

Peli- ja musiikkialojen kansainvälisyyden takia useat käsitteet ovat vakiintuneet yleiseen käyttöön englanninkielisinä. Käytän työssäni näitä alalla vakiintuneita käsitteitä vakiintumattomien suomennosten sijaan. Näin termit ovat työssä samoja kuin käytännön työelämässä käytetyt, ja työ on helpommin luettavissa ja sovellettavissa.

2 Videopelimusiikin erityispiirteet

Videopelimusiikin tuottamisessa on paljon samaa kuin minkä tahansa musiikin tuottamisessa. Perustyökalut ja työvälitteet ovat samat: musiikki voi olla ohjelmisto-instrumenteilla tuotettua tai äänitettyä tai näiden yhdistelmä, ja säveltäminen on pohjimmiltaan samanlaista. Tietyt videopelimusiikin erityispiirteet kuitenkin asettavat poikkeuksellisia haasteita ja toisaalta avaavat uusia mahdollisuuksia. Tärkeimmät näistä piirteistä ovat epälineaarisuus ja adaptiivisuus.

2.1 Epälineaarisuus

Videopelit ovat luonteeltaan interaktiivisia ja epälineaarisia². Tässä suhteessa ne eroavat merkittävästi esimerkiksi elokuvista tai äänilevyistä. Epälineaarisuus ja sen aiheuttamat haasteet leimaavat kaikkea pelinkehitykseen liittyvää, esimerkiksi käsikirjoitusta,

² Epälineaarisuudella tarkoitan tässä yhteydessä määrittelemätöntä muotoa ja rakennetta.

animaatiota ja myös äänisuunnittelua sekä musiikkia. Keskeistä on, että pelaaja hallitsee pelitilannetta, joten hänen toimintansa tai toimettomuutensa määrittää tapahtumien kulun. Esimerkiksi toinen pelaaja saattaa käyttää muutaman minuutin jonkin pelin tarjoaman tehtävän ratkaisemiseksi, kun taas toisella samaan tehtävään kuluu puoli tuntia. Käytännössä jokainen pelikokemus on erilainen. (Collins 2008, 4.)

Ilmeisin ongelma, jonka epälineaarisuus aiheuttaa, on ajoitusten ennakoimisen mahdollisuus. Ei voida käyttää tietyn mittaista kappaletta tai ylipäänsä säveltää ja soittaa kuvan mukaan kuten elokuvamusiikkia yleensä lähestytään. Tämä ongelma ratkaistaan tyypillisesti tuottamalla musiikkia, joka soveltuu loopattavaksi³. Käytännössä suurin osa pelimusiikista perustuu jollain tavalla looppaamiseen.

2.2 Adaptiivisuus

Epälineaarisuuden aiheuttama ennalta-arvaamattomuus on perimmäisin syy myös adaptiivisuuden⁴ tarpeelle. Kuten elokuvissa tai vaikka teatteriesityksissä myös peleissä musiikin on tarkoitus vahvistaa tapahtumien aiheuttamia tunnereaktioita, viedä niitä päinvastaiseen suuntaan tai esimerkiksi ennakoida jotain tulevaa – musiikki on siis tiukasti sidoksissa tarinankerronnan tapahtumiin. Koska peleissä tapahtumien ajoitus ja järjestys on pelaajan käsissä, täytyy musiikin voida reagoida pelitapahtumiin **niiden tapahtuessa**. Tätä juuri tarkoitetaan adaptiivisella pelimusiikilla. Esimerkiksi pelaajan ohjaaman hahmon ollessa rakennuksen ulkopuolella soi rauhallinen, huomiota herättämätön musiikki. Hahmon avatessa rakennuksen oven sisältä paljastuu vihollisia ja musiikki muuttuu kiihkeärytmiseksi taistelumusiikiksi. Tyypillisesti tämänkaltaisia yksinkertaisia implementaatioita on peleissä paljon, mutta niiden lisäksi käytetään monesti myös monimutkaisempia järjestelmiä.

Kirjallisuudessa adaptiivisesta musiikista käytetään usein myös termejä interaktiivinen ja dynaaminen musiikki. Näitä pidetään toisinaan synonyymeina, mutta toisaalta jokaiselle termille on myös oma määritelmänsä. Collinsin (2008a, 4) mukaan *interaktiivisella* (kuvio 1) äänellä tarkoitetaan ääntä, joka reagoi suoraan pelaajan syötteeseen eli esimerkiksi painikkeen painamiseen. *Adaptiivinen* ääni taas reagoi pelitilanteisiin – siis ikään kuin välillisesti syötteeseen. *Dynaaminen* ääni on kattokäsite, jolla tarkoitetaan

³ Engl. *looping*, looppaus eli silmukointi tarkoittaa saman äänimateriaalin jatkuvasti kertaavaa toistamista.

sekä interaktiivista että adaptiivista ääntä. Pelimusiikki on useimmiten adaptiivista, mutta se voi olla myös interaktiivista. Esimerkiksi *Guitar Hero* ja *Singstar* -tyyppisissä, niin kutsutuissa musiikkipeleissä, musiikki on nimenomaan interaktiivista. (Young 2012, 9.) Tällaiset pelit ovat kuitenkin oma kategoriansa eivätkä tyypillisesti tarjoa pelisäveltäjälle työnsarkaa. Dynaamisuus taas on käsitteenä etenkin musiikin yhteydessä niin moniselitteinen, että mielestäni adaptiivisuus kuvaa pelimusiikin luonnetta parhaiten ja yksiselitteisimmin.



Kuvio 1. Dynaaminen ääni

Muun muassa Nintendon kuuluisan *Super Mario* -pelisarjan säveltäjä Koji Kondo erittelee neljä adaptiivisen musiikin ominaisuutta seuraavanlaisesti:

1. kyky luoda musiikkia, joka muuttuu jokaisella pelikerralla
2. moniulotteisen kokonaisuuden saavuttaminen varioimalla teemoja
3. kyky luoda uusia yllätyksiä ja lisätä pelinautintoa
4. mahdollisuus lisätä musiikillisia elementtejä pelin ominaisuuksiksi.

(Collins 2008a, 139.)

Kaikki pelimusiikki ei suinkaan ole adaptiivista. Toisissa peleissä asiaan on panostettu enemmän, toisissa sitä ei ole pidetty tärkeänä. Riippuu myös paljon pelityypistä, onko adaptiivinen musiikki ylipäänsä tarpeellista. Esimerkiksi jos peli koostuu hyvin nopeasti vaihtuvista, lyhyistä pelikentistä, voidaan riittävän hyvään lopputulokseen päästä käyttämällä tavallisia, lineaarisia kappaleita. Adaptiivisen musiikin luominen vaatii myös enemmän aikaa ja rahaa, minkä takia siitä voidaan olla valmiita tinkimään. Kuitenkin

laatutietoisten pelaajien ja suuren suosion ansiosta pelien tuotantoarvot nousevat jatkuvasti, ja tämä puoltaa myös adaptiivisen musiikin käyttöä. (Collins 2008a, 139.) Jos aikaisemmin adaptiiviset musiikkijärjestelmät olivat vain suurten pelitalojen mahdollisuus, nykyään kaikenkokoiset pelialan yritykset ymmärtävät niiden positiivisen vaikutuksen pelikokemukseen. Adaptiivisesta musiikista tuleekin vuosi vuodelta enemmän ja enemmän standardi pelialalla, joskin adaptiivisuuden määrä vaihtelee huomattavasti eri pelien välillä (Young 2012, 32).

Peleissä on myös paljon tilanteita, jotka eivät välttämättä hyödy adaptiivisesta musiikista tai jotka erityisesti vaativat lineaarista musiikkia. Tarinankerrontaa viedään usein eteenpäin elokuvamaisilla, käsikirjoitetuilla pätkillä, joissa pelaaja ei voi hallita tilannetta. Tällaisia pätkiä kutsutaan nimillä *cinematic* tai *cut scene* ja niissä käytetään nimenomaan lineaarista musiikkia, aivan kuten elokuvissa. Lineaarista musiikkia käytetään myös muissa tilanteissa, joissa ei ole varsinaista pelimekaniikkaa - esimerkiksi lataus- ja valikkoruudut. (Marks 2001, 188–193; Philips 2014 182.)

3 Videopelimusiikin historia lyhyesti

Videopelien kehitys on tiukasti sidoksissa pelaamisen mahdollistavan teknologian kehitykseen. Pelialalla uudet teknologiset mahdollisuudet on aina osattu ottaa nopeasti käyttöön ja niitä on hyödynnetty ennakkoluulottomasti. Sama kehityskaari koskee myös peliääntä ja pelimusiikkia, mutta etenkin peliteknologian alkuaikoina ääniominaisuudet kehittyivät muita osa-alueita hitaammin, koska aluksi tietotekniikkaa kehitettiin ensisijaisesti bisneskäyttöön, jossa äänellä ei ollut niin suurta roolia. (Collins 2008b, 2.)

3.1 Ohjelmoitavat äänigeneraattorit

Ensimmäiset suuren yleisön saataville tulleet videopelit olivat niin sanottuja arcade-pelejä, verrattain suuria, yhden pelin sisältäviä laitteita, joita pelattiin pelihalleissa. Ensimmäinen massatuotettu arcade-peli oli *Computer Space*, joka julkaistiin vuonna 1971. Se sisälsi muutamia ääniefektejä, muttei vielä musiikkia. Ensimmäinen varsinaisen taustamusiikkiääniraidan sisältävä peli oli *Space Invaders* vuonna 1978. Musiikki koostui neljästä kromaattisesti laskevasta bassonuotista. Huomattavaa on, että pelin kehittäjät ymmärsivät jo tuolloin tarpeen jonkinlaiselle adaptiivisuudelle, ja kyseisen

pelin tapauksessa se toteutettiin nopeuttamalla musiikin tempoa pelaajan edetessä pelissä eteenpäin. Jo ensimmäinen varsinainen videopelimusiikkiääniraita oli siis adaptiivinen. (Collins 2008b, 8–12.)



Kuvio 2. Arcade-pelejä

Pelimusiikki perustui pitkään syntetisoituun ääneen, jota ohjattiin suoraan ohjelmointikielellä. Musiikin implementoiminen peliin oli siis hyvin työlästä ja työtavat kaukana tavallisesta säveltämisestä tai soittamisesta. Tuon ajan pelisäveltäjät olivatkin pääosin hyvin tekniikkaan orientoituneita. Keskeisenä komponenttina olivat eritasoiset ohjelmoitavat äänigeneraattorit, äänentuottamiseen erikoistuneet mikrosirut. Ääni oli alkeellisesti syntetisoitua, oskillaattorilla tuotettua ja aluksi mahdollisuudet äänenvärien muokkamiseen olivat olemattomat. Myös samanaikaisten äänien rajallinen määrä rajoitti pitkään sävellystyötä. Äänisirut kehittyivät pikkuhiljaa ja mahdollistivat useammat samanaikaiset äänet, erilaiset aaltomuodot ja kehittyneemmät syntetisointitekniikat. Erityisen suuressa roolissa sirujen ja niiden käytettävyyden kehityksessä oli Atari, joka muun

muassa julkaisi ensimmäisen kotitietokoneen, jossa oli integroitu MIDI-tuki⁵. (Collins 2008b, 12.)

Ohjelmoitavat äänigeneraattorit määrittivät peliääntä varsin pitkään, mutta niiden rinnalla alettiin jo 1980-luvulla käyttämään alkeellisia DA-muuntimia⁶, jotka mahdollistivat äänitettyjen PCM-äänien toistamisen. Muun muassa pelikonsoliklassikoksi muodostunut, vuonna 1983 julkaistu Nintendo NES pystyy toistamaan neljää syntetisoitua ääniraitaa ja yhtä 6-bittistä, 33 kilohertsin PCM-raitaa. Konsolin seuraava versio, vuonna 1991 julkaistu Super NES käytti jo pelkästään DA-muunnosta mahdollistaen kahdeksan samanaikaisen 8-bittisen, 32 kilohertsin PCM-raidan toiston. Äänenlaatu oli siis edelleen selkeästi heikompi kuin esimerkiksi Audio-CD-levyissä, joissa ääni on tallennettu 16-bittisenä ja 44,1 kilohertsin näytteenottotaajuudella. (Horowitz, Looney 2014, 27–28.)

3.2 Tekniikan kehitys

Vuonna 1995 julkaistu Sony PlayStation -konsoli mullisti monella tapaa videopelaamisen ja merkitsi myös selkeän käännekohdan peliäänen historiassa. Laitteen 24-kanavainen äänisiru mahdollistaa Audio-CD-tasoisesta äänenlaadun ja tukee erilaisia digitaalisia efektejä, esimerkiksi kaikuja. Huimasti kasvanut keskusmuistin ja tallennustilan määrä mahdollisti monipuolisemman äänimaiseman käytön. Samanlaisia ominaisuuksia alettiin PlayStationin julkaisun myötä hyödyntää myös pöytätietokoneissa, jotka nousivat 1990-luvulla kilpailemaan markkinoista konsolien kanssa. Tämän kehityksen myötä peliääneen liittyneet rajoitukset vähenivät merkittävästi ja pelimusiikinkin tuottamista oli mahdollista lähestyä enemmän kuten mitä tahansa muuta musiikintuotantoa. Peleissä yleistyivätkin muun muassa elokuvamaiset täysin orkestroidut musiikkiraidat. (Horowitz, Looney 2014, 31; Alten 2012, 453.)

3.3 Nykyinen tilanne

Pelikonsolit ja tietokoneet ovat kehittyneet kovaa vauhtia, ja nykyaikaiset konsoli- ja tietokonepelit tarjoavat monikanavaista surround-ääntä ja toistavat vaivatta satoja

⁵ MIDI eli *Musical Instrument Digital Interface* on standardi, joka mahdollistaa tietokoneen ja erilaisten instrumenttien ja ohjainten keskinäisen kommunikoinnin (Guerin 2005, 1).

⁶ DA-muunnin muuntaa digitaalisen signaalin analogiseksi.

ääniraitoja samanaikaisesti. Reaaliaikaiset efektit ja pelitilanteiden mukaan vaihtuva miksaus sekä toinen toistaan monimutkaisemmat adaptiiviset musiikkijärjestelmät ovat arkipäivää. Vanhat rajoitukset ovat historiaa, ja pääosin äänisuunnittelun rajana on enää taiteilijan mielikuvitus. Toki esimerkiksi tiedostokokoja ja muistinkäyttöä täytyy edelleen pitää silmällä, mutta niidenkin hallitsemiseen on olemassa parempia työkaluja ja -tapoja. (Horowitz, Looney 2014, 32.)

Samaan aikaan kun rajoitukset konsoli- ja tietokonepelien äänimaailman suhteen ovat käytännössä poistuneet, markkinoita hallitsemaan ovat nousseet mobiilipelit, joiden myötä vanhat tekniset rajoitukset ovat taas vahvasti läsnä pelinkehityksessä. Mobiililaitteet ovat heikompia suorituskyvyltään ja tallennuskapasiteetiltaan, minkä takia äänenlaadusta joudutaan usein tinkimään ja äänitiedostojen määrää rajoittamaan. Käytävissä olevien resurssien onnistunut käyttö vaatii huolellista suunnittelua. (Horowitz, Looney 2014, 196.)

4 Mobiilialusta

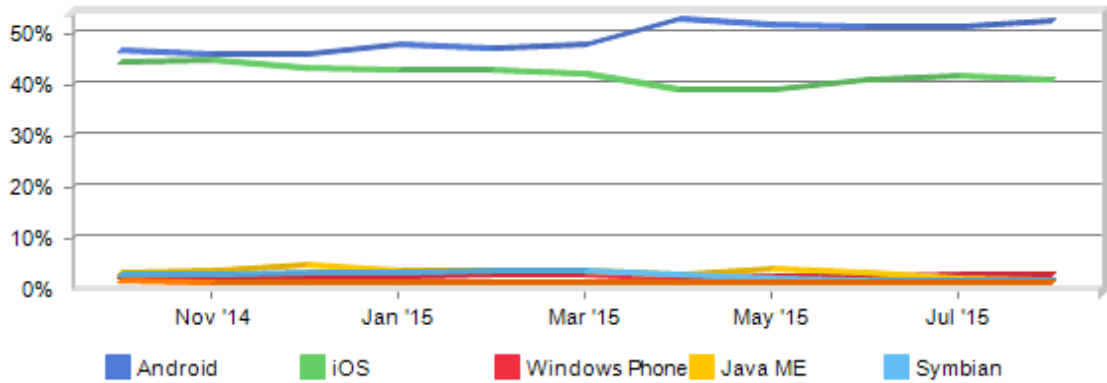
Mobiilipelien kehitykseen liittyy tiettyjä lainalaisuuksia, rajoitteita, ongelmia ja mahdollisuuksia verrattuna esimerkiksi konsoli- ja tietokonepeleihin. Äänisuunnittelun ja musiikin tuottamisen kannalta merkittävimmät rajoitukset liittyvät suorituskykyyn ja äänen-toisto-ominaisuuksiin. Merkittävimpiin etuihin kuuluvat ainakin valtavat, globaalit markkinat. Äänisuunnittelijan ja musiikin tuottajan on hyvä tietää mobiilipeleihin liittyvät peruskäsitteet, käyttöjärjestelmistä liiketoimintamalleihin.

4.1 Suosituimmat mobiililaitteet

Mobiililaitteilla pelinkehityksen yhteydessä tarkoitetaan älypuhelimia ja tabletteja. Kannettavista pelikonsoleista puhutaan sen sijaan käsikonsoleina, ja ne eroavat mobiililaitteista merkittävästi muun muassa käyttöjärjestelmiensä osalta ja erityisesti, koska ne ovat suunniteltuja pelkästään pelaamiseen. En käsittele käsikonsoleita sen tarkemmin tässä työssä, mutta niihin kohdistuvat pääosin samanlaiset rajoitukset esimerkiksi suorituskyvyn ja muistinkäytön suhteen kuin mobiilipeleihin.

Tämän työn kirjoittamisen aikaan ylivoimaisesti myydyimmät mobiililaitteet ovat Android- ja iOS-pohjaisia. Luonnollisesti niille tehdään myös eniten pelejä. Muita tunnettuja

mobiilikäyttöjärjestelmiä ovat muun muassa Windows Phone ja BlackBerry, mutta niiden markkinat ja pelitarjonta ovat huomattavasti pienemmät kuin tämänhetkisten markkinajohtajien, enkä siksi käsittele niitä tarkemmin.



Kuvio 3. Mobiililaitteiden käyttöjärjestelmien suosio maailmanlaajuisesti

Applen kehittämä iOS on vuonna 2007 julkaistu käyttöjärjestelmä, ja sitä käytetään vain Applen valmistamissa laitteissa, joista pelinkehityksen kannalta tärkeimmät ovat iPhone- ja iPad-tuotelinjat. (Scolastici, Nolet 2013, 18.) Vuonna 2008 julkaistujen iPhone 3G:n ja yksinkertaisen sekä helppokäyttöisen ohjelmistojen latauspalvelun App Storen myötä iOS-laitteet nousivat nopeasti suureen suosioon ja laajensivat kertaheitolla mobiilipelimarkkinoita merkittävästi. (Olson, Hunter, Horgen, Goers 2012, 25.)

Google lähti nopeasti kilpailuun mukaan kehittämällään Android-käyttöjärjestelmällä. Merkittävin ero iOS:iin verrattuna on avoin lähdekoodi ja käyttömahdollisuus minkä tahansa valmistajan laitteissa. Erilaisia Android-pohjaisia laitteita valmistavatkin useat kymmenet eri yhtiöt, tunnetuimpina Samsung ja Motorola. Ohjelmistoja ja pelejä Androidille tarjoaa myös usea internet-jakelualusta, tärkeimpinä Google Play ja Amazon App Store. (Scolastici, Nolet 2013, 12.)

Äänen kannalta merkittävin ero on juuri laitekantojen monimuotoisuudessa. Applen laitteilla ääniominaisuuksien voi olettaa olevan riittävän samanlaiset kautta linjan ja esimerkiksi pelin testaaminen yhdellä laitteella voi hyvin riittää, etenkin musiikin osalta. Android-laitteiden ääniominaisuudet sen sijaan voivat vaihdella enemmän, ja lopputulos voi olla arvaamaton jollain tietyllä laitteella. Käytännössä kaikilla mahdollisilla laitteilla testaaminen voi olla myös aika- ja budjettisyistä mahdotonta, joten äänisuunnittelijan on syytä varautua yllättäviin ongelmiin, jotka saattavat tulla ilmi vasta

pelin julkaisun jälkeen. Mobiilipelejä tavataan kuitenkin päivittää monesti hyvin tiheäänkin tahtiin, joten ongelmien korjaaminen julkaisun jälkeenkin onnistuu, ja on ainakin pienessä määrin enemmän sääntö kuin poikkeus.

4.2 Rajallinen suorituskyky

Teknisiltä ominaisuuksiltaan ja näin ollen suorituskyvyltään mobiililaitteet ovat luonnollisesti uusimpia pelikonsoleita ja keskimääräisiä pelikäyttöön tarkoitettuja tietokoneita heikompia. Syyt tähän ovat ilmeiset: mobiililaitteet ovat pienikokoisia ja toimivat pienellä akulla. Ne eivät myöskään ole suunniteltuja pelkästään pelaamiseen. Elektroniikan nopean kehityksen ja edullistumisen myötä mobiililaitteet ovat kuitenkin kehittyneet varteenotettaviksi, suorituskykyisiksi ja suosituiksi pelialustoiksi. Samaan aikaan konsolit ovat kehittyneet entistä tehokkaammiksi, suuremman kokonsa ja pelikeskeisyytensä ansiosta. Keskeiset suorituskykyyn vaikuttavat ominaisuudet ovat keskusmuisti (RAM), prosessoriteho (CPU) ja kovalevyn koko. (Alten 2012, 440–441.)

Äänisuunnittelun kannalta suurimmat huolenaiheet ovat muistinkäyttö ja tiedostokoot. Ne vaativat huomiota kaikissa peliprojekteissa, mutta ongelmat kärjistyvät mobiilipeleissä. Esimerkiksi tämän työn kirjoittamisen aikaan Applen uusimmassa tabletissa iPad Air 2:ssa on 2 gigatavua muistia kun taas Sonyn uusimmassa Playstation 4-konsolissa sitä on 8 gigatavua (IFIXIT, iPad Air 2 Teardown; Playstation 4, valmistajan verkkosivut). Mobiilipelejä kehittäessä yleensä pyritään myös mahdollisimman laajaan mahdolliseen laitekantaan eli pelit yritetään saada toimimaan myös vanhemmilla laitteilla, joissa muistia voi olla esimerkiksi vain 512 megatavua. Tämä vaikuttaa merkittävästi kaikkeen sisältöön ja äänen osalta erityisesti musiikkiin, koska tyypillisesti se kuormittaa muuta äänisisältöä enemmän muistia. Musiikkiraidat ovat äänitehosteita pidempiä, ja monesti niitä halutaan pakata vähemmän.

Mobiilipelien tiedostokoko pyritään pitämään mahdollisimman pienenä. Osittain tämä johtuu laitteiden rajallisesta tallennustilasta, mutta myös siitä, että pelit ladataan poikkeuksetta internetistä ja pitkät latausajat karkottavat osan mahdollisista käyttäjistä. Laitteet ja ohjelmistokaupat saattavat myös asettaa rajoituksia ohjelmiston tiedostokoolle. Esimerkiksi Applen App Storessa suurin sallittu tiedostokoko on 4 gigatavua ja mobiilidataa käyttäen voi ladata vain alle 100 megatavun tiedostoja, suuremmat vaativat Wi-Fi-yhteyden (Marsal, 2015).

Tekniikan nopean kehityksen myötä rajoitukset muuttuvat jatkuvasti ja kehityksen suunta on ainakin toistaiseksi ollut se, että mobiililaitteiden suorituskyky ottaa kiinni konsoleja. Tulevaisuudessa voikin olla, ettei samanlaisia suorituskykyyn liittyviä rajoitteita enää ole. Toisaalta paljon riippuu myös tietoliikenneyhteyksien kehityksestä; latausajat ja tiedostokoot voivat jatkossakin olla mobiilipelimarkkinoiden kipupisteitä. Joka tapauksessa suorituskyvyn huomioon ottaminen ja optimoiminen on erottamaton osa pelinkehitystä ja siten myös peliäänisuunnittelua.

Pelimusiikin kannalta muistinkäytön ja tiedostokoon optimoiminen täytyy ottaa huomioon käytännössä erityisesti käytettävien looppien määrää ja pituutta suunniteltaessa. Konsoli- ja tietokonepeleihin verrattuna mobiilipeleissä looppeja on keskimäärin paljon vähemmän, ja ne ovat lyhyempiä. Musiikintekijän täytyy jatkuvasti arvioida, koska musiikkipätkiä on riittävästi, jotta kuuntelukokemus ei käy puuduttavaksi, mutta kuitenkin riittävän vähän, jotta peli pysyy vaaditussa koossa ja muistinkäytössä. Lopulta yhtälöön liittyy vielä äänenlaatu, eli kuinka paljon ääntä pakataan.

4.3 Äänen pakkaaminen

Käytännössä pakkaamatonta ääntä käytetään mobiilipeleissä hyvin harvoin. Sen sijaan käytetään jollain tavalla pakattua ääntä eli ääni enkoodataan tiettyyn tiedostomuotoon peliä kehitettäessä ja dekodataan toistaessa. Pakatut tiedostomuodot eroavat toisistaan pääosin siinä, miten paljon tiedostokoossa saavutetaan etua ja miten paljon dekodaus kuormittaa prosessoria. Mitä enemmän tiedostomuoto vaatii prosessorilta sitä suurempi latenssi, ja toisaalta sama suorituskyky on poissa pelin muilta osa-alueilta.

Lyhyet äänitehosteet pakataan mobiilipeleissä tyypillisesti IMA ADPCM-tiedostomuotoon, jolla saavutetaan 4:1 etu tiedostokoossa ja joka kuormittaa vain hyvin vähän prosessoria. (Horowitz, Looney 2014, 206.) Äänitehosteissa latenssin pienuus pysyminen on yleensä tärkeämpää kuin musiikissa. Esimerkiksi miekan iskuun liitetystä äänitehosteesta huomaa välittömästi, jos se toistuu vähänkin myöhässä, kun taas musiikin vaihtuminen seuraavaan osaan kestää helposti sekunninkin latenssin antamatta kömpelöä vaikutelmaa. IMA ADPCM:n tarjoama pakkaussuhde on kuitenkin varsin pieni ja usein riittämätön pidemmille äänitehosteille ja musiikille. Toisaalta ADPCM-pakkaus tapahtuu myös vahvasti näytteenottajuuden kustannuksella ja sen alentamisen aiheuttamat lieveilmiöt erottuvat musiikista selvemmin kuin lyhyistä äänitehosteista.

Musiikki pakataan tyypillisesti joko MP3 tai Ogg Vorbis -formaatteihin. Näillä saavutetaan 10:1 tai jopa suurempi pakkaussuhde, mutta pakkauksen dekoodaaminen kuormittaa prosessoria enemmän. Musiikki ja pitemmät äänitehosteet kuten ambienssit yleensä niin sanotusti *streamataan* eli pakattu tiedosto luetaan toiston alkaessa muistiin pieni osa kerrallaan ja dekodataan lennosta. Tapa kuormittaa muistia vähemmän kuin jos koko tiedosto dekodattaisiin muistiin yhtenä palasena ennen toistoa. Streamaus kuormittaa kuitenkin prosessoria ja tallennusmediaa⁷, ja liian monen tiedoston yhtäaikainen streamaus saattaa aiheuttaa epätoivottuja ilmiöitä äänentoistossa tai pelin toiminnassa ylipäättään. Applen laitteet ja suuri osa Android-laitteista tarjoaa myös niin kutsutun *hardware decoding* -ominaisuuden yhdelle streamattavalle tiedostolle kerrallaan. Käytännössä tämä tarkoittaa, että dekodaukselle on laitteessa varattu oma pieni prosessori, jolloin tätä käytettäessä varsinainen prosessori ei kuormitu ollenkaan. (Unity verkkomanuaali, Audio clip; Horowitz, Looney 2014, 206.) Oman kokemukseni mukaan turvallisella alueella ollaan, jos samanaikaisesti käytetään korkeintaan yhtä streamattavaa tiedostoa hardware-dekoodatun streamin lisäksi. Tällöin vanhemmillakin laitteilla äänet toistuvat luotettavasti, eikä dekodaus vie liikaa prosessointitehoa muulta peliltä. Toisaalta jos peli on muilta osiltaan hyvin kevyt ja jos vanhempia laitteita ei ole tarkoituskaan tukea, useammat streamit eivät ole mikään ongelma.

4.4 Taajuusvaste ja stereokuva

Oman haasteensa asettavat myös runsaasti vaihtelevat kuunteluympäristöt ja laitteiden taajuusvaste. Konsolipelejä pelataan yleensä vähintään television kaiuttimilla, monesti erilaisten hifi-järjestelmien ja 5.1-kotiteattereiden kanssa tai kuulokkeilla. Mobiililaitteiden kaiuttimet sen sijaan ovat hyvin pienet, eivätkä siksi toista esimerkiksi käytännössä ollenkaan bassotaajuuksia. Toisaalta pelaaja saattaa käyttää myös kuulokkeita, jolloin taajuusvaste on aivan toisenlainen. Eryityisesti bassoalueen informaation hallittu ja riittävä välittyminen sekä laitteen kaiuttimilla että kuulokkeilla vaatii paljon työtä, luovuutta ja ongelmanratkaisutaitoja. (Horowitz, Looney 2014, 207)

Stereokuva on toinen ongelmakohta. Useimpien mobiililaitteiden kaiuttimet toistavat äänen monona, ja toisaalta stereokaiuttimellisissa laitteissa kaiuttimet ovat niin lähellä toisiaan, että äänikuva on käytännössä mono. Yksi vaihtoehto onkin miksata musiikki

⁷ Mobiililaitteiden tallennusmedia on useimmiten jonkinlainen flash-muisti eli perinteistä kiintolevyä huomattavasti kompaktimpi puolijohdemuisti.

monotiedostoiksi. Tällä tavalla myös tiedostokoko ja muistinkuormitus ovat luonnollisesti puolet pienempiä. Toisaalta kuulokekuuntelussa käytössä on taas koko stereokuva, mikä täytyy myös ottaa huomioon. Monoksi miksattu musiikki jää kuulokkeilla helposti hyvin vaatimattoman ja yksiulotteisen kuuloiseksi. Hyvä kompromissi on pitää musiikki stereona, mutta huolehtia monoyhteensopivuudesta kiinnittämällä siihen useasti huomiota ja varmistamalla, että tärkeimmät elementit erottuvat ja ovat keskenään riittävän hyvässä suhteessa molemmissa tilanteissa.

Samanlaiset rajoitukset ja haasteet pätevät toki jossain määrin kaikessa musiikin miksausessa, koska materiaalia saatetaan hyvin kuunnella myös mobiililaitteilla. Ongelmat kuitenkin kärjistyvät mobiilipeleissä, koska niitä pelataan pelkästään mobiililaitteilla. Hyvä työtapana onkin kuunnella työnsä jälkeä jatkuvasti sekä kohdelaitteilla että joillain keskivertokuulokkeilla. Parhaiten näillä molemmilla toimiva kompromissi saattaa lopulta kuulostaa tyypillisistä studiomonitoreista erikoiselta.

Toisaalta yleensä lopullinen kohdelaite on pelimusiikkia tehdessä tiedossa jo musiikin suunnitteluvaiheessa ja asiaa voi ottaa huomioon alusta lähtien. Näin esimerkiksi juuri bassoinformaation erottuvuuteen ja stereokuvan monoyhteensopivuuteen voi vaikuttaa jo säveltäessä, ei pelkästään miksatessa. Bassolinjojen tuplaaminen jollain enemmän alakeskialueelle sijoittuvalla soittimella tai vain yksinkertainen oktaavituplaaminen samalla soittimella on toimiva tapa varmistaa bassoäänten erottuminen kaikissa kuunteluympäristöissä.

4.5 Free-to-play vs. Premium

Mobiilipelit jakautuvat karkeasti liiketoimintamalleiltaan kahteen ryhmään. Niin sanotut *premium-pelit* noudattavat perinteistä konsoli- ja tietokonepeleistä tuttua mallia, jossa kuluttaja ostaa pelin maksamalla siitä tietyn summan ja voi sen jälkeen vapaasti pelata peliä kokonaisuudessaan. Monet tunnetuista hittipeleistä ovat premium-pelejä, esimerkiksi alkuperäinen *Angry Birds*, joka teki suomalaisen Rovio-yhtiön tunnetuksi maailmanlaajuisesti. (Scolastici, Nolte 2013, 283.)

Nykyään suosituimmat ja taloudellisesti eniten tuottavat mobiilipelit ovat kuitenkin pääosin niin kutsuttuja *free-to-play-pelejä*, jotka kuluttaja voi ladata itselleen ilmaiseksi. Tuotto saavutetaan pelissä tehtävien pienten maksusuoritusten tai pelin yhteydessä näytettävien mainosten avulla, monesti molemmat keinot ovat käytössä. Pelaaja voi

esimerkiksi ostaa itselleen pelissä käytettävää valuuttaa, jolla hän voi hankkia pelihahmolleen uusia varusteita tai rakentaa uusia rakennuksia. Ostettavat tuotteet vaihtelevat peligenren mukaan ja uudenlaisia rahastusmalleja kehitetään jatkuvasti. Myös osa pelin ominaisuuksista voi olla käytettävissä ainoastaan maksua vastaan. Valtaosa pelaajista ei osta mitään, mutta mallin toimivuus perustuu suureen, globaaliin käyttäjämäärään ja toisaalta siihen, että pieni osa pelaajista käyttää peliin suuriakin summia. (Vankka 2014, 1.)

Liiketoimintamalli ei välttämättä suoraan vaikuta pelin musiikkiin, mutta toisaalta pelin suunnittelussa sillä on yleensä merkittävä rooli ja siksi mielestäni asia on hyvä pitää mielessä myös musiikkia suunniteltaessa. Esimerkiksi premium-peleissä tarinan draaman kaari voidaan suunnitella tavallaan aivan vapaasti, koska käyttäjästä on tullut maksava asiakas jo ennen pelin pelaamisen aloittamista, hänhän on ostanut pelin. Pelimekaniikka voi olla monimutkaisempi, koska pelin alussa voidaan hyvin käyttää pitkiäkin opastusjaksoja, joilla pelaajaa opetetaan. Myös musiikin osalta draaman kaarta voi suunnitella samaan tapaan – alussa ei tarvitse vielä näyttää kaikkia temppuja.

Free-to-play-peleissä pelikokemus pyritään suunnittelemaan niin, että pelaajat palaisivat pelin pariin mahdollisimman usein ja toisaalta haluaisivat käyttää rahaa johonkin pelin sisäiseen ostoon. Vaikeat pelimekaniikat ja liian hidas tarinankerronta saattavat karkottaa osan mahdollisista asiakkaista nopeasti, eivätkä he ehkä palaa enää koskaan pelaamaan peliä. Tyypillisiä tapoja lähestyä tätä asiaa ovat pelin alkupuolen pitäminen helppona, mutta mahdollisimman mielenkiintoisena ja nopeasti etenevänä sekä pelin sisäisten ostojen houkuttelevuuden maksimoiminen kaikilla mahdollisilla keinoilla. Tämä lähtökohta on hyvä pitää mielessä myös musiikin suhteen: vaikutuksen tekeminen heti alusta alkaen voi olla tärkeämpää kuin pitkän draaman kaaren muodostaminen. Kehittelyä ja draamaa voi tuki rakentaa myöhemmin, mutta ensimmäisen pelikokemuksen tulisi olla pelaajalle mahdollisimman vaikuttava ja helposti lähestyttävä. Myös pelimekaniikkaa ja ominaisuuksia lähestytään usein tällä tavoin ja kehittyneempiä sekä vaikeammin omaksuttavia ominaisuuksia esitellään vasta pelin myöhemmissä vaiheissa, jolloin pelaaja on jo tykästynyt peliin.

Toisaalta musiikki voi itsessään olla myös tuote pelin sisäisten ostojen joukossa. Pelissä voidaan esimerkiksi tarjota erilaisia vaihtoehtoisia musiikkipakkauksia ladattavaksi maksua vastaan. Toistaiseksi tällaista mallia ei musiikin kanssa olla vielä käytetty ainoakaan tunnetuissa mobiilipeleissä, konsolimarkkinoillakin lähinnä musiikkipeleissä kuten

Guitar Hero ja *Rock Band*. Muun äänisisällön osalta sen sijaan löytyy esimerkkejä mobiilimarkkinoiltakin. Muun muassa Hi-Rez Games -yhtiön julkaisema *Smite* tarjoaa pelihahmoille vaihtoehtoisia puheäänipaketteja pelin sisäisenä ostona. (Hahl 2014, 47–50.)

4.6 Pelaako kukaan edes äänet päällä?

Peleissä äänimaisema mielletään usein lähinnä koristeeksi, ikään kuin erilliseksi kerrokseksi, joka on läsnä vain tukeakseen ja korostaakseen tiettyä tunnelmaa. Tämä onkin yksi peliäänen tärkeistä rooleista, mutta sillä voi olla myös muita funktioita. Äänitehosteet voivat tarjota pelitilanteesta informaatiota, joka suoraan vaikuttaa pelaajan toimintaan. Sama koskee myös musiikkia, esimerkiksi lähestyvän vaaran voi aistia musiikin sävyn muuttumisesta. (Collins 2008b, 163; Moormann 2013, 192.)

Mobiilipeleissä on perusteltua suunnitella pelikokemus niin, että siitä saa kaiken tarvittavan irti myös ilman ääntä, koska vähintään olosuhteiden pakosta pelaajat saattavat toisinaan pelata peliä ilman ääntä. (Collins 2008a, 127–128.) Suorituskyvyn takia ja ehkä kustannussyistä mobiilipelialalla äänisuunnittelua ja musiikkia ei aina priorisoida kovin korkealle ja monien suosituidenkin pelien äänimaailma vaikuttaa mielestäni varsin huolimattomalta. Silti esimerkiksi Appingtonin (2013) tutkimuksen mukaan valtaosa, jopa 73 % mobiilipelien pelaajista pelaa äänet päällä. Tämän perusteella myös huolellista panostamista pelien äänimaisemaan voi mielestäni pitää hyvin perusteltuna ja oleellisena. Kehityssuunta onkin positiivinen, ja yhä useammassa mobiilipeleissä on nykyään laadukkaat ja hyvin suunnittelut äänet sekä musiikki.

5 Pelimoottorit

Nykyään pelinkehitykseen käytetään käytännössä poikkeuksetta jonkinlaista pelimoottoria (engl. *game engine*). Pelimoottorin tarkoituksena on yksinkertaistaa ja nopeuttaa pelinkehitystä tarjoamalla esimerkiksi visuaalisia rakenteita ja työkaluja, joiden avulla pelin eri osa-alueita voidaan kehittää ja yhdistää. Suuri osa pelinkehittämiseen osallistuvista työntekijöistä, kuten animaattorit, käsikirjoittajat ja äänisuunnittelijat, työskentelevät osittain pelimoottorin parissa, vaikeivätkin välttämättä ymmärrä esimerkiksi ohjelmoinnista mitään. Merkittävä ominaisuus on myös kerran luotujen rakenteiden ja

ominaisuuksien uudelleenkäytön mahdollistaminen sekä samassa että muissakin projekteissa. Useasti myös kokonaisten projektien kääntäminen useammille alustoille on tehty mahdollisimman vähätoiseksi. (Ward, 2008.)

Ennen pelimoottoreita jokainen peli täytyi ohjelmoida käytännössä pienintäkin ominaisuutta myöten alusta alkaen ja jollekin tietylle alustalle. Jos sama peli haluttiin siirtää myös jollekin toiselle alustalle, esimerkiksi PC:ltä konsolille, täytyi kaikki ohjelmointi tehdä uudestaan. Sama urakka toistui jokaisessa projektissa ja vaati runsaasti aikaa ja resursseja. Jonkinasteisen standardisoinnin ja uudelleenkäytettävyyden tarve tuli nopeasti ilmeiseksi ja johti suoraan pelimoottoreiden syntyyn. (Horowitz, Looney 2014, 149.)

Nykyaikaiset pelimoottorit ovat yksiä monimutkaisimmista ohjelmistoista, joita on olemassa. Ne koostuvat useista erilaisista järjestelmistä, jotka vastaavat muun muassa pelin kuvista, graafisista malleista, valoista, fysiikasta, animaatioista, tekoälystä, tiedostohierarkioista ja äänistä. Lisäksi niihin on mahdollista liittää kolmannen osapuolen tarjoamia väliohjelmistoja (engl. middleware), joiden myötä saadaan tarvittaessa lisää ominaisuuksia ja työkaluja. (Horowitz, Looney 2014, 154.) Toisinaan peliyhtiöt saattavat luoda itse pelimoottorinsa, mutta useammin käytetään jotain jo olemassa olevaa vaihtoehtoa. Tämän työn kirjoittamisen aikaan suosittuja pelimoottoreita ovat muun muassa Unreal Engine, CryEngine ja Unity, jota käytetään paljon erityisesti mobiilipeleissä.



Kuvio 4. Unity-pelimoottorin muokkausnäkyvä

6 Ääniväliohjelmistot

6.1 Äänisuunnittelijasta implementoijaksi

Tyypillinen työtapa pelien äänisuunnittelussa oli pitkään, että äänisuunnittelija toimittaa äänitiedostot ja jonkinlaiset implementointiohjeet ohjelmoijalle, joka ohjelmoi äänet kiinni peliin. Tämän jälkeen parhaassa tapauksessa äänisuunnittelija testaa peliä, muokkaa äänitiedostoja tarpeen mukaan ja antaa kommentteja implementoinnin parantamiseksi, esimerkiksi äänenvoimakkuuksiin ja ajoituksiin liittyen. Tätä tapaa käytetään silloin tällöin vieläkin, ja se on etenkin äänisuunnittelun osalta periaatteessa toimiva, mutta vie runsaasti aikaa molemmilta osapuolilta. Ohjelmoijan ääni-implementointiin käyttämä aika on myös suoraan pois muusta ohjelmointityöstä. (Horowitz, Looney 2014, 124). Adaptiivisen musiikin kanssa tapa käy entistä hankalammaksi, koska musiikkiin liittyvät parametrit voivat olla hyvin monimutkaisia ja vaadittavat säädöt hienovaraisia sekä useita kokeiluja vaativia. Lisäksi ohjelmoija ei välttämättä ymmärrä musiikkia säveltäjän tavoin, jolloin esimerkiksi tempoon ja äänen miksaukseen liittyvät ongelmat voivat olla vaikeita ratkaista. Pahimmillaan tapa kannustaa yksinkertaisiin ja tylsiin ratkaisuihin ja huolimattoman kuuloiseen lopputulokseen.

Nykyään tätä edestakaisin pallottelua helpottamaan on olemassa niin kutsuttuja *ääniväliohjelmistoja* (engl. *audio middleware*). Ne on suunniteltu vähentämään ohjelmoijan tarvetta äänien implementoinnissa ja antamaan äänisuunnittelijalle työkaluja muun muassa äänenvoimakkuuksien, ajoituksien ja siirtymien säätämiseen sekä äänien testaamiseen pelissä. Ääniväliohjelmistot ovat kehittyneet nopeasti ja niiden käyttö alkaa olla nykyään enemmän sääntö kuin poikkeus. Suosituimmat vaihtoehdot ovat paljolti *DAW-ohjelmistojen*⁸ kaltaisia ja niiden oppimiskäyrä musiikkitekologiaan vähääkään perehtyneelle on verrattain lyhyt. (Horowitz, Looney 2014, 125.) Erytisesti musiikin kanssa toimittaessa moniraitaominaisuudet, ristihäivytykset ja testauksen yksinkertaisuus ovat merkittäviä etuja, jotka tekevät pelimusiikista säveltäjälle helpommin lähestyttävää ja toisaalta mahdollistavat laadukkaamman lopputuloksen.

⁸ DAW eli *Digital Audio Workstation* -ohjelmistoilla voidaan äänittää, käsitellä ja miksata ääntä tietokoneella. Tunnettuja DAW:eja on muun muassa Pro Tools, Logic ja Cubase.

6.2 Toimintaperiaate

Useimmat ääniväliohjelmistot toimivat kutakuinkin samalla periaatteella ja käyttävät samoja käsitteitä. Yleensä ohjelmistoista löytyy tapahtumia ja parametreja sekä erilaisia miksaus- ja efektointiominaisuuksia.

6.2.1 Tapahtumat ja parametrit

Tyypillisesti ääniväliohjelmiston ja pelimoottorin välinen kommunikointi perustuu tapahtumiin (engl. *event*) ja parametreihin (engl. *parameter*). Käytännössä eri pelitapahtumat määrätään kutsumaan tietyn nimisiä äänitapahtumia. Äänisuunnittelija voi tämän jälkeen väliohjelmistoa käyttäen itsenäisesti liittää tapahtumiin haluamansa äänitiedostot, testata niitä pelissä ja miksata äänimaailmaa haluamallaan tavalla tai esimerkiksi kokeilla vaihtoehtoisia äänitiedostoja. (Horowitz, Looney 2014, 126.)

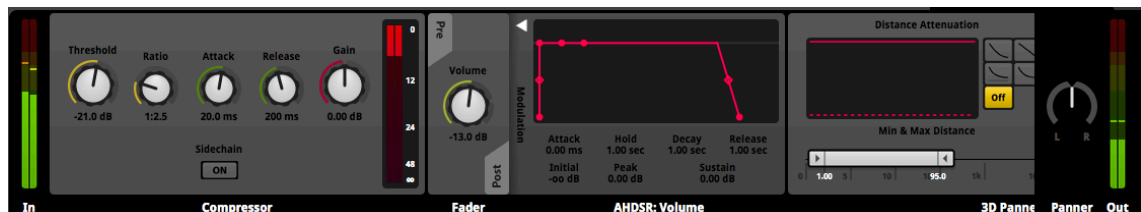
Parametrit ovat erilaisia arvoja, joita pelimoottori voi tapahtumien lisäksi välittää ääniväliohjelmistolle. Äänisuunnittelija voi väliohjelmistoa käyttäen määrittää, miten parametrit vaikuttavat äänitapahtumiin. Esimerkiksi autopelissä voitaisiin käyttää auton nopeudelle parametria, joka ohjaisi auton moottorin ääntä. Yksinkertaisimmillaan parametri voitaisiin asettaa ohjaamaan äänenvoimakkuutta ja sävelkorkeutta niin, että moottorin ääni soisi kovempaa ja korkeammalla taajuudella auton nopeuden kiihtyessä. Realistisempaan lopputulokseen päästäisiin käyttämällä useampaa äänitettä, joista siirryttäisiin nopeuden muuttuessa aina edelliseen tai seuraavaan ristihäivytyksen kautta. (Horowitz, Looney 2014, 126.)

Adaptiivisen musiikin rakentamisessa parametrit tarjoavat useita käteviä mahdollisuuksia. Yhden pelikentän koko musiikki voisi esimerkiksi olla yksi tapahtuma, jota ohjattaisiin useilla parametreilla. Kentän alkaessa tapahtuma käynnistettäisiin ja loppuessa pysäytettäisiin. Tällä välillä pelattaisiin pelkästään parametrien säätämällä. Jos pelaajan olisi tarkoitus selviytyä maalialueelle koko ajan lisääntyvän vihollisarmeijan läpi, parametreina voisi olla kentällä samaan aikaan olevien vihollisten määrä ja pelaajan etäisyys maalialueesta. Musiikin eri osia ohjattaisiin näiden parametrien avulla niin, että se muuttuisi dramaattisemmaksi vihollismäärän kasvaessa, mutta toisaalta voitokkaammaksi mitä lähemmäksi pelaaja pääsisi maalialuetta.

6.2.2 Miksaus ja efektit

Ääniväliohjelmistot tarjoavat erilaisia DAW:eista tuttuja mahdollisuuksia äänen miksaamiseen. Vähimmillään tämä tarkoittaa äänenvoimakkuuden ja panoroinnin säätimiä, mutta kaikissa käytetyimmissä ääniväliohjelmistoissa on myös useampia reaaliaikaisia efektointimahdollisuuksia, esimerkiksi kaikuja, viiveitä, dynamiikkaprosessoreita, ekvalisaattoreita ja modulaatioefektejä kuten chorus ja flanger. Näitä käyttämällä voidaan saavuttaa merkittäviä etuja erityisesti dynamiikan ja jälkikaikujen suhteen.

Esimerkiksi eri soitinryhmien kaiuttaminen reaaliajassa pelin käydessä on yksinkertainen tapa pehmentää adaptiivisen musiikkiraidan siirtymäkohtia. Jos esimerkiksi siirtymäkohdassa vaihtuva sellokuvio olisi kaiutettu etukäteen, kaiun leikkaantuminen kuulostaisi helposti häiritsevältä. Käyttämällä kaikua reaaliajassa ääni leikkaantuu ikään kuin ennen kaiuttamista, ja lopputuloksena on luonnollisemman kuuloinen siirtymä. Hyvin yleisesti käytetty reaaliaikainen efekti on masterkompressor tai -limitteri, jolla varmistetaan, että adaptiivisesti jatkuvasti muuttuva ääniraita pysyy dynamiikaltaan hallussa aktiivisimmissakin tilanteissa. Mobiilipeleissä verrattain runsaan masterkompressoinnin avulla voidaan myös varmistaa, että ääniraita välittyy tarpeeksi selkeästi hankalissakin kuunteluolosuhteissa, esimerkiksi älypuhelimien omilla kaiuttimilla kuunneltaessa liikennemelun ympäröimänä.



Kuvio 5. Masterkompressor FMOD-ääniväliohjelmistossa

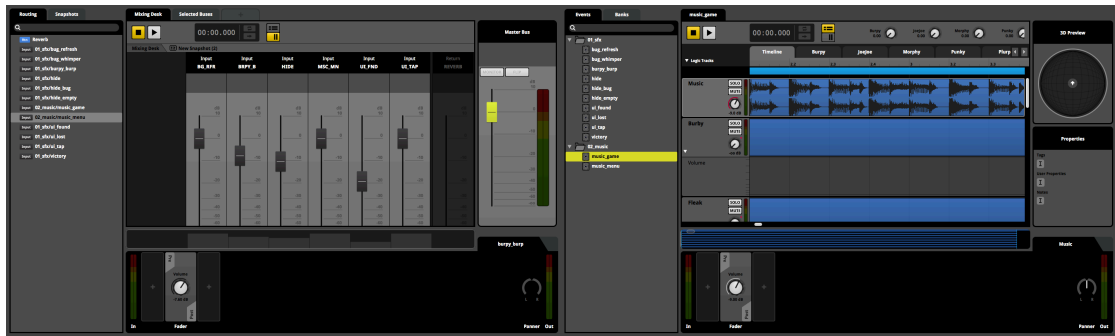
Reaaliaikaisten efektien käyttäminen vaatii kuitenkin paljon prosessointitehoa ja käy nopeasti liian raskaaksi suorituskyvyllä. Erityisesti mobiilipeleissä on tästä syystä suositeltavaa pitää efektien käyttö mahdollisimman vähäisenä. Kuten sanottu, masterkompressoinnista on kuitenkin hyötyä erityisesti mobiiliympäristössä, joten sen käyttäminen voi hyvin olla perusteltua. Äänisuunnittelijan on syytä myös varautua perustelevaan eniten prosessoria kuormittavia valintoihin muulle pelinkehitystiimille, koska etenkin projektin loppuvaiheessa niitä voidaan kyseenalaistaa, kun yritetään optimoida pelille parasta mahdollista suorituskykyä.

Äänenvoimakkuuksia, panorointeja ja efektejä voidaan ääniväliohjelmiston avulla myös säätää pelin aikana, jolloin myös miksausella voi rakentaa adaptiivisuutta sävellyksen lisäksi (Horowitz, Looney 2014, 127). Esimerkiksi kaikua lisäämällä ja äänenvoimakkuutta pienentämällä voidaan hetkellisesti tehdä musiikista etäisemmän kuuloista. Vastaavasti kaikua vähentämällä ja nostamalla äänenvoimakkuutta runsaasti master-kompressoria vasten voidaan musiikki saada hetkellisesti hyvin läheisen kuuloiseksi.

Monet ääniväliohjelmistot mahdollistavat myös erilaiset *ducking-tekniikat* eli järjestelmät, joissa yhden raidan voimakkuutta ohjataan toisella raidalla, tarkoituksena yleensä tehdä tilaa tärkeille äänille. Tyypillisin keino tämän aikaansaamiseksi on niin kutsuttu *side chain -kompressointi*, jossa raidan kompressoinnin määrää ohjaa jokin toinen raita. Esimerkiksi usein ääninäyttelijöiden dialogi halutaan saada erottumaan selkeästi pelin muun äänimaiseman ja musiikin seasta. Tällöin dialogiraidalla voidaan ohjata muun äänimateriaalin kompressointia, jolloin dialogia toistettaessa muu äänimateriaali vaimenee halutun verran ja dialogin loppuessa palaa täyteen voimakkuuteensa. (Collins 2008a, 103–104.) Ducking-tekniikkaa voi hyvin käyttää myös adaptiivisen musiikin miksausessa, esimerkiksi tärkeiden motiivien erottumisen varmistamiseksi.

6.3 FMOD Studio

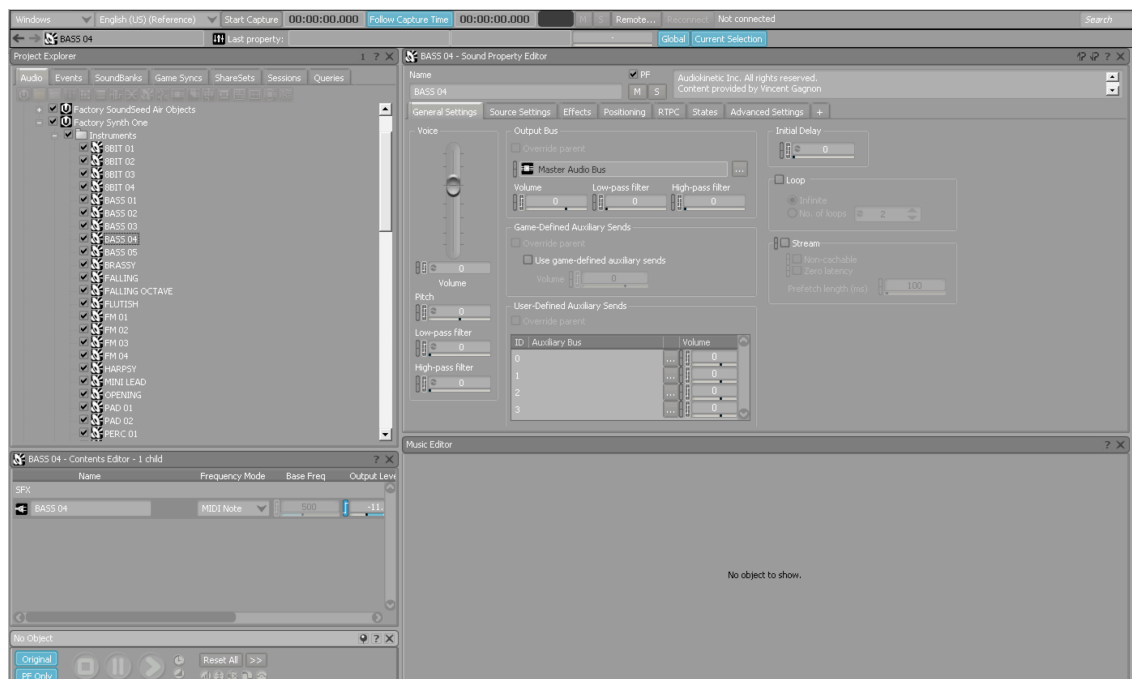
Fireflight Technologies julkaisi ensimmäisen version FMOD-ääniväliohjelmistostaan vuonna 2002. Ohjelmisto kasvatti nopeasti suosiotaan ja oli edelläkävijä DAW-ohjelmistoista tuttujen äänenmuokkaus- ja säätömahdollisuuksien sekä graafisen käyttöliittymän yhdistämisessä ääniväliohjelmistoihin. Vuonna 2013 ohjelmistosta julkaistiin uusi versio FMOD Studio, jossa muun muassa miksausominaisuudet ovat hyvin pitkälle vietyjä. (Horowitz, Looney 2014, 133.) FMOD Studio on tällä hetkellä alan johtavia ääniväliohjelmistoja ja lähinnä DAW-ohjelmistojen logiikkaa, minkä takia havainnollistan tämän työn esimerkkejä sen avulla. Sen voi myös ladata ilmaiseksi internetistä, ja pienistä projekteista ei tarvitse maksaa julkaisuvaiheessakaan mitään lisensointimaksuja. Tyypillinen pienten, itsenäisten peliyhtiöiden käyttämä yhdistelmä onkin FMOD Studio Unity-pelimoottorin kanssa. Lisensointimaksuttomuuden lisäksi yhdistelmä on luonteva, koska Unity käyttää äänen hallintaan ominaisuuksiltaan karsittua versiota FMOD:in äänimoottorista joka tapauksessa. Ohjelmistojen integraatio on siis luonnostaan hyvin toimiva.



Kuvio 6. Fireflight Technologies: FMOD Studio

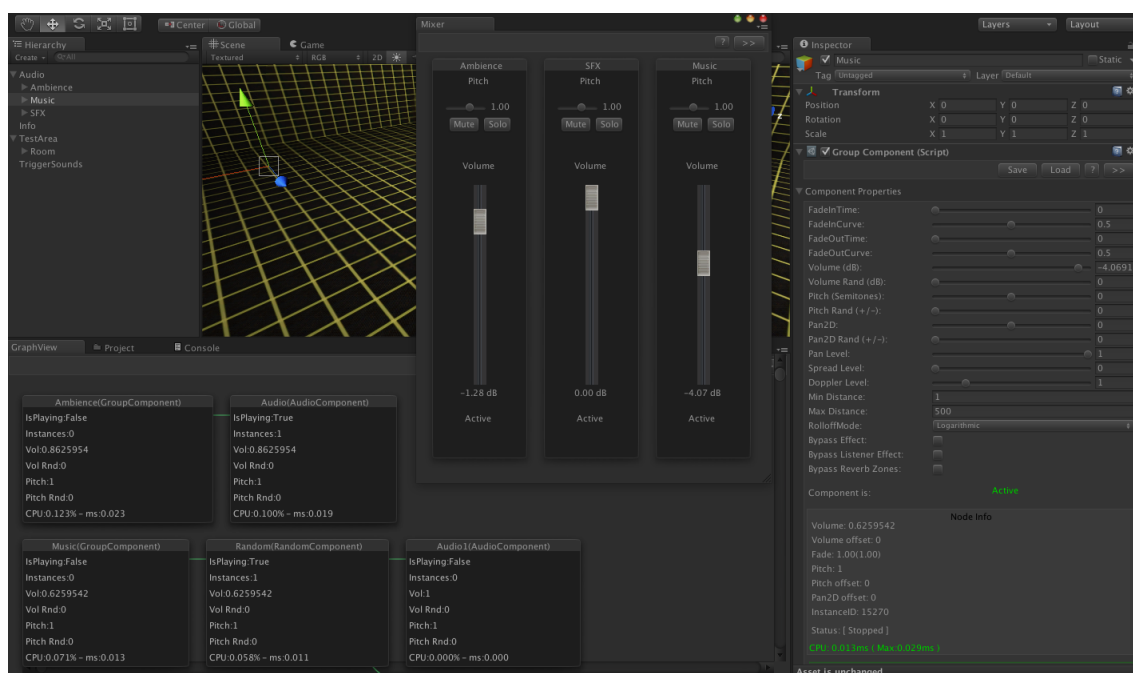
6.4 Muita vaihtoehtoja

Audiokineticin Wwise julkaistiin 2006. Se on ominaisuuksiltaan hyvin monipuolinen ja siksi käyttöliittymältään huomattavasti FMOD Studiota sekavampi ja vaatii siksi enemmän aikaa opetteluun. Tärkeimpiä ominaisuuksia ovat modulaarinen tapahtumajärjestelmä, joka mahdollistaa hyvin monimutkaisten kokonaisuuksien rakentamisen pienistä ja yksinkertaisista elementeistä sekä Interactive Music Engine, joka tarjoaa työkaluja adaptiivisen musiikin rakentamiseen. (Horowitz, Looney 2014, 134.)



Kuvio 7. Audiokinetic: Wwise

Osa ääniväliohjelmistoista on vähemmän DAW-tyyppisiä ja enemmänkin pieniä lisäosia pelimoottoriin. Esimerkiksi paljon mobiilipeleissä käytetty Tazman Audion kehittämä Fabric on äänilisäosa Unity-pelimoottoriin. Se ei siis aukea omana erillisenä ohjelmistonaan, kuten FMOD Studio ja Wwise, vaan lisää ääniominaisuusvalikoita ja työkaluja Unityn sisälle. (Horowitz, Looney 2014, 135.) Ominaisuuksiltaan Fabric on jokseenkin vajaampi kuin esimerkiksi FMOD ja käyttöliittymältään kömpelömpi, mutta suurin osa ydinominaisuuksista, kuten päällekkäiset raidat, tapahtumat ja parametrit ovat yhtälailla saavutettavissa ilman ohjelmointia. Ohjelmoinnin perusteet tunteva äänisuunnittelija pystyy kuitenkin pienellä opettelulla laajentamaan Fabric:in ominaisuuksia helposti ja toisaalta hallitsemaan esimerkiksi muistinkäyttöä vapaammin kuin DAW-tyyppisissä vaihtoehdoissa. Silti, erityisesti adaptiivisen musiikin rakentamisessa Fabric ei ole läheskään yhtä intuitiivinen ja nopeakäyttöinen kuin FMOD.



Kuvio 8. Tazman Audio: Fabric Unity-pelimoottorin sisällä

7 Tekniikoita adaptiivisuuteen

Lähtökohtana kaikelle adaptiiviselle musiikille on, että musiikki koostuu useasta eri osasta. Osia voi muodostaa eri tavoin, ja näistä tavoista voidaan puhua omina metodeinaan, joihin liittyy omat hyötynsä ja haittansa. Musiikista voidaan myös luoda adaptiivista, vaikkei sitä olisi tehty erityisesti adaptiivisuutta varten, esimerkiksi leikkaamalla kokonaisia kappaleita osiksi jälkikäteen. Tämä tapa voi toimia hyvin, riippuen paljolti kappaleista, mutta on usein hyvin ongelmallinen. Parhaat tulokset syntyvät huolellisella suunnittelulla, joka ottaa adaptiivisuuden huomioon jo sävellyksen alkumetreillä.

Käytännössä adaptiivista musiikkia voi lähestyä kahdella tavalla, horisontaalisesti tai vertikaalisesti. Pienissä, yksinkertaisissa peleissä voi saada hyviä tuloksia käyttämällä vain yhtä tapaa, mutta useasti adaptiivisuuden rakentamisessa käytetään molempia tapoja ja muodostetaan niistä erilaisia ja uniikkeja systeemejä. Muitakin tapoja on, mutta niitä käytetään harvemmin, ja ne ovat enemmänkin kuriositeetteja tyypilliselle pelisäveltäjälle.

7.1 Horisontaalinen resekvensointi

Musiikissa ajan kulumisen hahmotetaan yleensä horisontaalisesti, vasemmalta oikealle. Näin käyttäytyy myös nuottikirjoitus ja useimmat audio-ohjelmistot; tahdit ja aaltomuodot liikkuvat vaakasuunnassa. Osien uudelleenjärjestäminen vaakasuunnassa eli horisontaalinen resekvensointi on kautta pelimusiikin historian ollut, ja on edelleen, hyvin yleinen tapa rakentaa adaptiivisuutta. Yksinkertaisimmillaan se on teknisesti ja sävellyksellisesti varsin helppo toteuttaa, mutta vaatii taitavaa suunnittelua onnistuakseen hyvin ja antaa mahdollisuudet erittäin monipuolisten järjestelmien rakentamiseen. Käytännössä kappaleet jaetaan osiin ja pelitapahtumien mukaan valitaan aina seuraavaksi soiva osa. Peruseriaate on luoda osia niin, että ne voivat luontevasti seurata toisiaan missä tahansa järjestyksessä. (Philips 2014, 188; Marks 2001, 192.)

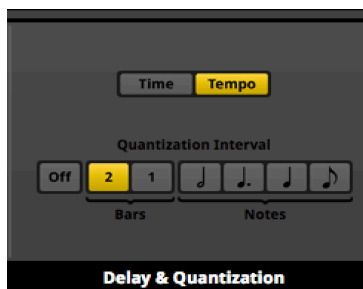
Esimerkiksi tyypillistä pop-kappaletta voisi käyttää pelissä horisontaalisen resekvensoinnin keinoin: ensin rauhallisessa pelitilanteessa loopataan kappaleen säkeistöä. Pelaajan löytäessä jotain uutta siirrytään kertosaakeeseen, jonka loputtua palataan taas looppaamaan säkeistöä, kunnes pelaaja taas löytää jotain uutta. Pelaajan kohdatessa vihollisen voitaisiin siirtyä kappaleen C-osaan, jota loopataan niin kauan kunnes

taistelu on ohi, jolloin siirrytään taas looppaamaan säkeistöä. Näin rakentuu yhden pelikentän musiikki ja seuraavaan kenttään siirryttäessä mekanismi on sama, mutta kappale vaihtuu.

Valmiin pop-kappaleen käyttäminen tällä tavalla kuulostaisi todennäköisesti kuitenkin hyvin kömpelöltä. Pop-kappale, kuten kaikki musiikkiäänitteet, on lineaarinen taide-muoto, eikä siksi ole tietenkään suunniteltu epälineaariseen käyttötarkoitukseen. Toki säkeistöä seuraa kertosäe ja kertosäettä säkeistö, mutta esimerkiksi saman säkeistön looppaamista tai esimerkiksi C-osasta säkeistöön siirtymistä ei ole äänitettä luodessa tarvinnut miettiä. Horisontaalisesti resekvensoitavaksi tarkoitettu kappale koostuu osista, jotka on suunniteltu niin, että kaikki osat sopivat toistensa jatkeeksi ja siirtymät ovat luonnollisia. Lähtökohtaisesti myös sanoitettu musiikki soveltuu tähän käyttötarkoitukseen huomattavasti paremmin ja onkin peleissä harvinaista, vain joissakin erikoistilanteissa käytettyä.

7.1.1 Siirtymät liukuen tai tempossa

Eri osien välillä voidaan liikkua eri tavoilla, karkeasti joko liukuen tai tempossa iskulta. Hyvin yksinkertainen ja käytetty tapa siirtyä osasta toiseen on ristiinhäivyttää (engl. crossfade) osat keskenään eli häivyttää uusi osa sisään (engl. fade in) edellisen osan häivytyessä ulos (engl. fade out). Tapa sopii erityisesti musiikkiin, jossa rytmisen pulssi ei ole suuressa roolissa, eli esimerkiksi maalaileviin ambient-tyyppisiin kappaleisiin. Pelitapahtuma voi tällöin laukaista uuden osan heti tapahtuessaan ja ristiinhävytys alkaa saman tien. Tällaisen siirtymän implementoiminen on yksinkertaista, koska pelimoottorin ei tarvitse ottaa tempoa huomioon, eikä äänisuunnittelijan tarvitse määrittää siirtymälle mitään ehtoja. Tunnistettavan pulssin sisältävä materiaali soveltuu kuitenkin yleensä huonosti tällaiseen siirtymään, koska käytännössä joka siirtymällä pulssi sekoittuu.



Kuvio 9. Siirtymäkohdat asetettu kahden tahdin välein FMOD-ääniväliohjelmistossa

Toinen vaihtoehto on siirtyä uuteen osaan tempossa ennalta määritetyltä iskulta. Säveltäjä voisi esimerkiksi määrittää, että osasta voidaan siirtyä toiseen osaan joka toisen tahdin ykköseltä. Kun pelitapahtuma laukaisee uuden osan, ääniväliohjelmisto odottaa seuraavaan sallittuun siirtymäkohtaan asti, käynnistää uuden osan ja pysäyttää edellisen. Käytännössä ääniväliohjelmistot tarjoavat tämänkaltaisten siirtymäehtojen toteuttamiseen useita vaihtoehtoja. Tyypillisesti on mahdollista määrittää musiikkipätkille tempot ja tahtilajit ja tämän jälkeen määrittää halutut välit siirtymille, esimerkiksi juuri aina kahden tahdin välein. (Horowitz, Looney 2014, 100–101.) Tämä tapa toimii parhaiten pätkiin, joissa rytmiset fraasit ovat koko ajan yhtä pitkiä eli pätkä rakentuu ikään kuin useasta samanmittaisesta fraasista. Sen sijaan jos pätkässä on esimerkiksi sekä kahden että kolmen tahdin mittaisia fraaseja, tapa ei enää toimi halutulla tavalla.

Siirtymäkohdat voidaan merkata myös jokainen erikseen pätkään asetettavilla merkeillä, niin sanotuilla markereilla (Horowitz, Looney 2014, 100–101). Tällä tavalla voidaan varmistaa, että siirtymä tapahtuu aina luonnollisessa kohdassa eli fraasien välissä. Pelitapahtuman laukaistessa siirtymän ääniväliohjelmisto odottaa, kunnes se havaitsee seuraavan markerin, jolloin siirtymä toteutetaan. Markereita voidaan sijoittaa niin tiheään tai harvaan kuin on tarve, yleensä juuri fraasien mukaan.



Kuvio 10. Markereilla merkatut siirtymäkohdat FMOD-ääniväliohjelmistossa

7.1.2 Fraasit

Fraasien pituus onkin horisontaalisessa resekvensoinnissa hyvin oleellinen seikka. Liian pitkiä fraaseja on syytä välttää, jotta pelitilanteisiin reagoiminen pysyy tarpeeksi nopeana. Toisaalta pelkkien lyhyiden fraasien käyttäminen voi käydä helposti yksitoikkoiseksi. Hyvä keino on rakentaa pitempiä fraaseja ikään kuin lyhyistä osista niin, että fraasi toimii hyvin kokonaisuutena, mutta sen keskeltäkin voidaan siirtyä luontevasti muihin osiin siirtymän kuulostamatta kömpelöltä.

Fraasiajattelussa on syytä ottaa huomioon melodian lisäksi myös kaikki harmonia ja rytmilliset elementit ja rakentaa ne loppumaan luontevasti siirtymäkohtiin. Esimerkiksi jousisektiolla tai syntetisoidulla padilla toteutettujen pitkien sointujen tulisi vaimentua mahdollisen siirtymäkohdan lähestyessä, koska niitä ei välttämättä ole enää seuraavassa osassa, jolloin ne saumaan asti soidessaan loppuisivat töksähtäen seinään. Toisaalta, vaikka seuraavassa osassa olisikin sama jousisektio, se saattaisi soittaa eri sointua tai käyttää erilaista artikulaatiota, jolloin leikkaus olisi joka tapauksessa töksähtävä. Pitkät sustain- ja legato-tyyppiset elementit ovatkin hankalia horisontaalisessa resekvenssoinnissa ja helpommalla parempaan lopputulokseen pääsee suosimalla harmonian muodostuksessa esimerkiksi arpeggioita ja staccato-tyyppisiä sointuja.

7.1.3 Stingerit

Siirtymiä pehmenemään voidaan käyttää niin sanottuja *stingereitä* eli lyhyitä musiikkipätkiä, jotka johdattelevat seuraavana osaan. Ideana on siis siirtymän tapahtuessa toistaa ensin stingeri ja sen jälkeen uusi osa. Stingerin käyttäminen on suositeltavaa erityisesti, jos osat ovat keskenään hyvin erilaisia. Esimerkiksi siirryttäessä hitaammas- ta osasta nopeampaan äkillinen siirtymä saattaa kuulostaa erikoiselta, mutta jos uuteen osaan siirrytään asteittain nopeutuvan stingerin kautta, lopputulos on luontevampi. (Horowitz, Looney 2014, 100.)

Stinger-termillä voidaan tarkoittaa myös lyhyitä musiikkipätkiä, joita käytetään johonkin muuhun tarkoitukseen kuin siirtymän pehmentämiseen. Tyypillisesti *stingereitä* käytetään ikään kuin äänitehosteita eli tietyt pelitapahtumat, esimerkiksi räjähdys, voivat laukaista tietyn stingerin. Tällöin stinger voidaan joko toistaa soivan musiikin päälle tai siihen voidaan siirtyä kuten mihin tahansa muuhun osaan, jonka jälkeen siirrytään takaisin stingeriä edeltäneeseen osaan. Stingereiden tulisi ihannetilanteessa joka tapauksessa olla tarpeeksi samankaltaisia muun musiikkimateriaalin kanssa, että pelaaja mieltäisi ne luonnollisiksi osaksi pelin musiikkia. (Philips 2014, 177–181.)

7.2 Vertikaalinen reorkestraatio

7.2.1 Työskentely stemmojen kanssa

Musiikin teoriassa ja musiikkiteknologiassa kokonaisuuden muodostavat samanaikaiset elementit hahmotetaan yleensä vertikaalisesti, pystysuunnassa päällekkäisinä. Esimerkiksi partituurissa eri instrumenttien osuudet ovat omilla viivastoillaan allekkain ja tyypillisissä DAW:eissa raidat järjestyvät samalla tavalla. Vertikaalinen reorkestraatio tai kerrostaminen tarkoittaa adaptiivisuuden rakentamista yhtä aikaa soivien raitojen määrää tai suhdetta muuttamalla. Käytännössä musiikki on tällöin osoitettu erilaisiksi stemmoiksi, jotka laukaistaan samaan aikaan ja joiden äänenvoimakkuutta ja muita parametreja muutetaan pelitapahtumien mukaan. (Horowitz, Looney 2014, 100.)

Yksinkertaisimmillaan kappaleesta voitaisiin esimerkiksi erottaa rummut omaksi stemmakseen ja loput soittimista omakseen. Pelitilanteen ollessa rauhallinen, rummut ovat mutella⁹ ja muut soittimet soivat normaalisti. Taistelutilanteessa rummut nostetaan vähitellen mukaan, jolloin musiikin tunnelma muuttuu kiihkeämmäksi. Kun taistelu on ohi, rummut vaimenevat ja vaikutelma on taas rauhallisempi. Käytännössä tunnelman muutos haluttaisiin todennäköisesti dramaattisemmaksi, jolloin mukaan voitaisiin nostaa muitakin elementtejä kuten hakkaavia staccatojousia ja särökitaraa. Taistelusta voitaisiin erottaa myös jonkinlainen huippukohta, esimerkiksi vihollisen joutuminen alakynteen, jolloin sekaan lisättäisiin vielä voimakkaita patarummun iskuja.

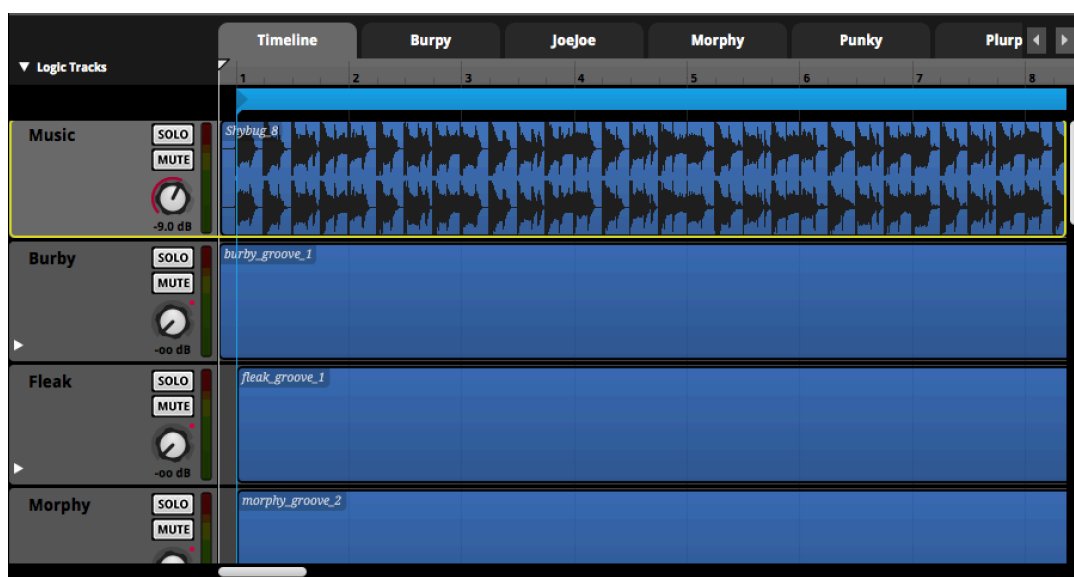
Useampien raitojen kanssa työskennellessä täytyy olla tarkkana muistinkäytön, prosessorin kuormituksen ja tiedostokokojen kanssa. Mitä useampi samanaikainen raita, sitä enemmän musiikki vie levytilaa ja rasittaa suorituskykyä. Tämä onkin vertikaalisen reorkestroinnin suurimpia ongelmia etenkin mobiilialustoilla. Lisäksi, koska yli kahden raidan samanaikainen streamaus ei ole suorituskyvyn kannalta mobiililaitteilla suositeltavaa, raitojen pituus ja pakkaus täytyy suunnitella huolellisesti. Yleensä ambienssiraita myös streamataan, jolloin musiikissa voidaan käyttää enää vain yhtä streamattua raitaa. Käytännössä tämä tarkoittaa, että vain yksi raita voi olla pidempi, ja muut täytyy rakentaa lyhyistä loopeista. Tällöin täytyy suunnitella, mitkä elementit rakennetaan lyhyemmistä osista ja mille varataan pidempi raita.

⁹ Engl. *mute* eli vaimennettu. Musiikkiteknologiassa käsitteellä tarkoitetaan ääniraidan asettamista täysin äänettömäksi.

7.2.2 Additiivisuus ja vaihtoehtoisuus

Stemmat voivat olla keskenään joko additiivisia tai vaihtoehtoisia (Philips 2014, 195). Edellisessä esimerkissä kaikki stemmat ovat additiivisia, eli niitä voidaan soittaa täydellä äänenvoimakkuudella samanaikaisesti. Lisäksi voitaisiin käyttää joitakin vaihtoehtoisia stemmoja. Esimerkiksi jos harmoniasoitimet ovat omana stemmana, niille voisi olla vaihtoehto, jossa kaikki ensimmäisen asteen soinnut ovatkin duurin sijasta mollisointuja. Pelitilanteen mukaan käytetään jompaakumpaa stemmaa ja siirtymät voidaan ristiävyttää keskenään. Melodian pysyessä samana lopputulos on tunnelmaltaan hyvin erilainen, mutta silti tuttu ja selkeästi samassa kontekstissa.

Toimiakseen tämä edellyttää eri harmoniavaihtoehtojen tiedostamisen melodian sävellysvaiheessa. Tämänkaltainen suunnittelu onkin ensiarvoisen tärkeää vertikaalista kerrostamista käytettäessä. Melodia voisi tietysti myös reagoida muutokseen, jolloin rummut olisivat ainoa samanlaisena jatkuva elementti. Tällöin lopputulos olisi kuitenkin hyvin samankaltainen horisontaalisen re-sekvenssoinnin kanssa, ja tähän käyttö-tarkoitukseen valittua metodia kannattaisi ehkä harkita uudelleen.



Kuvio 11. Vertikaalista reorkestraatiota FMOD-ääniväliohjelmistossa

Yllä olevassa kuvassa näkyy erääseen mobiilipeliin tekemäni musiikin FMOD-sessio. Hyödynsin pelissä hyvin yksinkertaista vertikaalista reorkestraatiota. Kyseessä oli lapsille suunnattu peli, jossa etsitään piiloistaan ötököitä, jotka löydyttyään hyppäävät

pelikentälle tanssimaan musiikin tahtiin. Ötököitä saattoi myös menettää takaisin piiloihinsa tekemällä virheitä. Tein yhdelle raidalle varsinaisen kappaleen ja sen lisäksi jokaiselle kahdeksalle ötökälle oman kahden tahdin mittaisen rytmimotiivin omille raitoilleen loopattuna läpi kappaleen. Jokaisen ötökkäraidan äänenvoimakkuutta ohjasi ötökän niminen parametri, jolle pelimoottori lähetti arvon 0, jos ötökkä oli vielä piilossa, ja arvon 1, jos ötökkä oli löytynyt ja tanssimassa. Arvo 0 merkitsi täysin vaimennettua raitaa ja arvo 1 täyttä äänenvoimakkuutta. Raitojen äänenvoimakkuus vaihtui pelitilanteen mukaan parin sekunnin häivytyksellä, jotta muutokset eivät kuulostaisi töksähtävilä. Näin pelissä soi aina ruudulla näkyvien ötököiden mukaiset rytmimotiivit muun musiikin seassa.

7.3 Sattumanvaraisuus

Yksinkertaisimmillaan molempia tähän mennessä käsiteltyjä tapoja ohjaa yksinomaan pelitapahtumat. Tietty tapahtuma laukaisee tietyn osan, seuraava taas laukaisee jonkin toisen ja niin edelleen. Tällä tavoin musiikki reagoi peliin ja pysyy siksi mielenkiintoisena, eikä pelkästään toista itseään. Tämä on kuitenkin vasta pintaraapaisu adaptiivisuuteen. Ongelmaksi muodostuvat erityisesti tilanteet, joissa pelaaja ei pitkään aikaan osallistu pelitapahtumiin, jotka laukaisisivat muutoksia musiikissa. Tällöin musiikki pelkästään toistaa itseään ja voi käydä nopeasti rasittavaksi. (Collins 2008a, 140–141.)

Toistavuuden ehkäisemiseksi pelien äänisuunnittelussa käytetään usein sattumanvaraisuutta eri tavoin hyödyksi. Äänitehosteiden kanssa esimerkiksi on tyypillistä käyttää sattumanvaraisesti pienellä alueella vaihtelevaa sävelkorkeutta. Äänitehosteissa esimerkiksi sävelaskeleen sisällä tapahtuvat sattumanvaraiset vaihtelut kuulostavat vielä luonnollisilta, eikä ääni tunnu toistavan itseään yhtä selkeästi. Luonnollisesti tämä tapa ei toimi musiikin kanssa, koska sävelkorkeus on siinä tärkeämmässä roolissa. Toinen yleinen tapa käyttää sattumanvaraisuutta on yksinkertaisesti arpoa useamman vaihtoehdoisen äänitehosteen kesken joka toistokerralla. Samanlaista lähestymistapaa käytetään usein myös adaptiivisissa järjestelmissä. (Collins 2008a, 32–35, 162, 169.)

Yksinkertainen tapa rakentaa sattumanvaraisuudella vaihtelua loopattaviin osiin on tuottaa jokaisesta osasta yksi tai useampi vaihtoehtoinen versio, esimerkiksi hieman erilaisella melodialla tai soitinnuksella. Aina kun osan olisi määrä alkaa alusta, järjestelmä arpoo, mikä vaihtoehdoista toistetaan. Kun jonkin pelitapahtuman laukaisemana

siirrytään seuraavaan osaan, sama prosessi toistuu senkin kanssa. Näin musiikki elää jossain määrin yllätyksellisesti koko ajan eikä käy yhtä helposti pitkästytäväksi. Useimmat ääniväliohjelmistot myös tekevät tämän toiminnallisuuden hyvin helpoksi toteuttaa.

7.4 Muita tekniikoita

Edellä esitellyt tekniikat perustuvat ennen kaikkea valmiisiin fraaseihin, joita eri tavoin yhdistelemällä saavutetaan adaptiivisuutta. Tämänkaltainen lähestymistapa on toistaiseksi vallitseva pelimusiikissa, ja useimpien pelien musiikkijärjestelmät voidaan nähdä erilaisina yhdistelminä horisontaalista resekvensointia ja vertikaalista reorkestraatiota. Toinen, toistaiseksi harvinaisempi lähestymistapa on rakentaa järjestelmä, joka perustuu fraaseja pienempiin yksikköihin, jopa yksittäisiin ääniin. Tällöin musiikin koostamiseen tarvitaan huomattavasti enemmän sääntöjä ja ohjelmointia, eikä prosessi ole enää kovinkaan lähellä tyyppillisiä musiikin tuottamisen työtapoja. Tämänkaltainen lähtökohta avaa monenlaisia uusia mahdollisuuksia adaptiivisuuden luomiseksi, ja siihen pohjaavia tekniikoita tullaan todennäköisesti käyttämään ja analysoimaan tulevaisuudessa enemmän. Tähän asti näistä tekniikoista on kirjallisuudessa tunnistettu ainakin yksi, generatiivinen musiikki.

7.4.1 Generatiivinen musiikki

Pidän hyvin mahdollisena, että lapsenlapsemme katsovat meitä ihmeissään ja kysyvät: "Tarkoitatko, että kuuntelitte täsmälleen samaa asiaa uudelleen ja uudelleen?"

-Brian Eno (Jones 2003, 187.)

Generatiivinen musiikki on termi, jonka Brian Eno teki tunnetuksi *Generative Music 1* -teoksellaan, joka julkaistiin vuonna 1996 levykkeellä PC ja Macintosh -tietokoneille. Termillä tarkoitetaan musiikkia, jonka jokin järjestelmä tuottaa sattumanvaraisesti reaaliajassa pienistä palasista määrättyjen sääntöjen pohjalta – siis ikään kuin tietokoneen improvisoimaa musiikkia. Tällä tavoin luotu musiikkiraita on joka kerralla erilainen ja tarjoaa videopeliympäristössä erinomaisen vaihtoehdon loppaamiselle. Sääntöjen muuttaminen pelitilanteiden mukaan taas avaa aivan uudenlaisia mahdollisuuksia adaptiiviselle musiikille. (Intermorphic 2015.)

Videopeleissä generatiivista musiikkia ei vielä ole käytetty kuin harvakseltaan, tunnetuimpana esimerkkinä vuonna 2008 julkaistu *Spore*, jonka musiikkijärjestelmää suunnittelemassa oli niin ikään Brian Eno. Pelissä pelaaja ohjaa uuden eliön kehityskaarta laajassa, avoimessa pelimaailmassa, jossa etenemismahdollisuudet ovat käytännössä rajattomat. Musiikin säännöt muuttuvat eliötä kehitettäessä, ja esimerkiksi taisteluominaisuuksien kuten terävien kynsien lisääminen vie musiikkia mollin suuntaan luoden pahaenteisempää tunnelmaa. (Kosak 2008; Philips 2014, 213.)

Käytännössä voidaan ajatella, että generatiivinen musiikki hyödyntää sekä vertikaalista reorkestraatiota että horisontaalista resekvenssointia, mutta käytettävät osat ovat hyvin yksinkertaisia ja lyhyitä, ja niitä on todella paljon. Lisäksi voimakas sattumanvaraisuus ja monimutkaisten sääntöjen tarve erottavat sen selkeästi omaksi kategoriakseen. Tapa soveltuu erityisen hyvin alati muuttuvan taustamusiikin tuottamiseen, mutta tunnistettavia ja mieleenpainuvia melodioita sillä ei ole tarkoituskaan luoda.

Generatiivinen musiikki kuormittaa aikaisemmin mainittuihin tapoihin verrattuna enemmän prosessoria ja muistia eli vaatii enemmän laitteistolta. Musiikin rakennuspalasia on huomattavasti enemmän, ja pelimoottorin täytyy toistaa useampia elementtejä samanaikaisesti sekä huolehtia sääntöjen toteutumisesta ja vaihtumisesta. Tämän takia generatiivisen musiikin hyödyntäminen peleissä on varsin tuore ilmiö. Tapa vaatii myös paljon enemmän työtä ohjelmoijilta ja säveltäjiltä, koska käytännössä täytyy rakentaa kokonainen eheää musiikkia tuottava järjestelmä muutamien kappaleiden ja osien sijaan. Saavutettu vaihtelu ja ennalta-arvaamattomuus on kuitenkin muihin tapoihin verrattuna niin ylivoimaista, että uskon tavan tulevan suosittumaksi jatkossa.

8 Pohdinta

Tutkin ja määrittelin työssäni videopelimusiikin erityispiirteitä, joista keskeisiksi nousivat epälineaarisuus ja adaptiivisuus. Vaikka musiikin säveltämisen ja tuottamisen perustyökalut ovat pelimusiikkia tehdessä samat kuin minkä tahansa musiikin kanssa, nämä erityispiirteet aiheuttavat erilaisia haasteita ja ongelmia, mutta toisaalta tarjoavat runsaasti uudenlaisia mahdollisuuksia. Vaikeimpien haasteiden joukossa on looppaavan musiikin itsensä toistavuuden aiheuttama rasittavuus. Muun muassa tätä varten nykyään peleissä käytetään paljon erilaista adaptiivista musiikkia. Erityisesti keskityinkin

työssäni adaptiivisen musiikin tuottamiseen eli siihen, miten musiikki voi käytännössä reagoida pelitilanteisiin niiden tapahtuessa ja näin pysyä mielenkiintoisena ja tiiviinäkin osana pelimekaniikkaa.

Videopelimusiikin historian ja nykyisen tilanteen käsittelyssä kävi ilmi, että mobiilipelien suosion noustua ollaan ikään kuin menty ajassa taaksepäin teknisten rajoitusten suhteen. Tietokone- ja konsolipelit ovat kehittyneet niin pitkälle, etteivät laitteiden suorituskykyyn liittyvät ongelmat juurikaan hankaloita äänisuunnittelijan tai säveltäjän työtä. Sen sijaan mobiililaitteilla rajoitukset ovat ilmeiset, ja esimerkiksi rajallisen suorituskyvyn mahdollisimman tehokkaaseen hyödyntämiseen pyrkiminen on mobiilipeleihin ääntä tuottavalle arkipäivää. Myös laitteiden äänentoiston laatu aiheuttaa ongelmia ja erityisesti taajuusvasteeseen ja stereokuvaan täytyy kiinnittää huomiota, ja niitä käsitellään usein eri tavoin kuin esimerkiksi äänitetuotannossa.

Työni tavoitteena oli tarjota videopelimusiikin tekemiseen hyvä pohja sen erityispiirteisiin ennestään tutustumattomalle lukijalle. Viime vuosien teknisten innovaatioiden myötä adaptiivisen pelimusiikin tekeminen on monella tavalla yksinkertaisempaa kuin ennen, ja käytännössä kenellä tahansa hieman teknologisesti orientoituneella musiikin tekijällä on kaikki edellytykset toimia pelimusiikin tuottajana.

Käsittelin nykyaikaisia pelimoottoreita ja ääniväliohjelmistoja, jotka mahdollistavat äänisuunnittelijan itsenäisen työskentelyn ilman, että tarvitsee esimerkiksi osata ohjelmoida. Käytännön työssä näillä työkaluilla pärjää nykyään hyvin, vaikka ohjelmointiosaamisesta on luonnollisesti myös hyötyä alalla. Valitsin käsiteltäväkseni FMOD-ääniväliohjelmiston, koska se muistuttaa paljon musiikkialalla käytettyjä DAW-ohjelmistoja ja on siksi musiikkia tietokoneella tuottavalle hyvin nopea oppia. Mielestäni ääniväliohjelmistojen ja pelimoottorien esittely onnistui hyvin, mutta pelimusiikkia tosiaan tekemään alkavan täytyy luonnollisesti tutkia asiaa vielä syvällisemmin ja harjoitella näiden ohjelmistojen käyttöä.

Lopuksi käsittelin yleisimpiä tekniikoita adaptiivisen musiikin luomiseksi. Esittelin horisontaalista resekvenssointia ja vertikaalista reorkestraatiota sekä niiden etuja ja ongelmia. Usein peleissä käytetään molempia tapoja sekaisin mahdollisimman monipuolisen ja kiinnostavan äänimaiseman luomiseksi. Käsittelin lyhyesti vielä generatiivista musiikkia, jota ei toistaiseksi ole käytetty muihin tapoihin verrattuna peleissä kuin

harvakseltaan, mutta joka voi hyvin tulevaisuudessa teknologian kehittyessä nousta suosituksikin vaihtoehdoksi.

Työtä kirjoittaessa yllätyin laadukkaan kirjallisen lähdetiedon määrästä. Oletin kirjoitus-työhön lähtiessäni, että käyttökelpoisia lähteitä olisi vaikea löytää tai niitä ei löytyisi ollenkaan. Lopulta huomasin, että aiheesta on kirjoitettu useita asiantuntevia teoksia, joskin suomenkielisiä lähteitä ei juuri löytynyt. Vaikka ala on vahvasti kansainvälinen ja englannin kieli on keskeisessä asemassa, kuvittelisin suomenkieliselle kirjallisuudelle olevan myös kysyntää. Itseäni alkoi työtä kirjoittaessani kiinnostaa myös laajemman kirjallisen työn tekeminen aiheesta, ehkä muutaman vuoden työkokemuksen kartuttua.

Luin siis useita mielenkiintoisia kirjoja työtä varten ja opin niistä runsaasti uusia asioita. Löysin myös useimpia väitteitäni tukemaan vahvistuksen jostain lähteestä. Kaiken kaikkiaan mielestäni valitsemani työmenetelmä, kirjallisuuden reflektointi oman kokemuksen pohjalta, palveli työn tarkoitusta hyvin, ja onnistuin myös kehittämään itseäni. Työ olisi voinut hyötyä muutamasta haastattelusta, mutta jouduin rajaamaan ne pois, jotta työ pysyisi mielekkään pituisena. Toisaalta käyttämässäni lähteissä oli runsaasti alan tunnetuimpien tekijöiden haastatteluja, joten uskon että asiantuntevia näkemyksiä oli työn kannalta lopulta tarpeeksi.

Mielestäni onnistuin alkuperäisessä alan oleellisen tiedon tiivistämisen tavoitteessa melko hyvin. Rajaaminen oli jokseenkin hankalaa, koska toisaalta joihinkin aiheisiin olisi mielellään pureutunut tarkemminkin, esimerkiksi adaptiivisuuden tekniikoiden esittelyyn. Halusin kuitenkin antaa kokonaiskuvan koko alasta ja siksi päädyin käsittelemään myös esimerkiksi mobiilipelien liiketoimintamalleja ja pelimoottoreita. Lopulta työn varsinaiseksi funktioksi muodostui juuri kokonaiskuvan antaminen, kiinnostuksen herättäminen ja sen osoittaminen, että pelimusiikki on pienen perehtymisen jälkeen vartenotettava vaihtoehto musiikintekijälle ja tulevaisuuden suuria työllistäjiä. Toivon, että työn lukemisen jälkeen aiheesta kiinnostuneet hakeutuisivat syventävien teosten ääreen ja alkaisivat kokeilemaan ja harjoittelemaan pelimusiikin tuottamista.

Jätin tietoisesti tulevaisuuden enemmän spekuloinnin pois tästä työstä, mutta sitä olisi mielenkiintoista harrastaa jatkossa tai lukea aiheesta kirjoitettua materiaalia. Teknologian kehittyessä muun muassa generatiivisen musiikin tarjoamat mahdollisuudet olisivat vartenotettava tutkimuskohde. Olisi kiinnostavaa lukea tai tutkia, onko todennäköistä, että jo lähitulevaisuudessa tullaan käyttämään enemmän

virtuaali-instrumentteja suoraan pelimoottorin tai ääniväliohjelmiston sisällä ja synte- tisoimaan musiikkia reaaliajassa. Tämä lähestymistapa saattaa ratkaista joitain adaptii- visen pelimusiikin ongelmakohtia, esimerkiksi ristiinhäivytyksen tarpeen ja pitkien soin- tujen suhteen. Toisaalta samaan aikaan varmasti muodostuu uusia, erilaisia ongelmia ja haasteita.

Jatkotutkimukselle otollisia aiheita olisivat myös adaptiivisuuden estetiikka ja tekniikoi- hin tarkemmin syventyminen. Myös ääniväliohjelmistojen toimintaa voisi tutkia, etenkin suomenkielistä tekstiä aiheesta löytyy varsin vähän. Jonkinlainen seikkaperäinen ja käytännönläheinen ääniväliohjelmisto-opas voisi olla mielekäs projekti ja hyödyllinen työkalu aloittelevalle pelimusiikin tuottajalle.

Pelialan kasvun ja pelimusiikin tarpeen lisääntymisen takia mielestäni musiikkialan kou- lutuksessa aihealuetta olisi syytä huomioida enemmän. Perinteisesti peliäänen kurse- ja tuntuu olevan lähinnä äänisuunniteluun painottuneilla linjoilla ja musiikkia opiskelevil- ta alan tarjoamat mahdollisuudet voivat mennä kokonaan ohi. Ymmärrän, ettei aihe kiinnosta kaikkia, mutta vähintään vapaavalintaisia kursseja olisi mielestäni syytä tarjo- ta kaikissa musiikkialan korkeakouluissa, etenkin musiikin säveltämistä ja tuottamista painottavilla linjoilla.

Lähteet

Kirjalliset

Alten, Stanley 2012. Recording and Producing Audio for Media. Boston, Massachusetts: Course Technology.

Appington 2013. How Many Mobile Users Play with Sound on? [Verkkodokumentti] Saatavuus <<http://blog.appington.com/2013/10/07/how-many-people-play-with-sound-on-73/>>

Collins, Karen 2008a. Game sound: an introduction to the history, theory, and practice of video game music and sound design. Cambridge, Massachusetts; London, England: The MIT Press.

Collins, Karen 2008b. From Pac-Man to pop music: interactive audio in games and new media. Burlington, Vermont: Ashgate Publishing.

Collins, Karen; Kapralos, Bill; Tessler, Holly 2014. The Oxford handbook of interactive audio. Oxford: Oxford University Press.

Guerin, Robert 2005. MIDI Power: The Comprehensive Guide (2nd Edition). Boston, Massachusetts: Course Technology.

Hahl, Kalle 2014. The Success of Free to Play Games and Possibilities of Audio Monetization. Tampere: Tampereen ammattikorkeakoulu.

Horowitz, Steve; Scott Looney 2014. The Essential Guide to Game Audio : The Theory and Practice of Sound for Games. Burlington, Massachusetts: Focal Press.

IFIXIT, iPad Air 2 Teardown. [Verkkodokumentti] Saatavuus <<https://www.ifixit.com/Teardown/iPad+Air+2+Teardown/30592>> (luettu 2.10.2015)

Intermorphic 2015. Generative Music. [Verkkodokumentti] Saatavuus <<http://www.intermorphic.com/sseyo/koan/generativemusic1/#gm1>> (luettu 15.10.2015)

Jones, Steve 2003. Encyclopedia of New Media: An Essential Reference to Communication and Technology. Thousand Oaks, California: Sage Publications.

Kosak, Dave 2008. The Beat Goes on: Dynamic Music in Spore. [Verkkodokumentti] Saatavuus <<http://pc.gamespy.com/pc/spore/853810p2.html>> (luettu 15.10.2015)

Marks, Aaron 2001. The Complete Guide to Game Audio. Lawrence, Kansas: CMP Books.

Marsal, Katie 2015. Apple increases size limit of App Store downloads to 4GB. [Verkkodokumentti] Saatavuus <<http://appleinsider.com/articles/15/02/12/apple-increases-size-limit-of-app-store-downloads-to-4gb>> (luettu 8.10.2015)

Moormann, Peter 2013. Music and game: perspectives on a popular alliance. Wiesbaden: Springer VS.

Olson, Scott; Hunter, John; Horgen, Ben; Goers, Kenny 2012. Professional Cross-Platform Mobile Development in C#. Indianapolis: John Wiley & Sons, Inc.

Philips, Winifred 2014. Composer's Guide to Game Music. Cambridge, Massachusetts: The MIT Press.

Playstation 4, valmistajan verkkosivut. [Verkkodokumentti] Saatavuus <<https://www.playstation.com/en-us/explore/ps4/techspecs>> (luettu 2.10.2015)

Scolastici, Claudio; Nolte, David 2013. Mobile Game Design Essentials A useful and detailed resource for designing games for mobile devices. Birmingham, Iso-Britannia: Packt Publishing.

Unity verkkomanuaali, Audio clip. [Verkkodokumentti] Saatavuus
<<http://docs.unity3d.com/Manual/class-AudioClip.html>> (luettu 12.10.2015)

Vankka, Esko 2014. Free to play games: professionals' perceptions. Tampere: Tampereen ammattikorkeakoulu.

Ward, Jeff 2008. What is a Game Engine? [Verkkodokumentti] Saatavuus
<http://www.gamecareerguide.com/features/529/what_is_a_game_.php>
(luettu 15.10.2015)

Kuviot

Kuvio 1. Arttu Aalto, 2015. Word-kaavio.

Kuvio 2. NetMarketShare, 2015. <<https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=9&qpcustomb=1>>

Kuvio 3. Rob Boudon, 2011.
<https://commons.wikimedia.org/wiki/File:Four_Arcade_Games.jpg>

Kuvio 4. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 5. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 6. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 7. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 8. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 9. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 10. Arttu Aalto, 2015. Ruutukaappaus.

Kuvio 11. Arttu Aalto, 2015. Ruutukaappaus.