



Visuaalisten rajojen yli – näkövammaisille omistetun videopelin suunnittelu ja toteutus

Henry Friman

Haaga-Helia ammattikorkeakoulu

Tradenomi

Opinnäytetyö

2023

Tiivistelmä

Tekijä Henry Friman
Tutkinto Tradenomi
Opinnäytetyön nimi Visuaalisten rajojen yli – näkövammaisille omistetun videopelin suunnittelu ja toteutus
Sivu- ja liitesivumäärä 36 + 24
<p>Opinnäytetyöni aiheena on suunnitella ja toteuttaa näkövammaiset henkilöt huomioon ottava videopeli. Koen pelini olevan onnistunut, jos se on toimintalogiikaltaan ja identiteetiltään selkeä, julkaistu jossain suositussa digitaalisessa kauppapaikassa, pelattavissa ilman näköaistin käyttöä, ja saanut mielellään positiivista palautetta ensisijaisesti näkövammaisilta tai -rajoitteisilta pelaajilta. Vastaan projektini tuotannon jälkeen kahteen tutkimuskysymykseeni: ”mitkä ominaisuudet ovat tärkeitä visuaalisista aistihavainnoista riippumattoman pelikokemuksen toteuttamisessa?” ja ”mitä väärinkäsityksiä ja virhetilanteita voi ilmetä ei-visuaalisessa pelissä?” Nämä linkittyvät varsinaiseen päätutkimuskysymykseeni: ”miten tarjota autenttinen pelikokemus näkövammaisille ja -rajoitteisille pelaajille?”</p> <p>Ennen peliprojektin varsinaista tuotantoa kartoitan näkövammaisille suunniteltujen tai näkövammaisten itsensä kehittämien videopelien historiaa, yrittäen löytää niille yhteisiä tekijöitä ja ominaisuuksia. Oleellista näkövammaisystävällisissä videopeleissä ovat muun muassa kolmiulotteinen äänimaailma, ruudunlukijajärjestelmä, pelaajan helpotettu ohjausjärjestelmä sekä kirkkaasti korostetut visuaaliset elementit. Tutustun myös videopelien saavutettavuusohjeistuksiin, joita noudatan pelini kehitysprosessin aikana.</p> <p>Lopputuotteena on Unity-pelimootorilla kehitetty videopeliprojekti ”The Metronome of Dissonance”, joka on äänipainotteinen, visuaalisista elementeistä riippumaton 3D-toimintaseikkailupeli. Julkaisen pelistä toistaiseksi vain ilmaisen demoversion. Toteutan pelini viiden eri vaiheen kautta. Alussa suunnittelen pelini yleisilmeen ja toimintalogiikan (suunnitteluvaihe), joiden pohjalta toteutan varhaisen version pelistä (prototyyppivaihe). Perustoiminnallisuuksien toteuduttua varmistan, että koko peli on pelattavissa virheettömästi (viimeistelyvaihe), minkä jälkeen valmistaudun pelini julkaisuun digitaalisessa palvelussa (julkaisuvaihe). Kun pelistä on tullut julkinen, analysoin siihen kohdistuvaa kritiikkiä (palautteenkeruuvaihe). Kuvaan tarkasti pelinkehitykseni prosessia eri valintoihin ja päätöksiin, ja vertaan valmista peliä henkilökohtaisiin tavoitteisiin sekä saavutettavuusohjeistuksiin.</p> <p>Projektin päätyttyä vastaan tutkimuskysymyksiini: visuaalisista aistihavainnoista riippumattoman pelikokemuksen toteuttamisessa on tärkeää pitää pelaaja tietoisena siitä, mitä pelissä tapahtuu, ja miten hän voi vuorovaikuttaa siinä. Ei-visuaalisessa pelissä väärinkäsityksiä ja virhetilanteita voivat aiheuttaa keskenään samankaltaiset ja kontekstiin sopimattomat äänitehosteet, interaktion puute pelaajan opastuksessa sekä pelin abstraktinomaisuus. Autenttisen pelikokemuksen tarjoaminen näkövammaisille ja -rajoitteisille pelaajille on kuitenkin mahdollista, kun pelinkehittäjä asettuu heidän asemaansa ja suunnittelee aputoiminnot varhaisessa vaiheessa osaksi pelin ydinlogiikkaa ja identiteettiä. Tutkimusta voi kehittää muun muassa tekemällä pelistä kokonaan visuaalisen, laajentamalla sen eri asetuksia, antamalla tuen ohjaimelle ja mikrofonille sekä lisäämällä kohdeyleisöön kuulo-, liikunta- ja muun vamman tai rajoituksen omaavat pelaajat.</p>
Asiasanat Saavutettavuus, näkövammaisuus, pelinkehitys, Unity, C#, ruudunlukija, äänipeli

Sisällys

1	Johdanto	1
2	Tietoperusta	4
2.1	Videopelien saavutettavuus ja näkövammaiset pelaajat	4
2.2	Videopelien saavutettavuusohjeistusten noudattaminen	6
3	Empiirinen osa.....	7
3.1	Projektin alkumääritelmät	7
3.2	Tuotannon eri vaiheet ja niiden päämäärät.....	8
3.3	Projektissa alustavasti käytettävien teknologioiden ja palveluiden esittely	8
3.3.1	Unity	8
3.3.2	Steam ja Itch.io	9
3.3.3	Blender	10
3.3.4	Narakeet	10
3.3.5	LingoJam Robot Voice Generator	10
3.3.6	Freesound.....	10
3.3.7	Pixabay	11
3.3.8	Incompetech	11
3.3.9	HitFilm	11
3.3.10	Audacity	12
3.3.11	Gimp	12
4	Pelin suunnittelu ja toteutus.....	13
4.1	Suunnitteluvaihe: pelin konsepti saa alkunsa	13
4.2	Prototyyppi- ja viimeistelyvaihe: konseptista lopulliseksi tuotteeksi.....	14
4.3	Julkaisu- ja palautteenkeruuvaihe: peli eheytyy muiden havaintojen avulla	27
4.3.1	Ensimmäinen testipelaaja antaa palautetta	27
4.3.2	Toinen testipelaaja antaa palautetta.....	28
4.4	Tavoitteiden toteutumisen arviointi	28
5	Pohdinta.....	31
5.1	Vastaukset tutkimuskysymyksiin ja kehitysideoita	31
5.2	Henkilökohtaisia suunnitelmia	32
	Lähteet.....	34
	Liitteet	37
	Liite 1. Videopelien saavutettavuusohjeistukset.....	37
	Liite 2. Valmis esimerkki pelin toimintalogiikasta.....	39
	Liite 3. Pelin prototyyppi- ja viimeistelyvaiheen aikataulus	41
	Liite 4. Luettelo peliä varten kehittämistäni C#-skripteistä.....	42

Liite 5. "The Metronome of Dissonance" -pelin etusivu Itch.io-palvelussa	50
Liite 6. Valmiin pelin rakenne kuvankaappauksineen.....	51

1 Johdanto

Toiminnallisen opinnäytetyöni tavoitteena on suunnitella ja toteuttaa videopeli, joka on omistettu näkövammaisille ja -rajoitteisille henkilöille: heille, jotka on joko virallisesti luokiteltu näkövammaisiksi tai joilla on heikon näkökykynsä vuoksi vaikeuksia aistia pelejä visuaalisesti, menettäen täten mahdollisuuden autenttiseen pelikokemukseen. Projektini taustalla on halu inspiroida ja kunnioittaa näkövammaisia ja -rajoitteisia pelaajia, houkutella normaalinäköisiä pelaajia kokeilemaan tavallisesta poikkeavia pelejä, sekä muistuttaa pelinkehittäjiä siitä, miten paljon luovia mahdollisuuksia videopeleillä voi pohjimmiltaan olla.

Ennen projekti-ideani syntyä olin hankkinut kokemusta kahdesta asiasta: *videopelinkehityksestä* ja *saavutettavuudesta*. Videopeliurani alkoi audiovisuaalisen viestinnän perustutkintoa suorittaessani vuonna 2018, jolloin toteutin *BelleGames*-nimiselle yritykselle videopelin *Bob's Cat Challenge*. Nyt neljä vuotta myöhemmin, Haaga-Helian ammattikorkeakoulun viimeisen vuoden opiskelijana, ymmärrän paremmin kaikki ne vastoinkäymiset, jotka tuolloin vaikuttivat itselleni ylitsepääsemättömillä. Halu kehittyä videopelien tekijänä on säilynyt voimakkaana debyyttistäni asti.

Saavutettavuus, eli digitaalisten palveluiden ja tietoteknisten ohjelmistojen tasavertaisen käytön mahdollistaminen näkö-, kuulo-, liikunta- ja muita rajoitteita omaaville henkilöille, oli vuosina 2022–2023 suorittamani työharjoittelun keskeisin teema: olin tuolloin mukana tekemässä erilaisia saavutettavuusparannuksia silloisen työnantajani verkkopalveluun. Tekemieni muutosten jälkeen sivuston eri elementit olivat tarkemmin navigoitavissa näppäimistön *Tab-näppäimellä*, erilaiset painikkeet aktivoivatissa *Enter-näppäimellä* sekä kaikki elementit tunnistettavissa ruudunlukijan kautta. Toimenpiteet edellyttivät aktiivista testausta ruudunlukijalla, attribuuttien lisäämistä elementteihin ja eräässä tapauksessa jopa kolmannen osapuolen komponenttikirjaston muokkausta.

Saavutettavuudesta tuli aihe, joka vaikutti minuun syvästi: sen toteutuminen on enemmän tai vähemmän ihmisoikeuskysymys, sillä jokaisella on oikeus digitaalisten palveluiden ongelmattomaan käyttöön. Samalla huomasin, miten saavutettavuuden parantaminen on – mahdollisista vaikeuksista huolimatta – luoviin ongelmanratkaisukeinoihin kannustava tehtävä. Ohjelmistokehittäjille tämä voi silti näyttäytyä turhauttavana prosessina ilman konkreettista hyötyä: negatiivinen suhtautuminen saavutettavuuteen ei kuitenkaan tee asioita paremmaksi heille, joille digitaalinen esteettömyys on välttämätöntä palvelun tai ohjelmiston käyttämiseksi.

Ajatukseni saavutettavuuden monimuotoisuudesta yhdistyivät videopeleihin, joissa pääpaino on visuaalisissa aistihavainnoissa ja käsillä suoritettavassa ohjauksessa. Kansainvälisen pelinkehittäjien järjestön mukaan ”10–20 % ihmisistä eivät pelaa videopelejä vammansa takia” (Pusey 2019).

Vaikka kyseisellä arviolla tarkoitettaneen monenlaisia vammoja, uskon henkilökohtaisesti, että suurimmat vaikeudet videopelien pelaamisessa esiintyvät nimenomaan näkövammaisilla henkilöillä.

Näkövammaisille pelaajille on suunniteltu useita ääni- ja värinäpainotteisia video- ja mobiilipelejä, ja muutamissa suuren budjetin tuotannoissakin on heidän tarpeisiinsa sopeutuvia pelimekaniikkoja: yleisesti ottaen saavutettavuutta ei kuitenkaan ole vielä priorisoitu pelinkehityksessä (Link 2021; Blue 2021). Tämä sai minut pohtimaan, miten monesta pelikokemuksesta näkövammaiset pelaajat jäävät paitsi: saavutettavuus ei saisi olla vain pelikohtainen lisäominaisuus.

Havaintojeni ja ajatusteni innoittamana sain päätöksen luoda videopelin, joka on alusta asti suunniteltu näkövammaisten tarpeet, tunteet ja odotukset huomioiden. Samalla se olisi taiteellisesti kokeillinen teos, joka herättäisi arvostusta ja positiivista palautetta erilaisilta pelaajilta. Haluan myös todistaa, että saavutettavuus osana pelin identiteettiä voi tarjota uusia, moniaistillisia elämyksiä, jotka edistäisivät videopeliteollisuuden monimuotoisuutta. Vastaan myös projektini kautta päätutkimuskysymykseeni: *miten tarjota autenttinen pelikokemus näkövammaisille ja -rajoitteisille pelaajille?* Tätä pääkysymystä selvitetään yksityiskohtaisempien alakysymysten avulla: *mitkä ominaisuudet ovat tärkeitä visuaalisista aistihavainnoista riippumattoman pelikokemuksen toteuttamisessa? Mitä väärinkäsityksiä ja virhetilanteita voi ilmetä ei-visuaalisessa pelissä?*

Tutkimuskysymysteni avulla olen määrittänyt projektini tavoitteeksi luoda aistihavainnoista riippumattoman pelikokemuksen, mikä toteutetaan kolmiulotteisen äänimaailman ja persoonallisen ruudunlukujärjestelmän avulla. Pelini ammentaa vaikutteita näkövammaisille suunnitelluista, äänipainotteisista teoksista kuten *The Vale: Shadow of the Crown* ja *Lost and Hound*. Lisäksi tutustun suuren budjetin videopeliin *The Last of Us Part II*, joka on herättänyt positiivista huomiota kattavilla saavutettavuusasetuksillaan (Lenstore s.a.; Link 2021). Inspiraatioista huolimatta pyrin tekemään pelistäni uniikin teoksen, joka määrittää omaa tyyliäni pelintekijänä.

Toteutan videopelini *Unity*-pelimoottorilla, käyttäen myös *Blender*-mallinnustyökalua sekä *Nara-keet*-äänigeneraattoria. Kehitystyössä kulminoituvat aiemmin hankkimani taidot Unityn käytön lisäksi *C#-ohjelmoinnista* ja *3D-mallinnuksesta*. Hyödynnän tekijänoikeusvapaita äänitehosteita esimerkiksi *Freesound*- tai *Pixabay*-sivustolta, mutta toteutan myös muutaman äänitehosteen itse käyttäen *LingoJam Robot Voice Generator* -työkalua. Tarvittavat elokuva- ja äänieditoinnit teen *HitFilm*- ja *Audacity*-työkaluilla. Tarvittavan markkinointimateriaalin toteutan *Gimp*-kuvankäsittelyohjelmalla. Tavoitteenani on luoda kaikki saavutettavuustoiminnot itse, mutta kriittisissä tilanteissa olen valmis käyttämään kolmannen osapuolen kehittämiä funktiokirjastoja, mikäli ne vain parantavat saavutettavuutta toivomallani tavalla. Projektini lopputuotoksena olisi joko demoversiona tai kokonaisuutena julkaistu videopeli. Peli julkaistaan ilmaiseksi tietokoneelle joko *Steamin*, *Itch.io:n* tai muun

digitaalisen pelikauppa-alustan kautta. Mainostan projektiani ihmisille internetissä foorumeilla ja sosiaalisen median kanavien kautta.

Pelinkehitysprosessini kestää arviolta 12–13 viikkoa, ja se koostuu viidestä vaiheesta: *suunnittelu*-, *prototyyppi*-, *viimeistely*-, *julkaisu*-, sekä *palautteenkeruuvaiheesta*. Määrittelen projektini kokonaisuudessaan onnistuneeksi silloin, kun peli on selkeä identiteetiltään ja toimintalogiikaltaan, se on julkaistu onnistuneesti suositussa digitaalisessa verkkokaupassa, se on täysin pelattavissa ilman näköaistia ja se on saanut mielellään positiivista palautetta ensisijaisesti näkövammaisilta tai -rajoitteisilta pelaajilta.

Lopputuotteen kohdeyleisönä ovat ensisijaisesti näkövammaiset ja -rajoitteiset pelaajat, mutta myös kaikki muut henkilöt, jotka ovat kiinnostuneet äänipainotteisista peleistä tai muuten uskovat pelini tarjoavan jotain heidän yksilöllisistä mieltymyksistään poikkeavaa. Kohdeyleisöni kuuluvien henkilöiden pelaajahistorialla ja -statuksella ei ole merkitystä: he voivat olla niin aloittelevia kuin kokeneita pelaajia.

Rajaan tämän opinnäytetyön lukijakuntaa priorisoimaan heitä, joilla on jonkin verran kokemusta Unity-pelimoottorin käytöstä: myös ymmärrys muista käytettävistä teknologioista ja palveluista on suositeltavaa, mutta ne esitellään lyhyesti luvussa 3.3.

2 Tietoperusta

2.1 Videopelien saavutettavuus ja näkövammaiset pelaajat

Saavutettavuus videopeleissä voi kohdistua useaan eri ominaisuuteen: pelaajan motorisiin taitoihin, kognitiiviseen käyttäytymiseen, näkö-, kuulo- ja puheastiin tai jopa taloudelliseen tilanteeseen. Usein saavutettavuutta pidetään synonyymina monipuolisuudelle, jolloin pelistä luodaan mahdollisimman erilaisiin tarpeisiin sopiva, eri tavoin pelattavissa oleva tuotos. Toisinaan pelinkehittäjillä on kuitenkin yksipuolisia tai vääristyneitä käsityksiä siitä, mitä saavutettavuus on: kyse voisi heidän mielestään olla esimerkiksi vain värisokeuden huomioimisesta tai monikielisyyden takaamisesta. (Eng, s.a.)

Saavutettavuutta parantavia lisäyksiä on eritasoisia, joista vaativimpien lisääminen on sitä vaikeampaa, mitä valmiimpi kehitettävä peli on. Ihanteellista onkin vammaan tai rajoitteisuuden huomioiminen jo pelin suunnitteluvaiheessa aina konseptikuvituksesta pelintestaukseen. Saavutettavuuden toteutumattomuus kasvattaa riskiä siitä, että pelistä tulee ”väärällä tavalla vaikea”. (Eng, s.a.)

Näkövammaiset huomioonottavilla videopeleillä on pitkä historia: ensimmäisenä vastaavana pidetään italialaisen *Dedalomedia*-yrityksen videopeliä *Blindness* vuodelta 1996, joka oli elokuvallista tarinankerrontaa sisältävä, hiirellä navigoitava seikkailupeli. Koska peli kertoi sokeasta henkilöstä, kehitettiin projektia varten *kolmiulotteinen holofoninen äänijärjestelmä*: pelaaja kuuli äänisignaaleja liikuttaessaan hiirikursoria peliympäristössä ja lähestyessään klikattavia elementtejä – tämän avulla myös näkövammaisen pelaaja olisi pystynyt navigoimaan videopelissä. Hiirikursorilla ja sen käyttämällä äänijärjestelmällä pyrittiin mukailemaan näkövammaisen käyttämää valkoista apukeppiä. (Gerli 2023.)

Näkövammaisille suunnitelluissa peleissä äänimaailma on ollut keskeisessä asemassa, mikä on luonut käsitteen *audio game* eli äänipainotteinen peli. Esimerkkejä tällaisista peleistä ovat muun muassa näkövammaisen *Liam Ervenin* nuoruudessaan kehittämä, sivustakuvattu tasohyppelypeli *Super Liam*, jossa äänimaailma jaettiin kahdelle kaiuttimelle sen perusteella, kuuluvatko äänet pelihahmon edestä (oikealta) vai takaa (vasemmalta). Erven on sittemmin jatkanut saavutettavuusaiheisten projektien parissa muun muassa ideoimalla prototyypin 360-asteisesta äänitutkasta *Microsoft Studiosin* luomaa *Forza Motorsport* -ajopeliä varten, sekä tekemällä saavutettavuusparannuksia *Rockstar Gamesin* menestypeli *Grand Theft Auto V:n* tietokoneversioon. (Link 2021.)

Kaupallisista, pienkehittäjien tuottamista äänipeleistä huomiota ovat herättäneet erityisesti kaksi teosta. Australialaisen *Daisy Ale Soundworksin* kehittämässä *Lost and Hound* -pelissä ohjataan pelastuskoiraa, jonka tavoitteena on jäljittää vaarassa olevia ihmisiä tai kadoksissa olevia esineitä:

koiran tarkka hajuaisti kuvataan pelissä kolmiulotteisena, hymäilyä muistuttavana äänisignaalina, joka voimistuu pelaajan ollessa lähempänä kohdettaan. Toinen esimerkki kaupallisesta äänipelistä on *Falling Squirrelin* ja *Creative Bytes Studiosin* yhteistyössä kehitetty *The Vale: Shadow of the Crown*, jossa visuaaliset elementit on korvattu akustisilla ääniefekteillä ja ammattiääninäyttelyllä. (Lupetti s.a.; Pusey 2019; Link 2021.)

Äänipainotteiset pelit eivät ole ainoa tapa tarjota näkövammaisille suunniteltuja pelikokemuksia: tuntopalautteen, eli värinän ja muiden tehosteiden käyttö on tyypillistä erityisesti mobiilipeleissä. Esimerkiksi ajopelissä *Blind Drive* kaikki auton sisä- ja ulkopuolella tapahtuvat asiat, kuten maaston rakenne, onnettomuudet tai kuskin radiomusiikki, signaloidaan erilaisin värinätehostein pelaajalle. Toinen näkövammaisille suunnattu mobiilipeli, *Sam Friedmanin* kehittämä *Blind Spot*, sisältää ainoastaan haptisia tehosteita ilman kuvaa ja ääntä, minkä vuoksi sitä kutsutaan ”kosketuspeliksi”. (Link 2021; Lupetti s.a.)

Suurten peliyhtiöiden kehittämät pelit, joista käytetään myös nimitystä *AAA-pelit*, eivät yleisesti ottaen panosta näkövammaisten pelikokemukseen: positiivisissa tapauksissa ne ovat kuitenkin sisältäneet tuen ruudunlukijalle, tekstin tulkkauksen puheeksi, sekä mahdollisuuksia muun muassa pelinäytön kirkkauden säätämiseen, pelioloiden värisävyjen korostamiseen, tekstifonttien ja näyttöelementtien kokojen skaalaamiseen, kameran heilumistehosteiden poistamiseen sekä erilaisten reaktiokohtausten sivuuttamiseen. (Heath 2021; Lenstore s.a.)

Tällä hetkellä yhtenä saavutettavimmista AAA-peleistä pidetään *Naughty Dogin* kehittämää toimintaseikkailua *The Last of Us Part II*, joka sulautti luontevasti näkövammaisten aputoiminnot, erityisesti ääni- ja värinätehosteet, osaksi pelin identiteettiä. Pelaaja voi esimerkiksi ”kuunnella ympäristöään” ja aistia siten lähellä olevia objekteja etäisyyksineen, tai kääntää liikesuuntansa automaattisesti kohti oikeaa määränpäättä heti avattuaan karttavalikon. Lisäksi pelin vuorovaikutteiset oliot, kuten ovet ja ikkunat, on suunniteltu reagoimaan pelaajahahmon läsnäoloon, vaikka tämä ei olisikaan suoraan niiden edessä. Pelissä on myös oma ruudunlukijajärjestelmänsä, joka käyttää valmiiksi nauhoitettuja, ammattiääninäyttelijöiden lukemia selostuksia. (Heath 2021; Link 2021; Lenstore s.a.)

Toinen esimerkki saavutettavasta AAA-pelistä on *Electronic Arts-yhtiön* kehittämä, amerikkalaista jalkapalloa sisältävä *Madden NFL* -pelisarja, joka saavutettavuusohjaaja *Karen Stevensin* johtamana otti käyttöön värinätehosteet, monipuoliset äänenvoimakkuussäätimet ympäristön eri äänenlähteitä varten. Toisinaan saavutettavuus näkövammaisten osalta voi toteutua ilman suuria lisäyksiä: esimerkiksi sivusta kuvatun *Mortal Kombat 11* -taistelupelin yksityiskohtainen äänimaailma paljastaa selkeästi pelaajahahmon sijainnin ja etäisyyden vastustajasta, vaikka pelaaja ei näkisi ruudun tapahtumia. (Link 2021, Lupetti s.a.)

Videopelikonsoleista *Microsoftin Xbox Series X ja S* sekä *Sonyn PS5* sisältävät omat ruudunlukijansa, mutta niiden toimintakapasiteetit ulottuvat harvoin itse videopeleihin: kehittäjät vastaavat itse omien peliensä ruudunlukujärjestelmästä ja muista saavutettavuusominaisuuksista. Myöskään tietokoneella käytettävä *Steam*-pelialusta ei tue yleiskäyttöistä ruudunlukijaa täysin ongelmitta. *Nintendo* ei sen sijaan ole panostanut saavutettavuuteen omissa konsoleissaan. (Lupetti s.a.; RNIB s.a.)

2.2 Videopelien saavutettavuusohjeistusten noudattaminen

Videopelien saavutettavuusohjeistukset, jotka ovat luettavissa liitteessä 1, jaetaan kolmeen kategoriaan: perustason, keskitason sekä edistyneen tason ohjeistuksiin. Näkövammaisille suunnatuista saavutettavuusohjeistuksista kutakin kategoriata yhdistävät selkeästi luettavat tekstielementit, tuki ruudunlukijalle, pelaajan liikkumista helpottavat äänikartat ja -opastukset sekä monipuoliset ja yksilölliset äänitehosteet eri objekteille ja toiminnoille. Tavoitteenani on noudattaa kaikkia näitä ohjeistuksia, priorisoiden monipuolisen äänimaailman, kuvailutulkkauksen, selkeän peliolioiden välisen kontrastin ja pelihahmon liikkumista tukevat toiminnallisuudet. Fonttien ja muiden näyttöelementtien säätömahdollisuudet eivät ole ensisijaisia projektissani, sillä aion luoda visuaalisista elementeistä riippumattoman, äänipainotteisen videopelin. Mikäli aion kuitenkin käyttää teksti- ja näyttöelementtejä lopullisessa tuotoksessani, pyrin tekemään niistä automaattisesti suurikokoisia, taustasta erottuvia, helposti luettavissa olevia sekä ruudunlukijalle soveltuvia. (Game Accessibility Guidelines s.a.).

3 Empiirinen osa

3.1 Projektin alkumääritelmät

Olen antanut peliprojektini nimeksi *The Metronome of Dissonance*.

The Metronome of Dissonance on visuaalisista elementeistä riippumaton, äänipainotteinen toimintaseikkailupeli, joka on suunniteltu alusta alkaen näkövammaiset pelaajat huomioiden. Mikäli peli saa suosiota, pyrin kehittämään siitä saavutettavan myös kuulo- ja liikuntavammaisille henkilöille – nämä laajennukset tapahtunevat kuitenkin vasta myöhemmin tulevaisuudessa. *The Metronome of Dissonance* kehitetään *Unity*-pelimootorilla, hyödyntäen erilaisia saavutettavuuskirjastoja ja -ominaisuuksia sekä omaa *C#-ohjelmointia*. Se julkaistaan ilmaiseksi tietokoneelle joko *Steam*- tai *Itch.io*-verkkopalveluun, aluksi lyhyenä kokeiluversiona, mutta mahdollisuuksien mukaan myös kokonaisena versiona. Projektilla ei ole erillistä toimeksiantajaa: kehitän ja julkaisen pelin omasta aloitteestani.

The Metronome of Dissonance on onnistuessaan uusi lisäys näkövammaisille suunniteltujen pelien valikoimaan, ja samalla positiivinen esimerkki saavutettavuuden määritelmät täyttävästä videopelistä. Parhaimmillaan se murtaa omalta osaltaan ennakkoluuloja siitä, millaisia aisteja tarvitaan autenttisen pelikokemuksen aikaansaamiseen. Pelin pääkohdeyleisönä ovat ensisijaisesti näkövammaiset tai -rajoitteiset pelaajat, mutta ihanteena on myös saavuttaa ne normaalinäköiset pelaajat, jotka pitävät äänipeleistä tai muuten haluavat näköaistista riippumattoman pelikokemuksen. Yksi projektini lähtökohdista on yhdistää eri fysiologisia ominaisuuksia omaavat pelaajat, vaikka tämä ei olekaan aina täysin mahdollista.

Ainoa projektiani rajoittava asia on aikataulu sekä pelin julkaisuun liittyvät prosessit: julkaisun kesto voi vaihdella riippuen siitä, onko kauppapaikkana *Steam* tai *Itch.io*. Olen tarvittaessa valmis kustantamaan niin pelin julkaisun mahdollisesti edellyttämät maksut kuin ylimääräiset laitehankinnat.

Pelini laadulliset kriteerit täyttyvät ensisijaisesti sillä, että se on selkeä omalta identiteetiltään, se on julkaistu onnistuneesti viralliselle kauppapaikalle, se on täysin pelattavissa ilman näköaistin käyttöä ja on koettu miellyttäväksi pelata erityisesti näkövammaisten ja -rajoitteisten toimesta. Seuraavaksi tärkeimmät onnistumisen merkit ovat pelin latauskertojen määrät (muiden tahojen kuin itseni toimesta), sekä kaikki pelistä saatu kritiikki, jonka perusteella on arvioitavissa pelini onnistuminen näköaistittoman pelikokemuksen tarjoamisessa.

Pyrin kehittämään peliini oman ruudunlukujärjestelmänsä, mutta mikäli prosessissa tapahtuu yllättäviä haasteita tai esteitä, käytän Unitylle suunniteltuja, kaupalliseen tarkoitukseen käytettäviä lisäosia. Pelin taustamusiikit valitaan tekijänoikeusvapaista kappaleistaan tunnetun säveltäjä *Kevin*

MacLeodin perustamasta *Incompetech*-verkkomusiikkikirjastosta. Äänitehosteiden hankintasivustoina käytetään ainakin *Freesound*- tai *Pixabay*-äänitehostepalveluita – tarvittaessa teen myös itse äänitehosteita.

Oma ruudunlukijajärjestelmäni sisältäisi esinauhoitettuja ääniklippejä, jotka perustuvat joko omaan ääneeni tai tietokoneella tuotettuun puheeseen. Aina kun pelaaja liikuttaa kursoria valikossa tai painaa eri ruudulla olevia elementtejä, ruudunlukija toistaa tapahtumat selkeästi. Puheopastusta käytetään myös pelin tutoriaaliosiossa, johon pelaaja voi mennä valikon kautta aina halutessaan. Itse pelissä ei kuitenkaan käytetä puhetta vaan ainoastaan erillisiä 3D-äänisignaaleja.

Pelin toimintalogiikka tulee perustumaan kolmiulotteiseen äänimaailmaan ja siellä havaittaviin äänisignaaleihin. Lisäksi pelissä on oma ruudunlukijajärjestelmänsä. Näppäimistö ja kaiuttimet riittävät autenttiseen pelikokemukseen, mutta kuulokkeiden käyttö tulee olemaan suositeltavaa – mahdollisuuksien pyrin lisäämään peliin myös tuen erilliselle ohjaimelle. Lisäksi harkitsen värinätehosteiden ja äänisyötteiden hyödyntämistä pelissä.

3.2 Tuotannon eri vaiheet ja niiden päämäärät

Pelin tuotantoprosessi koostuu *suunnittelu*-, *prototyyppi*-, *viimeistely*-, *julkaisu*- ja *palautteenkeruuvaiheesta*. Suunnitteluvaiheessa määritellään pelin perusolemus: mistä se kertoo, millainen on sen toimintalogiikka ja miten sitä pelataan. Prototyyppivaiheessa suunnitelma toteutetaan toimivaksi peliksi: lopputulos voi poiketa jonkin verran alkuperäisestä suunnitelmasta. Viimeistelyvaiheessa varmistetaan, että peli ei sisällä bugeja tai muita toiminnallisuuteen negatiivisesti vaikuttavia asioita, ja julkaisuvaiheessa valmistaudutaan itse pelin julkaisuun valitulla alustalla – tässä tapauksessa joko *Steamissä* tai *Itch.io:ssa*. Myös pelin markkinointi on osa julkaisuvaihetta. Tämän jälkeen monitoroin pelin menestystä: sen latausmääriä ja erityisesti siihen kohdistuvaa palautetta. Vertailen myös valmiin pelini toiminnallisuuksia videopelien saavutettavuusohjeistuksiin.

3.3 Projektissa alustavasti käytettävien teknologioiden ja palveluiden esittely

3.3.1 Unity

Unity on yksi tunnetuimmista pelinteko-ohjelmistoista pelinkehittäjien keskuudessa. Sen toimintaperiaatteen voi havainnollistaa kolmella käsitteellä: *kohtauksella* (*Scene*), *peliliolioilla* (*GameObject*) ja *komponenteilla* (*Components*). Peli koostuu yhdestä tai useammasta kohtauksesta, jonka sisällä kaikki peliin liittyvät asiat, eli pelilioliot, ovat olemassa: pelaaja näkee kohtauksen tapahtumat siellä

olevan kameraolion kautta, ja siten vuorovaikuttaa muiden peliolioiden, kuten pelaajahahmon tai tekstielementtien kanssa. Pelioliot voivat olla *3D- tai 2D-malleja*, erillisiä *ääni- valo- video- ja UI-elementtejä*, tai jopa kokonaan *tyhjiä*. Näitä peliolioita voi myös yhdistää toisiinsa hierarkkisesti. Pelikohtaukseen siirretystä, etukäteen luodusta ja määritellystä peliolioista käytetään nimitystä *Prefab*. (Dolan 2019, Unity Documentation 2023 a.)

Peliolioiden käyttötarkoitus riippuu niiden sisällä olevista komponenteista, joiden sisältö ja toiminnallisuus vaihtelevat sijainnin määrittelevästä *Transform-komponentista* aina itse tehtyihin *C#-kooditiedostoihin* eli *skripteihin (Script)*. Toistensa kanssa vuorovaikuttavilla peliolioilla tyypillisiä ovat myös niiden visuaalisen ulkomuodon esittävä *MeshRenderer-komponentti* sekä toisten olioiden törmäyksen tunnistava *Box Collider -komponentti*. Pelaajahahmon liikkumista voi hallita käyttämällä muun muassa *CharacterController-komponenttia* ja sen eri metodeja. (Dolan 2019, Unity Documentation 2023 b.)

Valitsen Unityn pelimoottoriksi, koska minulla on siitä eniten kokemusta aikaisemman videopeliprojektini ansiosta. Lisäksi halusin valita alustan, jota käyttäessä pääsisin toteuttamaan mahdollisimman itsenäisesti omaa C#-koodia testatakseni nykyistä ohjelmointiosaamistani.

3.3.2 Steam ja Itch.io

Steam ja *Itch.io* ovat verkkokauppa-alustoja, joiden avulla voi ladata (ilmaiseksi tai maksua vastaan) muiden osapuolten tekemiä pelejä, sekä julkaista omia tuotoksiaan. *Itch.io* on suunniteltu itsenäisille pelinkehittäjille, minkä vuoksi se ei myöskään rajoita tuotteiden hinnoittelu- ja julkaisuprosessia. *Steam* puolestaan soveltuu kirjavasti kaikenkokoisille ja -tyyppisille pelinkehittäjille, mutta siellä pelin julkaisu vaatii erillisiä toimenpiteitä, kuten 100 dollarin julkaisumaksun maksamista sekä erillisen arvioinnin läpäisemistä. (*Itch.io* s.a.; Khan 2023; Xcolla s.a.)

Olen valinnut kummatkin mahdollisiksi julkaisualustoiksi, koska olen saanut pelaajana käsityksen siitä, että nämä kaksi ovat suosituimmat pelinjulkaisualustat. Tavoitteenani on julkaista peli ilmaiseksi aikatauluni rajoissa, minkä vuoksi *Itch.io* olisi todennäköisin vaihtoehto.

Pelin prototyyppivaiheen aikana päätän lopullisesti julkaista tuotokseni *Itch.io*:ssa Steamin sijaan. Käsittelen asiaa luvussa 4.2.

3.3.3 Blender

Blender on avoimen lähdekoodin *3D-mallinnus- ja animointityökalu* (Blender s.a.). Valitsin Blenderin, koska minulla on siitä eniten kokemusta aiempien mediaprojektieni, erityisesti ensimmäisen videopelini, ansiosta. Lisäksi pyrin aina tukemaan avoimen lähdekoodin työkaluja. Luon tällä työkalulla kaikkein monimutkaisimmat 3D-mallit: yksinkertaisemmat mallit teen suoraan Unityssa.

3.3.4 Narakeet

Narakeet on verkkosivusto, jonka avulla kirjoitettua tekstiä voi muuttaa tekoälyn puhumaksi äänitiedostoksi. Jokaisella tekoälyllä on oma nimensä ja aksenttinsa. Narakeetin luoman äänitiedoston voi ladata itselleen *MP3*-, *WAV*- sekä *MP4-muodossa*. (Narakeet 2021; Narakeet s.a. a; Narakeet s.a. b.)

Valitsen projektini ruudunlukijajärjestelmää varten Narakeetin, koska sen tarjoamien tekoälyjen äänet muistuttavat oikeiden ihmisten puhetta. Käytän sen valikoimasta *Ashley*-nimistä ääntä, koska se kuulosti mielestäni selkeältä ja ystävälliseltä mutta etäiseltä: priorisoin päätöksessäni peliprojektini salaperäisen imagon ylläpidon.

3.3.5 LingoJam Robot Voice Generator

LingoJam Robot Voice Generator on vanhaan *eSpeak*-kirjastoon perustuva ohjelma, joka muuttaa kirjoitetun tekstin ”monotoniseksi robottiääneksi”, jonka voi myös ladata erillisenä äänitiedostona. Ohjelma on luotu hauskanpitoa varten, mutta sitä voi käyttää kaikissa käyttäjän haluamissa projekteissa. (LingoJam s.a.)

Valitsin LingoJam Robot Voice Generator -ohjelman luodakseni äänitehosteita kahdelle asialle: pelissä esiintyville vihollisolioille sekä niin sanotun ”ääniaseen” valitsemisesta syntyville äänitehosteille. Tein päätöksen tämän työkalun valinnasta vasta pelinkehityksen aikana prototyyppivaiheessa, josta kerron enemmän luvussa 4.2.

3.3.6 Freesound

FreeSound on äänitehosteita ilmaiseksi tarjoava verkkosivusto, jonka valikoima koostuu *Creative Commons -lisensseillä* varustetuista *mp3*-, *FLAC*-, *Ogg Vorbis*- sekä *AIFF*- ja *WAV*-

äänitiedostoista. Äänitehosteiden käyttö ja muokkaaminen kaupallisessa tarkoituksessa riippuvat äänitiedoston suojaaman lisenssin asettamista ehdoista. (Freesound s.a.)

Valitsin Freesoundin pääosin tottumuksesta: olen käyttänyt sen monipuolisia äänitehosteita vanhoissa mediaprojekteissani. Pelinkehityksen prototyyppivaiheen lopussa korvaan kuitenkin Freesoundista lataamani äänitehosteet kokonaan *Pixabayn* valikoimalla: kerron prosessista lisää luvussa 4.2.

3.3.7 Pixabay

Pixabay on erilaisia video-, ääni- ja kuvatiedostoja ilmaiseksi tarjoava sivusto. Sen materiaali on vapaasti muokattavissa ja käytettävissä *Pixabay-lisenssin* ansiosta, ellei ehdoissa erikseen sanota. (Pixabay s.a.)

Valitsin *Pixabayn* *Freesoundin* tilalle vasta pelini prototyyppivaiheessa. Freesoundin tavoin myös *Pixabay* on minulle tuttu menneisyyden mediaprojekteistani. Kerron lisää valintani syistä luvussa 4.2.

3.3.8 Incompetech

Incompetech on muusikko *Kevin MacLeodin* perustama verkkosivusto, joka tarjoaa hänen itsensä säveltämää musiikkia ilmaiseksi kaupallisiin tarkoituksiin (Incompetech s.a. a; Incompetech s.a. b). Valitsen *Incompetechin*, koska olen käyttänyt sen musiikkia aikaisemmissa mediaprojekteissani.

Pelini prototyyppivaiheessa luovun suunnitelmistani käyttää musiikkia, minkä vuoksi *Incompetechista* ladattua tuotantoa ei esiinny pelissä. Kerron päätökseni syyn luvussa 4.2.

3.3.9 HitFilm

HitFilm on videoeditointiohjelmisto, joka sisältää myös mahdollisuuden erikoistehosteiden luomiseen ja käyttöön. Palvelu on käytettävissä niin ilmaiseksi kuin maksullisina tilauksina: erot näiden välillä liittyvät suurimmalta osin erikoistehostetyökalujen käyttövapauteen. (Paris 2022.)

Valitsin *HitFilmin*, koska minulla on ohjelmistosta kokemusta entisten mediaprojektieni kautta. Käytän ohjelmistoa luodakseni pelissä esiintyviä äänielokuvia ja tutoriaali kohtauksia: kaikki tiedostot

muutetaan äänitiedostoiksi, koska pelini ei ole tule olemaan riippuvainen visuaalisista elementeistä.

3.3.10 Audacity

Audacity on äänieditointiin erikoistunut, avoimen ohjelmiston työkalu, jossa on myös tuki oman äänilähteen nauhoitukselle sekä erikoistehosteiden käytölle (Audacity s.a.). Valitsen tämän ohjelmiston, koska haluan tarvittaessa täyden vapauden äänitehosteiden muokkaamiseen: lisäksi minulla on Audacitystä jonkin verran kokemusta harrastuneisuuteni kautta.

3.3.11 Gimp

Gimp on vuonna 1996 alkunsa saanut avoimen lähdekoodin kuvankäsittelyohjelma, joka tunnetaan monipuolisista työkaluistaan ja ominaisuuksistaan (Educba s.a.). Valitsin Gimpin, koska olen käyttänyt sitä paljon menneisyyden mediaprojekteissani: lisäksi tavoitteenani on aina tukea avoimen lähdekoodin ohjelmistoja.

4 Pelin suunnittelu ja toteutus

4.1 Suunnitteluvaihe: pelin konsepti saa alkunsa

The Metronome of Dissonance kertoo nimettömästä pelaajahahmosta, joka on hukannut itsensä vieraaseen, äänisignaaleista koostuvaan todellisuuteen. Pelaajan tehtävänä on löytää *Riitasoinnun metronomi* (*The Metronome of Dissonance*), jonka avulla hän voi palata takaisin omaan maailmaansa.

Pelin ensimmäinen taso on toimintamekaniikaltaan labyrinthimainen maailma. Pelaaja aloittaa risteyspisteestä, josta hän voi liikkua vasemmalle, oikealle tai suoraan eteenpäin. Pelaaja liikkuu automaattisesti haluttuun suuntaan niin kauan, kunnes tämä kohtaa uuden risteyksen, ansan tai vihollisen. Aina ollessaan pysähdyksissä pelaaja voi käyttää erikoiskykyä, jonka avulla hän voi kuulla määränpäästä kantautuvan metronomin äänisignaalin etäisyyden ja suunnan: tämä auttaa pelaajaa tunnistamaan määränpään sijainnin. Aina kun pelaaja liikkuu, jakautuu kävelyäänitehoste äänilaitteiden vasemmalle, oikealle tai molemmille puolille pelaajan kulkusuunnan perusteella: sama pätee myös pelaajan ympärillä oleviin äänitehosteisiin.

Koska pelaajan edessä olevat äänet jakautuvat molemmille puolille kuulokkeita, on vaikeaa määrittellä pelaajan takana olevien äänien jakautumista kuulokkeille. Tämän vuoksi pelaaja ei voi kulkea taaksepäin, vaan ainoastaan vasemmalle, oikealle ja eteenpäin. Mikäli pelaaja joutuu umpikujaan, tämä palaa portaalin kautta takaisin siihen risteykseen, jossa hän oli ollut viimeksi. Toinen tapa helpottaa labyrinthissa kulkemista on tehdä selkeä ero risteyksestä alkavien teiden välillä: kolmesta reitistä yksi johtaa aina umpikujaan, loput kaksi puolestaan joko helpompaan tai vaikeampaan reittiin.

Pelissä kohdataan erilaisia uhkia. Pelaajan ollessa liikkeessä tätä kohti voi lähestyä varjomaisia ansoja, jotka uhkaavat viedä pelaajan mukanaan tyhjiöön. Jotta pelaaja voisi kulkeutua ansojen lävitse turvallisesti, on tämän painettava tiettyä suojautumispainiketta oikeaan aikaan. Ansan lähestyessä pelaaja kuulee voimistuvan huminan, minkä perusteella suojautuminen pitää ajoittaa oikein. Interaktio on verrattavissa tasohyppelypeleistä tuttuihin rotkoihin tai lattia-ansoihin: niiden yli on hypättävä oikealla ajoituksella tai muuten pelaaja häviää.

Pelaaja voi myös kohdata hirviöitä muistuttavia olentoja, jotka yrittävät hyökätä tämän kimppuun eri suunnista. Puolustautuakseen hyökkäykseltä pelaajan on tunnistettava vihollinen tämän äänen perusteella, valittava oikea *ääniloitsu* (myöhemmin ”ääniase”) ja ammuttava se hyökkäävän vihollisen suuntaan. Jos esimerkiksi pelaajan eteen ilmestyisi olento, joka päästää mehiläisen pörinän kaltaista ääntä, valitsee pelaaja ääniloitsujen joukosta sen, joka kuulostaa samanlaiselta pörinä-ääneltä, ja painaisi täten *ylänuolinäppäintä* ampuakseen loitsun edessään olevaa vihollista kohti.

Taistelutilanteessa pelaaja valitsee oikean loitsun kelloa muistuttavasta, pyöreästä valikosta: pelaaja voi liikuttaa kursoria myötä- ja vastapäivään käyttämällä *Q- ja E-näppäimiä*. Aina kun ääniloitsu on valittu, pelaaja kuulee äänitehosteen, jota hän voi vertailla vihollisesta kuuluvaa ääneen: jos nämä sopivat toisiinsa, on pelaaja valinnut oikean vaihtoehdon. Ääniloitsut ovat tarkoituksella kimakoita versioita vihollisäänistä – tämän tarkoituksena on helpottaa ääniloitsun ja vihollisäänten keskinäistä vertailua ja samalla tuoda jännitystä taisteluun.

Mitä lähempänä pelaaja on määränpäättään, sitä selkeämpänä tämä ääni kuuluu – samalla myös pelissä esiintyvät vaarat monipuolistuvat ja kasvavat. Pelaaja voi ottaa vastaan vain rajallisen määrän vahinkoa: vihollisen hyökkäys vie kolmasosan elämäpisteistä, varjoonsaan osuminen puolestaan laskee ne automaattisesti nolleen. Menetettyään elämäpisteensä pelaaja palaa takaisin häviötä edeltävään risteykseen.

Pelituotanto alkaa opinnäytetyöni toisessa vaiheessa, *prototyypivaiheessa*, ja päättyy sitä seuraavan *viimeistelyvaiheen* jälkeen. Yhteensä molemmat vaiheet kestävät vähintään viisi viikkoa. Prototyyppi- ja viimeistelyvaiheen tarkempi aikataulu on liitteessä 3.

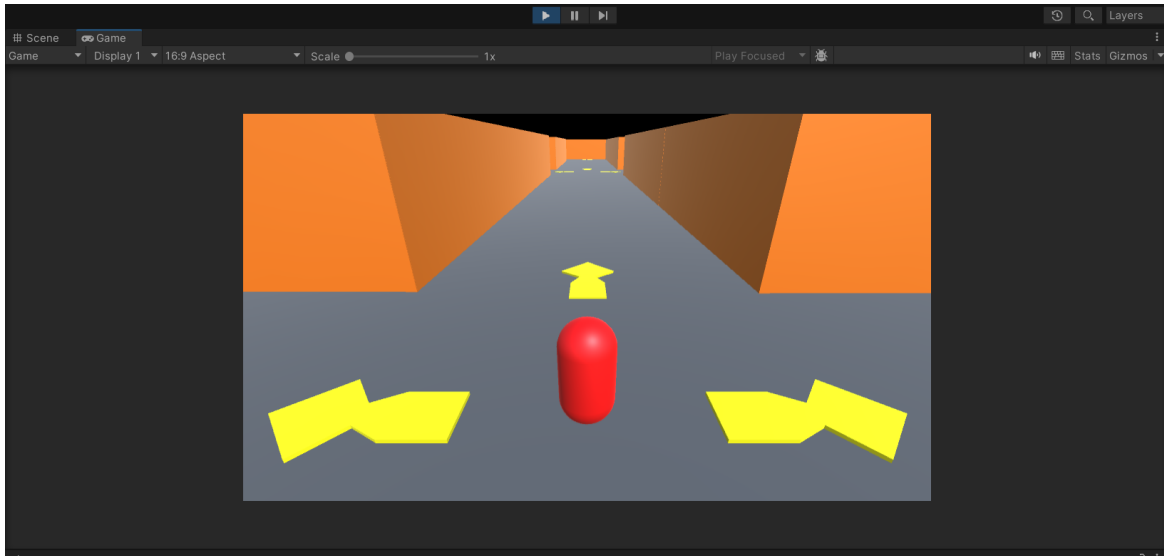
4.2 Prototyyppi- ja viimeistelyvaihe: konseptista lopulliseksi tuotteeksi

Pelisuunnitelman valmistumisen jälkeen aloitan prototyypivaiheen, jonka tavoitteena on luoda ensimmäinen versio toimivasta pelistä: tähän sisältyvät pelitasojen ja 3D-olioiden luominen sekä pelilogiikan suunnittelu C#-ohjelmoinnin avulla. Tarkempi kuvaus koko pelin lopullisesta pelilogiikasta ovat liitteessä 2.

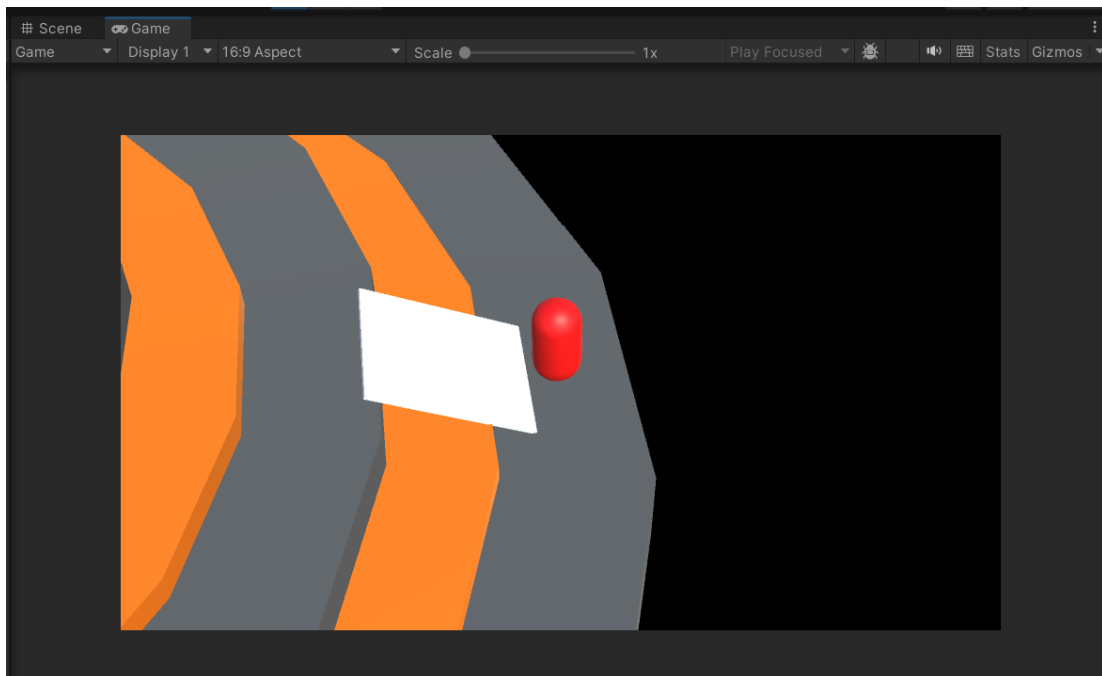
Luodessani pelin labyrinttikenttää, jonka olin määritellyt suunnitteluvaiheessa, aloin pohtia sille vaihtoehtoisia rakenteita. Olisiko pyöreä labyrintti parempi vaihtoehto kuin suorista seinämistä koostuva sokkelo? Olisiko jännittävämpää, jos pelaaja aloittaisi labyrintin ytimestä eikä ulkopuolelta? Miettiessäni näitä kysymyksiä, ja tehdessäni luonnoksia paperille, tulin johtopäätökseen: sekä suora että pyöreä labyrintti ovat molemmat kiinnostavia ja voisivat ongelmitta sijoittaa samaan peliin. Aloituspiste säilyisi kuitenkin molemmissa labyrinteissa aina niiden ulkopuolella, sillä pelaajan kulkeminen syvälle labyrinttiin olisi toimiva vertauskuva omien ongelmien kohtaamiselle: oman mielen sisälle meneminen on vaikeaa mutta sen arvoista.

Päätettyäni sisällyttää molemmat labyrinttivarientit (suoraseinäinen ja pyöreä labyrintti) osaksi peliäni, suunnittelin tasojen esiintymisjärjestyksen: peli alkaa suoraseinäisestä, eteenpäin ja sivuille kuljettavasta *sokkelosta*, ja kun pelaaja on löytänyt määränpänsä eli metronomin, siirtyy tämä suoraan pyöreään *kehään*, jossa pelaaja pyörii automaattisesti kiertoliikettä, tarkoituksenaan

päästä astetta lähemmäksi kehän ytimeen. Molemmista kentistä käytän vastedes nimityksiä *sokkelo* ja *kehä*.



Kuva 1: ensimmäinen pelitaso eli suoraseinäinen *sokkelo*, jossa pelaaja valitsee kulkusuunnan risteyksessä olevien nuoliolioiden perusteella ja ajautuu sen kautta erilaisiin reitteihin.



Kuva 2: toinen pelitaso eli pyöreä *kehä*, jonka ympärillä pelaaja on jatkuvassa, gravitaatiomaisessa pyörimisliikkeessä. Pelaajan tehtävä on päästä valkoisten siltaolioiden avulla lähemmäksi kehän ydintä.

Teen suurimman osan pelin 3D-objekteista suoraan *Unity-pelimoottorissa*, jonka kautta pystyy luomaan automaattisesti laatikon-, kapselin- ja pallonmuotoisia *peliolioita*: näitä olioita venytän ja värityn yksilöllisesti, jotta ne olisivat helpommin tunnistettavissa pelitasossa. Yksinkertaiset 3D-mallit riittävät todennäköisesti tähän peliin, mutta monimutkaisimmat mallit teen *Blender-ohjelmistolla*.

Teen peliä varten erilaisia 3D-objekteja. Seuraavat oliot kuuluvat 1. pelitasoon:

- *Pelaajaolio (PlayerObject)*
- *Pelaajaolion alaolio (Player)*
- *Pelaajaolion ympärille asetetut kävely- ja ääniasesensorit sekä Pelaajaolion sisälle asetetut sisäiset äänilähteet (WalkSoundSensors, SoundAttackSensors, SoundWeaponSelector ja InnerSoundSystem)*
- *Nuoliolio (new-arrowsign, mallinnettu Blenderillä)*
- *Risteysolio (CrossroadsParent)*
- *Risteysolioon kuuluvat osat (Crossroads)*
- *Checkpoint-olio (Checkpoint)*
- *Vihollisolio (Enemy)*
- *FightStop-olio eli Taistelupiste (FightStop)*
- *FightStopTrigger-olio (FightStopTrigger, osa FightStop-oliota)*
- *ShieldSpot-olio (ShieldSpot, aktivoi Shadow-olion)*
- *Shadow-olio (Shadow, aktivoituu ShieldSpot-olion kautta)*
- *Umpikuja-olio (DeathZone, esitellään pelissä nimellä VoidZone)*
- *InnerWall-olio (InnerWall, ensimmäisen pelitason seinäolio)*
- *Metronomi (Metronome, sama kuin toisessa pelitasossa)*

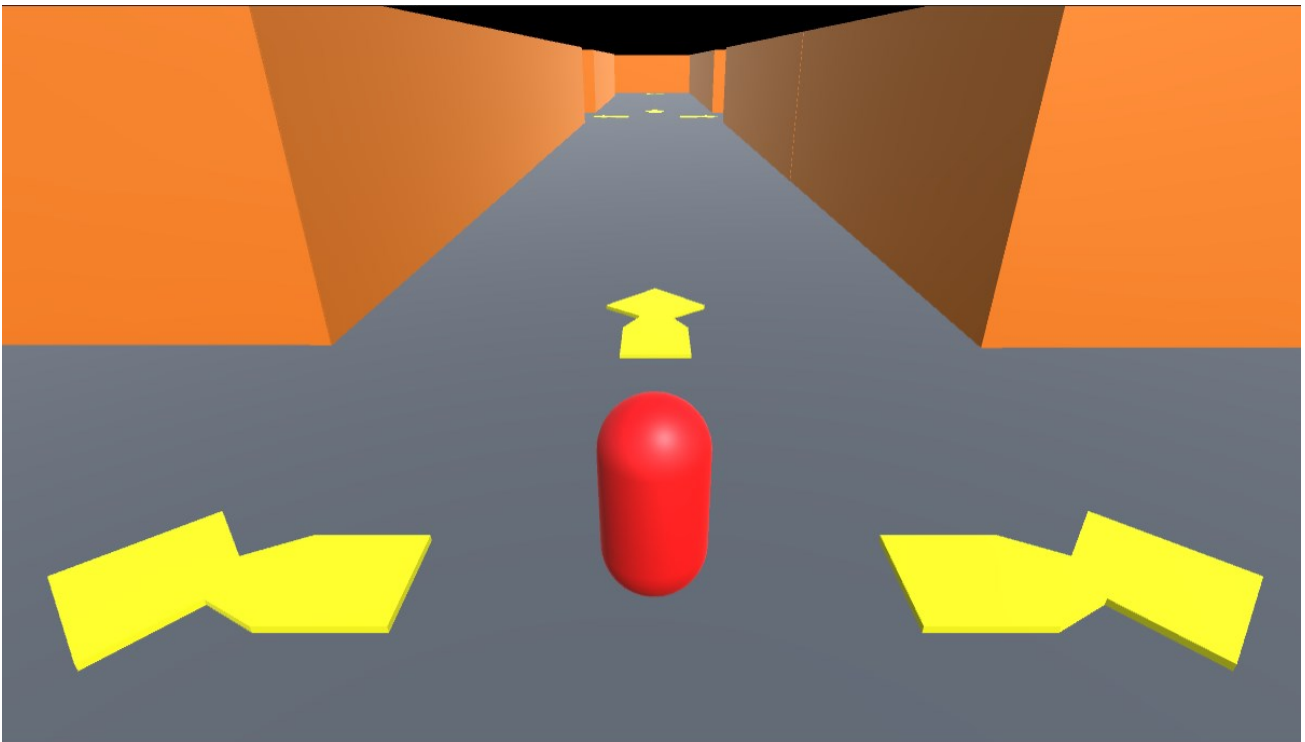
Seuraavat oliot kuuluvat puolestaan toiseen pelitasoon:

- *Pelaajaolio (Player, sama kun ensimmäisessä pelitasossa mainittu Pelaajaolion alaolio)*
- *Häkkiansan laukaisuolio (CageTrigger)*
- *Häkkiolion seinämä (Cage)*
- *Siltaolio (Bridge)*
- *Siltaolion nuoliolio (bridge-arrowsign, osa Siltaoliota)*
- *Varsinainen siltarunko (Bridge, osa Siltaoliota, vaikka molemmilla on sama nimi)*
- *Sillan päätepiste (BridgeEnd, osa siltaoliota)*
- *Metronomi (Metronome, sama kuin ensimmäisessä pelitasossa)*
- *Kehässä oleva harmaa rata, jonka päällä pelaaja pyörii (Circle Floor)*
- *Kehän oranssi seinämä, joka erottaa kehän harmaat radat toisistaan (Circle Wall)*

Vaikka pelissä esiintyvät oliot ovat prototyypivaiheessa yksinkertaisia, *Unityn* kautta luotuja *pelio-lioita*, käytän *Blender-mallinnusohjelmaa* luodakseni *Nuolioliot*, joilla pelaajan liikesuuntaa muutetaan tarvittaessa.

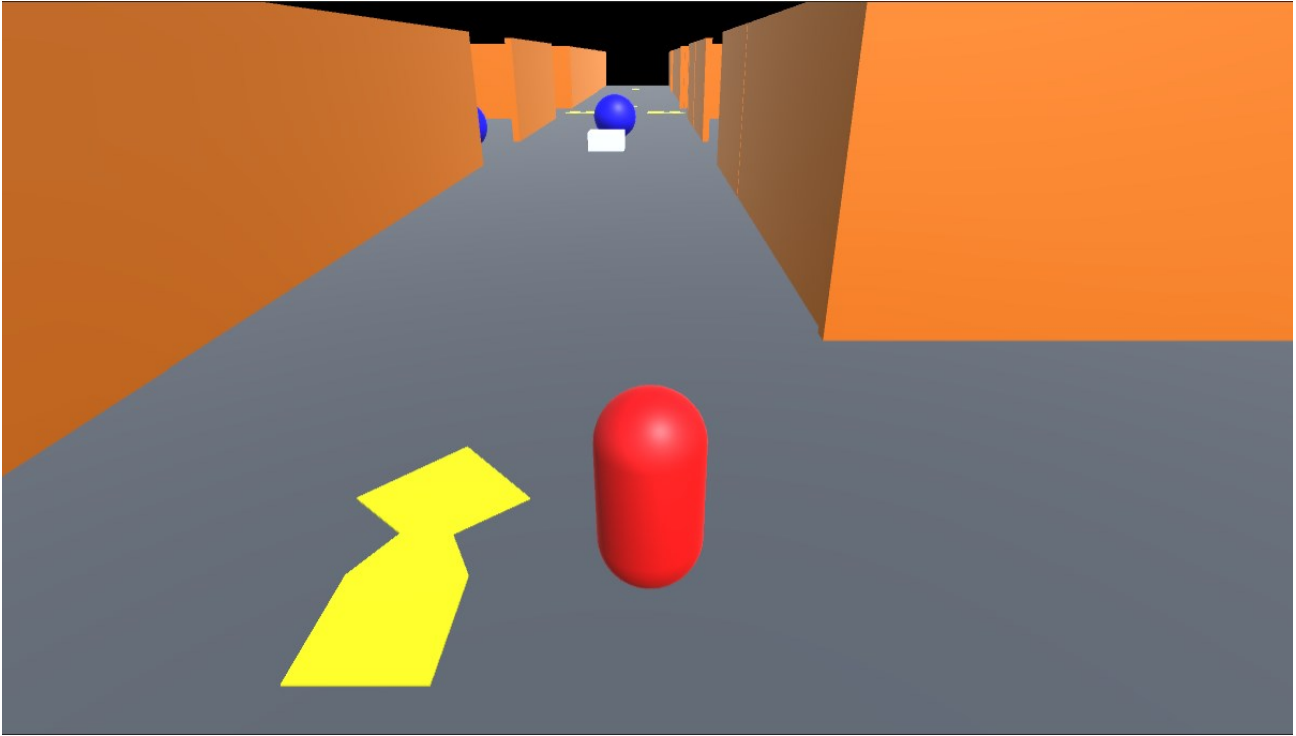
Unityssa käytettäviä *C#-kooditiedostoja* kutsutaan nimellä *script* eli *skripti* (Dolan 2019). Suunnittelen ja toteutan itse kaikki pelissä esiintyvät *C#-toiminnallisuudet*. Tarkemmat määritelmät skriptien toiminnallisuuksista ovat liitteessä 4.

Aloitan ohjelmoimisen pelaajan kontrolleista ensimmäisessä pelitasossa eli *Stage 1 -kohtauksessa*. Peruseriaatteena on pystyä liikuttamaan *Pelaajaoliota* haluttuun suuntaan aina kun tämä on kosketuksissa *Risteysolion* kanssa. Teen tämän siten, että lisään *Pelaajaoliolle* erillisen *interactionCheck-olion*, joka tarkastaa *Physics.CheckSphere -metodin* avulla, onko pelaaja kosketuksissa *Risteysolion* kanssa. Jos metodi palauttaa arvon *true (tosi)*, on pelaajalla mahdollisuus päättää oma liikesuuntansa joko *W*-, *A*- tai *D*-painikkeella. Pelaaja liikkuu tällöin automaattisesti valittuun suuntaan.



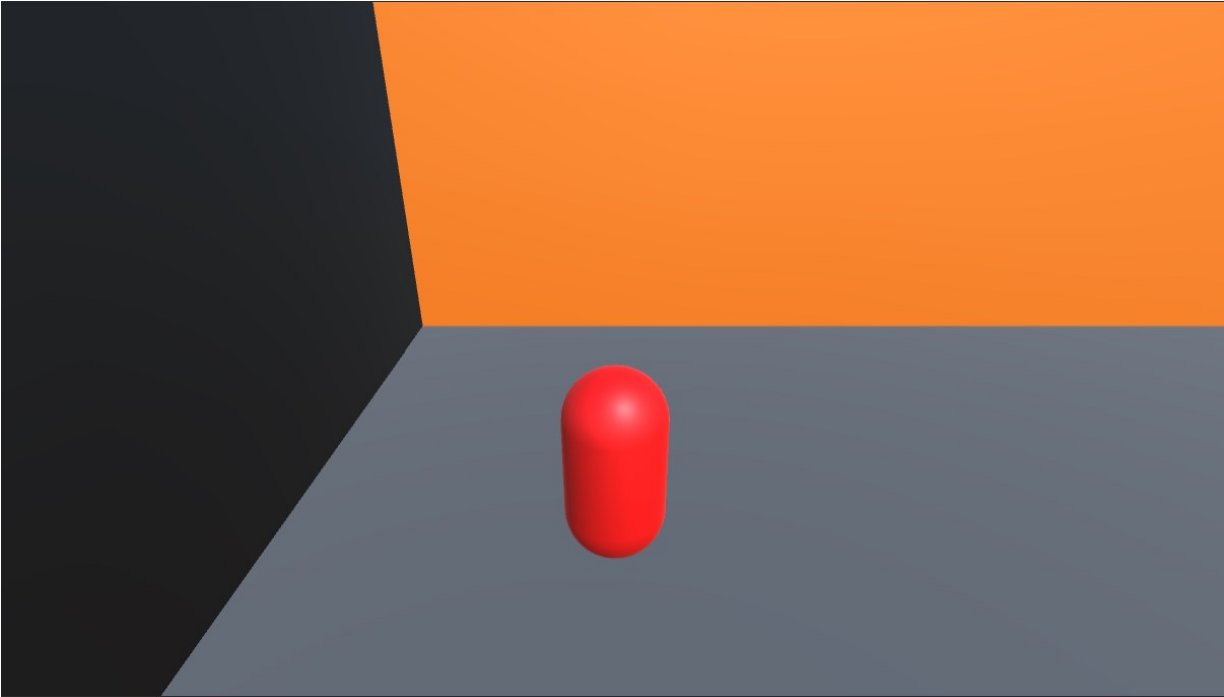
Kuva 3 Pelaajaolio on pelin ensimmäisen tason risteyksessä nuoliolioiden ympäröimänä.

Suunnittelen seuraavaksi *Nuoliolion*, joka antaa Pelaajaolion kontrolleista vastaavalle skriptille parametrina oman *rotaatiokulmansa z-akselissa*: Pelaajaolio vaihtaa liikesuuntaa automaattisesti sen perusteella, mihin Nuoliolio osoittaa. Tämän tekeminen vaatii jonkin verran testausta, sillä minun pitää selvittää tarkat rotaatioarvot jokaiselle nuolen eri suunnalle, ennen kuin pystyn tekemään yksityiskohtaista suunnanvaihdosfunktiota Pelaajaolille.



Kuva 4 Pelaajaolio on lähestymässä keltaista nuolioliota, joka kääntää hänen suuntaansa.

Seuraavaksi luon Pelaajaolion *elämäpistejärjestelmän*, pelaajan automaattisesti tuhoavat *DeathZone-alueet* ja niin sanotun *respawn-toiminnon*, jonka perusteella pelaaja palaa takaisin tiettyyn pisteeseen menetettyään kaikki elämäpisteensä. Elämäpistejärjestelmän tekeminen onnistuu suhteellisen yksinkertaisesti: määrittelen skriptissä täydet elämäpisteet ja jäljellä olevat elämäpisteet, joista jälkimmäinen vähenee aina kun vahinkoa tekevä funktio aktivoituu. Respawn-toiminnallisuuden toteuttamiseksi on deaktivoitava Pelaajaoliossa oleva *CharacterController-komponentti*, jotta Pelaajaolion voi siirtää minne tahansa paikkaan. Respawn-toiminnallisuuden täydennykseksi luon *Checkpoint-pisteen*, joka tallentaa aikaisemman risteyskohdan koordinaatit, sekä erillisen ”kuolinanimaation”, jossa Pelaajaolio nousee yläilmoihin ennen palautumistaan aikaisempaan risteyskseen: päätös kyseisen animaation toteutuksesta tapahtui suunnitellessani pelin taistelujärjestelmää, josta kerron lisää seuraavaksi.



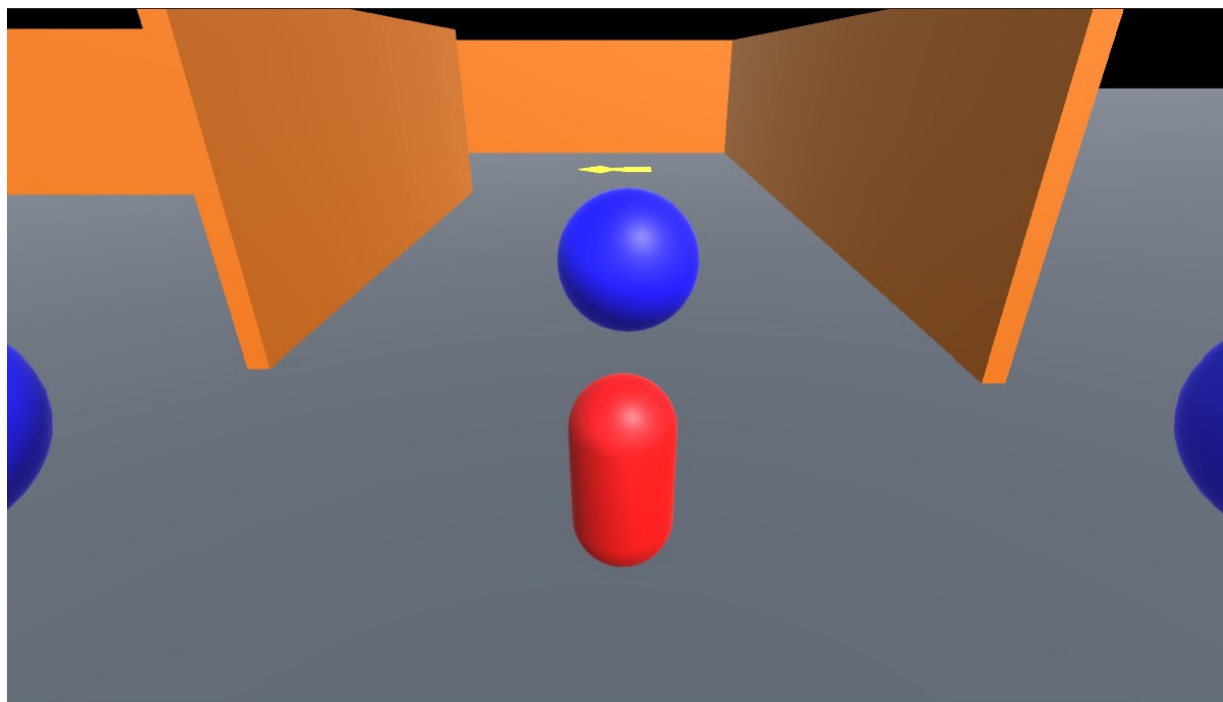
Kuva 5 Pelaaja on lähestymässä DeathZone-oliota.



Kuva 6 Pelaaja on menettänyt elämäpisteensä, ja nousee yläilmoihin näkymättömänä.

Pelin *taistelu- ja vihollisjärjestelmän* kehittäminen ovat suhteellisen pitkiä kokonaisuuksia. Aina kun pelaaja on *Taistelupisteessä*, tämä saa mahdollisuuden aktivoida kolme erilaista *ääniasetta* (alkuperäisessä suunnitelmassa *ääniloitsu*), joista kuuluu oma yksilöllinen äänitehosteensa. Pelaaja voi selata aseita *Q- ja E-näppäimillä*. Tehtyään valintansa pelaaja painaa *nuolinäppäintä*, jonka osoittamaan suuntaan tämä ampuu *Raycast-fysiikkaolion*. Määritän Raycastille seuraavan ehdon: jos se törmää viholliseen, jonka oma äänitehoste sisältää saman numeron kuin pelaajan valitsema ääniasse, tuhoutuu vihollinen heti. Ampumisen jälkeen pelaajalla on lyhyt vaihe, jolloin tämä voi selata ääniasseita muttei ampua: vaihe kestää muutaman sekunnin.

Taistelupiste koostuu aina tietyistä määrästä Vihollisolioita. Viholliset on ajastettu hyökkäämään sopivan sekuntimäärän välein, jolloin pelaajalla on aikaa tuhota kerrallaan hyökkäävä vihollinen. Valitsen sopivan sekuntimäärän miettimällä toimivaa äänitehosteiden ajoittamista. Vihollinen päästää aina oman äänensä ennen kuin se käy hyökkäykseen. Taistelupiste tarkistaa siinä olevien vihollisten aktiivisuutta jatkuvasti: kun kaikki viholliset on tuhottu, taistelu päättyy, ja pelaaja ohjataan liikumaan eteenpäin. Jos pelaaja menettää elämäpisteensä taistelun yhteydessä, nousee tämä yläilmoihin muutamaksi sekunniksi osana "kuolinanimaatiotaan". Syy tähän on Taistelupisteessä oleva *FightStop-skripti*, jonka *OnTriggerExit* -funktio pysäyttää Vihollisolioiden hyökkäykset vasta kun Pelaajaolio on kirjaimellisesti poistunut Taistelupisteen alueelta: tämä ei toteutuisi, jos Pelaaja vain yllättäen katoaisi paikalta, minkä vuoksi päätin "nostattaa" Pelaajaolion ensin yläilmoihin ja vasta sitten siirtää sen *checkpoint-pisteeseen*.

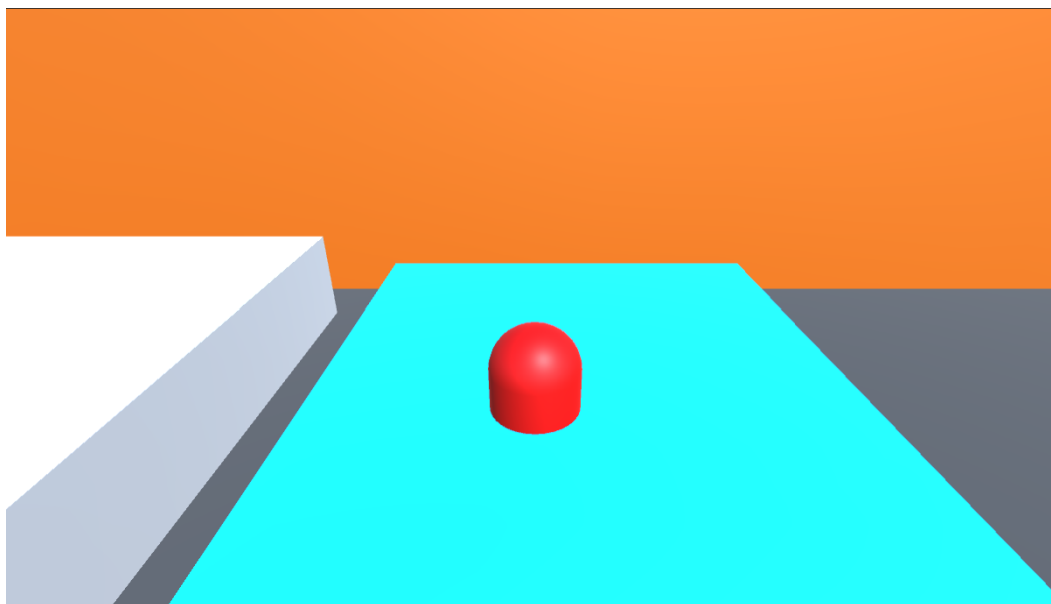


Kuva 7 Pelaajaolio on joutunut äänitaisteluun, ja hänen ympärillään on kolme vihollisoliota.

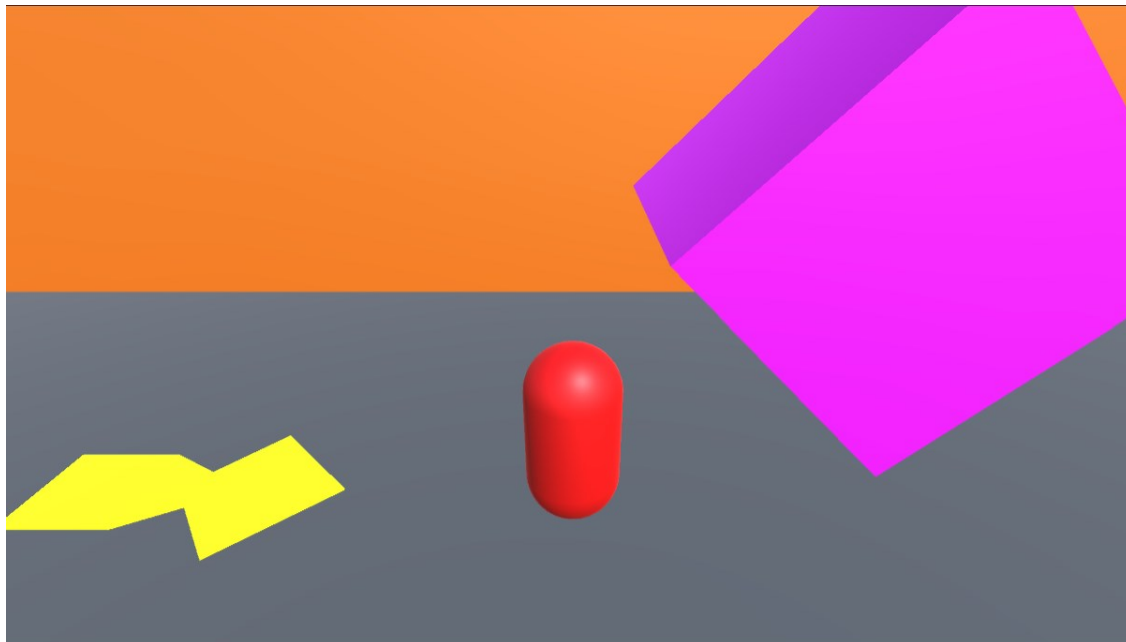
Taistelujärjestelmän viimeisessä versiossa otan käyttöön lopulliset äänitehosteet, niin Vihollisoli-
oille kuin itse pelaajan omille ääniaseille. Alkuperäisen suunnitelmani mukaan vihollisten äänite-
hosteet ovat erilaisia ääniä aina lasinsirpaleista mehiläisen surinaan, ja pelaajan on ammuttava
näitä kohti samoilla äänillä mutta vain kimeiksi muutetuilla versioilla: ajatuksena oli, että pelaaja
tunnistaa vihollisäänen ja vastaa heille ”samalla signaalilla mutta eri taajuudella”. Satunnaiset ääni-
tehosteet sekoittuivat kuitenkin helposti keskenään, mikä teki pelaamisesta hankalaa. Päätin lo-
pulta pitää alkuperäisen taisteluidean mutta vaihtaa satunnaiset äänitehosteet robottimaisiksi sa-
noiksi: täten pystyin tekemään vihollisäänistä matalampia ja ääniaseiden äänitehosteista kimeäm-
piä, mutta silti pitämään taistelutilanteen helposti aistittavissa kuuloastein. Päätin samalla myös
luoda toiminnon, jossa jokainen vihollinen arpoo aina yhden näistä kolmesta sanasta hyökätes-
sään uudestaan: tämä tuo peliin automatisaatiota ja jännitystä.

Annan vihollisobjekteille kolme erilaista sanaa: *Shita*, *Naka*, ja *Ue*. Sanat ovat japania ja tarkoitta-
vat tietyssä kontekstissa samaa kuin ”Ala-”, ”Keski-” ja ”Ylä-”. Vastaavasti pelaajan ääniasetta vali-
tessa toistuvat samat sanat mutta huomattavasti kimeämpinä versioina. Tämä pitää yllä ajatusta
siitä, että pelaaja voittaa vihollisen matalan äänen vastaamalla siihen kimakalla äänellä. Päätin va-
lita japaninkieliset sanat, sillä ne kertovat jotain mielenkiinnostani aasialaisia kieliä kohtaan, tuovat
tiettyä mystiikkaa peliin ja ovat samalla viittaus vanhaan japanilaistelevisiosarjaan, *Ultramaniin*,
jossa esiintyy robottimainen, omaa nimeään hokeva hirviö *Zetton* (*Tsuburaya s.a.*). Viittaukset it-
seäni kiinnostaviin asioihin ovat osa tavoitettani persoonallisen pelin luomiseen liittyen.

Taistelujärjestelmän jälkeen kehitän peliin *Ansaolioita*, joita lähestyessään pelaajan on painettava
välilyöntiä. Tämä vaihe oli ajoittain haasteellinen, sillä suunnitelmani vaihtelivat *Varjoansan* ja *rot-*
kon välillä: yritin toteuttaa hetken aikaa jopa hyppyfysiikkaa, jolloin pelihahmo oikeasti hyppäisi tie-
tyn ansan yli. Kyseinen vaihe sisälsi muutaman epäonnistumisen, joten palasin takaisin yksinker-
taisempaan ideaan: kun pelaaja lähestyy *Varjoansaa*, tämä välttyy *Shadow*-olion aiheuttamalta va-
hingolta painaessaan välilyöntinäppäintä. *Shadow*-oliota edeltävän *ShieldSpot*-olion kohdalla pe-
laaja kuulee ääniefektin, joka varoittaa lähestyvistä ansasta.



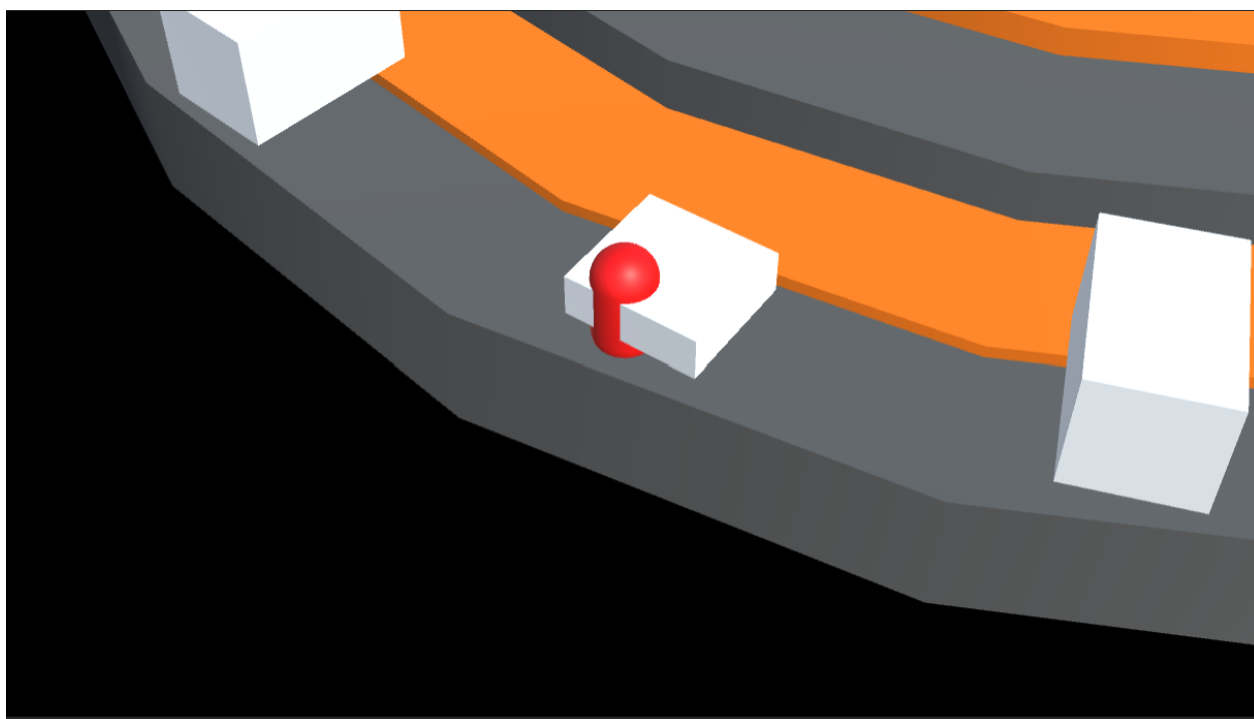
Kuva 8 Vasemmalle kulkeva Pelaajaolio on joutunut kosketuksiin turkoosin ShieldSpot-olion kanssa. Painaessaan välilyöntiä pelaaja deaktivoi valkoisen Shadow-olion ja pääsee turvallisesti ansan läpi.



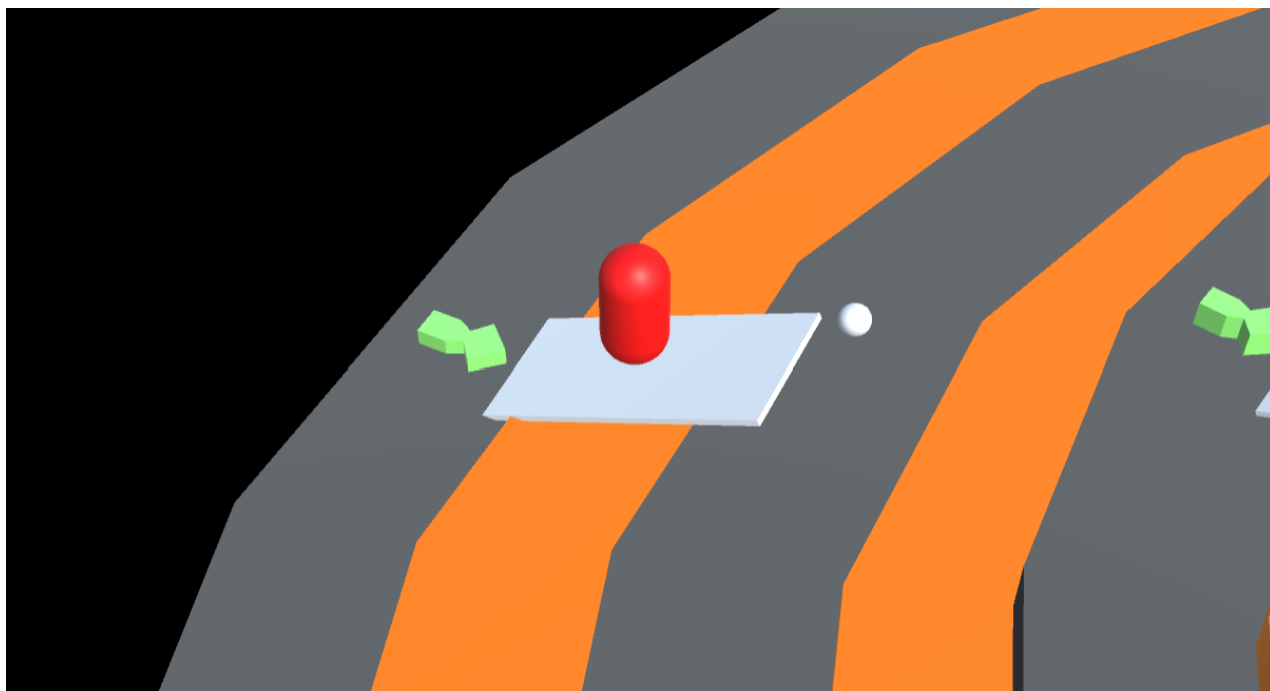
Kuva 9 Pelaaja on lähestymässä Metronomioliota, minkä seurauksena ensimmäinen taso päättyy ja uusi kohtaaus alkaa.

Suunniteltuani *ensimmäisen tason (sokkelo)* toiminnallisuudet, alan viimeistellä *toisen tason eli Stage 2 -kohtauksen* pelilogiikkaa. Kun pelaaja pyörii kehässä, aina kun hän osuu *Siltaolioihin*, hänen on painettava joko *W-, A- tai D-näppäintä* sen perusteella, mistä suunnasta hän kuulee ääniefektin: kyseessä on kaksiulotteinen ääni, joka kuuluu joko pelaajan vasemmalta, oikealta tai molemmilta puolilta riippuen siitä, millä puolella pelaajaa *kehän ydin eli metronomi* sijaitsee. Painaessaan oikeaa painiketta pelaaja liikkuu sillan toiseen päähän, lähemmäksi itse kehän ydintä, ja jatkaa sitten pyörimistä kellonmyötäiseen suuntaan.

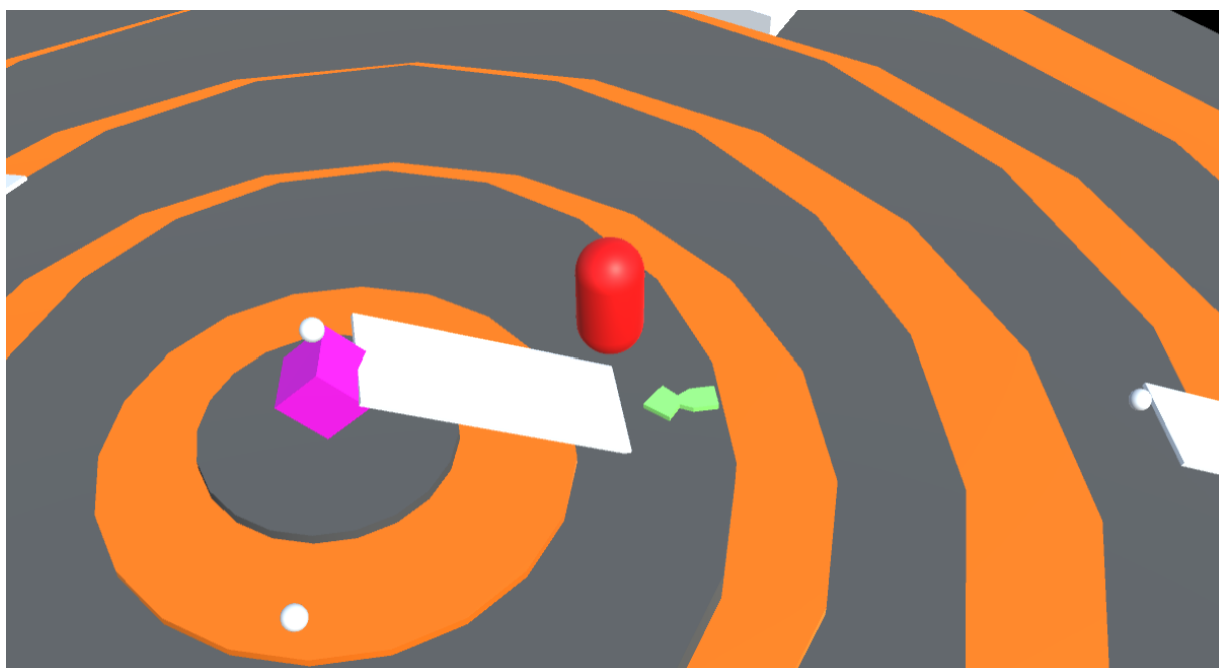
Toteutan peliin myös ansan nimeltä *Häkkiansan laukaisuolio*, joka aktivoituu aina kun pelaaja osuu siihen: tällöin *Häkkiansan laukaisuolion* vasemmalle ja oikealle puolelle ilmestyvät erilliset *Häkkiseinämät*: niihin osuessaan pelaaja ”kimpoaa” automaattisesti vastakkaiseen pyörimissuuntaan. Pelaajan on painettava välilyöntiä suurin piirtein samassa kohdassa, jossa Häkkiansan laukaisuolio oli aktivoitunut: tämän pystyy tunnistamaan kuuntelemalla Häkkiseinämien kimpoamisäänistä syntyvää taukoa, joka paljastaa koko ansan keskipisteen.



Kuva 10 Pelaaja on koskenut Häkkiansan laukaisuoliota ja joutunut kahden Häkkiseinämän väliin.



Kuva 11 Pelaaja löytänyt Siltaolion ja liikkuu lähemmäksi kehän keskipistettä.



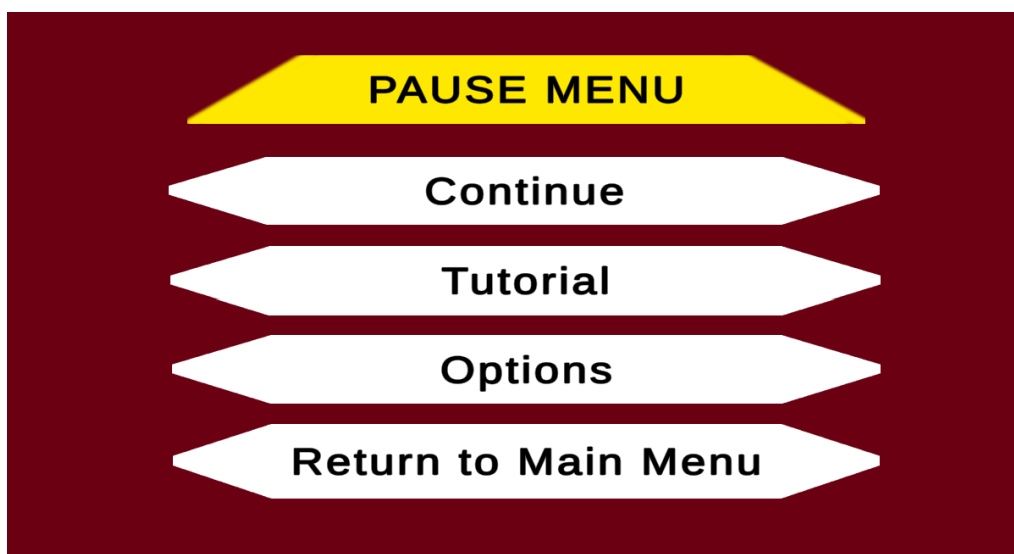
Kuva 12 Pelaaja on lähestymässä kehän ytimessä olevaa Metronomioliota. Pelaaja pyörii kelloonmyötäiseen suuntaan.

Molempien tasojen suunnittelun ja toteutuksen jälkeen aloitan pelin optimoinnin. Tässä tilanteessa poikkeen alkuperäisistä suunnitelmistani seuraavilla tavoilla: käytän *Freesound*-palvelun sijaan *Pixabayn* äänitehosteita, pois lukien itse kehittämäni robottiäänet, joita käytän vihollisissa ja ääniseessa (*LingoJam Robot Voice Generatorin* ja *Audacity*-editointityökalun avulla). Syy Pixabayn äänivalikoiman käyttöön *Freesoundin* sijasta on sen valikoima, jonka koen sopivan paremmin pelini visioon. Teen myös päätöksen olla käyttämättä lainkaan musiikkia, sillä peli oli mielestäni tarpeeksi tunnelmallinen ilman sitä. Jätän myös peliohjaintuen ja äänisyötemahdollisuuden pois projektistani, sillä en pystynyt huomioimaan niitä aikataulussani: peli on täten pelattavissa vähimmäisvaatimuksilla, eli näppäimistöllä sekä kaiuttimilla tai kuulokkeilla.

Seuraavaksi suunnittelen pelin *UI-elementit*, eli *pää- ja keskeytysvalikot* (*Main Menu* ja *Pause Menu*), jotka tukevat itse kehittämäni ruudunlukijärjestelmää. Tätä varten en tarvinnut erillisiä lisäosia: määrittelin päävalikon ja keskeytysvalikon sellaisiksi, että aina kun pelaaja navigoi niissä *W- ja S- tai ylä- ja alanuolinäppäintä* painaen, aktivoituu kerrallaan painikkeisiin liitetty äänitehoste, joka kertoo painikkeen nimen. Käytin ruudunlukijajärjestelmään *Narakeet*-ohjelman avulla luotuja äänileikkeitä.



Kuva 13 Pelin päävalikko.



Kuva 14 Pelin keskeytysvalikko.

Ruudunlukijajärjestelmän ohella toteutin pelin *varoitus-* ja *logotekstit*, *alku-* ja *lopetuselokuvat*, *opastusnauhoitteet* ensimmäiselle ja toiselle pelitasolle, lyhyen *kiitospuheen* sekä pelin *lopputekstit*: kaikki olivat täysin äänimuodossa. Elokuvien ja opastusnauhoitteiden editointiin käytin *HitFilm*-työkalua. Pelin kansikuvan toteutin *Blenderin*, *Gimpin* ja *Unityn* avustuksella: suunnittelin eksentrisen näköisen metronomin Blenderissä, jonka 3D-mallin siirsin Unityssä omaan pelikohtaukseen. Otin kohtauksesta kuvankaappauksen, jota jälkikäsittelin Gimpin avulla.

Lisäsin lopuksi peliin ajastimen, joka aktivoi valitun nauhoitteen sen perusteella, kuinka paljon peliaikaa on jäljellä minuutteina: ajastin ensimmäisen pelitason seitsenminuuttiseksi, toisen pelitason puolestaan kolmeminuuttiseksi. Kun aika päättyy, aktivoituu *Game Over* -ruutu, jolloin koko peli alkaa alusta.



Kuva 15 Pelin Game Over -ruutu, joka ilmestyy aikarajan päättyessä.

Koska halusin tehdä pelistäni täysin audiopainotteisen, lisäsin pelitasoihin erillisen värikalvon, joka estää näkemästä peliobjekteja. Tässä dokumentissa esitetyt kuvankaappaukset eivät sisällä kyseistä kalvoa. Lopullisessa pelissä on myös tilanteita, jossa pelaaja voi nähdä jotain: pelin alussa esiintyvät tekijätiedot sekä itse pelin logon kansikuvituksineen, jotka suunnittelin ja toteutin *Blender*- ja *Gimp*-ohjelmistojen avulla.



Kuva 16 Magentanvärinen kalvo, jota käytetään pelin ensimmäisessä ja toisessa tasossa, tarkoituksena peittää pelinäköymä.

4.3 Julkaisu- ja palautteenkeruuvaihe: peli eheytyy muiden havaintojen avulla

Julkaisen pelin *Itch.io-palvelussa*, jossa prosessi on nopea, helppo ja maksuton. Pelinkehitykseen liittyneiden aikatauluongelmien vuoksi *Steam-julkaisu* olisi saattanut viivästyttää pelin julkaisua merkittävästi. *The Metronome of Dissonancen* virallinen julkaisupäivä on 19. huhtikuuta 2023.

4.3.1 Ensimmäinen testipelaaja antaa palautetta

Vaikka *The Metronome of Dissonance* ei ole toistaiseksi kerännyt suuria latausmääriä, olen saanut palautetta eräältä normaalinäköiseltä pelaajalta. Hänen kritiikkinsä auttoi ymmärtämään tiettyjä epäselkeyksiä, joita en ottanut huomioon aiemmin: pelin alussa esiintyvä, tulkinnanvarainen äänielokuva oli pitkä ja epäselvä, mikä loi vaikutelman siitä, että peli olisi jäänyt jumiin bugin seurauksena. Korjasin ongelman lisäämällä äänielokuvan alkuun äänen "The audio movie starts now."

Listen and imagine” (”Äänielokuva alkaa nyt. Kuuntele ja kuvittele”). Tämä kannustaa pelaajia olemaan kärsivällisiä ja eläytymään pelin tunnelmaan.

Toinen tärkeä palaute liittyi pelin tutoriaaliäänitteisiin, jotka esiintyvät kahdessa muodossa: pidempänä versiona ennen varsinaisen pelitason alkua sekä lyhyempänä versiona keskeytysvalikon *Tutorial*-osiossa. Palautteen antanut pelaaja oli ohittanut pidemmän tutoriaalin ja kuunnellut lyhyemmän version keskeytysvalikosta, eikä hän ollut sen perusteella ymmärtänyt, mitä pelissä pitäisi tehdä. Korjasin ongelman lisäämällä keskeytysvalikkoon saman tutoriaaliäänitteen kuin mikä esiintyy ennen varsinaisen pelitason alkua: tein ainoastaan muutamia pieniä leikkauksia tauotuksessa ja sanavalinnoissa.

Pelin äänimaailma sai positiivista palautetta samalta palautteenantajalta.

4.3.2 Toinen testipelaaja antaa palautetta

Myöhemmin sain palautetta toiselta normaalinäköiseltä pelaajalta, joka auttoi minua löytämään muutaman epäkohdan pelistä. Kun pelaaja on säätänyt *keskeytysvalikon Options-ikkunassa* äänenvoimakkuutta ja sulkee sen välittömästi *Esc-näppäimellä* (ilman että palaisi valikon UI-painikkeiden kautta takaisin peliin), jää asetusvalikko vielä aktiiviseksi, vaikka se onkin jo muuttunut näkymättömäksi: täten kun pelaaja yrittää liikuttaa hahmoa, alkaa ruudunlukija toistaa asetusvalikossa esiintyvien painikkeiden ääniä. Tämä johtuu siitä, että olin unohtanut määritellä Options-ikkunan deaktivoituvaksi aina kun koko keskeytysvalikko suljetaan. Toinen havainto liittyi äänenvoimakkuuden muutosten tallentamattomuuteen aina kun pelissä vaihtuu kohta. Molemmat bugit on sittemmin korjattu.

Pelini sai samalta henkilöltä positiivista palautetta pelin ideasta ja toimintalogiikasta. Äänitehosteet saivat sen sijaan jonkin verran kehitysehdotuksia: pelaajahahmon kuolinääni voisi olla vähemmän vaimea, elämän heikentymisen voisi ilmaista erillisellä äänellä, jotkut äänitehosteista muistuttavat hieman toisiaan ja äänenvoimakkuuksien välillä on toisinaan huomattavia eroja.

4.4 Tavoitteiden toteutumisen arviointi

Henkilökohtaisia tavoitteitani olivat, että *The Metronome of Dissonance* on selkeä identiteetiltään ja toimintalogiikaltaan, se on julkaistu virallisessa kauppapaikassa, se on pelattavissa täysin ilman näköaistia ja se on koettu miellyttäväksi pelata erityisesti näkövammaisten tai -rajoitteisten pelaajien toimesta. Kolmen ensimmäisen tavoitteen osalta koen projektin olevan onnistunut: peli

heijastaa omaa tyyliäni, persoonallisuuttani ja mielenkiinnon kohteitani, ja se on nyt julkisesti kaikkien saatavilla. Olen myös pelannut pelin läpi onnistuneesti ilman näköaistini käyttöä. Palautetta peli on kerännyt tähän asti vähän, mutta se on auttanut peliä kehittymään. Projektini on ilmeisesti osoittanut kiinnostusta muutaman latauskerran ja muun muassa *Discord-keskustelupalvelussa* esitettyjen reaktioiden perusteella, mutta en voi vielä tarkalleen arvioida, ketkä kaikki ovat pelanneet peliä ja samalla pitäneet siitä.

Verrattuani lopputuotetta *videopelien yleisiin saavutettavuusohjeistuksiin* näkövammaisten osalta, huomaan suurimman osan ohjeistuksista täyttyneen: koska *The Metronome of Dissonance* on visuaalisista elementeistä riippumaton äänipeli, ei sen tarvitse täyttää *VR-tilaan* tai *3D-elementtien* näkemiseen liittyviä ohjeistuksia. Myöskään *hiirikursoriin* liittyvät ohjeistukset eivät ole tarpeen tässä pelissä, sillä siinä käytetään ainoastaan näppäimistöä.

Valikossa olevien *tekstielementtien koko ja jäsentely* on tehty mahdollisimman suuriksi ja selkeiksi, ja kaikki oleellinen teksti on aina sijoitettu pelaajan näköalueelle. Valikkojen ja tekstifonttien kokoja ei voi kuitenkaan vapaasti muuttaa. *Valikkopainikkeet* ovat selkeitä, vaikka ne voisivat olla enemmän erillään toisistaan. Kohdistettu valikkopainike on aina väriltään keltainen, vaikka väriä ei saisi yksinomaan käyttää elementtien tunnistamisessa: toisaalta ruudunlukija lukee jokaisen painikkeen ääneen, joten pelaajan ei tarvitse luottaa painikkeiden visuaaliseen puoleen. *Keskeytysvalikon* tummanpunainen tausta ei välttämättä erotu kunnolla painikkeista toisin kuin *päävalikon* musta tausta, mutta ruudunlukija kompensoi ongelmaa. Yksi puute, jota itsekin kadun jälkeinpäin, on lisätä *painikkeiden painamisesta* syntyvät äänitehosteet: ruudunlukija onneksi kompensoi tätä ongelmaa, sillä se lukee aina kohdistetun painikkeen nimen ja tarvittaessa ylimääräiset käyttöohjeet.

Vaikka pelihahmon mekaniikat ovat erilaiset tasoissa 1 ja 2, on niiden peliohjeet esitetty äänitutoriaaleissa selkeästi, mikä ennaltaehkäisee väärinkäsityksiä ja hämmennystä. Kameran liikkeisiin kohdistuvat muutokset eivät ole relevantteja visuaalittomassa äänipelissä: myöskään animaatiotehosteiden poiskytkemisen mahdollisuus ei ole tarpeen, vaikka sitä voisi teoriassa soveltaa pelin käynnistymisessä esiintyviin häivytysanimaatioihin. Ohjeistuksessa mainitut *surround-äänien* ja *binauraalisen äänen* käyttö on *The Metronome of Dissonance* keskiössä. Kaikki pelioliot ovat myös äänitehosteiltaan tarpeeksi erilaisia erottumaan toisistaan. Äänenvoimakkuus on säädettävissä pelissä mutta vain kokonaisvaltaisesti, sillä yksittäisten ääniraitojen kuuluvuutta ei voi valikoidusti muuttaa.

Pelin julkaisusivu Itch.io:ssa on käytettävissä ongelmitta tietokoneen ruudunlukijan avulla. Myös itse pelissä on oma *ruudunlukijajärjestelmänsä*, sillä jokainen valikkopainike ja pelikohtaukset selostetaan *esiäänitettyjen nauhoitteiden* avulla. *Ääniopastus* on pelissä osittain mukana, sillä pelaajan tehtävä selostetaan aina ennen varsinaisen pelitason alkua, jäljellä oleva peliaika ilmoitetaan

joka minuutin välien ja pelaajan ympärillä olevat äänet jakautuvat audiolaitteen eri puolille pelin tapahtumien mukaan. *GPS-opastuksen ja kompassipisteiden* konseptit on huomioitu pelissä, sillä pelaajan liikkumisjärjestelmä on yksinkertainen ja automatisoitu. Pelin tasoissa esiintyvä metro-nomi perustuu *äänikartan* konseptiin ja osaltaan kertoo pelaajan sijainnin ympäristössä, mutta sitä ei voine pitää tarpeeksi tehokkaana vastineena.

Koska pelissä on oma esinauhoitettu ruudunlukijajärjestelmänsä, en voi vaikuttaa siihen, että pelin asennusohjelma sisältäisi sellaista erikseen: kyseessä on *zip-tiedosto*, joka pitäisi pystyä avaamaan tietokoneen oman ruudunlukijan ohjeistuksella. Ohjeistus pelinsisäisestä *kuvailutulkkauksesta* toteutuu siinä mielessä, ettei mitään epäselvää jätetä tulkitsematta: salaperäisissä äänielokuvissa pelaaja luottaa omaan mielikuvitukseensa.

5 Pohdinta

5.1 Vastaukset tutkimuskysymyksiin ja kehitysideoita

Opinnäytetyöni tutkimuskysymykset liittyivät näkövammaisten ja -rajoitteisten henkilöiden huomiointiin videopeliä suunniteltaessa: *miten tarjota autenttinen pelikokemus näkövammaisille pelaajille? Mitkä ominaisuudet ovat tärkeitä visuaalisista aistihavainnoista riippumattoman pelikokemuksen toteuttamisessa? Mitä väärinkäsityksiä ja virhetilanteita voi ilmetä ei-visuaalisessa pelissä?*

Visuaalisista aistihavainnoista riippumattoman pelikokemuksen toteuttamisessa on erityisen tärkeää pitää pelaaja tietoisena siitä, mitä pelissä tapahtuu, ja miten hän voi vuorovaikuttaa siinä: tämän toteutumisessa kolmiulotteinen äänimaailma, pelaajan avustetut kontrollit, toisistaan selkeästi erottuvat äänitehosteet sekä opastava ruudunlukija ovat tärkeitä. Ei-visuaalisessa pelissä epäselvät tai liian samankaltaiset äänet voivat vaikuttaa pelaajan kykyyn hahmottaa ympäristöään tai pelin toimintalogiikkaa siitäkin huolimatta, että pelissä olisi yksityiskohtainen opastus eri äänien merkityksistä. Pelaajan tutustuttaminen pelin sääntöihin pienten interaktioiden kautta lienee parempi kuin pitkähkö tutoriaalivideo, vaikka sen voisikin toistaa uudelleen koska tahansa. Pelikokemukseen negatiivisesti vaikuttavia asioita voivat olla myös esimerkiksi epätasapaino äänitehosteiden keskinäisten voimakkuuksien välillä sekä pienistä yksityiskohdista viestivien tehosteiden puute. Myös pelin abstraktinomaisuus voi johtaa väärinkäsityksiin.

Autenttisen pelikokemuksen tarjoaminen näkövammaisille pelaajille vaatii kaikkein eniten asettautumista heidän asemaansa, jotka eivät kykene kokemaan peliä visuaalisesti täysin tai ollenkaan: kun pelaajaa avustavat tukitoiminnot on suunniteltu jo projektin varhaisessa kehitysvaiheessa, voi ne liittää erottamattomaksi osaksi pelin identiteettiä. Tämä ei ainoastaan saisi näkövammaisia tai -rajoitteisia pelaajia tuntemaan olonsa huomioiduiksi vaan myös murtaisi saavutettaviin videopeleihin, erityisesti äänipainotteisiin teoksiin, kohdistuvia mahdollisia ennakkoluuloja.

Tulevien tutkimusten kannalta projektia voisi kehittää eri tavoin: miten näkövammaisille suunniteltu autenttinen pelikokemus muuttuisi, jos projektissa olisivat kaikki visuaaliset elementit näkyvissä? Kuinka paljon fonttikokojen, yksittäisten ääniraitojen voimakkuuksien ja pelikappaleiden välisten kontrastien säätöasetusten mahdollistaminen vaikuttaisi pelikokemukseen? Voisiko pelin pitää epävisuaalisena mutta silti lisätä siihen kuulovammaistuen esimerkiksi *visuaalisten äänitehosteiden ja kuvailevan tekstityksen* muodossa? Miten *mikrofonisyötteen* ja *ohjainvärinätehosteiden* hyödyntäminen helpottaisi näkövammaisen pelaajan autenttisen kokemuksen saavuttamista, ja olisiko niistä hyötyä myös muita vammoja tai rajoituksia omaaville? Kuinka hyvin *interaktiivinen tutoriaalitaso* pelkän äänivideon sijaan ehkäisisi pelaajien väärinkäsityksiä siitä, mitä tämän tulee tehdä? Kuinka paljon peli muuttuisi, jos sen abstraktinomaiset äänitehosteet vaihdettaisiin

luonnollisempiin, oikean maailman tapahtumia esittäviin ääniin? Miten pelin taustatarina ja yleinen identiteetti vaikuttavat saavutettavuusominaisuuksien suunnitteluun? Kuinka laajasti saavutettava pelistä on mahdollista tehdä?

5.2 Henkilökohtaisia suunnitelmia

The Metronome of Dissonance -pelin jatkokehitysmahdollisuuksia on useita, koska se on vielä demovaiheessa: ensiksi voisin varmistaa, että peli täyttäisi ne puutteet, jotka havaitsin vertailemalla lopputulosta videopelien saavutettavuusohjeistuksiin. Näitä ovat lähinnä valikko- ja tekstielementtien vapaavalintaisen skaalauksen sekä monipuolisen äänenvoimakkuussäätimen mahdollistaminen, valikkopainikkeiden *painamisesta* syntyvien äänitehosteiden luominen, sekä hieman tehokkaampi *äänikarttatoiminnallisuus*. Varmistettuani pelin olevan mahdollisimman saavutettava näkövammaisille tai -rajoitteisille pelaajille, voin laajentaa pelin saavutettavuutta: kuulovammaisille soveltuvassa versiossa peli voisi sisältää *visuaalisia äänitehosteita* sekä esiaänitettyjen nauhoitteiden *tekstityksiä*. Myöhemmin voisin lisätä myös tuen peliohjaimille, etenkin sellaisille, jotka on suunniteltu erilaisia liikunta- ja motoriikkarajoituksia omaaville henkilöille. Ajatuksena on, että The Metronome of Dissonance olisi kaikille pelaajille tasavertainen, aistirajoista irtautuva pelikokemus.



Kuva 17 The Metronome of Dissonance -pelin kansikuvitus. Metronomi on mallinnettu Blender-ohjelman avulla, kuvattu Unity-pelikohtauksessa ja jälkikäsitelty Gimpin avulla.

Lähteet

Audacity. Home. Luettavissa: <https://www.Audacityteam.org/>. Luettu: 16.5.2023.

Blender s.a. About. Luettavissa: <https://www.blender.org/about/>. Luettu: 16.5.2023.

Blue, R. 2021. Accessibility in gaming: there's more work to be done. Luettavissa: <https://www.techradar.com/news/accessibility-in-gaming-theres-more-work-to-be-done>. Luettu: 16.5.2023.

Dolan, H. 2019. The Ultimate Beginners Guide To Game Development In Unity. Luettavissa: <https://www.freecodecamp.org/news/the-ultimate-beginners-guide-to-game-development-in-unity-f9bfe972c2b5/>. Luettu: 16.5.2023.

Educba s.a. What is Gimp. Luettavissa: <https://www.educba.com/what-is-gimp/>. Luettu: 16.5.2023.

Eng, D. s.a. Accessibility in Games. Luettavissa: <https://www.universityxp.com/blog/2020/3/12/accessibility-in-games>. Luettu: 16.5.2023.

Freesound s.a. Frequently Asked Questions. Luettu: <https://freesound.org/help/faq/>. Luettu: 16.5.2023.

Game Accessibility Guidelines, s.a. Full List. Luettavissa: <https://gameaccessibilityguidelines.com/full-list/>. Luettu: 16.5.2023.

Gerli, D. 2023. The Making of | Blindness, The 90's Breakthrough In Video Game Accessibility That Time Forgot. Luettavissa: <https://www.timeextension.com/features/the-making-of-blindness-the-90s-breakthrough-in-video-game-accessibility-that-time-forgot>. Luettu: 16.5.2023.

Heath, H. 2021. 13 Recent Games That Are Pushing Accessibility Features Forward. Luettavissa: <https://www.thegamer.com/games-with-accessibility-features/>. Luettu: 16.5.2023.

Incompetech s.a. a. Contact Me. Luettavissa: <https://incompetech.com/email.html>. Luettu: 16.5.2023.

Incompetech s.a. b. Frequently Asked Questions. Luettavissa: <https://incompetech.com/music/royalty-free/faq.html>. Luettu: 16.5.2023.

Itch.io s.a. About itch.io. Luettavissa: <https://itch.io/docs/general/about>. Luettu: 16.5.2023.

Khan, Z. 2023. What is Steam: The popular gaming platform explained. Luettavissa: <https://www.androidauthority.com/what-is-steam-3254981/>. Luettu: 16.5.2023.

LingoJam s.a. Robot Voice Generator. Luettavissa: <https://lingojam.com/RobotVoiceGenerator>.
Luettu: 16.5.2023.

Lenstore, s.a. The Most Accessible Games for the Visually-Impaired. Luettavissa:
<https://www.lenstore.co.uk/research/videogame-accessibility/>. Luettu: 16.5.2023.

Link, J. 2021. Triple-A Game Publishers Are Finally Taking Accessibility Seriously. Luettavissa:
<https://builtin.com/design-ux/video-game-accessibility-blind-low-vision>. Luettu: 16.5.2023.

Lupetti, M. s.a. Seen and heard: making games for the visually impaired. Luettavissa:
<https://whynowgaming.com/seen-and-heard-making-games-for-the-visually-impaired/>.
Luettu: 16.5.2023.

Narakeet 2021. Text To Speech: 5 tips for video narration. Luettavissa:
<https://www.narakeet.com/blog/text-to-speech-videos.html>. Luettu: 16.5.2023.

Narakeet s.a. a. Text to Speech Voices. Luettavissa: <https://www.narakeet.com/languages/>.
Luettu: 16.5.2023.

Narakeet s.a. b. Create high-quality text to speech audio. Luettavissa:
<https://www.narakeet.com/docs/how-to/create-high-quality-text-to-speech-audio.html>.
Luettu: 16.5.2023.

Paris, S. 2022. HitFilm 2022. Luettavissa: <https://www.techradar.com/reviews/hitfilm-2022>.
Luettu: 16.5.2023.

Pixabay s.a. FAQ. Luettavissa: <https://pixabay.com/service/faq/>. Luettu: 16.5.2023.

Pusey, M. 2019. Lost and Hound: Video games for the vision impaired. Luettavissa:
<https://phys.org/news/2019-06-lost-hound-video-games-vision.html>. Luettu: 16.5.2023.

RNIB, s.a. Design For Every Gamer. Luettavissa:
<https://www.rnib.org.uk/living-with-sight-loss/assistive-aids-and-technology/everyday-tech/tv-audio-and-gaming/design-for-every-gamer/>. Luettu: 16.5.2023.

Tsuburaya s.a. Zetton. Luettavissa: <https://tsuburaya-prod.com/encyclopedia/zetton>. Luettu:
16.5.2023.

Unity Documentation 2023 a. Work with multiple scenes in Unity. Luettavissa:
<https://docs.unity3d.com/Manual/MultiSceneEditing.html>. Luettu: 16.5.2023.

Unity Documentation 2023 b. CharacterController. Luettavissa:

<https://docs.unity3d.com/ScriptReference/CharacterController.html>. Luettu: 16.5.2023.

Xsolla s.a. Self-Publish on Steam: The Ultimate Guide. Luettavissa: <https://xsolla.com/blog/self-publish-on-steam-the-ultimate-guide>. Luettu: 16.5.2023.

Liitteet

Liite 1. Videopelien saavutettavuusohjeistukset

Perustason ohjeistukset:

- *elementtien ei pidä olla tunnistettavissa vain värin perusteella*
- *3D-pelien näkökenttä on määritettävä sopivalle vakiolaajuudelle*
- *virtuaalitodellisuuspeleissä (VR) on vältettävä tehosteita, jotka voivat johtaa pahoinvointiin*
- *tekstielementtien fontin vakiokoon on oltava sellainen, että teksti on helposti luettavissa*
- *tekstiformaation on oltava selkeää ja yksinkertaista*
- *näyttöelementtien ja taustan välisen kontrastin on oltava korkea*
- *pelissä esiintyvät interaktiiviset elementit, tai näytöllä käytettävät ohjainpainikkeet, on oltava kosketusalueiltaan suuria sekä sopivalla etäisyydellä toisistaan*

Keskitason ohjeistukset:

- *3D-pelin näkökentän laajuuden olisi oltava vapaasti säädettävissä*
- *kameran ja pelihahmon ohjaukseen kohdistuvia muutoksia on vältettävä, tai ne pitäisi olla kytkettävissä pois päältä*
- *pelissä käytettävän äänen tulisi olla Surround-ääntä*
- *pelin taustalla olevia animaatiotehosteita pitäisi pystyä halutessaan piilottamaan tai kytke-
mään pois päältä*
- *mobiilipeleissä olisi oltava tuki puhelimen ruudunlukijalle*
- *pelin sisäisen kontrastin olisi oltava vapaasti säädettävissä*
- *peliohjeiden ja tapahtumien aktivoitumisesta syntyvien ääni- ja musiikkisignaalien on oltava yksilöllisiä ja toisistaan selkeästi erottuvia*
- *jos pelissä käytetään kursoria, kuten hiirtä tai tähtäintä, on sen ikonin ulkonäön ja värin ol-
tava valittavissa eri vaihtoehtojen väliltä*
- *jos pelissä on interaktiivisia objekteja, on se pystyttävä päättämään olioiden olemuksesta*
- *jos pelillä on ohjekirja tai verkkosivut, on niiden oltava helposti luettavissa ruudunlukijalla*
- *äänitehosteiden, puheen ja taustamusiikin voimakkuuksien pitäisi olla yksilöllisesti säädet-
tävissä ja mykistettävissä*
- *ruudulla näkyvä informaatio tulisi pitää pelaajan katsealueen sisällä*
- *pelin ruudulla esiintyvien käyttöliittymien tulisi olla kooltaan vapaasti säädettävissä*

Edistyneen tason ohjeistukset:

- *tekstisisällön fonttikoon tulisi olla vapaasti säädettävissä*
- *pelissä tulisi olla kaikuluotaukseen perustuva äänikartta*

- *pelin tekstielementeille, niin itse pelissä, valikossa kuin itse asennusohjelmassa, tulisi olla esinauhoitettut ääniselostukset*
- *pelissä tulisi olla äänitetty GPS-opastus*
- *pelissä tulisi olla mahdollisuus kompassipisteiden käyttöön, mikä helpottaa pelihahmon kulkemista oikeaan suuntaan*
- *pelin tärkeiden toiminnallisuuksien tulisi olla käytettävissä ohjainten ja näppäinten avulla*
- *pelin valikossa ja asennusohjelmassa tulisi olla tuki ruudunlukijalle*
- *kaikille peliobjekteille ja tapahtumille tulisi antaa toisistaan poikkeavat ääni- ja musiikkitehosteet*
- *äänityksessä tulisi käyttää binauraalista ääntä*
- *pelissä tulisi olla kuvailutulkkaus*

(Game Accessibility Guidelines s.a.)

Liite 2. Valmis esimerkki pelin toimintalogiikasta

Taso 1:

- Pelaaja aloittaa sokkelon ensimmäisestä risteyksestä, ja hänen ympärillään on kolme polkua: yksi vasempaan suuntaan, toinen suoraan eteenpäin ja kolmas oikealle. Pelikamera on lukittunut pelaajaan tietyistä suunnasta ja etäisyydestä, eikä se muutu pelaajan kulkusuunnasta riippumatta.
- Pelaaja suorittaa *kuuntelutoiminnon*, jolla hän aktivoi kaikuluotaimen. Pelaaja kuulee määränpäänsä (*metronomin*) olevan jossain kaukana etuoikealla.
- Pelaaja haluaa mennä suoraan eteenpäin, joten hän painaa *W-näppäintä*. Pelihahmo lähtee liikkeelle suoraan eteenpäin. Pelihahmo kääntyy automaattisesti sellaisissa mutkissa, jotka eivät sisällä risteyksiä – tässä esimerkissä pelaaja kääntyy yhdestä mutkasta oikealle ja seuraavasta vasemmalle. Kävelyäänisensori on asetettu pelihahmon oikealle, vasemmalle ja eteen, joten pelaaja kuulee selkeästi pelihahmon kävelysuunnan.
- Pelihahmo lopettaa automaattisen kävelynsä. Äänisignaalin perusteella pelaaja on saapunut uuteen risteykseen. Hän suorittaa ääniskannauksen uudestaan: tällä kertaa määränpää on jossain pelaajan oikealla puolella.
- Pelaaja päättää mennä risteyksestä oikealle, joten hän painaa *D-näppäintä*. Pelihahmo menee oikealle ja jatkaa automaattisesti kävelyään, kääntyen yhdestä mutkasta vasemmalle.
- Pelihahmo lopettaa automaattisen kävelynsä. Tällä kertaa hän on joutunut *vihollisrobottien* piirittämäksi. Pelaaja kuulee kolmen robotin äänet ympärillään: vasemmalla puolella oleva robotti sanoo *Shita*, oikealla puolella oleva *Naka* ja edessä oleva *Ue*. Viholliset hyökkäävät yksi kerrallaan viiden sekunnin välein.
- Pelaaja valitsee *ääniasevalikoimastaan* (*Q- ja E-näppäimillä*) *ääniaseen*, joka kuulostaa kimakalta *Shita*-ääneltä, ja painaa vasenta nuolinäppäintä ampuakseen ääniaseella vasemmalla puolellaan olevaa robottia. Isku tehoaa ja robotti katoaa.
- Pelaaja valitsee *ääniasevalikoimastaan* toisen *ääniaseen*, joka kuulostaa kimakalta *Naka*-ääneltä, ja painaa oikeaa nuolinäppäintä ampuakseen ääniaseella oikealla puolellaan olevaa robottia. Isku tehoaa ja robotti katoaa.
- Pelaaja valitsee *ääniasevalikoimastaan* kolmannen *ääniaseen*, joka kuulostaa kimakalta *Ue*-ääneltä, ja painaa ylänuolinäppäintä ampuakseen ääniaseella edessään olevaa robottia. Isku tehoaa ja robotti katoaa.
- Pelaaja on nyt päihittänyt kaikki kolme robottia ja jatkaa täten automaattisesti matkaansa. Jos pelaaja olisi saanut kolme iskua vihollisilta, hän olisi menettänyt kaikki elämänpisteensä ja palannut takaisin siihen risteykseen, jossa oli ollut ennen taisteluun joutumista.
- Liikkuessaan eteenpäin pelaaja kuulee nopean *piippausäänen*: hän on lähestymässä *varjoansaa*. Pelaaja painaa nopeasti *välilyöntiä*, jolloin pelihahmon ympärille ilmestyy

suojakenttä: tämä estää pelaajaa tuhoutumasta varjoansan läpi kulkiessaan. Jos pelaaja ei aktivoinut suojakenttää, tämä menettää elämäpisteensä ja palaa takaisin siihen risteykseen, jossa oli ollut ennen varjoansa joutumista.

- Pelaaja selvisi varjoansasta ja jatkaa eteenpäin. Pelihahmo pysähtyy uuteen risteykseen. Pelaaja suorittaa ääniskannauksen: nyt määränpää kuuluu voimakkaana suoraan hänen vasemmalta puoleltaan. Pelaaja menee vasemmalle painamalla *A-painiketta*. Pelihahmo kävelee vasemmalle ja kävelee kohti metronomia– pelaaja on löytänyt määränpäänsä.
- Tässä kentässä on seitsemän minuutin aikaraja: peli päättyy, jos pelaaja ei ole siihen mennessä löytänyt määränpäätä.

Taso 2:

- Pelaaja aloittaa *kehän* uloimmalta reunalta, ja hän pyörii automaattisesti ytimen ympäri kellonmyötäiseen suuntaan. Ytimestä kuuluu automaattisesti *metronomin* tikitysäni.
- Pyöriessään kehässä, pelaaja kuulee ympärillään kalterien kolahduksen: hän on joutunut *hökkimäiseen ansaan*. Pelaaja törmää vasemmalla puolellaan olevaan *hökkiseinämään*, josta kuuluu *kimpoamisääni*: tämä saa pelaajan jatkamaan pyörimistä vastakkaiseen suuntaan, kunnes hän törmää oikealla puolellaan olevaan *hökkiseinämään*. Tämä jatkuu niin kauan, kunnes pelaaja painaa välilyöntipainiketta ollessaan häkkiansan keskipisteessä. Äänipainotteisessa pelissä pelaaja pystyy tunnistamaan tämän keskipisteen kuuntelemalla häkkiseinämien kimpoamisäänten välissä olevaa taukoa.
- Pelaaja painoi välilyöntipainiketta oikealla hetkellä, jolloin hän vapautuu ansasta. Pelaaja jatkaa pyörimistään joko kellonmyötäiseen tai -vastaiseen suuntaan riippuen siitä, kummanpuoleisesta häkkiseinämästä pelaaja oli kimpoutunut ennen vapautumistaan.
- Pelaaja kuulee oikealla puolellaan nopean merkkiäänen: hän on lähestymässä *siltamaista* oliota, joka siirtää pelaajan lähemmäksi kehän ydintä. Pelaajan on kuitenkin painettava oikeaa painiketta (joko *W-*, *A-* tai *D-painiketta*) sen perusteella, mistä suunnasta hän kuuli merkkiäänen. Koska merkkiääni tuli oikealta, pelaajan on painettava *D-painiketta*. Vasemmalta kuuluva ääni olisi edellyttänyt A-painikkeen painamista, yläpuolelta kuuluva ääni puolestaan W-painikkeen painamista. Pelaaja kuulee uuden äänimerkin, joka kertoo onnistumisesta: hän kulkeutuu sillan kautta sisemmälle reunalle ja jatkaa pyörimistä kellonmyötäiseen suuntaan.
- Yllä olevat vaiheet jatkuvat niin kauan, kunnes pelaaja on päässyt kehän sisäisimpään ytimeen metronomin luo.
- Tässä kentässä on kolmen minuutin aikaraja: koko peli päättyy, jos pelaaja ei pääse metronomin luo siinä ajassa.

Liite 3. Pelin prototyyppi- ja viimeistelyvaiheen aikataulutus

Viikko 1: Pelin ympäristön ja pelihahmon perustoiminnot

- 3D-ympäristön (*sokkelo ja kehä*) luominen
- Pelaajan *liikkumisjärjestelmän* kehitys
- Pelaajan *taistelujärjestelmän* kehitys
- Pelaajan *elämäpistejärjestelmän* kehitys
- Pelaajan *respawn-toiminnon* luominen ("henkiinherätystoiminto")

Viikko 2: Pelaajan interaktioiden kehitys

- Peliolioiden (*risteyskohdat, määränpää*) luominen ja muuttaminen vuorovaikutteisiksi
- Pelaajan *kuuntelutoiminnon* (kaikuluotaukseen perustuva äänikartta) luominen
- *Ansojen* ja *vihollisten* luominen
- Vihollisten *taistelu-*, *vahinko-* ja *tuhoutumistoimintojen* luominen
- *Ansojen vahinkotoimintojen* luominen

Viikko 3: Erikoiskentän viimeistely sekä äänimaailman ja valikkojen luominen

- *Taustamusiikin* ja *äänitehosteiden* lisääminen
- Äänitehosteiden muuttaminen kolmiulotteisiksi
- *Erikoistason* (*kehä*) pelilogiikan viimeistely
- *Valikkojen* sovittaminen *ruudunlukujärjestelmää* varten
- *Ruudunlukujärjestelmän* kehittäminen ja omien *äänitteiden* luominen

Viikot 4 ja 5: Ensimmäinen ja toinen viimeistelyvaihe

- *Alku- ja loppuanimaation* sekä *tutoriaaliosuuden* luominen
- Kohtausten välisten siirtymien luominen
- valikko → peli alkaa → tutoriaali → pelitaso → peli päättyy → valikko
- *Äänitysten* tekeminen tutoriaaliin
- Pelin *markkinointimateriaalin* luominen
- Aiemmilta viikoilta keskeneräisiksi jääneiden toimintojen viimeistely

Liite 4. Luettelo peliä varten kehittämistäni C#-skripteistä

PlayerController.cs:

- Kiinnitetään *Pelaajaoliioon* 1. pelitasossa
- Tarkistaa pelaajan ympäristön: jos tämä on risteyksessä, tämä voi liikkua eteenpäin, vasemmalle tai oikealle painamalla joko *W*-, *A*- tai *D*-painiketta
- Kutsuu Pelaajaoliiossa olevaa *CharacterController-komponenttia* ja liikuttaa pelaajaa oikeaan suuntaan sen perusteella, mitä painiketta on painettu
- Kun Pelaajaolio liikkuu tiettyyn suuntaan, aktivoituu kävelyäänitehoste yhdestä Pelaajaoliioon liitetystä *PlayerWalkSensor-olioista* sen perusteella, mihin suuntaan tämä kulkee
- Pysäyttää Pelaajaolion erillisen funktion kautta aina kun pelaaja osuu *FightStopTrigger*- tai *Checkpoint-olioon*
- Osuessaan *NewArrowSign -nuoliolioon*, kääntää pelaajan uuteen suuntaan sen perusteella, mikä on *NewArrowSign -nuoliolion* rotaatioarvo z-akselissa.
- Tallentaa liikesuunnan omaksi *Vector3-oliokseen*, jotta Pelaajaolio osaa jatkaa matkaansa oikeaan suuntaan aina kun pelaajan on liikuttava automaattisesti (aina kun taistelu vihollisia vastaan päättyy)
- Aktivoi *PrepareForDisappearance -funktion HealthManagerin* kutsusta aina kun pelaajan elämänpisteet ovat nollassa: tällöin Pelaajaolio nousee yläilmoihin. Tämä on tehty, jotta *FightStopTrigger-oliiossa* oleva laukaisin tunnistaisi Pelaajaolion poistuvan päältä ja täten keskeyttäisi *vihollisobjektien* toiminnot.

SpecialController.cs:

- Korvaa *PlayerController-skriptin* 2. pelitasossa
- Pyörittää pelaajaa automaattisesti tietyn keskipisteen ympäri kellonmyötäiseen suuntaan: tässä tapauksessa keskiössä on *Metronome-peliolio*
- Jos pelaaja osuu *Cage-olioihin*, tämän pyörimissuunta muuttuu päinvastaiseksi
- Kun pelaaja on osunut *Bridge-olioon* ja painanut oikeaa suuntapainiketta (*W*-, *A*- tai *D*-painike), tämä lopettaa pyörimisen ja liikkuu kohti sillan toisella puolella olevaa *BridgeEnd-oliota* kohti. Tämän jälkeen pelaaja jatkaa pyörimistä kellonmyötäiseen suuntaan.

PlayerCamera.cs:

- Kiinnitetään Pelaajaolioon ensimmäisessä ja toisessa pelitasossa
- Ottaa kohteekseen Pelaajaolion ja seuraa tätä valitulta etäisyydeltä x-, y- ja z-akselissa

AudioFollow.cs:

- Kiinnitetään Pelaajaolioon ensimmäisessä pelitasossa
- Pitää huolta siitä, että *AudioListener-olio* seuraa Pelaajaoliota: tämän avulla pelaaja kuulee tarkasti kaikki äänitehosteet Pelaajaolion ympärillä, kuin jos AudioListener olisi kiinni *Kameraoliossa (Main Camera)*

Crossroads.cs:

- Kiinnitetään *Crossroads-olioon* ensimmäisessä pelitasossa
- Kun pelaaja törmää Crossroads-olioon, käynnistyy äänitehoste, joka kuuluu vuorollaan pelaajan vasemmalla puolella, oikealla puolella sekä molemmilla puolilla
- Samalla skripti varmistaa, että Crossroads-oliossa oleva äänitehoste ei toistu koko aikaa pelaajan ollessa siinä: sen on aktivoiduttava vain kerran risteykseen tultaessa

DetectPlayer.cs:

- Kiinnitetään *NewArrowSign-olioon* ensimmäisessä pelitasossa
- Aina kun NewArrowSign-olioon osuu Pelaajaolio, tämä lähettää parametrinä tiedon omasta rotaatiostaan z-akselissa suoraan PlayerController-skriptin funktioon, joka vaihtaa Pelaajaolion liikesuunnan NewArrowSign-olion osoittamaan suuntaan

Checkpoint.cs:

- Kiinnitetään *Crossroads-olioon* ensimmäisessä pelitasossa
- Kun pelaaja törmää *Checkpoint-olioon*, tapahtuu seuraavat asiat:
 - o Pelaajaolion *PlayerController-skripti* tallentaa Checkpoint-olion sijainnin *Vector3-oliona*
 - o Pelaajaolion *PlayerController-skripti* pysäyttää Pelaajaolion liikkeen
- Kun pelaaja menettää elämäpisteensä, tämä palautuu automaattisesti takaisin viimeksi keräämänsä Checkpoint-olion paikalle

HealthManager.cs:

- Kiinnitetään *Pelaajaolioon* 1. pelitasossa
- Määrittää *liukulukuarvot täysille elämäpisteille ja nykyisille elämäpisteille*
- Vähentää pelaajan elämäpisteitä aina kun jokin muu olio (*Enemy, Shadow, DeathZone*) osuu pelaajaan: aktivoi myös eri äänitehosteet, kun pelaaja ottaa vahinkoa tai menettää kaikki elämäpisteensä
- Kutsuu Pelaajaoliossa olevan *PlayerController-skriptin PrepareForDisappearance-funktiota* aina kun pelaaja menettää kaikki elämäpisteensä

Respawn.cs:

- Kiinnitetään *RespawnManager-olioon* 1. pelitasossa
- Kun pelaaja menettää kaikki elämäpisteensä, aktivoituvat Respawn-skriptissä seuraavat toiminnot:
 - o Pelaaja poistetaan näkyvistä
 - o Kutsutaan *Pelaajaolion PlayerController-skriptissä* olevaa funktiota, joka nostaa pelaajan yläilmoihin
 - o Pelaajan fysiikkaominaisuuksia säätelevä *CharacterController* kytketään pois päältä
 - o Kutsutaan Pelaajaolion *HealthManager-skriptissä* olevaa funktiota, joka palauttaa pelaajan elämäpisteet täysille
- Pelaajaolion uudeksi sijainniksi määritellään tämän viimeksi keräämänsä *Checkpoint-olion* koordinaatit
- Pelaaja palautetaan näkyviin, ja tämän *CharacterController* kytketään takaisin päälle
- Lopuksi Respawn-skripti kytkee kaikki pelitasossa olevat *Enemy-oliot* takaisin päälle: täten Pelaaja joutuu kohtaamaan uudestaan ne viholliset, joille oli viimeksi hävinnyt

FightStop.cs:

- Kiinnitetään *FightStop-olioon* 1. pelitasossa
- Sisältää *Array-taulussa* tiedot ympärillään olevasta kolmesta vihollisoliosta (*Enemy*)
- Tarkistaa *Pelaajaolion* läsnäolon ja aktivoi sen perusteella Pelaajaoliossa olevan *Sound-WeaponSelector-olioon* kiinnitetyn *SoundFight-skriptin*.
- Aktivoi myös siihen liitettyjen vihollisolioiden *EnemyManager-skriptissä* olevat *hyökkäysfunktiot*

- Laskee ajoittain siihen kuuluvien vihollisten olemassaolon. Jos deaktivoitujen vihollisten lukumäärä on 3, se kutsuu SoundFight-skriptissä olevaa *EndSoundFight-funktiota*, joka päättää taistelun ja lähettää pelaajan taas liikkeelle

SoundFight.cs:

- Kiinnitetään *Pelaajaoliassa* olevaan *SoundFightSelector-oliioon* 1. pelitasossa
- Aktivoituu *FightStop-olion* kutsusta
- Ollessaan aktivoitunut, antaa pelaajalle mahdollisuuden painaa *Q- ja E-painikkeita*: tämä aktivoi kerrallaan yhden kolmesta eri *SoundWeapon-äänitehosteesta*
- Jos pelaaja on valinnut SoundWeapon-äänitehosteen ja painaa vasenta, oikeaa tai ylänuolinäppäintä, aktivoituu erillinen *SoundBeam-äänitehoste* joko pelaajaolion vasemmalla, oikealla tai edessä olevan SoundWeaponSensor-oliosta. Samalla aktivoituu erillinen *Shoot-funktio*, joka tarkistaa erillisen *Raycast-fysiikkaominaisuuden* avulla, onko pelaajan nuolinäppäimellä osoittamassa suunnassa vihollisoliota. Jos pelaajan ampumissuunnassa oli vihollinen, jonka äänitehosteen nimessä on sama numero kuin pelaajan sillä hetkellä aktivoimassa SoundWeapon-äänitehosteessa, vihollisoliion *EnemyManager-skripti* valmistautuu vihollisoliion tuhoutumiseen. Muussa tapauksessa mitään ei tapahdu.
- Ampumisen jälkeen pelaaja kokee lyhyen jäähdytysvaiheen, jolloin hän ei pysty aktivoimaan Shoot-funktiota nuolinäppäimillä. Muutaman sekunnin kuluttua tämä on taas mahdollista

EnemyManager.cs:

- Kiinnitetään *vihollisoliioon (Enemy)* 1. pelitasossa
- Tarkistaa erikseen, onko *Pelaajaolio* vihollisen alueella
- Valmistautuu hyökkäykseen (*FightStop-skriptin* aktivoimana)
- Jos hyökkäys on aktivoitu FightStop-skriptin kautta, tapahtuu seuraavaa:
 - o Niin kauan kuin pelaaja on vihollisen edessä, ja tämän elämäpisteet eivät ole nollassa, vihollinen arpoo itselleen äänitehosteen kolmesta vaihtoehdosta, jotka on tallitettu enemyIdleSounds-tauluun
 - o Vihollinen odottaa muutaman sekunnin ja aktivoi arpomansa äänitehosteen
 - o Odotettuaan vielä muutaman sekunnin, vihollinen aktivoi erillisen hyökkäysäänitehosteen merkiksi hyökkäyksen alkamisesta

- Jos vihollinen on vielä elossa, ja pelaaja on vielä tämän edessä, kutsuu vihollinen Pelaajaoliassa olevan *HealthManager-skriptin* funktiota, joka vähentää pelaajan elämäpisteitä 40:llä (noin kolmasosa kokonaiselämäpisteistä)
- Jos vihollinen on saanut osuman (tarkistetaan SoundFight-skriptin kautta), se valmistautuu omaan tuhoutumiseensa: ensiksi se muutetaan näkymättömäksi (kyttämällä *MeshRenderer-komponentin* pois päältä), sitten vihollisesta aktivoituu tuhoutumisääni, ja lopulta koko vihollisolio deaktivoidaan.

DamageManager.cs:

- Kiinnitetään kaikkiin niihin objekteihin, joiden halutaan antaa vahinkoa *Pelaajaolille* (*Enemy, Shadow, Deathzone*) 1. pelitasossa
- Aktivoi Pelaajaoliassa olevan *HealthManager-skriptin* funktion, joka vähentää parametrinä annetun lukuarvon verran elämäpisteitä pelaajalta

ShieldSpot.cs:

- Kiinnitetään *ShieldSpot-olioon* 1. pelitasossa
- Aktivoi *ShieldManager-skriptin* toiminnallisuudet aina kun *Pelaajaolio* koskettaa sitä

ShieldManager.cs:

- Kiinnitetään *ShieldSpot_and_Shadow -oliokokonaisuuteen* 1. pelitasossa
- Aktivoi erillisen ääniefektin *ShieldSpot-olion* käskystä, ja samalla antaa *pelaajalle* mahdollisuuden painaa välilyöntinäppäintä
- Jos pelaaja painaa *välilyöntinäppäintä*, aktivoituu erillinen kilpiääniefekti, ja samalla deaktivoituu *Shadow-olio*: pelaaja pääsee siis turvallisesti Shadow-olion ohi
- Jos pelaaja ei paina välilyöntinäppäintä ja törmää edessään olevaan Shadow-olioon, aktivoituu Shadow-oliassa oleva *DeathZone-skripti*

DeathZone.cs:

- Kiinnitetään sekä *Shadow-olioon* että erilliseen *DeathZone-olioon* 1. pelitasossa
- Sama toimintaperiaate kuin *DamageManager-skriptillä*, mutta se on suunniteltu viemään kaikki elämäpisteet *Pelaajaolilta*

BridgeManager.cs:

- Kiinnitetään *Bridge-oliossa* olevaan *BridgeArrowSign-olioon* 2. pelitasossa
- Jos pelaaja osuu *BridgeArrowSign-olioon*, *BridgeManager-skripti* aktivoi äänitehosteen, joka kuuluu *Pelaajaolion* sijainnin mukaan joko pelaajan vasemmalta puolelta, oikealta puolelta tai molemmilta puolilta (eli edestäpäin)
- Jos pelaaja painaa oikeaa painiketta, kutsuu *BridgeManager-skripti* *Pelaajaolion* funktiota, joka saa tämän lopettamaan pyörimisen ja kulkeutumaan kohti *Bridge-olion* päässä olevaa *BridgeEnd-oliota*. Samalla *BridgeManager* aktivoi *Bridge-oliosta* onnistumista kuvastavan äänitehosteen

BridgeEnd.cs:

- Kiinnitetään *BridgeEnd-olioon* 2. pelitasossa
- Kun *Pelaajaolio* koskettaa *BridgeEnd-oliota*, tämä jatkaa pyörimistä kellonmyötäiseen suuntaan *SpecialController-skriptin* mukaisesti

CageTrigger.cs:

- Kiinnitetään *CageTrigger-oliokokonaisuuteen* 2. pelitasossa
- Kun *Pelaajaolio* koskettaa *CageTriggeriä*, kytketään *CageTriggeriin* kuuluvat *Cage-oliot* päälle, estäen täten Pelaajan pyörimisen eteenpäin
- Kun pelaaja painaa välilyöntiä *CageTriggerin* kohdalla, *Cage-oliot* kytkeytyvät pois päältä

Cage.cs:

- Kiinnitetään *Cage-olioon* 2. pelitasossa
- Kun *Pelaajaolio* törmää *Cage-olioon*, tämän pyörimissuunta vaihtuu päinvastaiseksi, kunnes *Pelaajaolio* törmää toiseen *Cage-olioon*

Metronome.cs:

- Kiinnitetään *Metronome-olioon* 1. pelitasossa
- Kun *Metronome-olio* on kosketuksissa *Pelaajaolion* kanssa, sen äänitehoste aktivoituu, ja muutaman sekunnin päästä pelitaso päättyy: tällöin siirrytään seuraavaan kohtaukseen

EndSpecialStage.cs:

- Korvaa *Metronome.cs* -skriptin 2. pelitasossa
- Toimintaperiaatteiltaan sama kuin *Metronome.cs*: kohtausta vaihtuu muutama sekunti sen jälkeen, kun *Pelaajaolio* on osunut *Metronome-olioon*
- Toisin kuin *Metronome.cs*, ei *EndSpecialStage.cs* käsittele millään tavalla *Metronome-olion* äänitehosteita

Timer.cs:

- Kiinnitetään *Timer-olioon* 1. ja 2. pelitasossa
- Vähentää sille annetusta sekuntimäärästä yhden sekunnin kerrallaan
- Aktivoi oikean äänitehosteen aina kun sekuntimäärä vastaa tiettyä minuuttia
- Esimerkki: 120 sekunnin kohdalla aktivoituu äänitehoste *2 minutes left*
- Äänitehosteita on luotu 1–6 minuutin kohdalle
- Kun sekuntimäärä on 0, aktivoituu äänitehoste *Time's Up*
- *Pelaajaolio* lakkaa liikkumasta ja kaikki äänilähteet mykistetään
- Erillinen *Game Over -näyttö* ilmestyy nykyisen pelikuvan tilalle, jolloin aktivoituu *Game Over -äänitehoste*
- Tämän jälkeen peli nollautuu ja alkaa alusta

MainMenuController.cs, PauseMenuController.cs, ButtonSound.cs ja VolumeController.cs

- Osa samanlaista *UI-näyttökokonaisuutta*
- *MainMenuController* esiintyy *Main Menussa* eli *päävalikkokohtauksessa*, *PauseMenuController* puolestaan 1. ja 2. pelikohtauksessa
- *Pause Menu* eli *keskeytysvalikko* aktivoidaan *Esc-painikkeella*: tällöin pelikohtauksessa olevat äänilähteet mykistetään ja fysiikkaominaisuudet pysäytetään. Vain valikossa kuuluvat äänet säilyvät.
- Kun pelaaja navigoi *W- ja S- tai ylä- ja alanuolinäppäimillä*, aktivoituu jokaisessa valikossa olevassa UI-painikkeessa oma *ButtonSound-skriptinsä*, joka lukee painikkeen nimen
- Sekä *MainMenu-* että *PauseMenu-valikoissa* on eri kerroksia, ja jokaisella on oma indeksinsä: tällä varmistetaan, ettei pelaaja voi navigoida valikkojen ulkopuolelle
- Jokainen painike aktivoi eri funktion joko *MainMenuController-* tai *PauseMenuController-skriptistä*.

- VolumeController on myös osa molempia valikoita: kun pelaaja liikuttaa kyseistä liukukomponenttia *A- ja D-* tai *vasemmalla ja oikealla nuolinäppäimellä*, muuttuu samalla pelin *AudioListener-kuunteluolion* äänenvoimakkuus.

TitleCardFader.cs

- Häivyttaa mustan ruudun näkymättömäksi, paljastaen halutun kuvasisällön
- Käytetään pelin alussa, jolloin esiintyy varoitusteksti, tekijätiedot ja itse pelin logo
- Samalla aktivoituu tietty ääni, joka kertoo ruudulle ilmestyvän kuvan sisällön
- Äänen soittamisen jälkeen musta ruutu häivytetään takaisin näkyväksi

AudioMovie.cs ja SkipAudio.cs:

- Toistaa halutun äänitehosteen ja vaihtaa sitten automaattisesti *kohtausta*
- Käytetään pelissä seuraavissa kohtauksissa: *aloitus- ja lopetuselokuva, opastusnauhoitteet* 1. ja 2. pelitasossa, *kiitospuhe* ja *lopputekstit*.
- Viiden sekunnin kuluttua yllä mainittujen kohtausten alkamisesta pelaaja voi siirtyä suoraan seuraavaan kohtaukseen painamalla *Enter-painiketta*: tämä kaikki toteutuu SkipAudio.cs:n ansiosta

LockMouse.cs:

- Käytetään kaikissa kohtauksissa lukitsemaan ja piilottamaan *hiirikursori* pelistä: vain *näppäimistö* on tarpeellinen peliä pelattaessa.

Liite 5. ”The Metronome of Dissonance” -pelin etusivu Itch.io-palvelussa

The Metronome of Dissonance (Demo)

An Audio-Only Action Adventure Beyond Reality

Play as the nameless character, who has lost themselves in the abstract world of sound perceptions and lost memories.

Find the mysterious Metronome of Dissonance and become closer to finding yourself.

- Survive dangerous labyrinths and strange orbitals in your search for the Metronome of Dissonance
- Defeat the mysterious robotic enemies in a Sound Fight
- Avoid Shadow Traps, Void Zones and other dangers
- Discover your strenghts and weaknesses in a metaphysical dimension


This audio-only video game is designed for the visually impaired players. Headphones are strongly recommended for the best experience.

[More information](#) ▾

Download

Download

Metronome-of-Dissonance (demo).zip 57 MB



The Metronome of Dissonance

Kuva 18 The Metronome of Dissonance -pelin julkaisusivu Itch.io-palvelussa.

”The Metronome of Dissonance (demo): todellisuuden rajat ylittävä, äänipainotteinen toimintaseikkailu.

Pelaa nimettömällä hahmolla, joka menetti itsensä äänihavaintojen ja kadonneiden muistojen täytämässä, abstraktissa maailmassa. Etsi salaperäinen Riitasoinnun metronomi ja ole lähempänä itsesi löytämistä.

- *Selviydy vaarallisista labyrinteista ja oudoista kiertoradoista etsiessäsi Riitasoinnun metronomia*
- *Päihitä salaperäiset robottiviholliset Äänitaisteluissa*
- *Vältä Varjoansoja, Tyhjiöalueita ja muita vaaroja*
- *Löydä vahvuutesi ja heikkoutesi metafyysisessä todellisuudessa*

Tämä äänipainotteinen videopeli on suunniteltu näkövammaisille pelaajille. Kuulokkeiden käyttöä suositellaan parhaan pelikokemuksen takaamiseksi.”

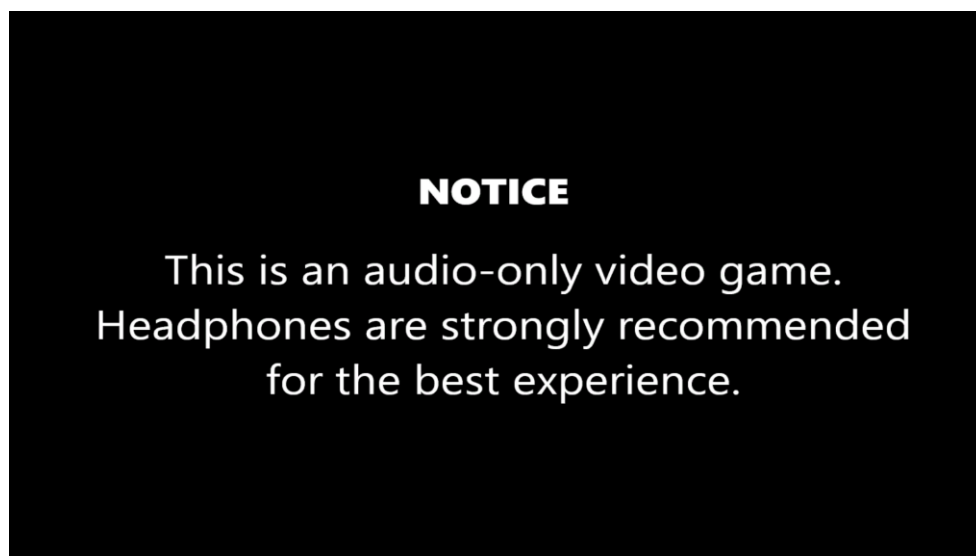
Linkki pelin julkaisusivulle: <https://henrysfriman.itch.io/the-metronome-of-dissonance-demo>

Liite 6. Valmiin pelin rakenne kuvankaappauksineen

Tässä liitteessä kuvataan esimerkki yhdestä tavasta pelata peli alusta loppuun.

Kohtaus 1: Notice

Kun peli käynnistyy, ruudulle ilmestyy tekstiä, jonka myös ruudunlukija toistaa: *Notice: this is an audio-only video game. Headphones are strongly recommended for the best experience* ("Varoitus: tämä on äänipainotteinen videopeli. Kuulokkeiden käyttöä suositellaan vahvasti parhaan pelikokemuksen saamiseksi").




Kuva 19 Pelin alussa esiintyvistä ilmoitusteksti.

Kohtaus 2: StartScreen

Ruudulle ilmestyy uusi teksti, jonka myös ruudunlukija toistaa: *Henry Sebastian Friman presents* ("Henry Sebastian Friman esittää").

Teksti häivyy pois, ja kuvaan ilmestyy pelin kansikuva ja nimi. Ruudunlukija toistaa: *The Metronome of Dissonance: Demo version* ("The Metronome of Dissonance: kokeiluversio"). Kohtaus häivyy takaisin mustaksi.



Henry Sebastian Friman
presents

Kuva 20 Pelin toinen kohtaus alkaa esittelyllä "Henry Sebastian Friman presents".

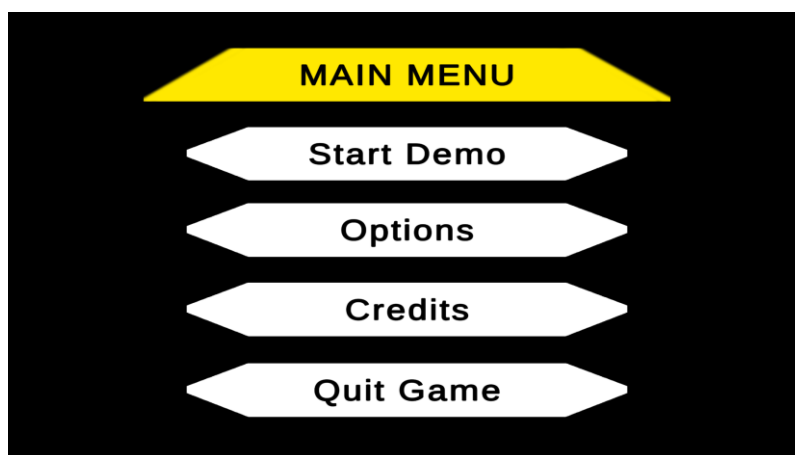


Kuva 21 Esittelyn jälkeen näytetään pelin kansikuva ja logo. Tähän päättyy pelin toinen kohtaus.

Kohtaus 3: MainMenu

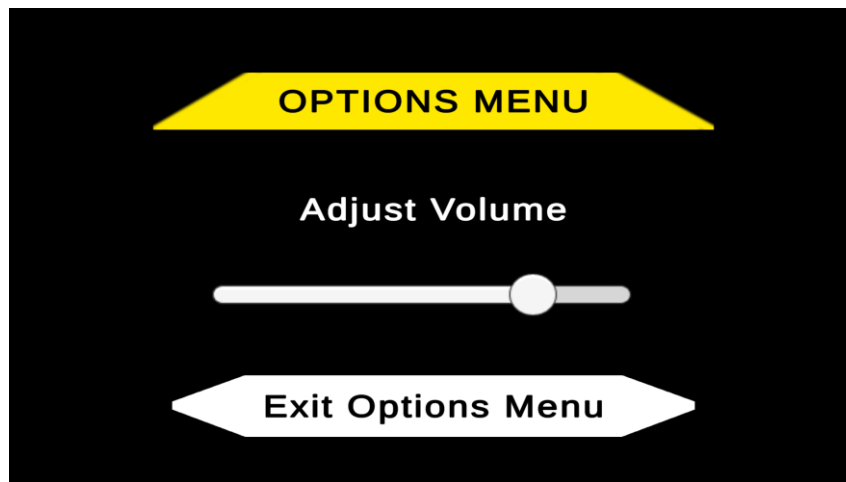
Ruudulle ilmestyy pelin valikko. Ruudunlukija sanoo: *Main Menu. Navigate with Up and Down arrow keys or W and S keys. Select with Enter key* ("Päävalikko. Navigoi ylä- ja alanuolinäppäimillä tai W- ja S-näppäimillä. Valitse painamalla Enter-näppäintä"). Valikossa esiintyvät painikkeet ovat *Main Menu* ("Päävalikko"), *Start Demo* ("Aloita demo"), *Options* ("Asetukset"), *Credits* ("Tekijätiedot") ja *Quit Game* ("Lopeta peli"). Aina kun pelaaja navigoi valikossa, ruudunlukija toistaa

painikkeen nimen ääneen, pois lukien *Main Menu*, jossa ruudunlukija toistaa ääneen sekä painikkeen nimen että valikon käyttöohjeet. Pelaajan kohdistama painike esiintyy keltaisena.



Kuva 22 Pelin päävalikko (Main Menu). Keltainen väri esittää kohdistettua painiketta.

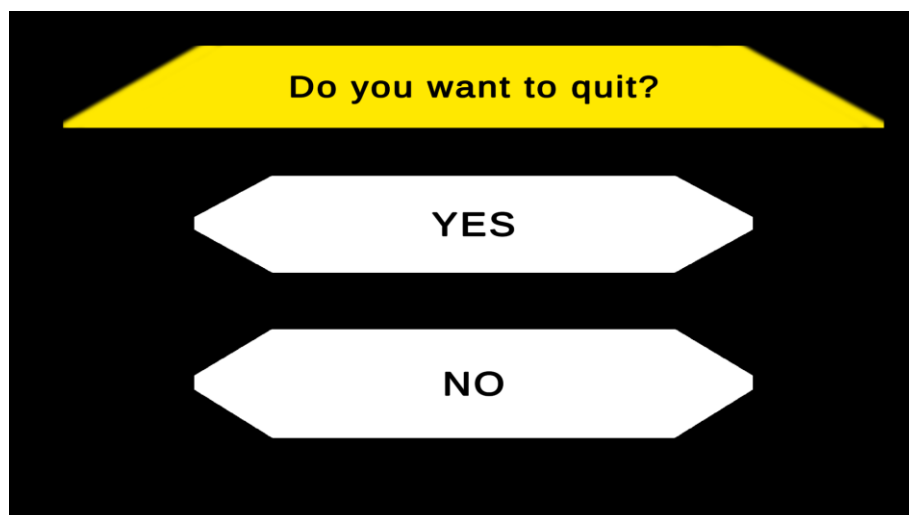
Painamalla *Options*-painiketta, avautuu uusi valikko asetuksille. Se koostuu kahdesta painikkeesta, joita ovat *Options Menu* ("Asetusvalikko") ja *Exit Options Menu* ("Poistu asetusvalikosta"). Keskellä on liukupainike *Adjust Volume* ("Säädä äänenvoimakkuutta"), jota pelaaja voi liikuttaa vasemmalle ja oikealle joko *nuolinäppäimillä* tai *A- ja D-näppäimillä*. Sekä *Options Menu*- että *Adjust Volume* -painikkeissa kuuluu myös niiden käyttöohjeet. *Options Menu* -painikkeessa kyse on samasta käyttöohjeesta kuin *Main Menu* -painikkeessa, mutta *Adjust Volume* -liukupainikkeessa ohjeet ovat hie-
man erilaisemmat: *Adjust Volume: Change the volume with Left and Right arrow keys or A and D keys* ("Säädä äänenvoimakkuutta: vaihda äänenvoimakkuutta vasemmalla ja oikealla nuolinäp-
päimellä tai A- ja D-näppäimillä").



Kuva 23 Pelin asetusvalikosta (Options Menu). Keltainen väri esittää kohdistettua painiketta.

Päävalikon alussa esiintyvä *Credits-painike* ei avaa erillistä valikkoa, vaan sitä painaessa ruudunlukija alkaa toistaa pelin tekijätietoja sisältävän äänitteen: tekijätiedot sisältävät enimmäkseen minun nimeni, mutta siellä mainitaan myös kaikki käyttämäni työkalut sekä äänitehosteet.

Quit Game -painike avaa oman valikkonsa, jossa on kolme painiketta: *Do you want to quit?* ("Haluatko lopettaa?"), *Yes* ("Kyllä") ja *No* ("Ei"). *Do you want to quit?* -painike sisältää valikon käyttöohjeen samalla tavalla kuin *Main Menu* -painike.



Kuva 24 Pelin lopettamisen vahvistusvalikko. Keltainen väri esittää kohdistettua painiketta.

Kohtaus 4: Opening

Kun pelaaja on painanut *MainMenu*-kohtauksessa *Start Demo* -painiketta, alkaa uusi kohtaus. Kyseessä on äänimuotoinen *alkuelokuva*, joka alkaa ruudunlukijan puheella: *Audio movie starts now. Listen and imagine* ("Äänielokuva alkaa nyt. Kuuntele ja kuvittele").

Äänielokuva sijoittuu meluisalle kadulle, josta kuuluu autojen kaahailua, tööttäyksiä ja ambulanssin ääniä. Vähitellen äänien seassa kantautuu voimistuva metronomin tikitys, jonka aikana kaupungin meteli katoaa. Tikitys vaimenee, minkä jälkeen on ympäristö vaihtunut rauhallisiin, metsäisiin ääniin. Vähitellen tikitys palaa taas voimakkaana, jonka jälkeen kuuluu voimakas lasersädetä muistuttava ääni. Hetken hiljaisuuden jälkeen ympäristön äänet ovat vaihtuneet kummallisiin, kaikumaisiin ääniin, jotka ovat kuin toisesta todellisuudesta. Samalla metronomi alkaa tikittää kolmannen kerran. Kohtaus päättyy.



Kuva 25 Pelin ensimmäinen äänielokuvakohtaus. Pelinäköymä esitetään harmaana ruutuna, eikä se esitä visuaalisia tapahtumia.

Kohtaus 5: Tutorial-1

Alkuelokuvan jälkeen alkaa uusi kohtaus, jossa esitellään ensimmäisen tason peliohjeet. Kyseessä on noin viiden minuutin pituinen *äänitutoriaali*, jonka voi halutessaan myös ohittaa painamalla *Enter*-painiketta.



Kuva 26 Pelin äänitutoriaali-kohtaus. Pelinäkömää esitetään vihertävän ruskeana ruutuna, eikä se esitä visuaalisia tapahtumia.

Tutoriaali kerrotaan tarkasti, miten peliä pelataan. Pelaaja on sokkelossa, tehtävänä on löytää *Riita-soinnun metronomi* ("The Metronome of Dissonance"). Ohjeet ovat tiivistetysti seuraavanlaiset: kun pelaaja kuulee äänitehosteen x, hän on tilanteessa y, jolloin hän voi tehdä asian z. Risteyksessä ollessaan hän voi liikuttaa pelihahmoa eteenpäin, vasemmalle tai oikealle käyttämällä *W*-, *A*- ja *D*-näppäimiä. Pelaaja voi myös painaa *Control*-näppäintä, jolloin hän kuulee metronomin tikityksen jossain kaukaisuudessa: tätä ääntä kohti pelaajan on kuljettava. Kun pelaaja liikuttaa hahmoa, tämä liikkuu automaattisesti valittuun suuntaan, kunnes törmää uuteen risteykseen tai vihollisiin: pelaaja myös kääntyy automaattisesti kulmakohdissa. Joutuessaan vihollisten ympäröimäksi pelaajan on kuunneltava edessään, vasemmalla ja oikealla olevien vihollisten äänitehosteita ("Shita", "Naka" tai "Ue"): pelaajan on valittava samaa ääntä esittävä *ääniase* *Q*- tai *E*-näppäimillä, ja sen jälkeen ampua oikeaan suuntaan *nuolinäppäimillä*. Pelaajalla on haasteenaan myös muita vaaroja, kuten *ShadowTrap*- ja *VoidZone*-objekteja: *ShadowTrap*-ansan lähestyessä pelaajan on painettava *välilyöntipainiketta* selvitäkseen ansasta, kun taas *VoidZone* on yksinkertaisesti yksi sokkelon umpikujista. Aina kun pelaaja saa kolme vihollisosumaa, törmää *ShadowTrap*-ansaan tai kohtaa *VoidZone*-umpikujan, pelaaja häviää ja palaa takaisin viimeisimpään risteykseen, jossa tämä oli ollut ennen häviötä. Lisäksi ohjeissa painotetaan 7 minuutin aikarajasta, jonka sisällä pelitaso on läpäistävä.

Kohtaus 6: Stage 1

Ensimmäisen tutoriaalin jälkeen alkaa pelin ensimmäinen taso eli sokkelo. Pelikuva on peitetty sekä ensimmäisessä että toisessa pelitasossa magentanvärillä kalvolla, jotta pelaajan olisi

luotettava kuuloaisteihinsa. Pelaajan sijainti (risteys, äänitaistelu, ansat) ja kävelysuunnat ilmoitetaan omilla äänitehosteillaan. Lisäksi pelissä ilmoitetaan aina erillisellä äänitehosteella, montako minuuttia peliaikaa on jäljellä. Ensimmäisessä tasossa esiintyviä konkreettisia tapahtumia kuvataan luvussa 4.2.



Kuva 27 Pelin ensimmäisen ja toisen tason näkymät on peitetty magentanvärillä ruudulla.

Tarvittaessa pelaaja voi avata keskeytysvalikon Esc-näppäimellä. Keskeytysvalikko on sisällöltään pääosin sama kuin MainMenu-kohtauksessa oleva päävalikko, mutta pienin poikkeuksin: siinä missä päävalikossa olivat painikkeet *Credits* ja *Quit Game*, on keskeytysvalikossa *Tutorial* ja *Return to Main Menu*. Toimintaperiaatteet ovat kuitenkin samat, sillä *Tutorial-painike* käynnistää vain tutoriaalissa kuullun ääniklipin (hieman editoituna), kun taas *Return to Main Menu* avaa uuden valikon, jossa pyydetään pelaajan varmistusta pelin lopettamisesta ja palaamisesta takaisin päävalikkoon.

Kohtaus 7: Tutorial-2

Kun pelaaja on löytänyt ensimmäisessä tasossa olevan metronomin, alkaa uusi *tutoriaalikohtaus*, jossa esitellään seuraavan tason peliohjeet. Myös nämä ohjeet voi ohittaa *Enter-painikkeella*. Tutoriaalain aikana ruudulla on sama vihertävän ruskea ruutu kuin ensimmäisessä tutoriaalikohtauksessa (Tutorial-1).

Tutoriaalissa pohjustetaan seuraavan tason tilanne: pelaaja on ajautunut *kehään*, jossa hän pyörii kellonmyötäisesti tikittävän metronomin ympärillä. Pelaajan on kuunneltava *teleporttisillasta* kantautuva äänitehoste ja painettava joko *W*-, *A*- tai *D-näppäintä* sen perusteella, mistä suunnasta

pelaaja kuulee äänen. Painettuaan oikeaa näppäintä oikealla hetkellä, pelaaja lopettaa pyörimisen, siirtyä sillan avulla lähemmäksi metronomin ydintä ja jatkaa sitten taas pyörimistään kellonmyötäiseen suuntaan. Tässä tasossa ei ole taistelujärjestelmää tai elämäpisteitä, mutta vaaroina ovat *hökkiansat*, jotka sulkevat pelaajan kahden *hökkiseinän* väliin: pelaaja kimpoaa aina päinvastaiseen suuntaan osuessaan yhteen seinästä, minkä seurauksena pelaaja kimpoaa edestakaisin kellonmyötäiseen ja -vastaiseen suuntaan. Pelaajan on kuunneltava seinämien kimpoamisesta aiheutuvien äänitehosteiden rytmiä, tunnistettava se hetki, jolloin hän on ansan keskipisteessä, ja painettava välilyöntinäppäintä. Tällöin seinämät katoavat ja pelaaja jatkaa liikettään kiertoradalla. Pelaajan liikesuunta on joko kellonmyötäiseen tai -vastaiseen suuntaan riippuen siitä, kumpaan suuntaan hän oli menossa ennen kuin vapautui ansasta. Tasossa on myös kolmen minuutin aikaraja.

Kohtaus 8: Stage 2

Toisen tason tutoriaalin jälkeen alkaa pelin toinen taso eli *kehä*. Pelaaja aloittaa pyörimisen kehän uloimmalta reunalta. Myös tässä tasossa varsinainen pelikuva on peitetty magentanvärisellä kalvolla, ja jokaisen jäljellä olevan peliminuutin kohdalla kuuluu oma äänitteensä. Pelin toisessa tasossa esiintyviä konkreettisia tapahtumia kuvataan luvussa 4.2, samalla tavalla kuin ensimmäisen tason tapahtumat.

Keskeytysvalikon ulkonäkö ja sisältö ovat toisessa pelitasossa samat kuin ensimmäisessä tasossa, mutta tällä kertaa *Tutorial-painike* aktivoi *Tutorial-2-kohtauksessa* esiintyneen *tutoriaaliääniraidan*. Pääajatuksena on, että jokaisen pelitason keskeytysvalikossa oleva Tutorial-painike käynnistäisi aina sen ääniklipin, jota käytettiin tasoa edeltäneessä tutoriaali-kohtauksessa, vaikka keskeytysvalikossa käytetty versio olisikin editoitu hieman lyhyemmäksi.

Kohtaus 9: Closing

Kun pelaaja on päässyt *kehän* ytimeen ja löytänyt *metronomin*, alkaa uusi kohtaus: lyhyt *äänielokuva*, jossa kuuluu rauhallisia ääniä metsästä. Metronomin tikitystä ei kuulu enää.



Kuva 28 Lopetuselokuvassa käytettävä vaaleanvihreä ruutu.

Kohtaus 10: ThankYou

Loppuäänielokuvan jälkeen avautuu uusi kohtaus, jossa toistetaan minun kirjoittamani viesti pelaajalle. Ääniklipissä kiitän pelaajaa pelaamisesta ja selitän perimmäisen syyn peliprojektilleni: painotan sitä, miten tärkeää on arvostaa itseään ja hyväksyä oma epätäydellisyytensä. Esitän myös viestissäni lupauksen siitä, että aion jatkaa projektia pidemmälle tulevaisuudessa.



Kuva 29 Kiitosviestin aikana esiintyvä kullanuskea ruutu.

Kohtaus 11: Credits

Viimeisessä kohtauksessa toistetaan ääniklippi, jossa luetellaan kaikki tekijätiedot: olen yksin vastuussa pelin toteutuksesta, mutta mainitsen myös kaikki käyttämäni työkalut ja äänitehosteet. Tekijätietojen jälkeen peli palaa takaisin ensimmäiseen kohtaukseen (Notice).



Kuva 30 Pelin tekijätietoja kertovassa kohtauksessa käytettävä vaaleankeltainen ruutu.