

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

Aki Karppanen

FLOW'N HYÖDYNTÄMINEN VIDEOPELEISSÄ

Opinnäytetyö
Toukokuu 2016



OPINNÄYTETYÖ
Toukokuu 2016
Tietojenkäsittelyn koulutusohjelma

Karjalankatu 3
80200 JOENSUU
(013) 260 600

Tekijä(t)
Aki Karppanen

Nimeke
Flow'n hyödyntäminen videopeleissä

Toimeksiantaja

Tiivistelmä

Opinnäytetyössä tutkittiin aiemmista peleistä löytyneitä positiivisesti flow-kokemusten saavuttamiseen johtaneita pelielementtejä ja –mekaniikkoja ja pyrittiin toistamaan niitä erillisessä käytännön toteutuksessa. Flow-kokemuksella tarkoitetaan peliä pelatessa tapahtuvaa tyytyväisyyden, keskittyneisyyden ja onnellisuuden tilaa, jossa pelin pelaaminen on itsessään nautittavaa ja toimii omana palkintonaan.

Tutkimustoteutuksessa tutkittiin kahta peliä (One Finger Death Punch, Guitar Hero-sarja), joissa on hyvä flow eri arvioiden perusteella ja joita on käytetty flow'n tutkimiseen aiemmin. Käytännön osuudessa luotiin pelidemo käyttäen tutkimustuloksia suunnittelun pohjana.

Opinnäytetyössä löydettiin joitakin suunnitteluratkaisuja ja mekaniikkoja jotka vaikuttavat pelien flow-kokemuksiin ja jotka ovat toistettavissa eri peleissä. Näitä ovat nopeatempoisuus, reaktiovetoinen pelirakenne ja värien käyttö symboloimaan eri oliotyyppejä tai pelimekaniikkoja. Näiden tietojen perusteella pystyttiin luomaan pelidemo, joka saavutti yli puolessa testaaajista jonkin asteisen flow-kokemuksen.

Flow'n käyttäminen pelisuunnittelun pohjana on potentiaalinen tapa suunnitella pelejä niin, että joko peli tai sen jokin osio, saavuttaa todennäköisesti flow-tilan pelaajassa. Flow-tilan saavuttamista voidaan ennustettavasti helpottaa pelimekaanisilla ratkaisuilla ja suunnittelulla.

Kieli
suomi

Sivuja 36
Liitteet 1
Liitesivumäärä 1

Asiasanat
peliohjelmointi, psykologia, flow-tila



THESIS
May 2016
Degree Programme in Business
information Technology
Karjalankatu 3
FI 80200 JOENSUU
FINLAND
+358 013 260 600

Author(s)
Aki Karppanen

Title
Using flow in video games

Commissioned by

Abstract

This thesis researched different game play elements and mechanics that positively affect the flow-state of a game and attempted to replicate those elements and mechanics in a separate practical demonstration. The flow-state means a state, where playing a game generates a feeling of happiness and concentration. In this state, playing a game is its own reward.

The research portion of the thesis looked at two games (One Finger Death Punch, the Guitar Hero-series), that have a good flow based on different reviews and that have been used to study flow before. In the practical portion a gameplay demo was created using the results of the study as a base for game design.

The study found some design practices and mechanics that affect a games flow-experiences and that are replicable in different games. These are high tempo, reaction based gameplay loop and the usage of colors to symbolize different object types or gameplay mechanics. Based on this knowledge it was possible to create a game demo that achieved some level of flow-experience in over half the testers of the demo.

Using flow as a basis for game design is a potential way to design games so, that the game or some part of it, will probably achieve a flow-state in the player. Achieving a flow-state can be predictably made easier by using certain gameplay mechanic solutions and designs.

Language
Finnish

Pages 36
Appendices 1
Pages of Appendices 1

Keywords

programming, psychology, flow-state

Sisältö

1	Johdanto.....	5
2	Määritelmiä.....	5
2.1	Flow	5
2.2	APM.....	7
2.3	FPS.....	7
2.4	Core gameplay loop.....	8
2.5	Unity.....	9
2.6	Trigger, Collider, Collision.....	9
2.7	Unity PlayerPrefs	10
2.8	Unity Scene	11
2.9	Unity AudioSource, AudioListener, AudioClip	11
3	Esimerkkitapaus: One Finger Death Punch.....	12
3.1	Pelin kuvaus	12
3.2	Flow'n vaatimukset	12
3.3	Tekninen toteutus	13
3.4	Arvioinnit	14
3.5	Yhteenveto.....	14
4	Esimerkkitapaus 2: Guitar Hero -sarja.....	15
4.1	Pelin kuvaus	15
4.2	Flow'n vaatimukset	16
4.3	Tekninen toteutus	17
4.4	Arvioinnit	17
4.5	Yhteenveto.....	18
5	Johtopäätökset esimerkkitapauksista	19
5.1	Reaktioajan vaikutus flow'n saavuttamiseen.....	19
5.2	Keskeytysten poistaminen pelistä.....	20
5.3	Värien käyttö symboleina.....	20
5.4	Musiikin ja rytmin vaikutus pelin flow'hun.....	21
6	Käytännön toteutus	22
6.1	Kehitysympäristö	22
6.2	Suunnittelu.....	22
6.3	Flow'n vaatimukset suunnitelmassa	24
6.4	Kehitystyö	24
6.4	Lopputulos	28
6.5	Käytännön osuuden testaaminen	29
6.6	Flow'n vaatimukset valmiissa demossa	30
7	Yhteenveto tutkimuksesta ja käytännön toteutuksesta.....	31
7.1	Kyselyvastaukset	31
7.2	Flow-kokemusten arviointi	32
7.3	Flow-kokemusten estymisen tai keskeytymisen syyt	33
7.4	Kehitystoiveet, parannusehdotukset	33
7.5	Yhteenveto.....	34
	Lähteet.....	35

Liitteet

Liite 1 Zombiacity Evac-demon flow-kysely ja sen vastaukset

1 Johdanto

Opinnäytetyön tarkoituksena on tutkia eri tapoja, joilla flow'ta on hyödynnetty onnistuneesti videopeleissä luomaan mielenkiintoisia pelikokemuksia ja niiden perusteella selvittää, miten flow'n hyödyntäminen tekee pelistä onnistuneen. Peleistä puhuttaessa flow määritellään yleensä tilaksi, jossa pelaaja keskittyy täysin pelin tapahtumiin ja aktiivisen päätöksenteon sijasta antaa alitajunnan ja intuition ohjata toimintaa. Pelaaja tietää mitä tekee ja tietää miten saavuttaa päämääränsä, mutta ei koe peliä haasteettomaksi. Flow-tilan saavuttaminen tekee pelaamisesta nautinnollista ja palkitsevaa itse itsensä vuoksi ja voi toimia tehokkaana pelisuunnittelun menetelmänä parantamaan pelikokemusta.

Tutkittujen pelien arvioinnissa käytetään kriteerinä luotettujen peliarvostelijoiden arviointeja, omia pelikokemuksia sekä yleisön arvioiden keskiarvoja yhdistettynä määritelmiin flow'n vaatimuksista videopeleissä.

Käytännön osuuden ja tutkimuksen yhteisenä tavoitteena on luoda lista asioista, jotka vaikuttavat positiivisesti tai negatiivisesti pelin flow-kokemuksiin, sekä suunnitella ja toteuttaa kootun listan perusteella käytännön demo. Käytännön demoa testautetaan puolueettomilla testajilla toteutuksen jälkeen, verraten sitä aiemmin luotuun listaan ja flow'n vaatimuksiin testauksen jälkeen.

2 Määritelmiä

2.1 Flow

“The state in which people are so involved in an activity that nothing else seems to matter; the experience itself is so enjoyable that people will do it even at great

cost, for the sheer sake of doing it" - Mihaly Csikszentmihalyi (Csikszentmihalyi 1990, 4).

Flow on termi, jolla kuvataan ihmisen täydellistä keskittymistä johonkin tehtävään ja siitä johtuvaan onnellisuuden tunteeseen. Kaikki tuntuu sujuvan, asiat etenevät kuin omalla painollaan, mutta silti tuntumatta liian helpolta. Flow-tila tai flow-kokemus saavutetaan yleensä kun tehdyn tehtävän vaikeus on sopivassa suhteessa omaan taitoon ja mikään ei häiritse tehtävään keskittymistä. Flow'ta on tutkittu urheilijoiden ja taitelijoiden avulla '70-luvulta lähtien ja viimeisen vuosikymmenen aikana yhä enemmän osana pelitutkimusta.

Videopelien osalta Curtiss Murphy on koonnut vaatimukset flow'lle:

- 1) Selvät tavoitteet, pelaajan tulee ymmärtää mitä hänen pitäisi milloinkin tehdä
- 2) Palaute, peli antaa välitöntä palautetta pelaajan etenemisestä ja siitä ovatko pelaajan tekemät päätökset hyviä vai ei. Palautteen ei tulisi keskeyttää pelaamista, vaan olla osa sitä.
- 3) Pelissä tulee olla järkevässä aikataulussa saavutettavissa oleva päämäärä. Pelin tehtävien on oltava haastavia, muttei niin haastavia, että eteneminen keskeytyy sen takia.
- 4) Keskittyminen, peli ei saa estää pelaajan etenemistä ja keskittymistä häiritsevillä keskeytyksillä (esim. liian pitkät välivideot) tai huonosti toimivilla valikoilla tai käyttöliittymällä. Tekniset ongelmat pelissä vaikeuttavat myös keskittymistä ja mielenkiinnon ylläpitoa ja rikkovat flow'ta.

Flow'n vaatimuksia peleihin voidaan yksinkertaistaa sanomalla, että flow-tila saavutetaan yleensä kun pelin vaikeus suhteessa pelaajan taitoihin on sopivalla tasolla. Jos peli on liian helppo, pelaaja tylsistyy ja lopettaa pelaamisen. Jos peli on liian vaikea, pelaaja turhautuu ja lopettaa. Pelin täytyy myös edetä pelaajan ehdoilla, eikä keskeyttää pelaamista häiriötekijöillä tai teknisillä ongelmilla (Murphy 2011).

Tässä opinnäytetyössä käytetään Curtiss Murphyn määritelmää flow'sta koska se yhdistää aiempaa flow'sta tehtyä tutkimusta yhdeksi kokonaisuudeksi nimenomaan videopeleistä keskusteltaessa. Murphyn määritelmässä on vaikutteita mm. Jenova Chenin ja Mihaly Csikszentmihalyin aiemmin kirjoittamista artikkeleista, mutta siinä keskitytään vain niihin asioihin jotka pätevät suoraan videopeleihin ja on jätetty ulkoiset tekijät pois.

2.2 APM

Actions Per Minute, tai toiminnot minuutissa, on termi jolla mitataan pelaajan tekemien toimintojen määrää ja sitä kautta pelin tempoa. Esim. shakki on erittäin hidastempoinen lautapeli jossa APM on usein alle 1, alle yksi siirto minuutissa. Pelit joissa on hyvä flow sijoittuvat yleensä APM-skaalan yläpäähän ja vaativat useita toimintoja sekunnissa, eli $APM > 60$. Odotusajat jolloin pelaaja ei tee mitään rikkovat flow-kokemuksen hyvin nopeasti.

Shakki ja Go ovat tässä huomattavia poikkeuksia, pelejä joiden APM on erittäin matala, mutta jotka silti aiheuttavat usein pelaajissaan flow-tilan. Tämä johtuu siitä, että pelaajat keskittyvät täysin pelaamaan peliä päänsä sisällä useita siirtoja eteenpäin myös silloin kun he eivät aktiivisesti tee siirtojaan.

Termiä APM käytetään usein strategiapeleissä mittaamaan pelaajan taitotasoa: korkea APM nähdään hyvän pelaajan vaatimuksena, vaikka se ei ole tae hyvästä pelitaidosta. Huonolla pelaajalla voi olla korkea APM, koska hän tuhlaa resurssejaan turhiin toimintoihin ja toimintojen toistamiseen, kun taas hyvä pelaaja saavuttaa paljon enemmän samalla APM:llä. Tämän takia puhutaan usein Effective Actions Per Minute-luvusta, jossa lasketaan vain toiminnot jotka tekevät jotain huomattavaa (jca 2003).

2.3 FPS

Frames Per Second, tai piirretyt ruudut sekunnissa, on termi jolla lasketaan pelien ruudunpäivitysnopeutta. Pelit, joiden FPS on matala, tuntuvat hitailta ja vastaavat pelaajan toimintoihin suuremmalla viiveellä. Peli vaikuttaa vähemmän sulavalta ja tämä rikkoo pelin flow'ta ja pelaajan keskittymistä. Vastaavasti korkea FPS tekee pelistä sulavan tuntuisen ja auttaa sitä ylläpitämään hyvää flow'ta nopeuttamalla pelin vasteaikaa pelaajan toimintoihin. Epätasainen FPS rikkoo myös peliin keskittymistä toimimalla ärsyttävänä häiriötekijänä. Normaalin FPS:n rajaksi hyväksytään yleisesti vähintään 30 FPS, hyvänä pidetty FPS on yleensä 60 tai enemmän, eli näytön tai television normaali virkistystaajuus (Korosh Ghazi 2012).

2.4 Core gameplay loop

Core gameplay loop kuvaa pelin yleisimpiä tapahtumia ja niiden yhdistymistä toisiinsa. Yksinkertaisimmillaan Core Gameplay loop voi olla esim. Tammea pelatessa:

Siirrä pelimerkkiä -> Vastustaja siirtää pelimerkkiä -> Toista.

Videopeleissä usein nähty esimerkki on *Juokse -> Hyppää -> Ammu -> Toista.*

Core gameplay loop ei ota kantaa pelin etenemiseen, tavoitteisiin, kenttien voittamiseen tai häviämiseen tms., se kuvaa ainoastaan pelissä yleisesti toistuvia tapahtumia.

Flow'n kannalta core gameplay loopissa on merkittävää se, että peli joko pakottaa pelaajan tekemään jotain core gameplay looppiin kuuluvaa aktiivisesti tai kannustaa siihen palkitsemalla kokoaikaista pelaamista. Esimerkiksi peli *Move or Die* (2016) vaatii pelaajaa liikkumaan koko ajan, tai pelaaja häviää erän.

Super Meat Boy (2010) taas antaa pelaajalle parempia palkintoja kentistä, jos ne suoritetaan riittävän nopeasti, joka on usein mahdollista vain, jos pelaaja ei pysähdy missään vaiheessa. Kentän läpipeluusta muodostuu yhtenäinen ketju core gameplay loopin tapahtumia, joka ylläpitää pelaajan flow-kokemusta.

Core gameplay loopin ulkopuolella on secondary gameplay loop, jossa otetaan huomioon pelin tavoitteet. Secondary gameplay loop voi olla esimerkiksi:

Tapa vihollisia -> Suorita tehtäviä -> Voita kenttä -> Toista.

Suorittamalla eri core gameplay loopin tapahtumia, saavutetaan secondary gameplay loopin tavoitteita.

Siinä missä core gameplay loop kuvaa tapahtumia jotka tapahtumat sekuntimittakaavassa, secondary gameplay loop kuvaa tapahtumia jotka tapahtuvat yhden pelisession, kartan tai ottelun mittakaavassa. Yleensä gameplay loopista puhuttaessa tarkoitetaan nimenomaan core gameplay looppia (Ask a Game Dev 2014).

2.5 Unity

Unity on Unity Technologiesin luoma sovelluskehitysympäristö, joka on tarkoitettu erityisesti pelien kehitykseen. Se yhdistää valmiin 3d-pelimoottorin koodieditoriin ja mahdollistaa ohjelmien luomisen suurelle määrälle eri laitteita samalla aikaa. Unityn käyttö on ilmaista (myös kaupallisiin tarkoituksiin) yksityishenkilöille ja yrityksille joiden vuositulot tai -rahoitus ovat alle sata tuhatta dollaria. Jos Unitylla luotua ohjelmaa myydään yli sadan tuhannen dollarin arvosta, voi kaupallisen lisenssin hankkia myös jälkikäteen (Unity FAQ 2016).

Unity tukee suoraan Microsoftin Visual Studiota ja Mono Developia IDE:nä (Integrated Development Environment), joiden lisäksi voidaan käyttää muita editoreita. Unity on saatavilla Windowsille ja Macille ja siitä on lisäksi kokeellinen Linux-versio.

Unity on nopeasti vallannut alaa muilta pelimoottoreilta sen suhteellisen yksinkertaisuuden, ilmaisuuden ja matalan aloituskynnyksen ansiosta. Sen markkinaosuus on kirjoitushetkellä Unityn oman arvion mukaan n. 45% (Unity Company Facts 2016).

2.6 Trigger, Collider, Collision

Unityn käyttämässä fysiikkamoottorissa kontaktit eri olioiden tai niiden osien välillä määritetään Collider- ja Trigger-komponenttien avulla. Käyttämällä näitä komponentteja voidaan määritellä milloin oliot osuvat toisiinsa ja mitä silloin tapahtuu. Tämän lisäksi voidaan määritellä eri tapahtumia törmäyksen alkamiselle, sen aikana tapahtuville asioille ja törmäyksen jälkeisille tapahtumille.

Yksinkertaisimmillaan oliot eivät vaikuta toisiinsa fyysisesti, jolloin toinen törmäyksen osapuoli on Trigger ja toinen Collider tai Trigger, tässä tapauksessa kyseessä on Trigger Event. Erilaisia Trigger Eventtejä voivat olla esim. kartan vaihtaminen, kun pelaaja saapuu maalialueelle, tai uusien vihollisten luominen, kun pelaaja saapuu tiettyyn osaan kartasta. Triggerit eivät usein näy pelialueella ollenkaan, ainoastaan kehitystyökaluissa.

Jos kumpikin törmäyksen osapuoli on Collider, kyseessä on Collision Event, jossa oliot vaikuttavat toisiinsa fyysisesti. Esim. biljardipelissä kun pallo osuu toiseen, kumpaankin palloon vaikuttaa fyysinen voima. Luomalla olioille Collider-komponentit voidaan varmistaa, etteivät oliot voi kulkea seinien läpi, tai tippua kartan lattian läpi (Unity Manual: Physics 2016).

2.7 Unity PlayerPrefs

Unityn käyttämä PlayerPrefs-luokka mahdollistaa ohjelman tietojen tallentamisen käyttäjäkohtaisesti niin, että tiedot säilyvät eri pelikertojen välillä. Normaalisti tällä voidaan toteuttaa pelitilanteen ja eri asetusten tallentaminen. Windowsissa tiedot tallennetaan rekisteriin, muissa järjestelmissä erilliseen tiedostoon jonka sijainti riippuu käyttöjärjestelmästä.

PlayerPrefs-tiedot tallennetaan automaattisesti, kun peli lopetetaan, joten se on erittäin käytännöllinen tapa pitää kirjaa pelitilanteesta. Lisäksi on mahdollista hakea ja tallentaa tietoja aina tarvittaessa myös pelin aikana skriptien avulla (Unity Scripting API: PlayerPrefs 2016).

2.8 Unity Scene

Unity jakaa jokaisen peliosion tai kartan omaksi Scenekseen. Scenellä on omat oliionsa, kamerat, valaistus ja skriptit. Normaalisti eri oliot eivät siirry Scenestä toiseen, ellei sitä erikseen vaadita. Erilaisia Scenejä joita peleissä on ovat mm. päävalikko, eri kentät, asetukset, ohjeet ja tekijälistaus. Peli on myös mahdollista toteuttaa kokonaan yhdessä scenessä pelkästään luomalla ja tuhoamalla olioita ohjelmallisesti, mutta se johtaa helposti hyvin monimutkaiseen Scenen rakenteeseen (Unity Manual: Scenes 2016).

Joissakin tilanteissa on tarpeellista varmistaa, että jokaisessa Scenessä on tasan yksi kappale jotain komponenttia, kuten AudioListeneria tai pelitilannetta seuraavaa skriptiä, riippumatta siitä miten Sceneen on päädytty. Tämä voidaan toteuttaa käyttämällä Singleton Design Patternia, jossa ensimmäinen luotu komponenttityypin sisältävä olio määritetään ainoaksi sen tyyppin olioksi ja kaikkien sen jälkeen luodut saman tyyppin oliot tuhotaan välittömästi. Näin saavutetaan tilanne, jossa voidaan aina kutsua tämän olion metodeja ja olla varmoja siitä, että olio on olemassa ja käyttövalmis. Tämän jälkeen vaadittu olio voidaan lisätä jokaiseen Sceneen ja olla silti varmoja siitä, että niitä on aina käytössä tasan yksi kappale (OODesign: Singleton Pattern).

2.9 Unity AudioSource, AudioListener, AudioClip

Unityn äänijärjestelmä pohjautuu kahteen komponenttiin, joiden on oltava paikalla jokaisessa Scenessä, joissa halutaan soittaa ääniä. Nämä komponentit ovat AudioSource, joka kuvaa äänen sijaintia 3d-tilassa ja hoitaa äänen soittamisen sekä AudioListener, joka kuuntelee ääntä ja toimii pelaajan ”korvina” 3d-tilassa. Normaalisti AudioListener on osa pelin kameraa, jolloin äänet kuuluvat aina suhteessa kameran sijaintiin.

AudioSource-komponentteja voi olla monta ja varsinkin 3d-peleissä se voi olla kannattavaa, koska jokaisella AudioSourcella on oma sijaintinsa pelialueella. Näin esim. pelaajan takana oleva vihollinen saadaan kuulostamaan erilaiselta

kuin pelaajan edessä oleva vihollinen. Tämä efekti voidaan saavuttaa myös yhdellä AudioSourcella, käyttämällä skriptiä joka kertoo AudioSourcelle, missä ääni tulisi soittaa riippumatta sen omasta sijainnista. AudioListener-komponentteja sen sijaan voi olla kerralla käytössä vain yksi.

AudioClip kuvaa mitä tahansa ääntä, efektiä, musiikkikappaletta, puhetta tms. joka halutaan soittaa pelissä. Äänet voivat olla useassa eri tiedostomuodossa, mm. MPEG layer 3 (mp3), Wave (wav) ja Ogg Vorbis (ogg) (Unity Manual: Audio 2016).

3 Case: One Finger Death Punch

3.1 Pelin kuvaus

Silver Dollar Games, 2013. Xbox 360, Windows (2014), Android ja iOS (2015).

One Finger Death Punch tiivistää tappelupelin mekaniikat äärimmäiseen yksinkertaisuuteen: pelissä on vain kaksi nappia: vasen ja oikea. Nappia painamalla pelihahmo hyökkää annettuun suuntaan, jos hyökkäyksen ajoittaa oikein, hyökkäys osuu. Väärä ajoitus rankaisee pelaajaa välittömästi pisteiden ja usein hahmon kestopisteiden menetyksellä. Kentät ovat lyhyitä intensiivisiä kokonaisuuksia joissa taistellaan suurta vihollismäärää vastaan. Pelin tempo ja vaadittu APM on niin äärimmäisen korkea, että peli pakottaa pelaajan keskittymään ainoastaan peliin ja rankaisee välittömästi virheitä. Peli myös nopeutuu normaalista jos pelaaja pelaa hyvin ja palautuu takaisin normaalinopeuteen vasta kun pelaaja alkaa tehdä virheitä.

3.2 Flow'n vaatimukset

- 1) Selvät tavoitteet: jokaisen kentän alussa pelaajalle kerrotaan kuinka monta vihollista on voitettava kentän läpäisemiseksi tai onko kentässä

muita erikoissääntöjä. Yhdistettynä äärimmäisen yksinkertaisiin pelimekaniikkoihin tämä tarkoittaa, että pelaaja tietää aina mitä pitäisi tehdä ja voi keskittyä täysin pelin pelaamiseen.

- 2) Palaute: jokainen hyökkäys antaa välitöntä palautetta hyökkäysanimaatioiden, äänien ja efektien avulla. Pelaaja tietää välittömästi hyökkäyksen jälkeen osuiko se vai ei ja osaa sen perusteella valita seuraavan toiminnon. Palaute on täysin integroitu osaksi peliä, eikä keskeytä sitä missään vaiheessa.
- 3) Päämäärät: pelin kampanja on jaettu tehtäviin, jotka ovat kaikki myös osa lyhempiä tehtäväkokonaisuuksia, karttoja. Pelin päämäärät jaetaan kolmeen ryhmään: kentän suorittaminen, kartan kaikkien tehtävien suorittaminen ja kampanjan suorittaminen. Pelaajan eteneminen näkyy kartalla tehtävien välissä joten pelaaja voi helposti seurata etenemistään ja eri päämäärien toteutumista. Haasteen määrä nousee varsin nopeasti ja voi johtaa tilanteeseen jossa pelaaja ei pysty etenemään toistamatta samaa kenttää moneen kertaan. Tämä voi joillakin pelaajilla johtaa turhautumiseen ja pelaamisen lopettamiseen.
- 4) Keskittyminen: pelin käyttöliittymä sisältää paljon tietoa jota ei tarvitse pelatessa ja joka voisi häiritä keskittymistä. Käytännössä suurin osa käyttöliittymästä jää täysin huomioimatta pelaamisen aikana ja siten sisältää lähinnä turhaa tietoa. Pistelaskurit ja kombotiedot kuuluvat genreen, mutta eivät tässä tapauksessa lisää mitään itse pelaamiseen.

3.3 Tekninen toteutus

One Finger Death Punchin graafinen tyyli on erittäin yksinkertainen tikku-ukkografiikka, jossa eri viholliset on värikoodattu niiden ominaisuuksien mukaan. Värikkäät erikoisefektit luovat peliin lisää dramatiikkaa ja antavat pelaajalle välitöntä palautetta pelin tapahtumista. Ääniefektit ovat enimmäkseen normaaleja lyöntiääniä ja toimivat myös yhtenä palautemetodina pelaajalle. Pelin FPS pysyy tasaisesti korkeana ja yhdistettynä sulaviin animaatioihin ja lyhyisiin vasteaikoihin tekee pelistä teknisesti erittäin miellyttävän pelattavan.

3.4 Arvioinnit

One Finger Death Punch on saanut enimmäkseen myönteisiä arvioita, pelin graafisen tyylin ja yksinkertaisten, mutta palkitsevien, pelimekaniikkojen ansiosta. Eurogamer antoi Xbox 360-versiolle 9/10[4], kehuen pelin mekaniikkoja ja graafista toteutusta. *”Soon enough, each mission [...] is a riot of colour and action. [...] What could so easily have become a disposable, short-lived distraction grows into a deep and complex series of tests of nerve, timing and dexterity as Silver Dollar Games twists its central premise into unusual and creative new shapes”* (Parkin 2013).

Pelin virheinä mainitaan usein itsensä toistaminen ja kenttien samankaltaisuus. Steamin käyttäjäarvioissa 98 % arvioista on myönteisiä ja peli onkin kirjoitushetkellä Steamin viidenneksi parhaaksi arvioitu peli (Steampowered.com 2014).

3.5 Yhteenveto

One Finger Death Punch on erittäin onnistunut esimerkki pelistä joka karsii pelimekaniikat minimiin hyvän flow’n luomiseksi. Peli vaatii täydellistä keskittymistä ja palkitsee pelaajaa siitä välittömästi hyvällä palautteella ja erinomaisilla animaatioilla. Jotkin ylimääräiset käyttöliittymäelementit voisivat häiritä peliin keskittymistä, mutta käytännössä ne on pakko jättää huomioimatta pelaamisen aikana.

Pelin nopeutuminen sitä mukaa kun pelaaja oppii pelaamaan paremmin auttaa ylläpitämään haastetta, kunnes pelaaja alkaa epäonnistumaan nopeuden takia jolloin nopeus laskee taas normaaliin ja antaa pelaajan aloittaa nopeuden keräämisen uudelleen. Jos pelaaja joutuu tilanteeseen jossa hän ei pysty läpäisemään kenttiä edes matalimmalla nopeudella hän voi turhautua vaikeuteen ja lopettaa pelaamisen.

4 Case 2: Guitar Hero -sarja

4.1 Pelin kuvaus

Guitar Hero: Harmonix, RedOctane, 2005. Playstation 2.

Guitar Hero II: Harmonix, Activision, 2006. Playstation 2, Xbox 360 (2007)

Guitar Hero III: Legends of Rock: Neversoft, Activision, 2007. Playstation 2, Playstation 3, Xbox 360, Microsoft Windows, Mac OS X, Wii

Guitar Hero World Tour: Neversoft, Activision, 2008. Playstation 2, Playstation 3, Xbox 360, Microsoft Windows, Mac OS X, Wii

Guitar Hero aloitti länsimaissa muovikitarapeligenren suosion ja perustui rakenteeltaan vahvasti japanilaiseen Guitar Freaks-kolikkopelisarjaan (Konami, 1998 - 2012). Guitar Freaksin tuominen Yhdysvaltoihin ei onnistunut kitaraohjaimen patenttiongelmien takia, mikä jätti oven auki Harmonix-studiolle oman version tuomiseen markkinoille (SiliconEra 2009).

Harmonixin luoma peli oli varsin häpeämätön kopio Guitar Freaksistä, mutta se ei estänyt sitä saavuttamasta välitöntä hittistatusta. Pelisarjan kehitys jatkui erittäin nopealla tahdilla ensimmäisen pelin suosion jälkeen ja jatko-osa Guitar Hero II valmistui jo seuraavana vuonna. Guitar Hero II lisäsi peliin tuen bassokitaralle.

Guitar Hero III ilmestyi seuraavana vuonna eri studion (Neversoftin) kehittämänä ja oli hyvin perusvarma jatko aiemmille peleille. Tämä oli sarjan ensimmäinen peli joka julkaistiin myös Nintendo Wiille ja PC-alustoille. Harmonix oli tässä vaiheessa myynyt oikeutensa Guitar Hero-sarjaan ja siirtynyt kehittämään kilpailevaa Rock Band-sarjaa. Guitar Hero World Tour lisäsi peliin tuen laulamiseksi ja rummuille ja mahdollisti neljän pelaajan monin pelin eri soittimilla.

Guitar Hero-peleissä yksi "kenttä" on aina yksi musiikkikappale, joka on soitettava alusta loppuun mahdollisimman tarkasti. Virheiden tekeminen tai nuottien jättäminen soittamatta vie pelaajalta energiaa jonka loputtua kappale päättyy epäonnistumiseen ja pitää aloittaa uudestaan. Onnistuneesti soitetut nuotit palauttavat energiaa pelaajalle ja nostavat pelin pistekerrointa yhdestä neljään. Yksikin virhe palauttaa pistekertoimen takaisin yhteen.

Pelin vaikeusaste vaikuttaa nuottien määrään ja nopeuteen jolla nuotit liikkuvat ja samalla myös pelin APM:ään. Täten korkeimmat pistemäärät on mahdollista saavuttaa vain vaikeimmalla vaikeustasolla jossa on eniten nuotteja soitettavana. Kappaleen loputtua pelaajan suoritus arvioidaan tähdillä: kolme – viisi tähteä. Alle kolme tähteä tarkoittaa epäonnistunutta kappaletta. Tämän lisäksi haasteena on kappaleen soittaminen täysin virheettömästi, eli "Full Combo", jossa pistekerroin ei tipu takaisin yhteen kertaakaan kappaleen aikana. Jokainen pelin soittolistoista vaatii tietyn määrän tähtiä avautuakseen, joten pelaamalla hyvin avautuvat uudet kappaleet nopeammin.

4.2 Flow'n vaatimukset

- 1) Selvät tavoitteet: Pelaajan välittömänä tavoitteena on aina soittaa seuraava kappale mahdollisimman täydellisesti ja saada korkein mahdollinen pistemäärä. Toissijaisena tavoitteena on saada kappale soitettua onnistuneesti, riippumatta pistemäärästä.
- 2) Palaute: Pelaajan tekemistä virheistä saa välitöntä palautetta. Jos soittaa nuotin väärin soitettuna kappaleen kitararaita keskeytyy ja peli soittaa epäsointuisen virheäänänen sen tilalla. Pelaajan energiapalkki laskee ja pistekerroin nolautuu. Mitä lähempänä epäonnistumista on, sitä vähemmän yleisö hurraa ja metelöi kunnes lopulta alkaa buuaamaan pelaajalle. Onnistumisesta palkitaan pistekertoimen kasvamisella ja erilaisilla graafisilla efekteillä. Onnistumiselle tärkein palautemetodi on kappaleen soiminen oikein kokoaikaisesti, ie. virheäänien välttäminen.
- 3) Päämäärät: Pelaajan päämääränä on kappaleiden oppiminen ja eri soittolistojen avaaminen ja suorittaminen ja lopulta kaikkien kampanjan

soittolistojen onnistunut suorittaminen. Pelaaja tietää koko ajan mihin asti hän on edennyt soittolistalla, mitä seuraavan soittolistan avaaminen vaatii ja miten lähellä pelin loppua hän on, eli montako soittolistaa ja kappaletta on vielä avaamatta.

- 4) Keskittyminen: Guitar Hero vaatii pelaajaa keskittymään täysin pelin etenemiseen; varsinkin vaikeammilla vaikeusasteilla nuotit tulevat vastaan nopeasti ja suurissa ryhmissä ja keskittymisen herpaantuminen johtaa nopeasti kappaleen epäonnistumiseen. Pelin käyttöliittymä tukee tätä varsin hyvin, ylimääräistä tietoa ei ole turhaan häiritsemässä ja tärkeimmät elementit on sijoitettu lähelle pelaajan keskittymiskohtaa pelialueella, eli nuottien ja kitaran nappien yhdistymiskohtaa.

4.3 Tekninen toteutus

Guitar Hero-sarjan pelit ovat parantuneet tekniseltä toteutukseltaan huomattavasti sarjan jatkuessa. Ensimmäiset pelit olivat varsin vaatimattomia ulkonäöltään ja sisälsivät paljon epävirallisia cover-versioita suosituista kappaleista.

Guitar Hero III, ensimmäinen Neversoftin kehittämä Guitar Hero-peli, oli graafisesti huomattavasti aiempia pelejä näyttävämpi ("*The game looks fantastic. [...] The art style has managed the developer transition intact, but it looks altogether more polished now.*") ja musiikkilista sisälsi lähes pelkästään alkuperäisiä versioita kappaleista ("*The track list makes or breaks any music game, of course, and thankfully it's impossible to complain about Guitar Hero III's. Gone are the covers [...]*") (MacDonald 2007).

4.4 Arvioinnit

Guitar Hero oli yllätyshitti joka möi 1,5 miljoonaa kopiota ja sai erittäin hyviä arvioita. Eurogamer antoi pelille arvion 8/10 ja kritisoi lähinnä vain pelin musiikkivalikoimaa (Reed 2006).

Guitar Hero II yritti korjata tämän ongelman laajemmalla musiikkivalikoimalla, mutta sai siitä huolimatta arvioksi saman 8/10. Pelin musiikkivalikoima todettiin liian metallivoittoiseksi. Eurogamerin arvioija kommentoi: *"It still gets an 8, then. But it's a really, really high one!"* (MacDonald 2006).

Guitar Hero 3 jatkoi samalla linjalla, parantaen aiemmista peleistä jonkin verran, mutta tekemättä mitään isompia muutoksia. Eurogamer palkitsi parannukset arviolla 9/10 (MacDonald 2007).

4.5 Yhteenveto

Guitar Hero onnistui erittäin hyvin flow'n ylläpitämisessä, varsinkin pidemmissä kappaleissa ja korkeammilla vaikeusasteilla. Hidastempoisissa kappaleissa flow keskeytyy helposti tekemisen puutteen takia. Guitar Heroa on aiemminkin käytetty flow'n testaamiseen: *"The study results revealed that majority of the participants experienced flow while playing guitar hero"* Tämä osoittaa, että aiempaa pelikokemusta ei vaadita Guitar Hero tapauksessa flow'n syntymiseen (Arslantas 2014).

Pelin musiikkikappaleisiin perustuva gameplay loop johtaa helposti "vielä yksi kappale"-tilanteeseen joka auttaa osaltaan ylläpitämään flow'ta. Pelissä käytetty musiikki voi myös itsessään johtaa flow-tilaan pelaajassa, jos kyseessä on kappale josta pelaaja pitää erittäin paljon (Arslantas 2014).

5 Johtopäätökset esimerkkitapauksista

Tutkimieni pelien ja niiden arvioiden perusteella olen päätenyt kolmeen pelimekaaniseen suunnittelukonseptiin jotka auttavat selvästi flow'n ylläpitämisessä:

1. Reaaliaikaisuus, nopea tempo, refleksimäinen core gameplay loop. Lyhyet reaktioajat kasvattavat keskittymisen merkitystä pelin pelaamiseen ja siinä onnistumiseen. Pelin pitää antaa palautetta onnistumisista ja epäonnistumisista välittömästi, jotta pelaaja tietää muokata pelitapaansa kesken pelin miettimättä sitä.
2. Selvästi eritelty peliosuus ja valikko-osuus, pelin alettua sitä ei keskeytetä ennen kuin peliosuuden päätyttyä. Peliosuus on yleensä lyhyt ja sillä on selvä päämäärä. Pelin aloituksen tulee olla helppoa: mitä enemmän oppimista peli vaatii ennen pelin alkua, sitä helpommin pelaaja turhautuu, eikä saavuta flow'ta. *"Games that have naturally shorter turnover tend to have better flow than those whose gameplay is based on long sessions"* (Slabinski 2013).
3. Värien ja symbolien käyttäminen pelimekaniikkojen esittämisessä yhdistettynä käyttöliittymän yksinkertaisuuteen. *"Visual simplicity also plays a part in flow. [...] economy of how much you're showing on screen is necessary to making sure the players never get frustrated with the information presented to them"* (Slabinski 2013).

Näiden asioiden perusteella olen luonut listan käytännön asioista jotka tulisi ottaa huomioon pelin suunnittelussa.

5.1 Reaktioajan vaikutus flow'n saavuttamiseen

Tutkimissani peleissä reaktioajan merkitys flow'n saavuttamiseen on selvä, pelaaja saa välitöntä positiivista palautetta onnistuessaan tehtävässä joka vaatii nopeita reaktioita ja tarkkuutta ja näiden tehtävien linkittäminen yhteen viiveettä

luo pitkäaikaisen flow-kanavan. Keskittymisen peliin on oltava kokoaikaista ja täysivaltaista tai peli rankaisee siitä välittömästi.

Hyvin toteutettuna pelaaja saa epäonnistuessaankin tunteen, että tehtävä on toteutettavissa harjoituksella, joka lisää pelaajan motivaatiota jatkaa pelaamista. Pelin antama palaute on tässä etusijalla, epäonnistumisenkin pitäisi tuntua siltä, että ainakin pelaaja oppi jotain uutta.

5.2 Keskeytysten poistaminen pelistä

Mikään ei tuhoa pelaajan flow'ta tehokkaammin kuin pelin keskeyttäminen tarpeettomalla välivideolla tai pitkällä animaatiolla kuoleman tai muun epäonnistumisen yhteydessä. Mitä lyhempi aika epäonnistumisen ja pelin uudelleen aloittamisen välillä on, sitä paremmin peli ylläpitää flow'ta. *"It's also important to not prolong the sense of failure a player experiences. If they lose, the fact that they lost at all is punishment enough. [...] Failure is immediately forgotten in favor of trying again"* (Slabinski 2013).

Tutoriaalit ja ohjeet annetaan pelaajalle ennen peliosiota tai välittömästi sen alussa ja pelaajalle annetaan kerralla kaikki tarvittavat tiedot pelissä etenemiseen. Jos tämä johtaa liialliseen tietoon (*"information overload"*), pelin suunnittelussa on ongelmia. Peli pitäisi pystyä selittämään parilla lauseella, jos tavoitteena on hyvä flow.

5.3 Värien käyttö symboleina

Yksinkertaisimmillaan värejä voidaan käyttää luokittelemaan peliolioita: punaiset oliot ovat vihollisia tai vaaroja, vihreät ystäviä tai apuvälineitä.

Yhdistämällä eri väreihin eri muotoja tai siluetteja voidaan luoda visuaalinen kirjasto eri peliolioista ja niiden merkityksistä, kertomatta näitä asioita pelaajalle suoraan missään vaiheessa.

5.4 Musiikin ja rytmin vaikutus pelin flow'hun

Pelit joiden musiikki sopii hyvin pelin tempon kanssa saavuttavat flow-tilan pelaajissaan helpommin. Musiikin kirjoittaminen tukemaan tapaa jolla pelaaja etenee pelissä vahvistaa pelaajan immersiota peliin, keskittymistä ja siten flow-tilan saavuttamista. Vaikka tämän tutkimuksen käytännön osuudessa ei keskitytä musiikkiin, tai sen käyttöön suunnittelussa, on tämä asia kuitenkin tärkeää mainita (Sweet 2014, 19).

Musiikki peleissä voidaan jakaa useisiin ryhmiin, joista flow'n kannalta tärkeimmät ovat "Music as Gameplay", jossa musiikki sitoutuu suoraan pelimekaniikkoihin (esim. Guitar hero, AudioSurf), sekä "Extra-Diegetic Music", jossa musiikkia käytetään vahvistamaan pelaajan tuntemia tunteita (Sweet 2014, 21-26).

Musiikin kuuntelu itsessään voi aiheuttaa kuulijassa flow-tilan ja yhdistettynä esim. urheiluun tai muuhun fyysiseen toimintaan musiikki auttaa keskittymään toimintaan ja sulkemaan häiriöt ulos. Musiikkia voidaan käyttää myös rytmittämään peliä jolloin se lisää pelaajan käsitystä pelin hallinnasta: jos toiminnat sijoittuvat musiikin rytmiin, siitä tulee osa pelin informaatiota jota pelaaja voi joko aktiivisesti tai alitajuisesti käyttää toimintojen ajoittamiseen (Whalen 2004).

6 Käytännön toteutus

Kerättyjen tietojen ja niistä tehtyjen päätelmien pohjalta lähdin toteuttamaan yksinkertaista Unity-pohjaista pelidemoa, käyttäen Unity-pelimoottorin 2D-tilaa. Pelin ideana on suojella kaupunkia hyökkäviltä zombeilta reaaliaikaisessa nopeatempoisessa taistelutilassa ja rakentaa kaupungin suojauksia taisteluiden ulkopuolella.

Peli on suunniteltu toimimaan sekä hiirellä että kosketusnäytöillä ja sitä on tarkoitus testata Windowsin lisäksi Android-tabletilla. Inspiraationa demoon oli Atom Zombie Smasher joka käyttää samoja perusideoita, mutta toteuttaa ne eri tavalla.

6.1 Kehitysympäristö

Demon kehitysympäristönä toimi 64-bittinen Unity (versio 5.3) Windowsissa, skriptauskielenä käytettiin C#-kieltä. Demossa käytetään ilmaisia Unity Storesta ladattuja ääniä ja musiikkia.

Skriptauksessa käytetään Visual Studiota, joka on korvannut MonoDevelopin Unityn virallisena ohjelmointiympäristönä.

Demosta toteutetaan erikseen Windows, Linux ja Android-versiot, joiden luonti onnistuu suoraan Unityn build-työkaluilla.

6.2 Suunnittelu

Tehdyn tutkimustyön perusteella löydettyjä flow'ta auttavia elementtejä käytettiin uuden pelin suunnittelun pohjana: pelin tulisi olla reaaliaikainen, ilman keskeytyksiä pelitilassa, graafisesti yksinkertainen, mutta yhtenäisesti värikoodattu ja helppo oppia.

Pelissä pääosassa oleva kaupunki on jaettu yhdeksään sektoriin, joista jokainen tuottaa pelaajalle resursseja, joita käytetään palkkaamaan sotilaita, ostamaan aseita ja tulitukea ja rakentamaan barrikadeja. Keskimäinen sektori 3 X 3 kartalla on pelaajan päämaja jonka tuhoutuminen lopettaa pelin. Peli loppuu myös jos pelaaja selviää niin pitkään, että jokainen selviytyjä on saatu evakuoitua. Jokainen päivä tietty määrä selviytyjiä evakuoidaan helikopterilla pois kaupungista, kunhan selviytyjiä on riittävä määrä odottamassa pelaajan päämajassa. Pelin päätyttyä pelaaja saa arvosanan sen perusteella, kuinka moni kaupungin asukkaista selvisi hengissä.

Menettämällä muita sektoreita zombeille tai joutumalla evakuoimaan ne pelaajan päämajaan täysin pelaaja menettää sektorien tuottamat resurssit pysyvästi: kun sektori on kerran tuhattu, sinne ei pääse takaisin. Pelissä tätä kuvataan ydinräjähdyksellä, joka tuhoaa menetetyn sektorin ja kaikki siellä olevat zombit kerralla. Tämä luo peliin jatkuvuutta ja lisää merkitystä jokaiseen taisteluun: jokainen hävitty taistelu tekee seuraavasta hyökkäyksestä hankalamman torjua resurssien heikentyessä. Samalla pelaaja oppii toivottavasti hyödyntämään resurssejaan tehokkaammin taistelusta toiseen.

Valmisteluvaiheessa pelaaja näkee seuraavan päivän aikana tapahtuvien hyökkäysten suunnan ja voi asetella kartalle sotilaita, räjähteitä ja barrikadeja. Hyökkäysvaiheessa pelaaja siirtää sotilaita pelastamaan siviilejä, käyttää räjähteitä, kutsuu ilmatukea ja ohjaa siviilejä evakuointiasemiin. Tehtävänä on siis saada mahdollisimman moni siviili turvaan, niin, että sektori ei tyhjene täysin ja jatkaa resurssien tuottamista. Pelin aikana zombien tappamat siviilit osallistuvat seuraavan päivän hyökkäykseen, joka osaltaan lisää jokaisen hyökkäyksen merkitystä.

Värejä käytetään pelissä kuvaamaan eri olioita: vihreät pallot ovat siviilejä, siniset sotilaita. Pallojen koot kuvaavat niiden suhteellista yksikkömäärää. Siniset osat kartasta ovat evakuointiasemia. Viholliset ovat punaisia, samoin vihollisten hyökkäysnuolet jotka kuvaavat zombien hyökkäyssuuntia, sekä omien ilmaiskujen yms. hyökkäysten osuma-alueet, koska ne tappavat sekä omia että vihollisia.

6.3 Flow'n vaatimukset suunnitelmassa

Valmiin suunnitelman vertaaminen flow'n vaatimuksiin:

- 1) Selvät tavoitteet: Pelaajan välittömänä tavoitteena on pelastaa mahdollisimman monta siviiliä. Toissijaisena tavoitteena on vihollisten tappaminen mahdollisimman tehokkaasti. Tavoitteen saavuttamisen hyödyt kerrotaan pelaajalle ennen tehtävien alkua.
- 2) Palaute: Pelastetut siviilit, tapetut siviilit ja kuolevat zombit soittavat kaikki oman ääniefektinsä joka toimii palautteena tehtävän etenemisestä. Kaikki pelissä suoritettavat toiminnat soittavat oman ääniefektinsä aktivoituessaan.
- 3) Päämäärät: Pelin päämääränä on kaupungin koko siviiliväestön pelastaminen mahdollisimman pienillä tappioilla. Peli seuraa ja raportoi pelaajan etenemisestä jokaisen taistelun jälkeen, joten pelaaja voi seurata päämäärän saavuttamista helposti.
- 4) Keskittyminen: Pelin tempo pakottaa pelaajan keskittymään eri tehtävien suorittamiseen mahdollisimman nopeasti ja tehokkaasti. Virheliikkeistä rangaistaan välittömästi ja koska peli on reaaliaikainen, virheitä on hyvin vaikea korjata jälkikäteen.

Flow'n vaatimukset täyttyvät suunnitelmassa kaikilta osin ainakin jossain määrin. Peli on jaettu erillisiin peli- ja valikko-osuuksiin, joista valikko-osuudessa ei välitetä flow'sta ja annetaan pelaajan rauhassa rentoutua ja tutkia pelin tapahtumia. Peliosuudessa ei sallita flow'ta haittaavia keskeytyksiä.

6.4 Kehitystyö

Demon kehitystyö alkoi maaliskuussa eri peliolioiden luokkien luomiselle, tässä vaiheessa pelissä olivat mukana zombit, siviilit, sotilaat ja rakennukset sekä siviileille evakuoitipiste johon pääseminen poisti siviilin pelikentältä ja lisäsi sen pelastettujen siviilien listalle. Pelaajan aktiiviset toiminnot puuttuivat vielä, pelaaja

pystyi vain klikkaamalla rakennusta tyhjentämään sen siviileistä, kaikki muu oli vielä automatisoitua.

Peliolioilla oli alusta lähtien yksinkertainen tekoäly, jossa siviilit pyrkivät aina pääsemään evakuointipisteeseen, zombit jahtasivat lähintä siviiliä ja siviilien loputtua hyökkäsivät rakennusten kimppuun joissa oli vielä siviilejä sisällä. Sotilaat seisoivat paikallaan ja hyökkäsivät riittävän lähellä olevien zombien kimppuun. Sotilaat eivät vielä olleet zombeille hyökkäyskohde.

Huomasin nopeasti eri kenttiä testatessa, että rakennukset eivät lisänneet demoon mitään järkevää tekemistä: ne toimivat vain ylimääräisenä häiriötekijänä joka oli erittäin helppo unohtaa kokonaan, kunnes kaikkien siviilien kuoltua tai paettua zombit hyökkäsivät rakennusten kimppuun ja pelaaja muisti niiden olemassaolon. Rakennukset olivat siten ensimmäinen pelimekaniikka joka karsiutui pois demosta, lopullisessa versiossa kaikki siviilit aloittavat kentät valmiiksi evakuointitilassa.

Perusrakenteen valmistuttua keskityin rakentamaan peliin mekaniikkoja, eli pelaajalla tekemistä. Tässä vaiheessa päädyin ideaan eri klikkausmoodeista, joissa pelaajan hiiren vasen nappi tekee eri asioita. Moodit valitaan klikkaamalla ruudulla olevia nappeja, joista aktiivinen nappi on aina vihreänä.

Perustila Interact mahdollistaa ylimääräisen nopeus-piikin ("Speed Boost") antamisen siviileille klikkaamalla siviilejä ja zombien hidastamisen väliaikaisesti klikkaamalla zombeja ("Zombie Distractor"). Jokainen kenttä alkaa Interact-tilassa ja nämä toiminnot ovat aina käytössä.

Toinen tila oli Soldier, joka loi kentälle sotilaan klikattuun paikkaan. Tässä vaiheessa sotilaiden tekoälyä oli parannettu ja ne jahtasivat nyt aktiivisesti zombeja jotka eivät olleet hyökkäysetäisyydellä. Zombit myös hyökkäsivät nyt sotilaiden kimppuun jos ne tulivat liian lähelle. Sotilaat pysyvät kentällä enintään viiden sekunnin ajan ja tuhoutuvat sen jälkeen. Käytännössä sotilaat pysyvät harvoin elossa täyttä viittä sekuntia.

Kolmas tila Artillery laukaisee tykistöiskun klikattuun kohtaan kartalla. 1.3 sekunnin viiveen jälkeen tykistöisku tuhoaa kaikki oliot osuma-alueella, myös siviilit ja sotilaat. Tykistöiskun vaikutusalue jää myös paikalleen sekunniksi jona aikana kaikki alueelle liikkuvat oliot tuhoutuvat.

Neljäs tila Barricade oli seuraava alkuperäisestä suunnitelmasta karsittu mekaniikka. Klikkaamalla kenttää Barricade-tilassa pelaaja luo zombien liikkumisen estävän muurin vähäksi aikaa, valitettavasti testauksessa totesin ajatuksen mahdottomaksi tasapainottaa yhdistettynä sotilaisiin ja tykistöiskuihin. Muurin taakse kerääntyvät zombiet oli aivan liian helppo tappaa yhdellä tykistöiskulla. Koska pidin tykistöiskun mekaniikkaa paljon mielenkiintoisempina kuin muurien pystyttämistä, Barricade-tila poistettiin demosta. Muurit myös toimivat selvänä flow'ta estävänä tekijänä koska ne keskeyttivät pelin etenemisen hetkeksi.

Tässä vaiheessa kehitystä tuli myös selväksi, että resurssien kerääminen eri sektoreilta ei olisi kannattavaa demon mittakaavassa. Jos pelaajalla ei olisi mahdollisuutta käyttää eri toimintoja kentän aikana, pelin pelaaminen jäisi sen katsomiseksi ja siviilien hoputtamiseksi ja zombien häiritsemiseksi, joka ei vastaa demon tarkoitusta, flow-kokemusten kartoittamista. Sen sijaan, että pelaaja ostaisi tykistöiskuja ja sotilaita, ne ovat aina käytettävissä, mutta vain muutaman sekunnin välein. Käytettyjen toimintojen väri muuttuu pelitilassa punaiseksi kertomaan, että toiminto ei ole väliaikaisesti käytettävissä. Tämä yksinkertaistaa pelikokemusta, mutta keskittää sen niihin asioihin joita ollaan tutkimassa.

Tässä vaiheessa demon perusrakenne oli valmis, pelimekaniikat olivat paikallaan, kaikki pelioliot olivat toimintavalmiita, puuttui vain kampanjan rakenne. Alkuperäisen suunnitelman idea eri sektoreista säilyi, mutta erillisen kaupunkikartan sijasta sektoreilla käydään yksi kerrallaan ja niitä on vain viisi. Kahdeksan sektoria tuntui liialliselta demon mittakaavassa. Näin demo on jaettuna viiteen sektoriin ja viiteen päivään joista jokaisena käydään eri sektorilla pelastamassa mahdollisimman monta siellä asuvaa siviiliä. Päivien välissä pelaaja palaa Base-näkymään aloittamaan seuraavan päivän.

Viidennen päivän jälkeen peli koostaa pelaajan tulokset yhteen ja nolaa pelitilan uutta yritystä varten. Tässä vaiheessa pelaajalle kerrotaan kuinka monta siviiliä on pelastettu, kuinka monta zombia on tapettu ja mikä pelaajan pistemäärä oli. Peli myös tallentaa pelaajan saavuttamat korkeimmat pisteet päävalikossa näkyvään highscoreen seuraavaa yritystä varten. Highscore tallennetaan Unityn Playerprefs-luokkaan, joten pistemäärä pysyy tallessa ohjelman eri käynnistyskertojen välissä.

Loin erillisen ohjeruudun testaajille, jossa selitetään demon päämäärä, eri toiminnot jotka pelaajalla on käytössään ja eri oliot joita pelissä tulee vastaan. Ruutu sisältää kuvaukset kaikista eri olioista ja tekstiohjeet niiden käyttöön. Tätä varten käytetyistä olioista on erilliset dummy-versiot joissa ei ole skriptejä, eikä collision- tai trigger-alueita. Samalla lisäsin pelin Base-näkymään uuden toiminnallisuuden, joka luo graafisen esityksen pelastettujen siviilien määrästä. Mitä enemmän siviilejä pelaaja on pelastanut, sitä useampi dummy-siviili-olio luodaan Base-näkymään.

Seuraava kokonaisuus, joka demosta puuttui, olivat äänet ja taustamusiikki. Näiden kehityksessä tuli avuksi Unityn Asset Store ja sieltä löytyvät ilmaiset äänipaketit. Kävin läpi useita äänipaketteja, mutta lopullisessa versiossa demosta on käytetty kahta äänipakettia ääniefekteille: Little Robot Sound Factoryn Horror Sfx eri olioiden kuolin äänille ja saman yrityksen Sci-Fi Sfx menuelementeille, sotilaiden aseille ja tykistöiskun äänelle. Taustamusiikkina toimii Ryunon Dark Ambient Loop, joka on myös ilmainen Unity Asset. Taustamusiikki soi koko pelin ajan vaihtumatta.

Päätin toteuttaa äänet erillisellä audioHandler-oliolla joka luodaan pelin alussa ja joka sen jälkeen hoitaa itsenäisesti kaikkien äänien soittamisen. Näin oli helppo keskittää äänien soittaminen yhteen paikkaan niin, että eri olioiden tuhoaminen ei häiritse äänien soittamista. Unityssa olion tuhoaminen keskeyttää myös sen soittamat äänet, joka lopettaa esim. kuolin äänien soittaminen välittömästi. Tämän ongelman voi ohittaa joko siirtämällä äänien hallinnan olion ulkopuolelle tai keskeyttämällä olion tuhoamisen kunnes ääni on soitettu loppuun. Näistä vaihtoehtoista totesin ensimmäisen järkevämmäksi toteuttaa.

Yhden audioHandler-olion ja sen AudioSourcen käyttö myös parantaa demon suorituskykyä, verrattuna tilanteeseen jossa jokaisella oliolla on oma audioHandler ja vain sen soittamat AudioClipit. audioHandler-Olion rakenteena on AudioSource-komponentti, joka määrittää äänen soiton asetukset kuten äänenvoimakkuuden, sekä kaikki demossa tarvittavat äänet omina AudioClip-muuttujinaan ja metodit jokaisen äänen soittamiseen. Tälle olisi vaihtoehtona se, että jokainen olio lähettää metodikutsussa tarvittavan AudioClipin audioHandlerille soitettavaksi. Tämä tekisi oliokirjaston laajentamisen helpommaksi ja olisi todennäköisesti paremmin skaalautuva ratkaisu jos demosta kehitettäisiin täysi peli.

Yhden audioHandler-olion ratkaisu teki helpoksi lisätä demon päävalikkoon äänenvoimakkuuden säätövalikon ja varmistaa myös eri äänien soivan aina samalla äänenvoimakkuudella. Valittu äänenvoimakkuus tallennetaan Unityn PlayerPrefs-olioon, joten se säilyy eri pelikertojen välillä.

Graafisesti demo pohjautuu lähinnä valmiisiin Unityn geometrisiin muotoihin, sekä joihinkin ohjelmoijataide-elementteihin, kuten kuolleiden olion jäljiltä jääneisiin veriroiskeisiin. Tässä oli taustalla ajatus pelin värikoodaamisesta, jossa jokaisen olion väritys vastaa tiettyä oliotyyppiä tai mekaniikkaa. Kaikki vihreät asiat ovat hyviä, kaikki punaiset asiat huonoja. Sininen väri kuvaa auktoriteettia, eli sotilaita ja evakuoitipisteitä. Tykistöiskun jäljeltä jäävä musta alue kuvaa kuolemaa.

Teknisiltä vaatimuksiltaan demo oli varsin kevyt ja sen FPS oli tasaisesti vähintään vaadittu 60, ilman FPS:n rajoitusta demo pyörii kehityskoneella n. 200 FPS.

6.4 Lopputulos

Demon ensimmäinen versio oli valmis 1.5.2016 ja se meni välittömästi testaukseen kahdelle henkilölle, joiden antaman palautteen avulla demoa

paranneltiin vielä saman päivän aikana. Tämän jälkeen versio lukittiin kyselyn avulla tehtävää testausta varten, jotta kaikilla testaajilla olisi käytössä sama versio demosta. Demosta luotiin erillinen Linux-versio Windows-version lisäksi, jotka saamani palautteen mukaan toimivat identtisesti. Android-versio luotiin samalla, mutta sitä ei laitettu julkiseen jakoon ja sen testaaminen jäi sisäiseksi kokeiluksi.

Demo on muuttunut paljon alkuperäisestä suunnitelmasta, osittain kehityksen aikana tapahtuneen testauksen takia, osittain demon mittakaavan skaalauksen takia. Osaan alkuperäisen suunnitelman asioista ei ollut riittävästi aikaa, osa olisi ollut haitallista flow'n saavuttamiselle. Lopputulos on virtaviivaistetumpi, mutta täyttää edelleen asettamani vaatimukset flow'n testaukselle.

6.5 Käytännön osuuden testaaminen

Demo laitettiin testaukseen 1.5.2016 illalla yhdistettynä demoon liittyvään Google Formsilla toteutettuun kyselyyn ja demon vapaaseen kommentointiin. Kyselyyn vastattiin anonymisti. Käytetty kysely pohjautuu tohtori Simon Mossin kokoamaan, Mihaly Csikszentmihalyin tekemään tutkimukseen pohjautuvaan listaan asioista jotka yhdistävät flow-tilassa olevia ihmisiä. Näitä asioita ovat mm. suoritettujen tehtävien haasteellisuus, hallinnan tunne, ajantajun hämärtyminen, huolettomuus ja keskittyneisyys (Moss 2008).

Kyselyssä pyydetään testaajia pelaamaan peliä ensin ja vastaamaan kyselyyn välittömästi sen jälkeen. Osana kyselyä pyydetään testaajia arvioimaan kuinka pitkään he demoa pelasivat, joten kyselystä ei anneta etukäteen aika-arviota kuinka pitkään osallistuminen vie. Aika-arvion saaminen voisi vaikuttaa testaajien omaan arvioon käytetystä ajasta. Kyselyssä oli 14 kysymystä ja avoin kommenttikenttä muulle palautteelle.

6.6 Flow'n vaatimukset valmiissa demossa

Flow'n vaatimusten vertaaminen suunnitelman ja valmiin demon välillä:

- 1) Selvät tavoitteet: Pelaajan välittömänä tavoitteena on pelastaa mahdollisimman monta siviiliä. Toissijaisena tavoitteena on vihollisten tappaminen mahdollisimman tehokkaasti. Tavoitteen saavuttamisen hyödyt kerrotaan pelaajalle demon ohjeistuksessa. Verrattuna suunnitelmaan, tavoitteiden saavuttamisen hyödyt eivät ole yhtä merkittäviä pelin etenemiselle. Tätä on jossain määrin korvattu lisäämällä demoon pistelasku ja highscore-listaus.
- 2) Palaute: Pelastetut siviilit, tapetut siviilit ja kuolevat zombit soittavat kaikki oman ääniefektinsä joka toimii palautteena tehtävän etenemisestä. Kaikki pelissä suoritettavat toiminnot soittavat oman ääniefektinsä aktivoituessaan. Tältä osin suunnitelman ja demon välillä ei ole eroa.
- 3) Päämäärät: Pelin päämääränä on kaupungin koko siviiliväestön pelastaminen mahdollisimman pienillä tappioilla. Toisin kuin suunnitelmassa, demossa kaupungilla ei ole ilmoitettua väestön maksimimäärää, joten päämäärän saavuttamisen seuraaminen ei onnistu yhtä helposti.
- 4) Keskittyminen: Pelin tempo pakottaa pelaajan keskittymään eri tehtävien suorittamiseen mahdollisimman nopeasti ja tehokkaasti. Virheliikkeistä rangaistaan välittömästi ja koska peli on reaaliaikainen, virheitä on hyvin vaikea korjata jälkikäteen.

Toisin kuin alkuperäisessä suunnitelmassa, demossa vältettiin lisäämästä peliin etenemistä keskeyttäviä toimintoja (resurssien keräys, barrikadit ja rakennukset), mikä paransi demon pelimekaniikkojen toimivuutta. Päämäärien seuraaminen ja niiden merkitys pelin etenemiseen jäivät paljon vähäisemmiksi kuin alun perin suunniteltiin. Palautteen ja keskittymisen osalta flow'n vaatimukset olivat samalla tasolla suunnitelman ja valmiin demon välillä.

7 Yhteenveto tutkimuksesta ja käytännön toteutuksesta

7.1 Kyselyvastaukset

Pelaajien käyttämä aika demon pelaamiseen vaihteli välillä 5 – 15 minuuttia. Antamissaan aika-arvioissa yksi vastaaja arvioi käyttäneensä peliin vähemmän aikaa kuin oikeasti oli käyttänyt, muiden aika-arviot olivat oikeat. Tämä vastaaja myös vastasi kokeneensa ajantajun hämärtymistä pelin aikana, joka on yksi tärkeimmistä flow'n kriteereistä. Kaikki vastaajat arvioivat ymmärtävänsä pelin päämäärän annettujen ohjeiden perusteella ja pitivät sitä kiinnostavana, mutta eivät kokeneet ymmärtävänsä pelin kontrolleja tai pelimekaniikkoja hyvin.

Pelaajien keskittyminen sijoittui 1-5 asteikolla välille 3-5, jossa 1 on täydellinen keskittymisen puute ja 5 täydellinen keskittyminen. Pelaajien keskimääräinen arvio pelin haasteesta oli haastava/liian haastava, vain yksi vastaaja arvioi pelin haasteen liian helpoksi. Keskittymisen aste suhteessa pelin haastavuuteen oli odotetulla tasolla, mutta keskittymisen, haasteen ja flow'n korrelaatio ei toteutunut kyselyvastauksissa odotetulla tavalla. Myös pelaajat jotka eivät vastausten perusteella kokeneet flow'ta, kokivat pelin haastavaksi ja keskittyivät siihen voimakkaasti.

Pelin ulkonäkö koettiin hyödylliseksi pelin pelaamisessa, helposti käsitettävät symbolit koettiin helposti tunnistettaviksi ja toisistaan erottuviksi. Yksi vastaaja kommentoi, ”...*vaikka grafiikka on selkeästi placeholderia, se toimii musta oikein hyvin: väri- ja muotokoodattuna hahmot tunnistaa [sic] ja erottaa toisistaan todella helposti*” (Vastaaja 2).

Teknisiä ongelmia ei raportoitu yhtään ja kaikki testaajat pelasivat pelin läpi ainakin kerran.

Kyselyn kysymykset ja vastaukset on liitetty opinnäytetyöhön erillisenä liitteenä.

7.2 Flow-kokemusten arviointi

Saatuisten vastausten perusteella voidaan välittömästi sanoa yhden vastaajan kokeneen flow-tilan ajantajun hämartyksen perusteella. Vastaja myös itse arvioi pelanneensa peliä vähemmän aikaa kuin oikeasti tapahtui. Muiden testaajien osalta flow'n arviointi on käytävä läpi yksi kerrallaan, käyttäen alla olevaa laskukaavaa.

Kysymys	Skaala	Pisteet
Kiinnostus pelin päämäärästä	Kyllä/Ei	-2/+1
Keskittyminen	1-5	-2/+2
Haaste	1-5	-2/+2
Pelin hallinta	Kyllä/Ei	+1/-1
Ajantajun hämartyminen	Kyllä/Ei	+3/-1
Pelin ulkonäkö	1-5	-2/+1
Pelin palkitsevuus	1-5	-2/+2
Halu pelata peliä uudestaan	Kyllä/Ei	+1/-2
Yhteensä		-11/+11

1-5 skaalalla 1 antaa vasemmanpuoleisen pistemäärän, 2 puolet siitä, 3 on aina neutraali 0, 4 antaa puolet oikeanpuoleisesta pistemäärästä ja 5 oikeanpuoleisen pistemäärän. Heikosti positiivinen lopputulos kuvaa mahdollista flow-kokemusta, vahvasti positiivinen lopputulos (5+) selvää flow-kokemusta.

Vastaja	Flow-arvio
Vastaja 1	-1
Vastaja 2	+7,5
Vastaja 3	+3,5
Vastaja 4	-0,5
Vastaja 5	+3

Näiden vastausten perusteella Vastaja 2 saavutti selvästi flow-tilan ja Vastajat 3 ja 5 kokivat mahdollisesti flow-tilan jossain vaiheessa peliä. Vastajat 1 ja 4 eivät todennäköisesti kokeneet flow-tilaa missään vaiheessa peliä.

7.3 Flow-kokemusten estymisen tai keskeytymisen syyt

Liiallinen vaikeus ja pelin nopea tempo olivat pääasialliset syyt joiden takia pelaajat raportoivat flow-kokemusten estyneen. Vastaajat 3 ja 5 kommentoivat pelin tarvitsevan pikanäppäimet eri toiminnoille sen korkean tempon takia. Tutoriaalin puute ja pelimekaniikkojen ymmärtäminen olivat toinen tuloksissa näkyvä ongelma, vastaajat eivät tunteneet olevansa pelitapahtumien hallinnassa, joka johti flow'n rikkoutumiseen.

7.4 Kehitystoiveet, parannusehdotukset

Yksi parannusehdotus joka tuli useammalta kyselyyn osallistuneelta testajaalta oli pikanäppäimien lisääminen eri toiminnoille, tai eri hiiren nappien käyttäminen eri toiminnoille. Jotkut testaajat kokivat demon tempon liian kovaksi pelkästään hiirivetoiselle pelille, jolloin he eivät enää kokeneet olevansa pelin tapahtumien hallinnassa. Vastaavasti riskinä pikanäppäimissä on pelikokemuksen liiallinen helpottuminen, joka voi johtaa tylsistymiseen ja flow'n rikkoutumiseen. Pelikokemusta pitäisi siten vaikeuttaa muilta osin.

Lisäksi parannusehdotuksena mainittiin Interact-tilan toimintojen palaute, joka koettiin riittämättömäksi. Toimintoihin kaivattiin parempaa äänipalautetta tai muuta efektiä joka näyttää jotain tapahtuneen pelissä. Tämä on selvästi flow'ta haittaava tekijä, jonka korjaamisessa ei olisi mitään haittapuolta.

Base-näkymää ei koettu tarpeelliseksi ja sen olisi voinut jättää pois demosta. Nykyisellään Base-näkymä näyttää vain pelastettujen siviilien määrän ja toimii eri kenttien latausruutuna. Näkisin paremmaksi ratkaisuksi näkymän toiminnallisuuden laajentamisen sen poistamisen sijaan, siirtyminen suoraan kentän tulosruudusta seuraavaan kenttään tuntuisi todennäköisesti liian äkkinäiseltä.

7.5 Yhteenveto

Tehdyn tutkimuksen pohjalta löydettiin lista ominaisuuksista, jotka kannattaisi ottaa pelisuunnittelussa huomioon, jos tavoitteena on hyvä flow pelissä. Näitä olivat pelin reaaliaikaisuus, nopea tempo, reaktioihin pakottava core gameplay loop, värien käyttö symboloimaan eri asioita ja keskeytysten poistaminen pelistä erottamalla peliosuuden valikoista. Negatiiviset asiat jotka löydettiin ja joita tulisi välttää, jos peliin halutaan hyvä flow, olivat keskeytykset pelin etenemisessä ja pelin hallinnan poistaminen pelaajalta, tekniset ongelmat (erityisesti matala FPS) ja niiden aiheuttamat häiriötekijät sekä haasteen tasapainottaminen pelaajan taitojen mukaan.

Tutkimuksen perusteella voidaan myös todeta, että flow ei ole itsessään hyvän pelin merkki, pelkästään flow'n saavuttamiseen pyrkiminen rajoittaa pelisuunnittelua huomattavasti. Realistisempi tavoite on jakaa peli eri osiin, joista osa suunnitellaan flow tavoitteena ja osa käyttäen normaaleja suunnittelumetodeja. Näin voidaan lisätä hidastempoiseen peliin nopeatempoisia osia jotka käyttävät flow'ta suunnittelupohjanaan. Flow'n saavuttaminen on myös erittäin subjektiivinen asia, joka todettiin selvästi toteutetussa kyselyssä: eri vastaajien arviot pelistä vaihtelivat laidasta laitaan.

Käytännön toteutuksessa löydettiin joitakin pelimekaanisia ratkaisuja jotka auttoivat flow'n saavuttamisessa. Näitä olivat symbolien käyttö kuvaamaan eri olioita ja olioiden värikoodaaminen kuvaamaan eri oliotyyppejä. Muuten käytännön toteutus jäi ohuemmaksi kuin olisin toivonut ajanpuutteen takia, kyselyvastauksien avulla demoon löytyi lista parannuksista, joiden toteuttaminen varmasti parantaisi pelin flow'ta.

Lähteet

- Csikszentmihalyi, M. 1990. Flow – The Psychology of Optimal Performance. New York: Harper Perennial.
- Murphy, C. 2011. Why Games Work and the Science of Learning. http://www.goodgamesbydesign.com/Files/WhyGamesWork_TheScienceOfLearning_CMurphy_2011.pdf. 17.12.2015.
- Chen, J. 2007. Flow in Games (and Everything Else). <http://www.ccs.neu.edu/home/lieber/courses/cs4500/sp09/resources/p31-chen-flow-in-games.pdf>. 23.3.2016.
- jca. 2003. BWChart FAQ. <http://bwchart.teamliquid.net/us/bwfaq.php>. 3.5.2016.
- Ask a Game Dev. 2014. <http://askagamedev.tumblr.com/post/101776602061/progression-or-skill-tree-systems-the-type-you>. 6.5.2016.
- Unity FAQ. 2016. <http://unity3d.com/unity/faq>. 8.5.2016.
- Unity Company Facts. 2016. <https://unity3d.com/public-relations>. 8.5.2016.
- Unity Manual: Physics. 2016. <http://docs.unity3d.com/Manual/PhysicsSection.html>. 8.5.2016.
- Unity Scripting API: PlayerPrefs. 2016. <http://docs.unity3d.com/ScriptReference/PlayerPrefs.html>. 8.5.2016.
- Unity Manual: Scenes. 2016. <http://docs.unity3d.com/Manual/CreatingScenes.html>. 8.5.2016.
- Singleton Pattern. 2016. <http://www.oodeesign.com/singleton-pattern.html>. 8.5.2016.
- Unity Manual: Audio. 2016. <http://docs.unity3d.com/Manual/AudioOverview.html>. 8.5.2016.
- Silver Dollar Games. 2014. One Finger Death Punch. <http://store.steampowered.com/app/264200/>. 17.12.2015.
- Parkin, S. 2013. One Finger Death Punch Review. <http://www.eurogamer.net/articles/2013-08-28-one-finger-death-punch-review>. 17.12.2015.
- SiliconEra. 2009. What Guitar Freaks Could Have Been Like in North America. <http://www.siliconera.com/2009/06/05/what-guitar-freaks-could-have-been-like-in-north-america/>. 26.1.2016.
- Reed, K. 2006. Guitar Hero. http://www.eurogamer.net/articles/r_guitarhero_ps2. 3.5.2016.
- MacDonald, K. 2006. Guitar Hero 2. http://www.eurogamer.net/articles/GH2_eg_rev. 3.5.2016.
- MacDonald, K. 2007. Guitar Hero III: Legends of Rock. <http://www.eurogamer.net/articles/guitar-hero-iii-legends-of-rock-review>. 23.3.2016.

- Arslantas, T, Arslan, O & Çağiltay, K. 2014. Investigating Game Player's Flow Experiences: The Case of First Time Guitar Hero Players. https://www.researchgate.net/publication/282804550_Investigating_Game_Player's_Flow_Experiences_The_Case_of_First_Time_Guitar_Hero_Players. 29.3.2016.
- Slabinski, M. 2013. Designing Games with Flow in Mind. http://www.gamasutra.com/blogs/MarkSlabinski/20130414/190449/Designing_Games_with_Flow_in_Mind.php. 29.3.2016.
- Sweet, M. 2014. Writing Interactive Music for Video Games: A Composer's Guide. Boston: Addison-Wesley Professional.
- Whalen, Z. 2004. Play Along - An Approach to Videogame Music <http://www.gamestudies.org/0401/whalen/>. 5.5.2016.
- Moss, S. 2008. Flow theory. <http://www.psych-it.com.au/Psychlopedia/article.asp?id=62>. 3.5.2016.

”Flow’n hyödyntäminen videopeleissä”-kysely

Timestamp	Tunsitko ymmärtäväsi pelin päämäärän?	Oliko pelin päämäärä mielestäsi kiinnostava?
5.1.2016 22:17:02	Kyllä	Kyllä
5.1.2016 23:02:13	Kyllä	Kyllä
5.2.2016 16:01:52	Kyllä	Kyllä
5.2.2016 18:08:54	Kyllä	Kyllä
5.2.2016 22:40:44	Kyllä	Kyllä

Kuinka hyvin koit ymmärtäväsi pelin pelimekaniikat ja kontrollit?	Arvioi keskittymisesi peliin	Kuinka haastavalta peli tuntui?
2	3	2
3	4	5
2	3	4
3	4	4
3	5	4

Tunsitko hallitsevasi pelin tapahtumia?	Koitko pelin ulkonäön haitalliseksi tai hyödylliseksi pelin pelaamisessa?	Koitko ajantajun hämärtymistä pelin aikana?
Ei	3	Ei
Ei	4	Kyllä
Kyllä	4	Ei
Kyllä	3	Ei
Kyllä	5	Ei

Tarkistamatta aikaa kellosta, arvioi kuinka monta minuuttia käytit pelin pelaamiseen.	Tarkista aika, kuinka monta minuuttia käytit pelin pelaamiseen?	Päätyikö peli voittoon vai tappioon? Jos pelasit useamman kerran, syötä viimeisimmän pelin tulos
Alle 5 minuuttia	Alle 5 minuuttia	Tappio
5 - 10	10 - 15	Voitto
10 - 15	10 - 15	Voitto
Alle 5 minuuttia	Alle 5 minuuttia	Voitto
5 - 10	5 - 10	Voitto

Syötä tähän saamasi pistemäärä pelin päätyttyä. Korkein saavuttamasi pistemäärä näkyy pelin päävalikossa	Kuinka palkitsevalta pelin pelaaminen tuntui?	Koitko halua pelata peliä uudestaan?
165	3	Kyllä
1800	4	Kyllä
365	3	Kyllä
1835	2	Ei
1290	3	Ei

Kommentoi vapaasti demoa ja pelikokemustasi (ei pakollinen)
-
<p>Kritiikki heti alkuun: olisin itse kaivannut selkeää feedbackkiä interact-toiminnon käytöstä - vaikka ääni tai joku graafinen efekti olisi saanut enemmän tunnetta että hei, teit juuri jotain. Tuntui kovin ponnettomalta muuten perus-interact. Lisäksi toimintojen valinta oli aika hektistä kun piti saada ohjattua ihmislaumaa kotiin, ja tuli fiilis että vaikeusaste ja haaste tulee kontroleista, ei pelistä. Voisin myös todeta että manuaalinen unpause kentän alussa ei toimi, se jotenkin katkasee juurikin haetun flown kokonaan. Muutenkin kenttäsiirtymä on kolmen napin painalluksen päässä, ja jokaista nappia pitää metsästää erikseen. Rikkoo flown aika hyvin</p> <p>Positiivisena totean että kyllä tohon helposti upottaa vartin siellä ja vartin täällä. Parhaimmillaan juuri viime hetken rutistus ennen kun kaikki ukot on turvassa johtaa paniikinomaiseen nikunakutteluun, joka epäonnistuessakaan on palkitsevaa. Muutamalla pienellä viilauksella tästä tosiaan voisi saada sen haetun flown hyvinkin kannustettua esille! Myös vaikka grafiikka on selkeästi placeholderia, se toimii musta oikein hyvin: väri- ja muotokoodattuna hahmot tunnistaa ja erottaa toisistaan todella helposti.</p>
Alkussa olisi hyvä olla ns. tutoriaali osuus jossa käydään läpi kontroleja. Lisäksi pikanäppäimet eri toimintoihin olisivat hyvät - peli on kuitenkin nopeatempoinen.
-
Pikanäppäimet olisi aika pakolliset, sen verran nopeasti syö zombit aivot siviileiltä.