

Sasu Kemppainen

Pelisuunnittelu pienessä peliprojektissa

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Tradenomi
Tietojenkäsittelyn KO
25.5.2011



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittely
Tekijä(t) Sasu Kemppainen	
Työn nimi Pelisuunnittelu pienessä peliprojektissa	
Vaihtoehtoiset ammattiopinnot	Ohjaaja(t) Janne Koponen
	Toimeksiantaja -
Aika Kevät 2011	Sivumäärä ja liitteet 55+1
<p>Opinnäytetyö dokumentoi pientä Kajaanin AMK opiskelijoiden toteuttamaa peliprojektia ja käy läpi teoriataustaa. Kaikilla peliharrastajilla on todennäköisesti jokin käsitys siitä, mitä pelisuunnittelu tarkoittaa, mutta nämä käsitykset tuntuvat olevan ristiriitaisia ja muuttuvan henkilöltä toiselle. Kyseessä ei ole vain satunnaisten ideoiden keksimistä tai tarinan kirjoittamista, vaan pelisuunnittelu koostuu kokonaisuudessaan useasta osa-alueesta. Kaikki niistä ovat tarpeeksi laajoja kokonaista opinnäytetyötä varten. Tarkoituksena on auttaa lukijoita ymmärtämään pelisuunnittelua ja antaa näkökulmaa peliprojektissa työskentelyyn.</p> <p>Käytännön osuuteen kuului työskentely usean kuukauden ajan pienimuotoisen peliprojektin pelisuunnittelijana. Projektin aikana täytyi työskennellä jokaisen pelisuunnittelun osa-alueen parissa ja opiskella erilaisia pelisuunnittelun teorioita sekä käytännön esimerkkejä. Opinnäytetyötekstissä käydään läpi pelisuunnittelun osa-alueet yleisellä tasolla käyttäen havainnollistavia esimerkkejä muista peleistä. Teoriaosuutta voi verrata opinnäytetyöpeliprojektissa käytettyihin käytäntöihin, joista kerrotaan teoriaosuuden jälkeisessä kappaleessa.</p> <p>Lopullista peliä ei saatu valmiiksi, mutta projektin aikana opitut teoreettiset käytännöt ja valmiiksi saatu materiaali olivat vaivan arvoisia. Tekstissä pohditaan hieman projektin epäonnistumisen syitä ja sen kautta lukija voi oppia välttämään yleisiä ongelmia vastaavanlaisissa peliprojekteissa.</p>	
Kieli	Suomi
Asiasanat	pelisuunnittelu, peliprojekti
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Kajaani University of Applied Sciences	Degree Programme Business Information Technology
Author(s) Sasu Kemppainen	
Title Game Design in a small Game Project	
Optional Professional Studies	Instructor(s) Janne Koponen
	Commissioned by -
Date Spring 2011	Total Number of Pages and Appendices 55+1
<p>The thesis documents a small game project made by Kajaani UAS students and goes through its theoretical background. Every gamer probably has some idea of what game design means but these ideas seem to contradict each other and change from one person to another. It is not about just coming up with random ideas or writing stories. Game design consists of many smaller fields, each of which is large enough to write a whole thesis about. The idea here is to teach the reader to understand game design and to give some perspective for work in a game project.</p> <p>The practical part of this thesis consists of work as a game designer in a small game project for several months. The project required working with each field of game design and studying different game design theories and practical examples. The text elaborates on the fields of game design on a general level with illustrative examples from other games. The theoretical part can be compared with the practices used in the thesis game project, which are described in the chapters following the theory.</p> <p>The game was never finalized but the theory learned and the assets produced made the project worth the effort. The text contemplates a little on the reasons for this failure and the reader may learn to avoid common problems in similar game projects.</p>	
Language of Thesis	Finnish
Keywords	game design, game project
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

SISÄLLYS

1 JOHDANTO	1
2 PROJEKTIN VALMISTELU JA VISIOINTI	2
2.1 Työnjako	3
2.2 Rajaukset	3
2.3 Projektin alku	4
3 PELISUUNNITTELU YLEISESTI	8
3.1 Maailman suunnittelu	8
3.2 Mekaniikan suunnittelu	9
3.3 Sisällön suunnittelu	13
3.4 Pelikäsikirjoitus	15
3.5 Kenttäsuunnittelu	17
3.6 Käyttöliittymäsuunnittelu	22
4 PELISUUNNITTELU PROJEKTISSA	25
4.1 Maailman suunnittelu	26
4.2 Mekaniikan suunnittelu	28
4.3 Sisällön suunnittelu	31
4.4 Pelikäsikirjoitus	36
4.5 Kenttäsuunnittelu	38
4.6 Käyttöliittymäsuunnittelu	40
5 YHTEENVETO JA POHDINTA	42
LÄHTEET	46
LIITTEET	

SYMBOLILUETTELO

AI	Tietokoneohjelman tekoäly, lyhenne sanoista artificial intelligence. (McCarthy, J. 2007.)
Arcade	Tässä tarkoitetaan arcadetyylisiä pelejä, eli nopeatempoisia toimintapelejä, jotka vaativat hyviä silmän ja käden koordinaatiokykyjä. (Webopedia 2011.)
Blogi	Internetissä oleva julkinen päiväkirja-tyylinen sivusto. Tulee sanoista web log. (Enzer, M. 2008.)
Brainstorming	Tekniikka, jossa ryhmä ihmisiä heittelee ideoita toisilleen, jotta saavutetaan suuri määrä ratkaisuja tietyn ongelman ratkaisemiseen. Ideoita kehitetään myöhemmin lisää. Tunnetaan myös nimellä aivoriihi. (jpb.com 2006.)
High Concept Document	Dokumentti, jossa pelin idea ja tärkeimmät peruspiirteet on lyhyesti lueteltu. Voidaan käyttää myyntitarkoituksiin tai muistilistana. (Adams, E. 2011.)
HUD	Kokoelma pelissä ruudulla näkyviä mittareita ja muita elementtejä, joiden tarkoitus on kertoa pelaajalle muun muassa pelihahmon sen hetkinen tila. Nimi tulee sanoista Heads-Up Display. (Wilson, G. 2006.)
Kasuaalipeli	Satunnaispelaajille ystävällinen, intuitiivinen ja helppo peli. (Welch, J. 2008.)
Kontekstitarkkuus	Ominaisuus, joka muuttuu sen mukaan, mitä käyttäjä tekee milläkin hetkellä. (Webopedia 2011.)
Kyberpunk	Scifin alatyyppejä, teknologian kehitykseen keskittyvä synkkä kuvaus lähitulevaisuudesta. (Kheper 2003.)

Käyttöliittymä	Käyttäjän ja ohjelmiston välillä oleva rajapinta, jolla käyttäjä vaikuttaa ohjelmistoon ja saa tietoa siitä. (Webopedia 2011.)
Level Flow Diagram	Diagrammi, jossa esitellään suunnitellun kentän pääpiirteet ja tapahtumat. (Gard, T. 2010.)
Lisäelämä	Lisäelämät antavat pelaajalle toisen mahdollisuuden pelissä epäonnistumisen jälkeen. (Giant Bomb 2011.)
Metroidvania	Metroid -ja Castlevania-pelisarjojen mukaan nimetty pelityyli, jossa pelaajan on tarkoitus seikkailla sokke-loisessa ympäristössä ja hankkia tavaroita tai kykyjä, joiden avulla hän pystyy etenemään ympäristön esteiden läpi. Pelaaja voi muuten seikkailla suhteellisen vapaasti tällä suurella alueella. (Giant Bomb 2011.)
Mockup	Eräänlainen malliversio testaamista, tutkimista tai esittelyä varten. Esimerkiksi pelistä voidaan piirtää kuvia, jotka auttavat hahmottamaan, miltä sen tulisi lopulta näyttää. (Merriam-Webster 2011.)
Narratiivi	Kertomus, jolla on alku, keskikohta ja loppu. (Kookas 2011.)
NES	Nintendo Entertainment System, Nintendon ensimmäinen pelikonsoli. (Open Dictionary 2011.)
Pelimoottori	Ohjelmistojärjestelmä, joka sisältää uudelleen käytettäviä komponentteja, kuten grafiikoiden piirron ja törmäystarkastuksen. (Game Career Guide 2008.)
Pick Up	Pick upit ovat pelien sisäisiä resursseja, kuten esimerkiksi raha ja energialisät. (Björk, S. & Holopainen, J. 2005, 87.)

Pitch	Esimerkiksi nopea peli-idean esittely, tarkoituksena ”myydä” se kuulijoille. (Make It Big in Games 2011.)
Power Up	Kerättävä pelinsisäinen esine, joista pelaaja saa hahmolleen jollain tavalla lisää voimaa, yleensä väliaikaisesti. (Björk, S. & Holopainen, J. 2005, 88-89.)
Pre-Alpha Testaus	Keskeneräisen pelin testaamista. Pre-Alpha- vaiheessa pelissä ei ole vielä kaikkia suunniteltuja ominaisuuksia. (Wiktionary 2009.)
Proseduraalinen generointi	Ohjelmallinen sisällönluomismenetelmä, esimerkiksi automaattinen pelikentän luominen ennalta syötettyjen arvojen perusteella. (Procedural Content Generation Wiki 2011.)
Quick Time Event	Tapa tehdä välivideoista hieman interaktiivisempia. Välivideon aikana saattaa esimerkiksi ruudulla välähättää erilaisia näppäinyhdistelmiä, joita pelaajan pitää painaa estääkseen hahmonsa kuoleman välivideoissa. (Giant Bomb 2011.)
Roguelike	Pelityyppi, jonka pelit ovat suurelta osalta proseduraalisesti generoituja ja epälineaarisia seikkailuja. (Temple of the Roguelike 2007.)
RPG	Pelityyppi, jonka monimuotoisissa peleissä ovat yleensä etualalla tarina, pelattavuus ja strategia. Suuri osa näistä peleistä sisältää erilaisia hahmonkehitysmekaniikkoja. (Barton, M. 2007.)
Sprite	Kaksiulotteinen kuva, joka voi olla esimerkiksi yksi kuva hahmon animaatiota 2D-pelissä. (msdn 2011.)

Steampunk	Scifin alatyyppeä, historialliseen ympäristöön sijoittuva kuvaus vaihtoehtoisen höyry - ja mekaniikkateknologian menestyksestä elektronisen sijaan. (Science Fiction Citations 2008.)
Tasohyppely	Pelityyppi, jossa pelaaja ohjaa hahmoa, joka hyppii ja kiipeilee eri tasojen välillä. (Oxford Advanced Learner's Dictionary 2011.)
Versionhallinta	Jokin menetelmä (usein ohjelmiston hoitama), jonka avulla säilytetään projektin tiedostoja eri, jotta voidaan esimerkiksi palata takaisin aikaisempaan versioon tarvittaessa. (BetterExplained 2007.)
Välivideo	Esimerkiksi videopätkä, joka keskeyttää toiminnan pelissä ja muun muassa kertoo tarinaa. (Gilbert, R. 2007.)
Wiki	Palvelinohjelmisto, jonka avulla käyttäjät voivat vapaasti muokata Internet sivua millä tahansa Internet selaimella. (wiki.org 2002.)

1 JOHDANTO

Tämän opinnäytetyöprojektin pohjana oli halu työskennellä osana pelikehitystiimiä ja saada yhteistyön tuloksena aikaan toimiva peli, josta tekijät voisivat olla ylpeitä. Perusajatus oli saada aikaan peliprojekti ja projektin alkaessa ideoida sekä suunnitella tuotettava peli. Projektille varattiin aikaa kolme kuukautta, mutta se ei lopulta riittänyt pelin toteuttamiseen. Tiimi joutui työskentelemään monta kuukautta yliaikaa, eikä lopputulokseksi siltikään saatu toimivaa peliä. Tähän olivat syynä erinäiset pelikehitysprosessin ongelmat, joita käydään läpi enimmäkseen tämän tekstin pohdintavaiheessa.

Projekti ei tästä huolimatta kuollut kokonaan. Se on yhä elossa ja paljon materiaalia saatiin kehitysvaiheessa valmiiksi. Kaikkein pisimmälle vietin pelin suunnittelupuoli.

Tämän opinnäytetyön kirjoittaja oli osa neljän hengen tiimiä ja toimi pääasiallisena pelisuunnittelijana. Kaikilla tiimin jäsenillä oli päätöidensä lisäksi muita, pienempiä vastuualueita, mutta pelisuunnittelussa päästiin pitkälle erikoistetun pelisuunnittelijan roolin ansiosta.

Pelisuunnitteludokumentti saatiin lähes valmiiseen kuntoon, joten projektin voi sen pohjalta helposti käynnistää uudelleen milloin vain. Dokumentti oli tarkoitettu koko projektin ajan muokattavaksi ja testauksen mukaan muuttuvaksi. Se jäi tämän takia hieman keskeneräiseksi, koska myöhempisiin testausvaiheisiin ei projektissa päästy. Tähän vaiheeseen jäänyt pelisuunnitteludokumentti antaa kuitenkin mahdollisuuden tässä tekstissä keskittyä pienempään osaan projektin pelisuunnittelua.

Tämän opinnäytetyön kirjallinen osuus käsittelee pelisuunnittelua tässä projektissa. Tarkoitus on käydä läpi projektin vaiheita ja selittää, miksi asiat päädyttiin tekemään niin kuin ne tehtiin. Pelisuunnittelu on monimutkainen ala, ja sen kaikkia puolia ei voi edes pintapuolisesti käydä läpi pienessä opinnäytetyössä. Suuri osa tekstistä on pelisuunnittelun tarkastelua melko pinnallisesta näkökulmasta, jotta yleisesti hyväksytyjä menetelmiä voidaan verrata tämän projektin käytäntöihin ja selittää sen pelisuunnittelullisia ratkaisuja.

2 PROJEKTIN VALMISTELU JA VISIOINTI

Ensimmäinen vaihe peliprojektissa oli kehitystiimin kokoaminen. Samaan aikaan valmistuivista pelilinjan opiskelijoista pystyi helposti kokoamaan tiimin, sillä vaikka opiskelijoita ei enää ollut kovin montaa, olivat jäljellä olevien taidot tarvittavan monipuolisia peliprojektia varten. Ainoastaan projektin johtamisesta ja tuottamisesta ei kenelläkään ollut erityisen hyvää koulutusta. Näiltä osin projekti oli kaikkein eniten oppimiskokemus.

Voidaan ajatella, että jonkinlaisen pelin voi minimissään saada aikaan yhden ohjelmoijan voimin. Pelihistorian alkutaipaleella kehittäjiä usein olikin vain muutama ja ohjelmoijat hoitivat muitakin töitä ohjelmoinnin lisäksi. Erityisesti pelisuunnittelu kuului ohjelmoijien sivutoihin. Tämä voi olla nykyäänkin pienien pelien tuotannossa. Opinnäytetyöprojektissa kuitenkin pyrittiin erottamaan pelituotannon elementit toisistaan ja erikoistaa tiimin jäsenet omiin osa-alueisiinsa. (Chavis, J. 2011)

Lopullinen tiimi koostui neljästä jäsenestä: pelisuunnittelijasta, graafikosta ja kahdesta ohjelmoijasta. Pelisuunnittelusta vastasi tämän opinnäytetyön kirjoittaja Sasu Kemppainen. Sen lisäksi hänen vastuualueeseensa kuuluivat erilaiset tuotannolliset seikat ja ryhmän johtaminen. Graafikkona työskenteli Markku Wiik, jolla oli kokemusta graafisesta suunnittelusta ja toteutuksesta niin työmaailmassa kuin opiskelunkin kautta. Hänelle on kehittynyt selviä vivahteita omanlaisesta piirrostyylisestä, mikä vaikutti koko peli-idean valintaan. Pelin kahtena ohjelmoijana toimivat Toni Räsänen ja Otto Hietala, jotka molemmat keskittyivät omiin osa-alueisiinsa tiiviissä yhteistyössä. Heidän ohjelmointiharrastuksensa on alkanut jo ennen ammattikorkeakouluopintoja ja portfolioon kuuluu molemmilla muutamia toimivia pelejä.

Peliprojektin suorittaminen päätettiin aloittaa heinäkuussa 2010 Kajaanin Ammattikorkeakoulun Pelilaboratoriossa. Tilasta varattiin neljä läheistä työpistettä noin 3-4 kuukauden ajaksi. Projektin alustavaa aikataulua ajateltiin hyvin venyvänä, mutta sen lopullinen kesto yllätti kaikki tiimin jäsenet. Julkaisemiskuntoista versiota ei oltu vielä edes tammikuuhun 2011 mennessä saatu valmiiksi. Tiimille oli kuitenkin tilaa Pelilaboratorion eri osissa lähes koko tarvittavalle ajalle.

Ennen projektin alkamista, tuotettiin vain hyvin karkea aikataulu opinnäytetöiden etenemisistä. Tarkemmat suunnitelmat itse pelistä ja tuotannollisista seikoista oli tarkoitus selvittää vasta projektin ensimmäisenä päivänä, jotta se muistuttaisi enemmän ”aitoa” peliprojektia.

2.1 Työnjako

Kaikkien tiimin jäsenien oli tarkoitus kirjoittaa peliprojektista oma opinnäytetyö, liittyen kyseisen henkilön projektissa tekemään työhön. Opinnäytetöiden aiheet valittiin ennen projektin alkamista ja valittujen osa-alueiden oli tarkoitus olla kyseisten tiimin jäsenien päätöitä projektin aikana.

Käytännönvaiheessa keskityttiin tämän opinnäytetyön osalta luonnollisesti projektin pelisuunnitteluun, mutta muitakin vastuualueita oli, kuten kaikilla muillakin tiimin jäsenillä. Nämä vastuualueet olivat lähinnä projektin johtamiseen ja ylläpitämiseen liittyviä, kuten esimerkiksi aikataulut. Ylimääräiset vastuualueet olivat tärkeitä projektin kannalta, mutta ne eivät olleet suuri osa itse opinnäytetyötä. (Kemppainen, S. 2010)

Graafikon työnä oli luoda kaikki peliin tarvittavat käyttäen Autodeskin Softimage-mallinnustyökalua ja piirto-ohjelmia. Pelin 2D spritet oli tarkoitus piirtää tehtyjen 3D-mallien pohjalta. Näin helpotettaisiin hahmojen animointia. (Wiik, M. 2010.)

Projektin kahdelle ohjelmoijalle asetettiin omat osa-alueet, joita voitaisiin tarpeen mukaan välillä vaihdella keskenään. Otto Hietalan vastuu-alueeseen kuului pelimoottorin ohjelmointi ja Toni Räsänen tehtäväksi jäivät proseduraaliseen generointiin liittyvät asiat. Pelattavuuden ohjelmointi kuului molemmille, riippuen siitä, miten heidän päävastuualueensa edistyivät. (Hietala, O. 2010; Räsänen, T. 2010.)

2.2 Rajaukset

Tämä opinnäytetyö on rajattu käsittelemään pelisuunnittelua ja pelikehitystiimin johtamista peliprojektissa. Tekstissä käydään läpi pelisuunnittelullisia asioita yleisellä tasolla, jotta niiden pohjalta voidaan antaa yleiskuva pelisuunnittelun perusteista ja selittää projektissa käytettyjä valintoja. Opinnäytetyöprojektiin suoraan liittymättömiä asioita käydään myös välillä läpi,

mutta ne ovat tarpeellisia pelisuunnittelun ymmärtämiseen. On tärkeää ymmärtää pelisuunnittelun teoriaa edes yleisellä tasolla, jotta projektissa tehdyt ratkaisut voivat valottua.

Opinnäytetyötekstin painotus on enimmäkseen pelisuunnittelussa. Projektin johtamiseen tässä tapauksessa lasketaan kaikki ylläpidolliset seikat, kuten esimerkiksi suunnittelupalaverit ja työtehtävien jakaminen. Johtamisen filosofiaa tai mitään vastaavaa aihetta ei tässä käsitellä.

Peli-idean kehittämiseen palataan tekstissä tarkemmin myöhemmin, mutta kaikkein ilmeisimmät rajoitukset pelisuunnitteluun olivat selvillä jo kauan ennen projektin virallista aloittamista. Erikoistetun graafikon olemassaolo pakotti ideoimaan pelin, jossa käytetään grafiikoita. Pelistä täytyi myös suunnitella graafisesti näyttävä, jotta graafikko saisi tarpeeksi materiaalia omaan opinnäytetyötekstiinsä. Tekstipohjaisen pelin tai graafisesti yksinkertaisen Roguelike-pelin tuottaminen ei olisi käynyt päinsä. Muidenkaan opinnäytetyösuunnitelmat eivät tukeneet vastaavanlaisia pelityyppejä.

Tiimin toinen ohjelmoija, Toni Räsänen, oli valinnut aiheekseen proseduraalisen generoinnin, mikä lopulta vaikutti kaikkein eniten pelisuunnitteluun. Tiimin jäsenillä oli suuri halu tehdä pelistä erikoinen ja moderni, joten proseduraalisen sisällöngeneroinnin käyttäminen perusideana toimi loistavasti.

Mahdollisesti rajoittavin elementti oli Otto Hietalan opinnäytetyö, johon kuului pelimoottorin rakentaminen SDL:ää käyttäen. Monimutkaisen pelimoottorin kokoaminen lopulta ideoitua peliä varten osoittautui myöhemmin vaikeaksi ja projektia hidastavaksi. Oma, pelin tarpeita varten määritelty pelimoottori sisältää kaikki peliä varten tarvittavat ominaisuudet, mutta valmista moottoria käyttäen olisi voitu säästää paljon aikaa pelin kehittämisessä. Tällöin olisi myös voitu keskittää yksi ohjelmoija kokonaan pelattavuuden ohjelmointiin.

2.3 Projektin alku

Projektin aloitettiin 5.7.2010 niin sanotusti puhtaalta pöydältä. Ainoastaan aiemmin asetetut rajaukset oli sovittu pitäviksi.

Projektin aloitettiin vapaamuotoisella alkupalaverilla, jossa oli tarkoitus sopia tarkemmin projektin käytänteet. Aiemmin sovitut vastualueet käytiin läpi ja pelilaboratoriosta varattiin so-

pivat lähekkäiset paikat koko tiimille. Aikataulu sovittiin aluksi venyväksi, sillä heinäkuun aikana oli mahdotonta hankkia avaimia koululle. Sisälle pääsi vain, jos joku muu oli aiemmin saapunut paikalle. Koululle saavuttiin varmuuden vuoksi kello kymmenen aikoihin ja töitä tehtiin niin pitkään kuin työtehoa riitti. Tällainen epätasallinen aikataulu ei välttämättä ollut kaikkein tehokkain mahdollinen, mutta se jäi päälle koko projektin ajaksi.

Ensimmäiseksi tiimi järjesti projektille versionhallinnan. Kun useampi ohjelmoija työskentelee saman koodin parissa, on jonkinlainen versionhallintaohjelmisto välttämätön. Tiimi päätyi yhteisestä sopimuksesta luomaan tilit SourceForge.net-sivulla. SourceForge tarjoaa mahdollisuudet useisiin ohjelmistokehitystyöhön tarvittaviin ominaisuuksiin, kuten juuri versionhallintaan. Tiimi päätti myös käyttää SourceForgen tarjoamaa tilaa pyörittääkseen projektiblogia (<http://sourceforge.net/apps/wordpress/vohweli/>), joka auttoi vähän kartoittamaan projektin etenemistä, mutta oli olemassa lähinnä hovin vuoksi. Rekisteröityessä vaadittiin tietysti myös projektin nimi, mutta koska itse pelistä ei ollut olemassa edes pohjaideaa, päätti kehitystiimi yhteismielin nimetä tiimensä Vohweliksi ja käyttää sitä myös projektin nimenä. Tässä opinnäytetyössä tiimiin voidaan viitata tästä eteenpäin nimellä Vohweli. (SourceForge 2011.)

Aluksi pyrittiin hankkimaan pelilaboratorion koneille kaikki todennäköisesti tarvittavat ohjelmistot. Ohjelmointia, pelisuunnittelua ja projektinhallintaa varten tarvittavat ohjelmat löytyivät jo koneilta, mutta grafiikan tuottamiseen tarvittavat ohjelmistot puuttuivat. Tiimin graafikko oli tottunut käyttämään Softimage-mallinnustyökaluja, eikä niitä saatu asennettua moneen viikkoon, sillä lupa uusien ohjelmistojen asentamiseen oli vaikea saada. Tämä loppujen lopuksi hidastutti hieman grafiikan tuottamista.

Alkuvalmistelut saatiin hoidettua suhteellisen nopeasti, joten jo ensimmäisenä päivänä päästiin ideoimaan lopullista peliä. Ideointi aloitettiin kevyellä brainstorming-sessiolla. Tarkoituksena oli aluksi vain kartoittaa tiimin jäsenten ajatuksia mahdollisesta peli-ideasta ja etsiä kaikkia kiinnostavia elementtejä.

Kehitystiimi sovelsi yleisiä brainstorming-käytäntöjä sopivaksi peliprojektia varten. Perinteisiä, tehokkaita todettuja käytäntöjä pyrittiin seuraamaan niin tarkasti kuin oli käytännöllistä seurata. Brainstorming-sessiossa on erittäin tärkeää saada määriteltyä kehitettävä ongelma hyvin, jotta sessio pysyy tiiviinä ja jäsenet tietävät, mihin pyritään. Hyvin suunniteltu ja määritelty kehitysongelma tuottaa parhaat ideat sen ratkomiseen. Vohwelin ensimmäisessä

brainstorming-sessiossa kehitysongelma oli määritelty yksinkertaisesti: ”Millaisia elementtejä haluaisit kehittää peliin?” Ongelma oli tietoisesti mietitty niin, ettei se rajoittaisi tiimin jäsenien mielikuvitusta ja että saataisiin aluksi yksinkertaisesti kartoitus tiimin ajatuksista tulevaa peliä kohtaan. (jpb.com 2006.)

Aikarajan käyttöönotto brainstorming-sessiossa on suositeltavaa. Toisaalta on mahdollista rajoittaa sessiota myös asettamalla raja heiteltyjen ideoiden määrään. Vohwelin brainstorming-sessioissa rajoitettiin pituus useimmiten ideoiden määrän mukaan.(jpb.com 2006.)

Alustavan session tuloksena saadut olennaisimmat ja mielenkiintoisimmat ajatukset tulevan pelin elementeistä olivat seuraavanlaisia:

- rikkinäinen robotti
- tasoloikka
- jäävoimat
- maailmanlopun keskellä
- satunnaisgeneroidut hahmot
- tapahtumat juuri ennen maailmanloppua
- kyberpunk
- hiiviskely
- RPG-hahmokehitys
- futuristinen väkivaltainen jalkapallo
- steampunk
- 1800-luvun vampyyritarina
- merirosvoteema
- AMK-manager
- cowboy-teema
- proseduraalisesti generoitu ”metroidvania”

Listasta voi huomata, kuinka laajalle skaalalle rajoittamattoman brainstorming-session ideat voivat sijoittua, mutta näkyvissä on myös tiettyä samanhenkisyyttä ideoiden pohjalta. Mukana on ajatuksia tarkoista, laajoihin ja ikiaikaisista peliteemoista, moderneihin suosikkeihin. (Giant Bomb 2011; Cyberpunkreview.com 2006; Moby Games 2011.)

Session jälkeen pidettiin ensimmäinen ideointipalaveri. Listattuja ideoita tarkasteltiin yksi kerrallaan ja kaikki tiimin jäsenet saivat sanoa niistä mielipiteensä. Lopuksi tiimin jäsenten annettiin pohtia palaverissa esitettyjä ajatuksia omassa rauhassaan ja myöhemmin jokainen sai esittää pitchin omasta ideastaan. Ideoista suosituimmaksi kohosi proseduraalisesti generoitu robottitasohyppely. Tässä vaiheessa pelille annettiin projektinimeksi Robodash.

Idean löydyttyä pidettiin vielä muutamia brainstorming sessiota, jotta voitaisiin kartoittaa pelin elementtejä tarkemmin. Tiimin jäsenet saivat ilmaista heitä kiinnostavia asioita peli-ideaan liittyen, jotta niistä voitaisiin mahdollisesti suunnitella pelin peruspilareita tai eräänlaisia kohokohtia.

Kun perusidea ja muutama pelillinen elementti oli selvillä, voitiin varsinainen työ aloittaa. Vielä myöhemmissäkin vaiheissa projektia pidettiin erilaisia suunnittelupalavereita pelin eri elementeistä, jotta koko tiimin ideoita voitaisiin hyödyntää.

3 PELISUUNNITTELU YLEISESTI

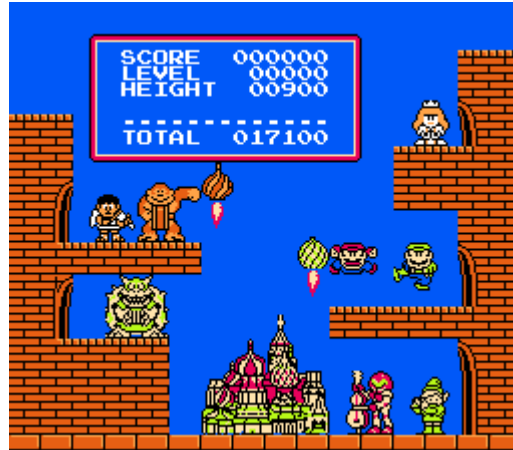
Tässä kappaleessa käydään läpi pelisuunnittelun yleisimmät osa-alueet ja kerrotaan niihin liittyvästä teoriasta yleisellä tasolla. Tätä tietoa voi käyttää vertaillakseen myöhemmässä kappaleessa kerrottuja Vohweli-tiimin käyttämiä ratkaisuja ja käytäntöjä yleisesti hyväksytyihin pelisuunnittelullisiin teorioihin.

Pelisuunnittelu on tässä jaettu kuuteen osa-alueeseen: maailman suunnitteluun, mekaniikan suunnitteluun, sisällön suunnitteluun, pelikäsikirjoitukseen, kenttäsuunnitteluun ja käyttöliittymän suunnitteluun. (Brathwaite, B. & Schrieber, I. 2008, 5.)

3.1 Maailman suunnittelu

Maailman suunnittelun voisi äkkiseltään sekoittaa kenttäsuunnitteluun, mutta ne kaksi eri pelisuunnittelun osa-alueita. Tähän osa-alueeseen kuuluu pelin taustatarinan, tapahtumapaikan ja teeman suunnittelu. Vaikka idea pelistä olisi hyvin yksinkertainen tai abstrakti, voidaan pelimaailman suunnittelulla selkeyttää muuta pelisuunnittelua ja määrittää rajoja tulevalle pelille. Teema on perusidea, jota pelin kaikkien elementtien täytyy tukea. Sen määrittäminen helpottaa peliin sisällytettävien asioiden määrittämistä: jos jokin elementti ei vahvista pelin teemaa, sen voi yleensä jättää pois. Lähes jokaisessa pelissä voidaan havaita vähintään jonkinlainen perusteema, joka sitoo sen elementit yhteen. (Brathwaite, B. & Schrieber, I. 2008, 5; Schell, J. 2008, 49.)

Vaikkei pelin olisikaan tarkoitus esimerkiksi kertoa minkäänlaista tarinaa, teeman asettaminen voi pelisuunnittelun helpottumisen lisäksi luoda peliin tunnelmaa tai herättää pelaajassa haluttuja mielentiloja. Jopa näennäisen yksinkertaisissa pulmanratkontapeleissä on yleensä jokin koko pelin yhdistävä teema. Esimerkiksi tunnetun Tetris-pelin NES-versiossa on selkeästi pelin kotimaahan, Venäjään, viittaava tunnelma niin musiikillisesti kuin esteettisesti (Kuvio 1). (|tsr's NES archive 2011.)



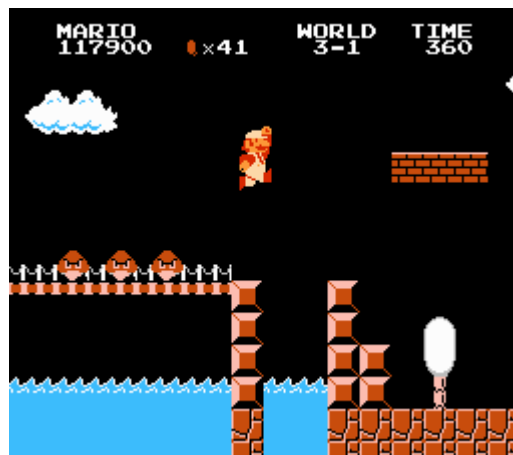
Kuvio 1: Tetris-pelin B-moodin loppukuvat (vgmuseum.com 2011.)

Pelin yhtenäisen teeman hyväksikäytön voi pelityypistä riippuen viedä pidemmällekin kuin vain grafiikkaan ja ääneen. Hyvä esimerkki tästä on koko Metal Gear Solid-pelisarja. Jokaisella pelisarjan osalla on selkeästi pelisuunnittelun alussa määrätty pääteema, joka näkyy niin pelimaailmassa kuin juonessa ja pelattavuudessakin. Pelisarjan ohjaaja Hideo Kojima on tiivistänyt pelisarjan kolmen ensimmäisen osan pääteemat lyhenteeseen MGS: Meme, Gene, Scene. Kaksi ensimmäistä osaa käsittelevät ajatusta: ”Mitä jätämme jälkeemme tuleville sukupolville?” Metal Gear Solid 1 käsittelee geenejä (Gene) ja niiden vaikutusta kun taas Metal Gear Solid 2 käsittelee ajatuksia, koskettelemattomia asioita (Meme). Kolmas osa käy läpi ympäristön ja ajan muutoksen vaikutuksia ihmisten arvoihin (Scene). Näiden lisäksi peleistä löytyy alateemoja ja koko sarjaa yhdistäviä teemoja kuten ydinaseiden vaarat ja yleinen sodan ja rauhan asioiden pohdinta. Nämä teemat näkyvät suurimmaksi osaksi käsikirjoituksessa, mutta vaikuttavat myös pelien kokonaistunnelmaan ja pelattavuuteenkin. (Konami 1998; Konami 2001; Konami 2004; konami.jp 2008.)

3.2 Mekaniikan suunnittelu

Mekaniikan suunnittelulla tarkoitetaan pelin sääntöjen ja pohjalla olevien matemaattisten järjestelmien luomista. Tämä on ainoa suunnitteluvaihe, joka on varmasti mukana jokaisen pelin suunnittelussa ja luomisvaiheessa, sillä jokaisella pelillä on jonkinlaiset säännöt. (Brathwaite, B. & Schrieber, I. 2008, 5.)

Yleisesti käytetty termi ”pelimekaniikka” tarkoittaa pohjimmiltaan samaa kuin sääntö. Kaikki pelit ovat täynnä erilaisia mekaniikkoja, joita peli ja pelaaja voivat käyttää hyväkseen muuttaakseen pelin tilaa. Esimerkiksi pelissä Super Mario Bros. yksi pelin perusmekaniikoista on hyppy (Kuvio 2.). Mekaniikan voisi kuvailla tarkalleen: ”Kun pelaaja painaa nappia 'A', Mario (tai Luigi) hyppää.” Kyseinen peli on itse asiassa pohjimmiltaan rakennettu juuri tämän mekaniikan ja sen ominaisuuksien ympärille. (Brathwaite, B. & Schrieber, I. 2008, 28; Nintendo 1985.)

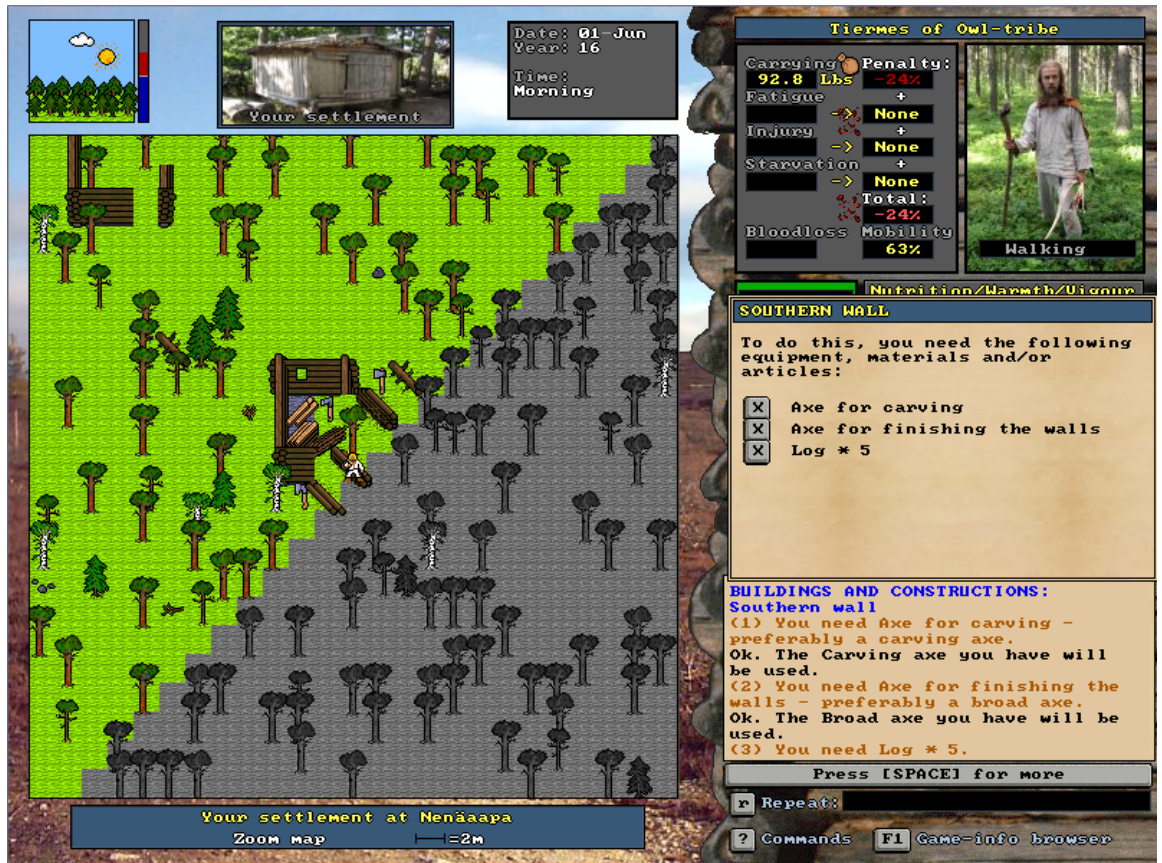


Kuvio 2: Mario hyppää vesiesteen yli (vgmuseum.com 2011.)

Pelimekaniikat voidaan määritellä erilaisiin luokkiin. Tässä on esiteltyä muutama yleinen mekaniikkaluokka.

Voittoehdot

Peleissä täytyy olla olemassa jokin sääntö, joka kertoo milloin pelaaja on voittanut pelin. On myös tärkeää antaa pelaajan tiedostaa kyseiset voiton ehdot, sillä muuten pelaaja voi tuntea itsensä eksyneeksi ja toimeettomaksi. Jonkinlainen maali on hyvä olla aina näkyvässä, vaikka pelaajalle ilmoitetut voiton ehdot muuttuisivatkin pelin edettäessä. Joissain avoimissa peleissä ei varsinaisesti ole voittoehtoja ollenkaan. Pelaaja voi esimerkiksi vain edetä tehtävästä toiseen loputtomasti. Tästä esimerkkinä toimii suomalainen Unreal World RPG, jota pelaaja voi käyttää virtuaalimaailmanaan, eläkkeeseen hahmonsensa kautta muinaisessa Suomessa. Lopullista maalia ei ole. Pelaaja voi vain elää pelimaailmassa ja esimerkiksi rakentaa itsellensä asuin-sijoja tai kesyttää eläimiä (Kuvio 3.). Peli loppuu, kun pelaaja kuolee. (Brathwaite, B. & Schrieber, I. 2008, 29; Enormous Elk 2010.)



Kuvio 3: Pelaaja seikkailee UnReal World RPG:n maailmassa (UnReal World RPG Official Homepage 2011.)

Pelaajan toiminnat

Mahdollisesti tärkeimmät mekaniikat luetaan ”pelaajan toiminnat”-luokkaan. Sinne kuuluvat säännöt siitä, mitä pelaaja voi tehdä ja miten pelaajan toiminnat vaikuttavat pelin tilaan. Pelaajan toiminnoista käytetään joskus termiä ”verbit”. Joissain vanhanaikaisissa peleissä pelaajalle käytettävissä olevat toiminnat olivat itse asiassa pelaajalle merkattu verbilistojen avulla. Esimerkiksi vanha seikkailupeli, Maniac Mansion, antoi pelaajalle mahdollisuuden ohjastaa hahmoja klikkaamalla ruudun alareunassa olevaa verbilistaa ja sitten jotain esinettä tai asiaa pelimaailmassa. Näin koostettiin sanoja kuten ”open door” ja ratkottiin pelin tarjoamia ongelmia (Kuvio 4). Tämä mekaniikka oli kehittyneempi versio vielä vanhempien tekstipohjaisen seikkailupelien käyttöliittymistä. (Brathwaite, B. & Schrieber, I. 2008, 30; Lucasfilm Games 1987.)



Kuvio 4: Pelaaja ohjastaa hahmoaan ottamaan mukaansa maalipurkin Maniac Mansionin parannellussa versiossa (The Maniac Mansion Fan Site 2007.)

Pelaamisen eteneminen

Pelaamisen etenemissäännöissä määritetään muun muassa onko peli reaaliaikainen vai vuoropohjainen ja missä järjestyksessä pelaajat saavat pelata. Hyviä esimerkkejä näistä mekaniikoista löytyykin erilaisista vuoropohjaisista peleistä. Pelissä Heroes of Might and Magic III pelaajat voivat hyökätä joukoillaan toisten pelaajien kimppuun, mikä laukaisee lautapelimäisen vuoropohjaisen taisteluvaiheen. Tässä pelivaiheessa erilaiset säännöt määrittävät esimerkiksi sen, kuka aloittaa ensiksi tai kuinka montaa joukkoa pelaaja voi liikuttaa ennen kuin on vastustajan vuoro liikuttaa omia joukkojaan. (Brathwaite, B. & Schrieber, I. 2008, 29; New World Computing 1999.)

Reaaliaikaisissa peleissä nämä säännöt voivat taas ratkaista muun muassa sen, mitä tapahtuu kun kaksi pelaajaa yrittää tehdä jotain yhtäaikaan. Street Fighter II pelin kaikissa versioissa pelaaja voi kamppailun aikana ottaa otteen toisesta pelaajasta ja heittää hänet. Tämä vahingoittaa vastustajaa. Jos kaksi hahmoa yrittää heittoa täsmälleen yhtä aikaa heitettävässä tilassa olevaan vastustajaan, arvotaan heittäjä ja heitettävä sattumanvaraisesti. (Brathwaite, B. & Schrieber, I. 2008, 29; Sonichurricane.com 2011.)

3.3 Sisällön suunnittelu

Tähän osa-alueeseen kuuluvat hahmojen, esineiden, tehtävien ja haasteiden suunnittelu. Nämä asiat ovat kuin lihaa gameplay-luiden päällä ja niiden suunnittelussa täytyy ottaa huomioon jo pohjalle rakennetun tai suunnitellun pelattavuuden ominaisuudet ja mahdollisuudet. (Brathwaite, B. & Schrieber, I. 2008, 5.)

Hahmot

Hahmot peleissä ovat tärkeä osa tarinankerrontaa. Hyvä tarina tarvitsee monesti muistettavia hahmoja, kuten voidaan huomata muidenkin medioiden kertomissa tarinoissa. Esimerkiksi yleisiä elokuvien ja kirjallisuuden päähahmoja voidaan verrata stereotyyppisiin pelipäähahmoihin. Näin voidaan saada jonkinlainen käsitys siitä, mitä pelihahmon pohjimmiltaan tarvitsee olla, ja miten ne eroavat muiden medioiden hahmoista. (Schell, J. 2008, 310)

Tarinankerronnan tyylit ja vaikutteet tietenkin leviävät mediasta toiseen sen verran kuin voivat, mutta kaikilla medioilla on omat ominaisuutensa. Esimerkiksi kirjat kertovat paljon hahmojen perimmäisistä tunteista ja ajatuksista kertojan kautta. Elokuviissa taas ei yleensä kuulla hahmojen ajatuksia, vaan heidän konfliktinsa, niin henkiset kuin fyysisetkin, näytetään visuaalisesti. Pelien päähenkilöitä ohjaa pelaaja, ja interaktiivisen luonteensa takia, pelaajan ja päähahmon kohtaamat ongelmat ovat yleisimmin fyysisiä. Pelit kuitenkin yleensä lainaavat menetelmiä vahvojen hahmojen kuvaamiseen muista medioista, ja vaikka pelaaja voi pitää peliä sarjana erilaisia fyysisiä haasteita, on pelihahmojen välille useasti rakennettu enemmän tai vähemmän monimutkaiset dramaattiset suhteet. (Schell, J. 2008, 311)

Avatar on ainoastaan peleissä esiintyvä hahmotyyppi. Tällä käsitteellä tarkoitetaan pelaajan ruumiillistumaa pelissä. Kaikki pelattavat hahmot ovat tietenkin jollain tasolla pelaajan avatareja, mutta tästä on olemassa eri tasoja. Joskus pelaaja saa pelin alussa luoda hahmonsa ulkonäköä myöten ja hahmoa rakennetaan täysin pelaajan valintojen kautta. Se on tyhjä taulu, joka ei ole mitään ilman pelaajan vaikutusta. Fallout-sarjan vapaan maailman pelit antavat pelaajan luoda itselleen hahmon, jonka kautta hän voi vaikuttaa pelimaailmaan, ja jonka persoonallisuus perustuu täysin pelaajan ajatuksiin ja toimintoihin. Toisessa päässä skaalaa ovat hahmot kuten Grim Fandangon Manuel Calavera. Manny on täyteläinen hahmo, jonka taustatarina, luonne, ulkonäkö ja kaikki dramaattiset piirteet ovat valmiiksi kirjoitettu. Pelaaja ot-

taa silloin osakseen Mannyn roolin ja ohjaa häntä ratkomaan ongelmia. Manny ei suostu tekemään aivan mitä tahansa ja pelaaja voi tunteakin olevansa pieni demoni päähenkilön pään sisällä vaikuttamassa hänen toimintoihinsa pinnallisella tasolla. Eräs välimuoto tämän tyyppisistä hahmoista on Half Lifen Gordon Freeman. Hänellä on valmiiksi määritetty ulkonäkö, nimi ja taustatarina. Hän ei ole suoraan pelaaja projektoituna pelimaailmaan, mutta pelin ensimmäisen persoonan kuvakulma, Gordonin täydellinen puhumattomuus ja persoonattomuus tekevät pelaajasta Gordon Freemanin. Hahmon persoona rakentuu kokonaan pelaajan pään sisällä. Sivuhahmot puhuttelevat Gordonia, mutteivät saa koskaan vastausta. Pelaajan ja hänen persoonallisuutensa välille ei näin tule ristiriitaa. (Schell, J. 2008, 312; Lucasarts 1998; Black Isle Studios 1997; Valve Corporation 1998.)

Esineet

Tässä esineillä tarkoitetaan pelin sisällä kerättäviä ja käytettäviä esineitä, aseita tai mitä tahansa keräilytavaroita. Kaksi olennaista peleissä käytettävää esinetyyppiä ovat niin sanotut pick upit ja power upit. (Björk, S. & Holopainen, J. 2005, 87-88.)

Pick upit ovat pohjimmiltaan resursseja. Ne voivat olla esimerkiksi kerättäviä kolikoita, joiden avulla pelaaja voi ansaita lisäelämiä, tai yksinkertaisesti energialisiä. Monissa Mario-sarjan peleissä pelaaja ansaitsee lisäelämiä jokaisesta sadasta keräämästään kolikosta ja Mega Man-sarjan viholliset voivat jättää jälkeensä energiakapseleita, jotka lisäävät pelaajan energiaa. Pick upit tuovat paljon monipuolisuutta pelisuunnitteluun. Niiden luonteesta riippuen niillä voidaan tasapainottaa peliä, luoda ansoja ja haasteita tai ohjata pelaajaa pelisuunnittelijan haluamiin paikkoihin. Pick upien kerääminen on yleensä hyvä alitavoite pelaajalle ja se voi pakottaa häntä miettimään strategioitansa. Kannattaako ottaa riskiä jonkin lisäenergian takia vai jatketaanko matkaa (Kuvio5.)? Kentän tavoitteeksi voi myös jollakin tavalla asettaa erilaisten pick upien keräämisen ja muuttaa näin hieman pelin luonnetta. (Björk, S. & Holopainen, J. 2005, 87; Nintendo 1985; Nintendo 1990; Nintendo 2009; Capcom 1988; Capcom 1990 ; Capcom 2008.)



Kuvio 5: Pelaaja joutuu päättämään, onko palkkio vaivan ja riskin arvoinen (vgmuseum.com 2011.)

Power upit taas ovat kerättäviä esineitä, joista pelaaja saa hahmolleen jollain tavalla lisää voimaa, yleensä väliaikaisesti. Esimerkki hyvin perinteisestä power upista on Pac-Manin voimapilleri. Kun Pac-Man syö pillerin, pystyy hän lyhyen ajan syömään häntä yleensä jahaavia kummituksia eikä vahingoitu kosketuksesta niiden kanssa. Power upit tuovat suuresti vaihtelua peliin muuttamalla hetkellisesti voimatasapainoa vastustajien ja pelaajan välillä. Koko peli voi jopa perustua power upien ympärille. (Björk, S. & Holopainen, J. 2005, 88-89.)

3.4 Pelikäsikirjoitus

Pelikäsikirjoitus on juuri sitä, miltä se kuulostaa. Siihen kuuluu pelin tarinan, dialogin ja kaiken muun tekstin kirjoittaminen. (Brathwaite, B. & Schrieber, I. 2008, 5.)

Käsikirjoittamista peleissä voidaan pitää melko irrallisena osana muuta suunnitteluprosessia. Pelattavuutta pidetään tärkeimpänä osa-alueena ja kaikki muut osa-alueet tukevat sitä. On mahdollista vain palkata peliprojektiin ulkopuolinen käsikirjoittaja, joka voi olla erikoistunut muihin tarinankerronnallisiin medioihin ja antaa hänen hoitaa pienet dialogin pätkät toiminnan välillä. Tällaista strategiaa käytetään liian usein, vaikka pelille olisi eduksi ottaa käsikirjoittajat tiiviiksi osaksi pelisuunnittelua, jotta he voivat osaltaan vaikuttaa pelin rytmiin, hahmoihin ja tunnelmaan. Muussa tapauksessa pelin ”peliosuus” ja ”tarinaosuus” joutuvat erilleen, eikä pelaaja huomaisi mitään, vaikka tarinan poistaisi kokonaan pelattavuutta häiritsemästä. Käsikirjoituksen laatu ei sinänsä myy pelejä, sillä niiden vetovoima koostuu suurimmalle osalle ihmisistä muistakin seikoista kuin niiden tarinankerronnallisesta potentiaalista. Tämän

takia erillisiä käsikirjoittajia voi olla vaikea oikeuttaa taloudellisista syistä. (McDevitt, D. 2010.) (The Escapist 2010.)

Useasti pienemmissä peleissä ei käytetä ollenkaan erillisiä käsikirjoittajia, vaan vaikkapa pääsuunnittelija hoitaa sen osa-alueen. Käsikirjoittaminen ja varsinainen pelisuunnittelu vaativat kuitenkin erilaiset pohjataidot ja hyvä pelimekaniikkojen luoja ei välttämättä osaa käsikirjoittaa sujuvaa tarinaa. Toisissa peleissä (enimmäkseen erilaisissa kasuaalipeleissä) ei varsinaisesti tarvita narratiivia ollenkaan. Jos kuitenkin pyritään luomaan tarinavetoinen peli, täytyy käsikirjoittamista pitää tasa-arvoisena muiden pelisuunnittelun osa-alueiden kanssa. Kiireisien kehitysaikataulujen takia voi olla vaikeaa yhdistää käsikirjoitusta ja muuta pelisuunnittelua, jos näihin töihin on palkattu erilliset työntekijät, mutta näitä osa-alueita ei silti saisi ainakaan kokonaan erottaa toisistaan. Lopullisesta pelistä näkee yleensä helposti, jos tarinaa on pidetty jälkijatuksena. (McDevitt, D. 2010; The Escapist 2010; Brathwaite, B. & Schrieber, I. 2008, 171.)

Joissain peleissä, joissa on vahva narratiivi, voi olla myös hyvin vähän dialogia. Esimerkiksi pelissä ICO pelaaja ohjaa hahmoaan läpi valtavan linnoituksen esteiden pitkiäkin aikoja ilman minkäänlaista keskeytystä välivideoilla tai dialogilla (Kuvio 6). Siltikin kyseisessä pelissä on vahva käsikirjoitus, sillä ympärillä pyörivä maailma ja tapahtumat ovat käsikirjoitettu tarkasti. Tunnelma vaihtelee tarinankerronnallisesti sekä tapahtumat ja ratkottavat ongelmat luovat pieniä tunnelmallisia palkkioita pelaajan etenemisen mukaan. (McDevitt, D. 2010.)



Kuvio 6: ICO:n maailma herää henkiin pelattavuuden kautta (Giant Bomb 2008.)

3.5 Kenttäsuunnittelu

Myös kenttäsuunnittelu kertoo nimenä itsestään paljon. Se on siis yksinkertaisimmillaan pelin kenttien suunnittelua, johon kuuluu karttojen piirtäminen sekä objektien ja haasteiden sijoittelu suunnitellulle kartalle. (Brathwaite, B. & Schrieber, I. 2008, 5.)

Kenttäsuunnittelu voi olla hyvin erilaista työtä riippuen suunniteltavasta pelistä. Hyvän peruskuvan kenttäsuunnittelusta saa kuitenkin ottamalla esimerkiksi perinteisen, enemmän tai vähemmän tarinaa kertovan toimintaseikkailun. Jotta pelisuunnittelija saa suunnitellun kentän idean heti selkeäksi itselleen ja muille, on hyvä luoda ideoista Level Flow Diagram (Kuvio 9.). Yksi tapa tämän diagrammin luomiseen on aluksi pohtia seuraavia elementtejä: (Gard, T. 2010.)

Motivaatio

Hahmon motivaatioiden täytyy jokaisen kentän alussa olla selkeät, jotta pelaaja ymmärtää tavoitteensa, eikä tunne päämäärättömyyttä. Pelihahmon motivaatioiden pitäisi saada muuttamaan pelaajan tavoitteiksi. Ei kuitenkaan ole tarpeeksi vain ilmoittaa: ”Hahmosi vihaa tämän kentän loppuvastusta heidän yhteisen menneisyytensä takia. Päihitä hänet.” Esimerkissä annetaan kyllä pelaajalle selvä maali, mutta on epätodennäköistä, että pelaaja olisi henkisesti täysillä mukana juonessa. Paras tilanne on, kun pelaajan ja pelihahmon motivaatiot saadaan samaistettua. Ei riitä, että hahmo vihaa kentän loppuvastusta, vaan pelaaja täytyy pyrkiä saada vihaamaan sitä myös. Tähän tarvitaan tietysti hieman pohjustusta. Tehokas tapa on laittaa loppuvastus aikaisemmassa kohdassa peliä ottamaan pelaajalta jotain pois, esimerkiksi hänelle tärkeän hahmon. Parempi vielä, jos kyseinen hahmo ei ole tehty tärkeäksi pelaajalle vain tarinan kautta, vaan hänellä oli myös paljon vaikutusta pelattavuuteen. Näin saadaan mukaan myös pelaajat, jotka eivät ole tempautuneet mukaan pelin tarinaan kovin paljoa. (Gard, T. 2010.)

Kuuluisa esimerkki tämän tyylisestä pelaajan motivoimisesta löytyy Final Fantasy VII:n tunnetusta juonipaljastuksesta. Pelin päävihollinen, Sephiroth, tappaa Aerisin, pelaajalle läheisen hahmon, kun peliä on jo pelattu pelaajasta riippuen todennäköisesti reilusti yli kymmenen tuntia. Sen lisäksi, että tarina on rakentanut Aerisista tärkeää hahmoa päähenkilölle ja jopa koko pelimaailmalle, on hänellä oma arvonsa myös pelattavuudelle. Hän on osa pelaajan

ryhmää ja häntä voi käyttää apuna taisteluissa. Koska peli on RPG, on Aerista tähän mennessä myös kehitetty paljon kokemuspisteillä ja muilla tehostuksilla. Kun Aeris kuolee, kukaan ei korvaa häntä. Hän on mennyttä ja kaikki pelaajan häneen käyttämä aika tuntuu hukkaan heitetyltä. Kohtausta on surullinen, mutta se ei ole ainoa syy, miksi monet pelaajat olivat järkyttyneitä sen takia. (Square 1997; Bowen Research 2011.)

Kohokohdat

Yleisessä pelin suunnitteluvaiheessa ja brainstorming-sessioissa tulee yleensä esiin yksittäisiä kohtauksia, jotka ovat hyvin voimakkaita tai mielenkiintoisia. Jopa koko peli-idea voi perustua johonkin tiettyyn kohtaukseen, jonka pelin tekijä haluaisi nähdä toteutettuna. Tällaisia kohtauksia voidaan pitää pelin kohokohtina, joiden ympärille kentät suunnitellaan. Kohtaukset, kuten lähes kaikki muukin pelissä, täytyy tietysti testata. Kaikki, joka kuulostaa äkkiseltään hyvältä, ei välttämättä toimi lopullisessa pelissä. Eräs aikaisissa esittelyfilmeissä esitetty kohtausta Metal Gear Solid 2: Sons Of Libertystä ei päätynyt lopulliseen peliin ollenkaan. Kohtauksessa pelaajan piti juosta sisälle virtaavaa vettä pakoon uppoavassa laivassa. Idea vaikuttaa juuri perinteiseltä kohokohdalta, mutta se ei selvästikään toiminut itse pelissä niin hyvin kuin luultiin. Syytä voi olla monia, mutta voisi esimerkiksi luulla, että kameraa päin takaa hyökkyviä vesimassoja pakoon juokseminen ei ollutkaan niin hauska pelata kuin miltä se kuulosti. (Gard, T. 2010; Konami 2002.)

Resident Evil kauhuselviytymispelisarjan kohokohdat kentissä ovat hyviä esimerkkejä, sillä niitä on paljon ja ne on helppo huomata. Kauhupelissä täytyy ollakin paljon asetelmia, jotka säikäyttävät pelaajan ja tuovat muutoksia perustilanteeseen. Sarjan ensimmäisessä osassa pelaaja saa suhteellisen vapaasti seikkailla pelottavassa kartanossa, joka on ripoteltu täyteen erilaisia toiminta-asetelmia. Hyvä esimerkki on eräs käytävä suhteellisen aikaisessa vaiheessa peliä. Se vaikuttaa tyhjältä ja ehkä tällä tuntemuksella myös psyykkää pelaajaa. Pelaajan edettyä käytävän keskivaiheille, eli turvattomimpaan kohtaan, särkyvät käytävän ulkoikkunat ja niistä hyppää sisälle hirviömäisiä koiria pelaajaa jahtaamaan (Kuvio 7). (Capcom 1996.)



Kuvio 7: Mutanttikoira yllättää selin olevan pelaajan (Learning Game Design with Arcade Berg 2010.)

Peruspilarit

Niin sanotut peruspilarit kuvaavat perusasioita, joita pelaaja voi tehdä pelissä. Toiminta-asetelmat ja muut kohtaukset kentissä täytyy suunnitella vastaamaan pelihahmon hallussa olevia kykyjä. Paras tapa suunnitella pelihahmon ominaisuudet ja niiden kehityskaari, on tehdä se vasta kaikkien pelin kenttien level flow-diagrammien suunnittelun jälkeen. Tällöin tiedetään, mitä kykyjä pelihahmo tarvitsee pelin kohokohtia varten. Päähahmo täytyy suunnitella avaimeksi pelikenttien tarjoamiin lukkoihin. (Gard, T. 2010.)

Jos pelihahmon peruskyyvyt eivät vastaa suunniteltuja kohtauksia, kohtausten interaktiivisuus laskee. Esimerkiksi tappelupeliin, jossa pelattavuus keskittyy pelimaailmassa ympäriinsä juoksentelusta ja vihollisten hakkaamisesta, voidaan yrittää tarinan edistämiseksi lisätä kohtaus, jossa päähahmo joutuukin yhtäkkiä ajamaan autolla pakoon perässä jahtaavaa kirahvilaumaa. Kyseiseen kohtaukseen voidaan vielä haluta lisätä pelaajalle mahdollisuus ampua auton ikkunasta. Kohtaus on pelattavuudeltaan kaukana pelin perimmäisestä olemuksesta, joten se saatetaan ratkaista pelkkänä välivideonä, johon on lisätty enemmän tai vähemmän interaktiivisuutta quick time eventtien avulla (Kuvio 8.). Hankalampi ratkaisu on ohjelmoida uudet pelimekaniikat vain tätä kohtaa varten. Tämä ei ole tehokasta pelikehityksen kannalta ja joskus

myös hieman heikko ratkaisu pelattavuuden kannalta. Jos suunnitellut kohtaukset eivät käy yhteen pääpelattavuuden kanssa, ne kannattaa yleensä jättää kokonaan pois. (Gard, T. 2010.)



Kuvio 8: Shenmue-peliä pidetään Quick Time Eventien keksijänä (101 Video Games That Made My Life Slightly Better 2009.)

Fiktio

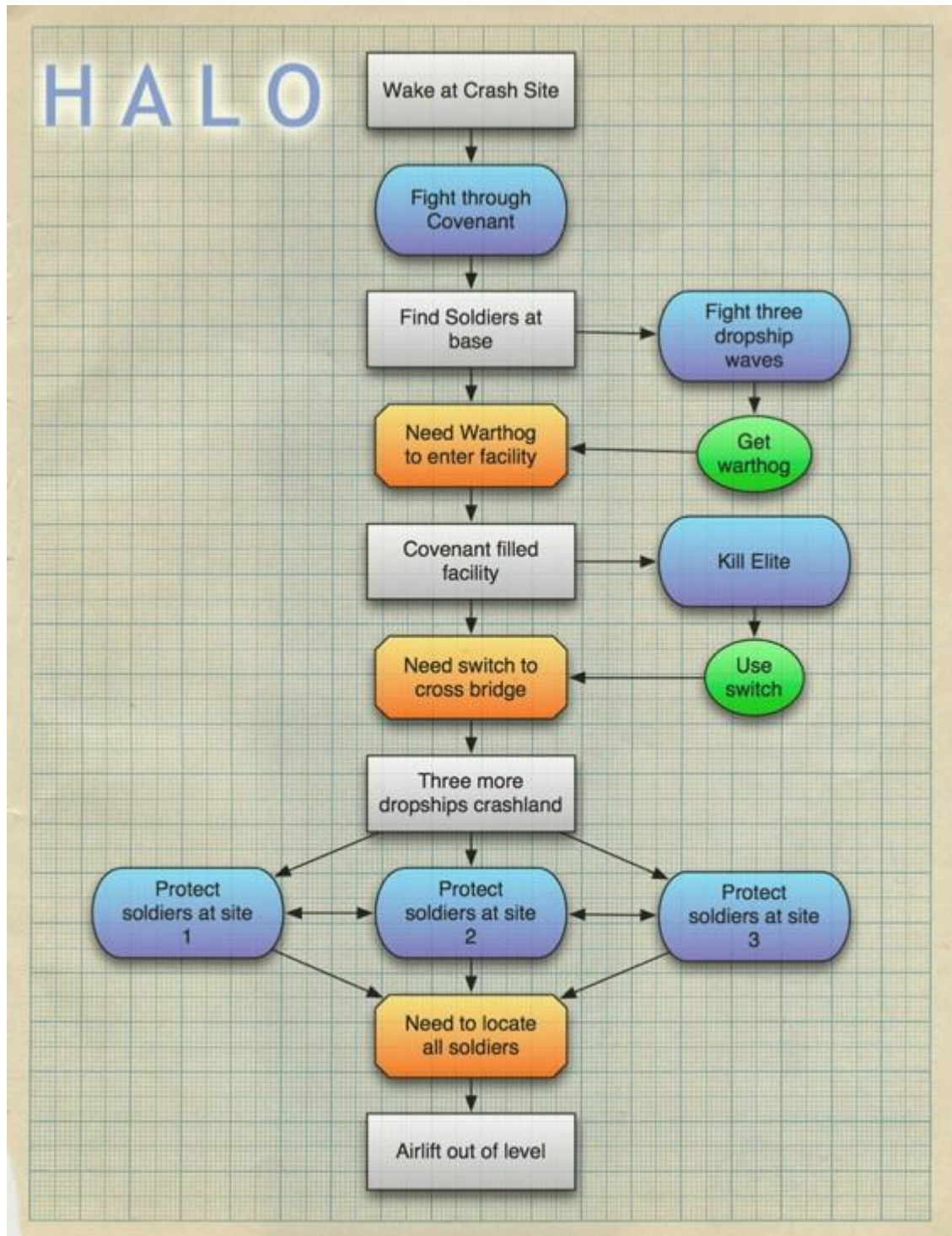
Jokainen pelin kenttä kertoo omalla tavalla tarinaansa. Jos jokin toimintapelin kenttä sijoittuu kuuuhun, täytyy kenttään ripotellut haasteet suunnitella sopimaan kuussa suoritettaviksi.

(Gard, T. 2010.)

Pelisuunnittelija saattaa hairautua suunnittelemaan kaikki esteet kentässä pelkästään mekaniikkojen kautta ja vasta jälkeenpäin pukemaan ne näyttämään joltain konkreettiselta. Kentän fiktio täytyy ottaa huomioon, jotta pelattavuus ja tarina eivät ole erotettu toisistaan. (Gard, T. 2010.)

Tiukka tarinan seuraaminen suunnitteluprosessissa voi tuntua kahlitsevalta, mutta lopullinen peli on heti parempi, jos sen fiktio saadaan pidettyä sisäisesti yhteensopivana. Kenttien esteetkin voivat olla helpompi suunnitella ja ne saavat varmasti uniikkeja piirteitä, sekä erottuvat toisten kenttien esteistä. (Gard, T. 2010; Björk, S. & Holopainen, J. 2005, 61.)

Lopullinen diagrammi



Kuvio 9: Flow Diagrammi osasta peliä Halo: Combat Evolved (ei alkuperäinen pelisuunnitelussa käytetty versio) (Gard, T 2010.)

Esimerkkikuvassa harmaat laatikot kuvaavat kentän virikkeitä, siniset laatikot pelaajan toimintoja ja tavoitteita, oranssit laatikot eräänlaisia pelillisiä lukittuja ovia ja vihreät laatikot

avaimia (Kuvio 9.). Tämän diagrammin mukaan tulee liittää aiemmin mainitut kenttäsuunnittelulliset elementit ja niiden sopivuus kyseiseen kenttään. (Gard, T 2010.)

Tämän jälkeen on jäljellä kentän arviointi ja mahdollinen hyväksyntä tai hylkääminen. Tämä vaihe on paras tehdä kun pelin kaikki kentät ovat jo tällä suunnitteluasteella ja niitä voi vertailla keskenään pelin alusta pelin loppuun. Näin voidaan huomata mahdolliset tyylikot, tylsät tai turhat kentät ja erilaiset sivuraiteille suistumiset. Tässä vaiheessa on mahdollista myös järjestellä kenttiä tai niiden elementtejä uudestaan, jotta peliin saadaan parempi rytmitys. (Gard, T 2010.)

Level flow-diagrammi on tarkoitettu kommunikoimaan kentän alkuperäinen idea lopulle tiimille. Näin ollen se on hyvä muistaa pitää niin yksinkertaisena ja lyhyenä kuin mahdollista. Se rakentuu kentän pohjimmaisesta luurangosta ja yksityiskohdat määritetään myöhemmissä suunnitteluvaiheissa tiimin kanssa. (Gard, T 2010.)

3.6 Käyttöliittymäsuunnittelu

Pelin käyttöliittymää suunnitellessa määritellään pohjimmiltaan kaksi asiaa: millä tavalla pelaaja voi vaikuttaa peliin ja kuinka peli antaa pelaajalle palautetta. Kaikissa peleissä on jonkinlainen käyttöliittymä, jonka kautta pelaaja on vuorovaikutuksessa pelin kanssa. Pelissä käyttöliittymän on hyvä olla sulava ja tuntua hyvältä, mutta ensisijaisesti on tähdättävä luomaan käyttöliittymä siten, että pelaaja tuntee olevansa pelikokemuksensa ohjaksissa. (Brathwaite, B. & Schrieber, I. 2008, 5; Schell, J. 2008, 222.)

Käyttöliittymä voi tarkoittaa monia asioita peliohjaimesta tietokoneen näyttöön ja pelin sisäisiin järjestelmiin. Pohjimmiltaan käyttöliittymä on kuitenkin se pinta, joka on pelaajan ja pelin välillä. Tähän kuuluvat pelaajan antama fyysinen tiedonsyöttö (esimerkiksi nappien painaminen) ja pelimaailman reagoiminen palauttamalla fyysistä ulostuloa (esimerkiksi kuvaa näytön kautta). Lisäksi on olemassa pelin sisäinen virtuaalinen käyttöliittymä, jos pelaaja ei ole toiminnoillaan täysin suoraan vuorovaikutuksessa pelimaailmaan. Strategiapelit ovat usein täynnä erilaisia mittareita ja valikkoja, jotka eivät varsinaisesti kuulu pelimaailmaan, vaan ovat osana käyttöliittymää. Command & Conquer-sarjan pelit näyttävät aina pienen energiapalkin valittujen armeijan yksiköiden päällä, joten pelaaja voi helposti nähdä aiheutetun tai vastaanotetun vahingon määrän, eikä hänen tarvitse päätellä sitä yksiköiden fyysisestä

ulkonäöstä, joka ei yleensä edes paljoa muutu vahingoituessa (Kuvio 10.). Jotkut pelit toimivat hyvin vähällä näkyvällä virtuaalisella käyttöliittymällä, mutta yleensä pelikehittäjille on helpompaa vain turvautua vaikkapa erilaisiin mittareihin. Tim Schafer on yksi pelisuunnittelija, joka on vuosien varrella vähentänyt ylimääräisiä käyttöliittymä-elementtejä peliruudulta ja monissa hänen peleissään pelaajat näkevät ruudulla suurimman osan ajasta pelkästään pelimaailman sekä sen visuaaliset ja äänelliset ilmaiset. Toiminnallisemmissa peleissä, kuten Psychonautsissa ja Brutal Legendissä, käyttöliittymä-elementit tulevat esiin piiloistaan vain silloin kun niitä tarvitaan ja antavat muuten tilaa pelimaailmalle. Aiemmin mainittu ICO toimii taas täysin ilman minkäänlaisia ylimääräisiä käyttöliittymä-elementtejä. (Schell, J. 2008, 223-224; Lucasarts 1993; Lucasarts 1995; Lucasarts 1998; Double Fine 2005; Double Fine 2009; Team Ico 2001.)



Kuvio 10: Command & Conquer 3: Tiberium Wars käyttää paljon HUD-elementtejä (IGN.com 2007.)



Kuvio 11: Brutal Legend piilottaa mittarinsa ja näyttää pelimaailman esteettömästi (Inglorius on Video Game UI & UX 2009.)

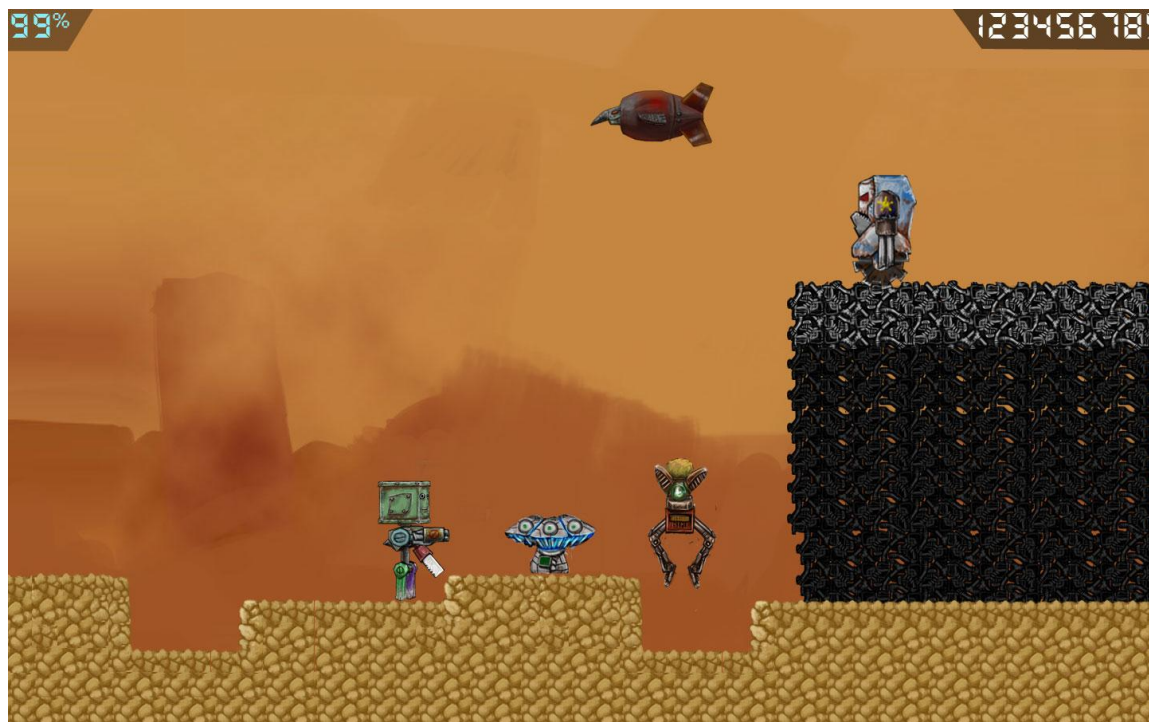
Perusasiat ovat selkeät, mutta tärkeä osa käyttöliittymäsuunnittelua on toimintojen kartoitus. Tällä tarkoitetaan tarkkoja määrittämiä esimerkiksi siitä, mitä eri nappien painaminen tekee pelimaailmalle. Jos pelaajalla on ohjaimessaan analoginen ohjausnappi, täytyy peliin määrittää, mihin vaikutetaan kyseistä nappia vääntelemällä. Näinkin itsestään selvään asiaan keskittyminen voi muuttaa pelattavuutta suuresti. Jos nappia käytetään pelaajahahmon liikuttamiseen, täytyy määrittää esimerkiksi miten nopeasti hahmo liikkuu ja vaikuttaako nupin vääntämisen voima hahmon nopeuteen. Samanlaista kartoitusta täytyy suunnitella kaikkiin muihinkin käyttöliittymän osiin. Miten esimerkiksi pelimaailma vaikuttaa virtuaaliseen käyttöliittymään ja mitä tietoja pelaaja näkee ruudulla? (Schell, J. 2008, 224-225.)

4 PELISUUNNITTELU PROJEKTISSA

Koko projektin ajan kestäneessä suunnitteluvaiheessa sivuttiin enemmän tai vähemmän josta aiemmin yksilöityä pelisuunnittelun osa-aluetta. Ajatuksena oli pitää pelisuunnittelullinen puoli joustavana ja kehittää uusia ideoita projektin edetessä. Pääsuunnitteludokumentti oli osa projektin ylläpitoa varten perustettua wiki-sivustoa, jotta sitä olisi helppo selata ja jatkuvasti muokata. Pohjan pelisuunnittelulle loi kuitenkin selkeä high concept-dokumentissa kuvailtu idea.

Robodashin perimmäisenä ajatuksena oli luoda proseduraalisesti generoitu arcadetyylinen, peruspelattavuudeltaan yksinkertainen peli. Pelattavuus on suunniteltu kaikin puolin heijastelmaan vanhan koulukunnan pelejä, mutta modernilla tavalla toteutettavaksi. Monet vanhat digitaaliset pelihallipelit olivat loppumattomia pisteidenkeräys-pelejä ja 2D tasohyppelytyylisiä pelejä on myös ollut markkinoilla hyvin kauan, mutta näiden kahden elementin yhdistäminen toimivaksi paketiksi vaati moderneja toteutustapoja. (Arcade History 2011.)

High Concept-dokumenttia mukailen, Robodash on siis peli, jossa pelaaja ottaa ohjatakseen hylätyn, keskeneräisen robotinraadon ja lähtee todistamaan pätevyyttään elottomaan maastoon (Kuvio 12.). Romumetallilla vuorattu vaarallinen maasto siintää loputtomiin ja pelaajan ei oleteta selviävän matkastaan. Mitä pidemmälle hän pääsee, sitä enemmän pisteitä hän saa, mutta peliä ei voi koskaan voittaa. Pelaaja voi, vanhanaikaiseen tyyliin, voittaa vain itsensä ja muut pelaajat, mikä sinänsä voi antaa paljon suurempaa tyydytystä kuin hengettömän pelin peittoaminen. (atariarchives.com 1984.)



Kuvio 12: Vanha mockup Robodashista antaa vielä hyvän kuvan perusideasta. (Project Vohweli 2010.)

4.1 Maailman suunnittelu

Robodash on tietoisesti suunniteltu yhden selkeän teeman ympärille. Tämä teema on ”äärettömyys”. Koska pelin ideointi perustui proseduraaliseen generointiin, joka mahdollistaa muun muassa loputtomien kenttien luomisen, oli kyseisen teeman valitseminen helppo päätös. Peli ideoitiin pelattavuudeltaan ”äärettömäksi”, joten tätä teemaa oli hyvä käyttää pelin muissakin osa-alueissa ja esimerkiksi selittää pelin äärettömyyttä tarinan kautta. (Compton, K. & Mateas, M. 2006.)

Pelin maailma on vieras planeetta, jolle pakkomieliteinen tieteilijä Sarai on haaksirikkoutunut. Sarain loputon robottitehtailu syrjäytti ajan myötä planeetan alkuperäiset asukkaat ja saasteet sekä enemmän tai vähemmän elollinen romu peittävät planeetan silmienkantamattomiin. Tämä jo sinällään kuvastaa pelin syklisyyttä. Jokainen pelisessio antaa pelaajan ohjata uutta robottia ja kyseinen robotti tuhoutuu pelisession päätteeksi ja päätyy osaksi planeettaa peittävää romua. Pelin tarina ei koskaan lopu, vain taustatarina paljastuu ja selittää pelaajalle tämän tilanteeseen johtaneita tapahtumia.

Maailman tilaa ei selitetä pelaajalle suoraan vaan pikemminkin vihjeitä siitä ripotellaan ympäri peliä. On paljon parempi herättää pelaajan mielenkiinto ja mielikuvitus kuin kertoa asiat suoraan. Lisäksi vihjeiden avulla pelaaja voi kokea niin sanottuja ”ahaa-elämyksiä”. Suuri mysteeri on, miten pelimaailma oikeastaan on loputon? Pelaaja ei välttämättä edes ajattele tätä seikkaa. Ehkä maailma ei ole loputon, vaan ainoastaan valtavan suuri? Jos pelaaja kuitenkin tuntee historiallisia symboleita, voi hän huomata erilaisia äärettömyyteen ja syklisyyteen viittaavia rakennelmia pelin taustamaisemissa, vihjaamassa planeetan alkuperäisasukkaiden kulttuurista ja myös selittämässä jotain. (Flogging the Quill 2006.)

Saastunut ja romukasoin peittyneet maailma luo peliin tietynlaisen tunnelman. Vastaavanlainen konsepti on suosittu ja tuhon jälkeisiä maisemia nähdään peleissä vähän väliä. Ympäristö on tietyllä tavalla klisee, mutta se sopii erinomaisesti aihepiiriin. Tuhoutuneen maiseman melankolinen kauneus on kuitenkin yhä pelaajien suosiossa, vaikka markkinat tuntuisivat olevan lähes kyllästettyjä tämän tyyllisillä kuvauksilla. Graafikkaa neuvottiinkin käyttämään referenssikuvina muita samanlaista kuvausta sisältäviä sarjakuvia, elokuvia ja pelejä. Esimerkkinä käytetyistä referensseistä ovat Yukito Kishiron Gunnm-sarjakuvan (<http://jajatom.moo.jp/E-top/frame.html>) ilmapiiri ja romukasat sekä Machinarium-pelin (<http://machinarium.net/demo/>) ”elolliset” romut ja hajonneet robotit. (Kuvio 13.) (Kuvio 14.) (Valve 2004; Epic Games 2006; Obsidian Entertainment 2010.)



Kuvio 13: Gunnm animaatioversiosta otettiin inspiraatiota pelin värimaailmaan. (Ingrid Richter 2011.)



Kuvio 14: Machinariumissa on näytillä paljon tuhon estetiikkaa (brej.org 2010.)

4.2 Mekaniikan suunnittelu

Koska Robodash ei vielä tässä vaiheessa ole valmis peli, on sen pelimekaniikkojen suunnittelukin keskeneräisessä tilassa. Ideatasolla perusmekaniikat ovat valmiina ja suunniteltu pelisuunnitteludokumenttiin, mutta tämä osa pelikehitystä ei oikeastaan voi olla valmis ilman pre-alpha-testausta. Paperilla hyvältä näyttävät ideat eivät välttämättä toimi yhdessä muun pelin kanssa tai eivät ole ollenkaan hauskoja kun niitä pääsee testaamaan. (Pedersen, R. E. 2009, 80.)

Voittoehdot

Robodash on tällä alueella hieman epätavallinen. Se on suunniteltu sykliseksi, ”loputtomasti” pelattavaksi. Lähin voittoa muistuttava tapahtuma on itse asiassa häviö, joka lopettaa pelin ja tallentaa pelaajan pisteet.

Voittoehdot myös kommunikoidaan pelaajalle epätavallisella tavalla. Pelin alkaessa pelaaja on omillaan ja pelin päämäärästä vihjaa ainoastaan matkamittari ja lyhyt alkuvideo, jossa pelaajan robottihahmon luoja hylkää robotin epäonnistuneena projektina. Pelaaja siis näkee ja tietää

kaiken, minkä hänen ohjaamansa robotti on nähnyt ja tiennyt koko lyhyen elämänsä aikana. Pelaajan motivointiin käytetään hyväksi pelaajille tuttuja pelimekaniikkoja. Vaikkei pelaaja pelkästä matkamittarin olemassaolosta ymmärrä pelin tarkoitusta, lähtee hän silti todennäköisesti tutkimaan avartuvaa pelimaailmaa. Viimeistään pelaajan ensimmäinen kuolema aukaisee pelin tarkoituksen. Pelihahmon tuhoutuessa sen luoja, Sarai, ilmestyy kommentoimaan pelaajan menestystä tyyliin: ”Pääsit pidemmälle kuin kuvittelinkaan. Ehkäpä et olekaan täysin epäonnistunut robotti...” ja pelaaja saa kirjoittaa nimensä pistelistalle, jossa pisimmälle päässeet robotit on listattu ylimpänä. Pelaaja voi silloin aloittaa uuden pelin ja pyrkiä rikkomaan ennätyksensä, avartaen siinä ohessa pelimaailmaa.

Pelaajan toiminnat

Robodashin suurin vaikutte on Capcomin Mega Man-sarja. Klassisissa Mega Man-peleissä on pelaajalla käytössä vain muutama perustoiminto. Mega Man osaa pohjimmiltaan hyppiä sekä ampua (myöhemmissä osissa myös liukua) ja näinkin yksinkertaisen hahmon ympärille on voitu rakentaa tyydyttävä pelattavuus. Kenttäsuunnittelu on rakennettu vastaamaan näitä toimintoja ja pienet päivitykset Mega Manin aseistukseen tuovat vaihtelua. Robodashissa pelaajan perustoiminnot on myös pyritty pitämään yksinkertaisina ja välittömästi opittavina. (Capcom 1988; Capcom 1990.)

Myös Robodashin pohja on hyppimisessä ja ampumisessa. Näitä ominaisuuksia on kuitenkin monella tapaa monipuolistettu. Peli perustuu paljon erilaisiin power upeihin, jotka muuttavat peruspelattavuutta. Pelaajan robotti pystyy keräämään power up-osia molempiin käsiinsä jalkoihinsa tai kehoonsa. Pelaajalla on siis käytössään neljä power up paikkaa ja niiden täytyttyä voi hän vaihtaa tietyt osat toisiin. Esimerkiksi robotin molempiin käsiin voi sijoittaa erilaisen aseensa ja käyttää niitä sopivissa tilanteissa. (Kuvio 15.)



Kuvio 15: Robo ampuu oikean käden aseellaan (Project Vohweli 2010.)

Lista pelaajan näppäimistä (Xbox 360-ohjaimelle) on seuraavanlainen: (Kuvio 16.)

Toiminta	Nappi
Liikkuminen / tähtääminen	Directional pad / Left stick
Vasemman käden toiminta	Left trigger
Oikean käden toiminta	Right trigger
Hyppy	A button
Ota / jätä vasemman käden power up	Left bumper
Ota / jätä oikean käden power up	Right bumper
Ota / jätä jalkojen power up	Alas + Left bumper / Alas + Right bumper
Ota / jätä kehon power up	Y button
Taukovalikko	Start button



Kuvio 16: Xbox360 ohjaimen nappien sijainnit (cuhkacs.org 2007.)

Näitä osia pelimekaniikasta oli tarkoitus kehittää jatkuvan pelitestauksen kautta, mutta projekti ei koskaan ehtinyt siihen vaiheeseen, että aivan pohjimmainen peruspelattavuus olisi saatu toimimaan tyydyttävästi. Esimerkiksi erilaisia näppäinkomentoja saattaa olla tällä hetkellä liikaa pelaajan käytettävissä ja niitä pitäisi virtaviivaistaa kontekstitarvuuksilla. (Giant Bomb 2011.)

4.3 Sisällön suunnittelu

Tätä vaihetta ei Robodashissa päästy kovin pitkälle toteuttamaan, koska perusmekaniikkojen ohjelmointi vei niin paljon aikaa. Tämä tarkoittaa myös sitä, että sisällön suunnittelu tapahtui enimmäkseen paperin ja kynän avulla, eikä ideoitua ja suunniteltua sisältöä päästy testaamaan ja kehittämään tarvittaviin suuntiin.

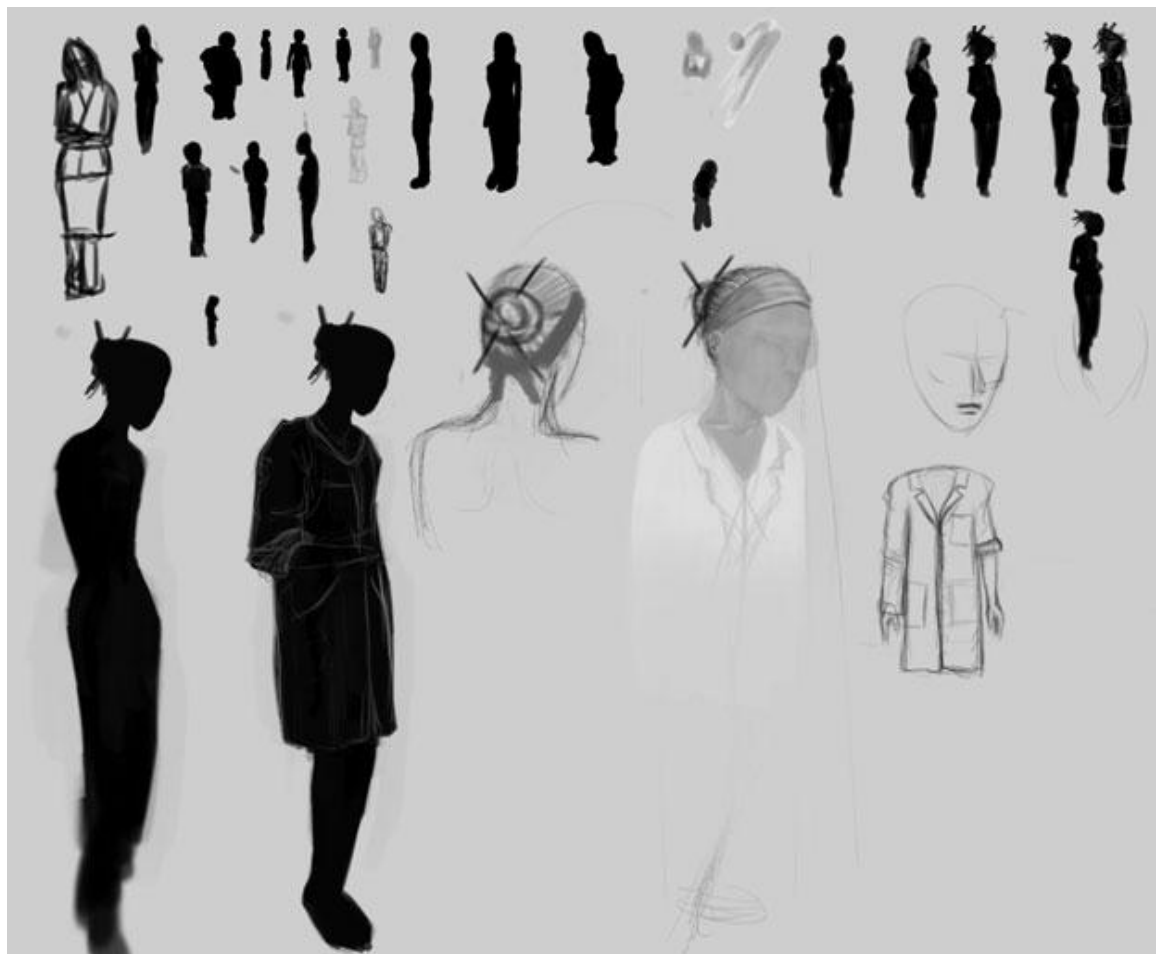
Hahmot

Jokaista näennäisen elollista olentoa käsiteltiin suunnitteluvaiheessa hahmona. Jopa kaikille vihollistyypeille pyrittiin kehittämään hauska persoonallisuus, joka vaikuttaisi niiden toimintaan ja lisäisi luonnetta pelimaailmaan.

Tärkein hahmo pelattavuuden kannalta on tietenkin pelaajan ohjaama robotti, jota voidaan kutsua kehitysnimellä Robo. Robo on melko suoraan avatar-tyyppinen hahmo ja pelaajaa itse

asiassa ajatellaan sen tekoälynä. Pelin päätyttyä pelaaja saa lisätä nimensä pistelistaan robotin tekoäly-sarakkeeseen. Robo rakennetaan jokaisen pelikerran alussa erilaisista paloista, mutta tekoäly pysyy samana ja kehittyy. (Schell, J. 2008, 312.)

Pelaaja kokee Robon koko tarinan kun taas Robon luoja Sarai on mystisempi hahmo (Kuvio 17.). Hänen funktionsa pelissä on enimmäkseen tarinankerronta, mutta hän myös selkeyttää pelin tavoitteita. Sarai on ei-pelattava hahmo, josta pelaaja ei tiedä aluksi mitään. Hän työskentelee telaketjuilla liikkuvassa laboratoriossaan ja rakentaa jatkuvasti uusia robotteja maailmaan. Hahmon mystisyyden ja oudon käyttäytymisen on tarkoitus herättää pelaajassa mielenkiinto. Pelissä etenemällä ja saavuttamalla parempia tuloksia voi pelaaja oppia lisää Saraista avaamalla hänen päiväkirjatekstejään, tulkitsemalla ympäröivää maailmaa ja kuuntelemalla hänen kommenttejaan pelisessioiden jälkeen. Hänen tarinansa ei oikeastaan etene mihinkään suuntaan, se vain on olemassa menneisyydessä ja sitä halutessaan selvittämällä voi pelaaja saada paljon enemmän irti pelistä.



Kuvio 17: Aikaisia Sarai-luonnoksia (Project Vohweli 2010.)

Naishahmoja on perinteisesti pidetty suhteellisen pinnallisina videopeleissä. Sarain suunnittelussa on pyritty välttämään tällaista pinnallisuutta, eikä hän ole pelkkää silmänruokaa. Hahmon sukupuolen ei tarvitse vaikuttaa hahmoon hirveästi, mutta mielenkiintoisia naishahmoja voidaan luoda miettimällä ensinnäkin sukupuolien geneettisiä ja sosiaalisia eroja ja soveltamalla niiden tuottamia odotuksia hahmon kehitykseen. (Escapist 2011.)

Sarai on, alussa pelaajalle tietämättä, androidi. Hänen taustatarinansa mukaan hän on biomekaaninen, ”täydellinen” kopio luojastaan, joka halusi pidentää elämänsä loputtomiin siirtämällä tietoisuutensa tähän ”nukkeeseen”. Asiat kuitenkin menivät pieleen ja Sarai päätyi vieraan planeetan pinnalle, luojansa tietoisuus tekoälyn rinnalle siirrettyinä ja pahoin vaurioituneena. Tämä teki hänestä hyvin sekavan ja hän yrittää ajatustensa sumussa seurata luojansa inhimillisiä piirteitä ja tahtoa, erottamatta itseään hänestä. Sanalla sanoen, Sarai on hyvin paljon päästään vialla. Hän kopioi luojansa käytöstä rakentaen jatkuvasti epätäydellisiä robotteja ja heittäen ne pois. Samalla hän on eräänlainen äitihahmo kaikille maailmansa roboteille, mutta tämä asia sekoittaa häntä yhä enemmän.

Robodashin muut hahmot ovatkin oikeastaan sen kenttäviholliset. Kaikki niistä on ensisijaisesti suunniteltu pelattavuuden tarpeiden mukaan, mutta niille on myös kehitetty samalla persoonallisuus niiden toiminnallisuuksiin liittyen (Kuvio 18.). Useat robotit ovat teemaltaan tragikoomisia. Esimerkiksi yksi kehitystiimin keskuudessa suosittu hahmo, Turtlebot, suunniteltiin pelillisesti yksinomaan ansaksi. Kun pelaaja vahingoittaa sitä, laukaisee se ilmoihin pienen ydinohjuksen, joka aiheuttaa valtavaa tuhoa. Muuten Turtlebot on harmiton. Konsepti muistuttaa erilaisista ydinaseturva-teorioista, joiden mukaan rauhaa voidaan ylläpitää maailmanlaajuisella ydintuholla pelottelemalla: jos joku laukaisee ohjuksensa, hyökkäyksen uhriksi jäänyt laukaisee heti omansa ja ketjureaktio aiheuttaa täystuhon. Turtlebot suunniteltiin tämän teorian kannattajaksi, rauhan ylläpitäjäksi, joka ei pelkää tarvittaessa painaa nappia. Tämä näkyy sen rauhanomaisessa ulkomuodossa ja symboliikassa. (Götterdämmerung.org 1990.)



Kuvio 18: Lentävät robotit pakottavat pelaajan varomaan hyppyjään ja liikkumisrytmiään (Project Vohweli 2010.)

Kaikkia puolia eri robottien persoonallisuuksista ei pysty noin vain esittämään niiden ulko-
muodossa tai käytöksessä. Robodashiin suunniteltiin myös vihollistietokanta, jota pelaaja voi
avata tuhoamalla vihollisia. Tässä vihollislistauksessa kerrotaan vihollisista tarinallisesti, mut-
ta seassa on myös vihjeitä niiden peittoamiseen.

Esineet

Ilman peliä muuttavia esineitä Robodash olisi hieman liian yksinkertainen tasohyppely. Eri-
tyisesti pelin power upit on suunniteltu monipuolistamaan pelikokemusta.

Pick upeja pelissä on melko vähän, mutta ne ovat silti tärkeitä. Tuhoutuessaan viholliset jättävät joskus jälkeensä energiakuulia, jotka lisäävät Robon energiaa. Idea on monesta pelistä tuttu, mutta nämä energiakuulat on suunniteltu hieman erilaisiksi, jotta ne sopisivat Robo-dashin pelattavuuteen. Ne eivät pysy paikallaan, vaan pomppivat Robon etenemissuunnan vastaiseen suuntaan. Yleensä pelaajan pitää jahdata niitä ja hän joutuu samalla liikkumaan taaksepäin. Tämä tuo hieman ylimääräistä mietittävää pelaajalle. Kannattaako hänen ottaa energiakuula kiinni vai jatkaa eteenpäin tuhlaamatta aikaa ja ottamatta turhia riskejä. Kaikki oikeastaan riippuu pelaajan sen hetkisestä energiatilanteesta ja pelaajan riskiarvioinnista. Kaiken lisäksi energiakuulia on kolmenlaisia: (Giant Bomb 2011.)

Energiakuula	Lisäenergia	Liike	Harvinaisuus
Pieni	3	Pomppii vain muutamaman kerran.	Melko yleinen.
Keskikokoinen	6	Pomppii melko korkealle ja kauas.	Ei kovin yleinen.
Suuri	12	Pomppii melko nopeasti loputtomiin.	Harvinainen.

Yllä olevan taulukon arvot on luonnollisesti tarkoitettu muutettaviksi projektin edetessä. Ilman pelitestaamista voi olla todella hankala sanoa, rikkovatko esimerkiksi liian suuret tai pienet energiapalautukset koko pelin.

Vielä suurempi rooli on pelin monilla power upeilla. Pelaajalla käytössä olevat neljä power up paikkaa mahdollistavat suuren valikoiman erilaisia yhdistelmiä ja jokainen power up muuttaa pelattavuutta huomattavasti. Seuraavalla sivulla taulukko muutamasta power upista ja niiden toiminnallisuudesta:

Power up	Selitys
Pea Shooter Gun	Pieni käsiase, jolla pelaaja voi tähdätä kahdeksaan suuntaan nuolilla. Heikko tuhovoi- ma, mutta kykenee sarjatulitukseen.
Peace Ray	Valtavan tuhovoimainen käsiase. Hyvin hi- das latautumaan. Sillä voi ampua vain eteen- päin, mutta säde on hyvin paksu ja kantaa kauas tehden suurta tuhoa.
Feather Battery	Robotin kehon lisäosa, joka antaa pelaajalle mahdollisuuden hypätä uudestaan ilmassa tehostaen hypyn ohjattavuutta.
Treads	Telaketjut robotin jalkojen tilalla. Tekevät robotista hitaamman ja huonomman hyppi- mään, mutta parantavat liikkuvuutta maassa. Lisäksi viholliset vahingoittuvat kosketukses- ta jyrääviin teloihin.

Myös power upien tasapainottaminen on hyvin vaikeaa ilman niiden kokeilua toimivan pelin vihollisia ja esteitä vastaan. Jokin power up-yhdistelmä voi olla myös liian ylivoimainen muihin nähden ja tehdä niistä turhia. Siksi siis tällaiset listat pelisuunnitteludokumentissa muut- tuvat jatkuvasti projektin edetessä. (Brathwaite, B. & Schrieber, I. 2008, 14.)

Vaikkei näiltä osin testausvaiheeseen päästykään, suunniteltiin power upeja varten muutamia tasapainottavia ratkaisuja. Jos esimerkiksi jokin power up todettaisiin liian voimakkaaksi, voi- taisiin sen ammuksista tehdä määrällisiä loputtomien sijaan. Varsinkin Peace Rayn kohdalla mietittiin valmiiksi erilaisia heikennyksiä. Yksi vaihtoehto oli laittaa se tuhoamaan muut käy- tössä olevat power up-aseet, kun sillä ammutaan.

4.4 Pelikäsikirjoitus

Robodash on suunniteltu sen pelattavuuden ympärille. Tämä ei silti tarkoita, että käsikirjoit- taminen olisi laiminlyöty. Vaikka pelistä voi nauttia pelkästään pelattavuuden pohjalta, eikä pelimaailman tarinan seuraaminen ole pakollista, on tarina silti olemassa, eikä se ole mikään jälkiajatus. Tarina pilkistää esiin pelin monista elementeistä. Siihen voi alkaa kiinnittää

enemmän ja enemmän huomiota jos kiinnostus riittää, mutta tarinalla ei hidasteta sulavaa pelattavuutta.

Pelin tarinan käsikirjoitusprosessi alkoi aikaisessa vaiheessa. Kun pelin perusidea (loputon tasohyppely) oli selvillä, alettiin peliä muun suunnittelun ohessa miettiä tarinallisesti. Miksi maailma on loputon? Mitä se symboloi? Miten tarinaa voidaan kertoa tällaisessa pelissä? Tarinan ideaa muutettiin pelin tarpeiden mukaan ja pelin elementtejä muutettiin tarinan tarpeiden mukaan.

Pelin hahmojen ja maailman taustatarinat kirjoitettiin kokonaisuudessaan yleisellä tasolla pelisuunnitteludokumenttiin. Tätä tekstiä ei missään vaiheessa tarkoitettu pelaajan käsiin. Se oli olemassa lähinnä referenssinä muuta suunnitteluprosessia varten ja sitä pystyi muuttamaan koko prosessin ajan. Kun tarinat olivat kirjoitettu enemmän tai vähemmän yksityiskohtaisesti, oli aika alkaa miettiä tarinankerrontaa. Mikä osa näistä tiedoista välitetään pelaajalle suoraan ja mikä osa jää hämärän peittoon? Missä kohdassa peliä nämä tiedot välitetään ja miten? (Tang, C. 2005.)

Laiska tapa hoitaa pelin tarina on vain laittaa pelaaja lukemaan taustatarinoita sellaisenaan. Silloin käytetään pelkästään kirjallisuuden tarinankerronnan keinoja eikä sovelleta pelin ominaisuuksia. Varsinkin vanhempien pelien täytyi joskus nojata tähän tekniikkaan. Pelit ovat mediana siitä erikoisia, että niissä voi käyttää ja käytetäänkin lähes kaikkien muiden tarinankerronnallisten medioiden tarinankerrontamenetelmiä. Suurimman hyödyn peleistä tarinankerronnan välineinä saadaan tietysti käyttämällä peleille ominaisia ominaisuuksia tarinankerrontaan ja sekoittamalla niitä muiden medioiden menetelmiin. (The Escapist 2010.) (Chandler, R. 2005.)

Pelin tarinaa avataan pelaajalle taustalla näkyvien objektien, hahmojen käytöksen, Sarain lopukommenttien ja erilaisten dokumenttien avulla. Pätkät Sarain päiväkirjasta ja vihollisten kuvaukset robottikirjastosta ovat pelkkää tekstiä kuvien höystämänä. Nämä tekstit ovat tavallaan ylimääräisiä osia pelissä ja niitä voi lukea jos pelaajan kiinnostus pelimaailmaa kohtaan kohoaa ja jos pelaaja haluaa selvennystä tulkintoihinsa. Ne eivät paljasta tarinaa suoraan, vaan niissä on pyritty käyttämään erilaisia tarinankerronnallisia tekniikoita. Päiväkirja on luonnollisesti kirjoitettu sekavan Sarain näkökulmasta ja robottikirjasto on kirjoitettu hieman luontodokumenttityyliin eräänlaisen tutkijarobotin kuvakulmasta. (Tang, C. 2005.)

4.5 Kenttäsuunnittelu

Koska Robodash perustuu proseduraaliseen sisällönluomiseen, ei kenttäsuunnittelua voitu tehdä tavanomaisen level flow-diagrammin avulla. Tästä huolimatta ”kenttiä” pystyi silti suunnittelemaan aiemmin yksilöityjen kenttäsuunnitteluelementtien avulla. (Gard, T 2010.)

Robodash kokonaisuudessaan sisältää tavallaan vain yhden kentän, joka kestää koko pelisesion ajan loputtomiin. Tapahtumia ei voitu tarkalleen käsikirjoittaa, vaan kentän rytmi täytyi pyrkiä ohjelmoimaan kenttägenerointiin ja kentän elementtien käytökseen.

Motivaatio

Robodash suunniteltiin pohjimmiltaan hyvin pelimäiseksi. Kun pelaaja aloittaa pelin, on pohjustuksena vain hyvin lyhyt välivideo, joka näyttää pelaajan ohjaaman robotin kokoamisen ja hylkäämisen. Kun robotti heitetään ulos, on pelaaja välittömästi ohjaksissa, tietämättä varsinaisesti tavoitettaan pelissä. Vihjeen pelin tavoitteesta tarjoaa yläkulmassa oleva matkamittari, joka kasvaa pelaajan edetessä. Ensimmäinen pelisessio on todennäköisesti lyhyt ja pelaaja tapattaa robottinsa hyvin nopeasti, jolloin lyhyessä loppuvideossa robotin rakentaja kommentoi hajonnutta robottia sen mukaan, kuinka pitkälle pelaaja on edennyt. Tämän pitäisi viimeistään vihjata pelaajalle pelin tavoite.

On osittain outo ratkaisu raottaa pelin tarkoitusta pelaajalle enemmän vasta pelaajan kontrolloiman hahmon kuoleman jälkeen, mutta Robodash on teemansakin mukaan loputon ja syklinen. Uusi robotti rakennetaan aina kun pelaaja aloittaa pelin ja jokainen epäonnistuminen raottaa pelin juonta. Juoni muutenkin on hieman salakavalasti pelimäisyyden takana piilossa, eikä pakollinen pelinautintoon. Se voi silti syventää kokemusta ja tällä tavoin pelaajaa motivoidaan sekä pelillisellä puolella kuin juonellisellakin.

Kohokohdat

Varsinaisia pelattavia kohokohtia ei pelin luonteen takia pystytty valmiiksi suunnittelemaan. Suunnitteluvaiheessa pyrittiin lähinnä keksimään pelaajalle paljon erilaisia toimintoja ja vastuksia, jotka voisivat tuottaa hienoja kohtauksia.

Yksi selkeä tapa, jolla kuitenkin yritettiin luoda vaihtelevuutta ja juonipaljastuksiakin oli niin sanottujen taustaobjektien käyttö. Nämä ovat pelkkiä graafisia mausteita, jotka menevät ohi taustalla tai etualalla kun pelaaja etenee maailmassa. Jos pelaaja etenee tarpeeksi pitkälle, voi hän esimerkiksi kohdata patsaita planeetan alkuperäisistä asukkaista tai maahan syöksyneitä avaruusaluksia. Nämä objektit osaltaan paljastaisivat pelimaailman tarinaa, toisivat vaihtelua ympäristöön ja saattaisivat jopa motivoida pelaajaa tutkimaan maailmaa lisää ja tavoittelemaan parempia tuloksia.

Robodashiin suunniteltiin myös osittain rikkoutuva maasto pelattavuuden monipuolistamista varten. Osa maastosta koostuu romukasoista, jota voi hajottaa enemmän tai vähemmän käytetystä aseesta riippuen. Tämä voi mahdollistaa pelattavuuden puolesta erilaisia kohokohtia. Pelaaja saattaa voida esimerkiksi kaivaa tiensä vaarallisen maaston läpi tai murskata maaston ja viholliset suurilla aseilla. Kaikki tällaiset lisäelementit tuovat lisää mahdollisuuksia tuottaa kohokohtia proseduraalisen generoinnin avulla.

Pelin peruspilarit

Kun Robodashin peruspelattavuuden idea oli selvillä, oli helppo suunnitella Robolle tarvittavat pelilliset ominaisuudet. Koska peli koostuu yksinomaan 2D-maaston navigoinnista ja vihollisten tuhoamisesta, Robon perustoiminnoiksi suunniteltiin tietysti hyppiminen ja ampuminen. Kaikki pelin viholliset on tehty tuhottavaksi tai ohitettavaksi millä tahansa power upilla.

Fiktio

Pelattavuus voi fiktion puolesta mennä oikeastaan melko epärealistiseksi, eikä sen pitäisi vaikuttaa pelinautintoon negatiiviseksi, koska pelimaailma on höystetty lähes fantasiamaailloilla, selittämättömillä scifi-elementeillä. Maailma ei siis ole kovinkaan rajoittava, mutta on olemassa silti muutamia sääntöjä, jotka estävät esimerkiksi täysin epärealististen, piirretyistä tuttujen fysiikoiden käytön. Vaikka peli ei olekaan realistinen ja vaikka jotkin animaatiot on tarkoitettu hieman humoristisiksi, pysytään kuitenkin näennäisen realistisella alueella pelimaailman nähdessä. Sisäisesti yhdenmukainen maailma on avain pelaajan ”epäuskon” välttämiseen, jos sellaista pidetään tarpeellisena. (Bishop, J. 2011.)

Näitä asioita täytyi miettiä enimmäkseen juuri pelin graafisessa puolessa. Vaikka pelin vihollisten käytöksen suunnittelussa oli oikeastaan taivas rajana, täytyi se oikeuttaa jotenkin pelimaailmaan nähden. Graafinen ulkonäkö selittää robottien toimintaa ja sitoo suunnitellut robotit pelimaailmaan. Koska konseptina oli täyttää maailma enemmän tai vähemmän elollisella rojulla, piti robottien ja kaiken muunkin näyttää siltä.

4.6 Käyttöliittymäsuunnittelu

Jo aiemmin esiteltyt pelissä käytettävät näppäimet antavat hyvän kuvan pelaajan keinoista vaikuttaa pelimaailmaan. Näitä näppäinasetuksia täytyi jo aikaisessa suunnitteluvaiheessa testata eräänlaisella kuivaharjoittelumenetelmällä. Koska peli ei ollut pelattavassa kunnossa, täytyi vain keksiä näppäinmäärittelyt ja kuvitella pelaavansa peliä, käyttäen määriteltyjä näppäimiä. Näin saatiin edes jonkinlainen kuva näppäimien sijoittelun toimivuudesta ilman oikeaa pelitestausta. Tämä ei tietenkään riitä ja näppäinasettelut jäivät hieman hiomattomiksi ilman kunnollista testaamista pitkien pelisessioiden myötä. (Pedersen, R. E. 2009, 80.)

Hieman pidemmälle päästiin pelin virtuaalisen käyttöliittymän suunnittelussa. Se koostuu pienistä ruudun yläreunaan sijoitetuista mittareista. Niiden graafisen ulkonäön on tarkoitus kuvastaa digitaalista HUD:ia. Pelaaja on siis tavallaan Robon pään sisällä kuin AI:n pitäikin olla. Tiedot ovat koko ajan ruudulla näkyvissä, mutta ne on pyritty asettelemaan niin, etteivät ne ole toiminnan edessä. Mittarit ovat pieniä ja hieman läpinäkyviä. (Wilson, G. 2006.)

Vasemmassa yläreunassa on Robon energialukema prosentteina. Lukema laskee pelaajan vahingoittuessa ja nousee pelaajan kerätessä tuhotuista vihollisista joskus saatuja energiapaketteja. Kun mittari näyttää nollaa prosenttia, peli päättyy. Pienenä, enimmäkseen visuaalisena mausteena mittari näyttää maksimissaan vain 99 %. Tämä kuvastaa sitä, miten Robon valmistus jäi puoliteihen ja se ei ole täydellinen.

Ruudun oikeaan yläreunaan on sijoitettu matkamittari. Se laskee koko ajan pelaajan etenemistä pelimaailman metreissä. Tietyt mittarilukemat laukaisevat tapahtumia, kuten tiettyjen taustaobjektien rullautumista taustalle pelaajan edetessä.

Näiden pakollisen tarpeellisten elementtien lisäksi suunniteltiin peliin vielä mittari, joka näyttää käytössä olevat power up-osat ja niihin sidotut näppäimet. Mittari olisi pieni kaavakuva

robotista, jonka jalat ja kädet muuttuisivat power upien mukaan. Tarvittaessa energiamittarin voisi myös siirtää tämän kaavakuvan päähän, jotta tilaa säästyisi. Tämä mittari todettiin kuitenkin ylimääräiseksi ja päätettiin luottaa pelaajan kykyyn muistaa, mistä napista mitäkin power upia käytetään. Näkyväthän Robon power upit graafisesti itse pelissäkin, joten voi olla turhaa selkeyttää tätä sen enempää. Tämä elementti oli kuitenkin tarkoitus palauttaa peliin jos testauksen aikana sen todettaisi helpottamaan pelaamista.

5 YHTEENVETO JA POHDINTA

Vaikka valmista peliä ei saatu varatussa ajassa luotua, oli kokemus silti tärkeä ja opettavainen tulevaisuutta varten. Pelkän teoriataustan osaaminen ei riitä projektin onnistumiseen, vaan on melkein pakollista olla jotain kosketusta käytännöntyöhön, jotta sen riskit ja muut seikat voidaan ymmärtää. Jos esimerkiksi tiimin pelisuunnittelijalla ei ole vahvaa kosketuspintaa grafiikoiden luomiseen tai ohjelmoimiseen, voi hänen olla hyvin vaikea arvioida, kuinka paljon aikaa yksittäiset työt voivat viedä. Esimerkiksi yhden hahmon käyttäytymisen ohjelmointi voi viedä paljon enemmän aikaa kuin mitä sitä varten on varattu. Tämä aiheuttaa ongelmia varsinkin, jos pelisuunnittelija on vastuussa projektin aikataulusta ja muista ylläpidon asioista. Kaiken lisäksi eri tiimin jäsenten taidot omilla osa-alueillaan voivat olla vaihtelevia. Kaikilla on myös omat tapansa työskennellä ja muun tiimin täytyy tiedostaa ne sekä sopeutua niihin.

Robodashin kehittämisen keskeytymisen syyt voisi tiivistää tekijöiden kokemattomuuteen. Aiemmin opittujen taitojen soveltaminen ei riittänyt ja suuri osa projektin ajasta kuului kaikenlaiseen opetteluun. Käytettävät työvälineet ja menetelmät olivat monille tiimin jäsenistä suhteellisen vieraita. Kokemattomuudesta johtui myös tarvittavan ajan aliarviointi, sillä kaikki pelin osa-alueet vaikuttivat teoriassa helpommilta toteuttaa kuin mitä ne olivat. Niinkin yksinkertainen asia kuin hahmon liikuttaminen ja perus törmäyksentunnistus kehitettyä pelimoottoria käyttäen vei todella paljon aikaa. Ajan säästämiseksi olisi suunniteltuja ominaisuuksia pitänyt karsia hieman rohkeammin. Valmiiksi suunniteltuja hienoja ideoita voi olla hankala jättää pois, mutta hienostelut ja hiomiset pitäisi jättää siihen vaiheeseen kun peli on saatu edes pelattavaan kuntoon. Esimerkiksi enimmäkseen graafiset ominaisuudet, kuten Robon kasaaminen erillisistä yhteensopivista palasista jokaisen pelikerran aluksi, tuottivat paljon vaivaa siihen nähden, miten pieni kyseisen ominaisuuden vaikutus olisi ollut lopulliseen peliin. Ominaisuus oli siksi hankala, että se piti oikeastaan olla pelissä mukana alusta lähtien, jos sitä haluttiin ottaa mukaan ollenkaan. Tiimi työskenteli päättäväisesti pitääkseen sen mukana ja päähahmon luominen peliin monimutkaistui huomattavasti.

Tarkemmat aikataulut ja määrä-ajat olisivat voineet helpottaa liian pitkien osuuksien karsimista. Kokonaisuus on helpompi ajatella ja ideoiden kokoa rajoittaa, kun työt jaetaan pienempiin ja hallittavampiin osiin. Tarkkojen aikataulujen tekeminen hylättiin enimmäkseen projektin edetessä, koska töihin tarvittavan ajan arviointi oli vaikeaa ja aikataulujen miettiminen pakotti miettimään liian pitkälle tulevaisuuteen. Työskentely haluttiin pitää suhteellisen

vapaana. Tämä oli virhe, sillä tiimin jäsenillä oli tapana jäädä hiomaan eri työvaiheita liian pitkäksi aikaa. Tiukempi aikataulu olisi pakottanut etenemään ja estänyt hidastelut.

Pelisuunnittelun näkökulmasta vaikeuksia tuotti testaamisen mahdottomuus. Projektin ensimmäisinä päivinä käytettiin jonkin aikaa Robodashin prototyyppi-version valmistamiseen ActionScript-ohjelmointikieltä käyttäen, jotta voitaisiin testata peli-idean toimivuutta ilman että se täytyisi todeta vasta kuukausien päästä. Prototyyppiä olisi myös voinut käyttää pelattavuuden testaamiseen ja hiomiseen samalla kun ohjelmoijat työskentelisivät oikean pelin parissa. Jopa yksinkertaisen prototyypin valmistaminen alkoi viedä viikkoja ja se alkoi siinä vaiheessa projektia käyttää liikaa resursseja. Prototyypin koodi ei koodikielestä johtuen edes sisältänyt uudestaan käytettävää koodia, joten sen kehittäminen oli pakko lopettaa. Ideoiden ja suunniteltujen ominaisuuksien toimivuuteen piti monesti vain luottaa sen mukaan, miten hyvältä ne paperilla näyttivät.

Pelisuunnittelijan tärkeimpiin ominaisuuksiin kuuluvat hyvät kommunikointikyvyt. Tähän puoleen kiinnitettiin projektin aikana kuitenkin liian vähän huomiota. Vohwelin jäsenet olivat kaikki tuttuja toisilleen, mutta ajatusten välittäminen pelisuunnittelusta sisällön tuottajille osoittautui odotettua haastavammaksi. Pelisuunnitteludokumenttina käytetty wiki-sivusto oli helppo käyttää, mutta se ei riittänyt. Peliprojektissa ei voi luottaa kaikkien ryhmän jäsenien pystyvän koko ajan pysymään ajan tasalla dokumenttien muutoksista ja pelin sisällöstä. Heillä on yleensä paljon työtä muutenkin. Pelisuunnitteludokumenttia täytyykin pääasiassa käyttää pelisuunnittelijan apuvälineenä ja eräänlaisena muistilistana. Pelisuunnittelijan vastuulla on kommunikoida pelin syvin olemus muille tiimin jäsenille ja erikseen selittää jokainen ohjelmitava tai graafinen elementti tarvittaessa. Pelin elementit täytyy myös osata oikeuttaa, sillä luotava peli on kaikille tekijöille tärkeä ja erimielisyyksiä sen suunnasta voi syntyä. Kaikki muut ideat hylkäävä pelisuunnittelija voi suuresti laskea muun ryhmän motivaatiota.

Vaikka pääpelisuunnittelija on vastuussa pelin suunnasta, tuovat kaikki projektiin kuuluvat oman osansa pelisuunnitteluun. Esimerkiksi peligrafiikka ei voi mitenkään olla täysin samanaista kuin millaiseksi pelisuunnittelija sen kuvittelee. Tämä ei ole huono asia. Kaikki erilaiset näkökulmat voivat rikastaa peliä, joten pelisuunnittelijan on osattava myös todella kuunnella muita ihmisiä. Tässä peliprojektissa opittiin, miten vaikeaa voi olla tarkastella omia ideoita puolueettomasti. Pelisuunnittelijan täytyy voida tarkastella asioita monesta kuvakulmasta voidakseen tunnistaa hyvät ideat huonoista. Tämä on hankalaa, jos muita tiimin jäseniä ja ulkopuolisia testaajia ei kuunnella tarkasti.

Ryhmä oli motivoitunut, mutta projektin vastoinkäymiset horjuttivat uskoa ja innostus luonnollisesti laantui hieman myöhemmissä vaiheissa. Proseduraalinen kenttägenerointi aiheutti vaikeuksia ohjelmoijille ja se täytyi projektin puolivälissä ohjelmoida kokonaan uudestaan. Myös pelimoottori täytyi koota osittain uudestaan yhdessä vaiheessa projektia, sillä siinä aluksi käytettyä Box2D-fysiikkakirjastoa oli lopulta vaikea soveltaa Robodashin tarpeisiin. Pelimoottorin ohjelmointi oli jo viety pitkälle, kun Box2D:n ongelmat todettiin. Fysiikkakirjaston poistaminen oli raskas ratkaisu. Tällaisille ongelmille ei aina voi mitään, mutta tässä tapauksessa koko tiimin olisi kaikista kiireistä huolimatta pitänyt olla tiiviimmin mukana pelimoottorin suunnitteluvaiheessa. Joskus on hankala keskittyä omaan osa-alueeseen ja samalla nähdä koko projektin kokonaiskuva, mutta se voi helpottua pitämällä usein pieniä kokouksia tiimin kesken.

Projekti opetti paljon kaikille osallisille ja se antoi mahdollisuuden soveltaa vuosien varrella oppimaa teoriataustaa käytännöntyöhön. Robodashia varten tehty tutkimustyökään ei mennyt hukkaan, sillä pelisuunnittelun teorian ja käytännön esimerkit voivat jatkossakin helpottaa suunnitteluprosessia ja ruokkia luovuutta. Kaiken lisäksi pelistä saatiin jonkinlainen pohjaversio valmiiksi. Vohweli voi jatkossa palata Robodashin pariin jossain muodossa, ilman opinnäytetyöprosessin mukanaan tuomia rajoituksia. Pelin voi ajatella olevan tauolla, mikä antaa mahdollisuuden miettiä sen alkuperäisen suunnittelun toimivuutta ja tarkastella sitä projektin ulkopuolelta. Pelin virheille ja ongelmille voi sokaistua ja pieni tauko antaa mahdollisuuden katsoa projektia uusin silmin.

Robodash toimisi muutenkin paremmin normaalimpana projektina. Opinnäytetyöprosessi toi mukanaan monia rajoituksia ja ylimääräisiä töitä. Varsinkin ohjelmoijilla oli vaikeaa tasapainotella opinnäytetyöaiheidensa ja muun ohjelmoinnin välillä. Pelattavuuden ohjelmoinnin jakaminen ohjelmoijien kesken ei lopulta toiminut kovin hyvin, sillä se vaikeutti keskittymistä. Kaikki ohjelmoinnin osa-alueet kärsivät tämän takia. Myöhemmin yritettiin saada projektiin mukaan ulkopuolinen ohjelmoija, joka olisi voinut keskittyä pelkästään ohjelmointiin. Häntä ei kuitenkaan saatu projektiin mukaan tarpeeksi aikaisessa vaiheessa. Koodin sekavuus vaikeutti sen seuraamista ja ulkopuolisen ohjelmoijan oli vaikea tulla mukaan kesken projektin. Peli olisi saatu paljon valmiimpaan kuntoon, jos kaikki ohjelmoijat olisivat olleet mukana alusta asti. Toisaalta, näin montaa ohjelmoijaa ei olisi välttämättä tarvittu, jos opinnäytetyöprosessi ei olisi määrännyt tiettyjä projektin piirteitä. Jos projekti aloitetaan uudestaan, voidaan näiltä ongelmilta välttyä.

Pelisuunnittelu on monimutkainen ala. Hyvän pelisuunnittelijan tulisi osata kirjoittaa hyvin, olla luova, ymmärtää mediaansa ja kaikkia siihen liittyviä medioita, olla hyvä kommunikoi-
maan ajatuksiaan muille, ymmärtää matematiikkaa, ymmärtää psykologiaa ja paljon muuta.
Robodash opetti tekijöilleen paljon peliprojekteista ja pelisuunnittelun todellisesta luonteesta.
Teoriapohjan tunteminen on tärkeää, mutta sitä täytyy myös osata soveltaa. Paras tapa oppia
pelisuunnittelua, on kokea se.

LÄHTEET

|tsr's NES archive 2011. The Tetris saga.

<http://www.atarihq.com/tsr/special/tetrishist.html>

(luettu 21.1.2011)

Adams, E. 2011. The High Concept Document.

<http://www.csc.kth.se/utbildning/kth/kurser/DH2640/grip08/HighConceptTemplate-Inl4.pdf>

(luettu 13.4.2011)

Arcade History 2011. Database.

<http://www.arcade-history.com/index.php?page=database>

(luettu 13.4.2011)

atariarchives.com 1984. Chapter 10: Game Design Theory.

<http://www.atariarchives.org/agagd/chapter10.php>

(luettu 17.4.2011)

Barton, M. 2007. The History of Computer Role-Playing Games Part 1: The Early Years (1980-1983)

http://www.gamasutra.com/features/20070223a/barton_01.shtml

(luettu 13.4.2011)

BetterExplained 2007. A Visual Guide to Version Control.

<http://betterexplained.com/articles/a-visual-guide-to-version-control/>

(luettu 13.4.2011)

Bishop, J. 2011. Analysis: A Wizard Did It – Internal Consistency In Video Games.

http://www.gamasutra.com/view/news/33189/Analysis_A_Wizard_Did_It__Internal_Consistency_In_Video_Games.php

(luettu 19.4.2011.)

Björk, S. & Holopainen, J. 2005. Patterns in Game Design. Charles River Media.

Black Isle Studios 1997. Fallout.

Bowen Research 2011. Can Videogames Make You Cry?

<http://www.bowenresearch.com/studies.php?id=3>

(luettu 17.2.2011)

Brathwaite, B. & Schrieber, I. 2008. Challenges for Game Designers. Course Technology.

Capcom 1988. Mega Man 2.

Capcom 1990. Mega Man 3.

Capcom 2008. Mega Man 9.

Chandler, R. 2005. Soapbox: Active Storytelling in Games.

[http://www.gamasutra.com/view/feature/2343/soapbox_active_storytelling_in_.ph](http://www.gamasutra.com/view/feature/2343/soapbox_active_storytelling_in_.php)

p

(luettu 13.4.2011)

Chavis, J. 2011. The History of Video Game Designing.

http://www.mania.com/history-video-game-designing_article_116390.html

(luettu 13.4.2011)

Compton, K. & Mateas, M. 2006. Procedural Level Design for Platform Games.

<http://www.aaai.org/Papers/AIIDE/2006/AIIDE06-022.pdf>

(luettu 17.4.2011)

Cyberpunkreview.com 2006. Cyberpunk Games.

<http://www.cyberpunkreview.com/cyberpunk-games/>

(luettu 16.5.2011)

Double Fine 2005. Psychonauts.

Double Fine 2009. Brütal Legend.

Enormous Elk 2010. Unreal World RPG.

Enzer, M. 2008. Glossary of Internet Terms.

<http://www.matisse.net/files/glossary.html#B>

(luettu 18.4.2011.)

Epic Games 2006. Gears of War.

Flogging the Quill 2006. How to show, when to tell.

http://www.floggingthequill.com/flogging_the_quill/2006/week6/index.html

(luettu 16.4.2011)

Game Career Guide 2008. What is a Game Engine?

http://www.gamecareerguide.com/features/529/what_is_a_game_.php

(luettu 13.4.2011)

Gard, T. 2010. Action Adventure Level Design, Part 1.

http://www.gamasutra.com/view/feature/4326/action_adventure_level_design_.php

(luettu 13.4.2011)

Giant Bomb 2011. Context Sensitive.

<http://www.giantbomb.com/context-sensitive/92-190/>

(luettu 17.4.2011)

Giant Bomb 2011. Health Pack.

<http://www.giantbomb.com/health-pack/93-255/>

(luettu 17.4.2011)

Giant Bomb 2011. Lives.

<http://www.giantbomb.com/lives/92-323/>

(luettu 17.4.2011)

Giant Bomb 2011. Metroidvania.

<http://www.giantbomb.com/metroidvania/92-2440/>

(luettu 13.4.2011)

Giant Bomb 2011. Quick Time Event

<http://www.giantbomb.com/quick-time-event/92-6/>

(luettu 13.4.2011)

Gilbert, R. 2007. Maniac Mansion in 9.

<http://grumpygamer.com/8139425>

(luettu 13.4.2011)

Götterdämmerung.org 1990. Nuclear Deterrence Theory: The Search for Credibility.

<http://www.gotterdammerung.org/books/reviews/n/nuclear-deterrence-theory.html>

(luettu 17.4.2011)

Hietala, O. 2010. Opinnäytetyösuunnitelma: Simple DirectMedia Layer.

jpb.com 2006. The Step by Step Guide to Brainstorming.

<http://www.jpb.com/creative/brainstorming.php>

(luettu 12.1.2011)

Kempainen, S. 2010. Opinnäytetyösuunnitelma: Pelisuunnittelu ja projektin johtaminen.

Kheper 2003. What is Cyberpunk?

http://www.kheper.net/topics/cyberpunk/what_is_cyberpunk.htm

(luettu 13.4.2011)

Konami 2008. Hideo Kojima Interview.

<http://www.konami.jp/mgs4/uk/interview/03.html>

(luettu 13.4.2011)

Kookas 2011. Narratiivi.

<http://www.kookas.fi/articles/read/218>

(luettu 26.4. 2011)

Lucasarts 1993. Day of the Tentacle.

Lucasarts 1995. Full Throttle.

Lucasarts 1998. Grim Fandango.

Lucasfilm Games 1987. Maniac Mansion.

Make It Big in Games 2011. How to Pitch Your Game.

<http://makeitbigingames.com/2007/09/how-to-pitch-your-game/>

(luettu 13.4.2011)

Merriam-Webster 2011. mock-up.

<http://www.merriam-webster.com/dictionary/mock-up>

(luettu 16.4.2011)

McCarthy, J. 2007. What is artificial intelligence?

<http://www-formal.stanford.edu/jmc/whatisai/>

(luettu 18.4.2011.)

McDevitt, D. 2010. A Practical Guide to Game Writing.

http://www.gamasutra.com/view/feature/6171/a_practical_guide_to_game_writing

.php? print=1

(luettu 10.2.2011)

Moby Games 2011. Post-Apocalyptic.

<http://www.mobygames.com/genre/sheet/post-apocalyptic/>
(luettu 16.5.2011)

msdn 2011. What is a Sprite?

<http://msdn.microsoft.com/en-us/library/bb203919.aspx>
(luettu 13.4.2011)

New World Computing 1999. Heroes of Might and Magic III.

Nintendo 1985. Super Mario Bros.

Obsidian Entertainment 2010. Fallout: New Vegas.

Open Dictionary 2011. Nintendo Entertainment System.

http://open-dictionary.com/Nintendo_Entertainment_System
(luettu 14.4.2011)

Oxford Advanced Learner's Dictionary 2011.

<http://www.oxfordadvancedlearnersdictionary.com/dictionary/platform-game>
(luettu 18.4.2011)

Pedersen, R. E. 2009. Game Design Foundations. Wordware Publishing.

Procedural Content Generation Wiki 2011. Main Page.

<http://pcg.wikidot.com/>
(luettu 13.4.2011)

Räsänen, T. 2010. Opinnäytetyösuunnitelma: Proseduraalinen sisällön luonti peleissä.

Schell, J. 2008. The Art of Game Design. Morgan Kaufmann Publishers.

Science Fiction Citations 2008. Steampunk.

<http://www.jessesword.com/sf/view/327>

(luettu 13.4.2011)

Sonichurricane.com 2011. SF2 Engine Randomness Rundown.

<http://sonichurricane.com/articles/sf2randomness.html>

(luettu 22.4.2011.)

SourceForge 2010. About, SourceForge.net: Project vohweli.

<http://sourceforge.net/about>

<http://sourceforge.net/apps/wordpress/vohweli/>

(luettu 13.1.2011)

Square 1997. Final Fantasy VII.

Tang, C. 2005. How To Write a Backstory Without Putting Your Reader To Sleep.

<http://storysensei.blogspot.com/2005/08/how-to-write-backstory-without-putting.html>

(luettu 19.4.2011)

Team Ico 2001. ICO.

Temple of the Roguelike 2007. Roguelike Definition 2.0.

<http://www.roguetemple.com/roguelike-definition/>

(luettu 13.4.2011)

The Escapist 2010. Extra Credits: Video Games and Bad Writing.

<http://www.escapistmagazine.com/videos/view/extra-credits/1887-Video-Games-Bad-Writing>

(katsottu 16.4.2011)

The Escapist 2010. Extra Credits: True Female Characters.

<http://www.escapistmagazine.com/videos/view/extra-credits/2868-True-Female-Characters>

(katsottu 16.4.2011)

Valve 1998. Half Life.

Valve 2004. Half Life 2.

Webopedia 2011. Arcade Game.

http://www.webopedia.com/TERM/A/arcade_game.html

(luettu 18.4.2011.)

Webopedia 2011. Context Sensitive.

http://www.webopedia.com/TERM/C/context_sensitive.html

(luettu 13.4.2011)

Webopedia 2011. User Interface.

http://www.webopedia.com/TERM/U/user_interface.html

(luettu 13.4.2011.)

Welch, J. 2008. Casual Games Summit 2008 Keynote.

Wiik, M. 2010. Producing Game Graphics: Project Plan.

wiki.org 2002. What Is Wiki

<http://www.wiki.org/wiki.cgi?WhatIsWiki>

(luettu 13.4.2011)

Wiktionary 2009. Pre-Alpha Version.

http://en.wiktionary.org/wiki/pre-alpha_version

(luettu 13.4.2011)

Wilson, G. 2006. Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design

http://www.gamasutra.com/view/feature/2538/off_with_their_huds_rethinking_hp
(luettu 13.4.2011)

Kuvat

101 Video Games That Made My Life Slightly Better 2009. #14. Shenmue.

<http://101videogames.wordpress.com/2009/04/20/14-shenmue/>

brej.org 2010. A couple of cute Linux games.

<http://brej.org/blog/?p=320>

cuhkacs.org 2007. [TV GAME] XBOX360購入篇（下）

<http://www.cuhkacs.org/~henryporter/blog/read.php/526.htm>

Gard, T. 2010. Action Adventure Level Design, Part 1.

http://www.gamasutra.com/view/feature/4326/action_adventure_level_design_php?page=3

Giant Bomb 2008. ICO.

http://www.giantbomb.com/ico/61-9638/all-images/52-181665/ico__10_/51-482670/

IGN.com 2007. IGN: Command & Conquer 3 Tiberium Wars Screenshots (PC).

<http://pc.ign.com/dor/objects/823989/command--conquer-3-tiberium-wars/images/command-conquer-3-tiberium-wars-20070316081951167.htm>

Inglorious on Video game UI & UX 2009. Brutal Legend tackles the No HUD challenge.

<http://inglorio.us/2009/10/21/brutal-legend-tackles-the-no-hud-challenge/>

Ingrid Richter 2011. Anime Breakdown: Battle Angel Alita (Gunnm).

http://www.ingridrichter.org/anime/battle_angel_alita.html

Learning Game Design with Arcade Berg 2010. "Outside the game"-expectations.

<http://arcadeberg.com/causerie/outside-the-game-expectations/>

Project Vohweli 2010. Exclusive Bullshots!

<http://sourceforge.net/apps/wordpress/vohweli/2010/11/19/exclusive-bullshots/>

Project Vohweli 2010. MockUpdate.

<http://sourceforge.net/apps/wordpress/vohweli/2010/10/22/mock-update/>

Project Vohweli 2010. Sketches!

<http://sourceforge.net/apps/wordpress/vohweli/2010/09/10/sketches/>

The Maniac Mansion fansite 2007. Maniac Mansion Screenshots.

<http://members.fortunecity.com/harang/screenshots.html>

UnReal World RPG Official Homepage 2011.

<http://www.jmp.fi/~smaarane/urw.html>

vgmuseum.com 2011. Ending for Tetris-B-Type(NES). Screenshots.

http://www.vgmuseum.com/end/nes/a/tetris_2.htm

vgmuseum.com 2011. NES Super Mario Bros.

<http://www.vgmuseum.com/images/nes/01/mario1.html>

vgmuseum.com 2011. SNES: Mega Man 7.

<http://www.vgmuseum.com/images/snes/01/megaman7.html>

LIITTEIDEN LUEITTELO

LIITE 1: ROBODASH HIGH CONCEPT DOCUMENT

ROBODASH HIGH CONCEPT DOCUMENT

Robodash

Sasu Kemppainen

High Concept

A mobile laboratory owned by an extremely strange scientist makes its way through an uncharted and possibly uninhabited planet. The planet's surface is littered with broken robots and junk thrown out by the scientist who frantically puts together new robots constantly only to throw them away in the end. Sometimes the robots aren't happy with their part and set out to prove their worth.

Features

- An infinite sidescrolling action game. The player works as a little robot's AI and makes his way through the wasteland as far as he can. Getting far unlocks new features in the game and records the player's achievements on an online leaderboard.
- The world, the Bot, the enemies, the names and the events are all procedurally generated, making each playthrough a unique experience and giving infinite amounts of variation.
- Powerups replace the Bot's limbs with weapons and items that the player can use pressing buttons assigned to each of them.
- No loading screens, no stops, a massive world where the player can travel as long as the Bot stays alive.
- Destructible environments that allow for alternative methods of getting through obstacles.
- Harder boss enemies appear from time to time. Fight or flight? The rewards make them worth fighting for though.

Player Motivation

While failure is the only option in an endless platformer, the player sets out to prove the worth of the little robot. The game keeps count on how far the robot manages to travel and going far enough unlocks prizes. The results of each game are also recorded on an online leaderboard.

Genre

Endless Tactical Arcade-Action.

Target Customer

Platforming fans, retro buffs, casual gamers.

Competition

Canabalt, Infinite Mario Bros

Unique Selling Points

- Truly endless non-interrupted gameworld.
- Move at your own pace: proceed stealthily or rip them all down.
-

Target Hardware

PC

Design Goals

- **Arcade Style:** Inspired by old fashioned arcade games, it's basically a survival mission. The player just tries to get as far as possible in the infinite world.
- **Atmospheric:** The graphics tell the story of an uncharted alien planet and makes the exploring interesting.
- **Simple:** Easy to get in and learn the basic mechanics.
- **Variable:** The variation offered by the procedural design and many powerups, enemies and objects makes the game a deep experience.

Characters

The character roster of the game consists of the Robot and all the enemies in the world. The Scientist is a story-only character that appears during the short intro and the ending.

The Bot: Little guy built of ill-matching spare parts and all kinds of junk. Determined to show it can make it in the world. Can upgrade itself easily with junk found during its travels.

The Scientist: Might be a bit off in the head. Decided to roam the planet to get an inspiration on the best robot design ever. Nothing is ever good enough though and all the half-finished concepts end up in the trash bin.

The Enemies

Rush: A running robot. Always training to develop its running skills. It doesn't actually mean to hurt the Bot but inevitably does so while running around in high speeds. The Bot loses energy and gets bumped backwards if hit by Rush.

Deter: A peace-loving robot specimen. It does, however, believe in the deterrence theory and it will launch a destructive nuke if attacked straight on.

Modrill: A drill robot that drills through the earth like a fish swims through water. Sometimes jumps on the surface damaging anyone getting in the way.
...etc...