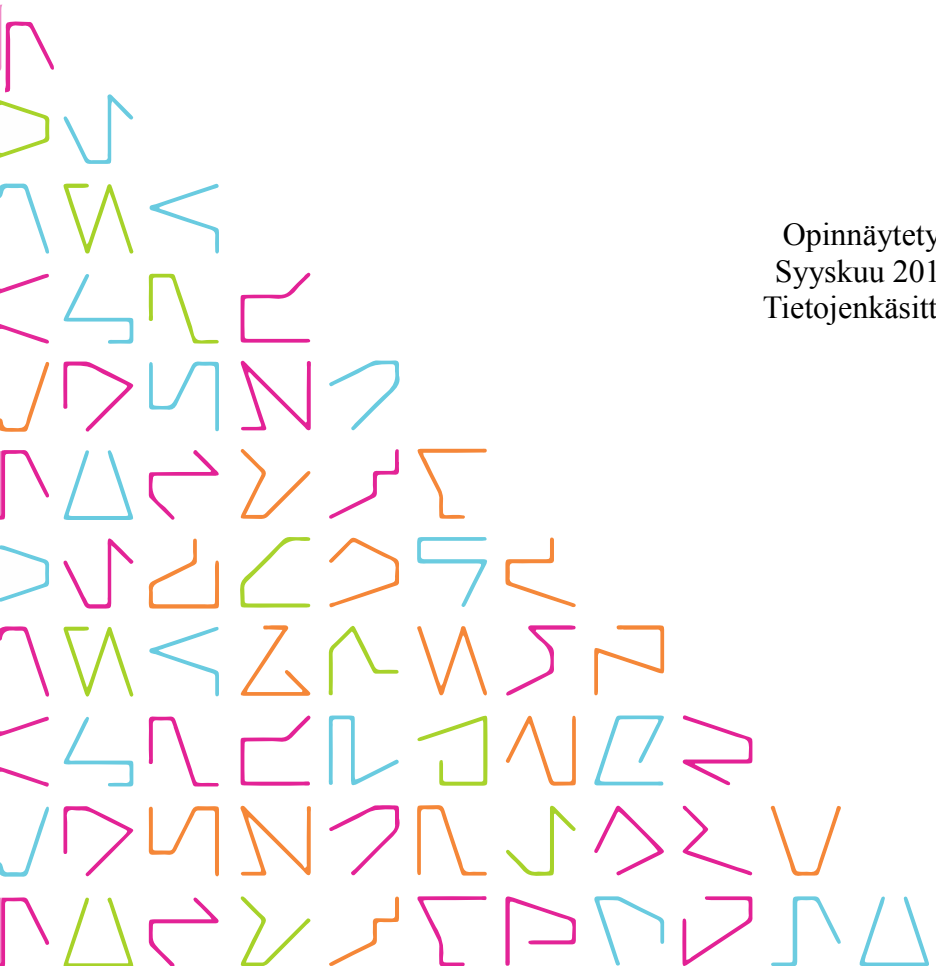
TAMPEREEN
AMMATTIKORKEAKOULU

HTML5-RETROPELIN KEHITYS CONSTRUCT 2 -SOVELLUKSELLA

Jukka Mattila



Opinnäytetyö
Syyskuu 2017
Tietojenkäsittely

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely

MATTILA JUKKA:
HTML5-Retropelinkehitys Construct 2 - sovelluksella

Opinnäytetyö 68 sivua
Syyskuu 2017

Tämän opinnäytetyön tavoitteena oli käsitellä Construct 2 toimivuutta HTML5-retropelien kehityksessä. Opinnäytetyö selvittää HTML5- merkkikielen perusteet pelintekijän näkökulmasta. Lisäksi opinnäytetyö käy läpi erilaisia HTML5-pelinkehitystekniikoita sekä työkaluja.

Pelinkehitys on Suomessa nousussa ja tarjoaa paljon mahdollisuuksia myös pienille pelinkehittäjille tuottaa pelejä massoille. Opinnäytetyö opastaa aloittelevalle pelintekijälle retropelien käsitteet. Se sisältää itseopiskelumateriaalia, jonka avulla käyttäjä oppii Construct 2 -työkalun HTML5-pelinkehityksen alkeet. Sen avulla käyttäjä voi luoda yksinkertaisen Construct 2 -pelin.

Tutkimuksen näkökulmasta, opinnäytetyön tarkoitus oli selvittää, onko järjestelmällä mahdollista tuottaa valmis peli omatoimisesti, joten opinnäytetyössä käydään läpi kokonainen, itse tuotettu retropeli. Peli on toteutettu Construct 2 - sovelluksella ja julkaistu verkkoselaimella toimivaksi. Kaikki pelin sisältö on tuotettu itse.

Opinnäytetyö näyttää toteen, että järjestelmä soveltuu ainakin pienimuotoiseen pelinkehitykseen, tai niinsanottuun indie-pelinkehitykseen. Opinnäytetyö ei selvitä pelinkehitystä taloudellisesta näkökulmasta vaan keskittyy pelinkehityksen filosofiaan sekä itseopiskelutaidon kehittämiseen.

Asiasanat: retro, peli, HTML5, pelinkehitys, pelijulkaisu

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Bachelor of Business Information Technology

MATTILA JUKKA:
HTML5 Retro Game Development using Construct 2

Bachelor's thesis 68 pages.
September 2017

The purpose of this thesis was to review the functionality of Construct 2 in HTML5 retro game development.

A guide for beginner game makers on the concepts associated with retro games is provided here.

Furthermore, self-study material is offered that helps the user learn the basic elements of HTML5 game development using the Construct 2 application. By following this thesis, user can create a simple game using Construct 2.

This thesis investigates if it is possible to self produce a finished game with the system.

The conclusion section states that, it is possible to self-produce HTML5 games with the tools and solutions presented, and that the Construct 2 is suitable for at least small-scale game development, or so-called indie game development.

Keywords: retro, game, HTML5, development, publishing

SISÄLLYSLUETTELO

1 Johdanto 5

2 HTML5 pelinkehityksessä 7

2.1 HTML5 7

2.2 Kehitystyökalut ja pelimoottorit 8

2.3 Yhteensopivat selaimet 10

3 Retropeliestetiikka 11

3.1 Retropeligrafiikka 12

3.2 Retropeliäänitehosteet 13

3.3 Retropelimusiikki 14

3.4 Retropelien teemat 15

4 Ensikertalaisena pelinkehityksen maailmaan 16

4.1 Aloita pienestä 17

4.2 Löydä sisäinen pelintekijäsi 18

5 Pelinkehitys Construct 2 – sovelluksella 19

5.1 Ideasta toteutukseen 19

5.2 Tavoite ja tarkoitus 20

5.3 Peliobjektien luominen 20

5.4 Vaarat ja vihollliset 28

5.5 Kerättävät bonukset 30

5.6 Pelilogiikka 31

5.7 Tasojen kehittäminen 42

5.8 Apua ongelmiin 42

5.9 Pelijulkaisun oikeudellisia kysymyksiä 42

5.10 Construct 2 pelin julkaiseminen yleisölle 44

6 Retropeli ”Contradiction” 45

6.1 Pelin esittely 47

6.1.1 Teema 47

6.1.2 Tarina 47

6.1.3 Sankarimme Billy Lancer 48

6.1.4 Kerättävät Bonukset 48

6.1.5 Käyttöliittymä 51

6.1.6 Vihollliset 52

6.1.7 Toiminnallisia elementtejä 55

6.1.8 Radat 57

6.1.9 Pomot 59

6.1.10 Musiikki 60

6.1.11 Äänitehosteet 62

6.1.12 Retrohenkinen salasana-systeemi 63

6.2 Pelinkehitysprosessi 63

6.3 Julkaisu, palaute, jälkitoimet 65

7 Pohdinta 66

Lähteet 67

1 JOHDANTO

Peliteollisuus on nykypäivän Suomessa kovassa nosteessa, ja alasta on povattu jopa ”uutta Nokiaa”. Supercell: in ja Rovion suurmenestys on synnyttänyt alan ympärille kovasti puhetta ja hypeä. (Wargh 2014, 5) Peliala on haastava, ja erottuminen työmarkkinoilla vaatii osaamisen näyttöä.

Oma osaamiseni on muodostunut vuosien varrella. Aloitin vaatimattomasti basic koodilla kikkailusta, jonka jälkeen ostin suomalaisesta kirjakaupasta The Games Factory – sovelluksen. Tämä pelitehtaaksi nimetty ohjelmisto oli kehittyneempi versio Click 'N Create ohjelmistosta, jolla oli mahdollisuus luoda erilaisia sovelluksia ja pelejä, ilman varsinaista ohjelmointikielen tuntemusta. Toiminnot muodostettiin visuaalispainotteisessa ohjelmointiympäristössä.

Ensimmäinen pelini oli sekoitus Arcanoidia ja Space Invaders: ia. Hajotettavien palikoiden sijasta mailalla ja pallolla tuhottiinkin avaruusaluksia, jotka vyöryivät yläruudusta kohti pelaajaa. Tämä oli avartava kokemus, ja pelinteon maailma alkoi kiehtomaan välittömästi. Olin täysin häkeltynyt siitä, että osasin tehdä pelin edellyttämät toiminnot. Vaikka varsinaista koodia en kirjoittanut, aloin kuitenkin hahmottamaan peliohjelmoinnin loogista puolta. Nykyään tämä taito kulkee mukana yhtä rakkaana ja tärkeänä kun piirtäminen, laulaminen tai soittaminen.

Koenkin, että opin parhaiten kokemalla asiat itse. Olisi hienoa kuitenkin jos matkan varrella olisi ollut enemmän tällaista omatoimista lähestymistapaa tukevaa kirjallisuutta. Suuri osa opuksista menee turhan suoraan asiaan. Alkaa välitön koodaaminen, ja älytön työmäärä kasaantuu, ja lukija luovuttaa samalla tuumien, että ehkäpä pelkkä pelien pelaaminen riittää. Kuka tahansa voi olla osa pelituotantoprosessia. Onkin tärkeintä, että löydät itsestäsi ne piirteet, jotka koet tuovan prosessiin jotain oleellista, ja saat tuotua ne esiin.

Mobiilipeliala on puhjennut kukkaan, ja tarjoaa ennen näkemättömiä mahdollisuuksia myös pienille pelinkehittäjille tuottaa pelejä massoille. Julkaisijoista riippumattomille indie-pelinkehittäjille on avautunut mahdollisuuksia toteuttaa omia ideoitaan helpommin kuin koskaan ennen. Tämä ei silti tarkoita helppoa tietä onneen ja menestykseen, vaan kovaa työtä ja omistautuneisuutta. (Wargh 2014, 7)

Lahti (2014, 38) toteaa että pienelläkin ryhmällä on mahdollista nykypäivänä toteuttaa pelejä. Saatavilla on monia työkaluja, joiden avulla pelien toteuttaminen on varsin helppoa, kunhan ryhmästä löytyy jonkinlaista kokemusta ohjelmoinnista, grafiikan teosta ja äänien toteuttamisesta.

Tässä opinnäytetyössä lähestyn pelituotantoprosessin vaiheita luovuutta kehittävästä näkökulmasta, otan kantaa siihen, miten asiaa vähemmän tunteva voi lähestyä asioita omatoimisesti, hakea tarvitsemansa informaation, ja oppia oman tapansa oppia. Peliteknisissä esimerkeissä olen valinnut teemaksi perinteisen kaksiulotteisen HTML5-retropelin, sillä tällaisista vanhanajan pelien replikoinnista alkoi oma polkuni. Se onkin siis se teema, josta minulla on eniten tarjottavaa. Lisäksi retropelit ovat aina suosiossa eikä tulevaisuuden horisontissa näy pistettä retopelien loppumiselle. Lisäksi se kaikki mitä oppii retropelin laatimisesta on kuitenkin hyödyllistä modernimpien pelienkin kehittämisessä.

Pelinkehitysalustana käytän erittäin suosittua, Scirran Construct 2:sta, joka on modernisoitu The Games Factorin kaltainen järjestelmä. Se tarjoaa joustavan tapauspohjaisen toimintojenmääritysjärjestelmän. Tästä johtuen oppaani ei edellytä minkään ohjelmointikielen opiskelua, vaan antaa perusedellytykset pelinkehityksessä alkuun pääsemiselle. Lisäksi ohjelmiston perusversio sisältää kaikki teeman mukaisten pelin kehittämiseen vaadittavat työkalut, ja siitä on saatavilla ilmainen versio hieman rajatuilla toiminnoilla. Construct 2 tuottaa HTML5-pohjaisia sovelluksia, jotka voidaan julkaista useammille moderneille julkaisualustoille, kuten Windows, Andoid, iOS, Linux jne. Lisäksi oleellinen ominaisuus HTML5-pelissä on niiden toimiminen suoraan internet selaimella. (Gullen 2011, 1)
Opinnäytetyön tavoitteena on näyttää toteen, että esitellyillä työkaluilla ja metodeilla voi tuottaa omatoimisesti kokonaisen HTML5-retropelin. Täten opinnäytetyön viimeisessä osassa esittelen sen yhteydessä julkaistavan valmiin HTML5-retropelini, ja kuvaan siihen liittyvän prosessin eri vaiheita ja haasteita. Esittelen myös pelin eri ominaisuudet.

Vaikka olen tehnyt paljon pelejä, on valmiin pelin aikaansaaminen omatoimisesti epävarmaa ja suuren työmäärän takana. Tätä peliä on kehitetty jo pitkään ennen opinnäytetyön kirjoittamisen alkamista. Aina kun tuntuu siltä, että on päässyt lähellä valmiin määritelmää, löytääkin jotain, jota haluaa lisätä, muuttaa tai viilata. Valmista peliä ei siis oikeastaan edes ole, ellei itse päätä sen olevan valmis.

2 HTML5 PELINKEHITYKSESSÄ

2.1 HTML5

HTML5 on useimpien modernien internetselainten tukema merkkikieli, joka julkaistiin vuonna 2014. HTML5 rakentuu vähittäisen kehityksen (evoluutioon eikä kumouksen (revoluution) ajatukselle. Vanhat sivut ja vanhat HTML-rakenteet toimivat, ja sivuihin voidaan lisätä uusia piirteitä sitä mukaa kuin niitä toeutetaan selaimiin ja ne koetaan tarpeellisiksi.(Korpela 2011, 24) Uusi kieli mahdollistaa entistä monipuolisimpien web-tekniikoiden hyödyntämisen, ja erilaisten sovelluksien, jopa näyttävän 3d-grafiikan ajamista selaimella. Uusi Canvas-rajapinta mahdollistaa tärkeimmät uudet asiat peliohjelmoijan näkökulmasta.

Canvas elementti mahdollistaa yksinkertaisten muotojen piirtämisen ja bittikarttojen manipulaation. Canvas-elementtiä voisi kuvailla dynaamisena kuva-tagina, joka on oleellinen osa HTML5-pelikehittämistä. (Tossavainen 2014, 10) Vielä merkittävämpää on kuitenkin joustavampi muuttaminen. Vaikkapa viivan värin tai leveyden, tai kuvassa olevan tekstin muuttaminen onnistuu muuttamalla yhtä käskyä JavaScript-koodissa. (Korpela 2011, 200)

HTML elementtiä käytetään grafiikan piirtämiseen JavaScriptin avulla. elementti on siis ns. kehikko grafiikalle, mutta grafiikka tuotetaan käytännössä JavaScriptillä.

Esimerkki <Canvas> kehikon määrittelystä HTML5-sivustolla:

```
”<canvas id="myCanvas" width="200" height="100"></canvas> ”
```

Tämä HTML5-ohjelma luo suorakulmion muotoisen alueen, jonka nimitunniste on ”myCanvas”.

Seuraava ohjelma piirtää tähän kyseiseen alueeseen vasemmasta yläkulmasta, oikeaan alakulmaan suoran janan:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
```

(W3schools.com – HTML5 Canvas)

Taustamusiikki ja ääniefektit ovat tärkeä osa pelisuunnittelua. HTML5 sisältää musiikkia ja ääntä tukevan audio-tagin. Tämän ominaisuuden ansiosta ei tarvita

selainlaajennusta (Flash) toistamaan musiikkia tai ääniä. (Ryhänen 2015, 12)

HTML5 on kaikkien käytössä ilmaiseksi. Se toimii loistavasti eri alustoilla, niin älylaitteilla, tietokoneilla ja smart tv:illäkin. (Tossavainen 2014, 10) HTML5 sopii täten hyvin aloittelevien pelintekijöiden julkaisualustaksi sen taipuvuudestaan johtuen.

HTML5-pelejä tukevat monet ilmaiseksi pelejä ylläpitävät portaalit, kuten Newgrounds.com, itch.io, Addictinggames.com

Pelikehittäjän näkökulmasta, lähtökohtaisesti Canvas-rajapinnan sekä JavaScript-kielen tuntemus riittää eikä tekstieditoria kummempaa työkalua vaadita. Helpompia ja aloittelijaystävällisempiä vaihtoehtoja on kuitenkin olemassa.

2.2 Kehittämistyökalut ja pelimoottorit

HTML5-pelin kehittämiseen on paljon vaihtoehtoja, esimerkiksi PHASER on täysin ilmainen. Se toimii selaimessa, mutta vaatii käyttäjältä JavaScriptin tuntemusta. GameMaker on melko hintava, mutta laajan käyttäjäryhmän tuki on taattu, ja alkuun pääsee melko helposti.

Aloittelevan pelintekijän eteen tuleekin ensin vaikea valinta. Mikä on se pelinkehitys ohjelmisto, josta kannattaa aloittaa? Tarjontaa ja vaihtoehtoja on paljon, eikä yhtä ylivoimaista voittajaa ole, joten on hyvä turvautua internetin vertailusivustoihin.

Alla HTML5 Game Engines sivuston julkaisema, suosion mukaan järjestetty HTML5-pelinkehitys alustojen listalta 10 suosituinta alustaa.

Nimi	Hinta	Julkaisuvuosi	Suosio
Construct 2	Rajoitettu 0€ Personal 99,99€ Bussiness 329,99€	2014	100
ImpactJS	83,00 €	2014	80
EaseIJS	Ilmainen	2017	64
Phaser	Ilmainen	2017	63
Pixi.js	Ilmainen	2014	57
GameMaker	169,00 €	2017	53
Three.js	Ilmainen	2017	52
PlayCanvas	Ilmainen	2017	51
Turbulenz	Ilmainen	2015	48

Suosioarvo perustuu alustalla luotujen pelien määrään sekä sen sosiaalisen median suosion (tykkäyksien, tweettien jne.) perusteella. Tämä laskutoimitus on toteutettu käyttäen Web crawler bottia, joka systemaattisesti selaa www sivustoja, samalla indeksoiden dataa. (Wikipedia – web crawler)

Listan kärjessä on suosituimpana alustana Scirra:n Construct 2. Sen eri versioita on ladattu vuodesta 2014 vuoteen 2017 yli neljä miljoonaa kertaa. (Scirra, total downloads)

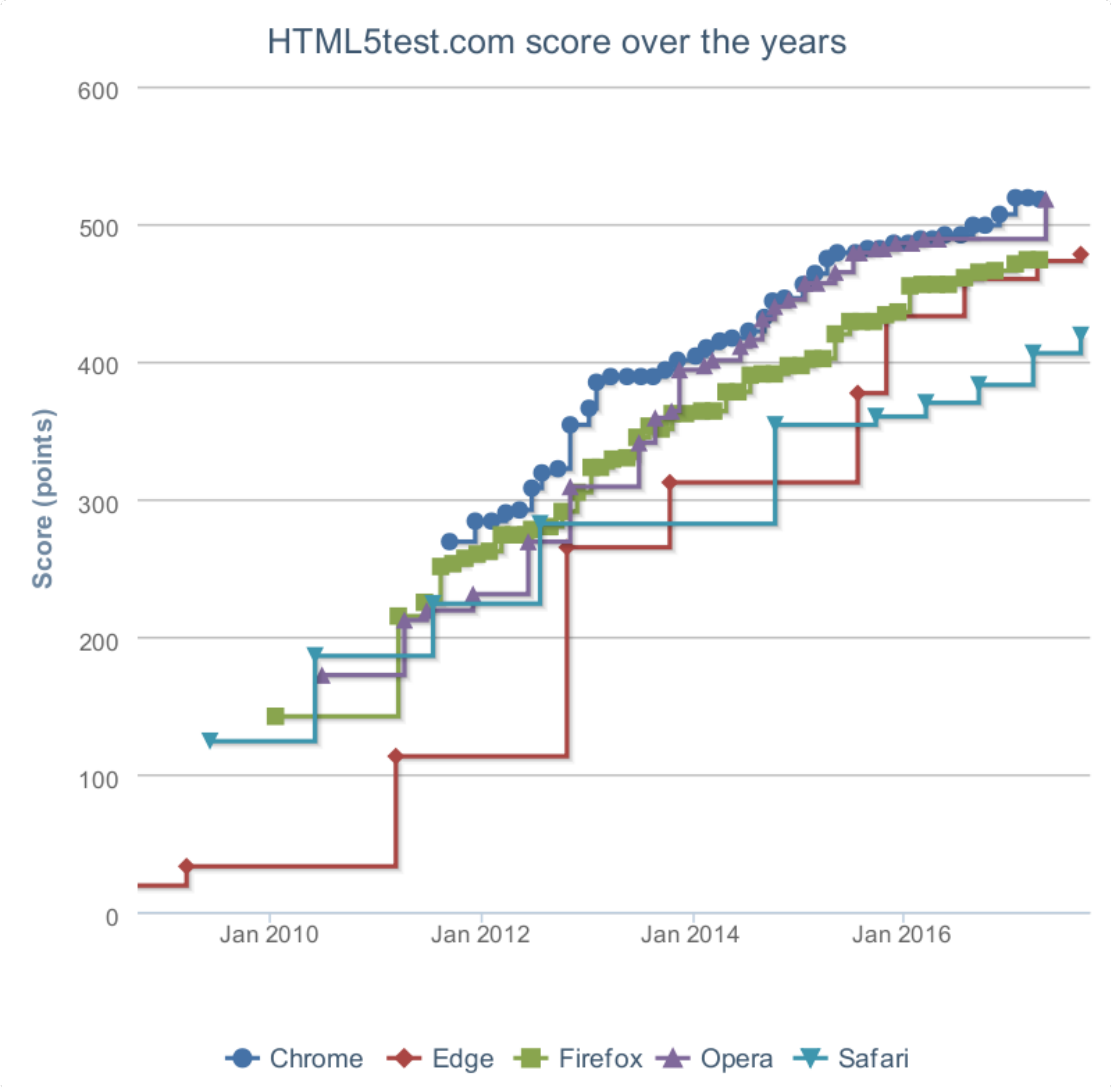
Scirra on vuonna 2011 Lontossa perustettu Gullenin veljesten yritys. Yrityksen sovelluksia ovat Construct 2:n lisäksi Construct Classic sekä beta testauksessa oleva Construct 3, joka julkaistaan pian kaikille moderneille alustoille. Selkein ero suhteessa Construct 2:een on se, että uusi tulokas toimii suoraan HTML5-selainympäristössä. Tämä mahdollistaa sujuvan pelinkehityksen suoraan internetselaimella. (Scirra.com)

Scirran sivuston (Scirra.com) mukaan Construct 2:n käyttö on mahdollista ilmaiseksi, mutta julkaiseminen ja monien ominaisuuksien käyttöä / määrää on rajoitettu. Maksullisia lisenssejä on kaksi. Personal ja Business. Personal lisenssi maksaa 99.99€ , ja business taas 329.99€. Ainoa ero lisenssien välillä on se, kuinka paljon saat sisällölläsi tienata tuloja. Personal lisenssiä käyttäen saat tienata koko elinaikanasi vain 5000$, kun taas business lisenssillä tienaamistasi ei ole rajoitettu. (Scirra.com)

Construct 2 tekee 2D-pelien teosta helppoa ihmisille riippumatta heidän taustastaan. Se sisältää täydellisen joukon tehokkaita ominaisuuksia, se tukee useita alustoja ja sovelluskauppoja, ja sillä on helposti ymmärrettävä visuaalinen ohjelmointijärjestelmä. Se on myös laajennettavissa plugin-järjestelmällä, ja yhteisö kehittää monia laajennuksia Construct 2:n laajentamiseksi (Subagio 2014, 16) Sovellus sopii perusominaisuuksiltaan hyvin esimerkkipelin kehittämiseen, sillä se sisältää kaikki oleelliset työkalut, joita oppaassani tarvitaan. Erityisen taipuisaksi Construct 2:sen tekee se, että sillä kehitetyt pelit voi julkaista suoraan www palvelimelle, tai vaikkapa omaan Google Drive tai Drop Box-palveluun. (Gullen 2011, 1)

2.3 Yhteensopivat selaimet

Lähes kaikki modernit selaimet tukevat HTML5:ttä, esimerkiksi Mozilla Firefox ja Google Chrome. Html5test.com-sivustolla voit testata eri selainten toimivuutta. Alla myös yleisemmät selaimet pisteytettynä eri HTML5 ominaisuuksien tuen mukaan. Data on muodostettu automatisoitujen testien perusteella.



3 RETROPELIESTETIIKKA

Mikä on retropeli? Se vaikuttaa helpolta kysymykseltä, mutta paljastuu yllättävän vaikeaksi vastata. Vastaaminen kysymykseen on kuitenkin tärkeää, sillä on vaikea opiskella asiaa, jota ei pysty selkeästi käsitteellistämään.

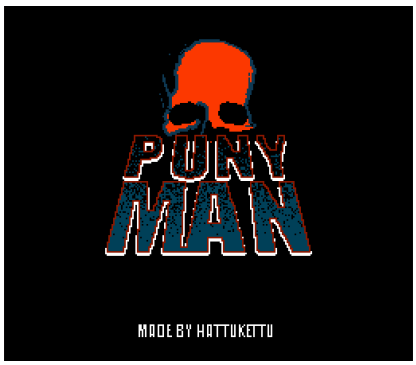
Yksi lähestymistapa onkin sanoa, että se on peli jonkun lapsuudesta, sellainen, joka herättää nostalgian tunnetta. Tämä kuvaus toimii kyllä yhdelle ihmiselle, mutta on hyvin subjektiivinen määritelmä eikä skaalaudu, sillä kaksi ihmistä ei voi jakaa täysin samaa lapsuutta, tai samoja nostalgisia tunteita. (Aycock 2016, 1)

Wikipedia ei siis tarjoa määritelmää haulle ”retropeli” vaan käsittelee asiaa termin ”retropelaaminen” kautta. Äkkiseltään ajateltuna retropelit ovat tietysti niitä peliklassikoita, joita pelattiin aina vanhoista hämyisistä arcadeluola-ajoista, aina Super Metroidin syvään henkilökohtaiseen seikkailuun saakka. Nykyään retroksi voisi mieltää kuitenkin jopa Super Mario 64:n, mutta mielestäni tässä kohtaa ollaan jo melko modernin peliajattelun tasolla. Termin määritelmä käytännössä elää jatkuvasti ja se on suhteessa elettävään aikaan ja kehityksen tasoon. Jonain päivänä vuoden 2017 pelit ovat retropelejä.

Retropelaaminen ilmiönä elää ja voi hyvin. Tampereella perustettiin tammikuussa 2017 pelimuseo, jossa pääsee itse oikein kokemaan vintage-retropelaamista oikeilla laitteilla. Suomen pelimuseo esittelee monipuolisesti suomalaista pelikulttuuria ja kertoo, kuinka digitaalinen pelaaminen Suomessa alkoi ja kehittyi vuosien saatossa. Digitaalisten pelien lisäksi museo kertoo myös lauta- ja roolipeleistä. (http://vapriikki.fi/pelimuseo/)

Retropelinkehityksestä kiinnostuneen kannattaakin tutusta näihin järjestelmiin ihan konsolikohtaisesti, ja pelimuseo onkin yksi niitä harvoja paikkoja maailmassa, jossa laitteita on saatettu julkisesti yhteen isommissa määrin.

Internet on nykyään pullollaan uusia pelejä, jotka tyyliltään jäljittelevät tätä vanhaa omalaatuista pelien kirjon tyyliä. Retropeli pelinkehityksessä tarkoittaakin siis sellaisen pelin kehittämistä, jonka teemat edustavat joitakin vanhoihin peleihin liittyviä tunnusmerkkejä. Räikeänä esimerkkinä CGA-väripalettiin rajattu värimäärä. Myös pelin toiminnot voivat tehdä siitä retron. Esimerkiksi jos täysin moderneilla grafiikoilla toteutettu peli käyttäisi ainoastaan tekstisyötteitä olisi se ainakin ominaisuuksiltaan retro. Henkilökohtaisesti ja äärimmäisen yksinkertaistetusti näkisin asian niin, että jokainen aikuiseksi kasvanut saa ehdottomasti kutsua lapsuutensa pelejä retropeleiksi.



Kuva 1 ja 2 Puny Man retropeli.

Minun tapauksessa retropeli voisi olla esimerkiksi 8-Bittisen Nintendon pelien tyyliä jäljittelevä kokonaisuus, jossa kuvapisteet on rajattu 256x224 resoluutioon. Värejä olisi käytetty rajallisesti ja äänissä olisi aitoa tunnelmaa, kuten yllä olevassa Puny Man-pelissäni.

3.1 Retropeligrafiikka

Yksiselitteistä retropelin graafista tyyliä ei ole, vaikuttavia tekijöitä ovat eri konsolien tekniset ratkaisut. Yleisesti ottaen kuitenkin kaksiulotteinen retropeligrafiikka koostuu matalasta resoluutiosta, ”spriteistä” ja ”tileistä” sekä rajoitetuista väripaleteista.

Sprite-grafiikka on etenkin vanhemmissa tietokonepeleissä käytetty tietokonegrafiikan menetelmä, joka perustuu pieniin pikseligrafiikkana toteutettuihin osittain läpinäkyviin kuvahahmoihin, spriteihin. Spriteillä on tyypillisesti toteutettu kaikenlaiset pelikentän liikkuvat hahmot, kun taas pelikenttä itse on toteutettu muilla menetelmillä. Ensimmäisenä pelinä, joka käytti spritejä, pidetään Taiton vuoden 1974 peliä Basketball, joka on myös ensimmäinen urheilupeli.
(Wikipedia – Sprite-grafiikka))

Ryhänen (2015, 23) tiivistää käsitteen seuraavasti: Sprite on yksi graafinen kuva, joka on integroitu osaksi isompaa pelin kohtausta, jotta se näyttäisi olevan osa sitä. Ja että spritet ovat suosittu tapa luoda suuria, monipuolisia kohtauksia, koska niitä on mahdollista manipuloida erillisenä muusta kohtauksesta. Tämä mahdollistaa paremman kontrollin siihen, kuinka kohtaus renderoidaan sekä siihen kuinka pelaajat voivat olla vuorovaikutuksessa kohtauksen kanssa.

Sprite onkin siitä oiva elementti, että sen sijaintia ja ominaisuuksia voi peliohjelmassa muuttaa lennosta. Construct 2 sovelluksessa, sprite objektilla on oletuksena seuraavat määritelmät: Animaatio, koko, näkymättömyys, oletusanimaatio, oletuskuva,

törmäyksentunnistus. (Scirra.com) Lisäksi objektille on lukuisia käsky, viite ja ilmaisumuotoja itse peliohjelmaa laatiessa, kuten ”Mikäli sprite objekti törmää toiseen sprite-objektiin, tapahtuu niin että molemmat sprite-objektit tuhoutuvat.”

Retronäkökulmasta, Dustmop.io:n blogin artikkelissa ”Nes graphics part 1” summataan, että NES:in väripaletti oli rajattu 64 väriin, joista ruudulla oli mahdollista esittää kerralla vain osa. Kuvaruutu jaettiin 16x16 laatikoihin, joista jokainen käyttää omaa neljän värin palettia. Tämä saa aikaan NES pelien laatikkomaisen yleisilmeen.

NES-konsolin tilet olivat 8x8 pikselin kokoisia osasia, joita yhdistämällä saatiin aikaan pelien taustat ja muut staattisemmat graafiset osat, tilejä voitiin kuitenkin myös manipuloida esimerkiksi värien osaltan, mikä mahdollisti monipuolisempaa visuaalista antia. Näistä mudoostettiin 16x16 pikselin laatikoita. Myöskin spritet olivat rajattu 8x8 kokoisiksi laatikoiksi, ja suuremmat sprite objektit koostuivatkin useammasta spritestä. Mm. Megaman hahmo koostuu 10 sprite objektista, mikä mahdollistaa myös useampien värien hyödyntämisen. Sprite objektin esittäminen ruudulla vei myös huomattavasti enemmän muistia kuin staattisempi tile objekti, sillä sprite objektilla oli useampia määritelmiä, kuten X ja Y sijainnit ruudulla. (Dustmop.io)

Retropeliä laatiessa, tyyliä valitessa onkin hyvä yrittää jäljitellä näitä rajattuja toiminnallisuuksia, jotta päästään mahdollisimman autenttiseen lopputulokseen. Nykyään kun vastaavia rajoitteita ei grafiikan osalta juuri ole, joudutaan ne retropeliä kehittäessä päättämään itse.

3.2 Retropeliäänitehosteet

Ääni on tärkeä pelielementti. Ääniefektit ja taustamusiikki lisäävät peliin paljon, kuten tunnelmaa, realistisuutta ja mielenkiintoa. Näiden lisäksi ne yleensä myös antavat pelaajalle tärkeää palautetta pelin tapahtumista. (Ryhänen 2015, 24)

Retroäänitehosteiksi voisi tunnistaa sellaisia äänitehosteita, joita tuotetaan digitaalisesti ja jotka kantavat mukanaan vanhojen konsolien äänitehosteiden tunnusmerkkejä, ominaisuuksia ja rajoitteita.

Eri konsoleilla oli omat ääntä tuottavat piirisarjansa ja esimerkiksi NES-konsoli käytti kustomoitua PSG-äänisirua, jossa oli käytössä yhteensä viisi kanavaa: kaksi pulssiaalto-, sahalaita-, ääninäyte- sekä kohinakanava. Toiseen pulssiaaltokanavaan voitiin vaikuttaa sweep ominaisuudella, jolla saatiin aikaan esimerkiksi laser-aseen äänitehosteet. (Matikainen 2014, 20)

Tämä mahdollisti Mega Man 2-Wily stage:n kaltaisten hittien luomisen, samalla sallien yhtäaikaiset Mega Manin mega busterin äänitehosteen sekä muista toiminnoista syntyvät äänet, kunhan osa musiikin raidoista uhrattiin niiden ajaksi. Tietyissä uudemmissa peleissä kanavia oli jopa kahdeksan, mutta tämä edellytti pelikasettiin asennettavan laajennuspiirisarjan hyödyntämistä, joka yleistyi vasta NES konsolin elinkaaren loppuvaiheilla. (Famitracker.com) Retropeliä kehittäessä onkin hyvä miettiä, mitkä ovat niitä keskeisiä äänitehosteita tuottavia tilanteita ja edustavatko ne haluttua retropelin tyyliä.

Yksinkertainen työkalu äänitehosteiden luontiin löytyy tietysti. Erittäin käyttökelpoinen vaihtoehto on BFXR, jolla toteutetut äänitehosteet ovat täysin ilmaiseksi käytettävissäsi.

Gamedevelopment.tutsplus.com artikkelissa ”Quick Tip: Make Retro, Low-Fi Game Sound Effects With Bfxr” summataan, että Bfxr on kätevä työkalu äänitehosteiden luomiseen NES aikaa mallintavalla tyylillä, kuten ääni, kuin Mario kolikoita kerätessään. Se toimii suoraan selaimessa ja tallentaa, joko Wav muotoon, jota lähes jokainen pelinkehitysalusta tukee, tai datamuotoon, jonka voi sisällyttää pelikoodiin käden käänteessä. Lisäksi sovellus generoi tietyn tyyppisiä äänitehosteita (hyppy, kolikon kerääminen, ampuminen jne.) napinpainalluksella, joten aloittelijakin saa aikaan perusäänet. Construct 2 tukee mm. wav, mp3, ogg, m4a äänitiedostoja.

3.3 Retropelimusiikki

Ensimmäisten laitesukupolvien ääniteknologia oli kovin rajoittunutta ja näin ollen sillä oli merkittävä vaikutus pelimusiikkiin. Kanavien vähäinen määrä loi omat ongelmansa moniäänisen musiikin tekemissä samoin kuin vähäinen muistitilan määrä. (Matikainen 2014, 45)

Pelimusiikin saundien kannalta keskeisessä asemassa ovat olleet äänisynteesimenetelmät. Varsinkin chptune-aikakauden videopelilaitteissa kullakin laitteella oli oma saundinsa, josta se vielä nykyäänkin muistetaan. Monet chiptune-aikakauden saundeista ovat jääneet elämään ja niille on muodostunut omat alakulttuurinsa. Monissa peleissä käytetään vielä nykyäänkin chiptunen kaltaisia saundeja. (Matikainen 2014, 45)

Pelimusiikki vapautui kanaviin liittyvistä rajoitteista 1990-luvun kehityksen myötä, kun massamuistilaitteet tulivat standardiksi peliteollisuudessa. Tällöin vapautui mahdollisuus käyttää valmiiksi äänitettyä musiikkia peleissä, eikä pelimusiikki ollut

enää sidonnainen kunkin koneen äänilaitteistosta. Samalla pelimusiikissa käytetty äänimaailma muuttui monipuolisemmaksi, koska saundit eivät olleet riippuvaisia videopelilaitteiden sisäänrakennetuista syntetisaattoreista. (Matikainen 2014, 45)

Teknisten rajoitteiden lisäksi pelimusiikin tekemistä rajoittaa tuotannolliset tekijät. Jokaisessa projektissa joudutaan pohtimaan musiikin merkitystä ja kuinka paljon siihen tullaan käyttämään aikaa ja rahaa. Varhaisimpien videopelikonsoleiden kohdalla tuotanto joutui päättämään myös siitä, kuinka paljon muistia musiikille ja äänille varataan. Yleensä määrä oli hyvin pieni ja tästä syystä johtuen monet pelimusiikin tekijät joutuivat kehittämään erilaisia ratkaisuja muistin säästämiseksi. Nykyään ongelmana ei ole välttämättä muisti, vaan esimerkiksi dynaamisen musiikin kehittäminen peliin, mikä on huomattavasti vaativampaa ja kalliimpaa kuin lineaarisen musiikin kehittäminen (Matikainen 2014, 45)

Retropeli-musiikkia laatiessa idea onkin lähestyä haetun aikakauden henkistä tyyliä, hyödyntäen niitä äänitysjärjestelmiä, joita tänä päivänä on käytössä. Retropeli-musiikin ei tarvitse välttämättä olla ohjelmoitua, vaan esiäänitetty kappale voidaan mallintaa ohjelmoidun kappaleen kaltaiseksi.

3.4 Retropelien teemat

Retropelejä on siinä missä modernejakin eli kaikenlaisia. Tyypillisiä toistuvia teemoja on maanisen maailmanvalloittajan pysäyttäminen väkivaltaisesti, prinsessan pelastaminen mörökölliltä, tai vaikka autolla ajaminen täysillä. Onkin siis hyvä oivaltaa, että lähes mistä tahansa aiheesta saa aikaan halutessaan retropelin.

Ensisijainen hyöty pelin kehityksessä selkeän teeman ympärille, on se, että kaikki pelisi elementit vahvistavat toisiaan, koska ne kaikki pyrkivät kohti yhteistä päämäärä. (Schell 2008, 48)

Kehittäjän kannattaakin selvittää mikä kehitettävän pelin teema on ja käyttää kaikkia keinoja hyödykseen selkeyttääkseen tätä teemaa. (Schell 2008, 49)

Teeman valintaan ja pelin suunnitteluun vaikuttaakin lähinnä pelin kehittäjää kiinnostava aihe ja halu toteuttaa siitä peli. Pelin suunnittelu onkin päättämistä siitä, mikä pelin tulisi olla. (Schell 2008, xxiv)

4 ENSIKERTALAISENA PELINTEON MAAILMAAN

Hello world. Tämä lausahdus lienee tuttu? Mikäli koskaan olut kokeillut ohjelmoida, olet todennäköisesti kehittänyt yksinkertaisen sovelluksen, joka tulostaa ruudulle kyseisen tekstinpätkän. Sekin vaati jo jokseenkin aivotyöskentelyä, tai siis ainakin jos halusi ymmärtää mitä ohjelman koodissa tapahtuu.

Alla W3schools.com-sivuston esimerkki Javascript koodista, joka tulostaa ruudulle tekstin ”Hello world!”.

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello world!";
</script>

</body>

</html>
```

Tämä ohjelma on kuitenkin vielä kaukana pelistä. Mikä siis on peli?

Jesse Schell (2008, 34) jakaa termin peli seuraavaan kymmeneen käsitteeseen:

1. Peliin osallistutaan tietoisesti
2. Peleissä on päämäärä
3. Peleissä on konfliktia
4. Peleissä on säännöt
5. Pelin voi voittaa tai hävitä
6. Pelit ovat interaktiivisia
7. Peleissä on haastetta
8. Peli voi tuottaa sen sisäistä arvoa
9. Pelit pitää pelaajan otteessa
10. Peli on suljettu, muodollinen järjestelmä

Mahdollisen tiivisti kuvattuna, peli on ongelmanratkaisuaktiviteetti. (Schell 2008, 36)

Peli muodostuu pääasiassa neljästä elementistä: estetiikka, mekaniikka, tekniikka ja tarina. (Schell 2008, 42)

Ajatus pelintekemisestä voikin tuntua mahdottoman suurelta kokonaisuudelta, mutta ei hätää. Pelin suunnittelu on yksinkertaisesti päätösten tekoa, voit oikeastaan suunnitella pelin päässäsi. Monia pelejä voi pelata ilman tietokoneita tai teknologiaa; lautapelit, korttipelit, urheilupelit esimerkiksi. (Schell 2008, xxvi)

Tehdäänkin siis yhdessä yksinkertainen ajatuspeli. Tämä peli ei vaadi mitään laitteita, sovelluksia tai ajattelua kummempaa osaamista.

4.1 Aloita pienestä

Ajattele pelin sankaria. Mitä ajattelit? Omaan päähäni tuli pizza, joka pyörii vaakatasossa, on varmaan nälkä. Ajattele seuraavaksi pelin tavoitetta. Omassani pizzan tulee pyöriä ruokapöytään. Mitä tavoitteeseen pääsemisen välissä tapahtuu? Huolestuttavaa: Pizza voi likaantua!
Näin saimme aikaan äärimmäisen yksinkertaisen konseptin pelille. Teen siitä heti prototyypin, eli mockup version Construct 2 -sovelluksella, ajatusleikin päätteeksi.

Kuva 3 peli-idea mockup.

Tässä vaiheessa peli toimii tietysti vain omassa mielessä, mutta lähes kaikki omat peliprojektini kumpuavat tällaisista lähes idioottimaisilta aluksi tuntuvilta ajatusleikeistä.

Prototyyppiä kehittäessä voi hyödyntää vaikka paperia ja kynää, paperileikkeitä tai muuta täysin tietotekniikasta erillään olevaa keinoa.

Voit luoda prototyypin hienosta peli-ideastasi vaikka lautapelin muodossa, tai niin sanotun paperiprototyypin. Miksi näin? Koska voit tehdä lautapelejä nopeasti, jotka samalla omaavat saman pelimekaniikan. (Schell 2008, 88)

Mutta mikäli kuvankäsittelyohjelma tuntuu tutummalta, voi peli-ideaa parsia kasaan vaikka Paint:illa.

4.2 Löydä sisäinen pelintekijäsi

Kuten kuitenkin historia on osoittanut, voi aluksi täysin typerältä tuntuva konsepti osoittautua äärettömän koukuttavaksi. Ajatellaanpa vaikka surullisen kuuluisaa FlappyBird peliä, jossa lintu väistelee Super Mariosta tuttuja putken pätkiä. Peli oli menestys, joka aiheutti tekijälleen suurta stressiä ja häpeää 50000$ päivittäisen tulon muodossa. (Hogging 2014, 1)

Nyt lukuisat upeat peli-ideat jäävät ajatustasolle, sillä niiden ajattelija ei tiedä pystyvänsä toteuttamaan konseptinsa käytännön tasolla. Olisikin hienoa ja jopa mahtavaa jos näitä ajatusten konsepteja saataisiin jatkossa käytännössä pelattavaan muotoon entistä useammin.

Peli on kuin savi, jota pelinkehittäjä muotoilee luodakseen mahtavia pelikokemuksia. (Schell 2008, 26) Tekemällä oppii. Construct 2 ilmainen kokeiluversio tarjoaa mainion ympäristön yksinkertaisen pelilogiikan hahmottamiseen ja konsepteilla leikkimiseen. Seuraava kappale käsittelee yksinomaan kyseisen sovelluksen hyödyntämistä pelinkehityksessä. Saat ladattua sen ilmaiseksi täältä: https://www.scirra.com/construct2/releases

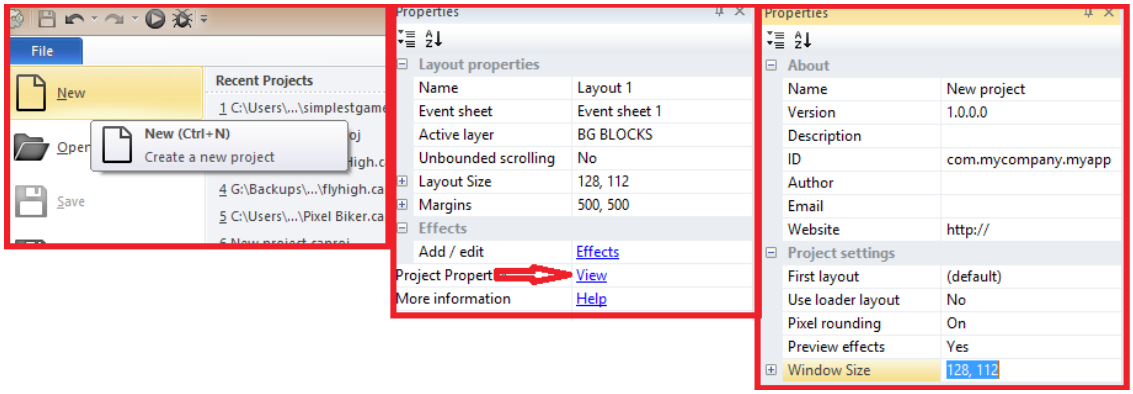
5 PELINKEHITYS CONSTRUCT 2 - SOVELLUKSELLA

5.1 Ideasta toteutukseen

Leikkiessäni ajatuksella pöytään päätyvästä pizzasta ja sitä hetken sulatelleena onkin idea jo hioutunut muotoon, josta saa sopivan pelinkehitysprosessia kuvaavan esimerkkiteoksen.

Kuvapisteitä säästääkseni päädyin retroteemasta poiketen minimalistiseen superretroon (itse keksitty nimitys, tämähän on taidetta ja toteutan tässä aiemmin mainittua pelin graafisen ilmeen rajoittamisen periaatetta). Esimerkkiprojektini resoluution onkin vain puolet NES-konsolin resoluutiosta eli 128,112. Täten spritejen sekä tilejen tuotantoon ei mene tuhottomasti aikaa tässä projektissa. Mitä pienempi resoluutio projektissasi on, sitä suurempana yksi kuvapiste näyttäytyy ruudulla.

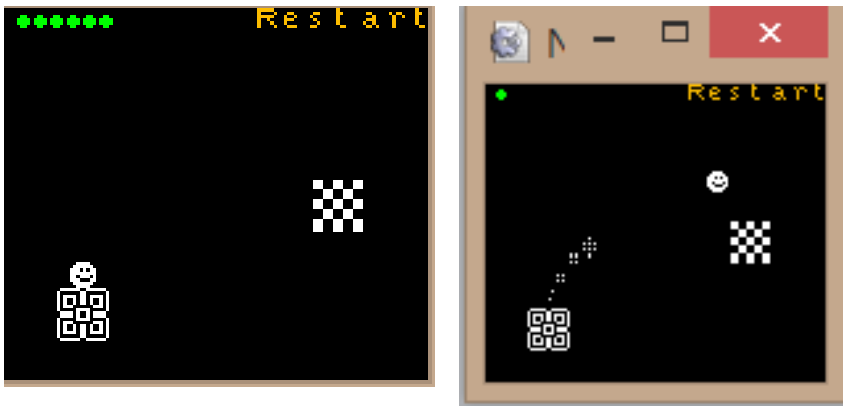
Kuvallinen ohjeistus uuden Construct 2 projektin aloittamiseen ja resoluution määrittelemiseen:



Kuva 4 uuden retropelin määrittely Construct 2 sovelluksella.

Valitse FILE välilehdeltä NEW ja klikkaa VIEW Project Properties valikosta. Määrittele kohtaan Window Size ensin vaakasuorat kuvapisteet 128 ja erottele pilkulla pystysuorat kuvapisteet 112.

Autenttisemman retromeiningin saavutat kun valitset lisäksi samasta valikosta Sampling valikkoon määritelmän Point, joka poistaa oletuksena käytössä olevan kuvapisteiden reunanpehmennyksen. Lisäksi Pixel rounding kannattaa klikata päälle. Tämä takaa sen, että pikselien väliin ei piirretä mitään vaikka kuva olisi skaalattu pelin resoluutiota suuremmaksi.



Kuva5 ja 6 selkeitä pikseleitä.

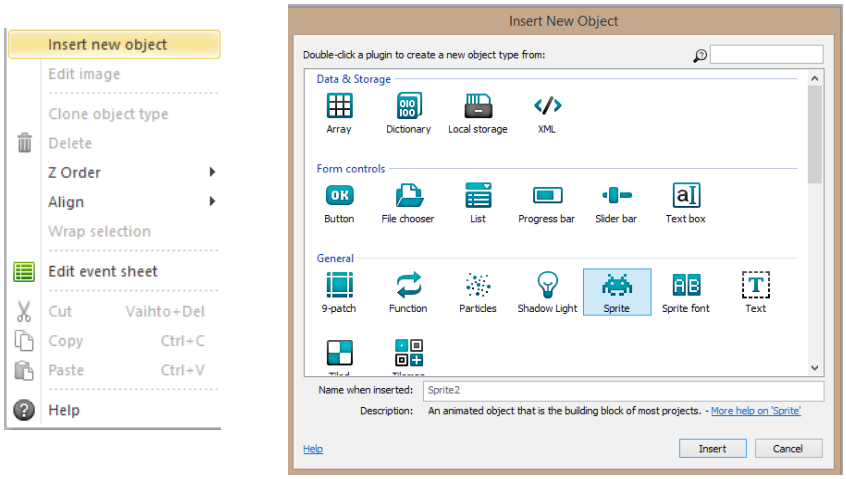
5.2 Tavoite ja tarkoitus

Tavoite on ennallaan, vie pizza pöytään. Pizza on vaihtunut sitä kaukaisesti muistuttavaan valkoiseen palloon, jonka animaatiot rajoittuvat lähinnä hymynaaman ilmeen vaihteluun. Pöytää kuvaa yksinkertainen shakkiruudukko. Tapa jolla pallo liikkuu hyödyntää kosketusnäyttökomentoja tai hiirtä sekä Construct 2 – sovelluksen omaa 2d fysiikkamoottoria.

5.3 Peliobjektien luominen

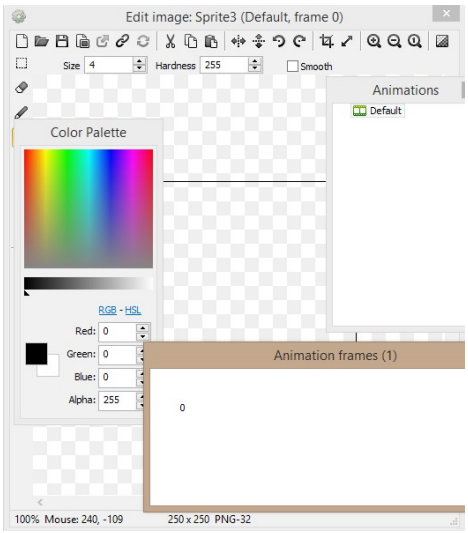
Sankari:

Pelisankarimme on käytännössä 8x8 pikselin sprite-objekti. Saat luotua spriten Construct 2 – sovelluksella klikkaamalla layout-näkymässä tyhjää aluetta hiiren oikealla näppäimellä, valitse Insert New Object. Valitse avautuvasta valikosta Sprite ja klikkaa Insert.



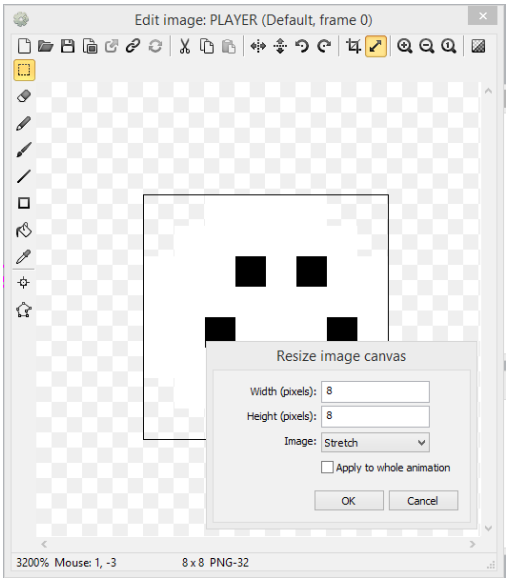
Kuva 7 insert new sprite.

Lopuksi klikkaa vielä layout-näkymää hiiren vasemmalla painikkeella. Ruudulle avautuu nyt Sprite-editor-näkymä, jolla määrittelet kuvapisteet, törmäyslaatikon, hotspotit sekä muut oleelliset sprite-määritteet.



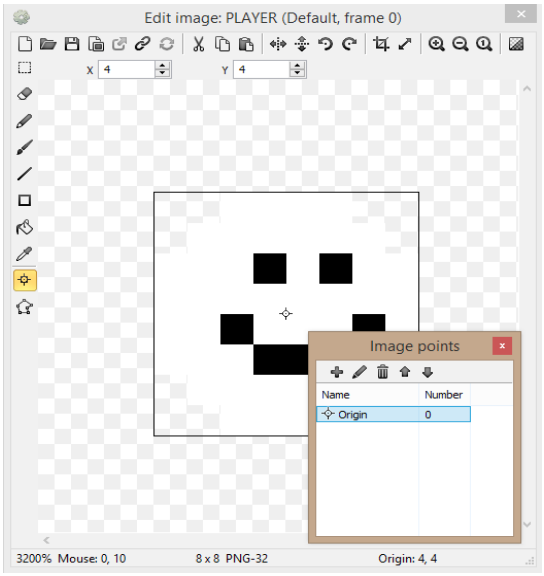
Kuva 8 sprite-editor.

Määrittele ensin spriten koko ylhäällä olevasta kahta nuolta kuvaavasta kuvakkeesta (8 x 8 pikseliä).
Piirrä alueelle haluamasi kuva tarjolla olevia työkaluja hyödyntäen.



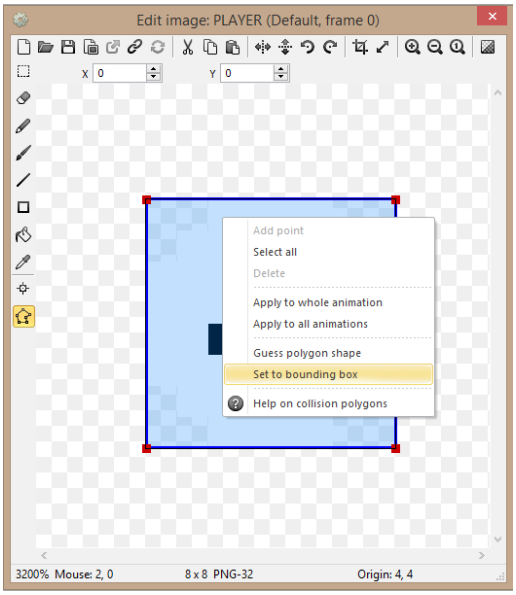
Kuva 9 spriten kokomäärittely.

Määrittele spriten keskuspiste vasemmassa alareunassa toiseksi alimpana olevaa työkalua hyödyntäen. Tämä piste määrittelee keskustan, johon viitataan kun tarkkaillaan objektin sijaintia ruudulla. Itse olen sen sijoittanut keskelle eli kohtaan X:4-Y:4



Kuva 10 spriten keskuspisteen määrittely.

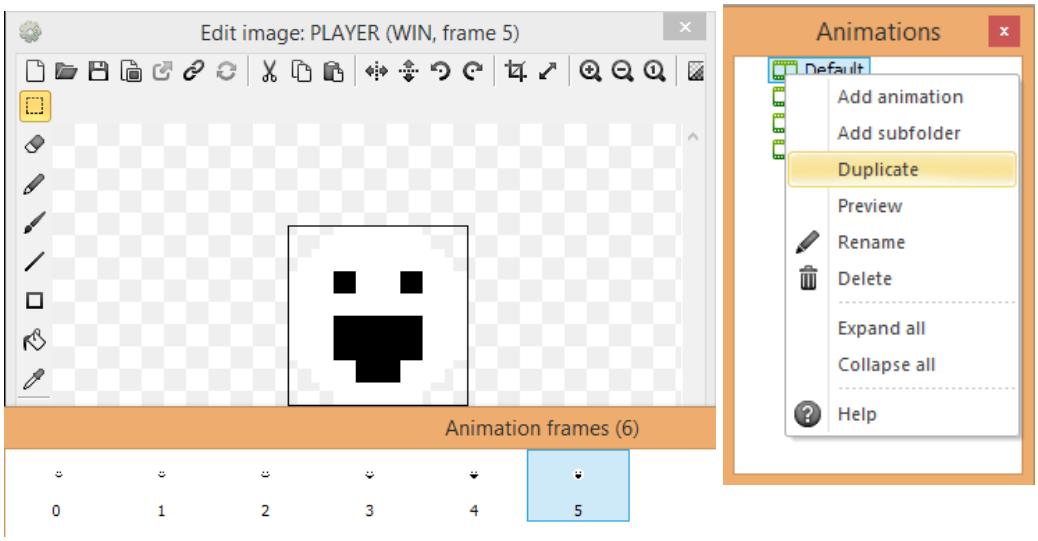
Määrittele vielä objektille törmäysalue kuvatulla tavalla. Set to bounding box määrittelee alueen koko kuva-alueen kokoiseksi. Tällä määritteellä ilmaistaan spriten fyysinen alue, joka havaitsee eri spritejen törmäykset. Esimerkiksi jos kaksi objektia on ruudulla päällekkäin, ohjelma tarkkailee sitä, että onko tämä määritelty alue kosketuksissa toiseen objektiin sen sijaan, että niiden kuvapisteen kokomääritelmien laatikoita tulkittaisiin törmäysalueiksi.



Kuva 11 spriten törmäystunnistimen määrittely.

Voit luoda spritellesi myös yksinkertaisen animaation, kopioimalla kuvaeditorin Aimation frames näkymästä, klikkaamalla hiiren oikealla näppäimellä alkuperäisen

kuvan päältä ja valitsemalla duplicate. Tämä metodi helpottaa prosessia, asettaen seuraavan framen määritteet, (resoluution, hitboxin sekä hotspotit) automaattisesti vastaamaan edellistä kuvaa. Tee animaatio mielesi mukaan ja luo sen lisäksi vielä seuraavat animaatioluokat: ”Default, WIN, LOSE, ja DRAG”.



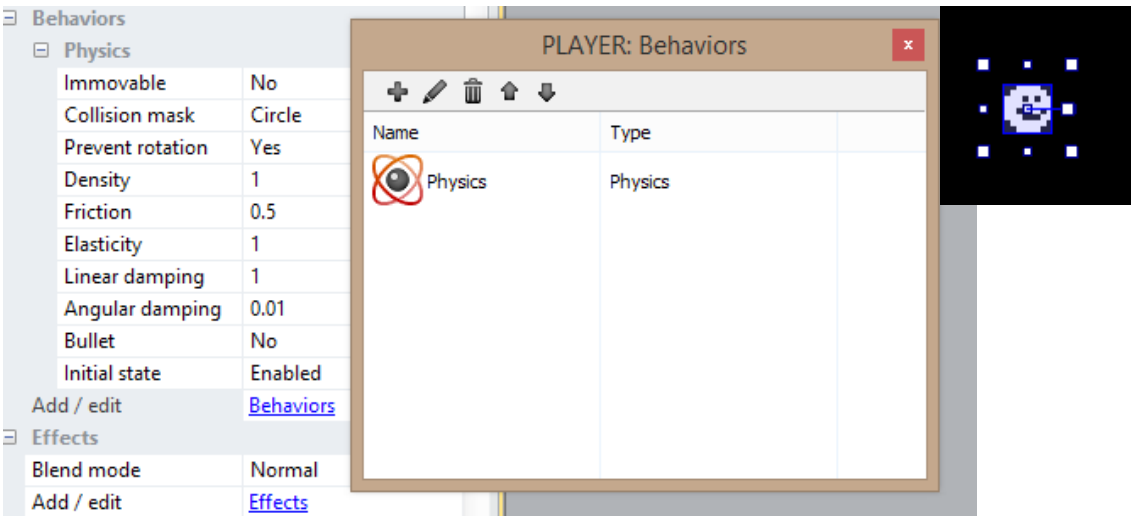
Kuva 12 spriten animaation duplikointi.

Määrittele animaatioihisi toisto päälle säätämällä properties ikkunasta määritelmä ”Loop” tilaan ”Yes”.

Sulje kuvaeditori ja ensimmäinen sprite objektisi on nyt lähtökohtaisesti luotu.

Valitse pelisankari-sprite klikkaamalla sitä hiiren vasemmalla näppäimellä. Näkymän vasempaan reunaan ilmestyy nyt objektisi määritelmät. Klikkaa sinisellä vahvistettua linkkiä Behaviours ja lisää + painikkeella toiminnallisuus Physics. Toiminallisuuden määritteet avautuvat pudotusvalikkoon automaattisesti. Tee lopuksi seuraavat muutokset fysiikkatoiminnallisuuden määritelmään. Vaihda ”Prevent rotation” määritelmä ”No”-tilasta ”Yes”-tilaan. Tämä estää spriten kääntymisen. Spriten törmäyslaatikko on neliön muotoinen, joten vaihdetaan objekti hyödyntämään oletustörmäyslaatikkonsa sijasta ympyrän muotoista. Vaihda siis ”Collision mask” määritelmä ”Circle” tilaan.

Törmäyksen sattuessa sen yhteyteen yleensä lisätään jokin toiminto. Tällainen toiminto on esimerkiksi toisen spriten tuhoutuminen.(Ryhänen 2015, 25)

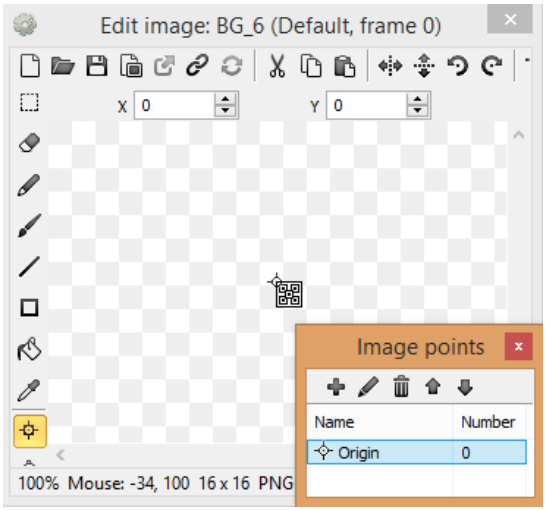


Kuva 13 spriten käyttäytymismalli, fysiikkamallinnus.

Kun hallitsee spritejen luomisen, on helppo jatkaa varsinaisen pelin rakentamiseen. (Ryhänen 2015, 25)

Seuraavaksi luodaan yksinkertaista sprite-objekteja, joista muodostuu pelin ratojen reunat ja tasot.

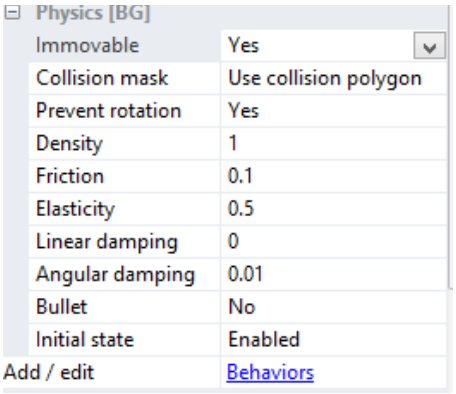
Luodaan jälleen uusi Sprite-objekti, kuten aiemminkin. Klikkaa layout-näkymässä tyhjää aluetta hiiren oikealla näppäimellä. Valitse Insert New Object. Valitse avautuvasta valikosta Sprite ja klikkaa Insert. Luo alkuun 16x16 sprite, tee siitä sen näköinen kun haluat, mutta määrittele keskuspiste käytännön syistä kohtaan 0,0 eli spriten vasempaan yläkulmaan. Tämä määritelmä on sprite-objekteissa oletuksena. Määrittele törmäyslaatikko jälleen koko sprite-objektin kokoiseksi. Objektin voi animoida, mutta se ei ole esimerkkiprojektissani tarpeen.



Kuva 14 taustaobjekti sprite.

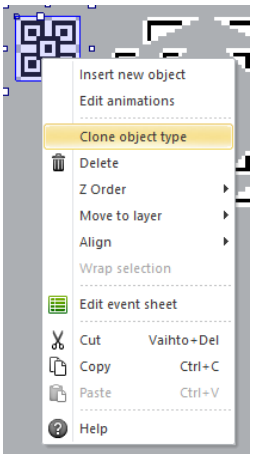
Kun olet tyytyväinen spriten ulkonäköön sulje sprite editori.
Valitse sprite klikkaamalla sitä hiiren vasemmalla näppäimellä. Näkymän vasempaan reunaan ilmestyy taas objektisi määritelmät. Klikkaa sinisellä vahvistettua linkkiä Behaviours ja lisää jälleen + painikkeella toiminnallisuus Physics.

Vaihda ”Immovable” määritelmä tilaan ”Yes”. Tee sama myös ”Prevent rotation” määritelmälle.



Kuva 15 taustaobjekti spriten käyttäytymismalli.

Kloonaa objekti klikkaamalla sitä hiiren oikealla näppäimellä ja valitsemalla aukeavasta valikosta ”Clone object type”.

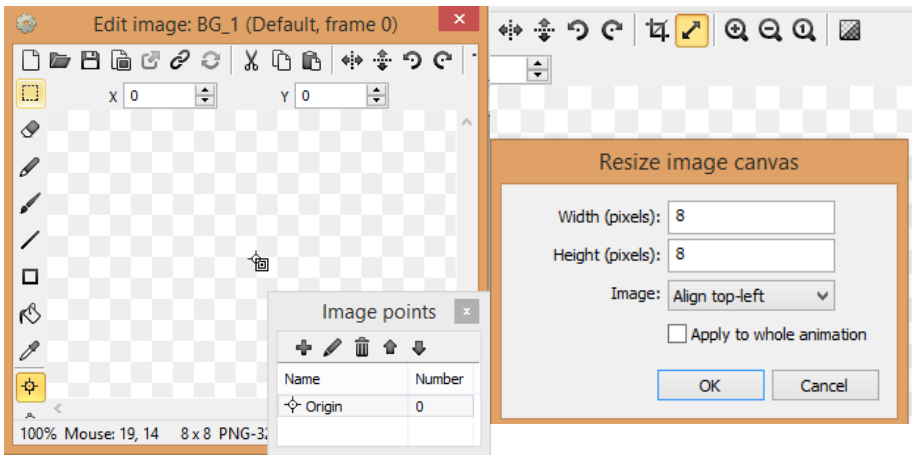


Kuva 16 objektin kloonaus.

Klikkaa vielä tyhjää aluetta ruudulla. Objektisi klooni ilmestyy nyt tähän. Kaikki määritteet ovat kuten alkuperäisessäkin versiossa.

Kaksoisklikkaa kloonia ja sprite editori avautuu jälleen.

Muuta kloonin koko 16x16 kuvapikselistä 8x8 formaattiin valitsemalla sprite editorin ylätyökaluriviltä kaksipäistä nuolta kuvaava painike.



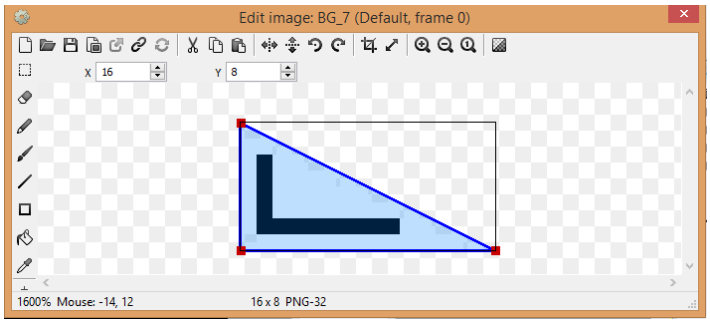
Kuva 17 ja 18 kloonatun taustaobjektin spriten kuvakoon muuttaminen.

Ennenkuin painat Ok painiketta, varmista, että olet valinnut ”image” määritelmään ”Align top-left” ”Stretch” määritelmän sijaan. Stretch venyttää ja sumentaa pikselit mössöksi ja Align työkalu käytännössä vain rajaa kuvan. Rajaus on järkevämpi metodi, sillä pyrimme pitämään kuvapisteet selkeinä ja käyttämään vain rajatusti värisävyjä. Stretch muodostaisi käytännössä kuvapisteiden sumeita keskiarvoja.

Voit taas tehdä objektistasi halauamasi näköisen kunhan määritelmät ovat muuten kohdallaan.

Sulje kuvaeditori kun olet tyytyväinen objektiisi.

Luo lisää klooneja. Voit halutessasi koittaa tehdä vaikka kaltevia tasoja, tai mitä mieleesi ikinä juolahtaakaan.

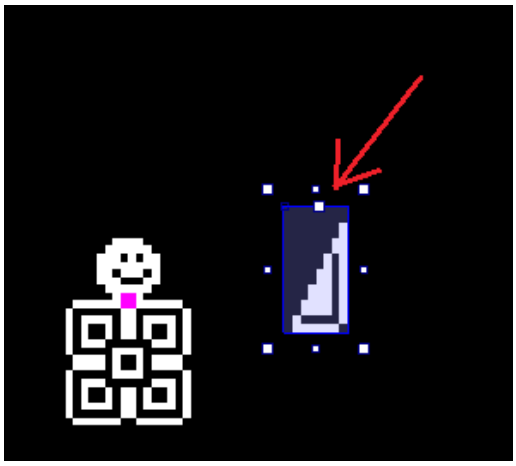


Kuva 19 kaltevaksi määritelty taustasprite objekti.

Kun olet tyytyväinen luomiisi sprite-objekteihin, valitse työkalurivin alta View välilehti. Määrittele Grid Width ja height määritelmiin arvot 8. Tämä mahdollistaa objektien sijoittamisen tasoon ruudukon avulla, joka helpottaa ratojen kehitystä huomattavasti.

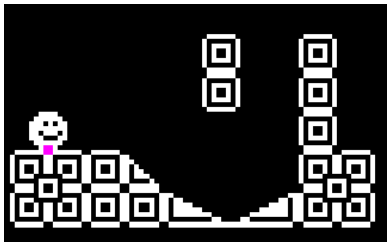
Voit nyt kokeilla kopioida ja liittää fyysisiä objekteja eri tavoin tasoon. Kaltevia

epäsymmetrisiä objekteja voit myös kääntää ruudulla. Klikkaa objektia kerran jotta objektin koko ja kulma työkalut ilmestyvät sen ympärille. Nuolen osoittama objektin yläreunassa oletusarvoisesti oleva piste mahdollistaa objektin kulman säätämisen. Kääntäessä objektia samalla shift painiketta pohjassa pitämällä, kääntyy objekti useampi aste kerrallaan, joka auttaa sijoittamaan objektin haluttuun kulmaan helpommin.



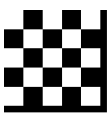
Kuva 20, objektien kääntö.

Voit sijoitella palikoita esimerkiksi näin:



Kuva 21, objektien sijoittelu.

Tarvitsemme vielä pizzalle pöydän, eli maalin johon pelaajan on tarkoitus radassa päästä. Luo siis vielä uusi sprite-objekti, esimerkiksi kokoa 32x32pikseliä. Määrittele jälleen törmäysalue koko objektin kokoiseksi ja hotspot keskelle objektia.



Kuva 22, maali

Itse tein siitä yksinkertaisen shakkilautakuvion, mutta voit jälleen tehdä siitä haluamasi näköisen. Objektille ei tarvitse tehdä muita määritelmiä.

5.4 Vaarat ja viholliset

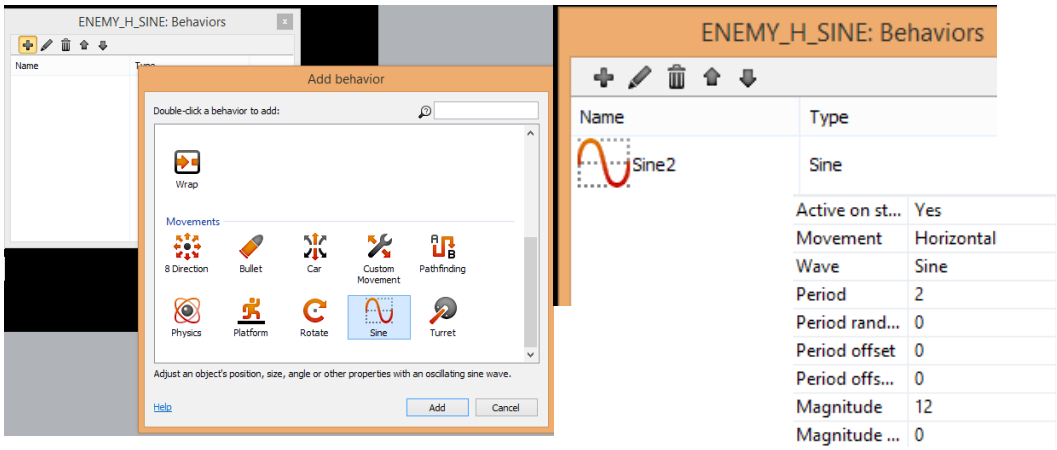
Pelissä ei olisi haastetta ilman vihollisia, joten luodaan vielä yksinkertainen liikkuva vihollinen.

Luo jälleen sprite-objekti, esimerkiksi kokoa 8x8pikseliä. Määrittele törmäysalue ja hotspot.

Voit luoda objektillesi animaation mikäli haluat. Omaa vihollistani edustaa yksinkertainen neljän framen surunaama.

Kuva 23, vihollinen.

Kun olet luonut objektin ja sulkenut kuva-editorin, klikkaa objektia kerran jotta se aktivoituu ja valitse Properties näkymästä Behaviour → add/ edit behaviours ja lisää objetille sine käyttäytymismalli.



Kuva 24 ja 26 vihollisen käyttäytymisen määrittely.

Properties näkymään avautuu nyt uusia määritelmiä joilla voit vaikuttaa vihollisen liikkeeseen.
”Acitve on start” määrittelee onko sine-käyttäytyminen alusta asti aktiivinen.

Movement valikosta voit valita vaaka tai pystytason liikkeen, joihin keskitymme tässä tutoriaalissa.

Valitse horizontal eli vaakatason liike. Wave määritelmästä voit valita eri aaltokäyttäytymisiä, mutta pitäydymme sine-aallossa.

Period määritelmä vaikuttaa liikeradan kestoon. Oman objektini liike kestää oletusarvoisesti 2 sekuntia.

Period random määritelmä lisää kestoon satunnaisesti maksimissaan sen verran sekunteja kun siihen määrittelet. Jos määrittelet tähän kohtaan 10 sekuntia saattaa satunnaisarvo olla siis väliltä 0-10 sekuntia, -10 taas o--10 sekuntia jne.

Satunnaisuudelle ei tällä erää ole tarvetta, joten jätän arvon nollaksi.

Period offset taas vaikuttaa siihen mistä kohtaa aikajanaa liike alkaa. Tähänkään ei nyt tarvitse puuttua. Period offset random vaikuttaa vastaavasti offsetin satunnaisuuteen vastaavasti kun period random vaikutti kestoon.

Magnitude vaikuttaa liikkeen kuvapisteiden määrään. Määrittelin omani liikkumaan 12kuvapistettä.

Tällekin arvolle on aiempia vastaava satunnaislisäarvo, Magnitude random. Jälleen 0.

Kun olet tehneet määritelmän, voit jatkossa kopioida vihollis-objektia ja muuttaa eri vihollisyksilöiden sine-käyttäytymisen arvoja. Voit myös luoda esimerkiksi toisenkin sine-käyttäytymisen, joka mahdollistaisi monimutkaisempia liikeratoja useampaan suuntaan.

Halutessasi voit myös kloonata vihollisobjektin, luoda kloonille eri ulkonäön ja valmiiksi määritellä erilaisen sine-käyttäytymismallin. Täten käytössäsi olisi kaksi tai useampi valmista vihollisluokkaa, joita voi näppärästi sijoitella ympäri pelialuetta.

Itse loin toisen vihollisen, joka liikkuukin pystyasennossa ja on selkeästi erilainen eli lila.

Kuva 27, lila versio vihollisesta.

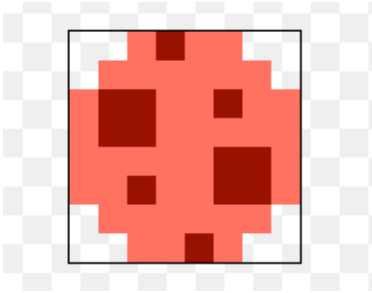
Voit toki luoda vastaavia vihollispalikoita ja poistaa niiltä sinekäyttäytymisen kokonaan jolloin saat luotua staattisia vihollisia, esimerkiksi piikkejä tai mitä mieleesi juolahtaa.

Kun olet tyytyväinen vihollis-obekteihisi, on aika siirtyä kerättävien objektien pariin.

5.5 Kerättävät bonukset

Peleissä usein kerätään pisteitä. Tässä esimerkissä pelaaja kerää pizzantäytteeksi salamiviipaleita, jotka on kerättävä, että pelin maaliruudukko aktivoituu ja radan voi läpäistä.

Salamiviipale objektini on 8x8 pikselin kokoinen sprite, jonka törmäysalue on koko objetin kokoinen ja jolla ei ole animaatiota lukuunottamatta mitään muita määrityksiä. Hotspot on objektin keskellä jälleen.



Kuva 28, kerättävä salamiviipale.

Luo haluamasi vastaava uusi sprite objekti.

Lisäksi luodaan painike, jonka avulla pelaaja voi käynnistää radan uudestaan. Itse tein yksinkertaisin 16x16 pikselin painikkeen sprite objektista jolla ei ole erityisiä määrityksiä:

Kuva 29, reset – painike.

Jotta pelaaja voisi jollain tavalla vaikuttaa pelin tapahtumiin luodaan vielä Touch objekti, joka mallintaa pc ympäristössä hiiren painallusta ja kosketusnäyttöympäristössä ruudun kosketusta:

Insert new object → touch
Luo vielä ääniobjekti jotta voimme jatkossa toistaa pelissä myös äänitiedostoja.

Insert new object → audio

5.6 Pelilogiikka

Nyt kun objektit ovat luotu, on meillä kasa epäinteraktiivisia palikoita, jotka eivät itsessään tuota minkäänlaista pelikokemusta. Perinteisesti pelin toiminnallisuudet ovat ohjelmoitu käyttämällä maallikolle monimutkaisia ohjelmointikieliä, mutta Construct 2 käyttää tapahtumapohjaista ”klikkaa ja kehitä” tyyppistä toimintomallinnusjärjestelmää.

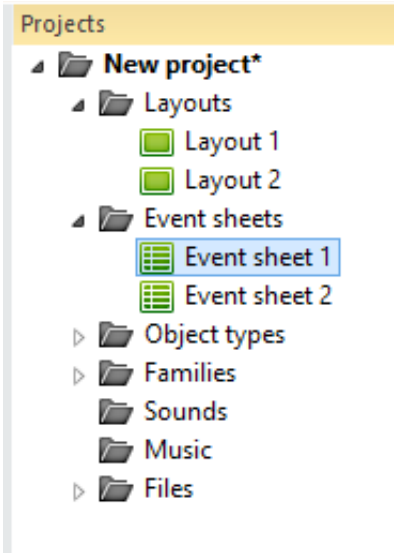
Sovellus muuntaa nämä käyttöliittymätason määritteet JavaScriptin ja HTML5:den harmooniseksi yhteistyöksi. HTML5-koodiin pääsee erikseen käsiksi ja siitä on mahdollista suoraa muuttaa pelin toimintoja.

Käyttöliittymä on intuitiivinen ja objektipohjainen. Erilaiset tapahtumat, kuten, ”jos kaksi objektia törmäävät toisiinsa”, saadaan aiheuttamaan reaktioita, kuten ”tuhoa objekti 1”, ”kun peli alkaa: soita kappale pelimusa.ogg” tai ”kun pelaaja sprite on pelialueen ulkopuolella-> aloita alusta”.

Alla olevassa kuvassa on kaikki esimerkki pelini vaadittu ohjelmoitu toiminnallisuus, muu aktiivisuus tulee aiemmin tehdyistä objektien määrityksistä.

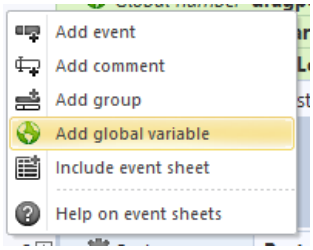
Kuva 30, peliohjelma.

Siirtyäksesi Events näkymään kaksoisklikkaa Projects näkymästä oletuksena luotua Event sheet 1:stä.



Kuva 31, event näkymään siirtyminen.

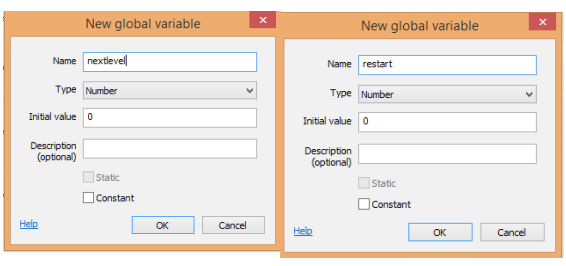
Luodaan ensin tarvitut muuttujat: dragpower, restart ja nextlevel.
Muuttuja luodaan klikkaamalla hiiren oikealla painikkeella event näkymän tyhjää valkoista aluetta ja klikkaamalla Add global variable.



Kuva 32, muuttujien määrittely.

Määrittele ensimmäisen muuttujan nimi: restart, tyyppi: numero ja alkuarvo: 0. Klikkaa ok.

Toinen muuttuja vastaavasti: nimi nextlevel, tyyppi: numero, alkuarvo 0.

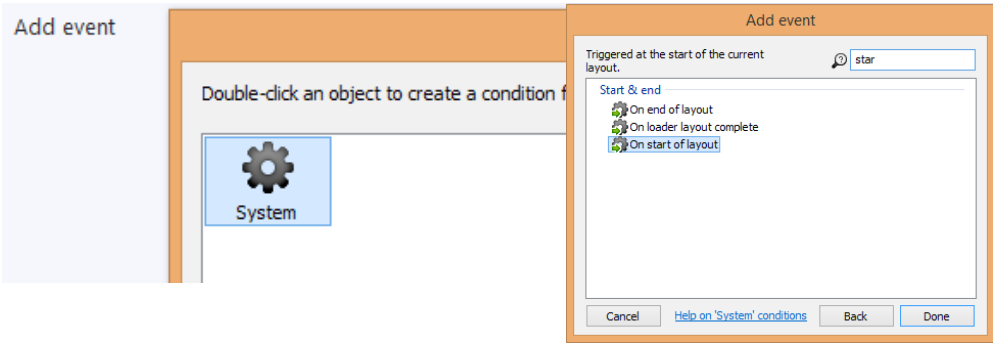


Kuva 33 ja 34, muuttujien määrittely, alkuarvot, tyyppi.

Esimerkin tarpeelliset muuttujat ovat nyt luotu.

On aika siirtyä komentojen pariin.

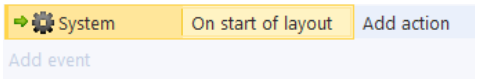
Klikkaa muuttujien alta Add Event. Kaksoisklikkaa system. Valitse näkymästä ”start of layout”



Kuva 35 ja 36, add event → start of layout.

Start of layout toiminto ajaa määrittelemäsi komennot kerran kunkin tason alussa.

Valitse näkymään ilmestyneen komennon oikealta puolelta Add action.



Kuva 37, add action

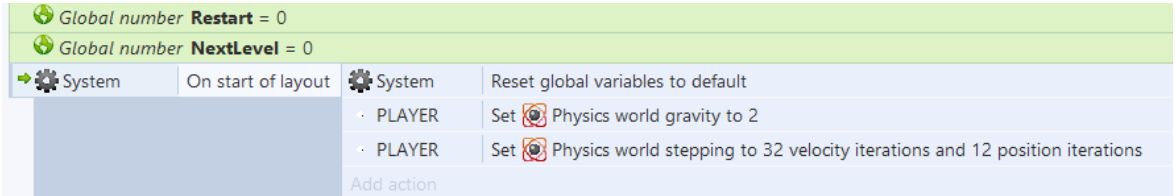
Lisää seuraavat toiminnot:

System –> Reset Global variables to default = jokaisen radan alussa muuttujat resetoituvat.

Pelaaja sprite → Set World Gravity to 3 = oletuspainovoima on 10, tiputetaan se 3:een.

Pelaaja sprite → set stepping iterations, velocity 32, position 12 = oletusarvot ovat 8 ja 3, mitä suuremmat arvot, sen tarkemmin sovellus laskee objektien fysiikkamallinnuksen sijainti ja liikedataa.

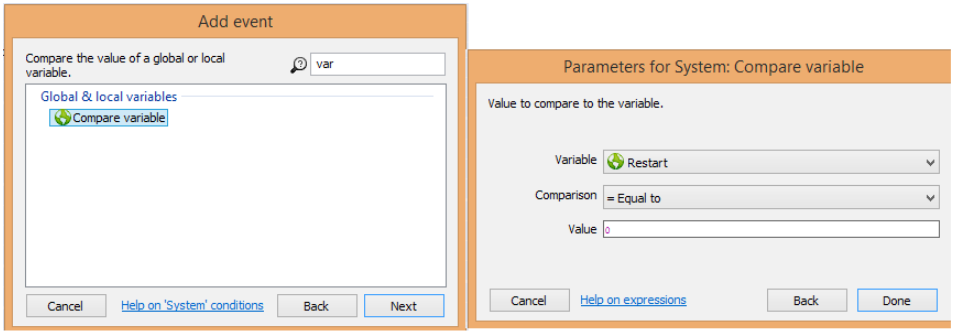
Määritteet tulisi näyttää ruudulla nyt tältä:



Kuva 37, komentojen määrittely.

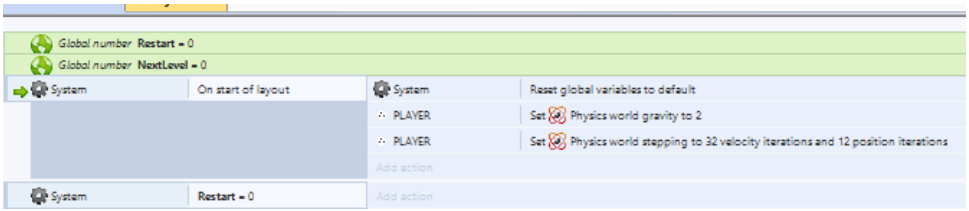
Lisää seuraavaksi komento, jonka alle luodaan alikomentoja:

Add Action → system → Compare variable → Restart = 0



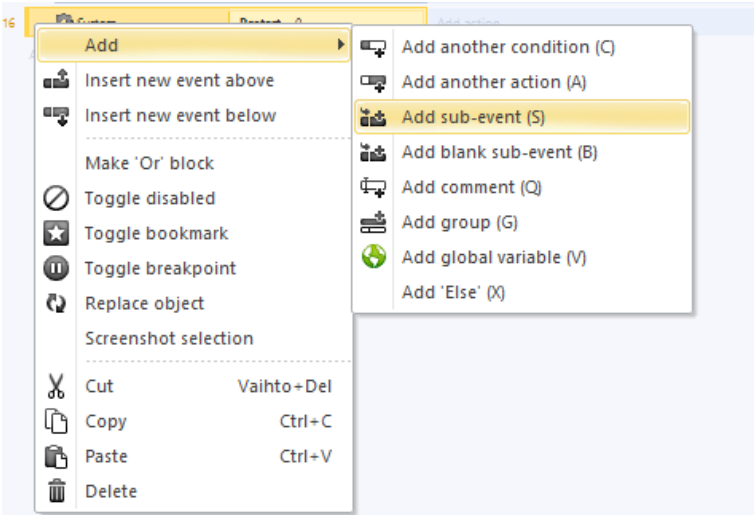
Kuva 38. if restart = 0

Klikkaa ok, ja toiminto ilmestyy jälleen vasempaan alareunaan.



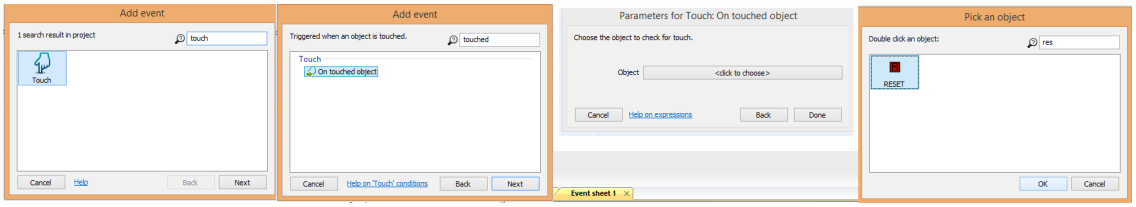
Kuva 38. toiminto system if restart = 0

Klikkaa luomasi System → restart = 0 toiminnon kuvakkeen vasenta puolta ja valitse avautuvasta valikosta add → add sub-event(s)



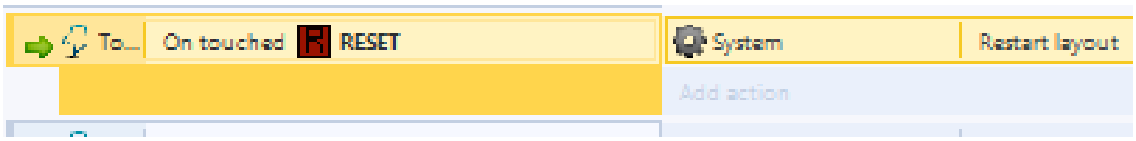
Kuva 39. add sub event.

Valitse Touch → On touched object → kehittämäsi uudelleenkäynnistyssprite.



Kuva 39 Uudelleenkäynnistys-spriten koskettamistoimintojen määrittely.

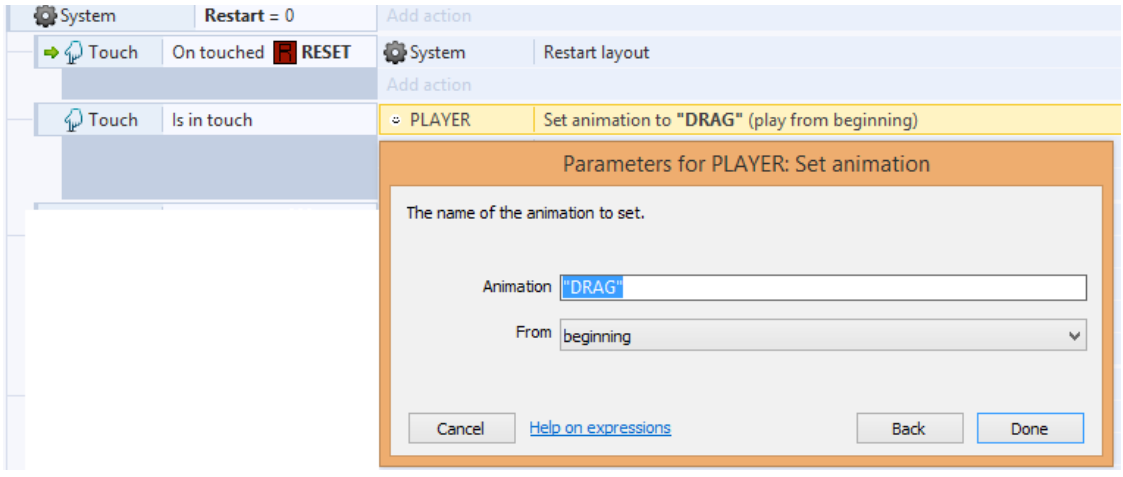
Lisää toiminnollesi komento System → Restart layout.



Kuva 40 Resetoi taso.

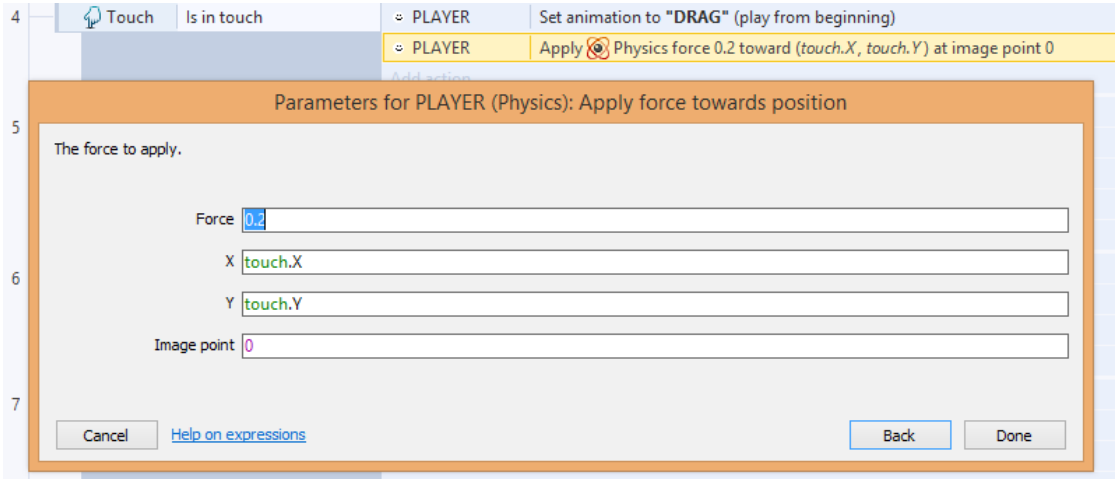
Valitse äsken määritellyn toiminnon alta, add event → touch → is in touch.

Mikäli teit pelisankariobjektillesi animaation ”Drag” niin lisää komento Pelaajaobjekti → set animation to ”drag”, from beginning.



Kuva 41 Jos animaatio DRAG toistuu.

Lisää tämän alle toinen komento: Pelaajaobjekti → apply physics force 0.2 toward: touch.x, touch.y, imagepoint 0.



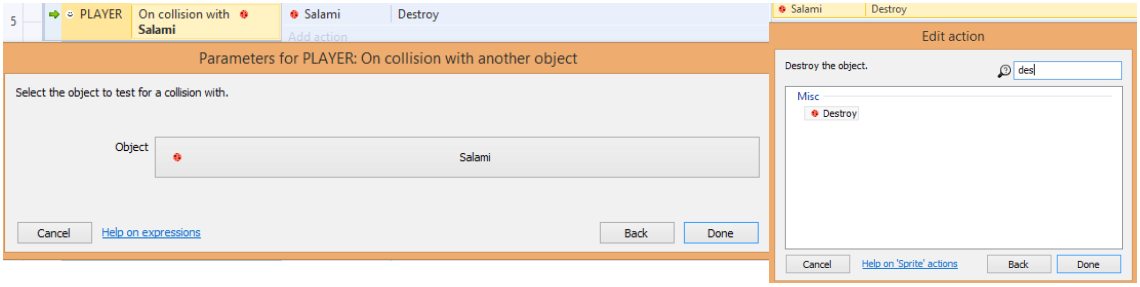
Kuva 42 apply force to touch

Nyt kun pelaaja koskettaa ruutua tai klikkaa pelialuetta hiirellä, hakeutuu peliobjekti kohti painalluksen sijaintia ruudulla. Tämä on käytännössä siis lähes yksinkertaisin pelisankarin ohjausmetodi.

Seuraavaksi määritellään salamiviipaleiden kerääminen ruudulta.

Lisää edellisen alapuolelle komento Pelaajaobjekti → on collision with → salamiobjekti.

Määrittele toiminto salamiobjekti -destroy

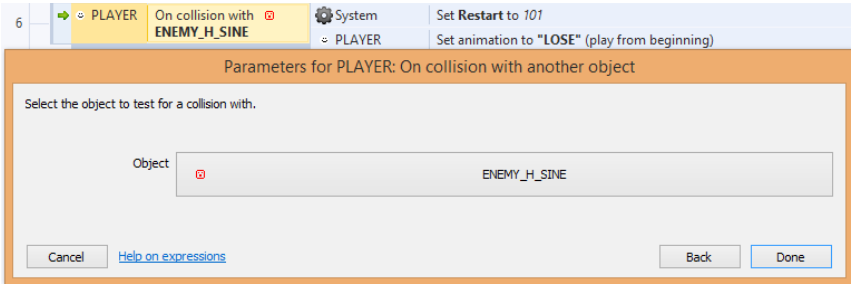


Kuva 43 Salamin kerääminen

Nyt kun pelaajaobjekti kohtaa ruudulla salamiobjektin, tuhoutuu salamiobjekti.

Luodaan seuraavaksi toiminto pelaajan ja vihollisen kohtaamiselle:

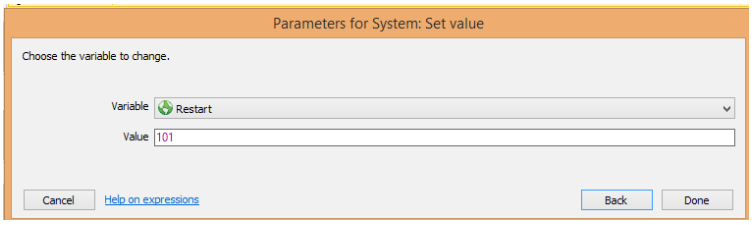
Pelaajaobjekti – on collision with → vihollisobjekti



Kuva 44, vihollisen kohtaaminen.

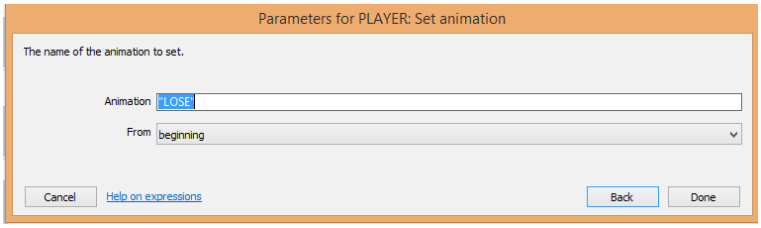
Määritellään seuraavat toiminnot:

muuttujan restart arvoon 101: System → set value restart to 101



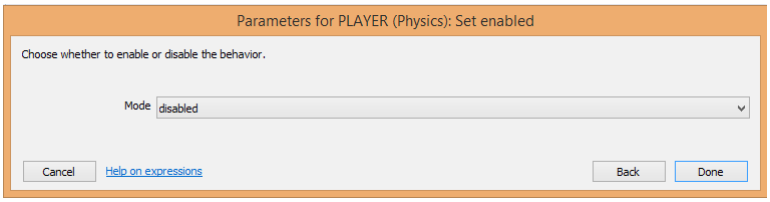
Kuva 45, kuoleman jälkeinen resetointi.

Pelaajaobjekti vaihdetaan animaation animaatioksi ”lose”: Player-> set animation to ”lose”



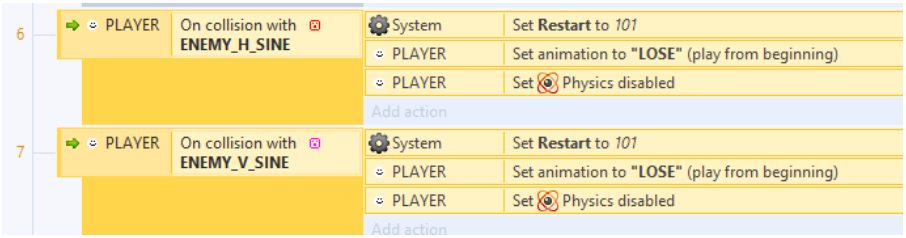
Kuva 46, kuolemisanimaatio

Pelaajaobjekti estetään fysiikkaobjektin toiminta: Player → physics → set enabled → mode disabled.



Kuva 46, fysiikkaobjektin toiminnan estäminen kuollessa

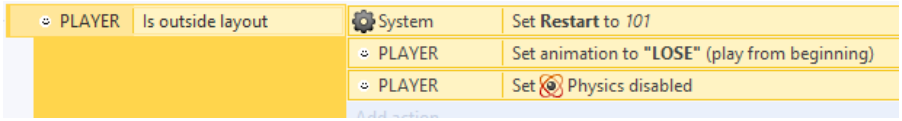
Mikäli teit useamman eri vihollisen, tee äskeiset toiminnot (esimerkiksi kopioimalla ja vaihtamalla objektin toiseen) jokaiselle vihollisobjektillesi.



Kuva 47, toimintojen kopiointi.

Tehdään samat komennet myös tilanteelle, jossa pelaajaobjekti joutuu pelialueen ulkopuolelle:

Pelaajaobjekti → is outside layout

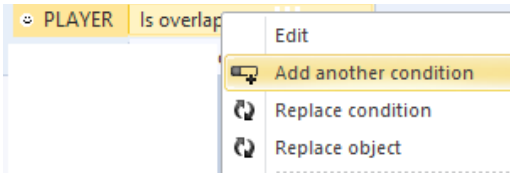


Kuva 48 pelialueen ulkopuolelle joutuminen johtaa kuolemaan.

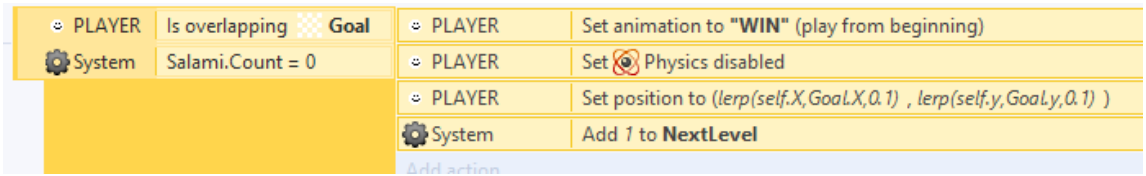
Seuraavaksi määritellään kaksoiskomento: Kun kaikki salamit ovat kerätty ja pelaaja kohtaa maaliobjektin.

Luo ensin normaalisti komento: Player → is ovelapping goal

Klikkaa luomaasi komentoa hiiren oikella painikkeella ja valitse ”Add another condition” ja lisää System → Salami.Count = 0



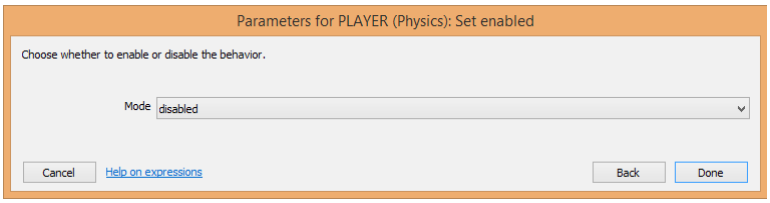
Kuva 49 , add another condition.



Kuva 50, kaksoiskomento.

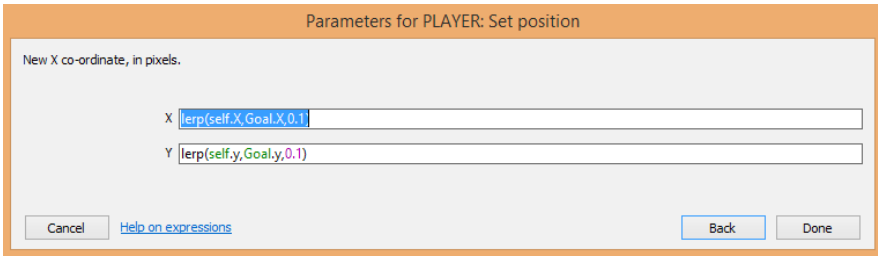
Määrittele kaksoiskomennollesi toiminnot Player → Set animation to ”win” from behinning.

Player → physics → set enabled → mode disabled.



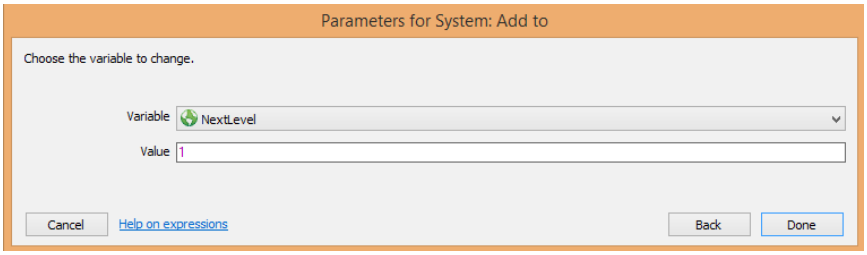
Kuva 51, fysiikkaobjektin disablointi.

Player → set position to (lerp(self.x, Goal.X, 0.1) , lerp(self.y,Goal,y.0.1)



Kuva 52, player lerp to position

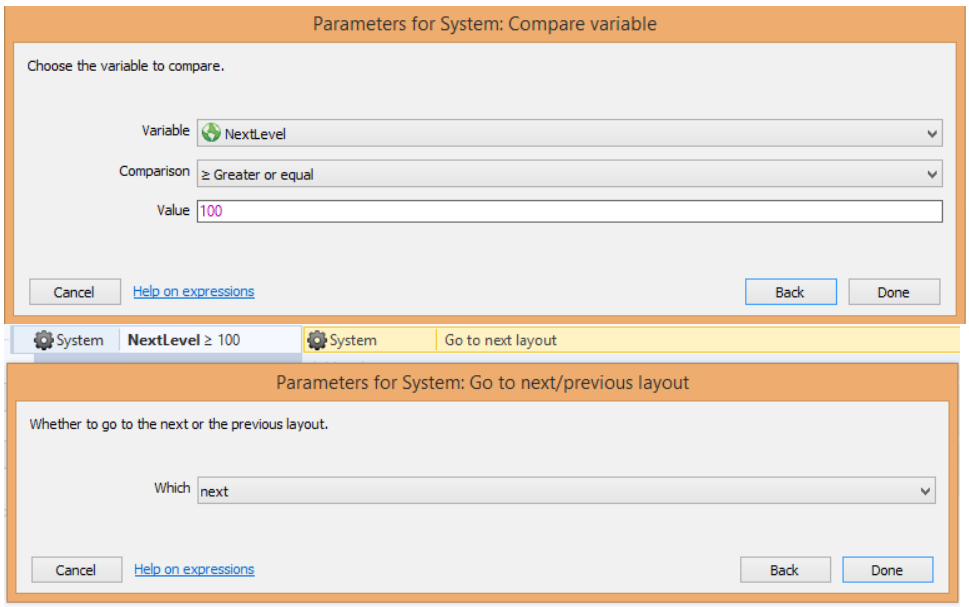
System → add to value → nextlevel → 1



Kuva 53, add 1 to nextlevel

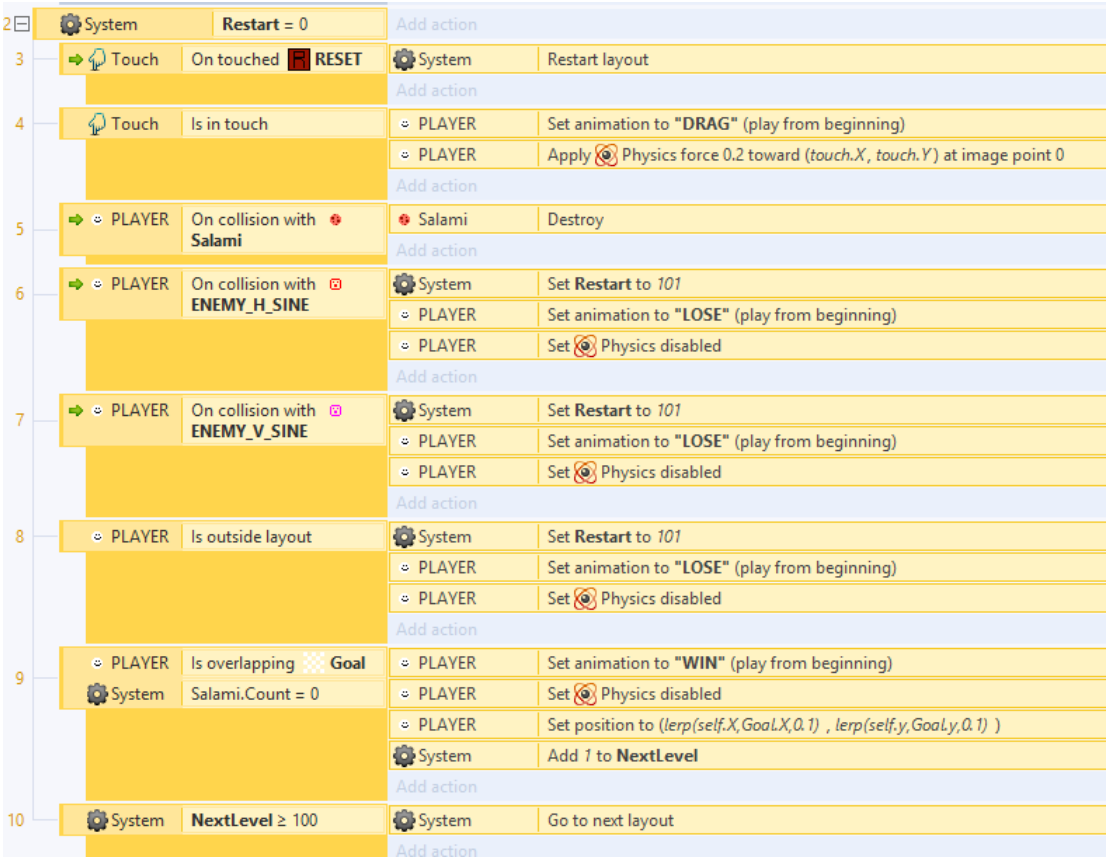
Seuraavaksi luodaan toiminto ”Jos muuttuja nextlevel => 100 siirrytään seuraavaan tasoon.

System → value next level: System → go to next layout.



Kuva 54, next layout

Näkymän tulisi näyttää nyt tältä, ja on oleellista, että nämä toiminnot ovat System → restart = 0 toiminnon alitoimintoja. Mikäli toiminnot eivät ole alikomentoja, voit vetää ja pudottaa ne kätevästi paikalleen jälkeenpäinkin.



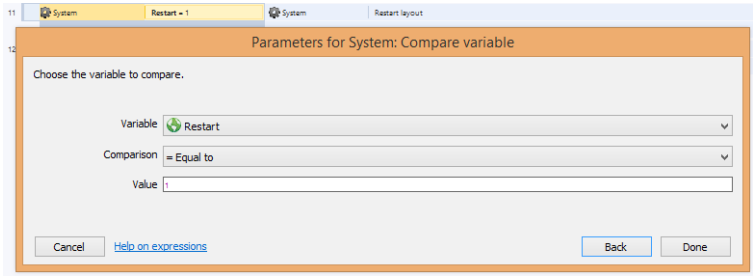
Kuva 55, komennot

Voit piilottaa nämä alitoiminnot klikkaamalla System → restart = 0 komennon vasemmalla puolella olevaa – merkkiä.

Luodaan vielä kaksi komentoa, jotka ovat irrallaan edellisistä alikomennoista.

Klikkaa add event ja luo komento → System → Value restart = 1

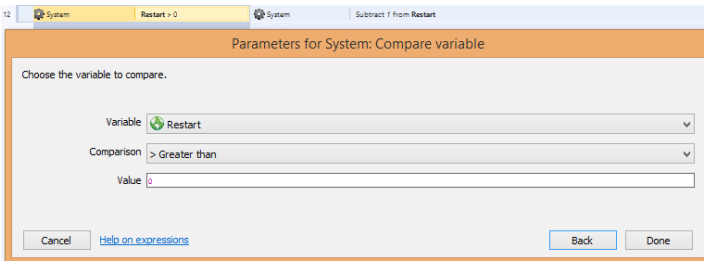
Luo sen alle toiminto: System → restart layout.



Kuva 56, jos muuttuja restart = 1, resetoi taso

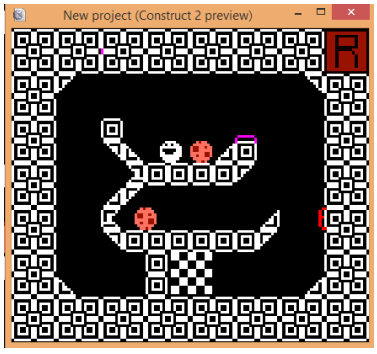
Luo vielä viimeinen uusi komento System → value restart > 0

Lue sen alle toiminto: System subtract from value restart, 1



Kuva 57 jos muuttuja restart => 0 vähennä siitä 1.

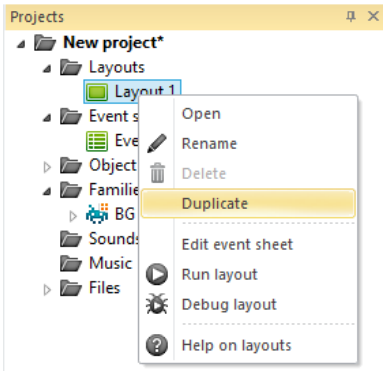
Nyt kun toiminnot ovat määritelty, voit palata aiempaan Layout 1 näkymään klikkaamalla sitä joko välilehdestä tai Projects näkymästä.Voit halutessasi sijoitella luomiasi palikoita radankaltaiseksi kokonaisuudeksi ja testata, että toiminnot toimivat.



Kuva 58, testirata.

5.7 Tasojen kehittäminen

Kun (tässä tapauksessa äärimmäisen yksinkertainen) pelilogiikka ja mekaniikka on todettu tyydyttäväksi, voit alkaa kehittämään ratoja. Helpoiten se käy luomalla Layout 1:stä klooneja.



Kuva 59, tasojen monistus.

Aiemmin kehitettyjä objekteja voi sijoittaa lukemattomin eri tavoin ja tarkoitus onkin, että uusia objekteja on helppo kehittää aiempien rinnalle kopioimalla ja luomalla uusia määritelmiä.

Ratojen kehityksen suhteen jätän lukijalle täysin vapaat kädet, leiki, kokeile, luo uutta. Voit tehdä myös pelillesi jonkinnäköisen alkuruudun ja vaikka radan valintaruudun oppaan metodeita hyödyntäen.

5.8 Apua ongelmiin

Tutoriaaleja, vinkkejä sekä muita avun ojennuksia voi lähteä hakemaan ihan googlesta, mutta Construct 2:sen valmistajan Scirran nettisivuilta ja foorumeilta saa runsaasti apua, ja jos olemassa olevaa ohjetta ei ole voi sitä muilta käyttäjiltä suoraan kysyä.

https://www.scirra.com/tutorials/37/beginners-guide-to-construct-2

5.9 Pelijulkaisun oikeudellisia kysymyksiä

Pelien tuottaminen ja julkaiseminen on entistä helpompaa, mutta on myös tärkeä tuntea omat juridiset oikeudet ja ehdot. Pelin eri versioita julkaistaessa voi olla tärkeää pitää ideoita salassa. Jos esimerkiksi teet projektia useamman henkilön joukolla, voit hyödyntää vaikkapa salassapitosopimusta. Kun työntekijä työskentelee yrityksellesi, hänen tuottama pelisisältö kuuluu yrityksellesi. Tilanne on toinen alihankkijoiden ja

konsulttien suhteen. Omaisuus pysyy tekijän hallussa, jos muuta ei ole sovittu. Kun raha astuu mukaan pelintekoon, on pelin sisällön omistussuhteen kannalta parempi käyttää freelance / palkkatyöläisiä. (Coe 2015, 1)

Tällä hetkellä mobiilipelaamisen ja selainpelaamisen tuoman räjähdysmäisen kasvun ansiosta on mahdollista levittää peliä monella eri tavalla. Niitä on esimerkiksi perinteinen vähittäismyynti, sovelluskaupat ja muut pelipalvelut.
Kolmannen osapuolen palvelua käyttäessä joudut hyväksymään heidän käyttöehtonsa. Yleensä myös pelintuottamiseen käytettävälle sovellukselle on omat ehtonsa, palvelun tuottajalla on oma kauppapaikkansa, jolla on taas omat käyttöehtonsa.(Coe 2015, 1) Esimerkiksi Construct 2 -sovelluksen käyttöehdoissa painotetaan usein lähinnä vastuun vapautusta sovelluksen tuottajan kannalta.

Seuraava lainaus suoraan Construct 2 -sovelluksen käyttöehdoista (Scirra.com) kuvaa hyvin tyypillisiä julkaisualustan käyttämiseen liittyviä ehtoja:
“” no event shall Scirra Ltd be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) rising out of 'Authorized Users' use of or inability to use the SOFTWARE PRODUCT, even if Scirra Ltd has been advised of the possibility of such damages. In no event will Scirra Ltd be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or otherwise. Scirra Ltd shall have no liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, moral rights or the disclosure of confidential information. ””

Käyttöehdoissa mukaan erimielisyydet ratkaistaan UK:n tuomioistuimen kautta.
“”7. GOVERNING LAW
All disputes are to be settled through courts in the United Kingdom. This agreement shall be construed in accordance with the laws in the United Kingdom. All parties hereby consent to the jurisdiction of the courts of the United Kingdom. If any part of this agreement is deemed unenforcable or invalid by the law, the remaining parts of the agreement will remain in effect.””

Kun henkilö alkaa tuottamaan pelejä ammatikseen, on ehtoja ja lakipykäliä hyvä tutkia tarkemmin, sillä mitään yleisiä periaatteita tai nyrkkisääntöjä ei varsinaisesti ole olemassa. Jokaisella julkaisualustalla, pelimoottorilla ja esimerkiksi sovelluskaupalla on omat ehtonsa.

5.10 Construct 2 pelin julkaiseminen yleisölle

File valikosta löytyy valinta Export Project, jonka kautta voit luoda projektistasi esimerkiksi HTML5-muotoisen nettiselaimella pelattavan version, jonka voi ftp:llä lähettää joko omalle kotisivutilalleen tai esimerkiksi HTML5 pelejä ilmaiseksi ylläpitäville pelisivustoille, kuten newgrounds.com.
Voit luoda pelistäsi myös Windows sekä muiden modernien käyttöjärjestelmien irrallisen sovelluksen, mutta se vaatii maksullisen version Construct 2 – sovelluksesta. (Scirra.com)

Julkaisupaikan suhteen valinnanvaraa on hyvin paljon. Internetistä löytyy erikokoisia toimijoita, ja aina vaihtoehtona on jakaa peliä omien sivujensa kautta. Isompiin kauppapaikkoihin oman luomuksen saaminen vaatii kuitenkin huomattavasti enemmän, verrattuna vaikkapa indie-kehittäjille suunnattuihin sivustoihin. Pienemmät sivustot ovatkin hyvä vaihtoehto, sillä niiden ympärille on monesti rakentunut kehittäjien ja pelaajien yhteisö.
Yhteisön kautta voi omasta pelistään saada arvokasta palautetta, vaikkei siitä heti hittituotetta syntyisikään. Saatu palaute auttaa pelinkehityksessä. Julkaisun voi tehdä myös useassa paikassa samaan aikaan, jolloin saatu palaute, tai tuotto voivat moninkertaistua, verrattuna yhden julkaisukanavan käyttämiseen (Lahti 2014, 35)

6 RETROPELI ”CONTRADICTION”

Opinnäytetyön varsinainen toiminnallinen osuus on ensimmäinen alusta loppuun omatoimisesti kehittämäni ja julkaisemani retropeli ”Contradiction”.
Contradiction on nimensä mukaisesti melko suora apinointi Contra nimisestä NES-konsolin klassikko toiminta-ampuilusta. Genrenä toiminta-ampuilut on erittäin suosittu. Vuonna 2016, 27.5% kaikista myydyistä peleistä oli toiminta-ampuilupelejä, kun esimerkiksi roolipelejä myytiin vain 12.9%, ja ajopelejä 3.3% kaikista myydyistä peleistä USA:n markkinoilla.

Tointa ampuilu	27,5%
Toiminta	22,5%
Roolipeli	12,9%
Urheilupelit	11,7%
Seikkailu	7,8%
Tappelu	5,8%
Strategia	4,3%
Ajopelit	3,3%
Muut	4,1%

(Statista 2017-www.statista.com-genre breakdown of video game sales in the United States in 2016)

Toinen syy miksi otin juuri tämänkaltaisen pelin työn alle, on se, että tiesin kyseisen pelimallin työmäärän laajuuden olevan huomattavasti pienempi, kuin esimerkiksi seikkailupelissä. Toivoinkin saavani aikaan ainakin neljä valmista rataa pomotaisteluineen, mutta määrä kasvoi yli odotusteni kuuteen.
Erilaisia vihollisia kehitin yhteensä 14 kappaletta. Jokaisella näistä on yksilöity graafinen ilme sekä toiminnot. Vihollisten skaala vaihtelee tekoälyn ohjaamista sotilaista eri aseineen, staattisempiin ansatyyppisiin esteisiin, kuten maasta nouseviin piikkeihin ja puista seitin vasassa pelaajan niskaan laskeutuviin hämähäkkeihin. Lopputaistelutkin ovat yksilöllisiä ja itsenäisiä kokonaisuuksia. Hiekkamato kaivautuu maasta ja pyrkii murskaamaan pelaajan, kun taas robottiseinä kourineen suojaa silmiään pelaajan ammuksilta samalla pyrkien tuhoamaan pelaajan.
Toimintapeleissä koenkin tarpeelliseksi, että pelaaja yllätetään uusilla haasteilla eikä peli toista itseään liikaa. Pelini on toiminnoiltaan yksinkertainen, mutta edellyttää

paikka paikoin useamman toiminnon yhdenaikaista suoritusta, ja onkin vaikeustasoltaan äärimmäisen haastava, kuten Nes/Snes-ajan vastaavat pelit ovat.
Jesse Schellin mukaan (2008, 195) yksinkertaisuus ja monimutkaisuus pelimekaniikassa voikin vaikuttaa paradoksiselta. Pelin kutsuminen yksinkertaiseksi voi olla kritiikkiä, kuten ilmaisu ”niin yksinkertainen että se on tylsä”. Se voi toisaalta olla myös kohteliaisuus ”Niin yksinkertaisen elegantti!” Monimutkaisuus voi myös olla kaksiteräinen miekka. Pelejä kritisoidaan ”liian monimutkaisiksi ja hämmentäviksi,” tai kohteliaisuutena kuten ”runsaan ja monipuolisen monimutkainen.” Varmistaakseen että pelissäsi on ”hyvää yksinkertaisuutta” ja ”hyvää monimutkaisuutta,” mutta ei huonoa, meidän on tarkasteltava pelien yksinkertaisuuden ja monimutkaisuuden luonnetta ja löytää oikea tasapaino niiden välillä.

Pelissäni onkin monenlaisia toiminnallisia dynaamisia elementtejä, joita pelaaja hyödyntää edetessään, kuten vaakatasossa ohjattava kiskoilla liikkuva taso, trampoliini, jonka avulla pelaaja saavuttaa korkeammat tasot sekä alta hajoavat palikat. Näitä yhdistelemällä pystyin luomaan monipuolisia tilanteita ja pientä aivojumppaa vaativia haasteita pelaajan ratkottavaksi. Kaikki ovat siis itsessään yksinkertaisia, mutta haaste muodostuukin usein vihollisten ja näiden toiminnallisten elementtien yhdistämisen myötä.
Pelaajalla on lisäksi käytössään ratoihin sijoitetuista laatikoista löytyvät aseet sekä muut bonustavarat. Eri aseet sopivat eri tilanteisiin toisiaan paremmin. Esimerkiksi haulikolla tekee helposti tuhoa eri suunnista hyökkääviin vihollisiin, kun taas laser-säde tekee yksittäiseen viholliseen enemmän tuhoa. Liekinheittimen lieska läpäisee kaikki esteet. Se mahdollistaa vihollisten muiluttamisen suojasta. Rynnäkkökiväärillä ampuu niin sarjoja, kuin laukauksiakin, ja kertakäyttöinen superpommi räjäyttää ruudun kaikki viholliset kerralla. Kun pelaaja kuolee, menettää hän hallussaan olevan aseen ja palaa joko radan alkuun, tai viimeksi haltuun otetulle lipputangolle. Kun pelaaja kuolee, tulee hän seuraavalla pelikerralla todennäköisesti saamaan eri aseet, kuin edellisellä pelikerralla. Tämä luo uudelleenpelaamisen arvoa, ja edellisen pelikerran vaikea tilanne saattaakin nyt sopivammalla arsenaalilla hoitua helpommin.
Retropeleissä on usein myös salakäytäviä ja niitä löytää ympäristöä tutkiva pelaaja Contradictionistakin.
Koen, että peli on kokonaisuudessaan valmis, Nes/Snes ajan pelin mittainen teos, jota voinkin pitää ensimmäisenä aidosti valmiina pelijulkaisunani.

6.1 Pelin esittely

6.1.1 Teema

Peli simuloi SNES-ajan grafiikkaa, resoluutio on 256x224 pikseliä ja väripaletti on 64 värin laajuinen.

Kuva 60, väripaletti.

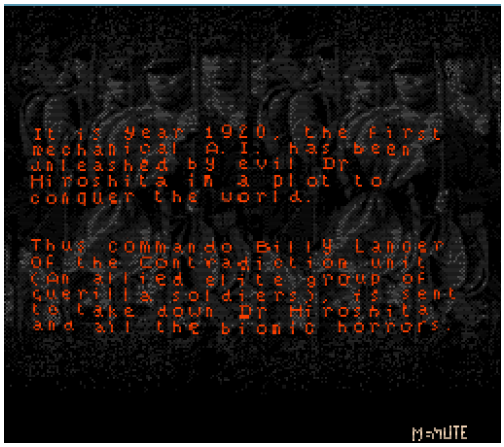
Lisäksi peli simuloi kuvaputkitelevisiota käyttäen ruudunpullistus ja scanline efekteiä.

Peli on genreltään toiminta-ampuilu. Peli on suunnattu kokeneemmille retropelaajille. Sen haastavuus saattaa rajoittaa tiettyjen pelaajien intressiä mutta samalla herättää osassa pelaajistoa kiinnostusta.

Pelin pikseligrafiikka sekä animaatiot ovat toteutettu Construct 2 – sovelluksen omalla kuvaeditorilla sekä Adobe Photoshop kuvankäsittelyohjelmalla.

6.1.2 Tarina

Tarina on aina ollut tärkeä elementti retrotoimintapeleissä. Yleensä se tosin on huomattavasti pelkistetympi kuin esimerkiksi seikkailu ja roolipeleissä. Jesse Schell (2008, 263) ilmaisee näiden pelien interaktiivisten tarinoiden olevan fundamentaalisesti erilaisia, ei-interaktiivisista tarinoista. Interaktiivisissa tarinoissa, kokija on aktiivinen, liitoksissa ja voi vaikuttaa niiden kulkuun. Oman pelini tarina on klassinen esimerkki vanhojen retropelien tarinosita.



Kuva 61, Contradiction tarina

”On vuosi 1920, paha tohtori Hiroshita on kehittänyt maailman ensimmäisen bionisen keinoälyn ja valmistanut kokonaisen armeijan koneistettuja sotilaita valloittaakseen maailman. Siispä Billy Lancer Contradiction Unitista (yhdistyneiden guerillasotilaiden liittoutuma) lähetetään omatoimisesti estämään Hiroshitan aikeet.”

6.1.3 Sankarimme Billy Lancer

Toiminnot

Juoksee, hyppii ja kyykistyy. Ampuu kahdeksassa kulmassa. Pystyy hyppäämään alas kielekkeiltä.

Ohjaus

Peliä ohjataan näppäimistöllä.

Oletuspainikkeet ovat nuolinäppäimet sekä z ja x painikkeet.

Nuolinäppäimillä ohjataan pelaajan suuntaa, liikettä sekä kyykistymistä.

Pelaaja ampuu Z painiketta pohjassa painamalla tai vaihtoehtoisesti painelemalla.

X painikkeesta pelaaja oletusarvoisesti hyppää, mutta tiettyjen tasojen päällä yhdistelmä NUOLI ALAS + X saa pelaajan pudottautumaan tasolta alas. Tämä on tyypillinen retrotasoloikka toiminto.

Kuva 62, Billy Lancer sheet

6.1.4 Kerättävät Bonukset

Kuten retropeleissä on tapana, on pelissäni ratoihin sijoiteltuja kerättäviä objekteja jotka vaikuttavat pelitilanteeseen. Objektit ovat sijoiteltu hyödyntäen satunnaisuusalgoritmiä. Täten jokaisella pelikerralla pelaaja saa kerättyä eri tilanteissa eri bonus-tavaroita. Tietyissä tilanteissa tosin esiintyy kiinteitä bonuksia, kuten lisäelämä sijoittuna salahuoneeseen. Lisäksi bonustavarat tukevat pelaajan palkitsemista. Jokainen pelaaja pitää siitä että heille ilmaistaan, että he tekevät hyvää työtä. (Schell 2008, 191). Siksi esimerkiksi lisäelämän poimiessa, toteutuu väreillä, äänillä sekä tekstin muodossa palautetta, josta ilmenee että näitä tosiaan kannattaa keräillä.

Kuva 63, Bonustavara-spriten kuvat.

Aseet

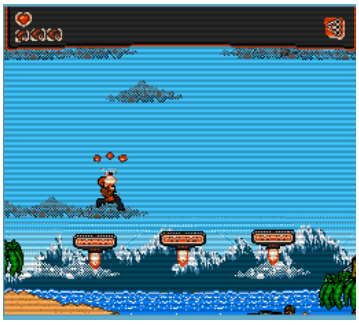
Sarjatuli

Ampuu lyhyitä sarjoja.

Kuva 64, sarjatuli

Haulikko

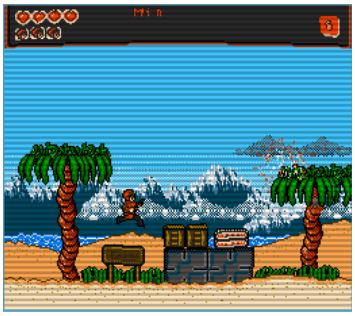
Ampuu 3 ammusta, 1 suoraan ja 2 kulmassa.



Kuva 65, haulikko

Granaatti

Heittää räjähtävän granaatin.



Kuva 66, granaatti

Liekinheitin

Ampuu lyhyen kantaman liekkiä, joka menee läpi seinistä.

Kuva 67, Liekinheitin

Laser

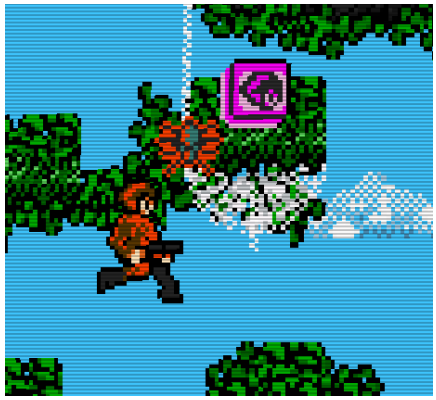
Ampuu tehokkaan laser-säteen. Tällä aseella on suurin viive.

Kuva 68, Laser

Muut bonustavarat

Lisäelämä

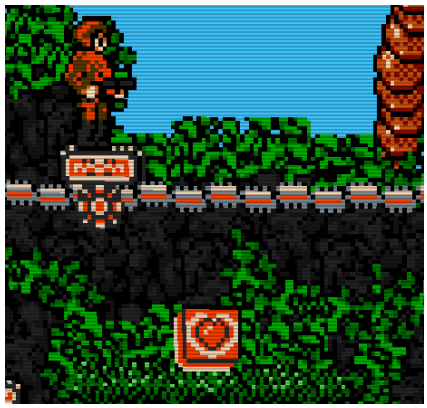
Pelaaja saa yhden lisäelämän.



Kuva 69, lisäelämä

Sydän

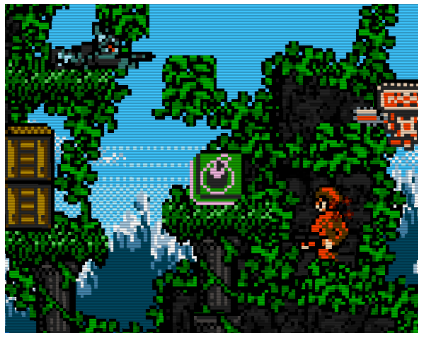
Pelaaja saa yhden osumapisteen. Kun pelaaja kohtaa vihollisen menettää pelaaja ensin osumapisteitä. Kun osumapisteet ovat nollassa menettää pelaaja yhden elämän.



Kuva 70, sydän

Megapommi

Tuhoaa kaikki ruudulla olevat viholliset, lukuunottamatta pomoja.



Kuva 71, megapommi

6.1.5 Käyttöliittymä

Jotta Schellin (2008, 195) ”hyvän monimutkaisuuden” ja ”hyvän yksinkertaisuuden” periaate toteutuu työssäni, on pelin käyttöliittymä minimaalisen yksinkertainen. Jättäen varaa pelitilanteiden monipuolisuudelle. Ohjaus tapahtuu näppäimistöllä ja peli tarjoaakin pelaajalle pelinäkymän lisäksi ainoastaan pienen alueen ylälaidasta, josta ilmenee osumien määrä (Sydämet) sekä elämien määrä (Naamat sydänten alla). Keskiosa on varattu pelin teksimoottorille, joka ilmaisee pelihahmon tunteita, tilanteita pelissä, jne. Oikea reuna ilmaisee pelaajan hallussa olevan aseen.

Kuva 72, käyttöliittymä

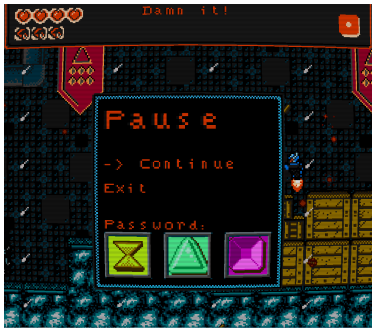
Alkuruudusta pelaaja pääsee salasanansyöttöön sekä aloittaa uuden pelin yksinkertaisesti valitsemalla nuolinäppäimillä jommankumman vaihtoehdon ja painamalla enter-painiketta.

Kuva 73, introruutu

Ennen kunkin radan alkua pelaajalle esitetään karttanäkymä. Se heijastaa samalla tulevan radan teemaa, tukien pelaajan ja tarinan suhdetta. Näin pelaaja tietää missä päin saarta hän tällä hetkellä kulkee. Samalla ilmaistaankin tulevan radan salasana.

Kuva 74, kartta

Pelin aikana pelaaja voi painaa joko ESC tai ENTER näppäintä laittaakseen pelin paussille. Samalla ruudulla aukeaa valikko, josta pelaaja voi joko jatkaa peliä tai poistua takaisin alkuruutuun. Koin myös oleelliseksi ominaisuudeksi esittää pelaajalle käynnissä olevan radan salasana tässäkin näkymässä.



Kuva 75, paussi

6.1.6 Viholliset

Jalkaväki

Puolielävä bioninen sotilas on joko hyökkäys tai puolustustilassa. Osaa hyppiä, ampua sekä maastoutua.

Kuva 76, jalkaväkisotilas

Granaatinheittäjä

Pysyy paikallaan ja heittää tasaiseen tahtiin granaatteja pelaajan suuntaan.

Kuva 77, granaatinheittäjä

Tarkka-ampuja

Ampuu tasaisin väliajoin vaakatasossa pelaajan suuntaan. Pysyy paikallaan, joskus hyödyntäen esimerkiksi pusikon suojaa.

Kuva 78, tarkka-ampuja

Birasotilas

Pysyy paikallaan, ampuu bira henkisellä konekiväärillä lyhyitä sarjoja.

Kuva 79, birasotilas

Tykkimies

Liikkuu hitaasti vaakatasossa ja ampuu yläviistoon pelaajaa kohti hakeutuvia ammuksia.

Kuva 80, tykkimies

Rakettireppusotilas

Pyrkii pitämään lentäen välimatkaa pelaajaa ampuen samalla pelaajan suuntaan.

Kuva 81, Rakettireppusotilas

Sala-ampuja

Kurkistaa luukusta ja ampuu kohti pelaajaa.

Kuva 82, sala-ampuja

Ninja

Juoksee kohti pelaajaa suurella nopeudella, lyö tappavalla samuraimiekalla hypätessään.

Kuva 82, ninja

Mutanttiapina

Tämä Hiroshitan kehittelemä mutanttiapinalaji himoaa verta ja puolustaa reviriään säntäämällä ympäriinsä ja hyökähdellen kohti pelaajaa.

Kuva 83, mutanttiapina

Vampyyrilepakko

Hiroshitan tarkkaan valittu, erityisen verenhimoinen lepakkolaji esiintyy saarella suurissa määrin. Lentää sivuttaissuunnassa ja hyödyntää pystysuuntaista sine-aaltoa liikkeessään. Ei välitä esteistä.

Kuva 84, vampyyrilepakko

Tarantula

Tämä myrkyllinen hämähäkki laskeutuu seittinsä varassa kohti pahaa-aavistamatonta pelaajaa.

Kuva 85, tarantula

Piikkiansa

Nämä neulan terävät teräspiikit nousevat maasta tasaisin väliajoin.

Kuva 86, piikkiansa

Sahanterä

Rattaan varassa pyörivät veitsenterävä sahanterä irrottaa jäsenen hetkessä. Tämän esteen voi joko kiertää tai tuhota.

Kuva 87, sahanterä

Miina

Pelaajan osuessa miinaan, se räjähtää. Miina kannattaa tuhota sopivalta etäisyydeltä.

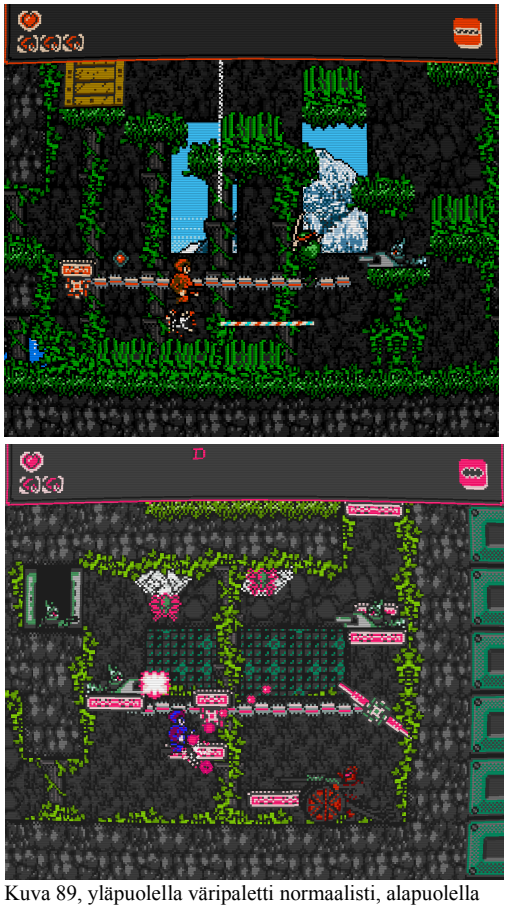
Kuva 88, miina

Tyypillisesti pelaaja kohtaa kerralla useampia vihollisia.

Kun pelaaja loukkaantuu ilmaistaan se pelaajalla äänitehosteen, veriroiskeen sekä erilaisten väripalettien hallintaan liittyvien efektien muodossa.

Kun pelaaja saa elämän ruudulla lukee 1-up ja peli toistaa palkitsevan äänitehosteen ja valaisee ruudun hetkeksi.

Olen huomannut, että pelin miellyttävyyteen voi vaikuttaa paljonkin näinkin pieniltä tuntuvilla asioilla.

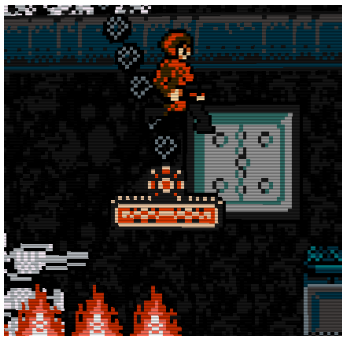


Kuva 89, yläpuolella väripaletti normaalisti, alapuolella väripaletin nopea muutos ilmaisee pelaajalle virheestä.

6.1.7 Toiminnallisia elementtejä

Ketjukiikku

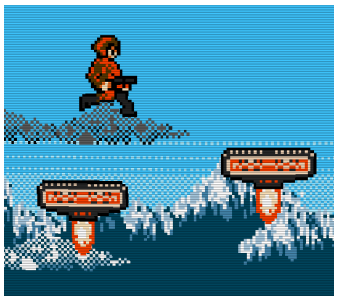
Perinteinen tasoloikkatyylinen kiikku, joka heiluu vaakatasossa tasaiseen tahtiin.



Kuva 90, ketjukiikku

Rakettitaso

Tämän tason moottori saa sen juuri ja juuri leijumaan ilmassa. Pelaajan seistessä tason päällä, laskeutuu se hitaasti alaspäin. Kun pelaaja ei ole tason päällä leijuu se takaisin lähtöasemiinsa.



Kuva 91, rakettitaso

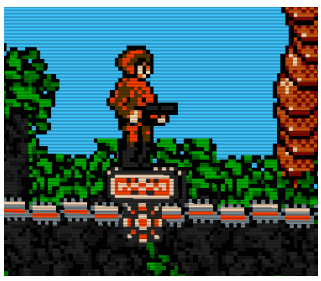
Hajoava taso

Nämä taso hajoavat pelaajan painosta.

Kuva 92, hajoava taso

Ketjuhissi

Tämän päällä ollessaan, pelaaja ottaa hissin hallinnan ja voi liikuttaa sitä koko ketjun mitalta vaakatasolla.



Kuva 93, ketjuhissi

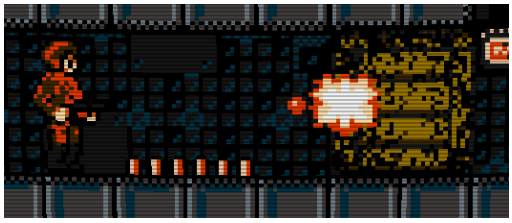
Trampoliini

Mahdollistaa korkeiden tasojen saavuttamisen.

Kuva 94, trampoliini

Iso laatikko

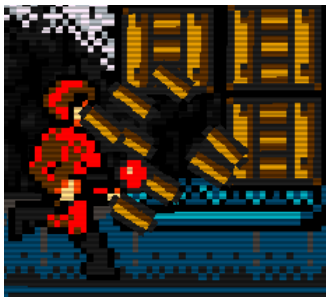
Saattaa sisältää joko miinoja, aseita tai bonuksia.



Kuva 95, iso laatikko

Pieni laatikko

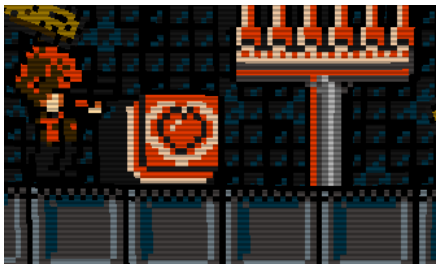
Vastaava kuin iso laatikko, pienemmässä koossa.



Kuva 95, pieni laatikko

Murskain

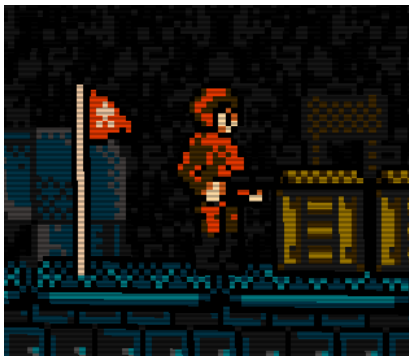
Ilmestyy taustaelementtien takaa, tätä vihollista et voi tuhota, sen voi vain kiertää, ylittää tai alittaa.



Kuva 96, murskain

Lipputanko

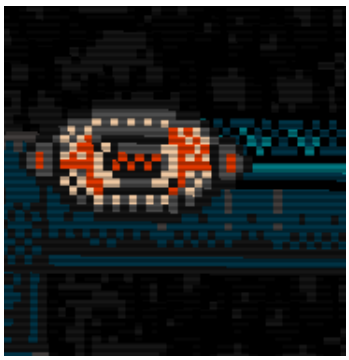
Kun pelaaja saavuttaa lipputangon radassa, tallentaa peli uudeksi kuoleman jälkeiseksi aloitussijainniksi viimeisimmän lipputangon sijainnin.



Kuva 97, lipputanko

Bonuspallo

Tiputtaa tuohoutuessaan satunnaisen bonus-tavaran.



Kuva 98, bonuspallo

6.1.8 Radat

Ratasuunnittelu on erilaista jokaista eri peliä kehittäessä, sillä jokainen peli on erilainen. (Schell 2008, 343).

Pelini sijoittuu Tohtori Hiroshitan saarelle, joka on trooppinen vuoristosaari. Tätä ajatusta olen pyrkinyt ammentamaan ratasuunniteluni visuaaliseen antiin.

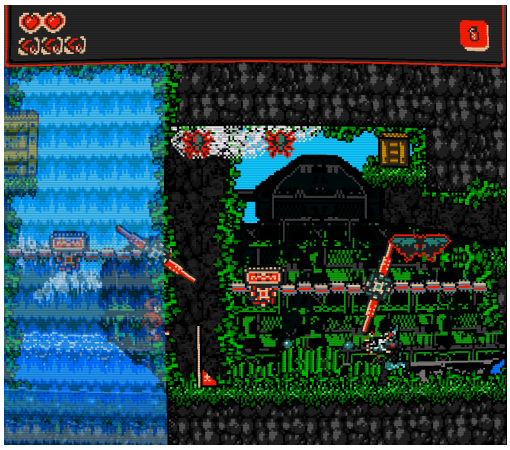
Rantautuminen saarelle

Kevyt aloitus. Pelaaja kohtaa ainoastaan osan perusvihollistyypeistä, joita on sijoiteltu vähän vähemmin tähän ensimmäiseen tutustumisrataan. Radan lopulla pelaaja kohtaa jättimäisen hiekkamadon.

Kuva 99, rantautuminen saarelle

Trooppinen vyöhyke

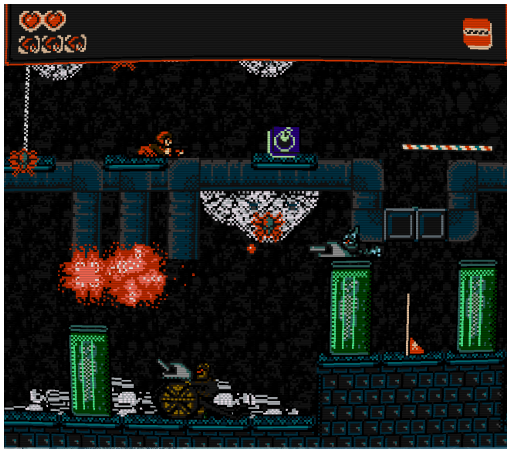
Siirtymä rannalta trooppiselle vyöhykkeelle, jossa palmut ja vihaiset apinat sekä ninjat ja muut kauhistuttavat asiat odottavat pelaajaa suurempina volyymeinä. Mikä mahtaa odottaa pelaajaa tämän kaiken jälkeen?



Kuva 100, trooppinen vyöhyke

Käytävä

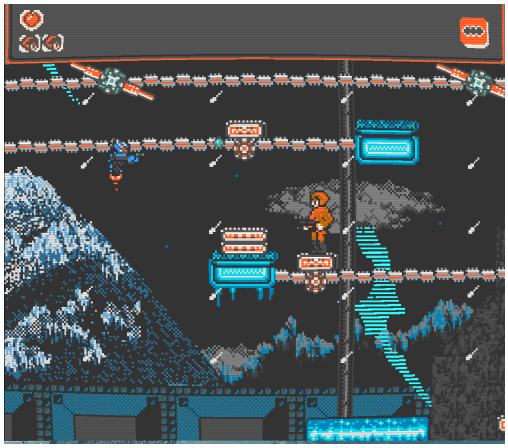
Maan-alainen tunneli johtaa kohti saaren ydintä, hämähäkinseittien takaa pelaaja kohtaa tohtori Hiroshitan hirveydet.



Kuva 101, käytävä

Vuoren valloitus

Tunneli päätyy vuoren huipun juurelle, suunta on vain ylöspäin. Korkealta putoaminen tietää tässä kentässä kuolemaa.



Kuva 102, vuoren valloitus

Pahuuden luola

Vuoren huipulta jälleen kohti maan uumenia aukeaa rakennettu luolasto, joka on täynnä toisiaan hirveämpiä esteitä ja vihollisia.

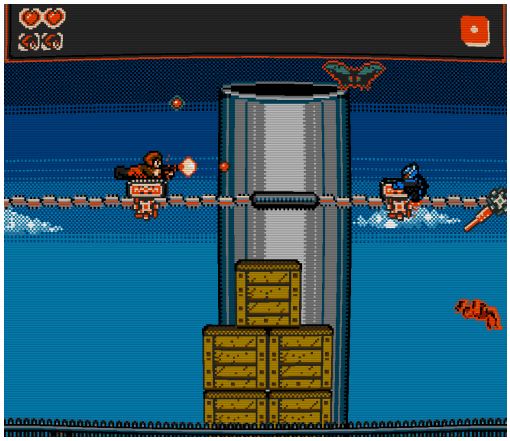
Puolivälissä rataa tulee kuumat paikat.

Kuva 103, pahuuden luola

Pakojahti

On aika poistua saarelta, mutta kuka jahtaakaan ketä?

Rata alkaa rannalta ja tie kulkee kahden sotalaivan halki. Toisen sotalaivan kannella alkaa tapahtua.



Kuva 104, pakojahti

6.1.9 Pomot

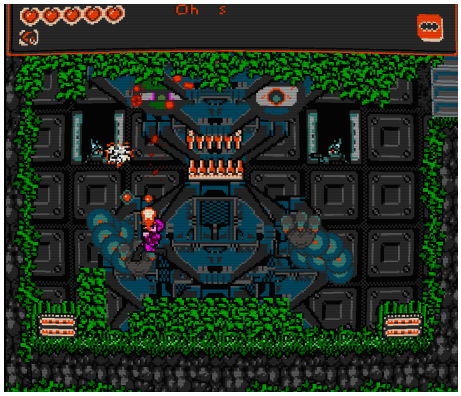
Hiekkamato

Näiden bionisten, puolielävien koneiden avulla hiroshita on kaivertanut saarensa täyteen erilaisia käytäviä vuosien saatossa. Tätä matoa ei noin vain laiteta koukkuun.

Kuva 105, hiekkamato

Robottiseinä

Onko se kone, jos se elää?
Kuoren alta paljastuu jotain entistä hirveämpää.



Kuva 106, robottiseinä

Bioninen tulilepakko

Hiroshitan vanha eläinkoe, jossa hän on saanut lepakon kuolan reagoimaan ilmassa niin, että se syttyy palamaan. Eläin on ollut kärsimyksissään vangittuna jo vuosia.

Kuva 107, bioninen tulilepakko

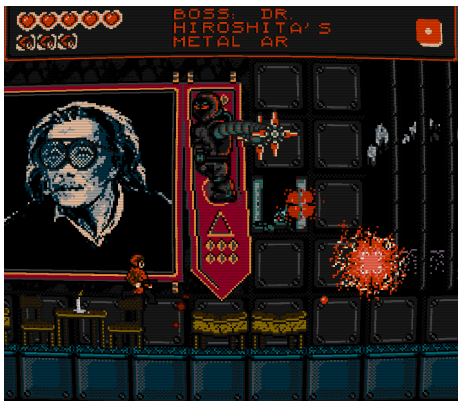
Panssaripataljoona

Jotta aika ei kävisi tylsäksi, pelaajalla on vastassa ohjautuvia ammuksia ampuva panssarivaunu ja ruutu täynnä ninjoja ja sala-ampujia. Lisäksi panssarivaunulla on käytössä Hiroshitan kehittämä laser-suojakilpi.

Kuva 108, Panssaripataljoona

Tohtori Hiroshitan biofyysinen erikoispanssari

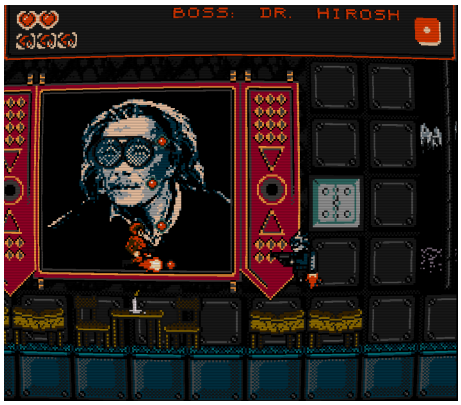
Tämän exoskeletonin avulla hiroshita laittaa todellakin haisemaan. Rakettisaappaiden avulla Hiroshita hyppää yli pelaajan helposti, tavoitteenaan murskata hänet moottorisahakourinensa.



Kuva 109, Tohtori Hiroshitan biofyysinen erikoispanssari

Tohtori Hiroshita

Kun exoskeleton tuhoutuu, on hiroshitalla vielä käytössään jetpack, granaatteja sekä perinteikäs haulikko. Lisäksi hän kantaa mukanaan laser-suojakilpigeneraattoria.



Kuva 110,Tohtori Hiroshita

Lentokonetaistelu

Kun pelaaja vihdoin saa pakoon pyrkivän roiston taas tähtäimeensä, tapahtuukin se yläilmoissa. Tämä on pelin kliimaksi ja loppu.

Kuva 111, lentokonetaistelu

6.1.10 Musiikki

Pelimusiikki on retropelien yksi tärkeimmistä ominaisuuksista. Useimmat pelit tunnistetaankin pelkän musiikin perusteella helposti. Kaikki tuntevat Super Mario:n tunnarin ja moni pystyy hyräilemään Mega Man 2 :sen Tohtori Wily-radan teemaa suvereenisti.

Contradictionin pelimusiikista vastaavat ammattimuusikot Nuutti Hannula, Ilari Hannula ja Erik Purdon.

Yhteistyö hoidettiin paikallisten sessioiden sekä etäyhteyden muodossa.

Pelissä on 14 itsenäistä kappaletta, eli enemmän kuin keskiverto pop-albumissa.

Jokaisesta kappaleesta on sovitettu eri versioita ja parhaiten rataan sopivat ovat valittu lopulliseen tuotokseen ja nämä valitut kappaleet ovat vielä miksattu ja masteroitu lopulliseen muotoonsa.

Seuraava haastattelu avaa pelimusiikin luomisprosessia:

Haastateltavana: Nuutti Hannula, Helsinki

Kuinka pelimusiikin laatiminen eroaa pop-musiikin laatimisesta?

Pelimusiikin säveltämisessä on paljon samoja piirteitä kuin pop-musiikin tekemisessä. Kyseessä ei ole kuitenkaan pelkkään musiikkin keskittyvä taidemuoto, joten kappaleiden on tuettava pelin kenttien tunnelmaa. Tämä on merkityksellistä varsinkin sovitusvaiheessa.

Mitä haasteita kohtasit?

Suurin haaste oli löytää oma tapa säveltää pelimusiikkia. Tein tutkimustyötä kuunnellen referenssikappaleita ja hakemalla inspiraatiota nuotinnuksista. Vaati kuitenkin muutaman kuukauden, että prosessi alkoi sujumaan vaivattomammin.

Mitä uutta opit projektin aikana?

Opin kokonaan uudenlaisen tavan tehdä musiikkia ja myös samalla monia uusia metodeja konemusiikin tekemiseen, joka ei rajoitu palkästään pelimusiikin säveltämiseen.

Mistä taidoista voisi olla hyötyä tulevien projektien kannalta?

Opin käyttämään Logig X sekvensseriä huomattavan paljon paremmin tämän prosessin aikana. Tästä on varmasti hyötyä myös tulevaisuudessa.

Mitä työkaluja käytit?

Käytin työkaluinani Logig pro X sekvensseriä, M-Audion sekä Casion kosketinsoittimia sekä Akain Mpd-218 rumpukontrolleria. Tämän lisäksi käytin sävellysvaiheessa myös akustisia sekä sähköisiä kitaroita.

Kuinka monta tuntia projektiin käytit?

Sanoisin, että projektin valmiiksi tekemiseen meni 150-200 tuntia.

Miten onnistuit?

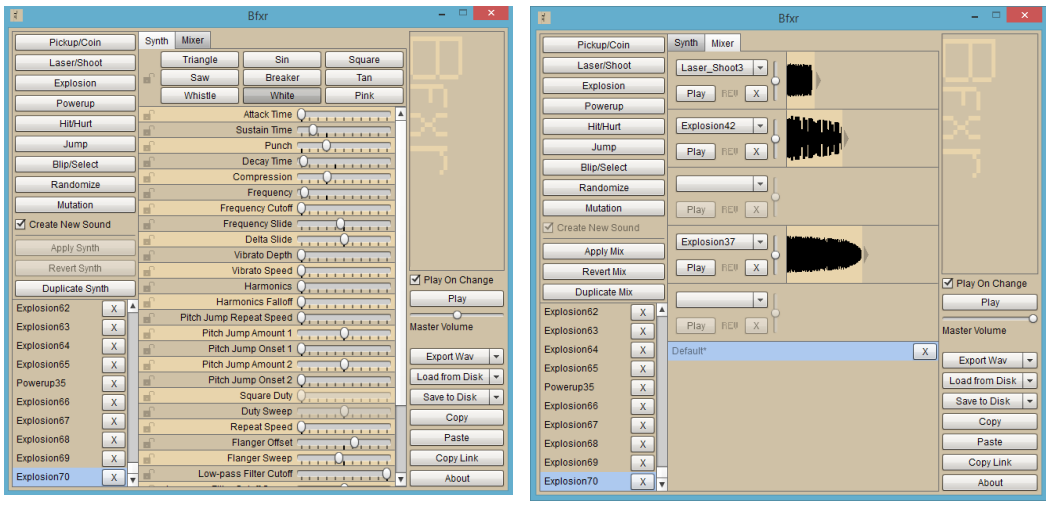
Mielestäni projekti onnistui erittäin hyvin. Musiikki ja peli sopivat hienosti yhteen.

Ketkä ovat pelimusiikin tärkeimmät esikuvasi?

Koji Kondo, Takashi Tateishi ja Kenji Yamamoto.

6.1.11 Äänitehosteet

Äänitehosteet ovat luotu Bfxr – sovelluksella. Eri efektejä loin yhteensä 30 kappaletta, jotka hyödyntävät monipuolisesti bfxr:n eri ominaisuuksia. Sovellus tarjoaa vasemmalla esimerkiksi satunnaisesti generoituvia hyppyäänen perusmalleja, mutta joitakin tehosteita oli kehitettävä ihan harkinnalla.



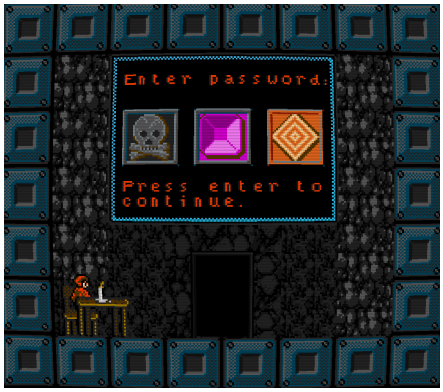
Kuva 112, Bfxr retroäänitehostesovellus

Mikseriválilehdellä voit yhdistää aiemmin luomiasi tehosteita useamman raidan kokonaisuuksiksi, joka monipuolistaa mahdollisuuksien kirjoa huomattavasti.

Sovellus on täysin ilmainen ja sen avulla kehitetyt äänitehosteet ovat käytettävissäsi ilmaiseksi, voinkin lämpimästi suositella jokaista retropelien kanssa puuhastelevaa tutustumaan tuotteeseen.

6.1.12 Retrohenkinen salasana-systeemi

Peliin olisi voinut kehittää session tallennusjärjestelmän, mutta halusin kokeilla vanhanaikaisen salasanajärjestelmän luomista ja se tuntuikin sopivan pelin henkeen erityisen hyvin. Käytännössä tämä hoituu kolmella muuttujalla, joilla on 9 eri arvoa.



Kuva 113, salasanan syöttö

6.2 Pelinkehitysprosessi

Pelinkehitysprosessi alkoi jo vuosia ennen opinnäytetyön kirjoittamista. Alkuperäisen mockup version pelistä julkaisin vuonna 2015 newgrounds palveluun. Pelissä oli vain yksi rata ja hieman kankeat kontrollit, mutta se herätti kiinnostusta ja rataa pelattiin yli odotusten joitakin tuhansia kertoja. Peli herätti eniten mielenkiintoa mockup peliprojekteistani, joten päätin ottaa sen työn alle ensimmäiseksi valmiiksi pelikseni.

Aloitin ensisijaisesti pelin perustoimintojen hiomisen. Suuri apu oli erittäin NES-kriittisen ystäväni Ville Vuoksialan suora ja estoton palaute ja korjauskehoitukset, joihin pyrin edelleenkin kiinnittämään huomiota. Tässä haluankin korostaa sitä, että omaan työhön tottuu ja tulee sokeaksi myös pelinteossa. Tuntuu ikävältä muuttaa täysin osa-alue omasta pelistä, johon on käyttänyt jo kymmeniä työtunteja, mutta yleensä se lopulta kannattaa, sillä mielestäni nämä muutokset ovat selkeästi vaikuttaneet pelin miellyttävyyteen positiivisesti.
Kun peli mekaanisesti tuntui toimivan tyydyttävästi aloin tuottamaan sisältöä uusien ratojen, vihollisten ja pomotaisteluiden muodossa.

Testaus on pelinkehityksessä todella tärkeää, jotta saadaan varmistettua pelin toimivuus pelaajan näkökulmasta ja minimoitua pelissä olevia virheitä.(Anglé 2014, 18)

Laatutestaus ei liity mitenkään siihen, onko peli nautittava vai ei, vaan keskittyy täysin bugien ja virheiden etsintään. (Schell 2008, 390)

Käytettävyystestauksella testataan sitä, että käyttöliittymä ja tuotteen systeemit ovat intuitiiviset ja helppokäyttöiset. Molemmat näistä ovat tarpeellisia nautittavan pelikokemuksen kehittämisen kannalta. (Schell 2008, 390)

Pelitestauksella taas tarkoitetaan sitä, että saatetaan pelaajat pelin ääreen ja katsotaan, kuinka pelin suunnitellut ominaisuudet toimivat suhteessa todelliseen pelikokemukseen. (Schell 2008, 390)

Testauksen on tarkoitus varmistaa ohjelman laatu, jotta ohjelmisto täyttää kaikki tarvittavat laatuvaatimukset.(Anglé 2014, 17)

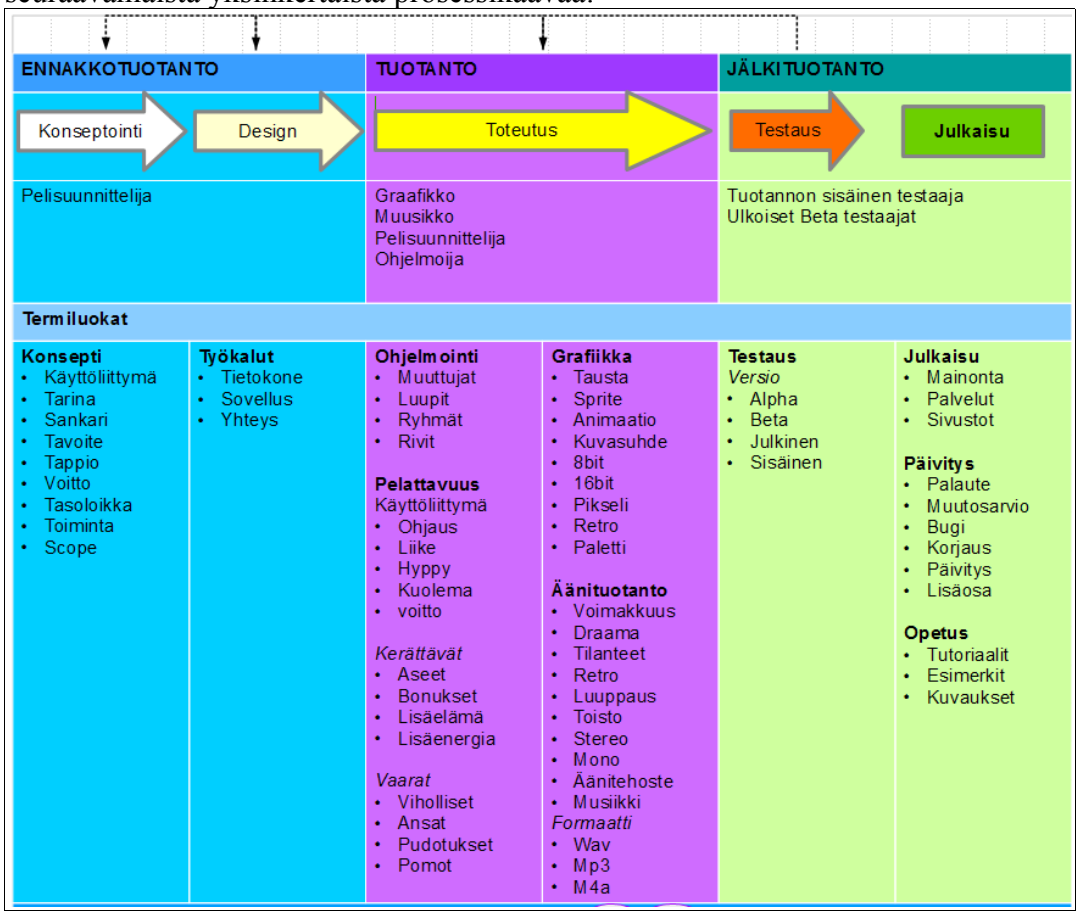
Omassakin projektissani, jokaista uutta asiaa piti testata useassa eri olosuhteessa, eri aseella, eri suunnista jne. Päätin, että pelini pelaamisen tulisi olla miellyttävää ja kontrollien piti olla alusta loppuun pelaajan hallussa.

Pelissä on tärkeää saada pelaaja ymmärtämään pelin logiikka mahdollisimman nopeasti. Jos pelaaja ei opi peliä tarpeeksi nopeasti, pelaaja yksinkertaisesti lopettaa pelaamisen eikä enää palaa.(Anglé 2014, 18)

Suurin osa testaamisesta tuli tehtyä itse, mutta seuraavissa projekteissani pyrin delegoimaan testaamistyötä jollekin yhteisölle, jos joku vaan on siihen valmis osallistumaan.

Testauksen ansiosta virheiden määrää saadaan tiputettua mahdollisimman alas, jotta julkaisuun saadaan mahdollisimman hyvin toimiva peli.(Anglé 2014, 19)

Tarkkaa työtuntimäärää mitä itse olen peliä kehittäessä en ole laskenut, mutta raaka arvio lopullisten viilausten ja hiomisten jälkeen lienee tuhannen tunnin paikkeilla. Peliä kehittäessäni koin oleelliseksi järjestää työni määrää, joten seurasin seuraavanlaista yksinkertaista prosessikaavaa:



Kuva 114, prosessikaava

6.3 Julkaisu, palaute, jälkitoimet

Pelin virallista julkaisua ennen, kehitän pelistä vielä pelivideon, joka toimii mainoksena youtube:ssa sekä sosiaalisessa mediassa. Lisäksi pelille on rakenteilla omat promootioverkkosivustot, josta peliä pääsee suoraan pelaamaan HTML5-muodossa.

Mikäli peli herättää kiinnostusta harkitsen siitä Steam Green Light version kehittämistä. Steam lienee tunnetuin tietokonepeliportaali ja Greenlight antaakin pelintekijälle mahdollisuuden julkaista pelinsä laajalle asiakaskunnalle.

> *Greenlightiin käyttäjä voi laittaa pelistään kuvia, videota ja muuta infoa. Steamin käyttäjät pystyvät antamaan palveluun laitetuille peleille ääniä ja antaa kehittäjälle palautetta pelistä. Mikäli peli saa tarpeeksi huomiota, sen on mahdollista saada julkaisulupa Steam-palveluun. (Lahti 2014, 36)*

Kuva 115, Steam Greenlight

Mikäli Steam yhteisö ei pelistä kiinnostu, lähetän pelin useammalle, ilmaiseksi pelejä ylläpitävälle pelisivustolle kuten newgrounds.com, html5games.com, kongregate jne.

Kuten Lahti (2014, 38) toteaa, useammassa paikassa pelin julkaiseminen laajentaa saadun palautteen sekä mahdollisten tuottojen määrään. Lisäksi niitä ympäröivän yhteisön palaute nostaa esiin pelin hyvät ja huonot puolet varmemmin kuin yhtä julkaisukanavaa ainoastaan hyödyntäen.

Kehitän peliä kohtaamani palautteen mukaan, tarvittaessa vielä julkaisun jälkeen. Suurin osa bugeista ja ongelmatilanteista on onneksi jo karsittu sitkeän pienen testausporukkani ansiosta.

7 POHDINTA

Ajatus retropelinkehitystä käsittelevästä opinnäytetyöstä syntyi jo opiskeluni alkutaipaleella. Pelinkehitys on ollut harrastuksenani lähes kaksikymmentä vuotta ja halusin hyödyntää omaa osaamistani mahdollisimman laajasti. Olen tehnyt pelejä siis vain harrastuspohjalta, joten varsinaisia deadlinejä tai muitakaan paineita projektin valmiiksi saattamisessa ei varsinaisesti ole ollut. Siksi koin erittäin hyväksi ideaksi alkaa kehittämään tällaisesta raakileesta kokonaista peliä, jonka deadlinena toimisikin opinnäytetyöni julkaisupäivä.

Tämän päätöksen jälkeen raakileeni on kuin onkin muotoutunut jopa yllättävän valmiilta tuntuvaksi peliksi, enkä ole siitä löytänyt virheitä tai välitöntä korjausta vaativaa tilannetta.

Omaehtoinen deadline nopeutti projektin etenemistä, ja työskentelylleni tuli selkeä ”tee nyt kaikki minkä voit” formaatti.

Yllätyin, kun ratojen valmiiksi kehittämisen jälkeen jäi aikaa vielä graafisen ilmeen ja yleisen viilauksen sekä jopa parin uuden vihollistyypin luomiseen.

Valmiin pelin omatoiminen kehittäminen on kuitenkin projekti johon en varmasti tästä opittuani enää hetkeen ryhdy, vaan pyrin saamaan ympärilleni joukon ihmisiä, jotka ovat valmiita jakamaan työtaakkaa joko yritys tai harrastemielessä.

Peliprojektin kuvaaminen tekstin muodossa osoittautui myös mielenkiintoiseksi projektiksi. Sen lisäksi, että tekstiä kirjoittaessa kehitin opinnäytetyötä, koin kehittäväni samalla käsitekarttaa omasta työskentelystäni pelien parissa. Tulenkin hyödyntämään pelinkehityspäiväkirjaa tulevien peliprojektieni tukena jatkossa.
Omakohtaisesti voin hyvinkin todeta että HTML5-retropelinkehitys on mahdollista omaehtoisesti, mutta suuren työmäärän takana. Mikäli pelinkehittäjän on mahdollista verkostoitua osaksi pelintekoryhmää, realisoituu laajempienkin kokonaisuuksien kehittäminen, alusta loppuun. Itseopiskelu on täysin mahdollista, ja omien voimavarojen selvittäminen ja niiden tukeminen uusilla opituilla asioilla avaa nopeasti ovia pelinkehityksen eri tehtäviin.

Jatkotutkimusta näkisin tarpeelliseksi tehdä yritysmaailman näkökulmasta sekä, kuinka realistista on toteuttaa taloudellisesti kannattavia tuotteita opinnäytetyössäni esitetyillä työkaluilla ja ratkaisuilla.

LÄHTEET

Ryhänen, J. 2015.HTML5 Pelikehityksessä. Tampereen ammattikorkeakoulu.

Tossavainen, A. 2014. HTML5-pelisovelluksen kehittäminen. Turun ammattikorkeakoulu.

Anglé, S. 2014. PELINKEHITYS Demosta julkaisuun. Oulun ammattikorkeakoulu.

Lahti, H. 2014. Pc-pelin kehitys ja julkaisu. Tampereen ammattikorkeakoulu.

Korpela, J. 2011. HTML5 Uudet Ominaisuudet. HAAGA-HELIA ammattikorkeakoulu.

Wargh, A. 2014 2D-pelin luominen Construct 2 -ohjelmalla Suunnittelusta ja tuotannosta demoon. Metropolia Ammattikorkeakoulu.

Subagio, A. 2014. Learning Construct 2. Packt Publishing Ltd.

Aycock, J. 2016. Retrogame Archeology Exploring Old Computer Games. Springer.

Schell, J. 2008. The Art OF Game Design – A Book of Lenses. Morgan Kaufmann.

Matikainen, Tuomas. 2014. Videopelilaitteiden teknologian aiheuttamat rajoitteet pelimusiikin kehityksessä. Jyväskylän Yliopisto.

Coe, Thomas. 2015. Legal issues for independent games developers. Wright Hassal.

Gullen, A. 2011. Publishing and promoting your Construct 2 game. Scirra.

Kelsey, B. 2012. Kickin' it old school: Setting up NES style chiptunes. Open Gameart. http://opengameart.org/forumtopic/kickin-it-old-school-setting-up-nes-style-chiptunes

Wikipedia-web-crawler
https://en.wikipedia.org/wiki/Web_crawler

Quick Tip: Make Retro, Low-Fi Game Sound Effects With Bfxr
https://gamedevelopment.tutsplus.com/tutorials/quick-tip-make-retro-low-fi-game-sound-effects-with-bfxr--gamedev-11579

Scirra – Construct2
www.scirra.com

HTML5 Game Engines
https://html5gameengine.com/

Suomen Pelimuseo
http://vapriikki.fi/pelimuseo/

W3schools
https://www.w3schools.com/html/html5_canvas.asp

Hoggins, Tom. 2014. The tragic tale of Flappy Bird. The Telegraph.
http://www.telegraph.co.uk/

Nes / Famicom sound hardware wiki
http://www.Famitracker.com

Statista, 2017. Genre breakdown of video game sales in the United States in 2016
http://www.statista.com