

Andreas Wargh

2D-pelin luominen Construct 2 -ohjelmalla

Suunnittelusta ja tuotannosta demoon

Metropolia Ammattikorkeakoulu

Medianomi

Viestintä

Opinnäytetyö

22.4.2014

Tekijä(t) Otsikko	Andreas Wargh 2D-pelin tuottaminen Construct 2-ohjelmalla – Suunnittelusta ja tuotannosta demoon
Sivumäärä Aika	35 sivua + 2 liitettä 22.4.2014
Tutkinto	Medianomi
Koulutusohjelma	Viestintä
Suuntautumisvaihtoehto	Digitaalinen viestintä
Ohjaaja(t)	Yliopettaja Pauli Laine
<p>Tämä opinnäytetyö koostuu toiminnallisesta ja kirjallisesta osasta. Toiminnallinen osa koostuu tuotetusta pelidemosta, joka löytyy liitteistä (ks. liite 2.).</p> <p>Opinnäytetyön toiminnallisena osiona tehtiin 2D-peli Construct 2 -ohjelmalla. Työn kirjallisen osion tavoitteena on esittää itsenäisen pelinkehittäjän eri työprosessit ja tuoda ilmi, mitä on mahdollista tehdä pienen budjetin ohjelmistolla. Opinnäytetyössä käy ilmi erilaiset tekniset ja visuaaliset keinot, joilla saavutettiin demon funktionaalisuus sekä visuaalinen lopputulos.</p> <p>Kirjallinen osio sisältää pohjustavan osion kotimaisesta pelituotannosta sekä alalla työskentelystä. Tämän lisäksi työ sisältää suunnittelu- ja tuotanto-osiot. Suunnittelu-osiossa kerrotaan hahmosuunnittelusta ja toivotusta visuaalisesta ulkoasusta. Osiossa käy ilmi kaikki suunnittelutyö koskien pelin graafista ulkoasua valikoista kenttiin.</p> <p>Tuotanto-osio pitää sisällään dokumentaatiota pelinkehityksen ajalta ja paljastaa käytännöllisiä keinoja, joilla saavutettiin toivottu lopputulos. Osiossa selostetaan tuotantoprosessin eri vaiheita, kuten animointia ja grafiikoiden implementaatiota Construct 2:een.</p> <p>Opinnäytetyön tavoitteena oli tuottaa toimiva 2D-tasohyppely-peli. Kirjallisen osion tavoitteena oli dokumentoida eri vaiheita 2D-pelituotannossa ja selvittää, millaiset mahdollisuudet on toteuttaa 2D-peli täysin itsenäisesti. Tekijän tavoitteena on saada toimiva pelidemo selaimelle.</p> <p>Työssä saatiin selville, että on täysin mahdollista luoda itsenäisesti yksinkertaisia ja toimivia 2D-pelejä. Yksinkertaisten toiminnallisuuksien rakentaminen peleihin on nykypäivän ohjelmistoilla huomattavasti helpompaa, eikä niiden käyttö edellytä pätevyyttä ohjelmoinnista. Construct 2:n kaltaiset pelintuotanto-ohjelmistot eivät silti korvaa ohjelmointitaitoja, mutta ne avaavat uusia mahdollisuuksia ohjelmointia osaamattomille.</p>	
Avainsana	Pelisuunnittelu, indie-pelit, 2D, After Effects, Construct 2

Author(s) Title	Andreas Wargh Producing a 2D Game with Construct 2 – Designing and producing a demo
Number of Pages Date	35 pages + 2 appendices 22 April 2014
Degree	BA
Degree Programme	Communications
Specialisation option	Digital media
Instructor(s)	Pauli Laine, Principal Lecturer
<p>This thesis consists of two parts: the written thesis and the game itself. The game is attached to this thesis (see appendix 2.).</p> <p>The topic of this thesis is to demonstrate the workflow of independently creating a 2D videogame with Construct 2. This thesis includes documentation of different processes such as concepting and production. The aim is to study how well you may develop a game with a low budget software. This thesis will provide detailed information about graphic design, character design and animation. It also demonstrates the functionality of Construct 2.</p> <p>This thesis includes an overall view of Finnish video game industry. The rest of the thesis consists of two sections: design and production. The design-section includes designing of graphic elements and early concepts. It also contains character-design and reveals what kind of look and feel was planned for the game.</p> <p>The production-section includes demonstrations about different tools and workflows of creating content. It also provides documentation of different stages of production.</p> <p>The aim of this thesis is to create a fully functional 2D-platformer and to study the opportunities of fully independent game-development. The purpose is to create a working demo for browsers.</p> <p>It can be concluded, that it is entirely possible to independently create simple 2D games with Construct 2. Nowadays, it is easy to create games without programming skills by using game developer software such as Construct 2. It is still notable, that these programs won't overthrow programming skills. They only work as mere tools to open opportunities for people who don't have knowledge of programming.</p>	
Keywords	Game design, indie games, 2D, After Effects, Construct 2

Sisällys

1	Johdanto	1
2	Keskeiset käsitteet	2
3	Tasohyppely genrenä	3
4	Peliala Suomessa	4
4.1	Kotimaisen pelituotannon kehittyminen	4
4.2	Pelinkehitys ammattina	5
5	Pelin konsepti	7
5.1	Synopsis	8
5.2	Pelattavuus ja mekaniikka	8
6	Graafinen suunnittelu	8
6.1	Värimaailma ja kenttäsuunnittelu	8
6.2	Hahmosuunnittelu	12
6.3	Valikot ja HUD	12
7	Tuotantoprosessi	14
7.1	Animointi	15
7.2	Pelin luominen Construct 2:een	23
7.2.1	Kontrollit	27
7.2.2	Viholliset	29
7.2.2	Pisteitys ja muuttujat	31
7.2.2	Äänimaailma	32
7.3	Haasteet	32
7.4	Epäonnistumiset	32
7.5	Onnistumiset	33
8	Yhteenveto	34
9	Lähteet	36

Liitteet

Liite 1. Pelin intro-animaation kuvakäsikirjoitus.

Liite 2. Save Mr.Puz -pelidemo.

1 Johdanto

Olen pienestä lähtien ollut hyvin innokas pelaamaan tietokonepelejä ja kasvoinkin hyvin pitkälle niiden keskellä. Pelit ovat merkinneet minulle hyvää viihdettä, ja ne ovat parhaimmillaan vetäneet täysin sisälle omaan maailmaansa. Pelaajan ja pelien välinen vuorovaikutus on yksi syy, miksi pelit ovat viihdemuotona kasvattaneet suosiotaan hurjasti.

Pelikulttuuri on ajan saatossa läpikäynyt useita kehitysvaiheita. Nykyään pelilaitteita on enemmän ja pelaajakunta on erilainen kuin kymmenen vuotta sitten. 1970- ja 1980-luvulla pelejä suunniteltiin nuorisolle ja vasta myöhemmin, 1990-luvulla, pelejä kehitettiin myös aikuisille. 2000-luvun alkupuolella tehdyt selvitykset osoittavat pelaajien keski-ikäen nousevan tasaisesti. Pelaajakunta on siis hyvin vaihtelevaa ja eri-ikäiset pelaajat suhtautuvat peleihin eri tavoin. (Sihvonen, Mäyrä, Saarenpää, Kultima, Paavilainen & Stenros 2010, 18–19) Osun siis itse juuri siihen taitteeseen, kun painopiste pelien kohdentamisesta vanhemmille ikäryhmille yleistyi.

Olen pitkään omaksunut roolin lähinnä pelien kuluttajana ja vasta hiljattain mahdollisuuksien auetessa halusin vihdoin pyrkiä tuottamaan oman pelin. Taustalla opinnäytetyöni aiheen valitsemiseen vaikutti merkittävästi Metropolian järjestämä Game Portal -innovaatiokurssi, jonka aikana opin hyvin paljon pelien tuotantoprosessista ja kehityin myös merkittävästi käytännön taidoissa. Kurssin aikana toimin graafikkona ja animaattorina. Mainittuihin rooleihin liittyviä käytännön taitoja olen myös kehittänyt sekä työelämässä että omalla ajallani.

On olemassa useita pelien tuotanto-ohjelmistoja, kuten Steamin GameMaker: Studio, TGC:n App Game Kit ja Unity 3D. Päädyin valitsemaan Scirran Construct 2:n tuotantovälineekseni, koska en osaa 3D-mallinnusta ja Construct 2:n helppokäyttöisyys vetosi minuun. Valintaani vaikutti myös Game Portal-kurssin opettajalta saamani suositus Construct 2:sta. Vain vajaan sadan euron investoinnilla saa Construct 2:een lisenssin, jolla ohjelmaa voi käyttää myös kaupallisiin tarkoituksiin. Lisenssi oikeuttaa myös julkaisemaan pelejä mobiilialustoille. Käytän pelin grafiikan ja animaatioiden tuotantoon Adobe CS 6 - Masters Collectionin opiskelijaversioon sisältyviä Adobe Photoshopia ja Adobe After Effectsiä.

Käsittelen työssäni pelien historiaa ja pelialan kehittymistä Suomessa. Sivuan myös kotimaisen peliteollisuuden tuotantoprosesseja peilaten niitä täysin itsenäiseen pelituotantoon. Esitän myös koko tuotantoprosessin eri vaiheita mahdollisimman kattavasti. Visuaalisten elementtien luomisessa käsittelen lähinnä sitä, millaista tyyliä ja teemaa haen. Syvennyn tarkemmin animointiin ja luonnollisesti myös Construct 2:n toimintoihin.

Opinnäytetyöni sisältää hyvin paljon eri metodeja ja tekniikoita, joita kaikkia en voi valitettavasti esittää kovin yksityiskohtaisesti. Syy on yksinkertainen; aiheesta täytyisi siinä tapauksessa kirjoittaa kokonainen kirja. Pyrkimyksenäni on silti selostaa eri vaiheita mahdollisimman selkeällä kielellä.

2 Keskeiset käsitteet

Sprite – Kaksiulotteinen grafiikka joka voi olla mikä tahansa näkyvä objekti pelikentällä. Spritejä on myös mahdollista animoida. (Wikipedia 2014a.)

Tekstuurit – Kaksiulotteista grafiikkaa, joilla peitetään jokin muoto tai pinta. 3D-grafiikassa tekstuureille voi asettaa useita eri tehosteita ja pintaheijastuksia (Wikipedia 2014a.). Niitä voi myös asettaa eri muotojen päälle, mikä täytyy ottaa huomioon suunnitteluvaiheessa. 2D-grafiikassa tekstuurit ovat tiiliä, joiden reunat sopivat saumattomasti yhteen. Tiiliä voi hyvin pienellä vaivalla asettaa 2D-maailmaan.

Framerate – Lukema, joka kertoo kuinka monta ruutua näytöllä toistuu sekunnissa. Yleisimmät videoiden framerateet ovat 23,976, 24, 25 ja 30 (Wikipedia 2014b). High Frame Rate (HFR) -elokuvissa framerate on 50 tai 60 (Wikipedia 2014c). Pienemmissä graafisissa animaatioissa sekä mm. timelapsessa käyvät myös 13–17.

Kompositio – After Effectsissä on mahdollista ns. paketoita useammat layerit efekteineen yhteen kompositioon. Jokainen kompositio sisältää oman aikajanan. Projektin siisteyden kannalta välttämätöntä. (Adobe After Effects Help 2014.)

Layerit – Vektorigrafiikkaa, ääntä, kuvia, sekä liikkuvaa kuvaa on käytettävistä ohjelmistosta riippuen mahdollista asettaa päällekkäin ja muokata eri työkaluin sekä efektein.

Pelattavuus – Kuinka pelaaja on vuorovaikutuksessa pelin kanssa. Pelin asettamat säännöt ja raamit määrittelevät pelattavuuden. (Wikipedia 2014d.)

Power-up – Pelimaailmassa kerättäviä esineitä, joista pelaaja voi saada ylimääräisiä pisteitä tai lisävoimia. Esimerkiksi Super Mario Brosin punaiset sienet ovat tyypillisiä power-uppeja. (Wikipedia 2014e.)

Wish list – Lista pelin ominaisuuksista, joita ei ollut aikaa tai mahdollisuuksia toteuttaa.

PNG-sekvenssi – Sarja animoituja PNG-kuvia joita voi tuottaa Adobe After Effectsillä

Renderointi – Prosessi, jossa käytettävä ohjelmisto koostaa työstetyn animaation, grafiikan tai videon toivottuun tiedostomuotoon.

HUD – Eli Head-Up-Display sijaitsee peleissä tyypillisesti ruudun reunoilla, tarjoten pelaajalle informaatiota pisteityksestä, ammuksista tms. (Wikipedia 2014f)

3 Tasohyppely genrenä

Tasohyppelypelit ovat useimmiten sivusta kuvattuja 2D-pelejä, joissa pelaajan tulee kulkea esteiden läpi vasemmalta oikealle. Kentissä tulee usein kerätä esineitä, joista saa pisteitä ja mahdollisesti myös power-uppeja. Useissa tasohyppelyissä tulee pelaajan tuhota vihollisia tyypillisesti hyppäämällä niiden päälle.

Ensimmäinen tasohyppely-peli Universalin *Space Panic* julkaistiin vuonna 1980. (Wikipedia 2014g.) Julkaistu peli ei onnistunut saavuttamaan yhtä suurta suosiota kuin Nintendon julkaisema Donkey Kong. Nintendo julkaisi myös myöhemmin Super Mario Brosin, joka saavutti nykypäivään asti kestävän valtavan suosion (Wikipedia 2014h.). Super Mario on saavuttanut kulttimaineen ja on ikoninen tasohyppelygenren edustaja. Nintendo julkaisee nykyäänkin useita uusia peliversioita Super Mariosta. Pelit ovat uskollisia alkuperäiselle pelisarjalle, mutta uusien laitteiden myötä nekin ovat uudistuneet innovatiivisella tavalla. Esimerkkinä innovatiivisesta Super Mario -pelistä toimii Nintendo Wiin moninpelattava New Super Mario Bros Wii. Peliä voi pelata useampi pelaaja samanaikaisesti eleohjaimilla.

4 Peliala Suomessa

Suomalainen peliteollisuus on synnyttänyt maailmanlaajuisesti arvostettuja hittipelejä kuten Max Payne- ja Flat Out -pelisarjat. Myös Sulakkeen Habbo-hotel keräsi paljon nuoria pelaajia seurustelemaan keskenään virtuaalimaailmassa. Suomalaisilla on myös ollut korkeaa osaamista mobiilipelien tuottamisessa. Digital Chocolaten omistukseen siirtynyt Sumea teki pitkään pelejä lukuisille eri mobiilialustoille. Nykyhetkenä Rovio ja Super Cell ovat saavuttaneet huomattavan maailmanlaajuisen suosion mobiilipelialalla. Uusia pelialan yrityksiä on myös perustettu kiihtyvällä tahdilla, ja peliteollisuuden liikevaihto on kasvanut hurjaa vauhtia. Vuosina 2010–2012 peliteollisuuden liikevaihto kasvoi vuosittain yli 50 % ja vuonna 2012 liikevaihto oli 250 miljoonaa euroa (Kajantie 2013). Vuonna 2013 liikevaihto kasvoi räjähdysmäisesti yli miljardin euron ja pelialan kokonaisarvon arvioidaan olevan yli kaksi miljardia euroa. Lisäksi kolmen vuoden aikana peliteollisuus on synnyttänyt Suomeen 125 uutta yritystä (Suvanto 2014). Suomalainen peliteollisuus ei siis näiden numeroiden valoissa ole suotta ollut otsikoissa niittämässä mainetta. Kuinka kotimainen peliteollisuus saavutti tämän menestyksen ja millaista alalla työskentely voi olla?

4.1 Kotimaisen pelituotannon kehittyminen

Suomalaisella peliteollisuudella on ollut etuna se, että varhaiset pelinkehittäjät olivat ajoissa liikkeellä ja maahamme syntyi innokas kehittäjien alakulttuuri. Kuten sanonta kuuluu, aikainen lintu madon vie. Tämän seurauksena Suomessa on useampia vanhoja tekijöitä jotka ovat vakiinnuttaneet asemansa pelimarkkinoilla. Näihin uranuurtajiin lukeutuvat muun muassa Remedy Entertainment, Housemarque ja WAH Software. (Wikipedia 2014i.)

Suomessa suunniteltiin tietokonepelejä 1950-luvulla, mutta niiden käyttö kasvoi vasta seuraavilla vuosikymmenillä pelaamisen alkaessa kiinnostaa yliopisto-opiskelijoita. Suomessa rakennettiin ensimmäinen NIM-pelikone keväällä 1955. Kyseinen NIM-peli on hyvin vanha ja yksinkertainen peli, josta on olemassa useita eri variaatioita. Pelissä on kaksi pelaajaa ja pelivälineenä toimii jokin lukumäärä tikkuja, kiviä tai mitä tahansa sopivia pelinappuloita pinoissa. Pelaajien tulee poistaa vuorollaan haluamansa määrän pelinappuloita tietyn ajan sisällä yhdestä tai useammasta pinosta. Pelaaja jolle jää viimeinen pelinappula, häviää pelin. Kyseistä ESKOksi nimettyä suomalaistekoista

pelikonetta ei voi kutsua nykyaikaiseksi tietokonepeliksi, vaan se liittyi varhaisiin tietokoneisiin. (Lehtonen & Kyllönen 2010, 4–5.)

Teknologian kehittyessä ja yleistyessä pelien määrä kasvoi hurjasti ja uusi teollisuudenala alkoi syntyä Suomeen. Suomeen syntyi 1980-luvun lopulla innokas demoskene-niminen alakulttuuri, jossa tietokoneharrastajat ohjelmoivat visuaalisesti näyttäviä demoja laitteistolle, jolla oli huomattavia teknisiä rajoitteita. Suomalaiset menestyivät hyvin demoskenessä, ja merkittävät suomalaiset peliyhtiöt, kuten Remedy Entertainment, Bugbear Entertainment ja Housemarque syntyivätkin demontekijöiden käsissä. (Wikipedia 2014j.)

Menestyviin suomalaisiin peliyrityksiin kuuluva FrozenByte syntyi myös peliharrastajien toimesta. Lauri Hyvärinen ja Juha Hiekkamäki perustivat armeijasta päästyään Frozenbyte Oy:n vuonna 2001. Yritys toimi aluksi vapaaehtoisvoimin pienessä opiskelija-asunnossa. Ryhmän kasvettua jatkui työskenteleminen autotallissa ja myöhemmin toiminnan muututtua ammattimaisemmaksi saivat he paikan yrityshautomosta. Frozenbyte keskittyy sähköisesti ladattavien pelien tuotantoon PC:lle sekä pelikonsoleille. Vuonna 2010 yritys työllisti 27 henkilöä ja on julkaissut suuren suosion saavuttaneen Shadowgroundsin sekä Trinen. (Kinnunen & Lyytikäinen 2010, 36–37.) Frozenbyte on hyvä esimerkki siitä, kuinka peleihin omistautunut ryhmä voi saavuttaa maailmanlaajuisia menestystä intohimoisella työpanoksellaan.

Tällä hetkellä Suomen peliteollisuus on kukoistanut erityisesti mobiilipelialalla. Suursuosion saavuttaneet Rovion Angry Birds -pelit ovat eräällä lailla ansainneet ikonisen maineen kosketusnäyttöpelien ”Tetrixenä”. Peli on helppo oppia, ja siihen on jokaisen kotimaasta ja kulttuurista riippumatta helppo jäädä koukkuun. Lisäksi japanilaisomistukseen siirtynyt Super Cell on erityisen tunnettu rahallisesti hyvin kannattavista mikromaksuja sisältävistä Hay Daysta ja Clash of Clansista. Pelit ovat hyvin koukuttavia, ja kynnyksellä ostaa virtuaalista sisältöä on saatu vedettyä matalalle.

4.2 Pelinkehitys ammattina

Peliteollisuus on nykypäivän Suomessa kovassa nosteessa, ja alasta on povattu jopa ”uutta Nokiasia”. Supercellin ja Rovion suurmenestys on synnyttänyt alan ympärille kovasti puhetta ja hypeä. Millaista työskentelyä alalla oikeasti on? Ville Vuorela jakaa näkemyksiään seuraavasti kirjassaan *Elämänpeli, pelintekijän maailmat*:

Olemme pelintekijöitä, unelmien ammattilaisia. Nykyisin on muotia korostaa pelien tekemisen olevan kovaa työtä, suorituskyvyn ääri rajoilla tapahtuvaa ohjelmistonkehitystä ja raadollista, perusvietteihin vetoavaa massaviihdeteollisuutta. Kovuus ei silti muuta sitä tosiasiaa, että tämä on unelma-ammatti: me pelintekijät teemme tätä mieluummin kuin mitään muuta – ja Nokian tuplapalkkaa nauttiva prässihousukansa tietää sen. (Vuorela 2009, 11–12.)

Alalla ollaan siis täysillä mukana, mutta työskentely ei ole aina ruusuilla tanssimista. Vuorela kertoo kaunistelematta peliprojektien turhauttavista seikoista, kuten julkaisijoiden kanssa työskentelystä ja pelinkehittämisestä eri mobiilialustoille. Hänen mukaansa peliala jakautuu kehittäjiin ja julkaisijoihin. Julkaisijat pyrkivät vaikuttamaan pelin sisältöön ja pelinkehittäjät joutuvat ajoittain alistumaan julkaisijan, eli rahoittajan, pyytämiin muutoksiin. Tämä ei ole aina itse pelin kannalta edullista, sillä julkaisijoiden tietotaito peleistä poikkeaa hyvin paljon pelinkehittäjien osaamisesta. Kehittäjän tulee siis kiltisti myöntyä julkaisijoiden toiveisiin, sillä rahoitus tulee heiltä. (Vuorela 2009, 19–20.)

Julkaisijat saattavat toimia kyseenalaisesti pelien tuotannossa ja julkaisemisessa. Vuorela kertoo aavistuksen katkerana alihankintatyönä tehdystä Burnoutin mobiilipeliversiosta, joka menestyi hienosti arvosteluissa ja tekijätkin olivat ylpeitä pelistään. Julkaisijan kanssa tehdyssä sopimuksessa oli kuitenkin jokin pykälä, jossa julkaisija voi mainostaa itseään pelintekijänä. Näin pelin oikeilta tekijöiltä oli viety kunnia heidän omasta työstään. Lisäksi alihankintatyönä tehdyissä projekteissa rahoitus tapaa tulee myöhässä ja erikoisehtojen kanssa. Julkaisijat usein omivat oikeudet pelin immateriaalioikeuksiin, jolloin heillä on mahdollisuus tuottaa pelien jatko-osat edullisemmin toisella pelistudiolla. (Vuorela 2009, 21–22, 26–27.)

Vuorela kertoo erityisesti mobiilipelien kehittämisen olleen äärimmäisen turhauttavaa useiden eri alustojen sekä puhelinten erojen takia. Ymmärrettävästi Vuorela puhuu ajasta ennen iPhonea ja sitä seuranneita muita kosketusnäyttöpuhelimia. Tällöin pelin jälkituotanto aloitettiin 5–8 erilaisesta versiosta, jotka myöhemmin haarautuvat lopulta pariksi sadaksi versioksi, jotta kännykkäpelejä välittävät operaattorit suostuisivat ottamaan pelin valikoimaansa. Luodut versiot voivat olla keskenään hyvin erilaisia ja 90 % niistä on taloudellisesti turhia. Vuorela kehuu iPhonea erinomaiseksi mobiilipelialustaksi ja kertoo Applen app-storen houkuttelevan kehittäjiä mm. sen operaattoriin riippumattomuuden vuoksi. (Vuorela 2009, 18–19.)

Vuorela kirjoittaa: ”Enää ei puutu kuin ne kaksisataa miljoonaa iPhonea. Sitä saadaankin varmasti odottaa.” (Vuorela 2009, 19.) Oli lainaus sarkasmia tai ei, on helppo nykyhetkessä todeta, että kaksisataa miljoonaa iPhonea on toteutunut ja rinnalle on noussut jopa suositumpi Googlen Android-alusta. Mobiilipelien taso ja pelattavuus on parantunut huomattavasti, ja pelejä on myös huomattavasti enemmän saatavilla. Mobiilipelit kävivät kivuliasta kasvukehitystä, johon saatiin helpotusta kosketusnäyttöpuhelinien ja yhtenäistyvien mobiilialustojen myötä. Mobiilipelien grafiikat ovat myös kehittyneet huimasti vuosien varrella, ja tämä kehitys muistuttaakin hyvin paljon aikoinaan tapahtunutta PC-pelien kehitystä.

Voi siis todeta, että 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.

Vuorela kertoo, että ne kehittäjäfirmat, joiden ei tarvitse julkaisijoista murehtia, ovat indie-studioita. Vuorelan mukaan heidät tunnistaa leveän hymyn lisäksi siitä, että he näkevät nälkää. (Vuorela 2009, 24.) Tämä on melko paikkansa pitävää ja käy erityisen selväksi *Indie Game : The movie* -dokumentissa. Dokumentti kertoo indie-kehittäjien luovan työn haasteista, paineista ja taloudellisesta kituuttamisesta. Indie-pelien työprosessi ei ole dokumentin mukaan kovin houkutteleva ja suoraviivaisimmasta päästä. Silti kehittäjien lopputuloksen muovautumista sekä heidän ylpeyttä ja intohimoa omaa työtä kohtaan oli ilo seurata dokumentissa. (Indie Game: The Movie, Kanada 2012)

5 Pelin konsepti

Pelin tuottaminen lähti ideasta, joka tupsahti mieleeni päästyäni koulusta. Taustalla oli muhinut idea yksinkertaisen ja klassisen tasohyppelyn tekemisestä. Suurempia ja pelimekaniikaltaan uniikkeja ideoita en kyennyt resurssien puutteen vuoksi edes harkitsemaan. Halusin tehdä pelin kokonaan itse, ja minun tuli valita helpoimmin lähestyttävä tapa. Lähdin siis tuottamaan ideaani huomioon ottaen käytettävät voimavarat ja työkalut.

5.1 Synopsis

Pelaaja ohjaa Mr. Puz -nimistä mustekalaa, joka on siepattu merestä. Kalastajien mukana Mr. Puz päätyy marketin kalatiskille ja sitä kautta keittiöön. Tätä tapahtumaketjua demonstroidakseni olen luonut kuvakäsikirjoituksen suunnitteilla olevasta intro-animaatiosta (ks. liite 1.). Pelaajan tulee palauttaa Mr. Puz takaisin sen omaan vesistöön. Matka käy erilaisten kenttien kuten keittiön ja viemäriputkien läpi. Mr. Puz kohtaa useita vaaroja, joista pelaajan tulee selvitä. Peli päättyy onnellisesti Mr. Puzin kotiinpaluuseen.

5.2 Pelattavuus ja mekaniikka

Peli on klassinen tasohyppely, jossa pelaaja pystyy liikkumaan vasemmalle ja oikealle, hyppäämään ja kiipeämään. Pelaajan tulee kerätä helmiä kentistä ja väistää erilaisia esteitä. Kentissä on myös vihollisia, joita voi tuhota hyppäämällä niiden päälle. Jokaisesta tuhotusta vihollisesta saa pisteitä. Kun kentän helmet on kerätty, pelaajan tulee mennä maalialueelle, minkä jälkeen rata on suoritettu. Pelaajan päästyä kentän läpi ilmestyy ruutuun pisteitysikkuna. Tämän jälkeen pelaaja pääsee etenemään seuraavalle kentälle.

6 Graafinen suunnittelu

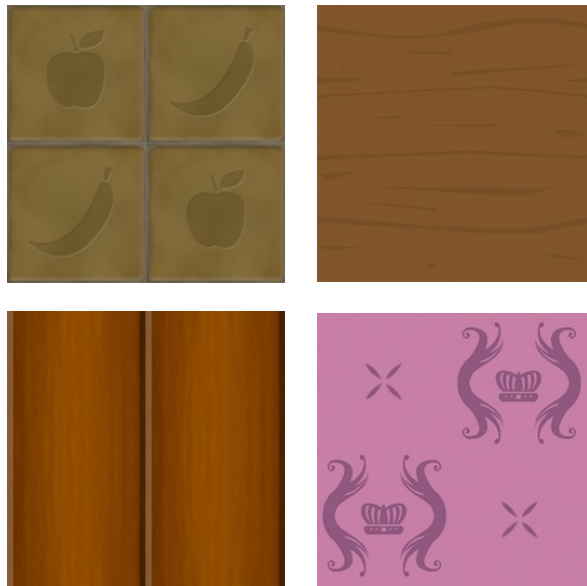
Loin grafiikat Photoshopissa ja tallensin tiedostot PNG -muodossa. Staattiset grafiikat jotka eivät sisällä animaatiota, pystyy suoraan viedä Construct 2:een. Animoitavat grafiikat täytyi luoda omalla tavalla Photoshopissa huomioon ottaen jatkotyöskentely After Effectsissä. Mikäli on tavoitteena julkaista mobiilialustalle, on hyvä ottaa huomioon että luo grafiikat riittävän suuressa koossa. Tällöin grafiikat ovat häviöttömästi skaalattavissa päätelaitteelle sopivaan kokoon. Loin grafiikat täysteräväpiirto-koossa, joka on riittävä mahdollista skaalausta varten.

6.1 Värimaailma ja kenttäsuunnittelu

Halusin luoda hillityn ja hemeän, hiukan lapsenmielisen tunnelman peliin. Tavoitteenani on nimenomaan tuottaa peli, jota voi kaikki ikäluokat pelaa. Tämän vuoksi pyrin luomaan lapsekkaan ja vetoavan värimaailman peliini.

Kenttien värimaailma vaihtelee sen mukaan, onko hahmo joko ihmisten luomassa ympäristössä vai luonnossa. Ihmisten luomat ympäristöt ovat karumpia ja värimaailmaltaan synkempiä, mutta silti maanläheisiä. Luonnolliset ympäristöt kuten vesistöt, ovat kirkkaita ja värimaailmaltaan pirteitä sekä raikkaita. Värien avulla rakennan tunnelman ympäristöihin, jotka vastaavat Mr. Puzin mieltymyksiä.

Suunnittelin kenttiä varten tiilitekstuureja sekä spriteja Photoshopilla. Tiilitekstuurit tulee suunnitella huolellisesti, jotta ne sopivat saumattomasti yhteen. On myös tärkeää, että niissä ei ole liian silmiinpistävää toistoa. Liian korostavien häivytysvärien käyttöä kannattaa välttää, koska niistä huomaa helpoiten tekstuurin toistuvuuden.



Kuvio 1. Suunnittelemani tiilitekstuureja keittiö-kenttään.

Kenttää varten täytyi suunnitella myös muita visuaalisia elementtejä, joiden kanssa hahmo on vuorovaikutuksessa. Kentissä tulee esimerkiksi olla tasaisia pintoja, joiden päällä hahmo voi liikkua. Lisäksi kenttiin tulee suunnitella sisältöä, jolla ei varsinaisesti ole pelillisesti merkitystä vaan joka tuo elävyyttä ja täytettä pelimaailmaan.



Kuvio 2. Keittiön pöytätason osia. Keskikappale on suunniteltu siten, että sitä voi toistaen käyttää myös pitkiin pöytätasoihin. Päätypalat asetetaan tämän pöytätason pätyihin.



Kuvio 3. Suunnittelemani ikkuna läpinäkyvyydellä ja "likaisuus"-efektillä (Photoshopissa *difference clouds*). Alla oleva piirtämäni maisemakuva ja ikkuna on asetettu pelissä eri parallaksi-tasolle, joka luo illuusion maiseman vierimisestä eri tahtiin pelissä. Jakkara toimii tasona, jolle hahmolla voi hypätä.



Kuvio 4. Piirtämiäni keittiön kaapistoja. Kaapistoilla ei ole pelillistä arvoa, vaan ne elävöittävät pelimaailmaa.

6.2 Hahmosuunnittelu

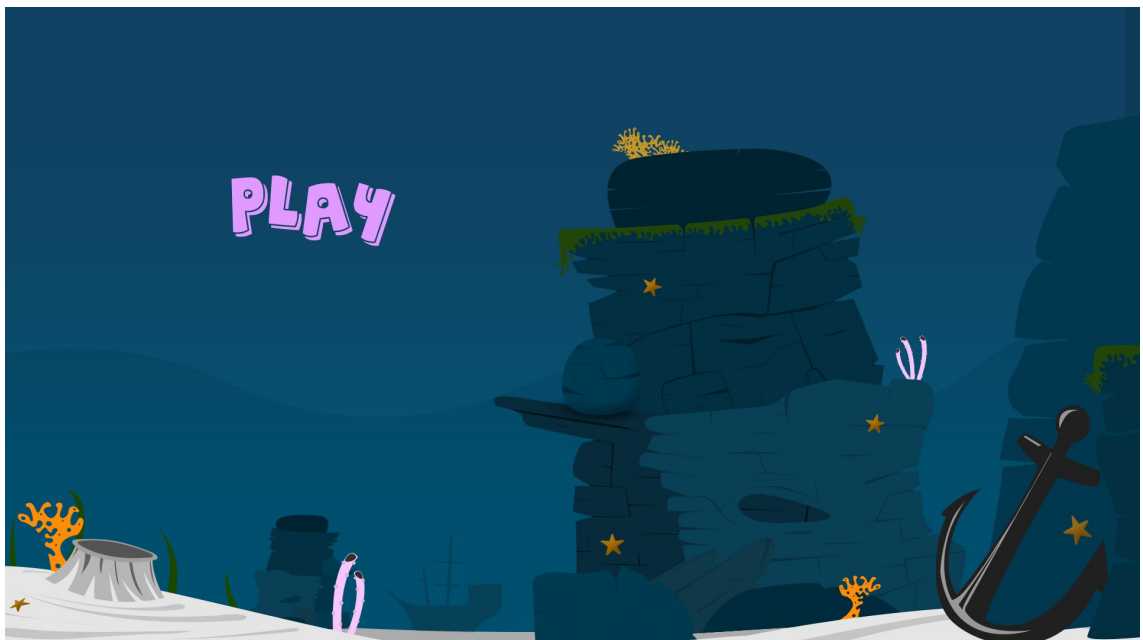
Hahmon väriksi valitsin pinkin saadakseni hahmoon haavoittuvuutta ja pehmeyttä. Alkuperäinen ideani ei juuri muuttunut tuotannon aikana, vaan päätin tiukasti seurata ideaani ns. herrasmies-mustekalasta. Saadakseni hahmoon ”herrasmiesmäisyyttä”, loin hahmolle pienen knallin ja monokkelin jäljitellen hienostuneen britin tyyliä. Jaoin kaikki liikkuvat osat omille layereilleen animointia varten.



Kuvio 5. Vasemmalla on ensimmäinen nopea hahmotelma ja oikealla lopullinen versio.

6.3 Valikot ja HUD

Halusin luoda merellisen ja hahmolle mieluisan ympäristön päävalikoksi. Halusin piirtää hillityn, mutta silti jännittävän merenalaisen maiseman. Tavoitteenani oli luoda dynaaminen valikko, joka kuhisee elämää. Valikkoa elävöittävät mm. lainehtivat merilevät, kalaparvet sekä merenpohjasta nousevat ilmakuplat.

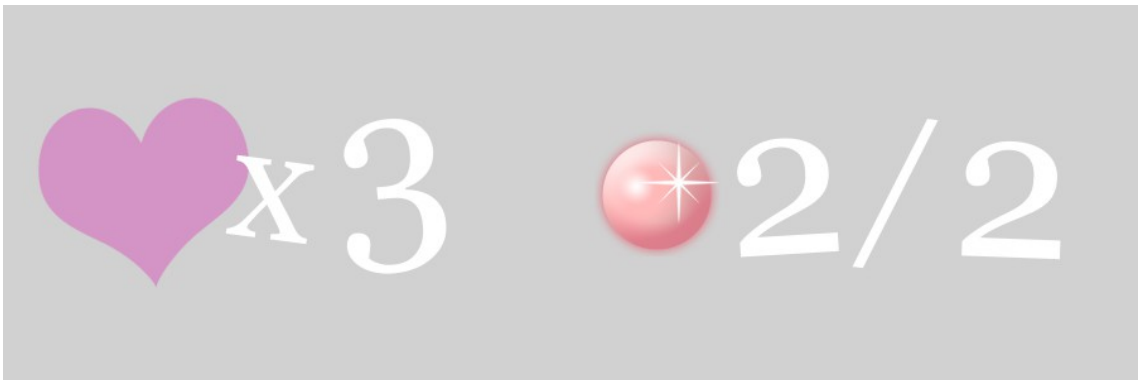


Kuvio 6. Piirros pelin päävalikosta.

Halusin tehdä HUD:sta yhteneväisen muun pelin kanssa. Pyrkimyksenäni oli luoda siitä leikkisä, mutta silti riittävän informatiivinen. HUD:n luominen ei ollut täysin suoraviivaista, koska informaatio täytyy esittää mahdollisimman selkeästi. Tiedot pelaajan elämästä, pisteetyksestä ja helmien määrästä täytyy olla nopeasti tulkittavissa vilkaisulla. HUD:n täytyi olla siis tarpeeksi kookas ja ennen kaikkea selkeä. Loin elämä- ja kerätyt helmet-indikaattorit sprite-objekteina. Kokonaispistemäärän indikaattori luotiin Construct 2:ssa tekstikenttänä.

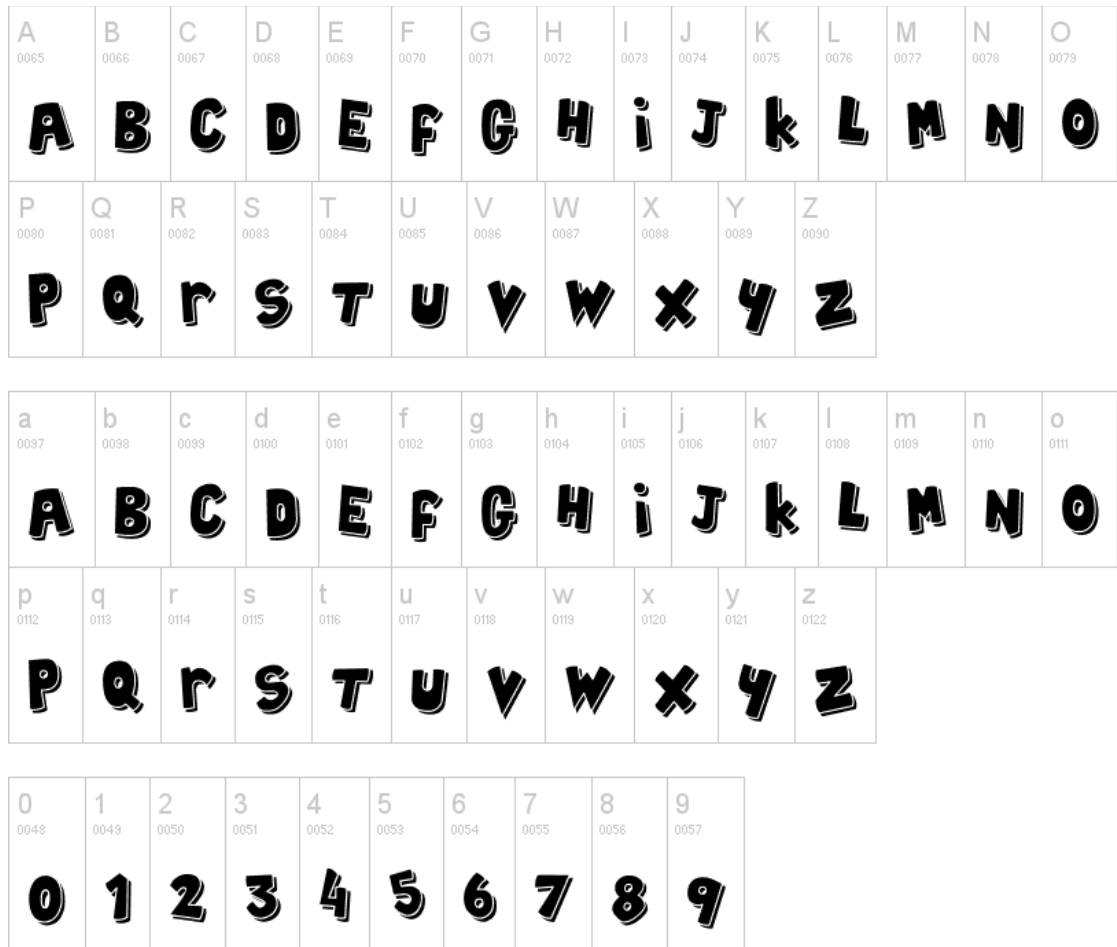


Kuvio 7. Luonnos elämä-indikaattorista. Tarkoituksena oli luoda pyöreä indikaattori, jota reunustaa kolme lonkeroa. Pelaajan menettäessä elämän, yksi lonkero katoaisi ja luku vähenisi yhdellä. Totesin, ettei tämä ollut tarpeeksi selkeä ja miellyttävä ratkaisu.



Kuvio 8. Lopulliset HUD-indikaattorit. Vasemmalla esitettynä elämä- ja oikealla kerätyt helmet-indikaattori. Päädyin huomattavasti yksinkertaisempaan ratkaisuun, jossa käytin selkeämpää grafiikkaa. Typografialla ja tekstin asetelulla sain indikaattoreihin leikkisän ulkoasun.

Mitä typografiaan tulee, yritin kovasti luoda hauskan ja uniikin tyylin pelin otsikolle. Totesin hyvin pian yksittäisten kirjainten piirtämisen olevan hyvin aikaa vievä prosessi ja päädyinkin käyttämään *Dafontista* löytämäni Geoff Cunninging Cobac-fonttiperhettä. (ks. kuvio 9.)



Kuvio 9. Geoff Cunningin Cobac-fonttiperhe

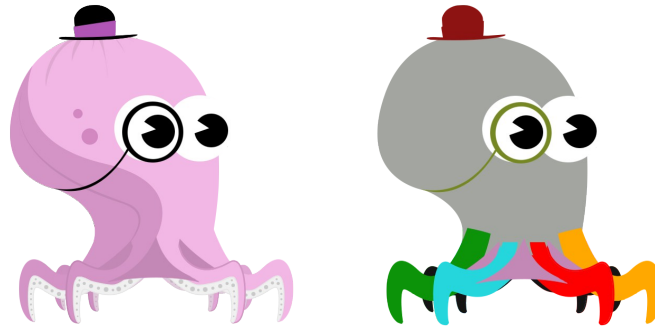
Fontti on juuri sopivan pelillinen, leikkisä ja aavistuksen jopa naiivi. Siitä tulee mieleen lasten aapiskirjat. Fonttia voi käyttää vapaasti ei-kaupallisiin tarkoituksiin, kunhan mainitsee fontin luojaan krediiteissä tai ilmoittaa asiasta suoraan artistille. Olen ottanut asiakseen antaa fontin luojalle ansaittua kunniaa suoraan sähköpostitse sekä pelin lopputeksteissä.

7 Tuotantoprosessi

Koin graafisen sekä animointityön helpoimmaksi tehtäväksi. Kuten odottaa saattoi, isoin haaste oli näiden elementtien kokoaminen toimivaksi peliksi. Tuotetun sisällön paketoiminen vieraassa Construct 2 -ohjelmassa oli tuotannon haasteellisin osuus.

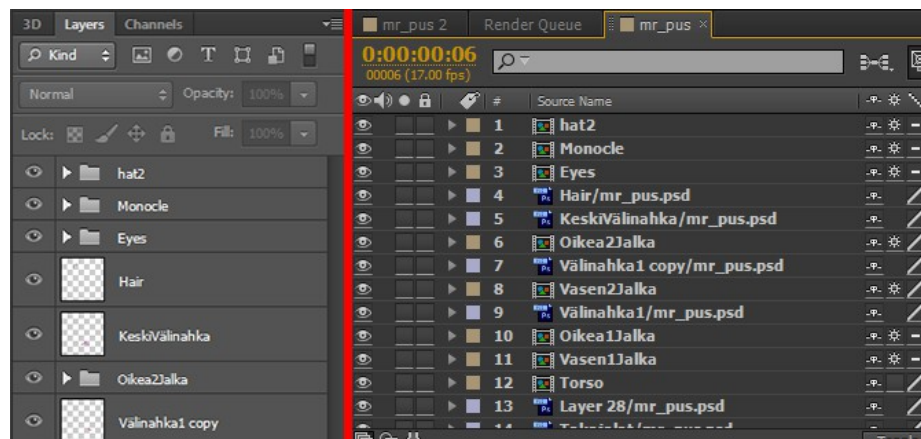
7.1 Animointi

Loin animaatiot Adobe After Effects -ohjelmalla. Animoiminen onnistuu luontevasti, kun ensin luo huolellisesti pohjagrafiikan Photoshopissa. Luodessa hahmoa, jota on tarkoitus myös animoida, on ehdottoman tärkeää ottaa huomioon eri osien tarkka nimeäminen ja asettaminen omille layereilleen. Photoshopin .PSD-dokumenttiedostot voi vaivatta tuoda After Effectsiin, jossa jokainen layeri säilyttää ominaisuutensa ja on vapaasti muokattavissa sekä liikuteltavissa.

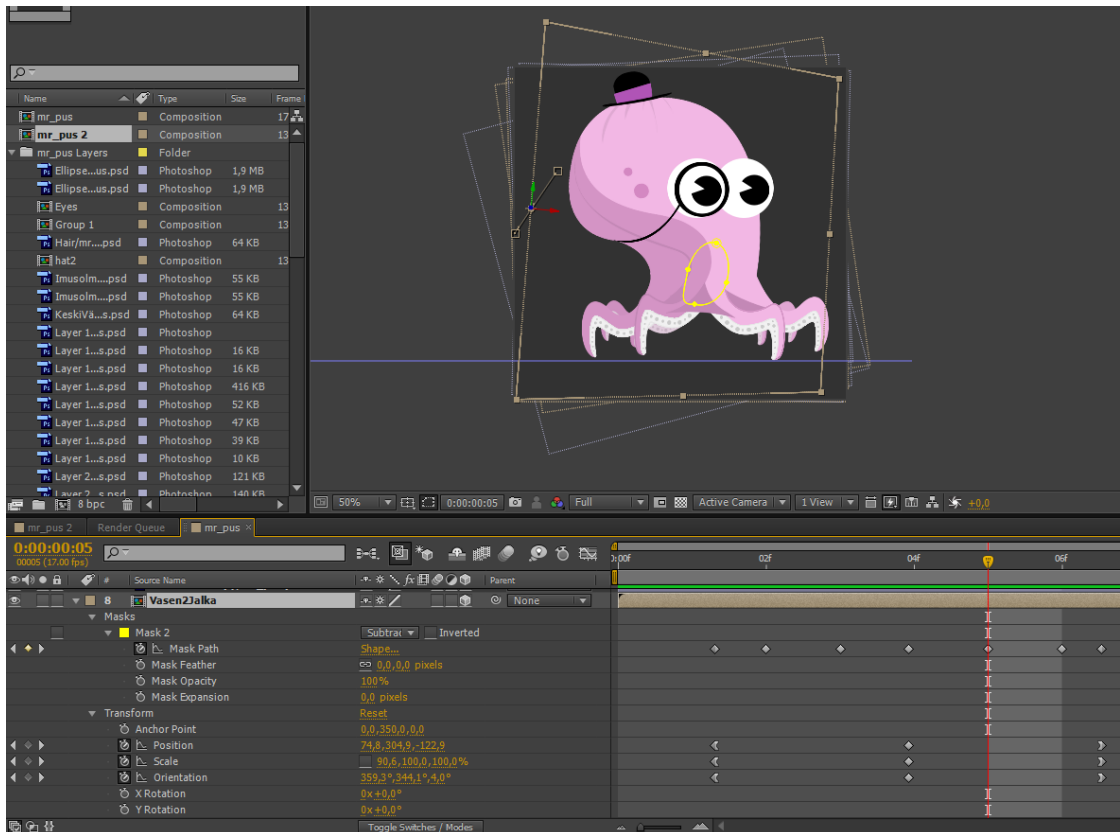


Kuvio 10. Layerit demonstroituna eri värein. Torso-osaan kuuluvia täpliä ja muuta grafiikka ei ollut tarkoitus animoida, vaan ainoastaan torso-osaa kokonaisuudessaan.

Hahmoa luodessa on suuri riski, että layereiden määrä kasvaa räjähdysmäisesti, joten on suositeltavaa lajitella animoitavat osat omiin ryhmiinsä. Nämä ryhmiin lajitellut layerit ilmestyvät After Effectsiin kompositioina, joiden sisällä layerit on lajiteltu samoin kuin Photoshopissa.



Kuvio 11. Kuvakaappaus Photoshopista ja After Effectsistä. Vasemmalla näkyvillä Photoshopin layerit ja oikealla layerit tuotuna After Effectsiin. Layereiden järjestys ja ominaisuudet pysyvät täysin ennallaan.

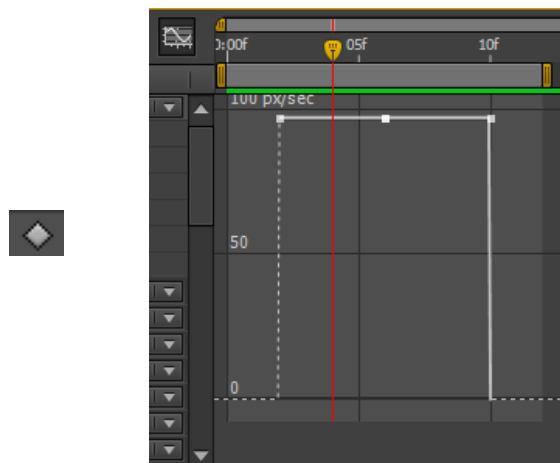


Kuvio 12. Kuvakaappaus After Effectsin käyttöliittymästä.

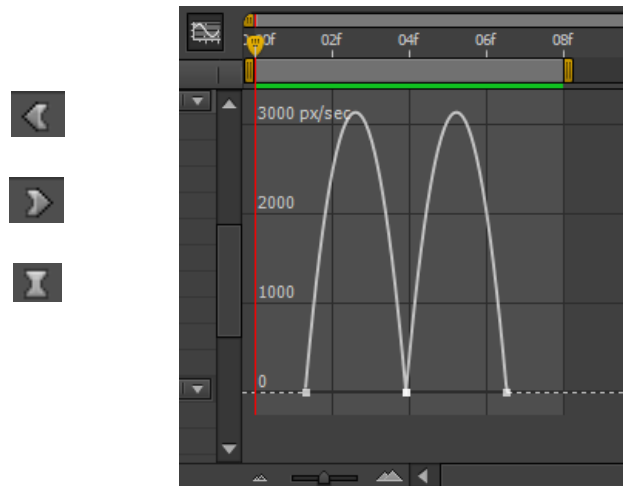
After Effectsin käyttöliittymä voi vaikuttaa aluksi hämmentävältä uudelle käyttäjälle. Ohjelmisto on eräänlainen Photoshopin ja Premierin sekoitus. Mikäli käyttäjällä on pohjatietoja molemmista ohjelmistoista, on After Effectsin opettelu kivuttomampaa. Alussa on suositeltavaa määrittellä työtilan layout. Oma suositukseni on käyttää työtilana *Standard* -layoutia.

Työtilan keskellä on esillä animaation esikatselu-tila. Vasemmassa yläosiossa näkyy projektiin tuodut grafiikat. Ruskein värikoodein merkityt layerit ovat kompositioita. Muut layerit ovat merkittynä oletusarvoisesti sinisellä. Alhaalla on timeline-osio, jossa animointi ja keyframien lisääminen tapahtuu. Punainen aikajanan lävistävä viiva osoittaa animaation sen hetkisen ajallisen kohdan. Sitä liikuttamalla näkee keyframien toiminnan ja esikatselutilassa olevien grafiikoiden liikkeen. Esimerkin kuvassa on näkyvillä mask -toiminnon animaatio. Hahmon päällä oleva keltainen ellipsi on maski *subtract*- toiminnolla. Maski piilottaa etualalla olevan toisen vasemman jalan turhaa grafiikkaa, joka muuten työntyisi liikkeessaan liikaa torso-layerin päälle. Maski on animoitu liikkumaan jalan mukana poistaen ei-toivotun grafiikan. Layeriin on myös luotu arvot erilaisin keyframein positiolle, koolle ja orientaatiolle. (ks. kuvio 12.)

After Effectsissä voi animoida hahmoja vaivatta. After Effectsiin tuoduilla layereilla näkyy oletusarvoisesti Transform -arvot, joita muokkaamalla saa lisättyä perusliikettä animoitavaan työhön. Arvoja lisätään projektin aikajanelle keyframein. Keyframeit ovat matemaattisten arvojen merkintöjä aikajanelle, jotka ohjelmisto punoo toisiinsa. Merkittäessä esimerkiksi grafiikan y-positiota arvolla "0" aikajanan alkuun ja seuraavalle frameille arvolla "10" grafiikka liikkuu kymmenen pikseliä ylöspäin yhden ruudun kuluessa. After Effects siis laskee aikajanan sisällä olevien keyframeiden arvojen erot, ja esikatselun avulla käyttäjä voikin tarkastaa miten animaatio piirtyy. Keyframeiden toimintoa ja laskutapaa toisiin keyframeihin nähden voi myös muokata joko lineaarisesti tai bezieriksi.



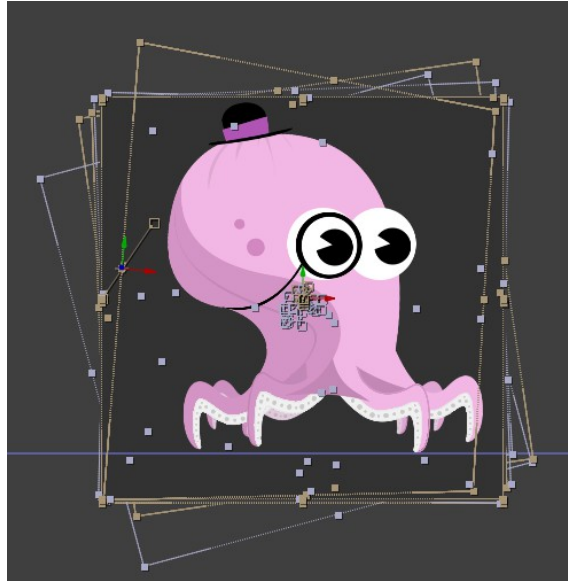
Kuvio 13. Vasemmalla näkyy linear keyframe-ikoni ja oikealla esitettynä lineaarisen keyframeiden liikerata *Graph editorissa*. Aikajanelle merkityn salmiakki-kuvioisen keyframeiden liikeradasta, huomaa ohjelmiston laskevan arvojen muutokset lineaarisesti. Liikerata näyttää silmään tasaiselta, mutta äkillisissä arvojen muutoksissa liike voi näyttää terävältä.



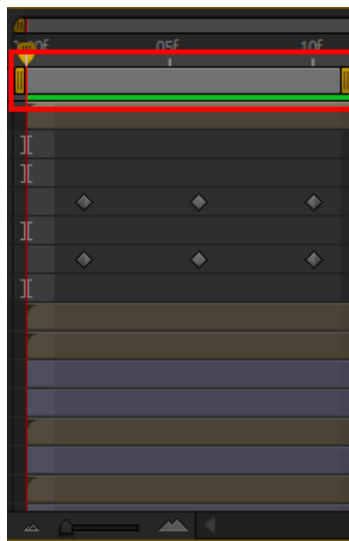
Kuvio 14. Ylin ikoni esittää *ease out*-keyframea, keskimäinen *ease in*- ja alin *easy ease*-keyframea. Graafissa 0:n kohdalla näkyvät laatikot ovat animaation keyframeja. Vasemmanpuoleinen keyframe on *ease out*, keskimäinen *easy ease* ja oikeanpuoleinen *ease in*. Kuten kuvioista näkee, *ease*-keyframeet laskevat liikkeen bezier-käyrinä. Liikeratoja voi tarpeen tullen muokata graph editorissa.

Käytin perustyökaluja kuten transformia, opacity ja rotationia animoimiseen. Tein myös joistakin layereista 3D-layereita, jotta pystyisin aavistuksen säätämään jalkojen Y-akselin rotaatiota. Käytin myös useampia maskeja. Saadakseni lisää ruutuja ja asentoja hahmolleni, käytin Muppet-työkalua. Muppet-työkalulla voi asettaa layereille ns. niveliä, joita timelinella siirtämällä saa vaivatta liikettä hahmoon. Muppet-työkalu ei silti toimi täydellisesti, ja tarkempaa grafiikkaa työstävälle sitä ei voi suositella. Työkalu vääristää työstettyä grafiikkaa, joten korkeampiresoluutioista hahmoa varten on suositeltavampaa vain perinteisesti piirtää ruutu ruudulta.

Muutettaessa layeri 3D-layeriksi transform-arvoihin ilmestyy lisää muokattavia arvoja x-, y- ja z-orientaatiolle. Tällöin layeriin voi myös lisätä muita 3D-toimintoja kuten materiaalin heijastavuutta ja paljon muuta. After Effectsissä on myös laaja määrä muita toimintoja, joita voi tarpeesta riippuen vetää hiirellä suoraan layerin päälle. Tällöin layerin alle ilmestyy jälleen liuta uusia muokattavia arvoja.



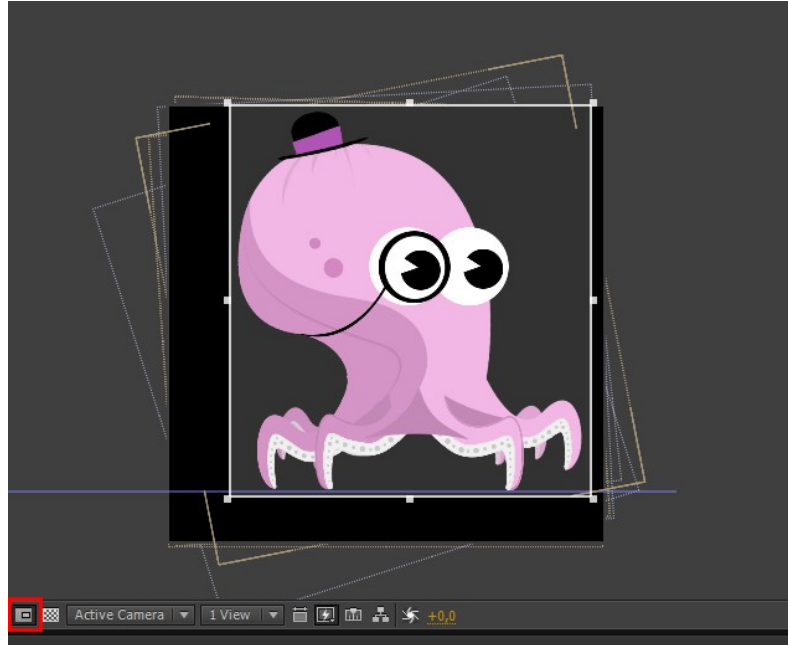
Kuvio 15. Esikatselutilassa näkee samanaikaisesti useampien layereiden liikeradat ja sijainnit. Hahmon keskellä olevassa rykelmässä näkyy layereiden liikerata. Etualalla olevan ruskean komposition vasemmassa reunassa näkyvät vihreä ja punainen nuoli osoittavat sen olevan 3D-layeri. Vihreä nuoli indikoi Y-akselia, punainen X-akselia ja niiden välissä oleva kameraan päin osoittava sininen nuoli, tässä tapauksessa piste, esittää Z-akselia.



Kuvio 16. Aikajanalla voi valita renderöitävän alueen asettamalla punaisella merkityn *Work arean* halutulle ajanjaksolle. Yllä näkyvät aikajanan merkinnät 0F, 5F ja 10F osoittavat, että ohjelmisto tuottaa animaatiosta 11 ruutua.

Kun animaatio on saatu valmiiksi, tulee aikajanalla rajata huolellisesti renderöitävä alue. On hyvä ottaa huomioon kävelyanimaatiota tehdessä, että renderöitävä animaatio palautuu sulavasti hahmon perusasentoon ja on myös toimiva looppi.

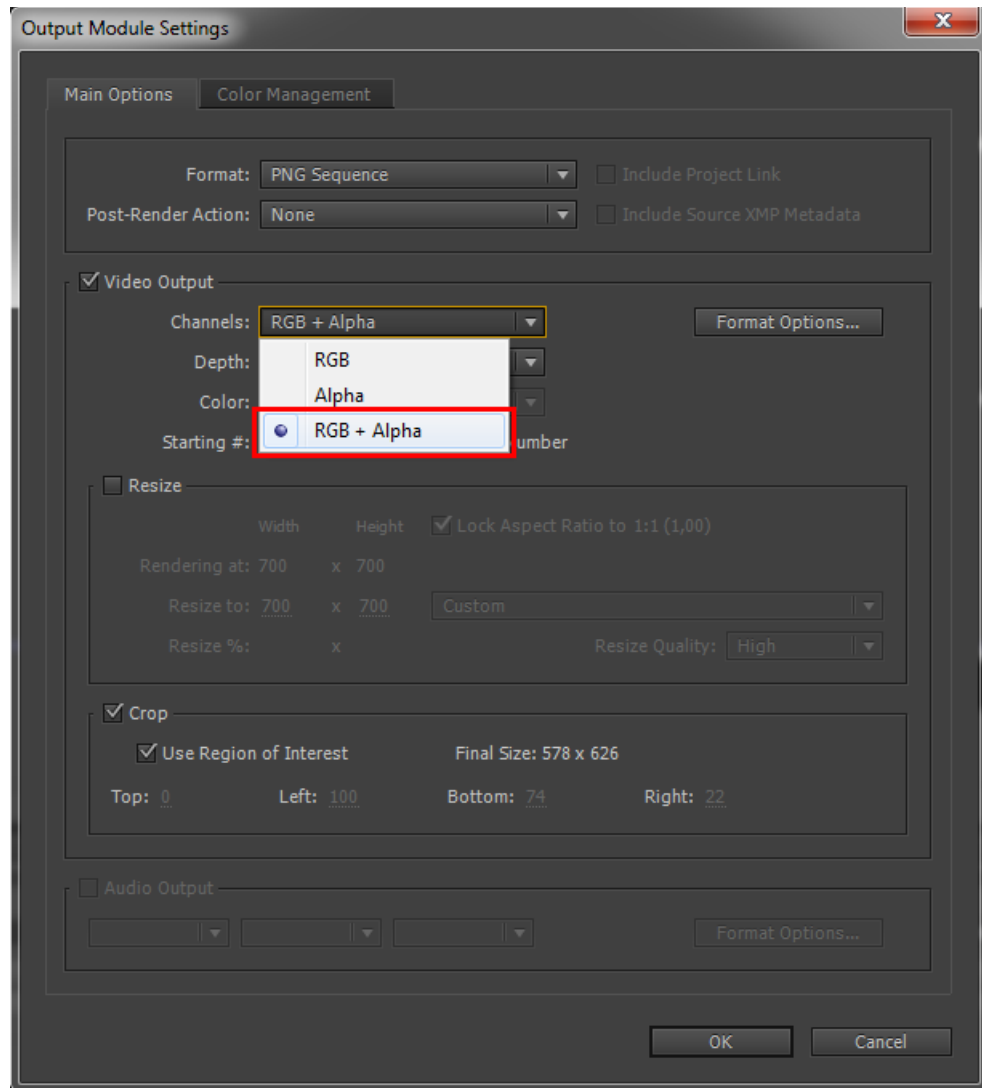
Optimoinnin kannalta on myös hyvä huomioida, ettei animaatioissa ole liikaa ”tyhjää” tilaa. Mitä isommat bittikarttagrafiikat pelissä on, sitä enemmän ne vievät muistia ja rasittavat päätelaitteen tehoja. On siis hyvä rajata tiiviisti animaatiosta alue, jonka sisällä hahmo pysyy koko liikeratansa aikana.



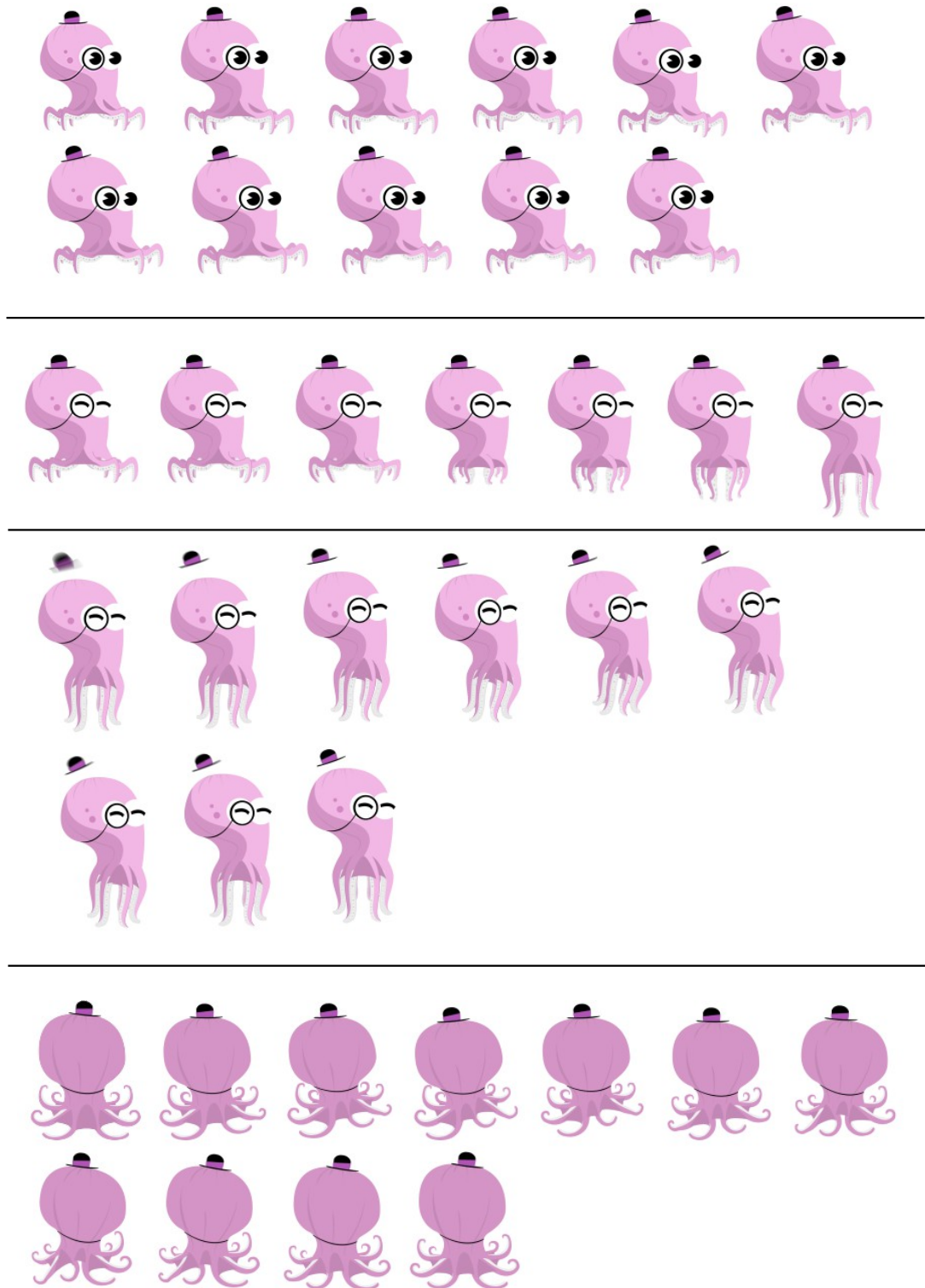
Kuvio 17. Punaisella korostettu ikoni on *Region of interest* -toiminto. Tämän toiminnon avulla voi vaivatta rajata alueen, jonka sisällä koko animaatio pysyy.

Kun mainitut valmistelut on tehty, voi aloittaa komposition renderöinnin. Tässä tapauksessa työstin animaatiot PNG-sekvensseiksi, jolloin After Effects tekee jokaisesta ruudusta läpikuultavan PNG-kuvan. Kyseistä metodia voi käyttää kattavasti myös muihin projekteihin. Esimerkiksi Flashin rajoitetut animaatiotyökalut on mahdollista välttää tuomalla Flash-projektiin After Effectsissä luodut PNG-sekvenssianimaatiot.

On hyvä ottaa huomioon, ettei renderöi PNG -sekvenssiä tarpeettoman korkealla framerateella, sillä liian suuri framerate rasittaa päätelaitteen prosessoritehoja. Käytin projektissani yleisesti frameratea 15 tai 17.



Kuvio 18. Renderöintiasetukset PNG-sekvenssiä varten. Format-valikosta valitaan *PNG Sequence* ja *Channels*-valikosta *RGB + Alpha*. Channels-valikon asetus määrittää ohjelmiston renderöimään animaation sekä RGB-värikanavalla että läpikuultavalla Alpha-kanavalla. Crop-osiossa voi määrittää After Effectsin rajaamaan animaatioalueella joka on valittu *Region of interest*-toiminnolla (ks. kuvio 15.).



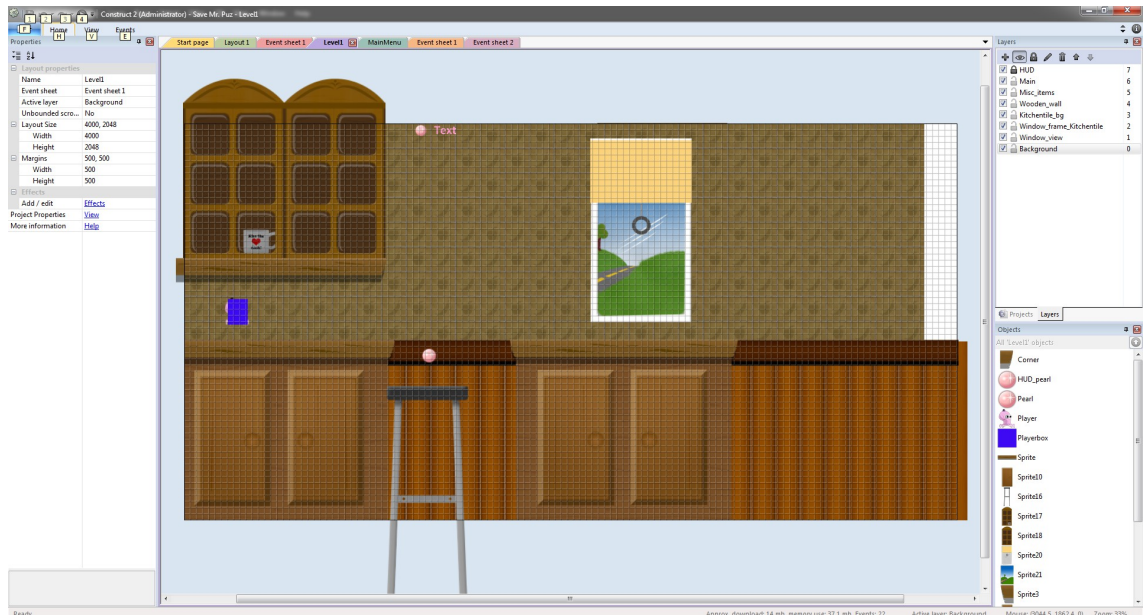
Kuvio 19.Ylimpänä esitettynä PNG-animaatiosekvenssi kävelystä. Kävely-animaatio on luotu loopiksi, eli animaatio toistuu sujuvasti, palautuen tarkoituksellisen vähäiseksi luodun liikeradan vuoksi koska vain hahmon perusasentoon (sekvenssin 1. ruutu). Toisessa sekvenssissä hahmo ponnahtaa hyppyasentoon ja lentoon. Toiseksi alimpänä on esitetty sekvenssi hahmon putoamisesta ja alin sekvenssi esittää kiipeämis-animaatiota.

7.2 Pelin luominen Construct 2:een

Construct 2 on HTML5-pohjainen pelimoottori, joka perustuu Javascript-ohjelmointikieleen (Scirra 2014a). HTML5 on verrattavissa Adobe Flashiin, mutta tukee huomattavasti laajempaa laitekantaa. Flash on erillinen Adoben omistama liitännäinen, joka täytyy erikseen ladata. HTML5 puolestaan on sisäänrakennettu ominaisuus verkkoselaimissa. Lisäksi HTML5 on web-ohjelmoinnissa alan standardi, toisin kuin Flash, jonka toiminnallisuudesta ja turvallisuudesta vastaa ainoastaan Adobe. Lisäksi HTML5 ei ole yhtä haavoittuvainen tietoturvariskeille kuin Flash. (Wikipedia 2014k.)

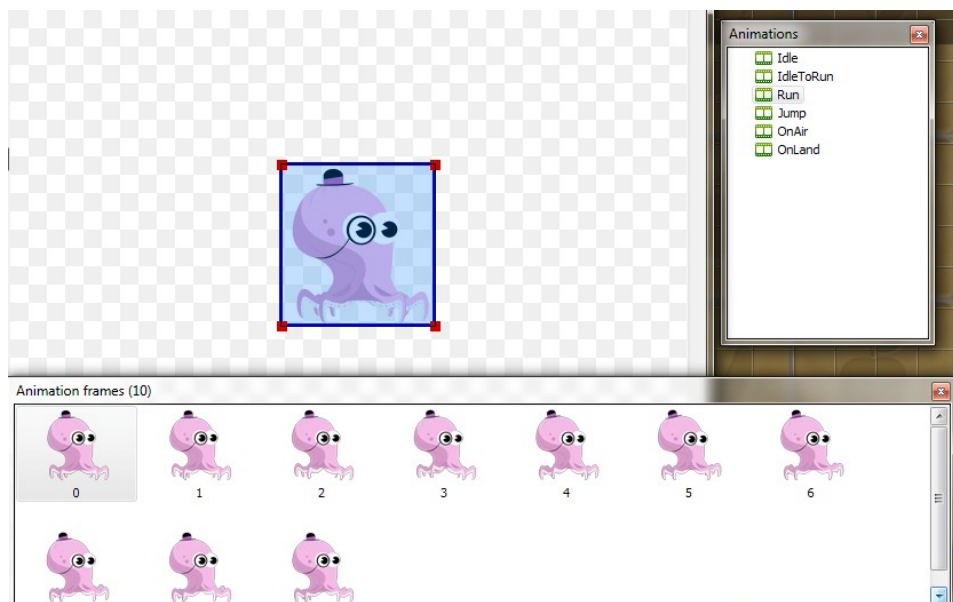
HTML5 siis tarjoaa sujuvampaa interaktiivisuutta laajemmalle laitekannalle. Opinnäytetyötä tehdessä Construct 2 oli suosituin HTML5-pelimoottori (HTML5 Game Engines).

Construct 2:ta on intuitiivista käyttää, ja erityisesti laaja tuki sekä monipuoliset tutoriaalit ovat hyvin tervetulleita ensikäyttäjälle. Käytin hyödykseni useita tutoriaaleja, joihin tulen viittaamaan tässä osiossa. Construct 2:n käyttöliittymän logiikan oppii nopeasti. Keskellä sijaitsee graafinen layout, johon voi asettaa 2D-spriteja sekä tiilitekstuureja omille layereilleen. Layerit näkyvät oikealla yläkulmassa ja projektin grafiikat sijaitsevat vasemmassa alakulmassa. Ruudun vasemmalla puolella olevassa osiossa voi muokata layoutin asetuksia ja määrittää eri grafiikoille ominaisuuksia. Layoutin yläpuolella sijaitsevien värikkäiden välilehtien avulla voi käyttäjä navigoida helposti eri layoutien ja event sheettien välillä (ks. kuvio 20.). Event sheetit ovat layout-kohtaisia sääntö-dokumentteja, joissa määritellään pelimekaniikka, rakenne, kontrollit, pisteytys ja paljon muuta.



Kuvio 20. Kuvakaappauksessa esitettyä Construct 2-käyttöliittymä.

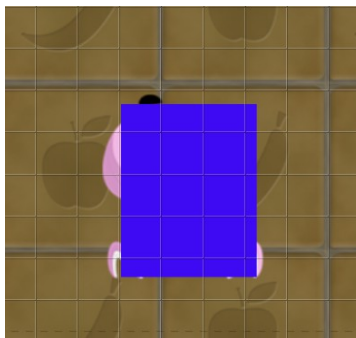
Grafiikoiden tuominen Construct 2:een on helppoa. Käyttäjän tulee vain kaksoisklikata layoutin ulkopuolelle ja valita joko *sprite* tai *tiled background*. Mikäli tarkoitus on toistaa tiettyä tiilitekstuuria isommalla alalla, on tiled background oivallinen ja nopea vaihtoehto. Tiilitekstuuria voi levittää ja skaalata vapaasti kokoon kuin kokoon. Spriteja käytetään yksittäisissä graafisissa elementeissä, joille voi määrittää erilaisia ominaisuuksia. Spriteille voi myös asettaa erilaisia animaatioita, jotka saa laukaistua tapahtumakohtaisesti event sheetissä.



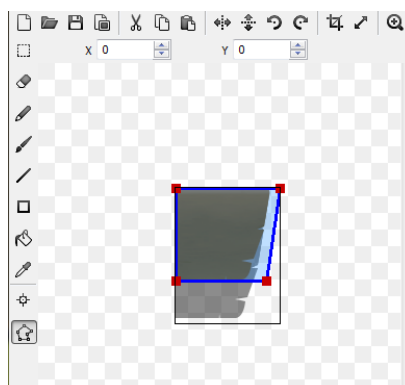
Kuvio 21. After Effectsissa animoidut PNG-sekvenssit ovat jaettuna omiin nimettyihin animaatioihin Construct 2:ssa. Yllä on esitettyä PNG-sekvenssi juoksuanimaatiosta.

Jotta animaatiot saa toistumaan toivottujen toimintojen yhteydessä, täytyy niille määrittää ehtoja ja toimintoja event sheetissä. Saadakseni toivomani toiminnallisuuden pelille, otin mallia useista eri Scirran tutoriaaleista. Tutoriaalit eivät silti olleet täysin linjassa pelini tarpeiden kanssa, joten jouduin keksimään omia keinoja selättämään teknisiä haasteita.

Tavoitteenani oli luoda toiminnallinen pohja perinteiselle tasohypelylle. Tämän pohjan tarjosi tyydyttävästi Scirran tutoriaali *How to make a platform game* (Scirra 2014b.). Tutoriaali tarjosi kattavat tiedot pelin pohjan luomisesta, kontroleista ja logiikasta. Tutoriaali opetti, että pelihahmoon toivotut toiminnot tulee ensisijaisesti määrittää hahmon päälle luotavaan näkymättömään neliögrafiikkaan. Syy tähän on se etteivät toiminnot toimi vakaasti animaatioita sisältävän spriten kanssa. Tällä keinolla vältetään myös ongelmat hahmon törmäysalueen kanssa. Jokaisella Construct 2:een tuodulla spritellä on oletusarvoinen *collision area*, joka merkitsee aluetta, jolla sprite tekee kontaktia muiden elementtien kanssa. Tämä alue on vapaasti muokattavissa muodon mukaisesti Construct 2:n editorissa.



Kuvio 22. Animaatioita sisältävän hahmo-spriteen päälle luotu sininen neliö, johon hahmon toiminnot ja vuorovaikutteisuus linkitetään. Sininen laatikko on määritetty pysymään hahmon päällä näkymättömänä.



Kuvio 23. Kuvakaappaus Construct 2:n sprite-/animaatio-editorista. Esitettynä pöydän päätypalan *collision area*, eli törmäysalue. Sinisin viivoin rajattu alue esittää aluetta, johon hahmo tulisi törmäämään. Huomioi, että pöydän varjo on rajattu törmäysalueen ulkopuolelle. Näin ollen hahmo ei törmää varjoon, vaan on kontaktissa pelkän pöydän kanssa.

	Start page	Level1	MainMenu	Event sheet 1	Event sheet 2	Climb
1	System	Every tick	Player	Set position to Playerbox (image point 0)		
			Text	Set text to "" & Score		
				Add action		
				Global number Score = 0		
2	Keyboard	Up arrow is down	Playerbox	Simulate Platform pressing Jump		
				Add action		
3	Keyboard	On Left arrow pressed	Player	Set Mirrored		
				Add action		
4	Keyboard	On Right arrow pressed	Player	Set Not mirrored		
				Add action		
5	Keyboard	Left arrow is down	Playerbox	Simulate Platform pressing Left		
				Add action		
6	Keyboard	Right arrow is down	Playerbox	Simulate Platform pressing Right		
				Add action		
7	Playerbox	Platform is jumping		Add action		
8	Playerbox	Platform is moving	Player	Set animation to " Jump " (play from beginning)		
				Add action		
9	Playerbox	Platform is moving	Player	Set animation to " Jump " (play from beginning)		
				Add action		
10	Run					
11	Playerbox	Platform On moved	Player	Set animation to " IdleToRun " (play from beginning)		
				Add action		
12	Player	On animation " IdleToRun " finished	Player	Set animation to " Run " (play from beginning)		
				Add action		
13	Playerbox	Platform On stopped	Player	Set animation to " Idle " (play from beginning)		
				Add action		
14	Playerbox	Platform On landed		Add action		
15	Playerbox	Platform is moving	Player	Set animation to " Run " (play from beginning)		
				Add action		
16	Playerbox	Platform is moving	Player	Set animation to " Idle " (play from beginning)		
				Add action		
17				Add action		
18	Playerbox	Platform is falling	Player	Set animation to " OnAir " (play from beginning)		
	Playerbox	Is overlapping Grab_area		Add action		
19				Add action		
20	Playerbox	On destroyed	System	Restart layout		
				Add action		
21				Add action		
22	Playerbox	Is overlapping	System	Add 1000 to Score		

Kuvio 24. Kuvakaappaus Construct 2:n event sheetistä. Event sheettien sinisellä pohjalla olevat vasemmanpuoleiset merkinnät ovat ehtoja, joiden seurauksesta ohjelma toteuttaa oikeanpuoleisia käskyjä. Kuvakaappauksesta näkee kuinka useat toiminnot linkittyvät siniseen *Playerbox* -laatikkoon. Eventsheetin ensimmäinen komento niputtaa *Playerbox* -laatikon ja pelihahmon sprite-grafiikat yhteen. Se myös määrittelee pisteytyksen. Pisteytys-komentojen jälkeen on määritelty kontrollit, animaatiot ja mitä tapahtuu pelaajan pudotessa pois kentältä.

7.2.1 Kontrollit

Kontrollien määrittäminen Construct 2:ssa on suoraviivaista. Hyppäämisen sekä sivuttain liikkumisen määrittäminen onnistuu pienellä vaivalla (ks. kuvio 25.)

3	Keyboard	Up arrow is down	Playerbox	Simulate Platform pressing Jump
			Add action	
4	Keyboard	On Left arrow pressed	Player	Set Mirrored
			Add action	
5	Keyboard	On Right arrow pressed	Player	Set Not mirrored
			Add action	
6	Keyboard	Left arrow is down	Playerbox	Simulate Platform pressing Left
			Add action	
7	Keyboard	Right arrow is down	Playerbox	Simulate Platform pressing Right
			Add action	

Kuvio 25. Ruutukaappaus event-sheetistä. Pelaajalle on määritelty nuolinäppäin-kontrollit liikkumiselle ja hyppäämiselle. Hahmo on määritelty pelaantumaan pelaajan painaessa nuolinäppäintä vastakkaiseen suuntaan.

Kiipeäminen osoittautui haasteellisemmaksi toiminnoksi. Käytin aluksi hyväkseni Scirran tutoriaalia kuinka määrittää toiminnot kiipeämiseksi (Scirra 2014c.). Tutoriaali perustui joko vanhempaan ohjelmistoversioon, tai sitten se oli muuten vain toteutettu hyvin monimutkaisella tavalla. Noudatin tutoriaalia huolellisesti, mutta lopputuloksesta tuli tahmea ja hyvin buginen.

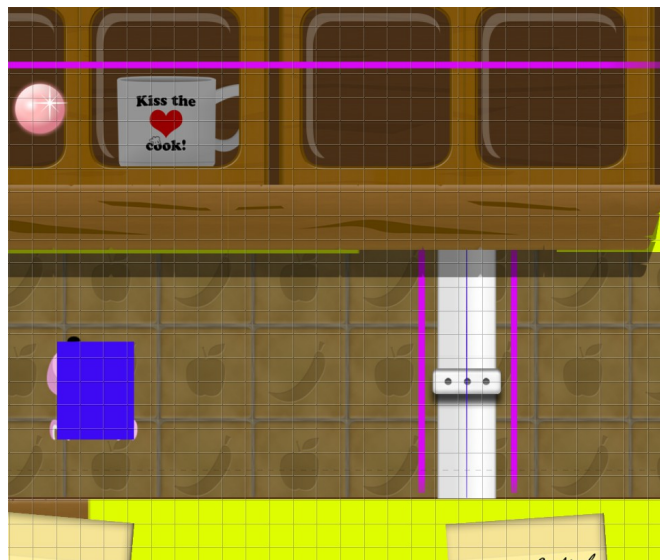
Start page	Level1	MainMenu	Event sheet 1	Event sheet 2	Climb
System	On-start-of-layout	Playerbox	Start-ignoring:8Direction-user-input		
			Add action		
Climb					
Playerbox	Is-CanClimbLadders				Add action
Keyboard	Up-arrow-is-down	Playerbox	Set-OnLadders-to-True		
		Player	Set-animation-to-"Climb_up"- (play-from-beginning)		Add action
Playerbox	8Direction-is-moving	Player	Stop-animation		Add action
Playerbox	Is-OnLadders				Add action
If the player is jumping, get on the ladder with down button					
Keyboard	Down-arrow-is-down	Playerbox	Set-OnLadders-to-True		Add action
Playerbox	Platform-is-on-floor				Add action
Get off ladder by jumping off it					
Keyboard	On-Left-arrow-pressed	Playerbox	Set-OnLadders-to-False		Add action
Playerbox	Is-CanClimbLadders	Playerbox	Move-120-pixels-at-angle-240		Add action
Playerbox	Is-CanClimbLadders	Playerbox	Set-Platform-Enabled		Add action
Playerbox	Platform-pressing-Jump				Add action
Get off ladder when above it					
Playerbox		Playerbox	Set-OnLadders-to-False		Add action
Get off ladder when touching the floor below					
System	Else	Playerbox	Set-OnLadders-to-False		Add action
Playerbox	Is-overlapping-Table-at-offset-(0,-20)				Add action
Playerbox	Y-<=Table.Y				Add action
Keyboard	Down-arrow-is-down				Add action
Get off ladder on pressing down					
Keyboard	Down-arrow-is-down	Playerbox	Set-OnLadders-to-False		Add action
Playerbox	Is-OnLadders	Playerbox	Set-CanClimbLadders-to-False		Add action
Playerbox	Is-overlapping-Table-at-offset-(0,-20)				Add action
Ladders implementation					
Ladders implement.					
Playerbox	Is-OnLadders	Playerbox	Set-Platform-Disabled		Add action
Keyboard	Up-arrow-is-down	Playerbox	Stop-ignoring:8Direction-user-input		Add action

Kuvio 26. Scirran tutoriaalin mukaisesti määritelty kiipeämistoiminto. Toteutus on hyvin monimutkainen, eikä se toiminut sulavasti.

Minun täytyi keksiä kiipeämistoiminnolle täysin oma ratkaisu. Kehitin oman toteutuksen hyödyntäen painovoiman arvoja ja näkymättömiä sprite-objekteja. Sprite-objektit rajaavat alueen jossa pätee omat säännöt kiipeämiselle. Putken kohdalla tulee voimaan miinus-arvoinen painovoima pelaajan painaessa nuolinappia ylöspäin. Tällöin pelaaja käytännössä "putoaa" ylöspäin kiipeämisen aikana. Tällöin käynnistyy myös *Climb* -niminen kiipeämisanimaatio. Pelaajan ollessa putken päällä ja vapauttaessaan ylänuolinapin, hahmo pysähtyy ja painovoima nollaantuu. Pelaajan osuessa kiipeämisaluetta rajaavaan pinkkiin sprite -objektiin palautuu pelikentän normaalit arvot kohdalleen. Olen hyvin tyytyväinen, kun keksin huomattavasti yksinkertaisemman sekä toimivamman ratkaisun kiipeämiselle.

23	<ul style="list-style-type: none"> Playerbox Is overlapping Putki Keyboard Up arrow is down 	<ul style="list-style-type: none"> Playerbox Set Platform gravity to -200 Table Set Solid Disabled Playerbox Set X to Grab_area.X +1 Player Set animation to "Climb_up" (play from beginning)
24	<ul style="list-style-type: none"> Keyboard On Up arrow released Playerbox Is overlapping Putki 	<ul style="list-style-type: none"> Playerbox Set Platform gravity to 200 Playerbox Set X to Grab_area.X +1 Player Stop animation
25	<ul style="list-style-type: none"> Playerbox Is overlapping Grab_area Keyboard Down arrow is down 	<ul style="list-style-type: none"> Playerbox Set Platform gravity to 400 Player Stop animation
26	<ul style="list-style-type: none"> System Else Playerbox Is overlapping Climb_border 	<ul style="list-style-type: none"> Table Set Solid Enabled Playerbox Set Platform gravity to 2500

Kuvio 27. Kuvakaappaus keksimästäni ratkaisusta. Hyödynsin näkymättömiä sprite-objekteja ja painovoimaa.



Kuvio 28. Layout-näkymä kiipeämisalueesta. Keltaiset sprite-objektit on määritelty toimimaan turvaverkkoina, ettei pelaaja putoa kiivetessä pöytätasojen läpi.

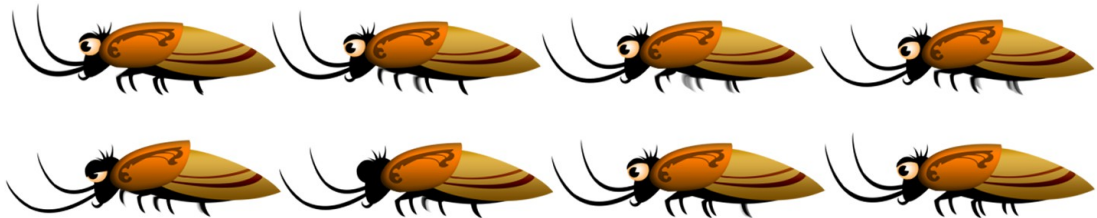
7.2.2 Viholliset

Loin peliin yhden staattisen sekä dynaamisen vihollisen. Staattisena vihollisena toimii liesi, joka hohkaa varoittavaa kuumaa hehkua. Piirsin liehen Photoshopissa ja lisäsin siihen yksinkertaisen hehku-animaation After Effectsissä. Määritin animaation Construct 2:ssa toistumaan looppina.

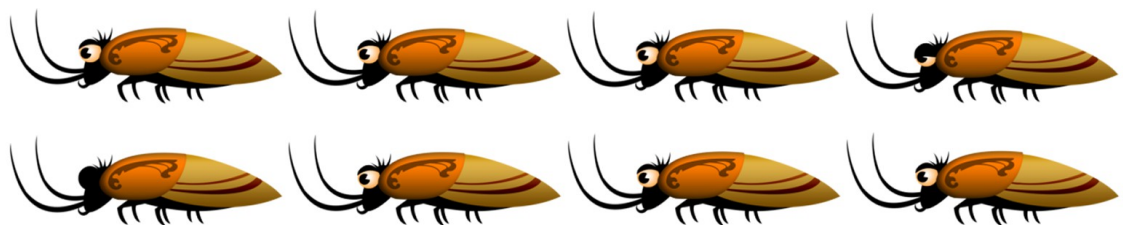


Kuvio 29. PNG-sekvenssi liehen hehkuanimaatiosta.

Dynaamiseksi viholliseksi loin torakan, jonka määritin liikkumaan itsenäisesti yksinkertaisen tekoälyn avulla. Sovelsin samaa workflowta torakan grafiikoiden ja animaatioiden tekoon, kuin mitä käytin pelin päähahmoon. Tarkoitukseni oli saada torakka kulkemaan edestakaisin tietyn alueen sisällä ja pysähtyä hetkeksi alueen ääripäissä. Tätä varten loin animaatiot kävelylle ja paikoillaan ololle (ks. kuviot 30 & 31.).



Kuvio 30. PNG-sekvenssi torakan juoksuanimaatiosta.



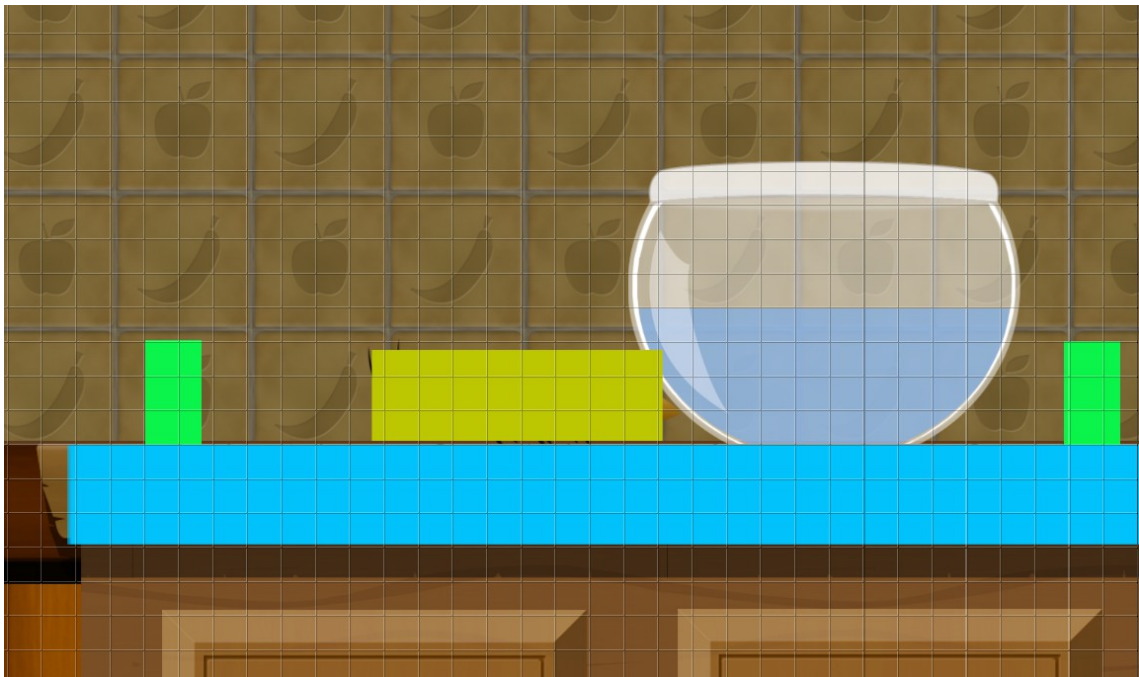
Kuvio 31. PNG-sekvenssi torakan paikoillaan ololle.

Saadakseni halutut toiminnallisuudet Construct 2:ssa minun tuli määritellä torakalle oma näkymätön neliögrafiikka, johon kohdistan toivomani animaatiot ja komennot. Lisäksi torakan liikkumisalueen ääripäihin täytyi luoda näkymättömät sprite -objektit laukaamaan kahden sekunnin tauko sekä pysähdysanimaatio. Tämän jälkeen tuli laukaista kävelyanimaatio sekä käännättää torakka liikkumaan vastakkaiseen suuntaan. Käytin torakan toimintojen määrittelemiseen Scirran tutoriaalia, jota minun tuli muokata saadakseni peliin omat kustomoidut toiminnallisuudet (Scirra 2014d.). Tutoriaalissa ei ollut määriteltynä toimintoja tauolle ja ylimääräiselle animaatiolle, joten minun tuli keksiä oma ratkaisu (ks. kuvio 32.).

34	Roach_box	action = "right"	Roach_b...	Simulate Platform pressing Right
			Roach	Set Mirrored
			Roach	Set animation to "Run" (play from beginning)
			Roach_b...	Set Platform maximum speed to 300
			Add action	
35	Roach_box	action = "left"	Roach_b...	Simulate Platform pressing Left
			Roach	Set Not mirrored
			Roach	Set animation to "Run" (play from beginning)
			Roach_b...	Set Platform maximum speed to 300
			Add action	
36	Roach_box	On collision with Edge_Trigger	Roach_b...	Set action to "idle"
	Roach_box	action = "right"	Roach_b...	Set Platform maximum speed to 0
			Roach	Set animation to "Idle" (play from beginning)
			System	Wait 2 seconds
			Roach_b...	Set action to "left"
			Add action	
37	Roach_box	On collision with Edge_Trigger	Roach_b...	Set action to "idle"
	Roach_box	action = "left"	Roach_b...	Set Platform maximum speed to 0
			Roach	Set animation to "Idle" (play from beginning)
			System	Wait 2 seconds
			Roach_b...	Set action to "right"
			Add action	
38	Playerbox	On collision with Roach_box	Add action	
39	Playerbox	Platform is falling	Roach_b...	Destroy
	Playerbox	Y < Roach_box.Y	Roach	Destroy
			Playerbox	Set Platform vector Y to -700
			System	Add 250 to Score
			Add action	
40	System	Else	Player	Flash: Flash 0.1 on 0.1 off for 1.0 seconds
			Playerbox	Subtract 1 from Health
			Add action	

Kuvio 32. Kuvakaappaus torakan toiminnosta event sheetissä. Torakalle on määritelty tietty liikkuma-alue, jonka ääripäissä se pysähtyy kahden sekunnin ajaksi. Tämän jälkeen torakka kääntyy takaisin vastakkaiseen suuntaan. Event sheettiin on myös määritelty interaktio pelaajan kanssa.

Vastaan tuli ongelma jossa torakka oli kadonnut alueelta jossa sen tuli liikkua edestakaisin. Syyksi paljastui pelihahmon kiipeämistoiminnolle määritetyt säännöt, jossa pöytätasot menettävät kiinteän-ominaisuuden pelaajan kiivetessä ylös pöydän "läpi". Ratkaisin ongelman luomalla näkymättömän kiinteän spriten torakan kulkualueen alapuolelle. Tämä sprite toimii itsenäisesti säilyttäen kiinteän ominaisuuden riippumatta siitä kiipeäkö pelaaja vai ei (ks. kuvio 33.).



Kuvio 33. Kuvakaappauksessa esitettynä näkymättömät sprite-objektit, jotka määrittelevät torakan kulkualueen ja logiikan (ks. kuvio 32.).

7.2.3 Pisteytys ja muuttujat

Pelin pisteytys on määritelty *Score*-muuttujalla. Muuttujan aloitusarvoksi on asetettu "0", ja pisteytys on määritelty erikseen tuhotusta torakasta ja kerätystä helmestä. Demoa varten määritin pelaajan saavan 750 pistettä helmestä, ja 250 pistettä torakan tuhoamisesta (Scirra 2014e.).

Pelihahmolle on määritelty *Health* -muuttuja, jonka arvoksi on asetettu "3". Tämä arvo on määritelty muuttuvan pelihahmon osuessa viholliseen tai pudotessa pelialueen ulkopuolelle. Mikäli näin käy, on event sheettiin merkitty : vähennä "1" muuttujasta *Health* (Scirra 2014e.).

HUD:ssa näkyvä elämä-indikaattori on linkitetty vastaamaan arvon muuttumiseen. HUD:in elämä-indikaattori on yksinkertainen sprite, joka sisältää kolme framea esittämään pelihahmon elämien määrän. Indikaattori määritelty toimimaan siten, että muuttujan ollessa arvoltaan 3, toistuu spritessa kolmas frame. Vastaava logiikka on määritelty arvoille 1-2. Myös pelihahmon spriteen on määritetty *Flash* -toiminto, jolla voi määrittää hahmon vilkkuvan osutessaan viholliseen. Tämä antaa pelaajalle visuaalista palautetta siitä, että vihollisia tulee varoa.

Muuttujan ollessa tasan 1 pelaaja häviää pelin, mikäli vielä törmää viholliseen tai putoaa pelialueelta. Tällöin muuttuja tulisi muuttumaan 0-arvoon, jolloin ilmestyy ruutu jossa kysytään haluaako pelaaja yrittää uudelleen. Kyseinen ruutu on oma erillinen layoutinsa, kuten päävalikko sekä pelikenttä. *Try again* -nappia painamalla kenttä käynnistyy uudelleen.

7.2.4 Äänimaailma

Käytin Construct 2:n mukana tullutta rojaliti-vapaata Matt Oglsebyn musiikkia sekä kentässä että päävalikossa. Ohjelmiston mukana tuli myös äänitehosteita joita käytin päävalikon *Play* -napin painallukseen ja helmien keräämiseen.

7.3 Haasteet

Opinnäytetyötä tehdessäni tuli vastaan aikataulullisia rajoitteita, joiden vuoksi jouduin karsimaan alkuperäisiä tavoitteitani pelin suhteen. Alunperin oli tarkoituksena tehdä peli Android- tai Windows Phone-alustalle. Tämä osoittautui olevan isompi työ kuin osasin aavistaa, ja minun täytyi keskittyä selain-pohjaisen pelin tuottamiseen. Mobiilialustalle julkaiseminen on silti prosessi, jonka haluaisin toteuttaa tulevaisuudessa. Haaveena oli myös useamman kentän suunnittelu ja intro-animaation luominen. Nämä työt osoittautuivat työtaakaltaan suureksi, joten jouduin jättämään ne välistä. Edellä mainitut tehtävät, joita ei ollut mahdollista toteuttaa, kuuluvat *wish* -listiin.

7.4 Epäonnistumiset

Animoimisessa tuli vastaan ongelmia hyppyanimaatioissa, koska se sisälsi yhteensä kolme eri asentoa, jotka täytyi sulauttaa saumatta yhteen. Yksinkertainen mutta aikaa vievä vaihtoehto olisi ollut piirtää jokainen ruutu kerrallaan. Käytin hyväkseni nopeampia keinoja käyttäen hyväkseni *muppet* -työkalua ja *frame blendingia*. Mainittu nopea vaihtoehto ei osoittautunut kovin toimivaksi. Koska muppet-työkalu väärästi

pahoin grafiikkaa, tuli minun käyttää frame blending -toimintoa peittääkseni tehdyt virheet. Tämä auttoi hiukan, mutta jätti silti sotkua grafiikoihin. Jouduin jälkikäteen korjaamaan muutamia frameja Photoshopissa. Totesinkin tämän prosessin jälkeen, että järkevintä olisi ollut ruutujen manuaalinen piirtäminen. Onnekseni pelihahmo on hyvin pieni ja animaatio toistuu sen verran nopeasti, ettei silmä ehdi nähdä pieniä kauneusvirheitä. Ymmärrettävästi grafiikoiden häviötön laatu on silti jokaisen artistin tärkeimpiä tavoitteita.



Kuvio 34. Kuvassa esitettynä frame blending -toiminnon tuottamia vääristymiä.

Ymmärsin peliä kehittäessäni sen, että hahmoni olisi ollut parempi luoda vektorigrafiikkana Adobe Illustratorissa. Tämä johtuu siitä, että hahmon grafiikoita pystyisi jälkikäteen skaalamaan tarpeen mukaan myös mahdollista markkinointia varten. Suurin osa grafiikoista oli silti tarpeellista luoda bittikarttagrafiikkana Photoshopissa.

7.5 Onnistumiset

Olen tyytyväinen lopputulokseen, vaikka jouduinkin tekemään kompromisseja pelin suhteen. Koen sen saavutuksena, että onnistuin opinnäytetyössäni saattamaan sekä kirjallisen että toiminnallisen osan toivomaani kuntoon. Työssä piti tuottaa hyvin paljon omaa sisältöä ja tämä osoittautui paikoittain erittäin työlääksi.

Opin ennen kaikkea eniten Construct 2:sta, joka oli minulle lähtökohtaisesti täysin tuntematon työkalu. Construct 2:n luonut yritys Scirra lupaa sivuillaan, ettei ohjelmisto vaadi ohjelmointitaitoja ja että sitä on intuitiivista käyttää. Tämän lupauksen voin todeta Scirran lunastaneen. Pidän työskentelystä ja uskon käyttäväni oppimiani taitoja jatkossa omien pelien tekemiseen.

8 Yhteenveto

Olen hyvin kiinnostunut pelien tuottamisesta ja olinkin positiivisesti yllättynyt, että pelejä on mahdollista luoda ilman ohjelmointitaitoja. En ole löytänyt itsestäni vaadittavaa pitkäjänteisyyttä ja intoa selättää koodaamisessa ilmeneviä haasteita – niitä todellakin tulee. Voin vain todeta, ettei ohjelmoiminen tule koskaan olemaan artistina itselleni mieluisa tehtävä. Olen siis huojentunut ja kiitollinen, että pelintuotanto-ohjelmistoja on luotu juuri minun kaltaisiani koodia taittamattomia varten.

Pelien tuotanto on helpottunut huomattavasti työkalujen kehittyessä. Nykyään ei ole välttämättömyys osata ohjelmoida halutessaan luoda yksinkertaisia pelejä. Ohjelmoiminen on silti välttämätön taito halutessaan luoda täysin kustomoitavan ja koodin kannalta puhtaan pelin. Lisäksi ohjelmointitaidoilla on mahdollista luoda huomattavasti sisältörikkaampaa tuotantoa. Pelintuotanto-ohjelmilla pystytään tällä hetkellä luomaan pelejä vain tiettyjen ehtojen ja rajoitusten puitteissa. Lisäksi koodin laatu ei välttämättä hivle ohjelmoijan silmää. Construct 2:n kaltaisten ohjelmistojen ei missään nimessä olekaan tarkoitus syrjäyttää ohjelmoijia, vaan antaa koodaamista osaamattomille mahdollisuus luoda pelejä.

Peliharrastajana ja artistina koen pelintuotanto-ohjelmistot erinomaisena mahdollisuutena saada omasta ideasta todellisuutta, vaikka koodi ei tulisikaan olemaan täydellistä. Roolini ja taitoni tiedostaen koen ensisijaisena saavutuksena pelin toimivuuden. Koodin puhtauden ja logiikan näen toissijaisena, sillä niihin en pysty taitojeni puitteissa puuttumaan. Työtäni tarkastellessa tulisi tämä siis pitää mielessä - koodi voi olla kaaottista, mutta se toimii.

Opinnäytetyöni hyödyntää niitä, joita kiinnostaa pelien tuottaminen itsenäisesti pienellä budjetilla. Uskon, että demonstroimani keinot ovat hyödyllisiä ja niiden avulla voi luoda yksinkertaisia 2D-pelejä. Toivon, että opinnäytetyöstäni saa inspiraatiota ja tukea

pelien tekoon. Vaikka työni keskittyy tiiviisti pelin tuotantoon, voi erityisesti After Effectsiä koskeva osio olla hyödyllinen siitä kiinnostuneille.

Construct 2 sisältää tiettyjä rajoitteita, mutta niiden puitteissa voi silti luoda ainutlaatuisia pelejä. Ainoastaan mielikuvitus on rajana.

9 Lähteet

Adobe After Effects Help: Composition basics [verkkosivu]

<<http://helpx.adobe.com/after-effects/using/composition-basics.html>>

(Luettu 18.3.2014)

HTML5 Game Engines

<<http://html5gameengine.com/>> (luettu 8.4.2014)

Indie Game: The Movie. 2012. Pajot Lisanne, Swirsky James. Kanada: BlinkWorks Media. 103 min.

Kajantie Noora 2013. Suomen pelialan loistoajat jatkuvat - silti 99 % hyvistä ideoista jää toteuttamatta. Yle Uutiset. [verkkosivu]

<http://yle.fi/uutiset/suomen_pelialan_loistoajat_jatkuvat_-_silti_99_hyvista_ideoista_jaa_toteuttamatta/6487733> (Luettu 6.4.2014)

Kinnunen Joel & Lyytikäinen Susanna, 2010. Peliharrastuksesta syntyi menestyvä yritys. Toim. Kyllönen Kimmo, Lehtonen Saija: Sähköistä pelikulttuuria Electrasta Pongiin, Pongista Pleikkaan. Ykkösoffset, Mediamuseo Rurpiikin julkaisuja 4. Mediamuseo Rurpiikki, Perinneyhdistys Elektra ry.

Kuvio 9. Dafont.

<<http://www.dafont.com/img/charmap/c/o/cobac0.png>>

[haettu 24.2.2014]

Lehtonen Saija & Kyllönen Kimmo 2010. Electrasta Pongiin – Pongista Pleikkaan. Toim. Kyllönen Kimmo, Lehtonen Saija: Sähköistä pelikulttuuria Electrasta Pongiin, Pongista Pleikkaan. Ykkösoffset, Mediamuseo Rurpiikin julkaisuja 4. Mediamuseo Rurpiikki, Perinneyhdistys Elektra ry.

Sihvonen Tanja, Mäyrä Frans, Saarenpää Hannamari, Kultima Annakaisa, Paavilainen Janne & Stenros Jaakko 2010. Pelikulttuurin muodot ja kehittyminen. Toim. Kyllönen Kimmo, Lehtonen Saija: Sähköistä pelikulttuuria Electrasta Pongiin, Pongista Pleikkaan. Ykkösoffset, Mediamuseo Rurpiikin julkaisuja 4. Mediamuseo Rurpiikki, Perinneyhdistys Elektra ry.

Scirra 2014a. Technology [verkkosivu]

<<https://www.scirra.com/manual/35/technology>> (luettu 8.4.2014)

Scirra 2014b. How to make a platform game [verkkosivu]

<<https://www.scirra.com/tutorials/253/how-to-make-a-platform-game/how-to-make-a-platform-game/page-1>> (Luettu 20.1.2014)

Scirra 2014c. Simple Ladder for Platformer Game [verkkosivu]

<<https://www.scirra.com/tutorials/548/simple-ladder-for-platformer-game>>
(luettu 11.3.2014)

Scirra 2014d. How to make a platform game: adding an enemy [verkkosivu]

<<https://www.scirra.com/tutorials/253/how-to-make-a-platform-game/page-7>>
(luettu 27.3.2014)

Scirra 2014e. Beginner's guide to Construct 2: Instance variables [verkkosivu]

<<https://www.scirra.com/tutorials/37/beginners-guide-to-construct-2/page-6>>
(luettu 14.2.2014)

Suvanto Soile 2014. Peliteollisuus on jo miljardibisnestä Suomessa. Yle Oulu.
[verkkosivu]

<http://yle.fi/uutiset/peliteollisuus_on_jo_miljardibisnesta_suomessa/7169571>
(Luettu 6.4.2014)

Vuorela Ville 2009. Elämäpelejä, pelintekijän maailmat. Helsinki : BTJ Finland Oy.

Wikipedia 2014a. Sprite computer graphics [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=Sprite_%28computer_graphics%29&oldid=604920650> (Luettu 18.3.2014)

Wikipedia 2014b. Frame rate [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=Frame_rate&oldid=604179268>
(Luettu 18.3.2014)

Wikipedia 2014c. High Frame Rate [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=High_frame_rate&oldid=603554578>
(Luettu 18.3.2014)

Wikipedia 2014d. Gameplay [verkkosivu]

<<http://en.wikipedia.org/w/index.php?title=Gameplay&oldid=600973724>>

(Luettu 18.3.2014)

Wikipedia 2014e. Power-up [verkkosivu]

<<http://en.wikipedia.org/w/index.php?title=Power-up&oldid=602635652>>

(Luettu 18.3.2014)

Wikipedia 2014f. HUD (video gaming) [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=HUD_%28video_gaming%29&oldid=604716576> (luettu 31.1.2014)

Wikipedia 2014g. Space Panic [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=Space_Panic&oldid=586215277>

(Luettu 23.2.2014)

Wikipedia 2014h. Platform Game [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=Platform_game&oldid=604452393>

(Luettu 31.1.2014)

Wikipedia 2014i. Suomen videopelialan historia [verkkosivu]

<http://fi.wikipedia.org/w/index.php?title=Suomen_videopelialan_historia&oldid=13698692>

(Luettu 6.4.2014)

Wikipedia 2014j. Demoskene [verkkosivu]

<<http://fi.wikipedia.org/w/index.php?title=Demoskene&oldid=13728058>>

(Luettu 23.2.2014)

Wikipedia 2014k. Adobe Flash [verkkosivu]

<http://en.wikipedia.org/w/index.php?title=Adobe_Flash&oldid=603781341> (luettu 8.4.2014)

Liite1. ScreenPlay_MrPuz_intro.pdf

Intro-animaation kuvakäsikirjoitus – pdf-dokumentti

Liite 2. Save Mr.Puz-pelidemo.zip

Save Mr. Puz-pelidemo – ZIP-tiedosto