



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Sami Porkka

# Peliprototyypin kehitys pienyrityksessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

18.11.2018

Tekijä Otsikko	Sami Porkka Peliprototyypin kehitys pienyrityksessä
Sivumäärä Aika	46 sivua 18.11.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintäteknikka
Ammatillinen pääaine	pelisovellukset
Ohjaajat	lehtori Antti Laiho toimitusjohtaja Henri Taussi
<p>Insinööriyössä pyrittiin selvittämään, kuvaamaan ja kokeilemaan pelin prototyypin kehityksen kaari alusta loppuun saakka. Työssä kokeiltiin nopean prototyypin kehitystä ja sen kaikkia vaiheita, peli-idean suunnittelua, peliin vaadittavan osaamisen tunnistamista, pelin sisällön suunnittelua, pelin sisällön hankintaa ja toteutusta.</p> <p>Projektissa aloitettiin pelinkehitys ideatasolta, jolloin päädyttiin ensimmäisen persoonan ammuntapeliin ja 2D-jalkapalloammuntapeliin. Työssä toteutettiin pelin ideointi ja idean skausaus, prototyyppi, lisäominaisuuksia ja hiottuja perusominaisuuksia sekä kaikki oleellinen prototyypin kehityksen vaiheista alusta loppuun.</p> <p>Havaittiin, että ensimmäisen persoonan ammuntapeli on mahdollista toteuttaa yksin tai kaksin, mutta projekti vaatii lähes täyspäiväistä sitoutumista, jos kehitysaika halutaan pitää järkevän lyhyenä. Insinööriyössä toteutettiin ensimmäisen persoonan ampuminen ja vuorovaikutus. Myös muun muassa vihollisten kuolema, pelaajan elämät, rekyyli, aseiden vaihto ja aseiden heitto ovat toteutettuja ominaisuuksia. Myös 2D-ammuntajalkapallopelistä saatiin hyvä prototyyppi ja huomattiin, että se on paremmin toteutettavissa osa-aikaisella työskentelyllä, koska pelin koko on pienempi kuin ensimmäisen persoonan pelin. Saatiin siis tehtyä kaksi hyvää prototyyppiä, joista toisen kehitystä jatketaan.</p> <p>Julkisesti peliä ei päästy näyttämään, vaikka se oli yksi tavoitteista. Nyt on kuitenkin tarkka suunnitelma, miten projektin kanssa edetään, miten peliä markkinoidaan ja missä se julkaistaan.</p> <p>Projekti opetti paljon prototyypin kehityksen eri vaiheista, ja varsinkin pelin koon suunnittelusta ja nopean prototyypin tekemisestä tiedetään nyt paljon enemmän. Muun muassa se, kuinka keskittyä oikeisiin asioihin peliä tehdessä, ja se, että tehokas ja nopea ominaisuuksien kokeilu ja suunnittelu ovat tärkeitä asioita prototyypin kehityksessä.</p>	
Avainsanat	pelinkehitys, nopea prototyyppi, pelisuunnittelu, prototyyppi

Author Title	Sami Porkka Developing a game prototype in a small business
Number of Pages Date	46 pages 18 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Game Applications
Instructors	Antti Laiho, Senior Lecturer Henri Taussi, CEO
<p>This project explains the whole cycle of developing a game prototype. Including: rapid prototyping, coming up with a doable game idea, recognizing the knowledge needed to make the game, designing the game and getting all the assets for the game.</p> <p>The project started from scratch with the idea of a first-person shooter and a platformer football shooter. The projects went through multiple faces, like scaling, prototyping and adding features. This work covered many important steps of prototype development.</p> <p>It was found out that a first-person shooter is doable alone or with a partner, but it needs almost daily commitment if a developer wants to keep the time of the development reasonable. This project was a success because there is a deep shooting mechanism and interaction with the game world. It was also found out that the football shooter game is manageable even as a part-time project because of its smaller scale.</p> <p>The games did not see daylight yet even if that was one of the goals. Nevertheless, the game now has a precise plan how to move on with the project when the time is right.</p> <p>This project taught a lot about game development and developing a game prototype and every step there is when making a game prototype. Especially scaling the game and rapid prototyping had a big impact. Lots of this is very important to know or at least familiarize with when making a game and wanting it to success.</p>	
Keywords	game development, rapid prototyping, game design, FPS

## Sisällys

1	Johdanto	1
2	Peliprototyypin ja pelin suunnittelu	2
2.1	Pelin prototyyppi ja kehitys	2
2.2	Prototyypin tai pelin idea	2
2.3	Yksin tai yhteistyönä	4
2.4	Projektin koko ja tapahtumat rajana	6
3	Peliprototyypin kehitys ja pelinkehitystyökalut	7
3.1	Pelimoottori ja muut työkalut	7
3.2	Nopea mielentila prototyypin kehityksessä	10
3.3	Prototyypin ydin	11
3.4	Kehityksen ajankäyttö ja riskien ottaminen	12
3.5	Jumiin jääminen ja sen välttäminen	13
3.6	Prototyypin tavoite ja jälkitoimet	14
4	Prototyyppien kehitys	17
4.1	Peli-idea ja pelimoottorin valinta	17
4.2	Prototyypin ensimmäiset askeleet	20
4.3	Prototyyppien perusominaisuudet	22
4.4	Prototyyppien lisäominaisuudet	25
4.5	Tasot, viholliset, aseet ja pelaaminen	28
4.6	Huumori, komedia ja tarina	31
5	Valmis prototyyppi	33
5.1	Prototyyppien lopputulokset	33
5.2	Prototyyppien seuraavat askeleet	36
6	Yhteenveto	37
	Lähteet	39

## 1 Johdanto

Insinööriyössä syvennytään pelin prototyypin kehittämisen eri vaiheisiin alusta loppuun: aloituksesta ja suunnittelusta kehittämiseen ja siihen, mitä sen jälkeen kannattaa ja voi tehdä. Tämä kaikki käydään läpi yksittäisen kehittäjän tai pienen studion näkökulmasta. Suurin osa asioista kuuluu myös isojen pelien kehitykseen, vaikka prototyypin ja pelien kehitysajat ovatkin paljon pidemmät ja rahaa on enemmän.

Prototyypin kehitystä voi lähestyä monelta suunnalta, mutta runko toimii yleensä hyvin samanlaisena ja samat säännöt ja toteutukset pätevät. Prototyypin kehitys on myös hyvä tapa kehittää itseään, minkä takia se on hauska ja hyvä tapa tulla paremmaksi työssään. Tämän insinööriyön luettuaan lukijan on mahdollista aloittaa prototyypin kehitys oikein ja ilman turhia kompastuksia tai ajan haaskausta.

Työn lähteet ja esimerkit on poimittu eri kohdassa uraansa olevilta pelinkehittäjiltä. Mukana on kehittäjiä isoista menestysyrityksistä, pieniä yksin kehittäjiä, jotka ovat menestyneet, ja keskikokoisia pelistudioita, jotka onnistuvat elättämään itsensä pelejä tehden.

Työssä pyritään myös tekemään suurin osa asioista itse, jotta on mahdollista jakaa omia kokemuksia, onnistumisia ja epäonnistumisia. Työssä on käytetty kahta eri prototyyppiä, täysin eri genreistä. Työssä tarkastellaan aihetta pienyrittäjän näkökulmasta, joka pyrkii luomaan oman menestystuotteensa. Kaikkia näkökulmia ei voi soveltaa isoissa yrityksissä, joissa menestyspelien resepti on jo olemassa. Työ on tehty oma peliyritys, Baracuda Disaster, mielessä. Se on tuore neljän hengen peliyritys, joka tekee tarinallisia, hauskoja ja erilaisia pelejä konsoleille, tietokoneille ja virtuaalitodellisuuteen.

Prototyypin tekeminen on paras tapa aloittaa pelinkehitys, minkä takia jokaisen pelejä kehittävän tai pelialalla työskentelevän on tärkeä tietää, miten prototyypin kehitys toimii, varsinkin jos on kyseessä oma yritys tai peli, koska silloin koko talous on kiinni pelin menestyksestä. Prototyypin kehittämällä voi huonot pelit tappaa jo heti alussa.

## 2 Peliprototyypin ja pelin suunnittelu

### 2.1 Pelin prototyyppi ja kehitys

Aloitteleville pelintekijöille on tärkeää, että ei mene liian nopeasti eteenpäin. On helppoa jäädä tutkimaan ja ihmettelemään työkaluja, alustoja, markkinointia, palkintoja ja muuta, ennen kuin on saatu mitään aikaiseksi. Siinä tilanteessa on helppo sokaistua ja jäädä jumiin. On tärkeää ruveta aktiivisesti tekemään pelejä [1].

Pelin kehittämisen aloitus on tärkeää. Pelin kehittämistä ei ole idean kirjoitus, kehitystiimin kokoaminen eikä edes grafiikan tai äänien teko. Se kaikki on pelikehityksen aloittamisen valmistelua. Hyvä muistisääntö on, että peliä pitää voida pelata. Jos sitä ei voi pelata, se ei ole peli. [2.]

Hyvä prototyyppi on jotain nopeaa ja halpaa, joka vastaa tiettyyn kysymykseen pelistä, oli se sitten tietty mekaniikka, toiminnallisuus tai se, onko peli-idea hauska. Avainsanat ovat nopea ja halpa. Prototyyppi ei ole pelin taso, ulkonäkö, eikä missään tapauksessa pelimoottori. Hyvä prototyyppi yrittää vastata kysymykseen. Kaikki kysymykset eivät kuitenkaan ole tasa-arvoisia. Kysymyksen pitää olla relevantti projektiin liittyen. Voiko asetusvalikko olla hyvännäköinen, ei ole tarpeeksi vaikea kysymys, jotta sille pitäisi tehdä oma prototyyppi. Voiko pieniä lentokoneita ohjata piirtämällä viivoja näyttöön, on paljon oleellisempi kysymys. Myös kysymys, onko peli hyvä, on melko vaikea, koska hyvä peli on niin laaja käsite. Se kannattaa muotoilla mieluummin muotoon: ”Miten saan tehtyä lentomekaniikasta kosketusnäytöllä vasteisen ja hyvältä tuntuvan?” Kysymyksiä kannattaa miettiä siltä kannalta, että pelaajat, jotka pelaavat, osaavat vastata niihin niin, että palautteesta on hyötyä. [3.]

### 2.2 Prototyypin tai pelin idea

Ensimmäisellä peli-idealla ei kannata aloittaa pelin tekoa. Yleensä tehdään niin, että kehitystiimi valitsee joukosta ideoita yhden hyvän tai jonkinlainen johtoryhmä päättää idean ja sitä lähdetään kehittämään, joko ensin prototyypiksi tai suoraan teknologiaa ja kirjoittamaan suunnitteludokumentteja. Jos yritys on onnekas tai peliä vie eteenpäin joku

erittäin hyvä peliohjaaja, pelistä saattaa tulla täydellinen. Tämä on kuitenkin harvinaista, koska yleensä ensimmäinen idea on vähän sinnepäin ja huonoimmassa tapauksessa kehitystiimi huomaa ensimmäisen tason tullessa valmiiksi vuoden päästä, että peli ei toimikaan oikein tai ole hauska. Tässä vaiheessa, koska projektiin on jo käytetty satoja tuhansia tai jopa miljoonia euroja, joudutaan tekemään päätös, perutaanko koko projekti vai yritetäänkö jo koossa oleva pelastaa. Kumpikaan tapa ei ole houkutteleva. [3.]

Prototyypin kehittäminen peliin, johon on jo sitoutunut, on turhaa. Kannattaa mieluummin tehdä useasta parhaasta ideasta prototyyppi, ennen kuin sitoutuu ideoista mihinkään tiettyyn. Hyvänä tapana voi olla esimerkiksi, että kun tulee uusi peli-idea mieleen, kirjoittaa sen muistiin, ja kun listassa on enemmän ideoita, sitä voi järjestellä niin, että parhaat ideat ovat ylimpänä. Tällöin kun on aika rakentaa prototyyppi, ideoita on jo valmiina ja voi aloittaa ensimmäisestä ja parhaasta ideasta. [3.]

Hyvällä prototyypillä on helppo selvittää, onko idea toimiva. Jos se ei ole, sen voi hävittää ja siirtyä seuraavaan. Jos on siinä ja siinä, onko prototyyppi hyvä vai keskinkertainen, voi päättää, jatkaako vielä hieman pidemmälle vai laittaako sen takaisin potentiaalisten ideoiden listalle. Myöhempi uusi inspiraatio saattaa viedä vanhaa prototyyppiä eteenpäin paremmaksi. Myös prototyypit, joista ei tule hyviä, saattavat synnyttää uusia parempia ideoita samoilla toiminnoilla tai muuten. [3.]

Jossain vaiheessa jokin prototyyppi on se oikea, josta lähdetään rakentamaan peliä. Jos onni potkaisee, se voi olla jo ensimmäinen idea, mutta silloinkin on suositeltavaa tehdä muutama prototyyppi. Jos sen jälkeenkin idea on vielä paras, nyt on myös tulevia pelejä varten prototyyppejä, joten aika ei ole mennyt hukkaan. [3.]

On olemassa kolme erilaista peli-ideatyyppiä: peli, jonka haluan tehdä, peli, jonka haluan olevan olemassa, ja peli, jonka tekemisessä olen hyvä. Tärkeää on kuitenkin, mikä tahansa se edellä mainituista olisikin, että se täyttää vaatimukset: ajan ja resurssit pelin toteuttamiseen.

Kuvassa 1 voidaan nähdä kaaviorakennelma pelinkehittäjän tilanteesta pelin aloitusvaiheessa. Peli-idean täytyy olla kaikista ideoista lupaavin.



Kuva 1. Potentiaalinen peli-idea. Kolme ideatyyppiä yhdistyvät yhdeksi hyväksi. [2.]

Peli, jonka haluan tehdä, on hauska kehittää, sen mekaniikalla on hauska leikkiä tai se on joku hahmo, joka olisi kiva animoida. Peli, jonka haluan olevan olemassa, on peli, jossa ollaan enemmän kiinnostuneita näyttävästä lopputuloksesta, eikä siitä, kuinka sinne päästään. Tällainen peli voi olla ilman rajoitteita kehitettävä miljoonabudjetin peli tai peli, jonka ajatus on hieno, mutta jota ei voi toteuttaa tarpeeksi hauskaksi. Peli, jonka tekemisessä olen hyvä, sopii kehittäjän osaamiseen ja kiinnostuksenkohteisiin. Se voi olla lempigenre tai jokin ominaisuus tai teema, joka on todella tuttu. [2.] Lupaavin idea on kaikkien edellä mainittujen ideoiden yhdistelmä: peli, jonka kehittämiseen on aikaa, resursseja ja oikeaa osaamista.

### 2.3 Yksin tai yhteistyönä

Hyvän yhteistyökumppanin löytäminen voi olla aikaa vievää, varsinkin kokemattomalle kehittäjälle. Ei riitä, että toinen osaa ohjelmoida ja toinen on hyvä piirtämään. Täytyy myös ottaa huomioon persoonallisuus, kokemus, ajoitus ja yhteiset kiinnostuksen kohteet. Projektia on mahdoton tehdä, jos toisella on eri prioriteetit tai vähemmän kiinnostusta projektia kohtaan. Tätä voi kokeilla tekemällä pienemmän projektin. Jos toinen lähtee projektista kuukausien tai vuosien päästä, se voi olla kohtalokas isku pelille.

Pelin kehittämisen voi myös aloittaa yksin. Internetistä löytyy paljon ilmaista musiikkia ja grafiikkaa ainakin siihen asti, että saa artistin tekemään grafiikan, jos ei itse tee. Artistit tulevat mielellään projektiin, joka on muuten valmis, mutta vain grafiikka puuttuu [2].

Ohjelmointia kannattaa kokeilla itse. Internetistä löytyy paljon oppaita ja videoita ja jopa visuaalisia ohjelmointiohjelmiä.

Pelinkehitykseen sisältyy useita rooleja. Monet näistä rooleista voi hoitaa yksi ihminen. On olemassa ihmisiä, jotka hoitavat nämä kaikki roolit, mutta se on hyvin työlästä ja harva pystyy siihen. Yksi tärkeä asia on tiimien yhteinen tiimihenki. Jos kaikki eivät tule täydellisesti toimeen toistensa kanssa pienessä yrityksessä, pelinkehitys ja yrityksen ylläpito on hyvin vaikeaa. Pienet yritykset eivät voi ottaa ketä tahansa tekemään pelejään kanssaan.

Ohjelmoija on yksi tärkeimpiä pelinkehityksen henkilöitä, koska hän tekee peli-ideasta toimivan. Ohjelmoija käyttää suurimman osan ajastansa koodin kirjoittamiseen ja työkalujen tekemiseen. Koska videopelit ovat digitaalinen viihdemuoto, ei ohjelmoijaa tarkoittaa yleensä ei peliä. Pienessä yrityksessä ohjelmoija hoitaa myös muun muassa ohjelmointivirheiden, koodiongelmien ja testien teon ja varmistaa, että peli on hiottu ja hyväksyttävä koodin kannalta. [4.] Ohjelmoijalla on iso rooli prototyypin kehityksessä, koska hän on yleensä ensimmäinen, joka fyysisesti koskee peliin.

Pienissä tiimeissä yleensä kaikki ovat mukana pelin suunnittelussa. Pelisuunnittelija voi olla ohjelmoija, artisti tai jopa kirjoittaja. Yleensä kannattaa kuitenkin olla johtava pelisuunnittelija, joka pitää suunnitelman uskollisena peli-idealle ja huolen, että uudet implementoitavat ideat lisäävät pelin kokemuksesta ja hauskuutta. [4.]

Ohjelmoijan ja pelisuunnittelijan lisäksi peli tarvitsee artisteja tekemään 2D- tai 3D-grafiikkaa, tekstuureja ja kaikkea pelin ulkoasuun liittyvää, äänisuunnittelijan ja äänituotannon sekä kirjoittajan tarinoita varten. Artisti toimii yleensä tiiviissä yhteistyössä suunnittelijan ja kirjoittajan kanssa, jotta peliin tuotettava grafiikka on oikeaa. Yleensä artisteja on enemmän kuin yksi, koska pelin visuaalisen puolen luominen on aikaa vievää. Äänet ovat pelin hyvin tärkeitä elementtejä pelikokemusta ajatellen, minkä takia siihen tarvitaan joku, joka tietää, mitä tekee. Äänisuunnittelija ja äänituottaja ovat vastuussa kaiken

musiikin ja ääniefektien luomisesta peliin. Äänet ovat tärkeä osa pelin atmosfääriä ja kokemusta. [4.]

Yrityksessä pitää olla myös tuottaja, joka pitää huolen, että peli menee eteenpäin. Tuottajan vastuulla ovat aikataulut, tiimihenki, tiimin tarpeet, työkalut, budjetti, markkinointi ja projektin managerointi ylipäättänsä. Tuottaja pitää huolen, että kaikki tekevät tehtävänsä, peli pysyy aikataulussa ja budjetti ei kasva liian isoksi. [4.]

Näitä kaikkia tehtäviä voi hoitaa yrityksen koosta riippuen yksi tai kaksi henkilöä, tai 50 tai 100 henkilöä. Kannattaakin ottaa huomioon, saako nämä kaikki roolit suoritettua annetulla henkilömäärällä ja kuka hoitaa minkäkin roolin.

Prototyyppiä kehitettäessä on kuitenkin oleellista, että prototyypin teossa on henkilö tai henkilöitä, jotka osaavat ohjelmoinnin ja suunnittelun. Muut roolit voi täyttää, kun on olemassa toimiva idea. Jos ei ole aikaisempaa kehitystiimiä, sen kokoaminen voi olla jopa helpompaa, koska on jo näyttää, että toimiva idea on olemassa.

## 2.4 Projektin koko ja tapahtumat rajana

Peliprojektin koon miettiminen on erittäin tärkeää. Kaikilla on paljon ideoita, miten tehdä uusi avoimen maailman jättiläispeli monilla ominaisuuksilla tai miten tehdä kaunein peli, mitä on olemassa. On kuitenkin tärkeää tietää, mihin itse tai ryhmän kanssa pystyy, ja se on tärkeä pitää mielessä jo suunnitteluvaiheessa. Hyvä idea on tehdä pieniä pelejä, jotka keskittyvät muutamaankin tiettyyn pelielementtiin ja jotka on mahdollista saada valmiiksi esimerkiksi kahdessa kuukaudessa [5].

Jos kyseessä on aloitteleva pelintekijä, pitää ajatella hyvin pienesti ja sen jälkeen vielä pienemmin. Jos aloittaa isoa projektia heti, sitä ei saa viimeistelyä, tai jos sen saa viimeistelyä, on polttanut itsensä loppuun. Yksi helpoimmista virheistä, mitä voi tehdä aloittaessaan pelien kehityksen, on antaa kunnianhimon ohjata tielle, jossa ei ole valmis kulkemaan. Kannattaa hidastaa, miettiä ja aloittaa yksinkertaisesta. Tämän takia prototyypin tekeminen on ratkaisevaa kaikille kehittäjille ja suunnittelijoille. [6.]

Kun tekee ensimmäistä peliä, törmää asioihin, joihin joka toinen kehittäjä törmää: liikaa ominaisuuksia ja liian iso projekti. Kehittäjät haluavat laittaa liikaa ominaisuuksia, mekaniikkoja ja sisältöä ensimmäiseen peliinsä. Kunnianhimo työntää helposti kehittäjää näihin ansoihin. Se on hyväksyttävää niin kauan, kuin se korjataan eli skaalataan alas. [7.]

Ensimmäiseen peliin pitää suodattaa se, mitä kehittäjä haluaisi nähdä yhteen mekaniikkaan, ja koettaa luoda se toimivaksi. Se pitää myös muovata yksinkertaiseksi mekaniikaksi, esimerkiksi ”etene pelin tarinan läpi valitsemalla vaihtoehtoista tekstivalikossa” tai ”älä päästä esinettä tippumaan”. Pitää ajatella enemmän Pong-peliä kuin Call of Duty-peliä. Ensimmäinen peli tuskin on seuraava iso juttu, mutta se tulee olemaan vähintäänkin prototyyppi ja jotain, mitä tiimi voi alkaa jatkokehittää kokonaiseksi peliksi. On tiettyjä piirteitä, joita kannattaa välttää ensimmäisessä projektissa, jos niistä ei ole paljoa kokemusta: moninpeliominaisuudet, verkkotoiminnot ja kolmiulotteisen pelin tekeminen. [7.]

Kilpailuja, messuja ja tapahtumia voi käyttää aikataulun takarajana. Kilpailujen takarajoihin ehtiminen vaatii usein tekemään päätöksiä pelin kehityksestä tai suunnasta. Se pakottaa myös katsomaan aikataulua paljon tarkemmin. Kilpailuihin on hyvä liittyä, koska takarajat ovat hyvin todellisia, mutta niin ovat myös palkinnot ja huomio. [2.] Tapahtumista voi myös löytää uusia suhteita, kuten kumppaneita, rahoittajia, median edustajia ja julkaisijoita.

### **3 Peliprototyypin kehitys ja pelinkehitystyökalut**

#### **3.1 Pelimoottori ja muut työkalut**

Projektin alussa kannattaa aina miettiä, mitä työkaluja tarvitsee. Esimerkiksi varsinkin alussa oma pelimoottori tuskin on tarpeellinen ja se saattaa viedä kehityksestä aivan liikaa aikaa: peliä ei ole edes vielä syntynyt, vaikka moottori olisikin valmis. Pelimoottori on arkkitehtuuri, jota kehittäjät käyttävät pelin pyörittämiseen. [10.]

Yleensä pelimoottori tarjoaa seuraavat ominaisuudet:

- fysiikan

- peliohjaimen näppäinten syötön
- grafiikan renderöinnin
- ohjelmoinnin
- törmäyksen havaitsemisen
- tekoälyn
- muita toimintoja ilman ylimääräistä ohjelmointia.

Pelimoottorit ovat uudelleen käytettäviä komponentteja, jotka tarjoavat kehittäjälle puitteet pelinkehittämiseen. Useat isot kehittäjät kehittävät omat moottorinsa. [10.] Pelimoottorit vaativat osaavia kehittäjiä ja kokemusta, koska ne ovat helposti muokattavissa ja mahdollisuudet ovat rajattomat. Ne kuitenkin säästävät paljon aikaa tarjoamalla valmiita kirjastoja, efektejä ja työkaluja, jotta kaikkea ei tarvitse tehdä tyhjästä. Vaikka ne ovat hyvin kehittyneitä, on suhteellisen helppoa opetella käyttämään niitä ajan kanssa, koska verkko on täynnä dokumentaatioita, ohjeita ja oppaita. [11.]

Pelimoottoreita, jopa ilmaisia, on jo saatavilla, esimerkiksi Unity, Unreal Engine, Game Maker ja CryEngine.

Unity, unity3d.com, on yksi suosituimmista pelimoottoreista, koska se on helppo oppia, se on erinomaisen toiminnallinen, sillä saa tehtyä korkealuokkaista sisältöä ja sillä voi tehdä käytännössä minkälaisen pelin tahansa [10]. Unity on myös käytetyin pelimoottori mobiilipeleissä. Unityllä voi tehdä 2D- tai 3D-pelejä, siinä on välitön pelitila, jotta peliä voit testata nopeasti ja hyvin voimakas animaatiojärjestelmä. Unity on myös virtuaalito-dellisuuspelien kehityksen eturintamassa. [11.] Unity Personal Edition on ilmainen, se sisältää kaikki Unityn ominaisuudet ja sillä voi julkaista kaupallisia pelejä.

Unreal Engine, www.unrealengine.com, on yksi vahvimmista ja käytetyimmistä pelimoottoreista kautta maailman. Alkuperäinen versio julkaistiin vuonna 1998, ja 17 vuotta myöhemmin sitä käytetään vieläkin markkinoiden suurimpien pelien tekoon. Unrealin iso vahvuus on, että sitä pystyy muokkaamaan niin paljon, että peleistä saa tehtyä todella uniikkeja kokemuksia. Viimeisimmän version, Unreal Engine 4, sanotaan olevan kaikista helpoin käyttää ammattilaisten käsissä. [10.] Guinness World Records nimesi Unreal Enginen menestyneimmäksi pelimoottoriksi. Unreal Engine on kuitenkin vaativampi kuin esimerkiksi Unity, minkä takia sitä suositellaankin kehittyneemmille pelin kehittäjille, jotka

tekevät isoja pelejä. Unreal Engine on ilmainen käyttää, mutta siitä joutuu maksamaan käyttökorvausta noin viisi prosenttia ensimmäisestä ansaitusta 3 000 dollarista neljännesvuodessa. [11.]

GameMakerista, [www.yoyogames.com/gamemaker](http://www.yoyogames.com/gamemaker), on tullut erittäin käytetty, koska toisin kuin moni muu pelimoottori, se ei vaadi ohjelmointiosaamista. Ohjelmoinnin sijaan kehittäjät voivat ”osoittaa ja klikata” luodakseen pelejä nopeammin ja helpommin kirjoittamatta natiivia ohjelmointikieltä. GameMakerilla on tehty menestyspelejä, esimerkiksi Spelunky ja Hotline Miami. Ongelma GameMakerissa ja muissa vastaavissa moottoreissa on se, että pelinkehittäminen on rajoittuneempaa. GameMakerista on ilmainen versio, mutta jos kehittäjä haluaa kaiken hyödyn, täytyy ostaa Professional- tai Master Collection -versiot. [10.]

CryEngine, [www.crytek.com/cryengine](http://www.crytek.com/cryengine), on ilmainen käyttää. Se sisältää koko lähdekoodin ja kaikki moottorin ominaisuudet ilman tarvetta maksaa lisenssimaksuja, käyttökorvauksia tai muita piilotettuja kuluja. CryEngine on visuaalisesti hämmästyttävä moottori. Grafiikka on kaunista ja hahmot eläviä. CryEngineä voi käyttää Xbox One-, PlayStation 4-, Windows PC- ja Oculus Rift -kehitykseen. Sen lisäksi CryEngine tarjoaa paljon ilmaisia oppaita ja ohjeita kehittäjille. Koska pelimoottori on niin voimakas, on oppimiskäyrä korkeampi kuin muissa mainituissa moottoreissa. [10.]

Grafiikkaa löytyy verkosta paljon erihintaista ja erilaista. Grafiikkaa varten voi myös etsiä freelancerin, joka tekee grafiikkaa palkkiota vastaan, tai grafiikan voi tehdä itse. Verkossa on paljon oppaita ja videoita ohjelmien käytöstä ja grafiikan teosta aina 3D-mallintamisesta piirtämiseen asti. Työkaluista ilmaisina esimerkkeinä ovat 3D-mallinnustyökalu Blender, [www.blender.org](http://www.blender.org), ja kuvankäsittely- ja piirto-ohjelma Gimp, [www.gimp.org](http://www.gimp.org).

Äänten tekeminen on oma taiteenlajinsa, ja sen takia on suositeltavaa ostaa ääniä verkosta. Niitä myydään paljon erilaisia monessa paikassa. Voi myös palkata äänistudion tekemään persoonalliset äänet ja musiikin. Ääniä voi myös generoida ohjelmilla kuten, esimerkiksi sfxr:llä, joka on ilmainen.

### 3.2 Nopea mielentila prototyypin kehityksessä

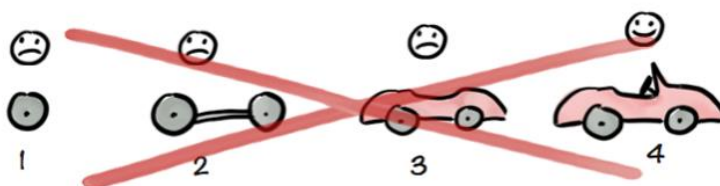
Nopea prototyyppi on yleensä nopea, karkea ja kokeilu: nopea, koska se tehdään yleensä nopeasti lyhyessä ajassa, karkea, koska sitä ei kuulu hioa kehityksen aikana, vaan saada vain toimivaksi, ja kokeilu, koska sillä yleensä kokeillaan jotain uutta ominaisuutta tai ideaa [12].

Jos kuitenkin mietitään hieman syvemmin, nopea prototyyppi onkin minimi, elinkykyinen ja vuorovaikuttava. Minimi tarkoittaa, että prototyyppiä ei tehdä vain nopeasti, vaan prototyypissä pitää olla vähimmillään kaikki, mitä se vaatii osoittaakseen tarkoituksensa. Elinkykyisellä taas haetaan sitä, että prototyyppi on koko ajan toimivassa tilassa ja sitä pystyy kehittämään eteenpäin. Lopuksi prototyypin pitää olla vuorovaikuttava, jotta lopukäyttäjät pystyvät käyttämään sitä ja sitä kautta saadaan kokeiltua prototyyppiä. [12.]

Nopea prototyypin rakentaminen on suosittu tapa testata ideoita ja ominaisuuksia pienissä pelistudioissa ja yksin pelejä kehittäville. Nopeaa prototyypin kehitystä käytetään myös Game Jameissa, joissa yleensä tehdään peli muutaman vuorokauden aikana.

Kuvassa 2 on hyvä esimerkki, kuinka prototyypin pitäisi edetä. Prototyypin on syytä olla toiminnassa ja pelattavissa koko ajan, niin kuin alemmassa kuvasarjassa on kuvattu.

**Not like this....**



**Like this!**



Kuva 2. Prototyypin kehittäminen askeleittain. Yläpuolella vasta lopussa on toimiva prototyyppi. Alapuolella joka askeleella lisätään toimivaa toiminnallisuutta. [12.]

Kehittäjän on hyvä muistaa, että epäonnistuminen kuuluu asiaan ja on jopa hyvin yleistä, kun nopeaa prototyyppiä kehitetään. Epäonnistumisen jälkeen tulee aina uusia projekteja, ja todennäköisesti jokaisesta epäonnistumisesta oppii jotain. [8, s. 1.] Prototyypin kehittämisestä saa myös välitöntä mielihyvää, koska projektiin tulee niin nopeasti uusia ominaisuuksia ja se kasvaa vauhdilla [12].

Nopeaa prototyyppiä kehittäessä pitää päästä pois kehitys- tai puuhailutilasta [9]. Se antaa mahdollisuuden keskittyä olennaiseen. Nopeaa prototyyppiä tehdessä ei aina tarvita parasta ratkaisua vaan nopeinta. Myöhemmin pystyy palaamaan hiomaan koodia tai parantamaan ulkoasua. Nopein ratkaisu vie eteenpäin ilman odottelua, mikä on tärkeää lyhyen kehitysajan projektissa. Ydin pitää aina pitää mielessä ja mennä sitä kohti.

### 3.3 Prototyypin ydin

Prototyypin ydin voi olla

- mekaniikka
- teema tai aihe
- miljöö
- tunteellinen kokemus.

Idean ydin on löydettävä ja selkeytettävä irti muista kannattelevista elementeistä, koska nopeassa prototyypin kehittämisessä keskitytään ytimeen. Sen takia on tärkeä tietää tarkkaan, mikä se on, jotta kehittäjä ei eksy vääriin asioihin.

Ensimmäinen asia, josta kannattaa aloittaa, on lelun rakentaminen. Lelu voi olla ritsa, jolla ammutaan esimerkiksi lintuja, hahmo, jolla liikutaan labyrintissä, painovoima tai mitä tahansa muuta. Tämä lelu on siis pelin ydin.

Tässä vaiheessa prototyypissä ei ole voittoa tai häviötä, vaan vain hauska juttu, jolla voi leikkiä. Mikä parasta, lelun rakentamisessa ei todennäköisesti mene edes koko päivää.

### 3.4 Kehityksen ajankäyttö ja riskien ottaminen

Kehitysaika nopean prototyypin kehityksessä on todella lyhyt, noin 1–2 viikkoa. Sen takia on tärkeää, että aikaa käytetään oikeiden asioiden tekemiseen. Esimerkiksi on turha asetella peliin puita tai hioa tasoja, jos niillä ei ole mitään tekemistä pelin ytimen kanssa. [8, s. 2.] Prototyypin on oltava nopea ja halpa. Mikä sitten on nopea? Nopeus riippuu paljon projektin pituudesta. Kahdessa kuukaudessa tehtävän mobiilipelin prototyyppi on paljon pienempi kuin kolmen vuoden konsolipelin. [3.] Asiaa on kuitenkin helpompi lähestyä, kun tietää tärkeimmät kysymykset ja saa itsenä oikeaan mielentilaan. Budjetti ja raha on myös suuri määräävä tekijä, koska prototyypin on oltava halpa siltä varalta, että se ei toimi, jotta rahaa ei mene hukkaan liikaa.

Yhden päivän prototyyppi on hyvä aika pienemmille kehittäjille tai tiimeille, varsinkin jos teknologia on jo olemassa. Silloin voi keskittyä pelkästään oleellisten ominaisuuksien kokeiluun tai kysymyksiin vastaamiseen. Jos yksi päivä ei kuitenkaan riitä, prototyypistä voi irrottautua ja miettiä, miksi tarvitaan lisää aikaa, keskityttiinkö kehityksessä vain ytimeen ja oleellisen vai onko pelin idea vielä epäselvä. [3.]

Kannattaa myös miettiä ohjelmoinnin ulkopuolelle. Prototyypin tekemiselle saattaa olla nopeampi vaihtoehto kuin ohjelmointi. Esimerkkeinä tästä toimii olemassa olevan pelin muokkaaminen tai jopa paperiprototyyppi, eli paperista tehty simulaatio. Ei kannata ajatella, että prototyyppi on pakko ohjelmoida, jos on nopeampiakin vaihtoehtoja. [3.]

Lyhyiden kehitysjaksojen noudattaminen on tärkeää, koska mitä enemmän on aikaa kehittää, sitä suurempi on riski, että kehittäjä poikkeaa sivuraiteille kehittämään tai korjaamaan jotain prototyypin ytimen kannalta merkityksetöntä asiaa. Pelattavuusidea voi testata tehokkaasti alle viikossa. Turhien teknisten riskien ottamista kannattaa välttää. Luovia riskejä sen sijaan on hyvä ottaa. Teknisiä riskejä kannattaa ottaa vain, jos prototyyppi on tekniikan kokeilu.

Prototyypeissä kannattaa muutenkin välttää monimutkaisia tietorakenteita ja algoritmeja, asioiden liian hätäistä tekemistä ja teknisiä toimia, jotka eivät paranna pelattavuutta. Prototyypin epäonnistumisen ei pitäisi johtua teknisestä ongelmasta, vaan prototyypin tylsyydestä.

Hyvä insinööri ei välttämättä ole hyvä prototyypin kehittäjä. Oikeat tai uudelleenkäytettävät ratkaisut eivät aina ole se, mitä haetaan. Kannattaa edetä prototyypin kehityksessä niin, että jokaista ongelmaa kohden on kourallinen ratkaisuja ja nopein ratkaisu voittaa. Loppukäyttäjät ei koskaan näe kehittäjän insinööritaitoja, eikä häntä se edes kiinnosta. [8, s. 2.]

### 3.5 Jumiin jääminen ja sen välttäminen

Jos on mahdollista huijata, se kannattaa tehdä. Kannattaa mennä sieltä, missä aita on matalin. Ei pidä tehdä monimutkaisia asioita, jos ne voi tehdä helposti ilman, että kukaan huomaa. Esimerkiksi hirviö, joka reagoi pelaajaan, on paljon vakuuttavampi kuin hirviö, joka ei reagoi pelaajaan, vaikka molemmat hirviöt olisivat käytökseltään yhtä yksinkertaisia.

Oikea ratkaisu ei ole aina paras. On suositeltavaa keskittyä ratkaisuun, joka toimii helposti oikein, eikä siihen, mikä on teknisesti oikein. Taktinen huijaaminen säästää myös aikaa ja rahaa. [8, s. 2.]

Jumiin jäämistä pitää välttää. On monia asioita, jotka saattavat aiheuttaa jumiin jäämistä. Ohjelmointivirheet, vaikeasti implementoitava koodi, oudot ohjaimet, sisällön määrä ja tasojen suunnittelu ovat vain muutamia esimerkkejä. Jos kuitenkin tulee tilanne, jossa ollaan jumissa siitä pitää hankkiutua pois. Hyviä apukeinoja jatkamiseen ovat seuraavat: olla välittämästä vioista, jotka eivät vaikuta pelattavuuteen, leikata jumittamisen aiheuttamat kohdat pois ja näin ollen vähentää sisältöä ja piilottaa ongelmakohta pelaajalta. [9.]

Umpikujaan vievät ideat kannattaa opetella tunnistamaan ajoissa. Kaikkea ei myöskään kannata yrittää pelastaa. Pelkkä hieno ominaisuus ei riitä saamaan aikaan peliä. Prototyypin pariin voi aina palata, jos saa ajatuksen, miten siitä tulee pelattavampi. Ei pidä jäädä tuhlaamaan aikaansa ideaan, joka ei toimi. [8, s. 3.]

### 3.6 Prototyypin tavoite ja jälkitoimet

Prototyypin tavoite kannattaa määritellä hyvin. Ilman tavoitetta prototyyppi on vain lelu eikä peli. Tavoite voi olla lähes mikä vain. Se voi olla jonkin asian keräämistä annetussa ajassa tai liikkumista avaruudessa osumatta mihinkään pahaan.

Vaikeus on löytää prototyypille hyvä tavoite. Esimerkiksi aikaraja saattaa tuntua liian pakotetulta. Paras tavoite on synnynnäinen osa pelattavuutta, esimerkiksi tornin rakentelu peleissä: rakenna ylöspäin.

Tavoite pitää muistaa koko ajan. Se helpottaa oikeiden ratkaisujen tekemistä. Se myös vaikuttaa työskentelyyn ja työskentelytapaan. Tavoitteesta kannattaa myös muistuttaa itseään tasaisin väliajoin. Jos tehtävä ei vie lähemmäs tavoitetta, sitä ei pidä tehdä.

Prototyyppi kannattaa simuloida mielessä. Se helpottaa prototyypin tekemistä, koska kehittäjä tietää tarkkaan, mitä pitää tehdä eikä tuhlaa aikaa suunnittelupäätöksiin, jotka tapahtuvat yrittämällä ja erehtymällä. [8, s. 2.] Peliä voi miettiä käyttäjän kannalta. Miksi he nauttivat pelistä? Mitä tunteita he tuntevat? Mitä pelissä tapahtuu, jotta he tuntevat mitä tuntevat?

Kun prototyyppi on valmis, on hyvä todistaa itselleen, että ydinominaisuus on hauska. Näin on valmiimpi tekemään päätöksen jatkotoimienpiteistä. Prototyypin kehittämisen jälkeen on hyvää aikaa ruveta hiomaan peliä ja ydintä. Pelistä voi tehdä kauniin, koodia voi korjata ja parantaa ja pelin voi kehittää loppuun valmiiksi tuotteeksi.

On myös tilanteita, joissa saattaa todeta, että prototyyppi ei ole tarpeeksi hyvä kaupalliseksi peliksi, jolloin on mahdollista julkaista se maksuttomana prototyypinä verkossa. Verkosta löytyy monta eri paikkaa, johon kehittäjät kasvattavat portfoliotansa; itch.io on yksi hyvä esimerkki.

Prototyypin julkista näyttämistä ei pidä pelätä. Tämä saattaa olla monelle vaikeaa. On hankala näyttää maailmalle keskeneräistä peliä, koska ihmiset eivät välttämättä näe visiota eikä prototyyppi täyttä visiota kerrokaan. Ihmiset tuomitsevat, mitä näkevät, ja joskus aika brutaalisti. On tärkeää vain hyväksyä se ja ottaa palaute vastaan ja ajatella asiaa

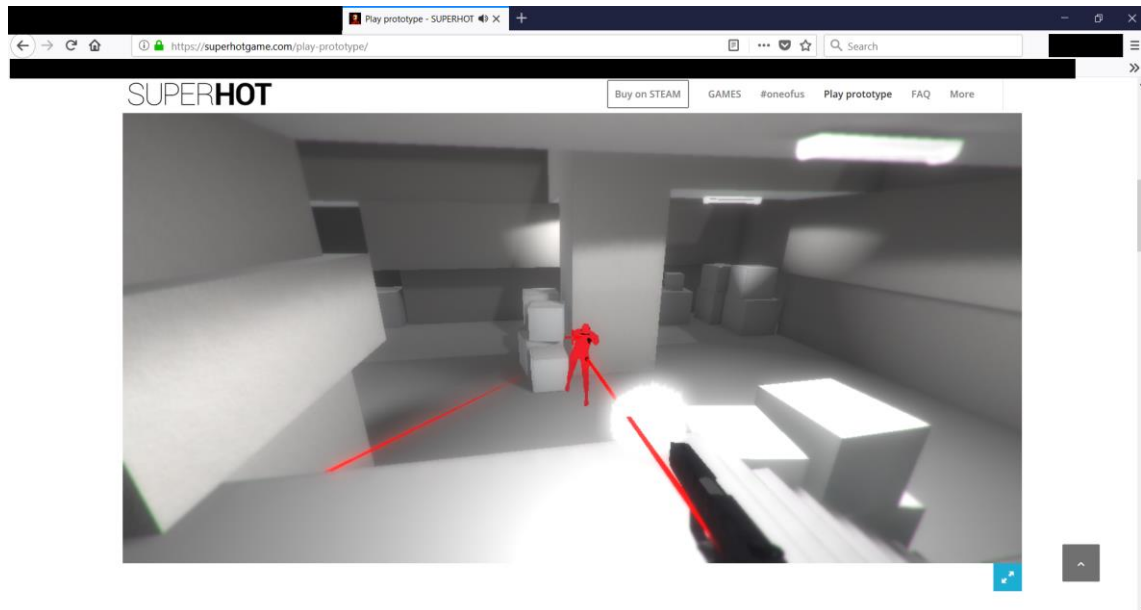
siltä kannalta, että hyvää peliä on vaikea tehdä, jos siitä ei saa rehellistä negatiivista palautetta. Tämä palaute on parempi saada ennen pelin julkaisua kuin sen jälkeen. [13.]

Kun prototyyppi on valmis ja todettu hyväksi, on monta vaihtoehtoa, miten edetä. Täydelliseen julkaisuun on vielä matkaa, vaikka julkaisijan hankkiminen prototyypille onkin yksi vaihtoehto. Muita vaihtoehtoja on esimerkiksi itch.io:ssa julkaiseminen. Itch.io on avoin markkinapaikka pelinkehittäjille, jotka kehittävät indiepelejä eli itsenäisesti pelejä. Se on alusta, joka antaa kenen tahansa myydä luomiaan pelejä siellä, ja kehittäjä päättää itse hinnan, hallinnoi myyntiä ja päättää sivujen ulkoasun. Iso ero muihin myyntipaikkoihin on, että ei tarvita tykkäyksiä, ääniä ja seuraajia, jotta saa pelin hyväksytyksi kaupapaikalle. [14.]

On myös paljon muita julkaisualustoja. Steam on kaikista tunnetuin ja isoin ja siellä on todella paljon pelejä. Steamilla on yli 100 miljoonaa käyttäjää. Steam-peli pitää julkaista palvelun nimeltä Steam Direct kautta ja liittymällä Valven yhteistyökumppaniksi. Tämä maksaa 100 dollaria tuotetta kohti. [15.]

Toinen hyvä vaihtoehto on Humble Bundle, jolla on noin 12 miljoonaa asiakasta. Pelin julkaiseminen siellä on ilmaista, ja Humble Bundlen oma tiimi käy läpi pelit, jotka sinne saapuvat. Humble Bundle on siitä hyvä, että siellä kehittäjä saa suuremman osan rahasta kuin Steamissa ja osa menee hyväntekeväisyyteen. [16.]

Kahdessa viimeksi mainitussa alustassa pelin pitää olla täysin valmis, jotta sivut hyväksyvät pelit tai jotta peli löytää yleisönsä.



Kuva 3. Superhot. Pelattava prototyyppi selaimessa. [17.]

Vaihtoehtoja on vielä tämänkin lisäksi. Esimerkiksi peli nimeltä Superhot aloitti tehokkaan markkinointikampanjan Game Jam -prototyypillään. Yksi sen tehokkaimmista eduista oli selaimessa pelattava prototyyppi selaimessa, joka näkyy kuvassa 3, ja helposti kuvailtava konsepti. Aika liikkuu, kun pelaaja liikkuu ja pelaajan pitää tappaa kaikki viholliset. [18.]

Julkinen raha ja avustukset ovat rahoitusvaihtoehto, joihin yrityksen kannattaa tutustua jo heti kehityksen alussa. Esimerkiksi Mediadeskin kehittelytuki peleille on yksi suurimmista julkisen rahan rahoituksista Euroopassa. Tukea myönnetään projekteille, joita kehitellään kansainvälisiä markkinoita varten. Peleille tarkoitettua kehitystukea voivat hakea itsenäiset eurooppalaiset pelituotantoyhtiöt, jotka ovat aiemmin tehneet vähintään yhden pelin, joka on ollut kaupallinen. Kehitettävä peli voi olla suunnattu mille tahansa alustalle, mutta sen pitää olla kaupallinen. Pelissä pitää olla narratiivista tarinankerrontaa. [19.]

## 4 Prototyyppien kehitys

### 4.1 Peli-idea ja pelimoottorin valinta

Ensimmäisenä askeleena kehitystä lähestytään pelimoottorin valinnalla. Pelimoottorin valinta oli tätä projektia tehdessä helppo, koska kehittäjällä oli Unitystä eniten kokemusta, jolloin aikaa ei tuhlaannu uusien työskentelymenetelmien opetteluun. Hyviä vaihtoehtoja ovat myös Unreal Engine, joka on hieman hankalampi käyttää, ja Game Maker, jolla voi tehdä vain 2D-pelejä. Unity tarjoaa myös ison yhteisön, joka osaa auttaa monessa tapauksessa ja valmiita ratkaisuja ongelmiin.

Idean keksiminen peliin on huomattavasti vaikeampaa. Tai idean keksiminen on helppoa, mutta toteutettavan idean keksiminen on vaikeaa. Ideaa kuitenkin lähestyttiin enemmänkin peligenren kannalta. Pelilajityyppien tekemisessä on isojakin vaikeusaste-eroja. Kuvassa 4 ne on listattu vaikeusjärjestyksessä.

Game genres in order of difficulty to produce a minimum viable product, from simplest to most difficult:

- |                               |                      |
|-------------------------------|----------------------|
| 1. Racing Game                | 7. FPS               |
| 2. Top Down Shooter           | 8. JRPG              |
| 3. 2D Platformer              | 9. Fighting Game     |
| 4. Color Matching Puzzle Game | 10. Action Adventure |
| 5. 2D Puzzle Platformer       | 11. Western RPG      |
| 6. 3D Platformer              | 12. RTS              |

Kuva 4. Pelilajityyppien tekemisen vaikeustaso. Lajityypit luetteloitu toteutuksen vaikeuden mukaan helpoin ensin. [20.]

Kuvasta 4 huomaa, että helpoimpana ovat ajopelien lisäksi 2D-pelit. Tämän takia ensimmäiset prototyypit tehtiin 2D-tasoloikkapelistä ja pelistä, jonka kuvakulma on ylhäältäpäin. Vaikka 2D-peleissä pysty hyppimään, loikkimaan, ampumaan vihollisia ja

puhumaan muille hahmoille, inspiraatiota ideaan ei tullut. Prototyypit tehtiin molemmista peleistä vain muutamassa päivässä, joten ne oli helppo jättää pois. Mikään tekeminen ei kuitenkaan mennyt hukkaan, koska nyt yrityksellä on valmis tasoloikka- ja ylhäältäpäin kuvattu ammutapeli pohja.

Tehtyjä prototyyppejä on olemassa monta, niin kuin esimerkiksi kuvan 5 mukainen ylhäältäpäin kuvattu ammutapeli, mutta seuraavaksi keskitytään kahden muun prototyypin kehitykseen ja kaikkiin vaiheisiin. Molemmista prototyypeistä käydään siis kehityksen samat vaiheet läpi ja vain toinen niistä pääsee jatkoon.



Kuva 5. Ylhäältäpäin kuvattu ammutapeli. Näyttökaappaus toisesta prototyypistä, joka oli ylhäältä päin kuvattu ammutapeli.

Kun mikään klassisista 2D-peleistä ei saanut innostumaan, kokeillaan toista lähestymistapaa. Ensin mietitään millainen peli olisi peli, jonka haluaa tehdä. Sen jälkeen pelistä mietitään pääominaisuudet, jota ilman peli ei toimisi. Koska peli nimeltä Doom, joka julkaistiin vuonna 2016, oli hyvin suosittu ensimmäisen persoonan ammutapeli ja koska kehitystiimillä oli jo aikaisempaa kokemusta 3D-maailmasta ja sen käsittelystä pelilajityypiksi, valittiin ensimmäisen persoonan ammutapeli. [21.]

Kun pelilajityyppi on valittu, sitä voi kannattaa vielä verrata kuvan 4 listaan. Ensimmäisen persoonan ammuntopeli on sijalla seitsemän "FPS". Jos listasta poistaa kaikki 2D-pelit, se sijoittuu sijalle kolme, joka on tarpeeksi hyvä sijoitus. Kutsutaan tätä ensimmäistä prototyyppiä nimellä Projekti Mojito, joka on siis ensimmäisen persoonan ammuntopeli.

Toinen prototyyppi-projekti syntyi niin, että mietittiin kaksi peruspeliä, jotka yhdistettäisiin. 2D-tasohyppelypelaaja oli jo valmis, ja se on ehkä yksi klassisimmista muodoista. Se valikoitui toiseksi, ja toiseksi valikoitui jalkapallo, koska sitä pelataan ympäri maailmaa. Niinpä toinen projekti on sivustapäin kuvakulman jalkapallopeli. Jotta siihen saataisiin vauhtia ja vaarallisia tilanteita, jalkapalloa ei potkita vaan sitä ammutaan. Ja koska jalkapalloon tarvitaan enemmän kuin yksi pelaaja tulisi tästä pelistä kahdesta neljään pelaajan pelattava peli. Tätä prototyyppiä kutsutaan nimellä Shootball.

Tässä vaiheessa Project Mojiton ideointi alkaa. Ensimmäisen persoonan pelejä on paljon, minkä takia peli tarvitsee jotain uniikkia. Tässä otetaan vaikutteita muista peleistä. Hotline Miami-nimisessä pelissä pelaaja pystyy heittämään tyhjiä ja käytettyjä aseita vihollisia päin, toisin kuin monessa ensimmäisen persoonan peleissä, joten se ominaisuus otetaan prototyyppiin. Kehittäjillä on hyviä vitsejä ja huumoria muissa peleistä, joten huumori ja komedia tulee olemaan osa peliä.

Pelin tai prototyypin pääominaisuudet ovat siis

- vauhdikas taistelu
- mahdollisuus käyttää aseita ja esineitä muutenkin kuin ampumalla
- huumori ja komedia.

Shootballin suunnittelu alettiin miettimällä, miten ampuminen toimii, onko se vasemmalle ja oikealle vai voiko ampumasuunta olla mihin päin vain. Ampumisesta päätettiin tehdä radiaalinen, eli pelaaja voisi tähdätä eri suuntiin ohjaimen tällä. Myös jalkapallon perussäännöt, kuten aika, maalit, aloitukset ja muut kirjattiin. Pallo pitää saada pomppimaan ja pelaaja hyppimään ympäri kenttää. Äänimaailma tulee olemaan iso osa peliä, koska peliin saa näin yleisö, selostajan, musiikkia ja muuta, mikä tuo juhlatunnelmaa. Myös graafinen ulkoasu, sijainti ja muu piti suunnitella. Pelistä pitää tulla hauska yhdessä pelattava kokemus.

Kun idea on olemassa, pelimoottori ja muut työkalut on valittu ja tehtävälisterat laadittu, kehitys voi alkaa. Huomioitavaa on, että tähän käytettiin aikaa alle viikko.

#### 4.2 Prototyypin ensimmäiset askeleet

Projekti Mojito on ensimmäisen persoonan ammuntopeli, joten pelin kehitys on hyvä aloittaa siitä, ensimmäisen persoonan ampumisesta. Jos pelaajan kameraa ei jaksata tehdä alusta asti itse, Unity tarjoaa käytettäväksi hyvän pohjan, jonka päälle rakentaa muu toiminnallisuus.

Tässä vaiheessa ei ole vielä tarpeellista miettiä grafiikkaa tai sitä, miltä peli tulee näyttämään lopuksi. Pääominaisuudet ovat ratkaisevia. On graafisia asioita, joita voi Unityssä tehdä heti alkuun, esimerkiksi muuttaa laatu-, grafiikka- ja pelaaja-asetukset kohdalleen ja ottaa jälkikäsitteleytehosteet käyttöön. Voi käyttää tarvittavina malleina, ääninä ja tekstuureina joko Unityn tarjoamia ratkaisuja tai etsiä verkosta. Vaikka ne eivät olekaan lopullisia, on artistin helpompi korvata vanhat, kuin tehdä kaikki alusta asti.

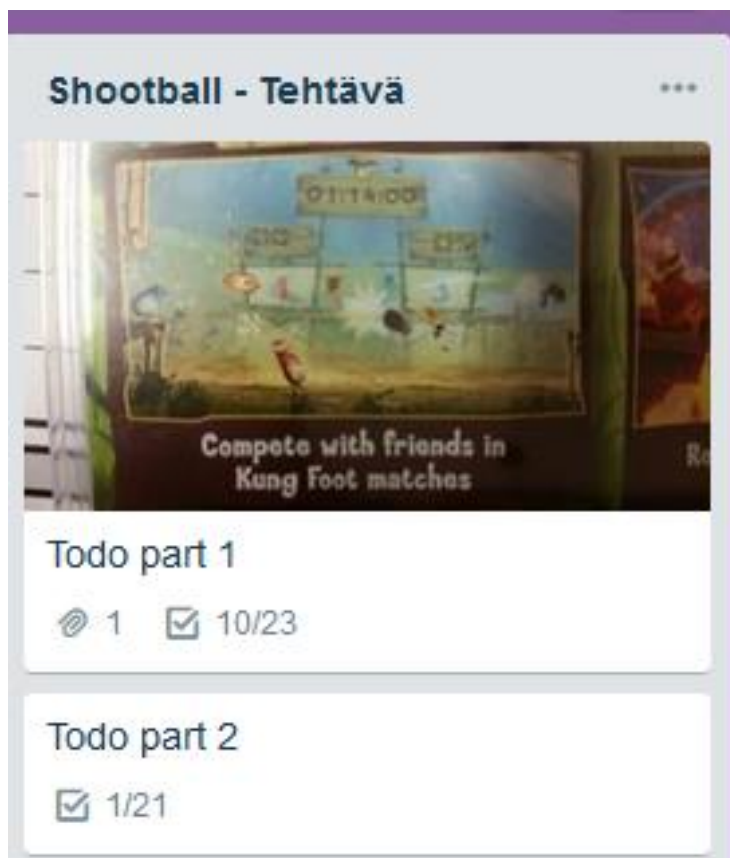
Kehitys alkaa siitä, että tehdään ensimmäinen taso, johon laitetaan jotain, missä kävellä, ja seinät muistuttamaan esimerkiksi huonetta. Sen jälkeen voi säätää valoasetukset kuntoon ja aloittaa kehityksen. Kaikki mallit, jotka kuvassa 6 näkyvät, on koottu alle tunnissa eri lähteistä. Ympäristö on koottu Unityn omista kuutioista, ase ja käsivarsi on ladattu Unityn omilta sivuilta, ja hahmot ovat koottu nopeasti Adobe Fusella, jolla pystyy tekemään ihmishahmon ilman mallinnusta muutamassa minuutissa.



Kuva 6. Mojito-peli. Näyttökaappaus, jossa näkyy väliaikaisia malleja ja testitaso. Kaikki kuvassa näkyvät mallit ja tekstuurit koottiin alle tunnissa eri lähteistä.

Shootballi-pelin kehitys aloitettiin hyvin perusasioista, kuten että pelaaja pystyy liikkumaan ja ampumaan palloa ja että se on hauskaa. Grafiikkapuoli tehtiin Unityn omilla oletuspaketeilla, joista saatiin myös pohja liikkumiselle. Ampumisen joutui tekemään itse. Äänet etsittiin verkosta aluksi, mutta toinen kehittäjä, joka tuli projektiin avuksi, rupesi nopeasti työstämään peliin oikeanlaista äänimaailmaa.

Kuitenkin tässä vaiheessa projektinhallintatyökalu Trellossa [22], kuten kuvasta 7 näkee, on kaksi eri tehtävälistaa osa yksi ja osa kaksi, joista ensimmäisessä osassa on paljon perusominaisuuksia, joita käsitellään seuraavassa luvussa, ja osassa kaksi on kehittyneempää ja yksityiskohtaisempaa kehitettävää, ominaisuuksia ja asioita. Ensimmäiseen osaan lisättiin asiaa sitä mukaa, kuin tuli vastaan kohta, joka tarvitsi hiomista tai asiaa piti muokata tai tehdä paremmin.



Kuva 7. Tehtävälista. Tehtävälista jaettiin kahteen osaan, jotta ensin tehtäisiin tärkeämmät asiat, jotka prototyyppi yksinkertaisimmillaan vaatii.

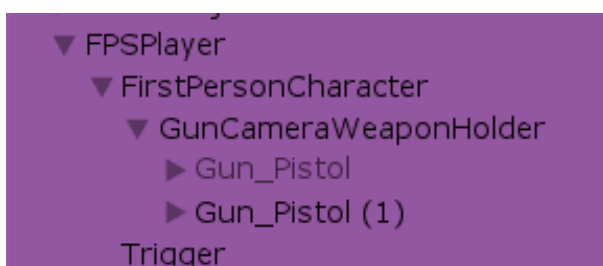
Kehityksen apuna toimivat tehtävälista, koska yksin tehdessä saattaa tekemisen määrä tuntua isolta. Silloin on hyvä järjestellä asiat tärkeysjärjestykseen ja keskittyä vain yhteen asiaan kerrallaan. Sovellus ja verkkosivu nimeltä Trello toimii hyvin apuna. Kun ohjelmointi, virheet ja toimimattomuus alkavat väsyttää voi, kehityksestä pitää tauon ja tehdä esimerkiksi grafiikkaa, estetiikkaa ja pelin ulkoasutekstejä tai hahmotella ja suunnitella vihollisia. On kuitenkin tärkeä muistaa, että vihollisten suunnittelu ei ole pelin kehitystä.

#### 4.3 Prototyyppien perusominaisuudet

Kun Mojiton kehitys on saatu alkuun, alkaa ensimmäisen ja tärkeimmän ominaisuuden kehitys ammutapelissä, eli ampuminen. Itse ampumismekaniikan tekeminen ei ole vaikeaa ja se toteutuukin yhdessä päivässä. Vaikea osuus on saada ampuminen tuntumaan hyvältä. Siihen onneksi on oppaita ja blogeja, ja mikä parasta, muita pelejä.

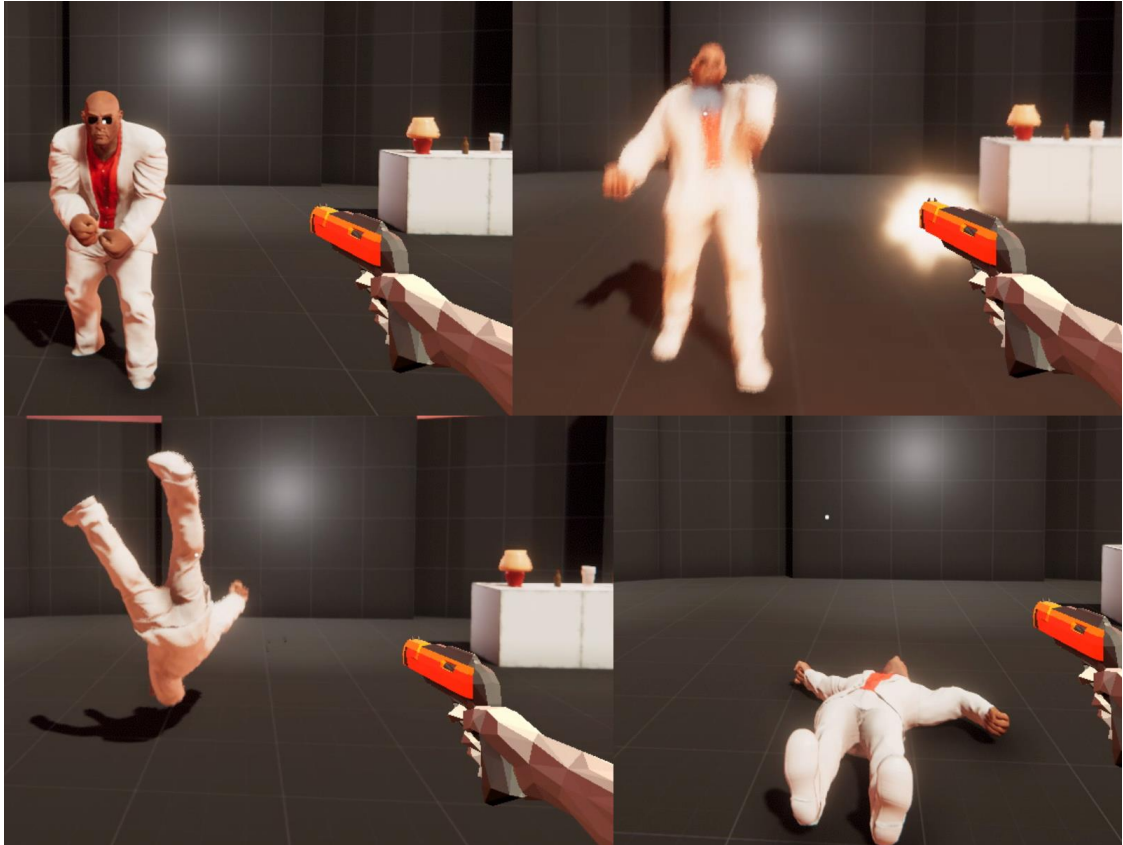
Hyvä ajatus on katsoa kauppapaikoilta, mitkä saman lajityypin pelit ovat suosiossa, selvittää miksi ja pelata niitä. Viitepelien löydyttyä voi lukea myös pelistä tehtyjä artikkeleita. Gamesradar, joka on pelialan mediataho, kirjoitti artikkelin, jossa pohdittiin miksi Doom on vielä vuonna 2017 paras konsoliammuntapeli, ja haastateltiin Doom-pelin kehittäjiä [23]. Kun perustuntuma on hyvä, siirrytään eteenpäin. Kehityksen loppupuolella on hyvä hioa kaikki täydelliseksi, mutta tässä vaiheessa se ei vielä ole tarpeellista.

Kuvassa 8 näkyy, kuinka pelaajaobjekti on järjestelty hierarkiassa, kun kaksi asetta on poimittu. GunCameraWeaponHolder-objekti hallinnoi aseita ja aseiden vaihtoa, kun taas itse aseobjekteissa on aseiden toiminnallisuus.



Kuva 8. Pelaaja-objektin hierarkia. Pelaaja-objekti näyttää tältä, kun kaksi asetta on poimittu.

Kun ampuminen toimii, laitetaan ympäristö ja viholliset reagoimaan siihen. Toteutus aloitetaan siitä, että valkoinen kuutio muuttuu punaiseksi ja siihen vaikuttavat fysiikat, kun pelaaja osuu siihen. Kehityksen edettyä viholliset saavat omat reaktionsa, ja dramaattisuuden lisäämiseksi vihollisista lentää verta ja he muuttuvat räsynukeksi kuollessaan, niin kuin kuvassa 9 näkyy.

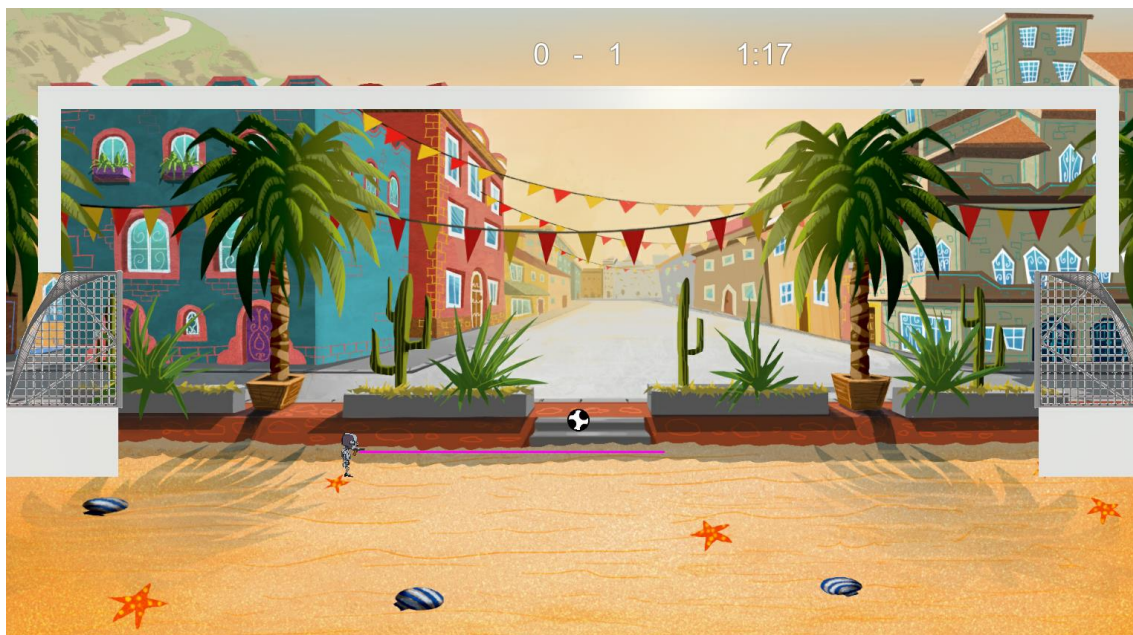


Kuva 9. Räsynukke. Dramaattisuuden lisäämiseksi viholliset muuttuvat räsynukeksi kuollessaan.

Ensimmäinen etappi on saavutettu. Pelaaja pystyy liikkumaan, hyppimään ja ampumaan vihollisia. Koossa on jokaisen ensimmäisen persoonan ammuntopelin ydintoiminta.

Shootballin kehitys aloitettiin pelaajan liikkumisesta, ampumisesta, tähtäämisestä ja pallon osumisesta. Tämän jälkeen tehtiin jalkapallopeliosuus peliin eli kaksi maalia, joihin pelaaja voi ampua pallon ja joista oikea joukkue saa oikean määrän pisteitä. Myös ampumisäänet, hurrausäänet ja taustamusiikki laitettiin peliin luomaan tunnelmaa. Googlen kuvahaun kautta pelille löytyi tausta ja maalit. Oli myös tärkeää saada Xbox-ohjain toimimaan pelaajalla, koska loppujen lopuksi peliä tulnaisiin pelaamaan ohjaimella. Piti siis päästä kokeilemaan, miltä pelaaminen tuntuu ohjaimen kanssa.

Tässä vaiheessa yksi pelaa pystyy liikkumaan ja ampumaan palloa maaliin ja hän saa pisteitä. Niin kuin kuvasta 10 näkyy, pelin ulkoasu on hyvin värikäs.



Kuva 10. Shootball. Sivustapäin jalkapalloa aseilla. Googlen kuvahaun kautta löytyi kaunis tausta prototyypin luomaan tunnelmaa [24].

Tausta muuttuu vielä, kun itse peli valmistuu, koska kehittäjällä ei ole oikeuksia kuvan käyttöön. Oikeuksien kanssa kannattaa olla tarkkana, jos prototyyppiä menee näyttämään kaupallisiin tapahtumiin tai haluaa myydä peliä.

#### 4.4 Prototyyppien lisäominaisuudet

Mojiton, eli ensimmäisen persoonan pelin, perustoimintojen, eli ampumisen ja liikkumisen, ollessa valmiina, mietitään, mitä tehdään seuraavaksi. Tässä kannattaa keskittyä alkuperäisen idean ydinominaisuuksiin, eli vauhdikkaaseen taisteluun ja mahdollisuuksien heitellä ja poimia esineitä ja aseita. Jotta peli näyttää ja tuntuu hyvältä, se tarvitsee myös animaatioita, efektejä ja ampumiseen ampumisen tuntua.

Animaatioiden ja ampumaefektien, muun muassa kuvan 11 koodin, ollessa paikallaan aletaan tehdä tärähdystä, kun pelaaja ampuu aseella. Tämä tehdään niin, että joka kerta, kun pelaa painaa liipaisinta, kamera tärähtää ja luo voiman tunnetta ampumiseen. Tämä termi kulkee nimellä "screenshake", ja se on tärkeä osa immersion luomista.

```

1reference
void WeaponSwayer()
{
    //left and right
    float movementX = -Input.GetAxis("Mouse X") * amount;
    //when gameObject's Y transform changes
    float movementY = gameObject.transform.position.y * amount;
    //float movementY = -Input.GetAxis("Mouse Y") * amount;
    movementX = Mathf.Clamp(movementX, -maxAmount, maxAmount);
    movementY = Mathf.Clamp(movementY, -maxAmount, maxAmount);

    Vector3 finalPosition = new Vector3(movementX, movementY, movementZ);
    transform.localPosition = Vector3.Lerp(transform.localPosition, finalPosition + initialPosition, Time.deltaTime * smoothAmount);
    movementZ = 0f;
}

1reference
void WeaponWalking()
{
    if(Input.GetAxis("Horizontal") > 0 || Input.GetAxis("Horizontal") < 0 || Input.GetAxis("Vertical") > 0 || Input.GetAxis("Vertical") < 0)
    {
        anim.SetBool("isWalking", true);
    }
    else
    {
        anim.SetBool("isWalking", false);
    }
}

1reference
public void Recoil()
{
    movementZ = -1f * amountZ;
    movementZ = Mathf.Clamp(movementZ, -maxAmount, maxAmountZ);
}

```

Kuva 11. Aseen liike. Koodia, jolla hallitaan aseiden heilumista pelaajan liikkua ja rekyä.

Koska pelissä pitää pystyä nostamaan ja heittämään tavaroita ja taistelu on vauhdikasta, tämä pitää toteuttaa niin, että se on mahdollisimman helppoa ja jouhevaa. Harvassa pelissä on nykyään niin sanottu "dual wielding" eli pelaaja pystyy pitämään kahta asesta samaan aikaan molemmissa käsissä, mutta prototyypissä kokeillaan sitä. Molempiin käsiin pitää pystyä ottamaan ase tai esine, ja pelaajan on pystyttävä heittämään jompikumpi pois milloin vain. Kokeilun ja tekemisen jälkeen paras tulos tuli niin, että molemmilla käsillä on omat painikkeensa aseiden ottamiseen, jos käsi on tyhjä, ja heittämiseen, jos kädessä on jo ase. Vasen käsi tottelee q-näppäintä ja oikea käsi e-näppäintä. Viholliset myös reagoivat, jos pelaaja osuu heittäessään heihin. Aseiden poimiminen toimii niin, että pelaaja katsoo aseeseen ja painaa näppäintä poimiakseen sen. On mahdollista, että tämä muutetaan niin, että riittää, kun pelaaja seisoo aseiden päällä ja painaa oikeaa näppäintä vauhdikkuuden ylläpitämiseksi.

Kun prototyyppi on valmis ja muut kokeilevat sitä, on mahdollista, että tulee muutoksia, esimerkiksi niin, että tuliase on aina oikeassa kädessä ja sitä ei voi heittää ja vasemmassa kädessä voi joko pitää lyömäasetta, jolla voi lyödä ja jonka voi heittää, tai toista tuliastetta. Tämän takia kannattaa kokeilla mikä, toimii parhaiten mahdollisten asiakkaiden pelatessa.

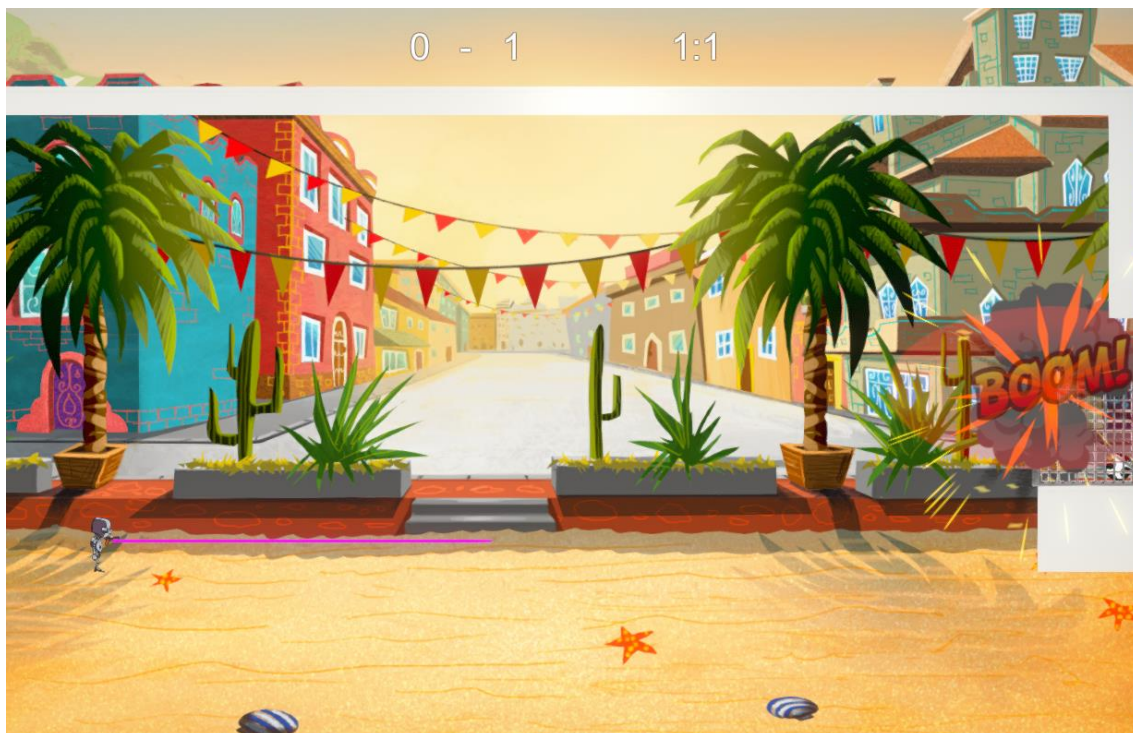
Kuvassa 12 on kuva tavoitetekstistä, joka kertoo pelaajalle, mitä tehdä seuraavaksi. Pelissä on myös animaatiot hyppimiselle, laskeutumiselle, kävelemiselle, ampumiselle ja vihollisille. Pää- ja taukovalikko ovat paikallaan ja toimivat. Pelaaja pystyy kiipeämään ylös tasanteen reunalta, eikä enää tarvitse vain hypätä päälle. Tämä ominaisuus auttaa tasojen suunnittelussa myöhemmin, kun pelaaja pystyy liikkumaan vauhdikkaasti pysty- ja vaakasuorassa. Toimiva tekstikenttä, joka näyttää seuraavan tehtävän tai tavoitteen, toimii myös perusominaisuuksiltaan, eli vaihtaa tekstiä, kun tietty asia tapahtuu.



Kuva 12. Tavoiteteksti. Tavoiteteksti vaihtuu aina, kun pelaaja suorittaa tavoitteen.

Koska Shootball, ammutajalkapallopele, on kahden pelaajan peli, on tässä vaiheessa lisättävä peliin kahden pelaajan yhteinen mittelo. Tämän eteni niin, että ensimmäinen pelaaja kopioitiin ja koodista vaihdettiin kaikki näppäimen painalluskäskyt niin, että niitä pystyi Unityssä muokkaamaan, jolloin molemmille pelaajille on yksi yhteinen skripti. Ongelmia tuli ohjainten kanssa, koska Unity käsittelee eri ohjaimia eri tavalla. Esimerkiksi Xbox-ohjaimen ja Playstation 4 -ohjaimen syötteet ovat aivan erilaiset, jolloin pelaaja, joka pelaa Playstation-ohjaimella, ampuu ja hyppii eri näppäimistä kuin Xbox -ohjaimella pelaava. Tähän etsittiin aluksi verkosta liitännäistä, joka olisi ongelman hoitanut, mutta moni liitännäinen oli joko liian monimutkainen käyttää tai tarkoitettu vain yksinpelien sisääntulojen uudelleenjärjestelyyn. Asia ratkaistiin itse muutamassa päivässä, minkä jälkeen ratkaisu olikin jopa nerokas: peli osaa nyt katsoa itse, mikä ohjain liitetään, ja sen mukaan käyttää oikeaa sisääntuloasetusta.

Myös itse peliin lisättiin paljon ominaisuuksia, kuten räjähdysisiä, hidastuksia ja ääniä kun pallo menee maaliin ja ajanotto jokaiselle ottelulle. Muun muassa maaliräjähdysten näkee kuvassa 13. Peli suorastaan heräsi eloon.



Kuva 13. Maaliräjähdys. Dramaattinen räjähdys tapahtuu, kun pallo menee maaliin. Myös aika hidastuu hetkeksi.

Kaksi pelaajaa pystyi nyt pelaamaan kahdella ohjaimella, ja äänet vaihtelivat sen mukaan, osuuko pallo maahan, maaliin, ylärimaan vai toiseen pelaajaan.

#### 4.5 Tasot, viholliset, aseet ja pelaaminen

Mojiton tasojen suunnittelu ja tekeminen ovat tärkeä osa pelin tekemistä, varsinkin kun kyseessä on vauhdikas ammuntopeli, koska pelaajan pitää pystyä pysymään vauhdissa ja lähestymään taistelutilanteita niin, että se on hauskaa ja monipuolista. Tässäkin apuna toimivat hyvin muut ensimmäisen persoonan ammuntopelit ja niistä kertovat dokumentit. Mojitossa keskityttiin siihen, että tasoissa on riittävästi korkeusvaihtelua, oikoteitä, kiertoteitä, kapeita ja leveitä kohtia ja että pelaajalla on mahdollisuus lähestyä vihollisia ja kenttää ylipäätänsä eri suunnista tai eri reittejä.

Myös yksityiskohdat pitää ottaa huomioon, muun muassa, että valoilla on helppo ohjata pelaaja oikeaan suuntaan ja ovet, huoneet ja alueet aukeavat vasta, kun kaikki viholliset on hoidettu pois kuvioista. Huomiota kiinnitetään myös siihen, että pelaajaa

kannustetaan pelaamaan aggressiivisesti eikä kyykkimään piilossa ja etenemään varovasti. Se toteutetaan niin, että läheltä tapahtuvat tapot näyttävät ja tuntuvat hienoilta ja vihollisen taposta saa lisää elämää tai voimaa pelaajalle, jolloin kuolemaa vältellään liikumalla nopeasti ja tappamalla paljon. Lyhyesti: jos pysyy paikallaan, kuolee.

Kenttiin tai tasoihin lisätään sen lisäksi vielä poimittavia elämä- tai terveyspaketteja, panoja aseisiin ja mahdollisia pelin muuttajia, esimerkiksi hetkellinen kuolemattomuus. Myös muita aseita ja vihollisia on erilaisia. Pelaaja tarvitsee myös tässä vaiheessa tavan kuolla eli elämämittarin, joka tyhjenee vihollisten hyökätessä.

Vihollisten tekemisessä on monia vaiheita. Vihollisille pitää tehdä tekoäly. Tähän tuodaan mielenkiintoa ajattelemalla taistelua ongelmanratkontana. Viholliset eivät vain ilmesty tyhjästä ja ammu pelaajaa, vaan pelaaja voi oppia vihollisen liikkeitä ja käyttää oikeita työkaluja vihollisen kukistamiseen. Vihollisen voi siis tappaa nopeammin, jos teittää, mitä tekee. Monessa pelissä viholliset ovat niin sanottuja luotipesusieniä, eli heitä pitää ampua todella paljon ennen kukistamista.

Viholliset käyttäytyvä ja hyökkäävät eri tavoin. On esimerkiksi vihollinen, joka katsoo hetken pelaaja ja lähtee sen jälkeen rynnistämään suoraa linjaa pelaajaa päin ja pelaaja voin näin väistää vihollista sivulle ja kääntyä ympäri ja ampu. Toinen esimerkki on vihollinen, joka lataa hetken iskuaan ja keskeyttää sen, jos pelaaja ehtii ampu vihollista tarpeeksi latauksen aikana.

Pienen yrityksen tulisi olla hyvin innovatiivinen ja luoda jotain aivan uusia vihollisia, jotka toimivat uudella tavalla ja tuovat siten jotain uutta nykyisen Doomien kaltaisiin mekaniikoihin. Koska parempaa Doomia on vaikea tehdä, on oltava luova. Pitää siis tehdä jotain radikaalisti eri tavalla. Pelkästään samanlaiset viholliset, kuin mitä muissa genren peleissä on, eivät kanna peliä pitkälle. Tämä tuo pelaamisen taktikointia ja mielenkiintoa. Vihollisia kannattaa suunnitella huolellisesti ja miettiä innovatiivisesti. Esimerkiksi vihollinen, jonka pää räjähtää osumasta ruiskuttaen sateenkaaren värejä, alkaa tappaa kaikkea ympärillään villisti juosten kohti ääntä, on vihollinen, jota ei kaikissa peleissä tule vastaan.

Tietysti vihollisille on myös perusasioita, kuten että ne huomaavat pelaajan tietyn etäisyyden päästä, tietyt viholliset menevät piiloon ja suojaan ja viholliset pudottavat asioita, muun muassa lisäelämää ja aseensa. Vihollisilla on myös animaatiot liikkumiseen, kiipeämiseen, hyppäämiseen, ampumiseen ja kuolemiseen tarvittaessa.

Tämän lisäksi tasoihin laitetaan esineitä ja asioita, jotka reagoivat tulitukseen, esimerkiksi kuvassa 14 nähtäviä rikkoutuvia laseja, lampuja ja pulloja. Myös räjähtävät tynnyrit ovat klassinen tapa antaa pelaajalle erilainen tapa tuhota vihollisia, mutta tässäkin kannattaa miettiä innovatiivisesti: vaihtaa esimerkiksi räjähtävät tynnyrit myrkyä suihkuttaviksi nallekarhuiksi.



Kuva 14. Tuhoutuvat esineet. Pelissä on esineitä, jotka tuhoutuvat niihin ammuttaessa. Palaset myös lentävät ylöspäin eivätkä vain hajoa.

Shootballiin tehtiin myös asioita, jotka tekivät pelaamisen paremmaksi, esimerkiksi pelaajalle annettiin mahdollisuus hypätä helpommin kohti maalia menevän pallon perään ja aseiden ampumatiheyttä pienennettiin, jotta pelaajilta vaaditaan enemmän tarkkuutta laukauksissa. Peliin on myös suunniteltu kerättäviä kuvakkeita, jotka muuttavat pelin kulua esimerkiksi tekemään kenttään liikkumista vaikeuttavia ansoja ja erilaisia käyttäytymisiä pallolle, kuten lisää painoa tai painovoimattomuutta.

#### 4.6 Huumori, komedia ja tarina

Tarinan kirjoittamisessa kaikilla on oma prosessinsa ja tapansa. Koska toiseen peleistä kuuluu huumori, komedia ja tarina, jotka ovat osa kokemusta, kerrotaan tarinasta ja hauskan tekemisestä ja prosessista tässä. Koko pelin tekoa voi lähestyä eri näkökulmasta ja keksiä ensin tarinan, jonka haluaa kertoa, ja sen jälkeen parhaan tavan tai genren kertoa se. Mutta koska on päätetty tehdä ensimmäisen persoonan ammuntopeli, mekaniikka tehtiin ensin ja tarina sen jälkeen.

Ensimmäisenä keksitään puitteet tai ympäristö: kertooko peli agentista kylmän sodan aikana vai onko päähenkilö demoneja tuhoava supersotilas. Tässä voi käyttää inspiraationa omia sen hetkisiä kiinnostuksenkohteita. Johnny English- ja Kingsman-elokuvien ollessa viimeisimpänä katselulistalla agenttiteema kuulostaa hyvältä ratkaisulta kahdesta syystä: jo edellä mainitut elokuvat todistavat, että agenttikomedia toimii ja on suosittu laji, ja sen takia, koska agenttitoiminta toimii hyvin ensimmäisen persoonan kuvakulmasta, hyvin usein siihen liittyvät aseet ja se on vauhdikasta.

Huumori ja komedia peleissä on oma taiteenlajinsa. Se vaatii tietynlaista mielenlaatua ja mielikuvitusta. Tietysti hyvä tapa on pelata muita komediapelejä, varsinkin omasta genrestään, jos niitä löytyy. Tässä tapauksessa hyvänä esimerkkinä toimisi Serious Sam 3: BFE, joka on ensimmäisen persoonan komedia-ammuntopeli. Kun vitsit on kirjoitettu ja laitettu peliin, niitä kannattaa testata koyleisöllä: tutuilla ja vieraililla.

Hyviä ja hauskoja asioita ovat esimerkiksi viittaukset muihin popkulttuuriasioihin, elokuviin, tv-sarjoihin ja muihin peleihin. Toinen, millä saa hauskaa sisältöä, on keksimällä hauskoja hahmoja, jotka voivat vaikka parodioida tosi elämän henkilöitä. Tärkein on kuitenkin se, mitä oman pään sisällä liikkuu. Esimerkiksi jos ammut vihollista päähän, pää räjähtää ja syntyy sienipilvi, joka on sateenkaaren väreissä. Tämä esimerkki ei tietenkään tarinaan vaikuta.

Tarinan kirjoittaminen etenee monella eri tavalla. Monesti tarinaa voi alkaa kirjoittaa yksin, mutta siitä saa huomattavasti paremman ja yksityiskohtaisen, kun on vastapuoli, jonka kanssa pallotella ideoita. Sen takia projektin tarinointiosuuteen hankittiin tukea

Barracuda Disasterin toiselta pelisuunnittelijalta ja äänivastaavalta, jonka kanssa tarinan tekemisestä oli jo aikaisempaa kokemusta.

Koska molemmat kehittäjät ovat hyvin samanmielisiä, kirjoitusprosessi eteni hyvin. Yleensä tavataan rauhallisessa paikassa kannettavan tietokoneen kanssa, lämmitellään noin puoli tuntia kahvin tai vastaavan piristeen parissa ja aletaan toimiin. Asiat etenevät suurin piirtein niin, että kirjoitetaan tarinaa askel askeleelta, ja kun tulee kohtaaus, jossa pitäisi tapahtua jotain, sitä jäädään pohtimaan, kunnes molemmat ovat tyytyväisiä. Molemmat siis heittelevät ideoita toisensa perään ja parasta, tai sitä, mihin jompikumpi keksii jatkoa, lähdetään rakentamaan. Asia lähtee yleensä ihan perusasiasta liikkeelle, mutta ajatus on, että mikään ei ole tylsää. Kun kohtausta naurattaa tai kun molemmat ovat tyytyväisiä, jatketaan eteenpäin. Lopussa tarina luetaan alusta asti ja kohtauksia muutellaan paremmiksi ja hauskemiksi. Sama tehdään vitsien, viittauksien ja hahmojen kohdalla.

Tarinaa kirjoittaessa on myös tärkeää pitää mielessä niin sanotut punaiset langat, jotta tarina ja kaikki tarinaan kuuluva liittyy ja kietoutuu toistensa kanssa hyvin yhteen. Vitsit eivät tarinan kannalta voi kaikki olla yksittäisiä hauskoja juttuja vaan paras on jos ne liittyvät maailmaan, tarinaan tai hahmoihin. Tämä näkyy kirjoitusprosessissa siten, että kun vitsi keksitään, sitä lähdetään hiomaan ja muokkaamaan, kunnes se on yhtä hauska kuin alun perin, mutta siinä on vielä lisäksi tietynlainen syvyys, eli se kietoutuu peliin ja sen maailmaan. Kun tarina on valmis, palataan kehittämään peliä. Valmiit tasot muokataan noudattamaan tarinaa, ja tarinalle tehdään uusia tasoja ja alueita. Myös kaikki kohtaukset, hahmot ja vitsit laitetaan toimiviksi.

Tarinan, pelattavuuden, vitsien ja tasojen ollessa hyvällä mallilla voidaan peliä ruveta näyttämään ja kokeilemaan oikeilla pelaajilla ja näyttämään maailmalle.

## 5 Valmis prototyyppi

### 5.1 Prototyyppien lopputulokset

Insinööriyön tulokset olivat positiiviset. Vaikka pelejä ei saatu julkaisukuntoon, projekteja tehdessä opittiin paljon. Yrityksellä on nyt käytössään hyvin monipuolinen ja muokattava ensimmäisen persoonan pelaajaobjekti, joka toimii hyvänä lähtökohtana mahdollisille tuleville ensimmäisen persoonan ammutapeleille.

Itse projektin kehityksestä opittua:

- Pelin koko ja ominaisuudet kannattaa skaalata tarpeeksi pieneksi.
- Olennaisten pääominaisuuksien teko on ensisijaista.
- Oman osaamisen tunnistaminen on tärkeä osa pelin suunnittelua.
- Ajanhallinta ja ajan löytäminen projektin tekoon on vaativaa.

Rahaa projekteihin ei mennyt muun kuin ajan muodossa. Mojitoon tehtiin noin viisitoista eri ominaisuutta pelaajalle ja noin kolme vihollisille. Shootballissa sen sijaan on ominaisuuksia noin 11.

Kuva 15 kertoo, miltä peli tässä vaiheessa näyttää ja miltä se toistaiseksi jää näyttämään, mutta se ei kerro pelin sisältämiä ominaisuuksia.



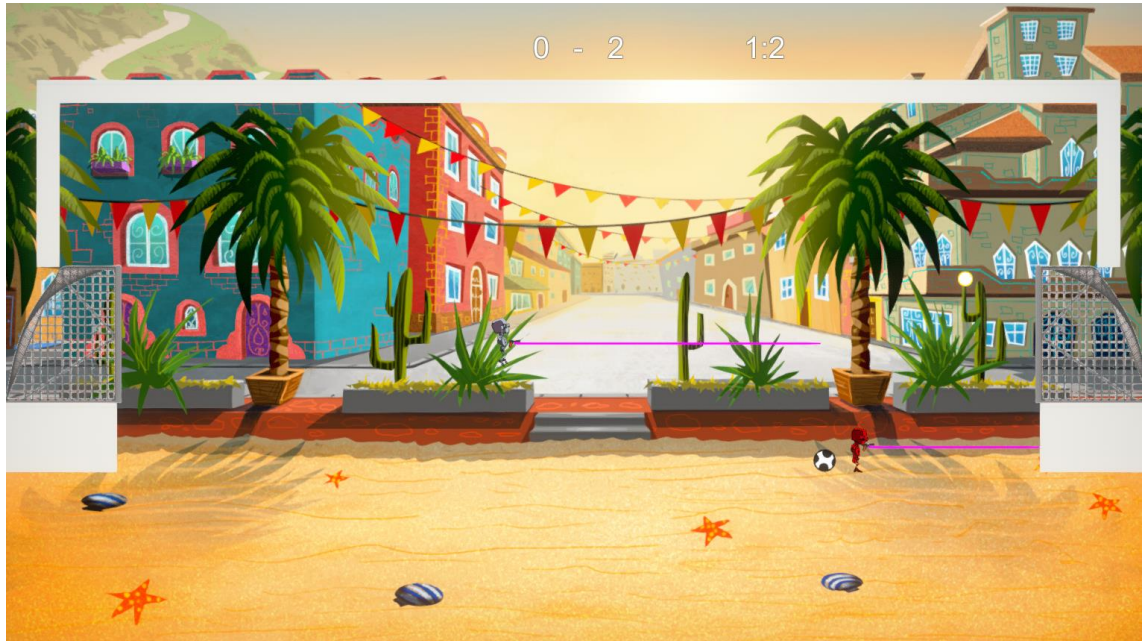
Kuva 15. Käytävä ja valot. Peliin tehtiin jo hieman grafiikkakokeilua ja valaisuolosuhteiden testaamista.

Ominaisuudet ovat ensimmäisen persoonan ammutapelissä seuraavat:

- ampuminen ja tähtääminen aseella
- liikkuminen pelaajahahmolla
- vuorovaikutus esineiden ja asioiden kanssa
- vihollisten kuolema
- pelaajan kuolema
- rikkoutuvat tavarat
- vihollisten liikkuminen ja partiointi
- palkki, joka kuvastaa pelaajan elämää
- aseiden poimiminen
- poimittujen aseiden selaaminen
- asiaankuuluvat animaatiot, muun muassa kävely ja vihollisanimaatiot
- fysiikka pohjainen rekyyli
- näytön täriä ammuttaessa
- äänet kaikessa edellä mainitussa.

Näiden ominaisuuksien tekemisestä tuli yhteensä noin 1 600 riviä koodia, ja animaatioita, materiaaleja, tekstuureja, malleja, ääniä ja vastaavia asioita syntyi noin 60.

Shootball on jo pelattavassa kunnossa kahdella pelaajalla, ohjaintuella, niin kuin kuvassa 16 näkyy.



Kuva 16. Kahden pelaajan ottelu. Shootball on täysin pelattavissa kahdella ohjaimella kahden pelaajan kesken.

Shootballin olemassa olevat ominaisuudet ovat seuraavat:

- kahden pelaajan liikkuminen ja ohjaintuki
- ampuminen
- pallon fysiikka
- maalit ja maalien teko
- aikalaskuri
- pelin loppuminen
- efektejä maaliin menosta, muun muassa hidastus ja räjähdys
- ääniefektit yläriimaan, maahan ja toiseen pelaajaan
- musiikki
- kahden pelaajan vastakkain pelaaminen

- pelin uudelleen aloitus.

Näiden ominaisuuksien tekemisestä tuli yhteensä noin 1 200 riviä koodia, ja animaatioita, materiaaleja, tekstuureja, malleja, ääniä ja vastaavia asioita syntyi noin 70. Työtunteja prototyyppeihin ja insinööriyöhön on mennyt yhteensä noin 250 tuntia, josta suurin osa itse työsuuteen, eli lähteiden etsimiseen, tutkimiseen, kokeiluun ja kirjoittamiseen.

Eniten hankaluuksia molemmissa peleissä aiheutti ampumisen saaminen hyvántuntuiseksi. Tämä on kuitenkin tärkeä osa molempia pelejä, ja koska markkinoilla on paljon ampumapelejä, on ampumisen rimakin aika korkealla. Tätä voi kompensoida muilla ominaisuuksilla ja ulkoasulla, mutta ampumisen on silti oltava vähintään hyvää. Muuten molemmat projektit etenivät mallikkaasti prototyypeiksi.

Ensimmäisen persoonan ammuntopeli ei valmistu toistaiseksi, koska se vaatisi täyspäiväistä työskentelyä noin puolen vuoden verran tai osa-aikaista työskentelyä noin vuoden verran. Sille tarvittaisiin myös vähintään yksi henkilö lisää tekemään äänimaailmaa tai grafiikkaa. Myös se, että hyviä ammuntopelejä on markkinoilla paljon, vaikuttaa siihen, ettei peliä kannata tällä hetkellä jatkaa, ellei sitä saa skaalattua paljon pienemmäksi ja nopeammaksi tehdä, kuitenkin niin, että pelistä saa laadukkaan ja hyvän.

Shootball taas ei tämän insinööriyön aikana valmistunut, koska raportin kirjoittaminen vaatii paljon aikaa ja projekti aloitettiin hyvin loppuvaiheessa tätä työtä. Shootballin kehittäminen kuitenkin jatkuu osa-aikaisesti, koska projekti on huomattavasti pienempi ja yksinkertaisempi ja silti hauska. Tavoite olisi saada se valmiiksi alkuvuodesta 2019.

## 5.2 Prototyyppien seuraavat askeleet

Seuraavaksi Shootball kehitetään loppuun ja julkaistaan, jolloin portfolio kasvaa, osaaminen kehittyy ja saadaan mahdollista yleisöä tai seuraajakuntaa. Shootballin kehitys yrityksen ison projektin lisäksi on tehtävissä, koska se vaatii paljon pienempiä resursseja toimiakseen. Seuraavat projektit ovat jo mielessä, ja kun aika on oikea, kehitetään niistäkin prototyypit toimivuuden testaamiseksi. Silloin tästä insinööriyöstä ja siinä opitusta on paljon apua tehokkuuden takaamiseksi. Mitä ensimmäisen persoonan ammuntopeliin ja siinä potentiaalisen huumorin maailmaan tulee, tulevaisuudessa genreen varmasti

palataan joko yrityksenä tai muuna projektina, varsinkin kun tämän jälkeen aiheesta on paljon uutta tietoa, osaamista ja ymmärrystä.

## 6 Yhteenveto

Insinöörityössä selvitettiin, kokeiltiin ja kuvattiin pelinkehityksen kaari alusta loppuun saakka: kokeiltiin nopean prototyypin kehitystä ja sen kaikkia vaiheita, peli-idean suunnittelua, peliin vaadittavan osaamisen tunnistamien, pelin sisällön suunnittelua, hankintaa ja toteutusta sekä markkinointia ja pelin julkaisua.

Pelin kehitys aloitettiin ideatasolta, jolloin päädyttiin ensimmäisen persoonan ammunta-peliin ja 2D-ammuntajalkapallopeleihin. Näitä lähdettiin lähestymään erilaisten menetelmien avulla askel kerrallaan, aina nopeasta prototyypin teosta vaativampiin ominaisuuksiin. Työssä toteutettiin pelin ideointi ja idean skaalaus, prototyyppi, lisäominaisuuksia ja hiottuja perusominaisuuksia. Kaikki oleellinen prototyypin kehityksen vaiheista alusta loppuun käytiin läpi.

Havaintoja, jotka eivät niinkään yllättäneet, vaan kävivät toteen, ovat pelin koko ja aika, jonka yhden pelin kehitys vaatii. Havaittiin, että ensimmäisen persoonan ammunta-peli on mahdollista toteuttaa yksin tai kaksin, mutta projekti vaatii lähes täyspäiväistä, vähintään puolen vuoden, sitoutumista, jos kehitysaika halutaan pitää järkevänä. Aikaiseksi kuitenkin saatiin ensimmäisen persoonan ampuminen ja vuorovaikutus. Myös muun muassa vihollisten kuolema, pelaajan elämät, rekyyli, aseiden vaihto ja aseiden heitto ovat toteutettuja ominaisuuksia. Barracuda Disasterilla on nyt käytössä hyvin muokattava ja kattava ensimmäisen persoonan pelaajaohjaus. Myös markkinointiin ja julkaisuun paneuduttiin. Sosiaalisen median markkinointisuunnitelma tehtiin ja julkaisuprosessi on nyt tuttu. Toisessa prototyypissä havaittiin, että jos pelin sisällön ja toteutustavan suunnittelee oman osaamisen ja toimivien keskeisten ominaisuuksien ympärille, peli on tehtävissä järkevässä ajassa myös osa-aikaisella työskentelyllä. Tämä on tärkeää, jos haluaa pelialalle, koska portfolio on kaikki kaikessa ja itse tekeminen.

Työn tavoitteet saavutettiin hyvin. Julkisesti pelejä ei päästy vielä näyttämään, vaikka se oli yksi hieman toiveikkaammista tavoitteista. Aika monen projektin ja muun työasian

keskellä ei riittänyt täyden julkisen prototyypin tai pelin tekemiseksi. Shootballilla on nyt kuitenkin tarkka suunnitelma, miten projektin kanssa edetään, miten sitä markkinoidaan ja missä se julkaistaan, kun pelin työstämistä jatketaan.

Projekti opetti paljon asioita pelinkehityksen eri vaiheista ja varsinkin pelin koosta ja nopean prototyypin tekemisestä, muun muassa sitä, kuinka keskittyä oikeisiin asioihin peliä tehdessä ja tehokas ja että nopea ominaisuuksien kokeilu ja suunnittelu ovat tärkeitä asioita, jotka pelinkehityksestä kannattaa tietää.

Barracuda Disasterilla on käytössä hyvin muokattava ja kattava ensimmäisen persoonan pelaajaohjaus, jota pystytään hyödyntämään yrityksen nykyisissä ja tulevista projekteissa.

## Lähteet

- 1 Yu, Derek. 2013. Making it in Indie Games: Starter Guide. Verkkoaineisto. <<http://makegames.tumblr.com/post/44181247500/making-it-in-indie-games-starter-guide>>. 27.2.2013. Luettu 13.3.2018.
- 2 Yu, Derek. 2010. Finishing a Game. Verkkoaineisto. <<http://makegames.tumblr.com/post/1136623767/finishing-a-game>>. 17.9.2010. Luettu 13.3.2018.
- 3 Prototyping: You're (Probably) Doing It Wrong. 2010. Verkkoaineisto. Games from Within. <<http://gamesfromwithin.com/prototyping-youre-probably-doing-it-wrong>>. 12.8.2010. Luettu 20.9.2018.
- 4 How To Form A Solid Indie Game Development Team. 2014. Verkkoaineisto. New York Film Academy. <<https://www.nyfa.edu/student-resources/forming-solid-indie-game-development-team>>. 25.11.2014. Luettu 23.9.2018.
- 5 Bestebroer, Pascal. 2017. Stop being a stupid indie-gamedeveloper. Verkkoaineisto. <[http://www.gamasutra.com/blogs/PascalBestebroer/20170809/303382/Stop\\_being\\_a\\_stupid\\_indiegamedeveloper.php](http://www.gamasutra.com/blogs/PascalBestebroer/20170809/303382/Stop_being_a_stupid_indiegamedeveloper.php)>. 9.8.2017. Luettu 23.9.2017.
- 6 McMillen, Edmund. 2009. Opinion: Indie Game Design Do-s and Don't-s: A Manifesto. Verkkoaineisto. <[https://www.gamasutra.com/view/news/117521/Opinion\\_Indie\\_Game\\_Design\\_Dos\\_and\\_Donts\\_A\\_Manifesto.php](https://www.gamasutra.com/view/news/117521/Opinion_Indie_Game_Design_Dos_and_Donts_A_Manifesto.php)>. 30.12.2009. Luettu 20.2.2018.
- 7 Quinn, Zoe. 2013. A Beginner's Guide To Making Your First Video Game. Verkkoaineisto. <<https://www.kotaku.com/5979539/a-beginners-guide-to-making-your-first-video-game>>. 1.28.2018. Luettu 21.2.2018.
- 8 Gray, Kyle; Gabler, Kyle; Shodhan, Shalin & Kucic, Matt. 2005. How to Prototype a Game in Under 7 Days. Verkkoaineisto. <[https://www.gamasutra.com/view/feature/130848/how\\_to\\_prototype\\_a\\_game\\_in\\_under\\_7\\_.php](https://www.gamasutra.com/view/feature/130848/how_to_prototype_a_game_in_under_7_.php)>. 26.10.2005. Luettu 20.11.2017.
- 9 Tulleken, Herman. 2014. Rapid Game Prototyping: Tips for Programmers. Verkkoaineisto. <[https://www.gamasutra.com/blogs/HermanTulleken/20140119/208901/Rapid\\_Game\\_Prototyping\\_Tips\\_for\\_Programmers.php](https://www.gamasutra.com/blogs/HermanTulleken/20140119/208901/Rapid_Game_Prototyping_Tips_for_Programmers.php)>. 19.1.2014. Luettu 20.11.2017.
- 10 The Top 10 Video Game Engines. 2018. Verkkoaineisto. Gamedesigning.org. <<https://www.gamedesigning.org/career/video-game-engines>>. 12.5.2018. Luettu 3.10.2018.

- 11 ElHady, Hady. 2017. Top Game Engines In 2018. Verkkoaineisto. <<https://instabug.com/blog/game-engines/>>. 7.12.2017. Luettu 3.10.2018.
- 12 Barret, Mark. 2017. Hitchhiker's Guide to Rapid Prototypes! Verkkoaineisto. <<https://www.youtube.com/watch?v=sYWkiv1hTPM>>. Katsottu 6.9.2017.
- 13 Forest, Danny. 2017. Tried and True: 5 tips on building a game prototype. Verkkoaineisto. <<https://medium.com/power-level/tried-and-true-5-tips-on-building-a-game-prototype-ad0273b12697>>. 17.7.2017. Luettu 8.10.2018.
- 14 About itch.io. 2018. Verkkoaineisto. Itch.io. <<https://itch.io/docs/general/about>>. Luettu 20.10.2018.
- 15 Steam Direct Now Available. 2017. Verkkoaineisto. Steam. <<https://steamcommunity.com/games/593110/announcements/detail/1328973169870947116>>. 13.6.2017. Luettu 13.6.2017.
- 16 11 Places to Publish & Release Your Indie Game. 2017. Verkkoaineisto. Ninichi. <<https://ninichimusic.com/blog/2017/9/1/11-places-to-publish-release-your-indie-game>>. 12.9.2017. Luettu 20.10.2018.
- 17 The Original Gameplay Prototype. 2014. Verkkoaineisto. Superhot. <<https://superhotgame.com/play-prototype>>. 2017. Luettu 20.10.2018.
- 18 Francis, Bryant. 2015. Crucial lessons from 7 game jam prototypes that went commercial. Verkkoaineisto. <[https://www.gamasutra.com/view/news/250332/Crucial\\_lessons\\_from\\_7\\_game\\_jam\\_prototypes\\_that\\_went\\_commercial.php](https://www.gamasutra.com/view/news/250332/Crucial_lessons_from_7_game_jam_prototypes_that_went_commercial.php)>. 25.8.2015. Luettu 20.10.2018.
- 19 Hankekehittelytuki peleille. 2017. Verkkoaineisto. Mediadesk. <<http://www.mediadesk.fi/tuet/hankekehittelypeleille.shtml>>. 2017. Luettu 15.1.2018.
- 20 Making Your First Game: Minimum Viable Product - Scope Small, Start Right. 2015. Verkkoaineisto. Extra Credits. <<https://youtu.be/UvCri1tqIxQ>>. Katsottu 15.3.2018.
- 21 Doom, PlayStation 4. 2016. Verkkoaineisto. Metacritic. <<https://www.metacritic.com/game/playstation-4/doom>>. 13.5.2016. Luettu 20.09.2018.
- 22 Trello. 2018. Verkkoaineisto. <<https://trello.com/>>.
- 23 Smith, Edward. 2017. A year later, Doom is still the best FPS on PS4, Xbox One, and now Switch. Here's how it happened. Verkkoaineisto. <

<https://www.gamesradar.com/a-year-later-doom-is-still-the-best-fps-on-ps4-xbox-one-and-now-switch-heres-how-it-happened/>. 10.11.2017. Luettu 20.4.2018.

- 24 Declercq, Alexis. 2017. AD Design. Verkkoaineisto. <<http://alexisdeclercq.com/game-art/>>. 2017. Luettu 26.10.2018.

