



Pelinkehitys Android - laitteelle

Juho Lahti

OPINNÄYTETYÖ
Kesäkuu 2019

Tieto- ja viestintäteknikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikan koulutus
Ohjelmistotekniikka

LAHTI, JUHO:
Pelinkehitys Android - laitteelle

Opinnäytetyö 25 sivua
Kesäkuu 2019

Tässä opinnäytetyössä käsiteltiin pelinkehitysprosessia Android – laitteelle yhden kehittäjän näkökulmasta. Tarkoituksena oli suunnitella peli Android – laitteille ja kehittää suunniteltu peli käyttäen Unity – ohjelmistoa kehitysympäristönä. Pelimekaniikka, konsepti, käyttöliittymä, laajennettavuus ja jatkokehitys, tarina sekä hahmot suunniteltiin peliä varten. Kaikki peliin vaadittava grafiikka ja äänimateriaali oli tarkoitus tehdä itse. Grafiikoiden luomiseen käytettiin ArtRage – piirto-ohjelmistoa ja animointi tapahtui Unity – ohjelmistolla.

Tuloksena opinnäytetyöstä syntyi suunnitelma pelisovelluksesta ja sen toteuttamisesta, jonka kehitys on vielä kesken.

Yksittäiselle pelin kehittäjälle yksi suurimmista haasteista oli grafiikoiden luominen. Piirtotyölin ja teeman suunnittelu oli suhteellisen vaikeaa ja aikaa vievää piirrettäessä asioita monta kertaa uudelleen eri tavalla. Kokemuksen puute oli haasteena grafiikoiden luonnissa. Pelinkehitysprosessissa olisi hyvä olla mukana sekä ohjelmoija, että graafikko hyvän lopputuloksen takaamiseksi ajallaan.

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

LAHTI, JUHO:
Game development on Android

Bachelor's thesis 25 pages
June 2019

This thesis covered the development process of a mobile game which was developed for Android devices by a single developer. The purpose of this thesis was to design and create a mobile game application for Android with Unity. The game mechanics, concept, user interface, modularity, further development, story and the characters were all designed for the game. All necessary graphics and audio were supposed to be made for the game by the single developer. Drawing program called ArtRage was used to create the artwork for the game and animating was done in Unity.

The result of this thesis was a plan of a mobile game and its' game mechanics. The game is still in development.

One of the biggest challenges for a single developer was creating the artwork for the game. Drawing style and the theme of the game were quite challenging and time consuming for inexperienced artist to create. Resulting in redrawing some of the artwork multiple times. In the game development projects to come, it would be good to have a programmer and a concept artist both to ensure a good result on time.

Key words: unity, game development

SISÄLLYS

1	JOHDANTO	5
2	KEHITYSYMPÄRISTÖ	6
	2.1 Android.....	6
	2.2 Unity.....	6
	2.3 Visual Studio	7
3	GRAFIKKA- JA ÄÄNITYÖKALUT	9
	3.1 ArtRage.....	9
	3.2 Audacity	9
4	SUUNNITTELU.....	10
	4.1 Kehitysympäristö.....	10
	4.2 Konsepti	11
	4.3 Pelimekaniikka	11
	4.3.1 Liikkuminen	11
	4.3.2 Väistäminen.....	11
	4.3.3 Taistelu.....	13
	4.4 Kehittyminen	14
	4.5 Käyttöliittymä.....	14
	4.6 Tarina.....	15
	4.7 Hahmot	15
	4.8 Laajennettavuus.....	16
5	TOTEUTUS	17
	5.1 Sovellus	17
	5.1.1 Hahmot.....	17
	5.1.2 Tasot	18
	5.2 Grafiikka.....	18
	5.3 Käyttöliittymä.....	21
	5.4 Testaus	24
6	LOPPUTULOKSET JA JATKOKEHITYS.....	25
	6.1 Pelisovellus	25
	6.2 Julkaisu	25
	6.3 Jatkokehitys	26
7	POHDINTA	27
	LÄHTEET.....	28

1 JOHDANTO

Tässä opinnäytetyössä käsitellään pelinkehitysprosessia Android – laitteelle yhden kehittäjän näkökulmasta valituilla menetelmillä. Käytetyt työkalut ja kehitysympäristöt esitellään lyhyesti, jonka jälkeen käsitellään pelinkehitysprosessia. Pelinkehitysprosessi on jaettu kolmeen eri osaan, suunnitteluun, toteutukseen ja lopputuloksiin. Suunnitteluvaiheessa esitetään pelisovelluksen konseptin ideointia, pelimekaniikan suunnittelua ja rakentamista sekä laajennettavuuden suunnittelua. Toteutus käsittelee pelinkehitysprosessia ja pelisovelluksen tekoa. Lopputuloksissa esitetään pelinkehitysprosessin tämän hetkinen tilanne sekä jatkokehitys suunnitelmat.

Opinnäytetyönaiheen toimeksiantajana on opinnäytetyöntekijä. Aiheen taustalla on tekijän oma mielenkiinto pelimekaniikan suunnitteluun ja rakentamiseen sekä nykyaikaisten mobiililaitteiden lisääntynyt suorituskyky, mikä mahdollistaa responsiivisen ja tarkan kontrolloinnin mobiilipeleissä.

Tavoitteena tässä opinnäytetyössä on suunnitella ja toteuttaa Android – laitteille haastava peli, jossa kontrollointi on mahdollisimman tarkkaa ja viiveetöntä käyttäjän näkökulmasta. Kaikki peliin tarvittava grafiikka ja äänimateriaali on tarkoitus luoda itse. Pelisovelluksessa pelaajan on tarkoitus voittaa vastustajia ja kehittää omaa peli hahmoaan. Pelaaja saa voittaessaan kokempisteitä, joita voi käyttää hahmon kehittämiseen. Sovellus toteutetaan kaksiulotteisena, eli pelaajan näkymä on sivulta päin ja hahmoa voi liikuttaa joko oikealle tai vasemmalle. Taistelumekaniikka ja liikkuminen toteutetaan painikkeilla. Painikkeilla pelaaja voi väistää, iskeä tai liikkua vasemmalle ja oikealle. Pelaaja voi myös muuttaa käytössään olevia iskuja pelin asetusvalikosta. Pelin ensimmäiseen versioon on tarkoitus toteuttaa yksi haastava vastustaja.

2 KEHITYSYMPÄRISTÖ

2.1 Android

Android on Googlen kehittämä mobiilijärjestelmä, jota käytetään puhelimissa, autoissa, kodinelektroniikassa sekä sulautetuissa järjestelmissä. Tämä Linux-pohjainen avoimen lähdekoodin järjestelmä on kehitetty useisiin eri tarkoituksiin ja erilaisille laitteille. Android – järjestelmä on laajassa käytössä maailmalla ja sitä käyttää yli 2 miljardia ihmistä. (Google Developers n.d.)

Pelinkehitykseen Android – järjestelmille on olemassa useita eri työkaluja, joita kehittäjä voi hyödyntää. Kehitykseen on lukuisia kirjastoja, joita hyödyntäen voi käyttää Android – laitteiden ominaisuuksia sekä sensoreita. Pelejä varten on myös olemassa valmiita pelimoottoreita, kuten Unity ja Unreal, joiden avulla sovelluksen luonti, testaus ja muokkaaminen on yksinkertaista. (Android game development n.d.)

Google Play Store mahdollistaa pelien julkaisun ja levittämisen kaikille yhteensopiville Android – laitteille. Google esittää useita eri ohjeita julkaisua varten, tarjoaa palautekanavan kehittäjille sekä mahdollisuuden ennakkotilauksiin. Google Play antaa myös mahdollisuuden paikalliseen levittämiseen, jotta kohderyhmä on helpompi saavuttaa. Googlessa on myös valmiita työkaluja käytön seurantaan ja analysointiin. (Android game development n.d)

2.2 Unity

Unity on laajasti käytössä oleva pelimoottori ja kehitysympäristö, jota voi käyttää pelien ja sovelluksien kehitykseen. Unity on ilmainen kehittäjille, joiden liikevaihto ei ylitä sataa tuhatta dollaria vuodessa. Kehitykseen on tarjolla runsaasti ohjeita ja esimerkkejä useille eri pelitoiminnoille ja ominaisuuksille. (Unity 2019.)

Unity tukee yli 25 eri alustaa, kuten konsoleita, tietokoneita, televisioita ja puhelimia. Sen avulla voi luoda sovelluksia kaikille näille järjestelmille. Unity tukee myös

VR ja AR – tekniikoita, eli virtuaalista- sekä lisättyä todellisuutta. Unity antaa kehittäjille myös mahdollisuuden animoida ja luoda elokuvamaista sisältöä peleihin. Pelimaailman luonti, testaus ja muokkaaminen on myös mahdollista samassa kehitysympäristössä. (Unity 2019.)

Unity Editor tarjoaa lukuisia eri työkaluja ja mahdollistaa nopean kehityksen sekä testauksen reaaliajassa. Kehitysympäristö on saatavilla Windows, Mac ja Linux – käyttöjärjestelmille. Sovelluksia pystyy demoamaan koko kehityksen ajan ilman Android – laitetta suoraan kehitysympäristössä. Unity tukee ja tarjoaa työkaluja 2D sekä 3D kehitykselle. Pelin fysiikkaa ja tekoälyä varten on myös valmiita työkaluja, joilla niiden muokkaaminen ja luonti on helppoa. Unity Editor antaa myös mahdollisuuden kehittäjille luoda omaa sisältöä ja jakaa sitä eteenpäin muille kehittäjille. Unity Editor tukee monia eri ääni-, video-, kuva- ja tekstiformaatteja, joiden ansiosta luotua sisältöä voi siirtää suoraan kehitysympäristöön. (Unity 2019.)

Pelimoottori mahdollistaa nopean kehityksen, koska sovelluksia pystyy testaamaan reaaliaikaisesti suoraan Unity Editorissa. Pelimoottori kykenee myös käsittelemään suuria määriä 2D grafiikka, joka mahdollistaa nesteiden, pilvien ja tulen luonnin peleissä. Kehittäjät pystyvät myös halutessaan laajentamaan ja muokkaamaan pelimoottoria. Pelilogiikkaa pystyy ohjelmoimaan esimerkiksi C# ja Unity Script – kieliä käyttäen ja ohjelmointi sekä koodin testaus onnistuu Visual Studio – ohjelmistolla reaaliajassa natiivin tuen ansiosta. (Editor 2019.)

2.3 Visual Studio

Visual Studio on ohjelmointiin tarkoitettu ohjelmisto, jolla pystyy ohjelmoimaan ja testaamaan esimerkiksi C# - ohjelmointikieltä, joka soveltuu hyvin Unity:n kanssa työskentelyyn. Visual Studio on mahdollista asentaa Unity – ohjelmiston asennuksen yhteydessä ja Unity:n omat ohjelmointikirjastot toimivat sen kanssa suoraan asennuksen jälkeen, niitä erikseen lisäämättä.

Visual Studio tarjoaa valmiita työkaluja ohjelmakoodin nopeaan testaukseen ja diagnosointiin. Työkaluja on myös tarjolla ryhmätyöskentelyyn sekä ohjelmakoodin ajamiseen pilvessä sekä testaukseen oikeilla laitteilla. (Visual Studio 2019)

Visual Studio antaa kehittäjälle mahdollisuuden asentaa vain ne työkalut ja kehitysympäristöt, joita kehittäjä tarvitsee. Visual Studio tukee esimerkiksi Azure, Python, Node.js sekä ASP.NET tekniikoita web- ja pilvipalveluiden kehityksessä. Pelienkehityksessä Visual Studio tukee .NET ja C++ - ohjelmointikieliä sekä se toimii yhdessä Unity – ohjelmiston kanssa. Datan prosessointiin ja analysointiin on myös valmiita työkaluja. (Visual Studio 2019.)

3 GRAFIKKA- JA ÄÄNITYÖKALUT

3.1 ArtRage

ArtRage – ohjelmisto on piirtotyökalu, joka tarjoaa realistisia piirtovälineitä ja piirtoalustoja artisteille. Ohjelmisto on maksullinen, mutta kilpailijoihinsa, kuten Adobe Photoshop, verrattuna kyseessä on kertamaksu, jonka jälkeen lisenssi on elinikäinen. Ohjelmisto soveltuu aloittelijoille sekä ammattilaisille ja se tarjoaa monipuolisia työkaluja piirtämiseen. (ArtRage 2019.)

Ohjelmisto mahdollistaa piirtämisen eri tasoissa, joille on mahdollista määrittää erilaisia ominaisuuksia ja efektejä. Ohjelmistossa on myös perspektiiviä varten työkaluja sekä piirtovälineitä voi muokata haluamallaan tavalla. Ohjelmisto soveltuu luonnosteluun sekä useiden piirtovälineiden avulla voi luoda öljypohjaisia maalauksia, vesiväripiirroksia ja monenlaista muuta. ArtRage – yhteisö tarjoaa apuja ja ohjeita piirtämiseen sekä käyttäjien itse luomaa sisältöä, kuten erilaisia piirtovälineitä ja kehyksiä. (ArtRage 2019.)

Pelinkehitystä ja sovelluksia varten pystyy ArtRage – ohjelmiston avulla luomaan taustattomia grafiikoita, joiden kokoa ja tiedostomuotoa pystyy hallitsemaan. Animointi – ominaisuutta ohjelmistossa ei ole, mutta sillä pystyy luomaan Unity – ohjelmistolle kuva-arkkeja, jotka voi Unity ohjelmistolla animoida.

3.2 Audacity

Audacity on ohjelmisto, jolla voi muokata äänitiedostoja ja nauhoittaa ääniä. Kyseessä on ilmainen avoimen lähdekoodin ohjelmisto, jota voi käyttää Windows, Mac ja Linux – järjestelmillä. Ohjelmistolla voi leikata, kopioida ja sekoittaa äänitiedostojen sisältöä ja se tukee monia eri tiedostomuotoja. Audacity tarjoaa myös monia eri työkaluja, joilla ääniä voi muokata. (Audacity 2019.)

4 SUUNNITTELU

4.1 Kehitysympäristö

Android – järjestelmälle on helppo kehittää sovelluksia ja kehitykseen on useita eri työkaluja, joilla voidaan saavuttaa sama lopputulos. Yksi vaihtoehtoista on Unity ja sen pelimoottori, joilla pelisovelluksen voi tehdä melko yksinkertaisesti sekä kehitys on ilmaista sadan tuhannen dollarin liikevaihtoon asti, mikä soveltuu projektiin hyvin. Kehittäjänä ei ole yritys ja sovelluksesta tulee ilmainen, joten Unity sopii kehitysympäristöksi erinomaisesti.

Pelisovelluksen voisi myös kehittää käyttäen vastaavaa Unreal – ohjelmistoa, mutta Unity:n 2D ominaisuuksien ja työkalun käyttökokemuksen johdosta ei pelisovellusta kehitetä Unreal – ohjelmistolla ja sen pelimoottorilla.

Android Studio – ohjelmistolla kehitys myös onnistuisi, mutta Unity ja Unreal tarjoavat pelisovelluksille helpon testausympäristön, koska peliä pystyy niiden avulla demoamaan ja testaamaan kokonaisuudessaan suoraan kehitysympäristössä.

Kehitysympäristön valinnassa on erityisen tärkeää, että pelin toimintoja, grafiikoita ja tasoja voidaan testata nopeasti pienellä vaivalla. Valintaan vaikuttaa myös käytössä olevat työkalut, kuten Unity:n peliobjekti rakenne, joka mahdollistaa ohjelmakoodin ja peliobjektien helpon yhdistämisen toisiinsa. Unity:n valmiit kirjastot ja komponentit, peliobjektien muokkaukseen ja hallinnointiin helpottavat myös kehitystä huomattavasti.

Unity – ohjelmiston kanssa toimii natiivisti Microsoft Visual Studio – ohjelmisto, jolla voi kirjoittaa C# - ohjelmointikielellä pelilogiikkaa. Visual Studio:ssa on Unity:n omat kirjastot valmiina, mikäli sen asentaa Unity - ohjelmiston yhteydessä. Ohjelmistojen natiivi yhteensopivuus mahdollistaa ohjelmakoodin välittömän testauksen Unity – ohjelmistossa.

4.2 Konsepti

Pelisovelluksesta tulee hack-and-slash – tyyppinen peli, eli käyttäjä kontrolloi pelihahmoa, jonka tavoitteena on voittaa vastaan tulevat vastustajat. Pelistä tulee sivulta päin katsottava, jossa liikutaan joko vasemmalle tai oikealle. Käyttäjä kykenee liikuttamaan pelihahmoa, väistämään iskuja sekä iskemään vastustajia. Konseptin valintaan on lähinnä vaikuttanut mielenkiinto tämän tyyppisen taistelumeکانikan suunniteluun ja toteutukseen.

4.3 Pelimeکانikka

4.3.1 Liikkuminen

Hahmon liikutteluun on muutamia eri vaihtoehtoja, käyttöliittymässä voi olla painikkeita liikuttelua varten, joilla pelaaja voi liikkua vasemmalle tai oikealle. Vaihtoehtoisesti voidaan hyödyntää mobiililaitteiden kosketusnäyttöjä, eli hahmo liikkuu siihen suuntaan mihin pelaaja sormenpäällään koskettaa.

Tässä tapauksessa painikkeiden käyttö on järkevää, koska hahmo kykenee liikumaan vain vasemmalle tai oikealle. Kosketusnäyttö toimii paremmin kolme ulotteiselle pelille, jossa voidaan liikkua kaikkiin suuntiin.

4.3.2 Väistäminen

Väistämistä varten peliin on mahdollista laittaa myös painike tai vaihtoehtoisesti se voisi toimia esimerkiksi pyyhkäisemällä näyttöä tiettyyn suuntaan. Liikkuminen on mahdollista kahteen eri suuntaan, joten väistäminen pyyhkäisemällä on toimiva vaihtoehto, mutta liikkuminen toteutetaan painikkeilla, mikä tekisi pyyhkäisemällä väistämisestä epäkäytännöllistä pelaajan kannalta. Painike väistöä varten on tässä tapauksessa parempi vaihtoehto, kun hahmoa myös liikutetaan painikkeilla. Väistö on myös mahdollista toteuttaa painamalla liikuttelupainiketta kaksi kertaa peräkkäin, jolloin hahmo tekee väistöliikkeen siihen suuntaan mihin painike liikuttaa hahmoa.

Väistöliikkeen toimintaperiaatteelle on eri vaihtoehtoja. Se voi toimia niin, että hahmo liikkuu vastustajasta kauemmas, niin ettei vastustajan liike osu pelaajan hahmoon laisinkaan. Tämä vaihtoehto vaatii tarkkuutta ja hyvin pientä viivettä väistöliikkeen ja pelaajan painalluksen välillä, jotta pelaajalle tulisi hyvä tunne hahmon kontrolloinnista. Pelaajan pitäisi saada sellainen vaikutelma, että väistöliike tapahtui heti painiketta painaessa.

Vaihtoehtoisesti väistöliike voi perustua niin kutsuttuihin ”invulnerability frames” eli haavoittumattomuuskehysiin. Tällä tarkoitetaan sitä, että hahmon suorittaessa väistöliikkeen animaatiota, vastustaja ei pysty vahingoittamaan pelaajaa. Hahmo voi olla vahingoittumaton myös vain osan väistöliikkeen animaatiosta.

Väistöliikkeen toimintaperiaatteella on suuri vaikutus pelin vaikeusasteeseen. Ensimmäisellä periaatteella vastustajan liikenopeudella, väistöliikkeen nopeudella ja pelaajan reagointinopeudella on suora vaikutus pelaajan kokemaan vaikeusasteeseen. Väistöliikkeen ollessa hidasliikkeinen verrattuna vastustajan iskujen nopeuteen, on pelaajan ajoitettava väistöliike erittäin tarkasti, jos tässä tapauksessa pelaaja ei kykene reagoimaan tarpeeksi nopeasti, voi lopputuloksena syntyä mahdoton haaste pelaajalle.

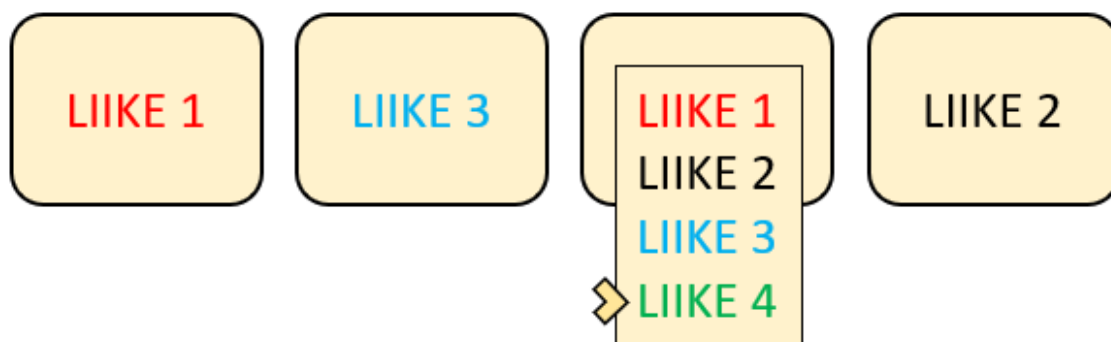
Toisessa periaatteessa erityisesti painikkeen painalluksen ja suoritettavan väistöliikkeen välinen viive tulee olla pieni tai sillä on suora vaikutus pelin vaikeusasteeseen. Tässä tapauksessa merkitystä on myös sillä, onko hahmo koko animaation ajan haavoittumaton vai vain osan siitä, ja onko haavoittumaton osa animaation alussa, keskellä vai lopussa. Sen sijoituessa animaation alkuun korostuu pelaajan reagointinopeus, koska pelaaja voi tällöin väistää iskuja ilman suurempaa ennakkointia. Keski- ja loppuvaiheilla kyse olisi taas väistön ajoittamisesta oikeaan hetkeen, mikä tekisi pelistä entistä haastavamman opettelematta vastustajan liikkeitä ensin.

Pelisovelluksessa käytetään ensisijaisesti ensimmäistä toimintaperiaatetta, eli hahmon tulee väistyä vastustajan iskujen alta pois. Mikäli väistöä ei saada tarpeeksi responsiivisesti toteutettua tällä tavalla, tehdään se käyttäen toista periaatetta.

4.3.3 Taistelu

Taistelu toteutetaan myös painikkeella, jota painellen hahmo tekee iskuja. Pelaajan tulee liikuttaa hahmo tarpeeksi lähelle vastustajaa, jotta iskut osuvat siihen. Vastustajan liikkeitä kontrolloi satunnaisgeneraattori, joka vaihtelee vastustajan suorittamaa liikettä. Vastustajilla tulisi olisi useita eri liikkeitä, jotta taistelu pysyy mielenkiintoisena, jos se toistaa vain yhtä tai kahta erilaista liikettä, tulee pelissä liikaa toistoa ja se vaikuttaa pelaajan käyttökokemukseen. Eri liikkeitä voi jakaa eri vaiheisiin, joihin siirrytään tietyn ajan tai pelitilanteen jälkeen. Esimerkiksi pelaajan tehdessä tarpeeksi vahinkoa, vastustaja siirtyy seuraavaan vaiheeseen ja sen liikkeitä muuttuvat. Näin taisteluun saadaan vaihtelua ja mielenkiintoa.

Pelaajan hahmon liikkeille toteutetaan valikko, jolla pelaaja voi muokata millaisia liikkeitä hahmo tekee ja missä järjestyksessä. Valikkoon tulee 4 laatikkoa, joihin pelaaja voi asettaa haluamansa liikkeitä. Hahmolle tehdään sarja erilaisia liikkeitä, joita pelaaja voi järjestellä laatikoihin (Kuva 1).



KUVA 1. Hahmon liikkeiden valintatoiminto esimerkki

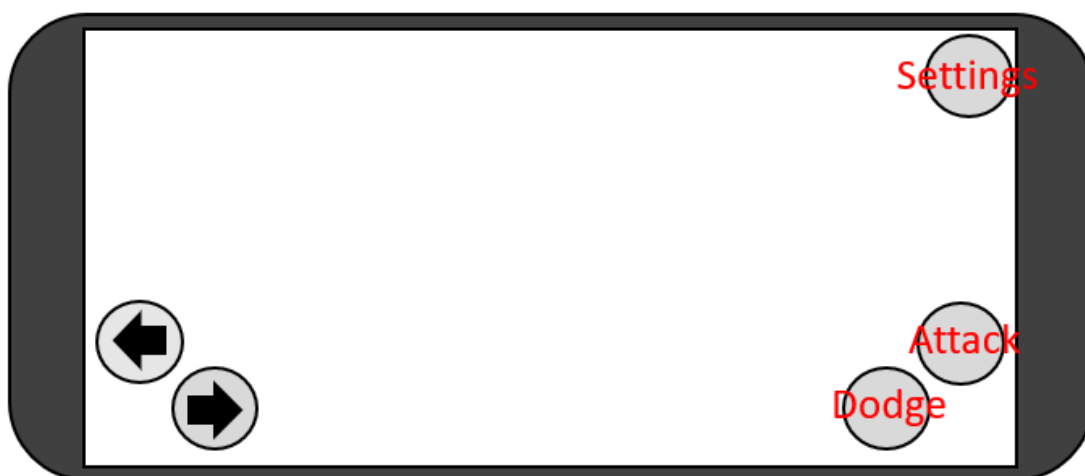
Pelaajan hahmolla on tietty määrä elämäpisteitä, joita se menettää vastustajan osuessa hahmoon iskulla. Vastustaja puolestaan menettää elämäpisteitään pelaajan hahmon osuessa siihen iskulla. Hahmolle tulee nopeita ja hitaita liikkeitä, joiden aiheuttama vahinko vaihtelee liikkeen mukaisesti. Tämä eri liikkeiden aiheuttama vahinko voidaan toteuttaa asettamalla liikkeille eri kertoimia. Näin pelaaja saa syyn vaihdella liikkeitä ja selvittää, millainen liikesarja toimii parhaiten.

4.4 Kehittyminen

Pelaajan voittaessa vastustajan, palkitaan pelaaja kokemuspisteillä, joita pelaaja voi käyttää parantaakseen hahmon kestävyttä ja iskujen voimaa. Kestävyyden parannus toteutetaan kahdella eri vaihtoehdoilla, joista toinen on hahmon elämäpisteet ja toinen hahmon kestävyys, joka vähentää hahmon vastaanottamaa vahinkoa. Iskujen voima toteutetaan voima-arvolla, joka lisää iskujen aiheuttamaa vahinkoa.

4.5 Käyttöliittymä

Pelin käyttöliittymään tulee 4 eri näkymää. Sovelluksen käynnistyessä tulee aloitus näkymä, jossa esitetään pelin nimi. Aloitusnäkömystä pääsee karttanäkymään, jossa pelaaja voi valita pelattavan tason tai muokata asetuksia. Asetusnäkömässä on hahmon liikkeiden valinta, äänen säätäminen sekä hahmon kokemuspisteiden asetus. Pelinäkömässä on hahmon liikutteluun, väistöön ja iskuun tarvittavat painikkeet, sekä mahdollisuus pysäyttää peli ja muokata asetuksia sekä palata karttanäkymään. Pelinäkömän painikkeet voisi sijoittaa esimerkiksi kuvan 2 osoittamalla tavalla.



KUVA 2. Painikkeiden sijoitus pelinäkömässä

Sijoittamalla painikkeet tällä tavoin pelaaja voisi kontrolloida peliä esimerkiksi peukaloilla, pitämällä puhelinta kaksin käsin. Pelaaja pystyisi myös käyttämään etu- ja keskisormea molemmista käsistä pelin kontrollointiin, jos puhelin on las-kettuna jonkin tason päälle.

Käyttöliittymän värimaailma sovitetaan piirretyn grafiikan mukaan ja muotoilussa pyritään käyttämään pyöreitä muotoja.

4.6 Tarina

Pelin tarina pidetään hyvin yksinkertaisena, koska pelissä keskitytään taistelumekaniikkaan ja hahmon sekä vastusten eri liike vaihtoehtoihin. Hahmo voisi esimerkiksi pyrkiä kasvamaan vahvemaksi ja sen johdosta lähtee taistelemaan.

4.7 Hahmot

Pelaajan hahmona on "slime" eli hyytelömäinen pyöreä olento, joka esiintyy useissa eri videopeleissä. Valinnan perusteena on yksinkertaisesti piirtokokemuksen puute ja yksinkertaisella hahmolla säästää aikaa grafiikan luomisessa.

Alustavan suunnitelman mukaan pelissä on aluksi yksi vastustaja, jolla on kolme eri vaihetta. Ensimmäinen pelin taso on metsä, joten vastustajasta tulee isokokoinen orava. Ensimmäisessä vaiheessa vastus on ihan tavallisen näköinen orava, mutta toiseen vaiheeseen siirtyessä se menettää turkkinsa, kynnet ja hampaat kasvavat ja silmät muuttuvat punaisiksi. Kolmannessa ja viimeisessä vaiheessa vastus on maassa ja sen kehosta tulee esiin isokokoinen mato, jolla on päässään hampaiden ympäröimä suu.

4.8 Laajennettavuus

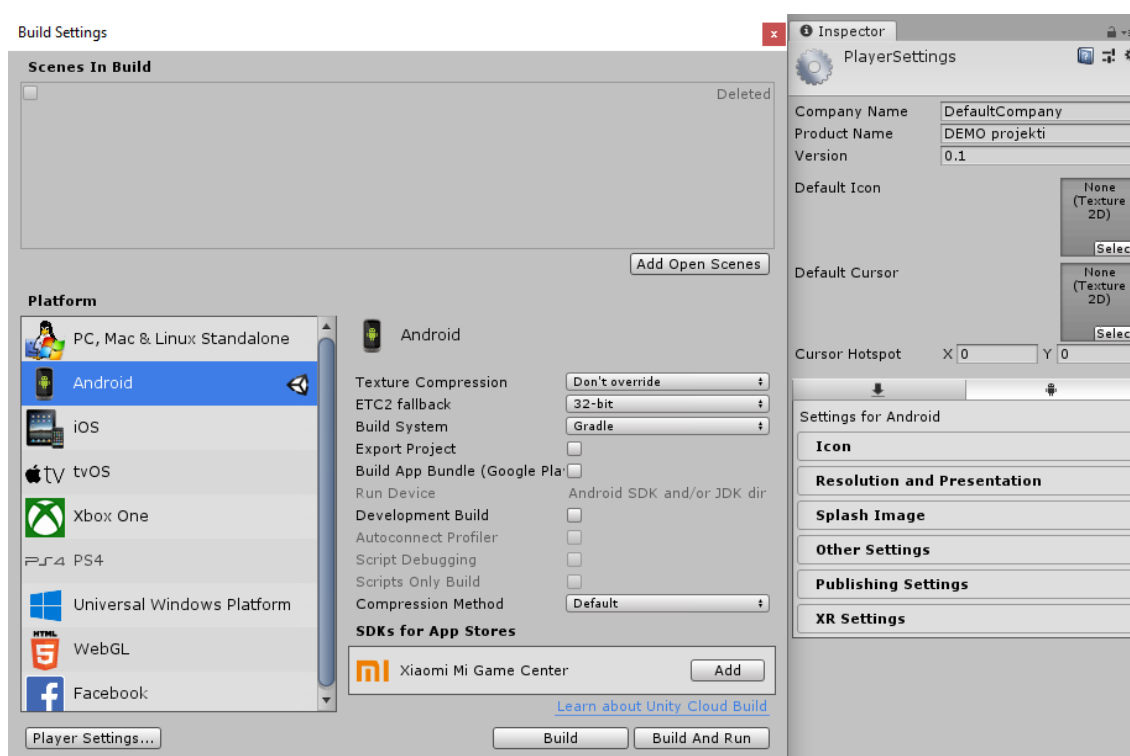
Pelinkehityksessä tulee jo suunnitteluvaiheessa huomioida sen jatkokehitys ja laajennus mahdollisuuksia. Konseptin tasolla tähän pelisovellukseen on mahdollista lisätä lisää tasoja ja vastustajia, mikä laajentaa pelin kokonaissisältöä. Pelaajan hahmolle voi lisätä liikkeitä, mikä lisäisi pelin monimuotoisuutta ja pelaajan vaihtoehtoja.

Ohjelmakoodin ja peliobjektien rakenne tulee toteuttaa siten, että uusia hahmoja, tasoja ja liikkeitä on helppo lisätä. Perusrakenne, jolla Unity – ohjelmistossa tehdään sovelluksia, antaa tähän hyviä mahdollisuuksia. Ohjelmakoodin rakenteesta kerrotaan lisää kappaleessa Toteutus.

5 TOTEUTUS

5.1 Sovellus

Pelisovellusta varten Unity – ohjelmistolla luodaan uusi projekti ja projektin asetuksista yksinkertaisesti vain valitaan, mille järjestelmälle projekti käännetään. Android – järjestelmälle käyntö vaatii valitulle järjestelmäversiolle sopivan SDK – paketin lataamisen. Kuvassa 3 esitetään Build Settings – valikko, josta järjestelmä valitaan sekä Player Settings – valikko, josta kehittäjä voi asettaa sovellukselle kuvakkeet, nimet ja muut asetukset.



KUVA 3. Build Settings ja Player Settings - valikot

5.1.1 Hahmot

Pelaajan hahmo lisätään omana peliobjektinaan ja hahmon käyttämät liikkeet ovat erillisiä animaatioita, jotta liikkeiden muuttaminen onnistuu pelaajalta, täytyy isku – painiketta painellessa tehtävä liikesarja määrittää koodissa siten, että se saa järjestyksen asetuksista. Asetuksien valinta voidaan tallentaa julkiseen luokkaan, josta liikeanimaatioiden järjestys voidaan hakea. Pelaajan hahmoa varten

tehdään erillinen luokka, joka hoitaa elämäpisteiden vähennyksen ja kestävyuden sekä voiman laskemisen.

Vastustajat lisätään myös omina peliohjeiksi, jotta peliä olisi helppo laajentaa. Tehdään vastustaja – hahmon tekoälylle oma luokka, joka hoitaa sen liikesarjan sille annetuilla liikkeillä sekä vastustajan elämäpisteiden vähennyksen. Tätä luokkaa voi siten kutsua kaikkia eri hahmoja liikuttaessa. Hahmoja varten luodaan tietorakenne, josta saadaan niiden elämäpisteet, kestävyys, kokemuspisteet, liikkeet ja liikkeiden aiheuttama vahinko.

5.1.2 Tasot

Tasot toteutetaan yhtenä peli – näkymänä, jonka tausta ja vastustaja vaihdetaan valitun tason mukaan. Pelaajan painaessa kartta – näkymässä tasoa, tallennetaan valinta ja siirrytään peli – näkymään. Peli – näkymään siirryttäessä valinnan perusteella, tasolle asetetaan tausta ja vastustaja – hahmo. Sen jälkeen hahmon tekoäly – luokka hoitaa liikkeiden satunnaisen generoinnin.

5.2 Grafiikka

Pelin hahmon liikkeet, vastustajien liikkeet, taustat ja valikot piirretään ArtRage – ohjelmistolla. Liikkeet piirretään aluksi eri kerroksia hyödyntäen ja liike piirretään kerroksittain, niin että lopputuloksena on sulava liike (Kuva 4).



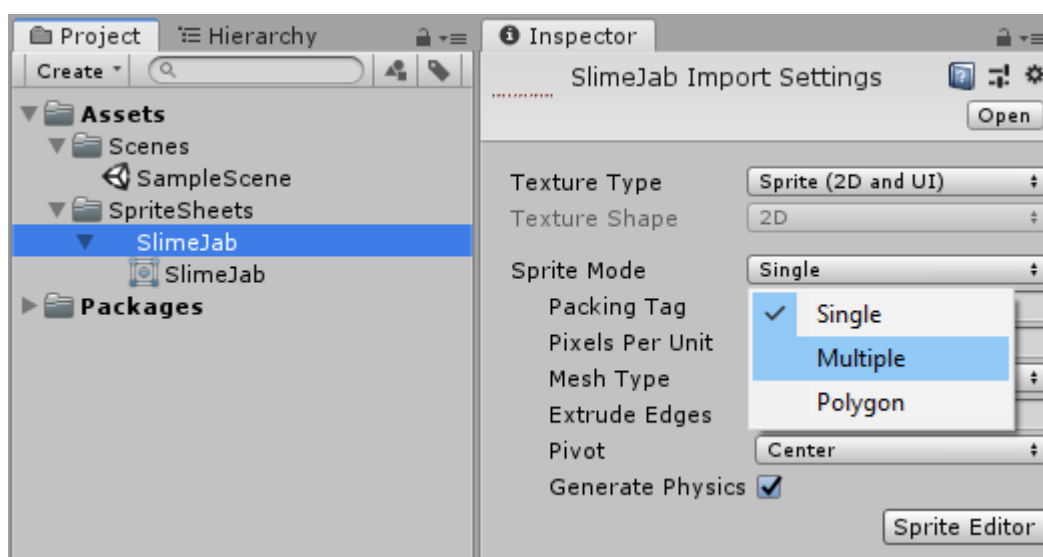
KUVA 4. Pelaajan Slime - hahmon nopea lyönti-isku

Sen jälkeen liike järjestellään kuva-arkeiksi, joita Unity – ohjelmistossa voi animoida. Kuvassa 5 esitetään sama liikesarja kuva-arkkina.



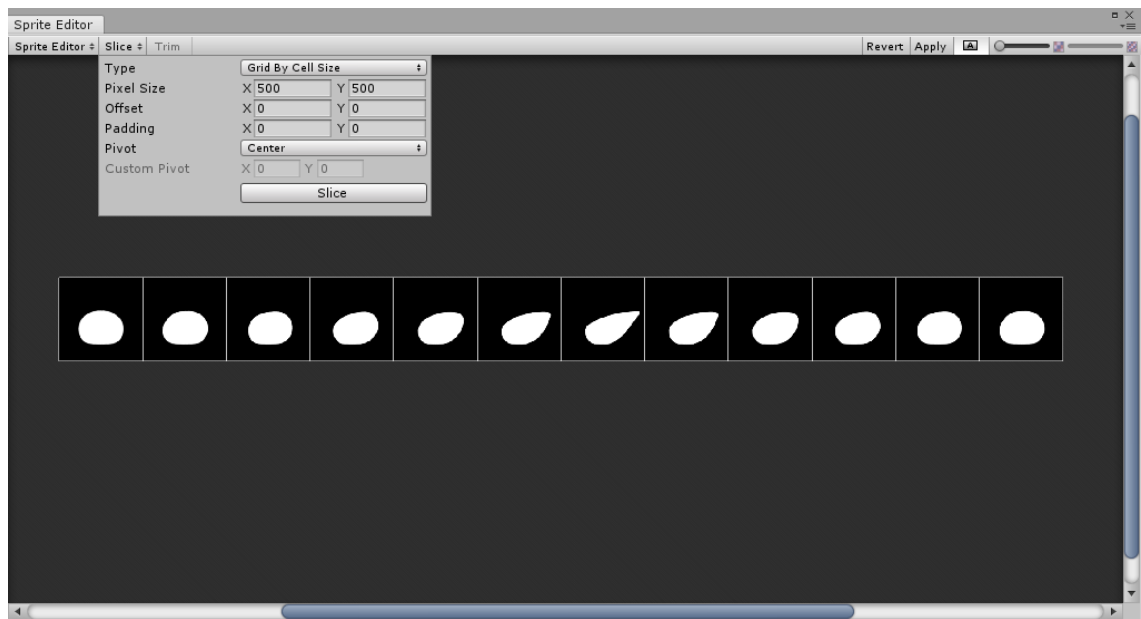
KUVA 5. Pelaajan Slime – hahmon nopea lyönti-isku kuva-arkkina

Kuva-arkin ollessa valmis sen voi muuttaa animaatioksi Unity – ohjelmistossa. Kuva-arkki lisätään Unity projektiin yksinkertaisesti vain raahaamalla se kansioon. Kuva-arkin animoimiseksi täytyy se jakaa ensin osiin, joita voidaan sitten vuoron perään järjestyksessä toistaa. Kuva-arkin osiin jakamisessa on 2 eri vaihetta. Ensin kuva-arkki muutetaan moniosaiseksi (Kuva 6).



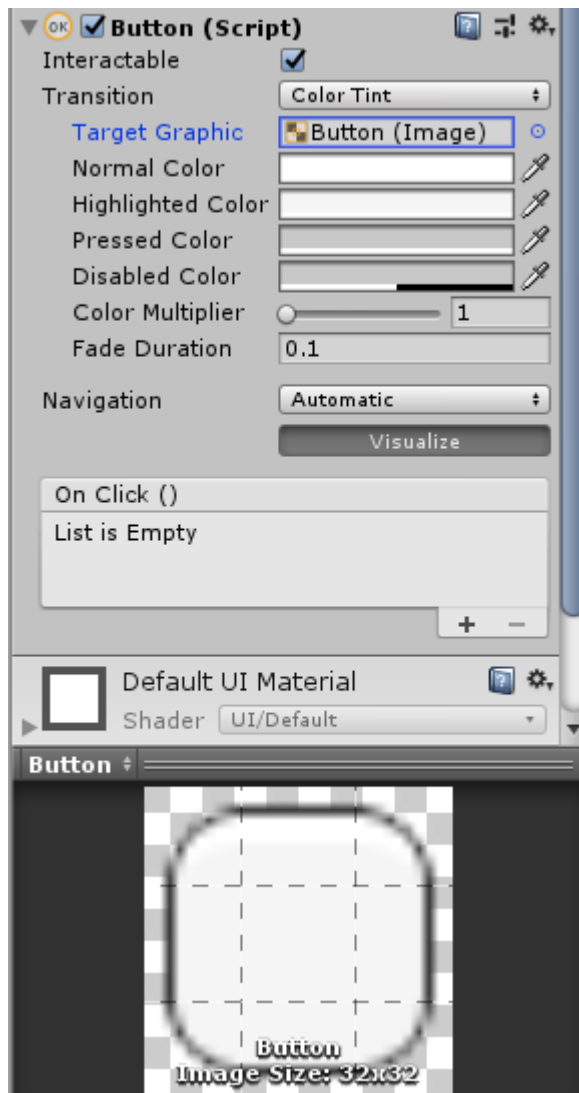
KUVA 6. Kuva-arkin muunto moniosaiseksi

Sen jälkeen kuva paloitellaan osiksi käyttäen Sprite Editor – toimintoa. Kuvassa 7 esitetään Sprite Editor – näkymä ja slice toiminto, jolla kuva-arkki jaetaan osiksi.



KUVA 7. Osiin jaettu kuva-arkki

Pelin taustat voi piirtää isoina yhtenäisinä alueina tai vaihtoehtoisesti luoda pienemmän alueen, joka toistuu uudestaan ja uudestaan sivulle päin liikuttaessa. Taustan voi Unity – ohjelmistossa lisätä yksinkertaisesti kuvana taustalle. Painikkeet lisätään objekteina, joille voi lisätä piirretyn grafiikan oletuksen tilalle (Kuva 8).



KUVA 8. Painikkeen grafiikka - kenttä

5.3 Käyttöliittymä

Sovelluksen käyttöliittymä rakennetaan Unity – ohjelmistossa käyttäen UI – komponentteja, jotka sijoitetaan Canvas – objektin sisään. Kuvassa 9 on esitetty käyttöliittymän ensimmäinen näkymä ilman valmiita grafiikoita. Pelaajan käynnistäessä sovelluksen, tämä näkymä tulee ensimmäisenä.

TITLE

PLAY

KUVA 9. Grafiikaton aloitusnäky

Pelaajan painaessa Play – painiketta siirrytään karttanäkymään, jossa näkyy valittavissa olevat tasot sekä Settings – painike, josta pääsee asetusnäky

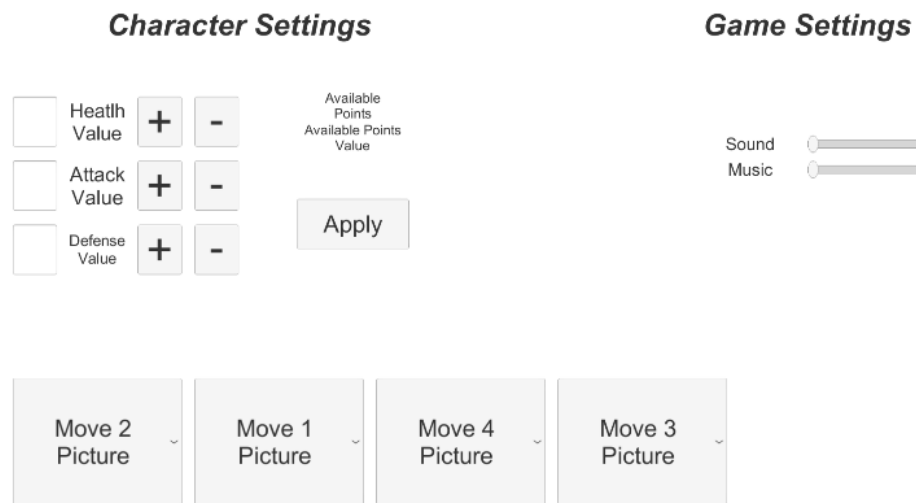
SETTINGS

FIRST
LEVEL

KUVA 10. Karttanäky ilman grafiikoita

Pelaajan painaessa Settings – painiketta siirrytään asetusnäky

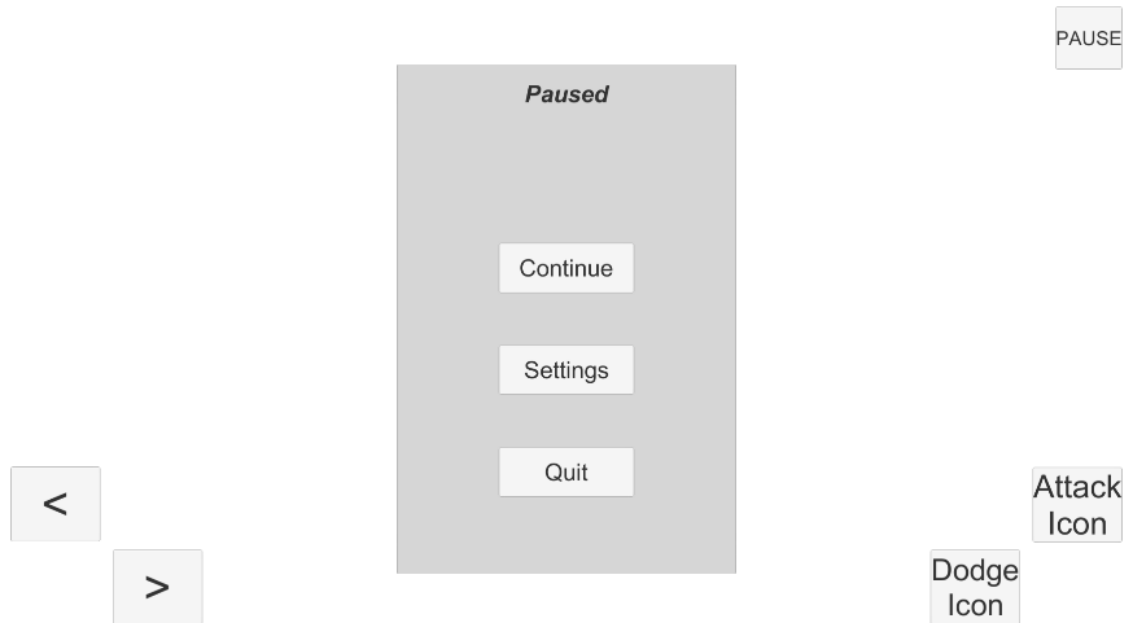
käyttämää perusliikesarjaa sekä muuttaa ääniasetuksia. Kuvassa 11 on asetusnäkö näkymä ilman grafiikoita.



KUVA 11. Asetusnäkö ilman grafiikoita

Pelaaja voi asettaa saatavilla olevat kokemuspisteet haluamallaan tavalla, käyttäen asetusnäkömän + ja - - painikkeita, jotka sijaitsevat elämäpisteiden, voiman ja kestävyden arvojen vieressä. Arvojen toisella puolella on niitä kuvaava pieni kuvake. Pelaajan asetettua kokemuspisteet saa ne tallennettua painamalla Apply – painiketta. Liikesarjaa voi muokata painamalla alareunassa näkyviä dropdown – painikkeita, joista pelaaja voi valita haluamansa liikkeet.

Näkymien välillä siirrytään Android – laitteen natiivin paluu painikkeen avulla. Kuvassa 12 on pelinäkömä, johon pelaaja pääsee painaessaan tason kuvaketta karttanäkömässä.



KUVA 12. Pelinäkö ja keskeytysvalikko

Pelinäkymässä pelaajan käyttämät liikuttelupainikkeet ovat vasemmassa alareunassa. Isku – painike ja väistö – painike sijaitsevat oikeassa alareunassa ja painikeihin tulee niitä kuvaavat kuvakkeet. Keskeytys – painike sijaitsee oikeassa yläkulmassa, jota painamalla kuvan 12 keskellä näkymä keskeytysvalikko ilmestyy. Keskeytysvalikosta pelaaja pääsee asetusnäköön, voi jatkaa peliä tai lopettaa pelin, jolloin pelaaja palaa karttanäkymään.

5.4 Testaus

Pelisovelluksen toimintoja testataan sen kehitysvaiheessa suoraan kehitysympäristössä, koska se on Unity – ohjelmistolla mahdollista. Pelin vaikeusastetta ja tasoja testataan muutamalla lähipiirin henkilöllä.

Animaatioiden ja grafiikan testaus suoritetaan Unity – ohjelmistossa ja mahdolliset muutokset tehdään ArtRage – ohjelmistossa, jos käytössä olisi Adobe Photoshop – ohjelmisto, testauksen voisi tehdä suoraan sillä.

6 LOPPUTULOKSET JA JATKOKEHITYS

6.1 Pelisovellus

Pelisovellus ei ole vielä valmis, mutta mekaniikka ja ensimmäinen taso sekä käyttöliittymä on jo suunniteltu. Käyttöliittymä on grafiikoita lukuun ottamatta valmis. Ohjelmakoodia on myös suunniteltu ja se toteutetaan grafiikan ollessa valmis. Grafiikoiden valmistuttua peli julkaistaan Google Play – kaupassa.

6.2 Julkaisu

Pelisovellus julkaistaan valmistuttuaan Google Play – kaupassa. Julkaisua varten sovelluksen tulee täyttää Googlen vaatimukset. Googlen Launch checklist – ohje määrittää muutamia ehtoja kehittäjille, jotka haluavat julkaista sovelluksia Google Play – kaupassa. Sovelluskehittäjän täytyy luoda kehittäjä – tunnukset ja mikäli kehittäjä aikoo julkaista maksullista sisältöä, tulee hänen luoda myös kaupallinen tili. Luodessaan kehittäjä – tilin sitoutuu noudattamaan Google kehittäjien sääntöjä. Näiden lisäksi julkaistaville sovelluksille on laatuvaatimuksia. (Launch checklist n.d.)

Googlen kehittäjien säännöt käsittävät muun muassa seuraavia asioita (Luodaan yhdessä maailman luotettavin sovellusten ja pelien lähde n.d.):

- rajoitettu sisältö
- toisena esiintyminen ja immateriaaliomaisuus
- tietosuoja, turvallisuus ja petokset
- roskasisältö ja vähimmäistason toimivuus

Googlen laatuvaatimukset sisältävät sovelluksen rakennetta, ulkoasua ja suorituskykyä koskevia ohjeita. Erillisiä ohjeita televisio ja tablet – sovelluksia varten sekä myös puettaville laitteille ja autoille tehdyille sovelluksille. Google myös tarjoaa vinkkejä houkutteleville peleille ja pelien ulkoasulle. (Quality Guidelines n.d.)

6.3 Jatkokehitys

Tämän opinnäytetyön jälkeen sovellus viimeistellään, siihen lisätään äänet ja julkaistaan Google Play – kauppaan ilmaisena. Sen jälkeen mielenkiinnon mukaan peliin voi lisätä hahmoja ja tasoja. Peliin voisi myös lisätä Google Play – ominaisuuksia, kuten kirjautumisen, saavutukset ja tallennuksen Google Play – tilille. Peliin voi lisätä pelin sisäisiä tehtäviä ja saavutuksia, kuten useissa mobiilipeleissä on tapana.

7 POHDINTA

Opinnäytetyössä saatiin aikaiseksi suunnitelma pelikonseptista ja sen mekaanisesta toteuttamisesta sekä pelin käyttöliittymä luotiin Unity -ohjelmistolla. Pelin toteutus on vielä kesken eikä se ole vielä julkaistavassa kunnossa, koska suurin osa peliin tarvittavista grafiikoista puuttuu. Työssä esitetyt ratkaisut toiminnoille voi myös toteuttaa muilla tavoilla. Työssä käytettyjä työkaluja on myös useita vastaavia, joten niiden sijasta voi myös käyttää muita työkaluja kehitykseen. Valitut työkalut ovat kuitenkin tarkoitukseen sopivia.

Yksittäiselle kehittäjälle kenties suurimpia haasteita on pelin grafiikat. Työmäärä, mitä sopivan teeman ja piirtotyylin tutkimiseen ja sen jälkeen grafiikoiden piirtämiseen menee, on huomattavan iso. Mitä enemmän pelisovellukseen haluaa sisältöä, sitä enemmän se vaatii grafiikkaa. Erinomaisesta piirtotaidosta ja harrastuneisuudesta piirtämiselle olisi ehdottomasti ollut työtä tehdessä hyötyä, pelkkä lahjakkuus ei riitä. Ohjelmointitaidoista on kuitenkin ehdottomasti hyötyä pelinkehityksessä, koska pelilogiikka ja -mekaniikka vaativat kokemusta ohjelmoinnista tai kaikkia ominaisuuksia ei pysty toteuttamaan. Peliyrityksissä käytetään eri tekijöitä kumpaankin osaamiseen.

Kokonaisuutena työstä sai kokemusta pelinkehityksestä ja suunnittelusta. Suunnitteluvaiheessa täytyy miettiä monia eri asioita ja ottaa huomioon käyttäjäkokemus. Alkuperäisessä suunnitelmassa ei pysytty, sovellus ei ole valmis ja julkaistu Google Play – kaupassa.

LÄHTEET

Ambient Design Ltd. 2019. ArtRage. Luettu 28.05.2019 <https://www.art-rage.com/>

Audacity. 2019. Audacity. Luettu 28.05.2019. <https://www.audacityteam.org/about/>

Google Developers. N.d. Android game development. Luettu 28.05.2019. <https://developer.android.com/games>

Google Developers. N.d. Android Platform. Luettu 28.05.2019. <https://developer.android.com/about>

Google Developers. N.d. Launch checklist. Luettu 28.05.2019. <https://developer.android.com/distribute/best-practices/launch/launch-checklist>

Google Developers. N.d. Platform Architecture. Luettu 28.05.2019. <https://developer.android.com/guide/platform>

Google Developers. N.d. Quality guidelines. Luettu 28.05.2019. <https://developer.android.com/docs/quality-guidelines/index.html>

Google Play. N.d. Luodaan yhdessä maailman luotettavin sovellusten ja pelien lähde. Luettu 28.05.2019. https://play.google.com/about/developer-content-policy/#!?modal_active=none

Microsoft. 2019. Visual Studio. Luettu 28.05.2019. <https://visualstudio.microsoft.com/>

Microsoft. 2019. Visual Studio 2019. Luettu 28.05.2019. <https://visualstudio.microsoft.com/vs/>

Unity Technologies. 2019. Editor. Luettu 28.05.2019. <https://unity3d.com/unity/editor>

Unity Technologies. 2019. Unity. Luettu 28.05.2019. <https://unity3d.com/unity>