

## Mobiilipelisovellus: Streamer Simulator

Julius Pusila



<b>Tekijä(t)</b> Julius Pusila	
<b>Koulutusohjelma</b> Tietojenkäsittely	
<b>Opinnäytetyön otsikko</b> Mobiilipelisovellus: Streamer Simulator	<b>Sivu- ja liitesivumäärä</b> 31 + 1
<p>Tämä toiminnallinen opinnäytetyö tehtiin oman yrityksen näkyvyyden edistämiseksi. Kokeukseni mobiilipelien kehityksestä ennen työn aloittamista oli yhden Google Play Storessa julkaistun mobiilipelisovelluksen verran, joka toteutettiin Unityllä. Tässä opinnäytetyössä valmistuva fiktiivinen suoratoistosimulaatio "Streamer Sim Tycoon"-mobiilipelisovellus on ainutlaatuinen ja se tarjoaa mukavaa sekä rentoa ajanvietettä kaikenikäisille. Pelissä suoratoistetaan pelaajan hahmolla sisältöä tietokoneelta, kerätään seuraajia, kehitetään puitteita ja nouseaan suoratoistajien huipulle.</p> <p>Työn tavoitteena oli julkaista suorituskyylyltään hyvä sekä suunnitellut ominaisuudet sisältävä mobiilipelisovellus Google Play Storessa, joka olisi täysin pelattava sekä helposti jatkokehittävissä. Itse opinnäytetyön raportin tavoitteena oli myös uusien pelikehittäjien auttaminen mobiilipelisovelluksen suunnittelussa sekä kehityksessä.</p> <p>Mobiilipelisovellusta lähdettiin työstämään lähes tyhjästä. Ainoastaan pelinsisäinen musiikki ja äänet sekä muutamat kuvakkeet olivat ulkoistettu. Google Play Developer käyttäjä oli jo olemassa. Mobiilipelisovellus toteutettiin Unityllä ja se sisältää valinnaisia videomainoksia. Peli sisältää pikseligrafiikkaa, joka on toteutettu Paint.NET ohjelmalla.</p>	
<b>Asiasanat</b> Mobiilipelit, pelinkehitys, optimointi	

# Sisällys

1	Johdanto .....	1
2	Simulaatiopelit ja niiden markkinatilanne .....	2
3	Toteutusmenetelmät .....	5
3.1	Pelin kulku .....	5
3.2	Pelimoottori .....	6
3.3	Grafiikka .....	7
3.4	Käyttöliittymä .....	8
3.5	Äänet .....	8
4	Ratkaisut ja toteutus .....	9
4.1	Projektin aloitus Unityssä .....	9
4.2	Käyttöliittymä .....	10
4.3	Katsoja- ja keskustelujärjestelmä .....	12
4.4	Taitojärjestelmä .....	13
4.5	Pelivaluuttajärjestelmä .....	16
4.6	Satunnaistaminen ja tapahtumajärjestelmä .....	19
4.7	Aloittelijan opastus .....	20
4.8	Tallennusjärjestelmä .....	21
4.9	Äänijärjestelmä .....	22
5	Pelisovelluksen julkaiseminen .....	24
5.1	Optimointi .....	24
5.2	Valmistelu ja julkaiseminen Google Play Storessa .....	25
6	Pohdinta .....	27
6.1	Projektin onnistuminen .....	27
6.2	Jatkokehitys .....	28
	Lähteet .....	29
	Liitteet .....	32
	Liite 1. Tehtäväluettelo .....	32

# 1 Johdanto

Tässä opinnäytetyössä valmistuvassa ja ”Streamer Simulator”-projektinimellä kulkevassa mobiilipelissä pelaaja ohjaa hahmoaan joka suoratoistaa sisältöä tietokoneelta, kerää katsojia sekä seuraajia päivä kerrallaan. Pelaaja kehittää laitteistoaan, hahmoaan sekä muita puitteitaan suoratoistokanavansa tuottamalla varoilla ja kerää tasoja eri taitoihin, jotka auttavat pelaajaa saavuttamaan yhä enemmän katsojia, seuraajia, sponsoritarjouksia ja muita pelissä eteenpäin vieviä parannuksia. Satunnaisuus on suuressa osassa pelin kulkua ja sen sisältöä.

”Streamer Simulator”-mobiilipelisovellus tehdään omalle yritykselleni, joka on julkaissut jo yhden Unityn itsenäisen opiskelun tuloksena syntyneen mobiilipelisovelluksen Google Play Storessa ja saanut tälle lähes 2000 latauskertaa. Yritykseltäni löytyy omat internetsivut sekä myös toiminimi tulevaisuutta varten. Opinnäytetyön tavoitteena on tuoda yritykselleni lisää näkyvyyttä alalla, suurentaa asiakaskuntaa sekä mahdollisesti saada mielenkiintoisia työtarjouksia.

Tammikuussa 2018 aloitetun opinnäytetyön tavoitteena on suunnitella, kehittää, optimoida, sekä julkaista mobiilipelisovellus Google Play Storessa huuhtikuun 2018 alkuun mennessä. Mobiilipelisovelluksen tulee sisältää kaikki tähän suunnitellut ominaisuudet, olla hyvä suorituskyvyltään sekä tarjota mukavaa ja rentoa ajanvietettä kaikenikäisille. Opinnäytetyön raportti tarjoaa ideoita uusille pelikehittäjille mobiilipelisovelluksen suunnitteluun ja kehitykseen sekä auttaa kehitysympäristön valinnoissa. Uuden mobiilipelisovelluksen kehityksen tavoitteena on oppia lisää mobiilipelikehityksestä ja käyttää jo opittua hyödyksi uuden mobiilipelisovelluksen pohjaa tehdessä, jotta jatkokehitys olisi helppoa ja yksinkertaista.

Opinnäytetyössä valmistuva mobiilipelisovellus on ilmainen ja se tehdään vain englannin kielelle. Mobiilipelisovellus ei myöskään sisällä Google Playn palveluita, sovelluksensisäisiä ostoja, eikä raportissa käydä läpi mobiilipelien markkinointia eikä itse koodin kirjoittamista. Opinnäytetyö ei myöskään ole opas kehitysympäristön pystyttämiseen, Googlen palveluihin eikä tarvittavien käyttäjien tai lisenssien hankintaan. ”Streamer Simulator”-mobiilipeli on todella yksinkertaistettu ja epärealistinen 2D-peli, vaikka kyseessä onkin simulaatiopeli. Mobiilipelillä ei ole lopetusta, vaan peli jatkuu niin kauan kuin käyttäjä jaksaa tätä pelata.

Tässä opinnäytetyön raportissa lähdemme työstämään projektia toteutusmenetelmistä aloittaen ja tämän jälkeen ominaisuus kerrallaan aikajärjestyksessä. Tässä raportissa py-

rin käyttämään hyödyksi ulkopuolisista lähteistä löytyviä asiatekstejä valintojen tekemiseen koko prosessin ajan.

## **2 Simulaatiopelit ja niiden markkinatilanne**

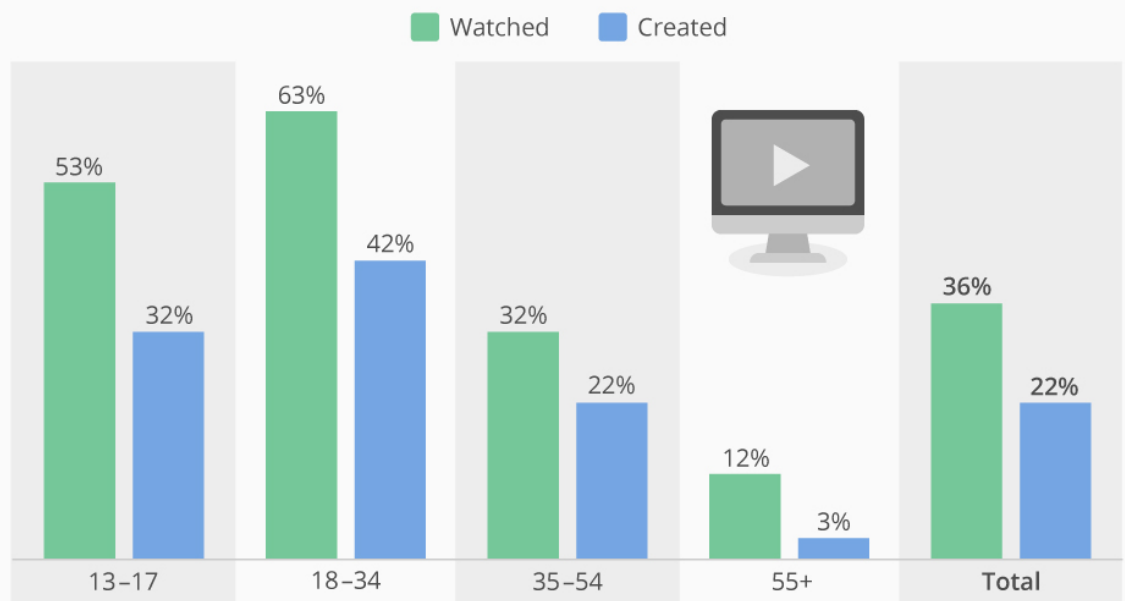
Simulaatiopeleissä simuloidaan eli jäljitellään todellisuuden tai virtuaalitodellisuuden tapahtumia. Simulaatiopelit voivat olla realistisia tai epärealistisia. ”Streamer Simulator”-mobiilipelissä simuloidaan nousevan suoratoistokanavan ylläpitäjää todella yksinkertaistelta sekä epärealistiselta kantilta. Simulaatiopelit ovat laajoja kokonaisuuksia ja tämän vuoksi projektissa valmistetaan helposti jatkokehitettävä pohja, johon ominaisuuksia sekä satunnaisia tapahtumia voidaan lisätä vaivattomasti. Pelisovelluksen käyttäjäkokemuksen sekä järjestelmien tasapainottamiseksi se julkaistaan beta-versiona, jonka aikana on tarkoitus kerätä palautetta jatkokehitystä varten. Raportin lopusta löytyvästä pohdinta osiossa käydään läpi myös jatkokehityksen mahdollisuuksia.

Syy ”Streamer Simulator”-mobiilipeliprojektin käynnistämiseksi on se, että markkinoilla ei tällä hetkellä ole tämän tyyppistä mobiilipelisovellusta ja monet mobiilipeleistä ovat suunniteltu tuottamaan maksimaalinen voitto rahallisesti. Peliin pyritään toteuttamaan ennemminkin hyvä käyttökokemus ja se sisältää ainoastaan valinnaisia videomainoksia. ”Streamer Simulator”-mobiilipeli ei tule sisältämään pakollisia mainoksia eikä muita odotusaikoja käyttökokemuksen parantamiseksi. Projektissa valmistuu siis peli, jota voit pelata milloin vain, missä vain ja mitään rajoittamatta.

Simulaatiopelin kehityksen syynä on myös se, että niitä kehittäessä voidaan keskittyä toteuttamaan yhdessä hyvin toimivia ja skaalautuvia järjestelmiä monimutkaisemman sisällön ja pelimekanismien sijasta. Peli pelataan kokonaan valikoissa, jossa valintamahdollisuudet ovat annettu pelaajalle. ”Streamer Simulator”-mobiilipeli ei myöskään tarvitse paljon grafiikkaa, erikoistehosteita eikä tarinan tai juonen suunnittelua, joten simulaatiopeli-genre ja tämän tyyppisen pelin kehitys sopii loistavasti omiin taitoihini sekä opinnäytetyöni aiheeksi.

## Who's Watching & Creating Live Streaming Video in the U.S.

Internet users who watched or created live streaming video on social media

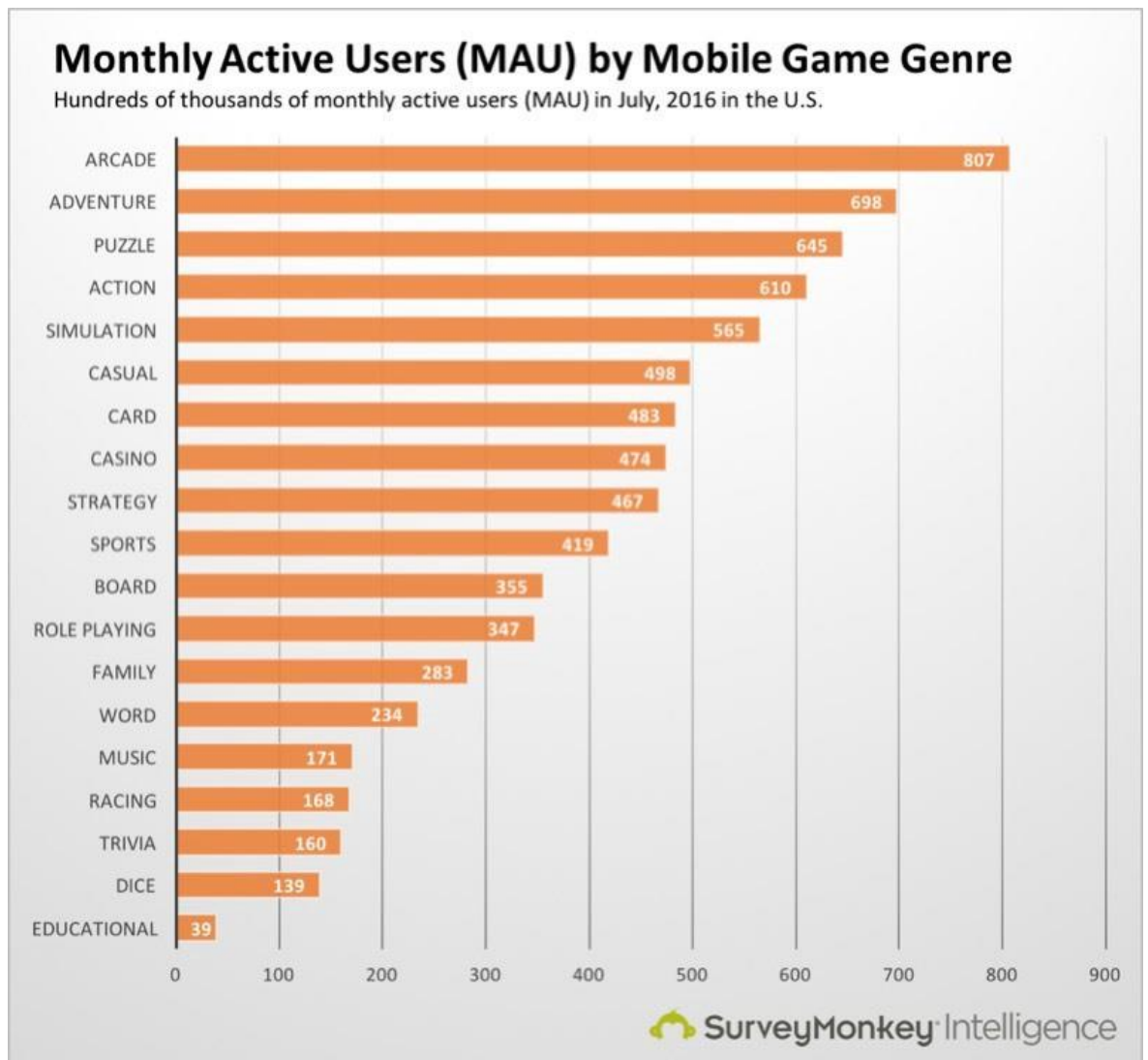


As of January 2017  
Sources: eMarketer, UBS

statista

Kuva 1. Kuinka moni katsoo ja luo suoratoistolähetyksiä Yhdysvalloissa (Patel 2017)

Potentiaalisia käyttäjiä voidaan ajatella olevan todella paljon. Noin 63% 18-34 vuotiaista katsoo ja 42% luo suoratoistolähetyksiä säännöllisesti (Patel 2017).



Kuva 2. Aktiiviset käyttäjät eri mobiilipeligenreille kuukausittain Yhdysvalloissa (Sonders 2016)

Suoratoistolähetysten suuren suosion lisäksi myös simulaatio-mobiilipeligenre on hyvin suosittu. Simulaatiomobiilipeleillä on neljänneksi eniten latauksia ja viidenneksi eniten aktiivisia käyttäjiä kuukausittain. Simulaatiomobiilipeleillä on pelaajia kuukausittain jopa 56,5 miljoonaa. (Sonders 2016.)

### 3 Toteutusmenetelmät

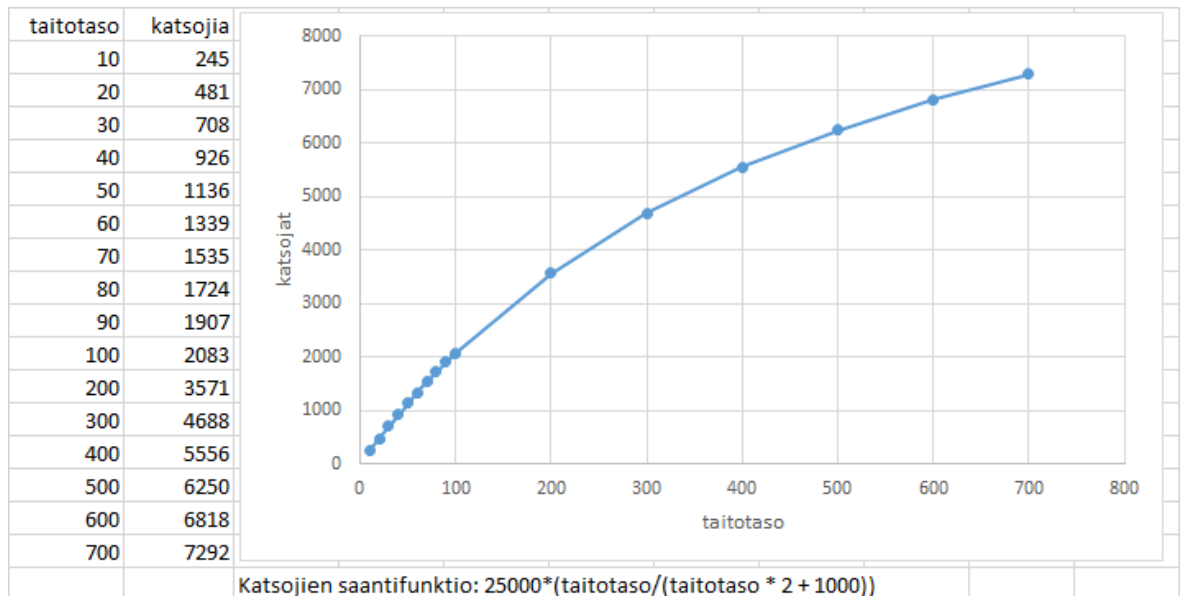
Mobiilipelisovellusta lähdettiin työstämään lähes tyhjästä. Ainoastaan pelinsisäinen musiikki ja äänet sekä muutamat kuvakkeet oli ulkoistettu. Mobiilipelisovelluksen pelimoottoon, grafiikkaan, käyttöliittymään sekä ääniin liittyvät valinnat selvitetään tässä osiossa.

#### 3.1 Pelin kulku

Pelin ideana on suoratoistaa sisältöä tietokoneelta ja päivittää puitteita päivä kerrallaan, jotta pelaaja menestyisi ja saisi yhä enemmän katsojia. Pelin kulku on seuraava; herättään uuteen päivään, päivitetään mahdollisesti puitteita, aloitetaan päivän suoratoistolähetys, suoratoistolähetyksen aikana reagoidaan tämän aikaiseen katsojien keskusteluun ja tietyn ajan päästä lopetetaan lähetys ja saadaan päiväraportti luettavaksi. Tämän jälkeen seuraa yö ja alkaa uusi päivä, jolloin sama sykli jatkuu.

Päivien aikana tapahtuu satunnaisia tapahtumia, jotka ovat suuressa osassa pelin kulkua ja sisältöä. Tällaisia tapahtumia voivat olla esimerkiksi sponsori- ja muut tarjoukset, katsojien lahjoitukset tai erilaiset tapahtumat, joihin pelaaja voi osallistua. Valinnaiset videomainoksien katselut ovat osana näitä tapahtumia, mutta eivät niin satunnaisina muihin tapahtumiin verrattuna. Satunnaistamisen ansiosta pelin kehityksessä voidaan keskittyä luomaan järjestelmiä sisällön sijaan. Taitavasti suunnitellut järjestelmät voivat skaalata uskottoman hyvin tuottaen pelaajille valtavia määriä sisältöä, johon uppoutua. (Wicksteed 2015.)

Suurin osa pelin järjestelmien ominaisuuksista, kuten katsojien ja seuraajien saamisesta perustuu taitojärjestelmään. Pelaajan hahmolla on eri taitoja, joiden tasot ovat numeraalisesti esitetty. Pelaaja kehittää eri taitotasoja pelissä päivittämällä puitteitaan ja osallistumalla kurseille. Pelaaja voi osallistua yhteen kurssiin kerrallaan pelin valikosta, joiden päivittäin muuttuva lista on satunnaisesti luotu. Pelaaja voi myös päivittää peruspuitteitaan päivitysvalikosta, joita ovat hahmon ulkoasu, tietokone, näyttö ja asunto. Pelin satunnaiset tapahtumat saattavat sisältää muiden puitteiden parantamista, pelaaja saattaa saada esimerkiksi tarjouksen ostaa kissan, joka edistää katsojien kiinnostusta pelaajan tuleviin suoratoistolähetyksiin.



Kuva 3. Katsojien saantifunktion toiminta

Vaikka kaikki perustuvat taitojärjestelmän tasoihin, ne eivät kuitenkaan toimi vaikutuksensa suoranaishana kertoimena. Ylläolevasta kuvasta huomataan, että aikaisemmat tasot vaikuttavat tulokseen suhteessa enemmän kuin suuremmat tasot ja samanlaisia funktioita käytetään myös muissa järjestelmissä, kuten seuraajien saannissa ja niiden osallistumisesta lähetykseen. Tällä funktiolla pyritään rajoittamaan pelin edetessä suuresta rahavirrasta aiheutuvaa nopeaa kehitystä ja siirtämään kehitysnopeutta pelin alkuun kun pelaajan tulot ovat pienet.

”Streamer Simulator”-mobiilipelisovellus pyrkii antamaan välitöntä palautetta pelaajan toiminnasta ja ohjaamaan pelaajaa tavoitteiden saavuttamisessa. Palaute edistymisestä voi edistää sitoutumista peliin entisestään ja mahdollistaa saavutuksia. Tämä tarkoittaa, että pelaajat, jotka saavat palautetta haluavat pelata enemmän. (Baron 2012.)

### 3.2 Pelimoottori

Pelien kehitys tyhjästä on todella hidasta ja vaatii erityistä taitoa. Tämän vuoksi suuremmatkin pelinkehittäjät käyttävät apunaan eri pelimoottori-ohjelmia, jotka nopeuttavat kehitystä huomattavasti. Pelinkehittäjät kohdistavat pelinsä yleensä useammalle alustalle, joita ovat mm. iOS, Android, eri pelikonsolit ja tietokoneet. Pelimoottorit tarjoavat ratkaisun tähän ongelmaan ja tämän lisäksi muita valmiita perustoiminnallisuuksia muun muassa grafiikan renderöintiin, fysiikkamoottoriin, ääniin, animaatioihin, muistinhallintaan, valaistukseen sekä varjostukseen liittyen ja näin ollen pelituotteen nopeampaan kehitykseen sekä julkaisuun (Enger 2013). Pelimoottoreista lähes kaikki ovat täysin ilmaisia ladata ja asentaa, mutta ne saattavat sisältää lisenssimaksuja, rojalteja tai molempia.

Suosittuja pelimoottoreita ovat mm. Unity, Unreal Engine ja CryEngine. Unitystä on saatavilla Personal-, Plus- ja Pro- lisenssit. Unity Personal- lisenssi on ilmainen, mutta mikäli yrityksesi tienaa vuodessa yli 100 000 dollaria bruttotulona, olet velvollinen maksamaan 35 dollaria kuukaudessa Unity Plus-lisenssistä. Mikäli tienaat yli 200 000 dollaria, olet velvollinen maksamaan 125 dollaria Unity Pro-lisenssistä (Unity Store 2018). Unityllä on tehty 34% tuhannesta suosituimmasta ilmaismobiilipelistä ja Unity onkin monien mobiilipelisovelluskehittäjien suosiossa. Unityllä on tehty monia hittipelejä, kuten Pokemon Go ja Angry Birds 2 (ElHady 2017). Unreal Engine on ilmainen, mutta siitä joutuu maksamaan 5% rojalteja yli 3000 dollarin tuloista peliä kohden joka vuosineljännes. Unreal Engine on pääasiallisesti tarkoitettu suuriin, vaativiin ja monimutkaisempiin peleihin, joihin tarvitaan vahvan 3D pelimoottorin (ElHady 2017). Unreal Enginen tavoin myös CryEngine on sopiva ratkaisu näyttäviin ja vaativiin 3D peleihin. CryEngine on myös ilmainen, mutta joudut Unreal Enginen tavoin maksamaan rojalteja. CryEnginen rojalitit ovat 5% projektisi tuotosta 5000 dollarin taikka 5000 euron vuositulorajan jälkeen (CryEngine 2018).

Unityn 2D- ominaisuuksien, helppokäyttöisyyden, hinnoittelun sekä aiemman kokemuksen perusteella ”Streamer Simulator”-mobiilipelisovellusta lähdettiin työstämään uusimalla Unity 2017-versiolla. Unity on yksi suosituimmista pelimoottoreista ja sen osaaminen on erittäin suotavaa alan työmarkkinoita ajatellen. Sopivia pelimoottoreita tämän kaltaiseen projektiin on monia. Unity mahdollistaa myös alun perin Androidille tarkoitettun mobiilipelin rakentamisen muillekin laitealustoille, kuten iOS:ille taikka tietokoneelle.

### **3.3 Grafiikka**

Pelien graafinen tyyli on merkittävä osa pelaajan kokemusta pelistä ja yksi osatekijöistä, joka tekee pelistä mielenkiintoisen sekä hauskan pelata. Graafinen tyyli on pelin esittelynäkökulma, jonka pelaajat näkevät (Lee, Gee & Dolah 2016).

”Streamer Simulator”-mobiilipeli sisältää itsetehtyä pikseligrafiikkaa ja ulkoistettuja kuvakkeita. Pikseligrafiikka oli alun perin ratkaisu huonoon laitteistoon sekä pieneen näyttöön. Pikseli on pienin rakennuspalikka siitä, mitä näytössä näkyy, mutta nykypäivän suurissa näytöissä yksi pelin pikseli voi vastata näytön koosta riippuen useampaa pikseliä. Nykypäivän pikseligrafiikan omaavat pelit eivät välttämättä yritä herättää nostalgisia tunteita eivätkä tällaiset pikselipelit näyttävyydeltään koskaan ikäänny. (Byford 2014.) Pikseligrafiikan lisäksi pelin käyttöliittymä sisältää Googlen materiaalikuvakkeita toimimaan visuaalisena kielenä ja näin ollen helpottamaan käyttöliittymän käyttöä sekä parantamaan pelin visuaalista olemusta, ymmärrettävyyttä sekä opittavuutta. Peli sisältää myös hieman muu-

ta ulkoistettua ei-pikseligrafiikkaa taitojen puutteen, ilmaisten käyttöoikeuksien sekä nopeamman kehityksen vuoksi.

Pikseligrafiikkaa on helppo lähestyä ja se pysyy terävänä näytössä kuin näytössä eikä se vaadi laitteelta suurta suorituskykyä. Tämän, aiemman kokemuksen sekä graafisten taitojen vuoksi ”Streamer Simulator”-mobiilipeli sisältää juuri tätä grafiikkaa. Pelin kaikki pikseligrafiikka toteutetaan yksinkertaisella Paint.NET ohjelmalla tämän helppokäyttöisyyden, yksinkertaisen grafiikan ja aiemman kokemuksen perusteella. Pelin käyttöliittymä sisältää myös ulkoistettuja kuvakkeita. Googlen Material Design Icons kuvakkeiden helposti ymmärrettävä muotoilu mahdollistaa kaikkien käyttäjien kykyä navigoida, ymmärtää ja käyttää käyttöliittymää onnistuneesti (Material Design 2018).

### **3.4 Käyttöliittymä**

Pelin käyttöliittymän tavoite on olla yksinkertainen sekä helppokäyttöinen. Mobiililaitteiden pienten näyttöjen vuoksi tilaa on rajoitetusti, joten näkyvillä on tavoitteena olla vain välttämätön määrä sisältöä. Aiemmin mainitut Google Material Design Icons kuvakkeet ajavat käyttöliittymän painikkeiden helppokäyttöisyyttä, ymmärrettävyyttä sekä opittavuutta. Käyttöliittymän fonttina toimii ulkoistettu pelin pikseligrafiikkaan sopiva ilmainen fontti.

”Streamer Simulator”-mobiilipeli on ylhäältäpäin kuvattu. Pelin peruseriaatteisiin kuuluu pelaajan oma hallinta valinnoistaan ja tekemisistään. Ihmisten tarve hallita sekä valinnanvapaus ovat yksi pelien suosion tärkeimpiä tekijöitä (Shvebish 2014).

### **3.5 Äänet**

Pelin musiikki sekä äänet ovat ulkoistettu taitojen puutteen sekä nopeamman julkaisuvaiheeseen pääsyn vuoksi. Käytössä olevan musiikin sekä äänien käyttöoikeudet ovat Unityn Asset Storesta aiemmin ostettuja.

Unity Asset Store-kauppa on jatkuvasti kasvava kirjasto, josta löytyy ilmaisia ja kaupallisia asetteja eli valmiita osia, joita Unity Technologies ja yhteisön jäsenet ovat luoneet. Saatavilla on monenlaisia asetteja, jotka kattavat kaiken tekstuureista, malleista, äänistä ja animaatioista kokonaisen projektin esimerkkeihin, opetusohjelmiin ja Unity-ohjelman laajennuksiin. (Unity Technologies 2018a.)

## 4 Ratkaisut ja toteutus

Projektin työvaiheet jaetaan kolmeen suurempaan osaan jotka ovat konseptin ja konseptin toteutuksen suunnittelu, grafiikan teko ja itse sovelluksen koodaus. Näiden työvaiheiden jälkeen seuraa julkaisu. Työvaiheet eivät etene aivan tässä järjestyksessä, sillä grafiikkaa ja ominaisuuksia suunnitellaan sekä lisätään myös myöhemmässä vaiheessa. Työvaiheista työläin on ehdottomasti itse sovelluksen koodaus.

Konseptin suunnitteluun kuuluu arviointi siitä, miltä pelisovellus tulee näyttämään sekä mitä se tarjoaa ja miten tämä kaikki toteutetaan. Minkälaista grafiikkaa sovellus sisältää sekä miten taito- sekä katsojajärjestelmät toimivat yhdessä ja erikseen. Grafiikkavaihe pitää sisällään spritejen eli hahmojen, talojen, muiden objektien ja käyttöliittymäkuvakkeiden teon. Grafiikkavaihe sisältää myös graafista suunnittelua. Sovelluksen koodausvaihe sisältää ominaisuuksien koodaamisen lisäksi Unity-projektin luomisen, spritejen tuomisen sovellukseen ja näkyville, käyttöliittymän teon sekä ensimmäisen sovelluspaketin luonnin ja testauksen.

Kokonaistyön vaiheistus ja ajoitus löytyy liitteestä 1. Liite ei kuitenkaan sisällä tämän raportin kirjoitukseen varattua 50 tuntia, joka tapahtuu mobiilipelisovelluksen edistymisen ohella.

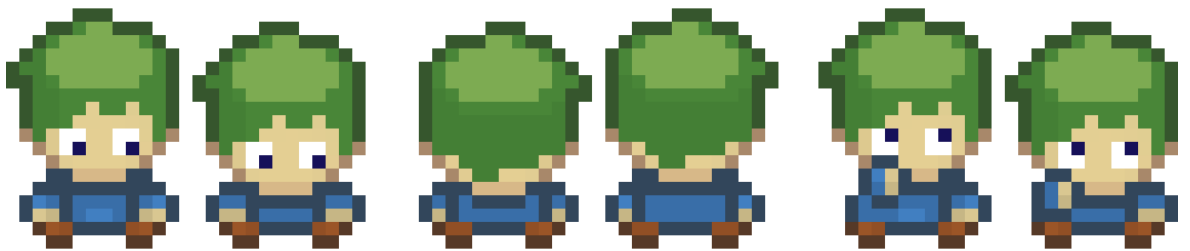
### 4.1 Projektin aloitus Unityssä

Projektin toteutus aloitettiin luomalla 2D Unity-projekti. Jotta kehitys olisi järjestelmällisempää, projektin sisälle rakennettiin kansiot ja ala-kansiot kaikille eri osa-alueille kuten scripteille eli komentosarjoille, grafiikalle, fonteille ja Unityn prefabeille. Unity-projekteissa paljon käytetyn prefab- ominaisuuden avulla voi tallettaa pelin objektin, jossa on valmiina komponentteja sekä ominaisuuksia. Kaikki tähän tehdyt muokkaukset näkyvät välittömästi kaikissa sen esiintymissä, mutta on mahdollista myös ohittaa komponentit ja ominaisuudet kullekin esiintymiselle erikseen (Unity Technologies 2018b).

Unity tukee kahta ohjelmointikieltä, C#:ia ja UnityScriptiä. C# on alan standardi ohjelmointikieli, joka muistuttaa Javaa ja C++:ssaa. UnityScript on erityisesti suunniteltu käytettäväksi Unityn kanssa ja joka on mallinnettu JavaScriptin jälkeen. Näiden lisäksi monia muita .NET-kieliä voidaan käyttää Unityllä (Unity Technologies 2018c). ”Streamer Simulator”-projektissa käytetään C#-ohjelmointikieltä, alan standardin ohjelmointikielen, aiemman kokemuksen sekä mahdollisen hyödyllisyyden vuoksi myös Unityn ulkopuolisessa ohjelmoinnissa.

Kaikki grafiikka tuodaan suoraan Unity-projektin grafiikka-kansioon ja fontti omaan kansioonsa. Pikseligrafiikan pienen koon vuoksi ei tarvitse käyttää pakkausta ja suodatustilan voi asettaa pikseligrafiikalle tyypilliselle pistesuodatukselle, jossa ei ole mitään interpolatiota. Eri interpolaatioissa pikselin väri ja läpinäkyvyys on muokattu lähellä olevien pikselien mukaan ja tällä esimerkiksi korjataan grafiikan särkyvyyttä ja muodostetaan sileitä käyriä. Pistesuodatustilassa kun asettaa pikseligrafiikalle ne pikselin arvot, jotka ne kattavat näytöllä, ne näkyvät näytöllä kristallinkirkkaana (Soher 2012). Muun ei-pikseligrafiikan kanssa voidaan käyttää Unityn oletustuontiasetuksia, kuten bilineaarista suodatustilaa ja normaalia pakkausta.

Kristallinkirkkaan pikseligrafiikan takaamiseksi tulee varmistaa, että jokainen sprite-pikseli esiintyy yhden näytön pikselin tai muun tasaluvun kanssa. Tämän tuloksen saavuttaminen on kiinni pelin kameran ortografisen koon muuttamisesta, mutta sen asettaminen joudutaan tekemään kaikille erikokoisille ruuduille (Tham 2015). Unityn Asset Storesta löytyvä "Pixel Perfect Camera"-työkalu, säätää kameran koon käyttäjän ruudun mukaisesti ja ratkaisee tämän ongelman. Tämä työkalu löytyi Unity-käyttäjältäni, sillä se oli tälle aiemmin lisätty. Nykyään "Pixel Perfect Camera"-työkalu on maksullinen.



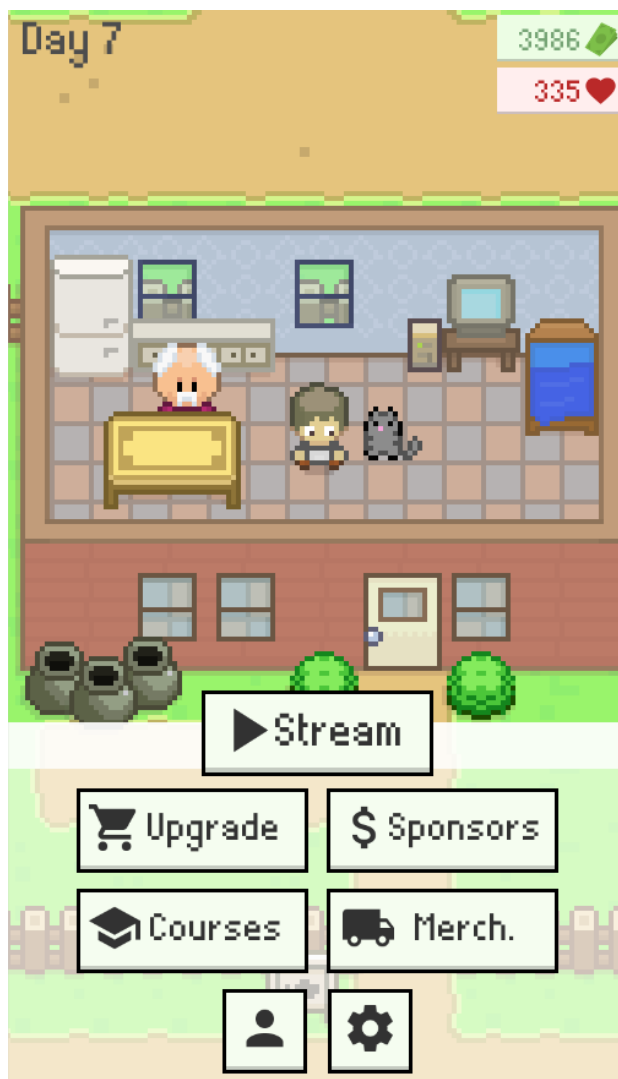
Kuva 4. Pelaajan hahmon sprite-kuvatiedosto, josta eri animaatiot toteutetaan

Pikselihahmot animoidaan Unityn animaatiojärjestelmällä. Pelaajan hahmolle lisätään hengitysanimaatio edestä- ja takaapäin sekä miettimisanimaatio edestäpäin kuvattuna. Animaatiossa hahmon sprite vuorottelee pikselin verran alempana ja ylempänä olevien hartioiden välillä, joka kuvastaa hahmon elollisuutta.

## 4.2 Käyttöliittymä

Pelin pohjalla on pikseligrafiikasta pala palalta rakennettu maasto Unityn Tilemap- ominaisuudella. Tämä ylhäältäpäin kuvattu maasto sisältää mm. huonekaluja, aitoja, puskia ja animoituja hahmoja, joista pelaaja on keskellä laitteen näyttöä.

Unityn Canvas-käyttöliittymä muodostaa pelin valikot, jotka sisältävät pelin tämänhetkiseen tilanteeseen nähden tärkeitä asioita, kuten rahan määrän tai valikon, jonne pelaaja on itsensä navigoinut käyttäen Canvas-käyttöliittymän valikkopainikkeita. Painikkeet ohjelmoidaan suorittamaan erilaisia scriptejä, kuten sulkemaan nykyisen valikon ja avaamaan painikkeeseen liittyvän valikon tämän tilalle. Canvas-käyttöliittymän grafiikka ei sisällä pikseligrafiikkaa, toisin kuin pohjalla oleva maailma sillä sen pitää skaalata eri näytökokoihin mahdollisimman tarkasti peittämättä näkyvyyttä esimerkiksi suoratoiston aikaisesta keskustelusta. Lähes kaikki pelin ohjaaminen tapahtuu Canvas-käyttöliittymässä, kuten navigointi ja reagointi katsojien lähettämiin viesteihin.



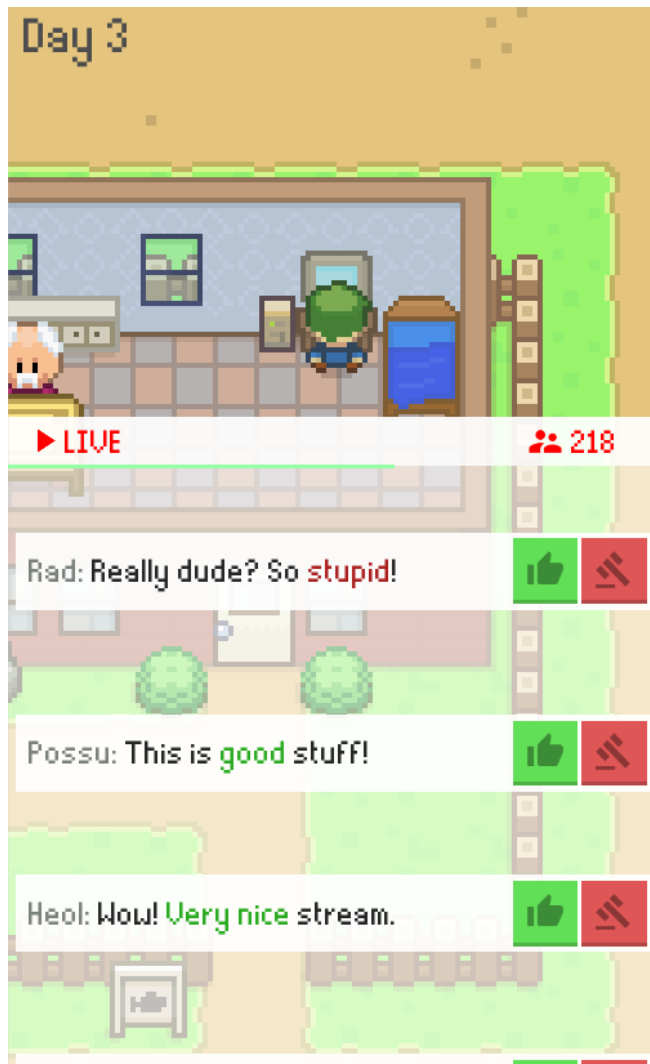
Kuva 5. Päävalikko on yksinkertainen ja helposti opittavissa

Käyttöliittymää tehdessä on hyvä ottaa huomioon käyttöliittymän selkeys, tiiviys, tuttavuus, reagoivuus, johdonmukaisuus, viehättävyys, tehokkuus ja anteeksiantavuus (Fadeyev 2009). Canvas-käyttöliittymä rakennettiin olemaan hyvin selkeä ja ymmärrettävä. Mobiililaitteiden pienten näyttöjen vuoksi tilaa on rajoitetusti ja kaikki näytöllä oleva teksti täytyy

jättää minimiin. Googlen Material Design Icons kuvakkeet kertovat paljon painikkeen painamisen mahdollisista seurauksista ja osa painikkeista onkin varustettu pelkällä kuvalla. Nämä kuvakkeet tekevät myös valikoista tuttaviamuulla nähtävien valikkojen ja kuvakkeiden samankaltaisuuden vuoksi. Päävalikosta muualle navigoidessa hahmo siirtyy miettimisanimaatioon, jolloin käyttäjä ymmärtää varmasti että on pelissä kyseinen hahmo. Jokaisen valikon yläosaan sijoitettiin valikon nimi, jotta käyttäjä tietää kokoajan missä on ja lisäksi jokaisessa valikossa on tarjolla myös selkeä takaisin painike. Useasta valikosta, kuten статистиikkavalikosta, löytyy oma info painike jota painamalla saa lisätietoa kyseisestä valikosta. Useasta toiminnasta, kuten tietokoneen päivittämisestä saat myös varmistusvalikon. Navigointi valikoiden välillä on näin ollen anteeksiantavaa, johdonmukaista ja tehokasta. Käyttöliittymän valikoiden värit, kuvakkeet ja fontti lisäävät käyttöliittymän yhtenäisyyttä ja viehättävyyttä.

### **4.3 Katsoja- ja keskustelujärjestelmä**

Suoratoiston aikana pelaaja saa katsojia lähetykseensä, joiden määrä perustuu taitotasoihin ja seuraajien määrään. Pelaaja voi lähetyksen aikana reagoida katsojien lähettämiin viesteihin joko positiivisesti tai negatiivisesti. Vihreää tekstiä sisältävä viesti on positiivinen, jolloin tähän tulee vastata vihreällä painikkeella ja punaista tekstiä sisältävään punaisella painikkeella. Oikeat vastaukset tuovat lisää katsojia ja väärät laskevat niitä. Lähetyksen aikana voi myös ilmetä lisää katsojia antavia painikkeita hahmon lähetyville kuvastamaan onnistunutta lähetystä.



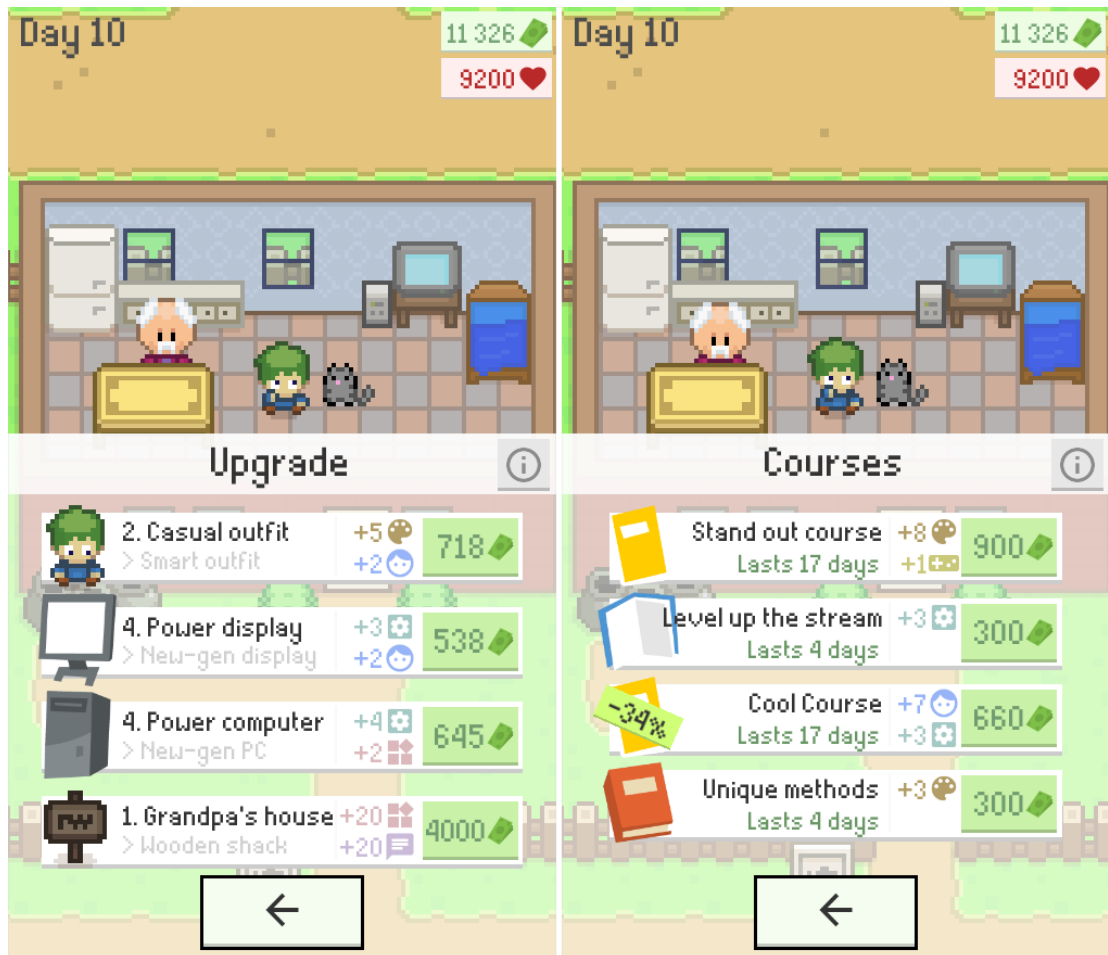
Kuva 6. Suoratoiston aikainen käyttöliittymä

Keskustelu ja tähän reagointi tapahtuu Canvas-käyttöliittymässä. Viestit pitävät sisällään satunnaisen nimen sekä tekstin. Viesti-peliobjekti on ohjelmoitu reagoimaan painikkeiden painalluksiin sekä liikkumaan alhaalta ylös ja häviämään. Viesti-peliobjektin usean komponentin ja ominaisuuden vuoksi tästä tehtiin Prefab- ominaisuuden avulla talletettu peliobjekti muokkauksen ja uudelleen käytön helpottamiseksi. Viestien ilmentymisen tiheys perustuu katsojien määrään.

#### 4.4 Taitojärjestelmä

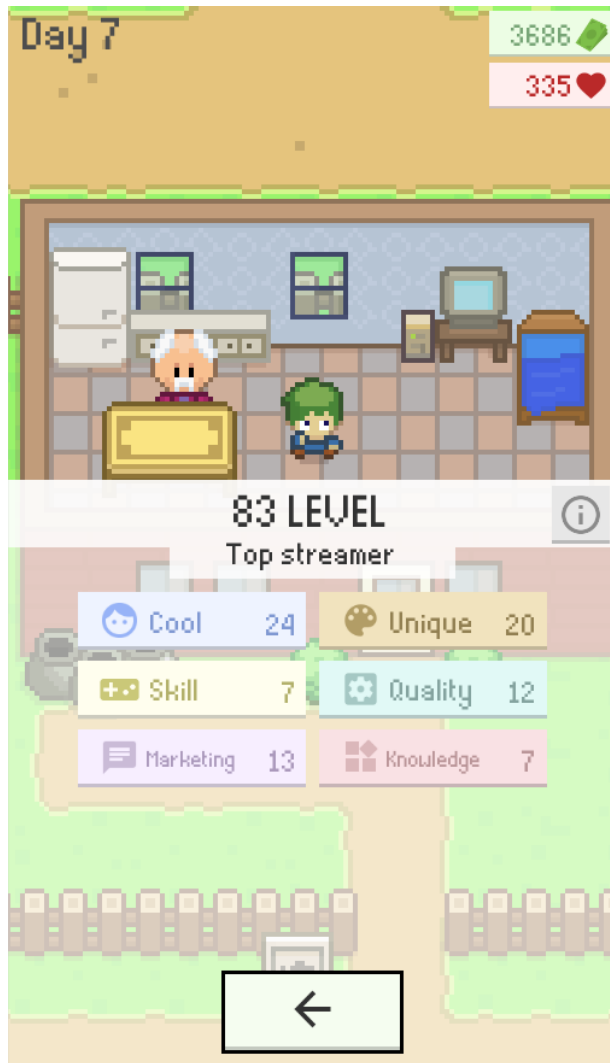
Pelaajan hahmolla on kuusi eri taitoa, joiden taso on esitetty numeraalisesti. Pelaajalle on myös annettu hahmon kokonaistaso, joka on kaikkien taitotasojen summa. Joka kymmenes kokonaistaso parantaa tämän alta löytyvää tittelä. Taitoihin kuuluvat cool, unique, skill, quality, marketing ja knowledge. Näistä kaksi ensimmäistä auttavat uusien katsojien saamisessa, seuraavat kaksi katsojien seuraamishalukkuudessa ja viimeiset seuraajien

osallistumisesta pelaajan suoratoistolähetysiin. Taitava pelaaja saattaa oivaltaa, että pelin alussa on tärkeää saada katsojia, jonka jälkeen on syytä edistää seuraamishalukkuutta ja seuraajien kertyessä heidän halukkuuttaan osallistua lähetysiin. Lisäksi onnistuneen lähetysten painikkeen tehokkuuteen vaikuttaa skill-taito, pelaaja saa myytyä paremmin fanituotteitaan suuremmalla marketing-taidolla ja sponsoritarjoukset ovat suosivampia suurella quality- ja knowledge-tasolla.



Kuva 7. Puitteiden päivitys ja satunnainen kurssivalikko

Pelaajan on myös syytä taidon vaikutuksen lisäksi huomata rahankäytön kohteita mietittäessä, että taitotason päivityksen lähtöhinta on 100 rahaa per taso ja jokainen päivitys lisää kyseisen päivityksen seuraavan tason hintaa. Jatkuvasti vaihtuvat ja satunnaiset kurssit ovat myös pelaajan valinnanvapautta syventävä järjestelmä. Pelaaja voi olla vain yhdelle kurssilla kerrallaan ja kurssin hinnan lisäksi tämän pituus riippuu sen antamasta taitosopalkkiosta, mutta hinta saattaa olla myös alennettu tai ilmainen jolloin kurssin pituus on yleensä pidempi. Pelaajan taitavuutta mitataan siis muidenkin simulaatiopelien tavoin järkevien rahankäyttövalintojen kautta. Peli tarjoaa myös rennolle, huolettomalle ja välinpitämättömälle pelaajalle saman kokemuksen, mutta kehitys ei välttämättä tapahdu yhtä nopeasti.

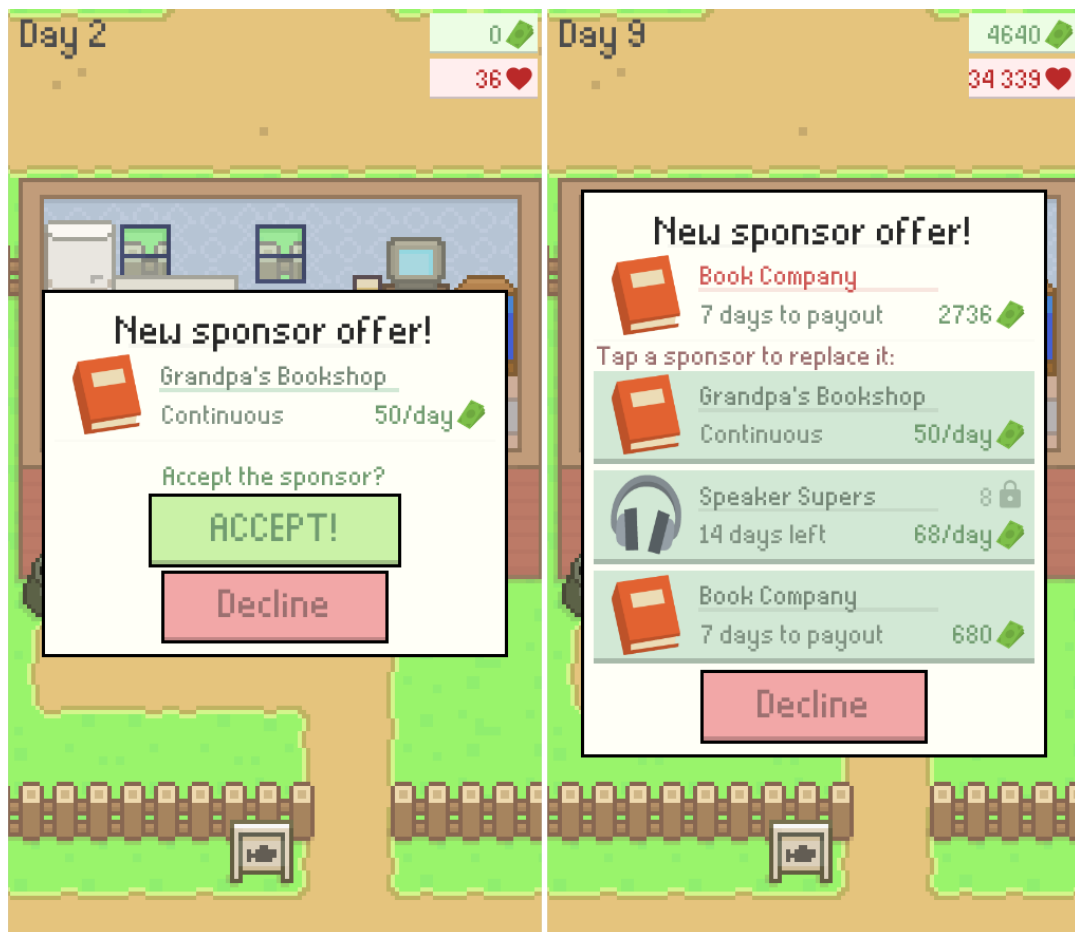


Kuva 8. Taitotasot ovat listattuna статистиikka valikossa

Pelaaja saa lisää tasoja päivittämällä puitteita, osallistumalla kursseille sekä mahdollisesti myös satunnaisista tapahtumista. Ihmiset suhtautuvat luonnostaan vastenmielisesti eri asioiden menetykseen ja päinvastoin kokevat suurta halua niiden keräämiseen (Cleary 2011). Numeraalisesti esitetyt taidot mahdollistavat helpon päämäärän asettamisen sekä vertailtavuuden. Kaikki parannukset mobiilipelisovelluksessa nostavat siis taitotasoa, jotka vaikuttavat suoranaisesti pelin eri järjestelmiin. Tästä syntyy yksinkertainen ja helposti ymmärrettävä käyttöliittymä, vaikka tämän alla toimivat järjestelmät ovatkin monimutkaisempia. Ne ovat silti helposti jatkokehitettävissä, muokattavissa ja tasapainotettavissa. Peli on näin ollen helppo aloittaa ja pitkä aikaisemmille pelaajille kiinnostavuutta löytyy järjestelmien toimintatapojen opiskelusta ja testailusta. Pelaajalla on mahdollisuus saada lisätietoa painamalla taitoa tai valikon infopainiketta.

## 4.5 Pelivaluuttajärjestelmä

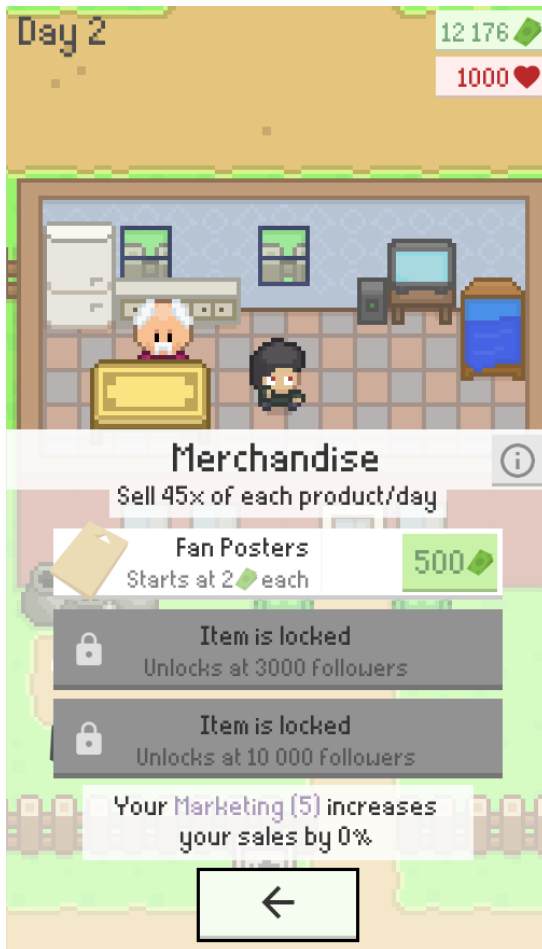
Pelaajan puitteiden parantaminen maksaa pelirahaa, joka on kuvattu vihreällä setelillä. Pelirahaa saadaan muun muassa sponsoreista, joista pelaaja saa tarjouksia ja fanituotteiden myynnistä, joita pelaajalla on mahdollisuus myydä tietyn seuraajamäärän jälkeen sekä satunnaisista tapahtumista, joihin kuuluu lahjoitusten lisäksi myös valinnaiset video-mainokset. Pelin lopussa saadaan siis paljon enemmän tuloja ja tätä kautta parannuksia kuin juuri aloittaneena, mutta kehitystä pyritään tasoittamaan aiemmin mainitulla funktiolla, jossa aikaisemmat tasot vaikuttavat suhteessa enemmän kuin myöhemmät. Tämän lisäksi puitteiden päivittämisen hinta nousee hieman jokaisen ostetun päivityksen seurauksena.



Kuva 9. Vasemmalla on sponsoritarjous ilman voimassa olevia sponsoreita, sekä oikealla kaikki sponsoripaikat varattuna

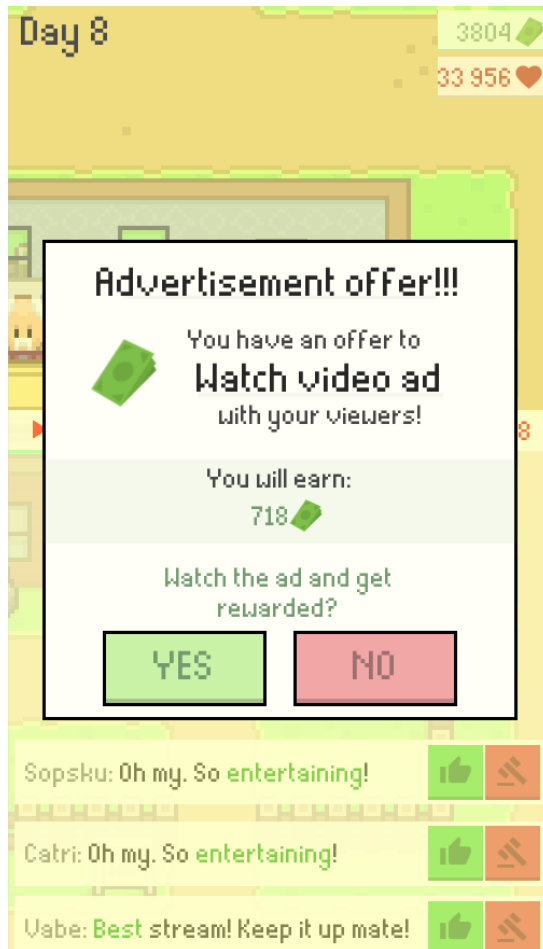
Mahdolliset sponsoritarjoukset annetaan pelaajalle uuden päivän alkaessa omalla valikollaan, joka rakentuu nykyisten sponsori määrän mukaan. Mikäli pelaajalla on alle 3 sponsoria, niin valintana on ainoastaan myönteinen ja kielteinen, mutta jos käyttäjän kaikki 3 sponsoripaikka ovat täynnä, pelaajaa pyydetään vaihtamaan tarjottu sponsori voimassa olevan kanssa. Sponsorit ovat satunnaisesti luotuja ja ne voivat olla jatkuvia ja maksaa päivä kerrallaan tai koko summan tietyn ajan päästä. Sponsoritarjouksien tihey-

teen vaikuttaa nykyisten sponsorien määrä ja niiden tuottavuuteen seuraajamäärän lisäksi pelaajan quality- ja knowledge-taito. Sponsorijärjestelmä suunniteltiin antamaan valinnan ja miettimisen pelaajalle.



Kuva 10. Fanituote valikon käyttöliittymä

Fanituotteiden myynti on lukittuna pelin alussa ja ne aukeavat yksi kerrallaan tietyn seuraajamäärän ylittyessä. Esimerkiksi pelaajan saavuttaessa 1000 seuraajaa, hänellä on mahdollisuus alkaa myymään fanijulisteita, joiden myynnin aloitus on maksullista. Fanituotteiden myyntiä voi myös kehittää ja ne toimivat investointeina, jotka maksavat itsensä takaisin aikanaan. Fanituotteiden myyntiin vaikuttaa seuraajamäärän lisäksi pelaajan marketing-taito. Fanituotteet ovat suunniteltu antamaan pelaajalle uusia ominaisuuksia ja mahdollisuuksia sekä asettamaan tavoitteita pelaajan seuraajamäärälle. Investoimalla fanituotteisiin pelaaja voi moninkertaistaa tulonsa, mutta hänen on syytä miettiä, kannattaisiko rahat käyttää mieluummin puitteiden parantamiseen ja näin ollen suurempaan katsoja- ja seuraajavirtaan ennen fanituotemyyntien aloittamista tai päivittämistä.



Kuva 11. Videomainos tarjouksen valikko

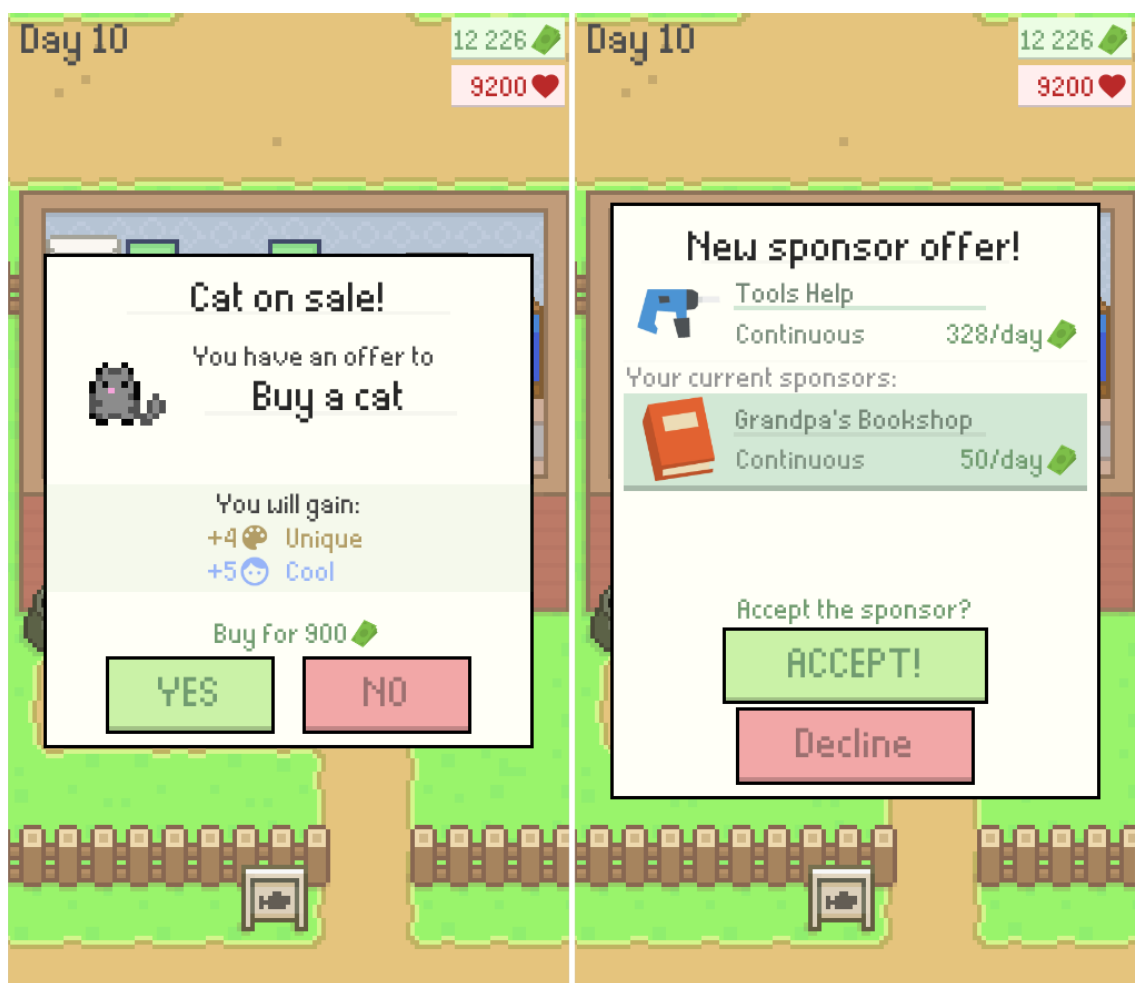
Myös eräät satunnaiset tapahtumat palkitsevat pelaajan pelirahalla. Opinnäytetyön lopussa valmistuneessa mobiilipelissä näitä tapahtumia ovat ainoastaan videomainokset ja lahjoitukset. Näitä tapahtumia on tarkoitus lisätä ja laajentaa jatkossa, josta kerrotaan lisää pohdinta osiossa. Valinnaiset videomainokset pyytävät pelaajaa katsomaan videomainoksen, jonka katsottuaan pelaaja palkitaan. Lahjoitukset toimivat samalla periaatteella, mutta eivät vaadi pelaajalta mitään toimintaa. Lahjoitukset ovat kuitenkin harvinaisia ja palkitsevat pelaajaa pienemmällä summalla. Videomainokset ovat toteutettu Unityn sisäänrakennetulla mainosjärjestelmällä aiemman kokemuksen vuoksi.

Lahjoitusten sijasta videomainokset eivät ole pelaajalle täysin satunnaisia, vaikka kuuluvatkin tähän samaan satunnaistapahtumien kategoriaan. Videomainostarjoukset tulevat pelaajalle noin joka neljäs pelipäivä ja ne ovat täysin valinnaisia. Syy tähän sykliin on se, että mitä enemmän pelaaja katsoo mainoksia putkeen, sen arvottomampia näyttökertoista tulee. Palkitut mainokset tuottavat tuloja paremmin kuin pakotetut, sillä ne ovat korkeampilaatuisia näyttökertoja. Ne auttavat myös pelaajan sitoutumisessa ja pelin pelaamisen jatkamisessa. (Unity Ads 2018.) Kuten aiemmin mainittiin, pelin ehdoton tavoite ei ole rahallinen tuotto vaan jalansija pelialalta ja markkinoilta. Mainoksia ei kuitenkaan jätetty pe-

listä kokonaan pois siitä syystä, että tällä olisi mahdollisuus tuottaa sen verran, jotta aikaa voidaan järjestää ja käyttää enemmän tämän jatkokehitykseen.

#### 4.6 Satunnaistaminen ja tapahtumajärjestelmä

Pelin kulun mielekkyyttä ja sisältöä on pyritty lisäämään satunnaistamisella. Opinnäytetyössä valmistuneessa mobiilipelin versiossa on hyvin rajallinen määrä satunnaisia tapahtumia, mutta järjestelmä ja sen ohjelmointi toteutettiin niin, että jatkokehityksessä näiden lisääminen on helppoa ja vaivatonta. Satunnaisia tapahtumia ovat kesken lähetyksen tapahtuvat edellisessä luvussa mainitut lahjoitukset ja videomainostarjoukset sekä uuden päivän alkaessa sponsori- ja muut ostotarjoukset.



Kuva 12. Aamulla tapahtuvia satunnaisia tapahtumia ovat sponsori- ja muut tarjoukset

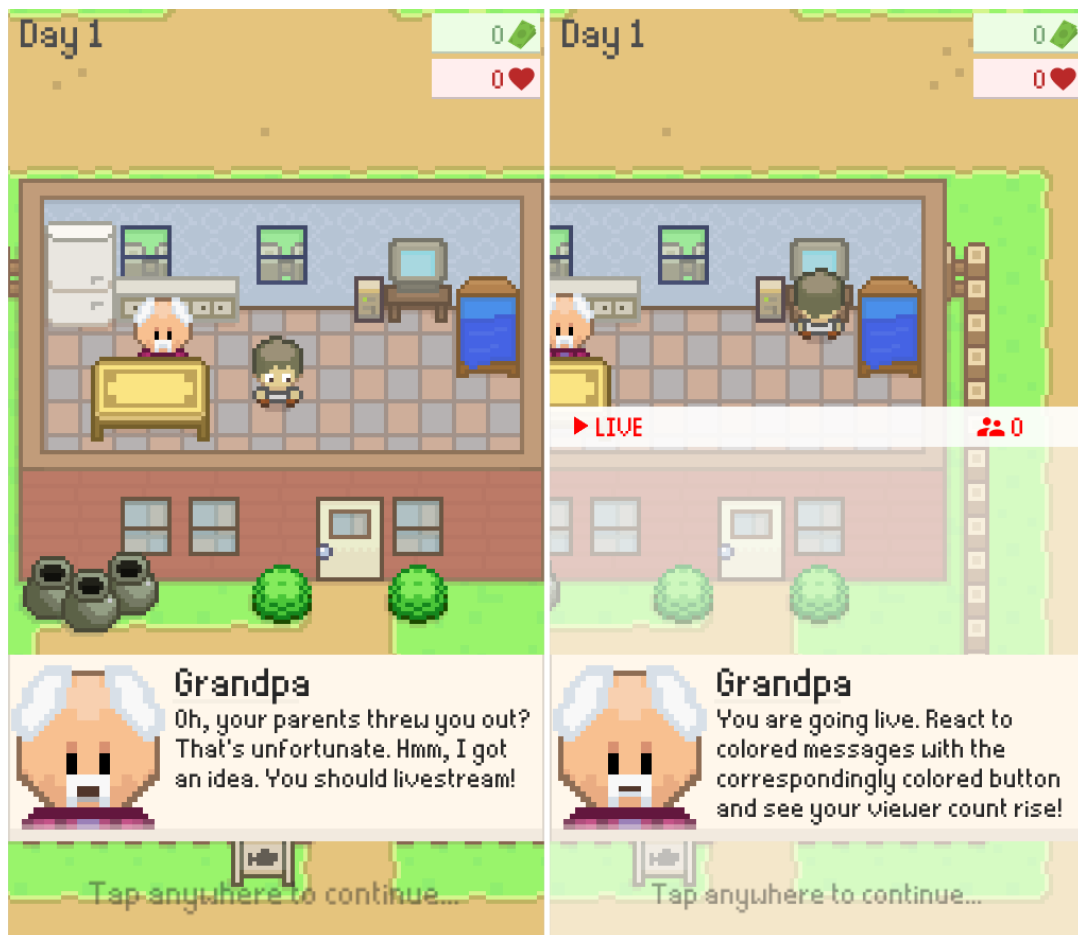
Aamulla tapahtuviin satunnaisiin tapahtumiin kuuluu ensisijaisesti sponsoritarkoukset, jotka ovat varsinkin pelin alussa tärkeä rahanlähde pelaajalle. Jos pelaaja ei saa sponsoritarkousta, hänellä on pieni mahdollisuus saada muunlainen tarjous, kuten tarjous kissasta tai esimerkiksi paremmasta puhelimesta. Järjestelmään voidaan rakentaa erilaisia yhdistelmiä, kuten kissatarjouksen tapauksessa tämän voi ostaa vain kerran jonka jälkeen kissan

sijasta voi saada jatkuvasti tarjouksia kissan leluista. Toisin kuin kissan tapauksessa, voit ostaa esimerkiksi paremman puhelimen monesti, mikäli saat siitä useampaan kertaan tarjouksen. Kissatarjousta lukuun ottamatta, tarjoukset tarjoavat satunnaisen määrän taitotasoa, jotka ovat yhdistettävissä tarjouksen tuotteeseen, kuten parempi puhelin parantaa ainakin marketing- ja mahdollisesti myös cool-taitoa. Tarjouksien hinnat saattavat myös vaihdella, mutta ne maksavat vähintään aiemmin mainitun 100 rahaa per taitotaso.

Satunnaiset tapahtumat tarjoavat pelaajalle satunnaisesti luotuja rahankäytön valintoja ja luovat pelaajalle jännittävää ilmapiiriä, sillä mitä vain voi tapahtua milloin vain. Satunnais-ten tapahtumien määrää ja monipuolisuutta tullaan lisäämään jatkokehityksen aikana, jota on suunniteltu pohdinta osiossa.

#### 4.7 Aloittelijan opastus

Aloittelijan saattaminen peliin on tärkeä osa tämän kiinnostuvuutta ja sitoutumista peliin. Aloittelijan opastus suunniteltiin sisältävän hyvin pienimuotoisen ja humoristisen tarinan siitä kuinka pelaajan hahmo on heitetty ulos vanhempiensa luota ja kuinka hän päätyy opastuksen aikana suoratoistamaan sisältöä isoisänsä vanhalta tietokoneelta.



Kuva 13. Ensimmäisen päivän opastus on minimaalinen, mutta informatiivinen

Järjestelmien toimintatavat ja muu on jätetty opastuksesta pois sillä ne ovat monimutkaisia, mutta lyhyen opastuksen jälkeen pelaajalla on mahdollisuus navigoida eri valikoissa ja lukea näistä lisätietoa painamalla infopainiketta. Hyvän ensikokemuksen luomiseksi pelin valikot eivät saa olla tulvillaan painikkeita ja sisältöä heti peliä aloittaessa (Shvebish 2014). Pelin opastuksen tarkoitus on siis näyttää mikä on pelin tarkoitus ja jättää loput ominaisuudet pelaajan itse opiskeltavaksi ja pähkäiltäväksi. Opinnäytetyön jälkeen alkaneessa beta-versiovaiheessa kerätään palautetta siitä mihin suuntaan opastusta tullaan kehittämään.

#### **4.8 Tallennusjärjestelmä**

Aikaisemmasta mobiilipelisovellusprojektin tallennusjärjestelmästä oli paljon hyötyä sen toteuttamisesta tässä projektissa. Muita järjestelmiä ja istunnon tietojen säilytystä miettiessä tallennusjärjestelmä oli kokoajan tietoisesti mukana kehityksen suunnittelussa, vaikka sen toiminnallisuus toteutettiin projektin loppuvaiheessa.

Pelaajan tiedot ovat pelatessa tallessa tämän hahmoa ja peliä ohjaavassa scriptissä eli komentosarjassa. Tällaisia tietoja ovat muun muassa rahat, seuraajat, päivä, taitotasot, päivityksien ja fanituotteiden tasot, nykyiset sponsorit sekä mahdollinen nykyinen kurssi ja tämän hetkinen kurssien tarjonta. Tiedot voisivat olla hajautettuna omiin järjestelmiinsä, mutta se tekisi niiden hallinnasta hankalampaa.

Jokaiselle kokonaisuudelle kuten päivityksille, kursseille, sponsoreille, fanituotemyynnille ja asetuksille luotiin oma scripti eli komentosarja mitä voidaan käyttää myös pelaajan hahmon oman scriptin kautta. Jokainen näistä scripteistä toteutettiin niin että ne sisältävät funktion, joka asettaa kyseistä järjestelmää koskevat tiedot ajan tasalle pelaajan hahmon scriptistä löytyvien tietojen mukaisesti ja päivittävät valikkonsa. Tämä mahdollistaa hahmon scriptin tietojen vaivattoman tallentamisen laitteeseen, sillä ne ovat kaikki yhdessä scriptissä sekä niiden yksinkertaisen lataamisen peliä käynnistettäessä, jolloin myös kaikkien järjestelmien tiedot asetetaan pelaajan tietojen mukaisiksi funktioidensa avulla.

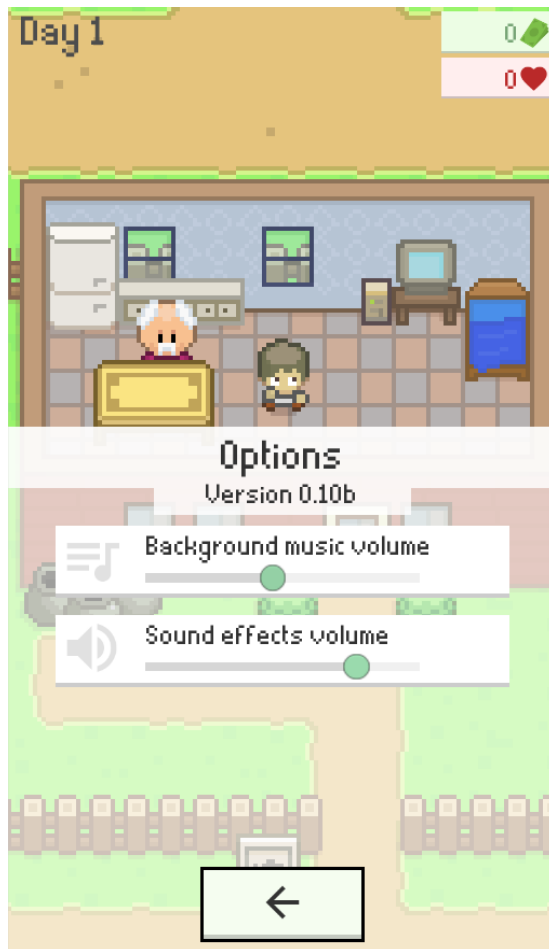
Tallennettu tiedosto on siis tekstitiedosto, joka sisältää yllä mainitut pelaajan tiedot seuraavaa peliä varten. Tallennustiedostoon lisätään vielä tallentaessa tallennustiedoston versio, joka voidaan tarkistaa ladatessa ja esimerkiksi kaikki tiedot nollata tämän mukaan mikäli peli kokee suuria muutoksia jatkokehityksen aikana. Tekstitiedosto on kuitenkin helposti avattavissa ja muokattavissa, joten tämän muokkaamisen estämiseksi tiedoston arvot muutetaan binaariseen muotoon tallennettaessa. Tietojen suojaaminen ei ollut

välttämätöntä opinnäytetyössä valmistuneelle versiolle, sillä muokkauksilla ei voitu tehdä suoranaista harmia pelille. Jatkokehityksessä voi kuitenkin löytyä tarvetta tietojen suojaamisen lisäämiselle, jolloin tallennustiedostoa ei ole tarvetta nollata, sillä tiedosto on jo binäärimuodossa.

#### **4.9 Äänijärjestelmä**

Taustamusiikki sekä äänet ovat ulkoistettuja, eivätkä suuressa osassa opinnäytetyötä, mutta kuitenkin tärkeä osa pelin ilmapiirin luonnissa. Taustamusiikki ja äänet valikoitiin aiemmin ostetuista laajoista musiikki- ja äänipaketeista, jotka olen hankkinut aikaisempaa mobiilipelijärjestelmää varten. Taustamusiikiksi valikoin rentoon kategoriaan kuuluvan musiikin, sillä simulaatiopelit ovat yleisesti ottaen rentoja ja valikoiden navigointi sekä eri toimintojen ääniksi niihin yleisesti sopivia lyhyitä ääniä. Olin huomannut myös aikaisemmassa projektissa Unityn käyttävän omaa äänijärjestelmäänsä Android-käyttöjärjestelmässä, mikä kuitenkin aiheutti viivettä äänien kuulumisessa. Tämän korjaamiseksi olin kuitenkin löytänyt ilmaisen lisäosan Unityn foorumeilta, joka ohjaa äänet suoraan Android-käyttöjärjestelmän natiiviin eli alkuperäiseen äänijärjestelmään, joka toistaa äänet ilman viivettä.

Asensin saman lisäosan ”Streamer Simulator”-mobiilipelille, sillä äänet olivat osa välitöntä palautetta joka pelaajalle annetaan hänen tehdessään valintoja. Esimerkiksi suoratoiston aikaiseen viestiin reagoidessa käyttäjälle soimitaan eri ääni eriväristen lentävien peukkujen lisäksi kuvastamaan oliko reagointi oikea vai väärä. Tällaiset välittömät palautteet on tultava viiveettä pelin sujuvuuden kannalta.



Kuva 14. Asetusvalikon käyttöliittymä

Musiikille sekä äänenvoimakkuudelle oli syytä rakentaa omat osiot asetusvalikkoon, jotta käyttäjät voisivat muokata käyttökokemustaan heille mieluisammaksi. Taustamusiikki on yli minuutin mittainen, mutta se saattaa osoittautua itseään toistavaksi pitkäaikaisemmille pelaajille, jolloin pelaajalla tulisi olla mahdollisuus äänien hiljentämiselle. Mobiililaitteella pelatessa voidaan myös kuunnella musiikkia toisesta sovelluksesta, jolloin pelin taustamusiikki halutaan mykistää ja ääniefektejä mahdollisesti koventaa. Mobiilipelisovelluksen ensimmäinen versio ei sisällä muita asetuksia.

## 5 Pelisovelluksen julkaiseminen

Ennen ”Streamer Simulator”-mobiilipelisovelluksen julkaisua oli syytä käydä läpi pelisovelluksen optimointia ja julkaisuun liittyvää valmistelua. Peli julkaistiin beta-versiona ja ainoastaan Androidille Google Play Storen kautta. Beta-version aikana oli tarkoitus kerätä palautetta, jonka pohjalta pelin järjestelmiä kehitetään eteenpäin sekä tasapainotetaan.

### 5.1 Optimointi

Tallennusjärjestelmän tavoin myöskään optimointi ei ollut erillisenä projektitehtävänä vaan sitä pyrittiin aina miettimään järjestelmiä ja toimintoja suunniteltaessa sekä tehdessä. Optimoinnilla saavutettiin pidempi akun kesto ja pienempi sovelluksen koko.

Pelin suoritustehon ja akun keston parantamiseksi käyttämätön fysiikkamoottori laitettiin pois päältä kuten myös valaistus, jota käytetään muun muassa varjostuksen luomiseen 3D-peleissä. Object pooling eli peliobjektien, kuten suoratoiston aikaisen keskustelun viestien käyttäminen uudelleen on paljon tehokkaampaa kuin niiden tuhoaminen ja uudelleen luominen. Tällöin käytössä olevassa muistissa ei ilmene suuria muutoksia, jotka voivat aiheuttaa pätkimistä kesken pelaamisen (Unity Technologies 2018d). Toteutuksen kannalta tämä tarkoitti sitä, että kun viestiin reagoidaan tai kun se menee ylös ja häviää, se asetetaan epäaktiiviseksi tuhoamisen sijasta. Uuden viestin saapuessa yksi näistä epäaktiivisista viestiobjekteista siirretään alkuun ja sisältö muutetaan, jolloin ne voidaan aktivoida uudelleen pitäen kuitenkin niiden toiminnan täysin ennallaan. Uusi viestiobjekti siis luodaan vain jos kaikki aikaisemmin luodut viestiobjektit ovat jo aktiivisia. Uudelleenkäyttö ei tässä mobiilipelisovelluksen kehitysvaiheessa tuota näkyvää parannusta, mutta jatkokehitystä varten se oli hyvä toteuttaa jo tässä vaiheessa. Kaikki scriptit eli komentosarjat olivat suunniteltu ja tehty niiden nopeutta ajatellen.

Sovelluksen kokoon vaikuttaa siinä käytetyt tiedostot. Pikseligrafiikan kuvatiedostot olivat hyvin pieniä. Myös ulkoistetut kuvakkeet olivat tuotu tarvittavassa koossa projektiin. Taustamusiikki oli suurin tiedosto koko projektissa ja tämän yli minuutin mittaisen ääniraidan soittaminen voisi olla raskasta puhelinten pienelle RAM-muistille. Musiikkitiedoston lataustyyppi muutettiin stream-tyyppiseksi, jolloin ääni toistetaan suoraan laitteen pysyvästä muistista ja tämän ääniraidan laatua pienennettiin, jolloin sen koko tippui lähes kahdella megatavulla. Sovelluksen kokoon vaikuttaa myös suuresti se, mille prosessityypille sovellustiedosto rakennetaan. Sovellus voidaan rakentaa molemmille samalla sovellustiedostolla, mutta tällöin koko on lähes kaksinkertainen. Molemmille prosessityypeille samanaikaisesti rakentaessa sovelluksen kooksi tuli 22,5 megatavua. Erillisiksi sovellustiedostoiksi

rakentaessa ARMv7-tyyppisen prosessorin sovelluksen kooksi tuli 12,7 megatavua ja x86-tyyppisen prosessorin sovelluksen kooksi 13,7 megatavua. Unity rakentaa sovellustiedostot vain sovelluksessa käytettävien tiedostojen kanssa, vaikka Unity-projekti sisälsikin paljon muutakin, kuten kokonaisen äänikirjaston, joista äänet valittiin.

## 5.2 Valmistelu ja julkaiseminen Google Play Storessa

Mobiilipelisovelluksen nimeksi jalostui "Streamer Sim Tycoon". Samankaltaisia nimiä käytetään paljon simulaatiopelissä, joita ovat muun muassa "Game Dev Tycoon", "Rollercoaster Tycoon" ja "SimCity". Nimen "sim"-osio tulee luonnollisesti simulaatiosta ja "tycoon" voidaan ajatella tässä kontekstissa tarkoittavan suurta ja tunnettua suoratoistolähetäjää. Nimi on mielestäni osuva ja se saattaa herättää tuttavuutta sekä mielenkiintoa lukijoissa.

Google Play Storeen julkaistaessa on syytä miettiä pelin versiointia. Pelille voi Unityn sovelluksen rakennusasetuksista täyttää kaksi versioon liittyvää kenttää, jotka ovat version nimi ja numeraalinen paketin versiokoodi. Google Play Store vaatii päivitettyjen sovellustiedostojen versiokoodien olevan suurempia aikaisempiin sovellustiedostoihin verrattuna. Ensimmäisen version nimeksi annoin "0.10b", joka kuvastaa keskeneräisyyttä sekä beta-vaiheessa olevaa pelisovellusta. Annoin versiokoodiksi 100, jolloin versio "1.0" eli julkaisun versiokoodi olisi 1000. Syy ylimääräiseen numeroon versiokoodin lopussa oli se, että Google Play Storeen voidaan asettaa useampi aktiivinen sovellustiedosto, jolloin sovelluksen lataajan laite valitsee hänen laitteeseen uusimman sekä sopivimman sovellustiedoston. Tämä tarkoittaa sitä, että pystyin lataamaan Google Play Storeen molemmille prosessortyypeille sovellustiedoston erikseen säästäten sovelluksen koossa, kunhan muuttan toisen viimeistä versiokoodin lukua suuremmaksi. Päivitykset tulevat siis aina olemaan kahden desimaalin tarkkoja. Pelille on annettava myös "package name" eli paketin nimi, joka muodostuu yleensä kehittäjän ja sovelluksen nimestä. Paketin nimen tarkoituksena on yksilöidä sovellus laitteessa ja Google Play Storessa. Paketin nimi annettiin yleisten ohjeiden mukaisesti, jolloin siitä muodostui "com.iferalgames.streamersimtycoon".

Julkistettavalla Android-sovellustiedostolle on myös annettava Android "keystore" eli avainnippu, johon on luotuna tälle sovellukselle oma "key" eli avain, joka suojaa sovellusta luvattomalta käytöltä. Avainnippu on siis tiedosto, joka sisältää avaimia. Avainnippuun on oma salasana, kuten myös tämän sisältämille avaimille. Aiempien Android-sovellusprojektien takia luotua avainnippua käytettiin myös "Streamer Sim Tycoon"-mobiilipelisovelluksessa, jolloin avainnippuun luotiin tälle sovellukselle kuuluva oma avain.

Ennen julkaisua pelille luotiin logo ja kieliästä korjattiin suurimmat virheet pois. Mobiili-pelisovelluksen julkaisu beta-vaiheeseen onnistui Google Play Developer Consolen kautta vaivattomasti. Pakollisiin tietoihin kuului pelisovelluksen nimi, kieli, lyhyt- sekä pitkäku-vaus, logo, kaksi kuvaa itse pelistä, informatiivinen kuva sovelluksesta, sovelluksen tyyppi ja kategoria sekä sisältöluokitus kysely. Kyselyn tuloksesta pelille muodostui ikäraja ja tämän tyyppinen peli sai kaikilta tahoilta pienimmän mahdollisen ikärajan. Lisäksi tietoihin piti antaa sähköpostiosoite käyttäjien yhteydenottoja varten, mahdollinen linkki sovelluk-sen tietosuojakäytäntöön, sovelluksen hinta, onko sovellus kohdennettu alle 13-vuotiaille sekä sisältääkö se mainoksia. Lopuksi piti tietenkin hyväksyä myös käyttöehdot ja ladata itse sovellustiedostot Google Play Storeen, jonka jälkeen sovellus jäi odottamaan hyväk-syntää.

## 6 Pohdinta

Projektissa syntyi pelisovelluksen Unity-projekti ja Unityllä rakennetut APK- eli "Android application package"-tiedostot, jotka julkaistiin Google Play Storessa beta-versiona. Lisäksi syntyi tämä opinnäytetyön raportti. Toteutusmenetelmät olivat kaikki loistavia valintoja aiemman kokemuksen, helppokäyttöisyyden sekä yksinkertaisen jatkokehityksen vuoksi.

### 6.1 Projektin onnistuminen

Tavoitteena oli suunnitella, kehittää, optimoida, sekä julkaista mobiilipelisovellus Google Play Storessa huuhtikuun 2018 alkuun mennessä. Edellä mainitut tavoitteet täyttyivät ja mobiilipeli oli valmis keräämään palautetta Google Play Storessa avoimen beta-version muodossa. Liite 1 sisältää projektin aikataulutuksen ja sen eri osa-alueiden toteumat sekä tekoviikot. Aikataulusta huomataan, että toteumat pitivät lähes paikkaansa. Suurin ero on kuitenkin uusien käyttöliittymä ominaisuuksien koodaamisessa, johon kuului sponsori-, fanituotemyynti-, tallennus-, äänijärjestelmä, asetusvalikko sekä satunnaiset tapahtumat. Näiden hajauttaminen omiksi osioikseen olisi voinut olla järkevämpää, sillä niiden laajuus oli erittäin suuri. Projektin teossa oli muutamia taukoviikkoja, joita ei ole merkitty luetteloon, mutta aktiivista työskentelyä projektin parissa oli kolmentoista viikon ajan. Itse projektin tekoon varattu 350 tuntia ylittyi myös lievästi, kuten myös raportin kirjoitukseen varattu 50 tuntia. Yhteensä projektiin kului aikaa noin 450 tuntia.

Tavoitteena yritykselleni oli tuoda lisää näkyvyyttä alalla, suurentaa asiakaskuntaa sekä mahdollisesti saada mielenkiintoisia työtarjouksia alalta, mutta niitä ei voi tällä hetkellä analysoida sen tarkemmin. Projektista on varmasti hyötyä uralleni pelinkehittäjänä ja yritykselleni on nyt kaksi erilaista mobiilipelisovellusta, jotka varmasti nostavat toistensa ja tätä kautta myös yrityksen näkyvyyttä.

Uuden mobiilipelisovelluksen kehityksen tavoitteena oli myös oppia lisää mobiilipelikehityksestä ja käyttää jo opittua hyödyksi uuden mobiilipelisovelluksen pohjaa tehdessä, jotta jatkokehitys olisi helppoa ja yksinkertaista. Aiempi mobiilipelisovellus auttoi "Streamer Sim Tycoon"-mobiilipelisovelluksen kehityksen kanssa valtavasti, sillä minulla oli kokoajan tiedossa, mitä seuraavat tehtävät tulevat pitämään sisällään. Näistä tehtävistä varsinkin tallennus-, mainos- ja äänijärjestelmä sekä optimointi valmistuivat aiemman kokemuksen vuoksi nopeasti. Tekisin monet aiemman mobiilipelin järjestelmät nykyään aivan toisin, mutta "Streamer Sim Tycoon"-mobiilipelin järjestelmistä en muuttaisi mitään. Projekti on

siis selvästi kasvattanut myös itsevarmuutta alalle, kuin tehtynä on muutakin kuin ensimmäinen opiskelun ohella valmistunut mobiilipeli.

Opinnäytetyön raportissa pyrittiin myös tarjoamaan ideoita uusille pelikehittäjille mobiilipelisovelluksen suunnitteluun ja kehitykseen sekä auttamaan kehitysympäristön valinnoissa. Raportti on kirjoitettu sellaisen pelin tekemisestä, missä valintoja on mietitty tarkkaan, mutta jonka aloittamista ennen on jo opiskeltu itse Unityn käyttäminen. Toisin sanoen projektissa tehdään ensimmäisen kunnollisen eli harkitun pelin pohja, jossa pohjatiedot ohjelmoinnista ja toimintatavoista on jo opittu. Tähän tilanteeseen uskon monen muunkin aloittelevan pelinkehittäjän samaistuvan ja voisinkin suositella heille raporttini lukemista, koska uskon heidän hyötyvän siitä.

## **6.2 Jatkokehitys**

Jatkokehityksestä on monia ideoita, mutta tärkeintä on kuitenkin beta-vaiheen palautteen analysointi ja järjestelmien tasapainottaminen sekä sisällön lisääminen. Palautteesta olisi tärkeää saada selville muun muassa se, miten eri käyttäjät kokevat ja ymmärtävät eri järjestelmät sekä tajuavatko he niiden tarkoituksen. Jatkokehityksessä haluan ehdottomasti tarjota pelaajille yhä enemmän mahdollisuuksia ja monipuolisuutta. Pelin vieminen myös iOS laitteiden App Storeen on mahdollisesti kehityksen myöhäisemmässä vaiheessa. Myöskin tietokoneversion rakentaminen ja julkaiseminen on mahdollista, mikäli peli saa suuren suosion.

Satunnaistapahtumatyyppien lisääminen tuo peliin enemmän jännittävyttä, mahdollisuuksia sekä monipuolisuutta. Satunnaiset tapahtumat tulevat pitämään sisällään erilaisia sekä monipuolisempia tapahtumia, jotka ovat mahdollisesti kokonaan omia pelikokemuksiaan. Tällainen tapahtuma voisi olla esimerkiksi pelimessut, jossa pelaaja käyttää päivänsä omalla pisteellään mainostamalla suoratoistokanavaansa. Monipuolisuutta tullaan lisäämään myös pelin muuttumisella siinä edetessä. Suoratoiston aikaisen keskustelu voisi muuttua nopeaksi suuren katsojaluvun seurauksena tai esimerkiksi sellaiseksi, että viestit olisivat vain palloja, joita käyttäjän on tarkoitus painaa. Valintamahdollisuuksia voitaisiin antaa myös suoratoiston sisältöön, eli pelaaja voisi itse päättää mitä suoratoistaa ja tehdä sitä mahdollisesti poissa tietokoneeltaan. Tällaisien järjestelmien suunnittelu ja toteutus on kuitenkin työlästä ja jatkokehitys alkaa käyttäjälähtöisesti heidän palautteiden sisällöstä.

## Lähteet

Baron, S. 2012. Cognitive Flow: The Psychology of Great Game Design. Luettavissa: [https://www.gamasutra.com/view/feature/166972/cognitive\\_flow\\_the\\_psychology\\_of\\_.php](https://www.gamasutra.com/view/feature/166972/cognitive_flow_the_psychology_of_.php). Luettu: 21.3.2018.

Byford, S. 2014. Pixel art games aren't retro, they're the future. Luettavissa: <https://www.theverge.com/2014/7/3/5865849/pixel-art-is-here-to-stay>. Luettu: 20.3.2018.

Cleary, C. 2011. Analysis: Option Availability And Loss Aversion in Game Design. Luettavissa: [https://www.gamasutra.com/view/news/125537/Analysis\\_Option\\_Availability\\_And\\_Loss\\_Aversion\\_in\\_Game\\_Design.php](https://www.gamasutra.com/view/news/125537/Analysis_Option_Availability_And_Loss_Aversion_in_Game_Design.php). Luettu 17.4.2018.

CryEngine 2018. Get CryEngine Today. Luettavissa: <https://www.cryengine.com/user/registration>. Luettu: 20.3.2018.

ElHady, H. 2017. Top Game Engines In 2018. Luettavissa: <https://blog.instabug.com/2017/12/game-engines/>. Luettu: 20.3.2018.

Enger, M. 2013. Game Engines: How do they work? Luettavissa: <https://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-theywork/101529/>. Luettu: 20.3.2018.

Fadeyev, D. 2009. 8 Characteristics Of Successful User Interfaces. Luettavissa: <http://usabilitypost.com/2009/04/15/8-characteristics-of-successful-user-interfaces/>. Luettu: 21.3.2018.

Lee, L., Gee, S. & Dolah, J. 2016. Graphic Styles Appearance in Educational Games to Enhance Malaysian Students Learning: A Preliminary Study. Luettavissa: [https://www.researchgate.net/publication/301678773\\_Graphic\\_Styles\\_Appearance\\_in\\_Educational\\_Games\\_to\\_Enhance\\_Malaysian\\_Students\\_Learning\\_A\\_Preliminary\\_Study](https://www.researchgate.net/publication/301678773_Graphic_Styles_Appearance_in_Educational_Games_to_Enhance_Malaysian_Students_Learning_A_Preliminary_Study). Luettu: 20.3.2018.

Material Design 2018. Usability. Luettavissa: <https://material.io/guidelines/usability/accessibility.html>. Luettu: 20.3.2018.

Patel, N. 2017. Why You Should Care About Live Streaming in 2018. Luettavissa: <https://neilpatel.com/blog/live-streaming-importance-2018/>. Luettu: 28.2.2018.

Shvebish, G. 2014. 10 ingredients that concoct a subconsciously addictive mobile app. Luettavissa: <https://thenextweb.com/dd/2014/10/28/10-ingredients-concoct-subconsciously-addictive-mobile-app/>. Luettu: 21.3.2018.

Soher, S. 2012. Texture Import Settings for GUI & 2D Games. Luettavissa: <http://www.sarpersoher.com/texture-import-settings-for-gui-2d-games/>. Luettu: 21.3.2018.

Sonders, M. 2016. New mobile game statistics every game publisher should know in 2016. Luettavissa: [https://medium.com/@sm\\_app\\_intel/new-mobile-game-statistics-every-game-publisher-should-know-in-2016-f1f8eef64f66](https://medium.com/@sm_app_intel/new-mobile-game-statistics-every-game-publisher-should-know-in-2016-f1f8eef64f66). Luettu: 28.2.2018.

Tham, P. 2015. Pixel Perfect 2D. Luettavissa: <https://blogs.unity3d.com/2015/06/19/pixel-perfect-2d/>. Luettu: 21.3.2018.

Unity Ads 2018. Best practices guide. Luettavissa: <https://unityads.unity3d.com/help/monetization/best-practices>. Luettu 18.4.2018.

Unity Store 2018. Unity Personal. Luettavissa: <https://store.unity.com/products/unity-personal>. Luettu: 20.3.2018.

Unity Technologies 2018a. Importing from the Asset Store. Luettavissa: <https://docs.unity3d.com/Manual/AssetStore.html>. Luettu: 21.3.2018.

Unity Technologies 2018b. Prefabs. Luettavissa: <https://docs.unity3d.com/Manual/Prefabs.html>. Luettu: 21.3.2018.

Unity Technologies 2018c. Creating and Using Scripts. Luettavissa: <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>. Luettu: 21.3.2018.

Unity Technologies 2018d. Optimizing Scripts. Luettavissa: <https://docs.unity3d.com/Manual/MobileOptimizationPracticalScriptingOptimizations.html>. Luettu: 18.4.2018.

Wicksteed, T. 2015. Design Analysis: Randomness in games. Luettavissa:  
<https://www.twicecircled.com/2c/design-analysis-randomness-in-games/>. Luettu  
21.3.2018.

