

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Mediatekniikka

2015

Aaron Holanne

# MOBIILIPELIN KONSEPTOINTI JA TOTEUTUKSEN SUUNNITTELU LIBGDX- SOVELLUSKEHYKSELLÄ



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

Aaron Holanne

## MOBIILIPELIN KONSEPTOINTI JA TOTEUTUKSEN SUUNNITTELU LIBGDX-SOVELLUSKEHYKSELLÄ

Tämän opinnäytetyön tavoitteena oli suunnitella mobiililaitteille suunnattu kaksiulotteinen peli ja miettiä sen toteutusta Libgdx-sovelluskehysellä. Suunnitelman pohjaksi tutkittiin menestyviä mobiilipelejä, mobiilipelien markkinoita sekä Libgdx-sovelluskehystä.

Teoriaosuudessa eriteltiin menestyvien mobiilipelien ominaisuuksia sekä tutkittiin mobiilipelien monetisaatiota ja asiakassegmentointia. Menestyvillä mobiilipeleillä osoittautui olevan selkeitä yhteisiä ominaisuuksia sekä pelattavuudessa että sisällössä. Monetisaatiotutkimuksessa havaittiin kolme mobiilipeleille tyypillistä monetisaatiostrategiaa: ilmainen peli, maksullinen peli ja mobiilimainonta, joista ilmainen peli -strategia on ylivoimaisesti suosituin. Mobiilipelien asiakassegmentoinnissa saatiin selville, että valtaosa pelaajista ei laita rahaa pelaamiseen ja jäljelle jäävä pieni osuus jakautuu innokkaisiin ostajiin ja vastahakoisiin ostajiin. Lisäksi teoriaosuudessa perehdyttiin Libgdx-sovelluskehysten tarjoamiin ominaisuuksiin kaksiulotteisen mobiilipelin kehityksen näkökulmasta.

Tutkimustyön pohjalta tehtiin suunnitelma, joka kattaa pelin idean, ominaisuudet ja pelaamisen sekä toteutuksen Libgdx-sovelluskehysellä. Peliin suunniteltiin ominaisuuksia, jotka nousivat esille tutkittaessa menestyviä mobiilipelejä. Mobiilipelien markkinatutkimuksen pohjalta pelin monetisaatiostrategiaksi valittiin ilmainen peli -strategia. Pelin toteutus suunniteltiin sellaisten ominaisuuksien mukaan, joita Libgdx-sovelluskehys tarjoaa kaksiulotteisen mobiilipelin kehitykseen. Opinnäytetyön tutkimusten perusteella laaditun suunnitelman pohjalta voidaan siirtyä saumattomasti käytännön toteutukseen.

### ASIASANAT:

mobiilipelit, pelisuunnittelu, Libgdx

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Media Technology

May 2015 | 36

Instructor: Principal Lecturer Mika Luimula

Aaron Holanne

## DESIGN OF A MOBILE GAME AND PLANNING OF IMPLEMENTATION WITH LIBGDX FRAMEWORK

The purpose of this study was to design a 2D mobile game and plan the implementation of the game with the Libgdx framework. Successful mobile games, mobile game markets, and Libgdx framework were explored for the basis of the design.

The theoretical part of the thesis analyzes the features of successful mobile games and examines the customer segmentation and monetization of mobile games. Based on the analysis of successful mobile games, they appear to have several shared features both in gameplay and contents. The monetization study suggested that there are three typical monetization strategies: free-to-play, paid game, and mobile advertising, the first of which is by far the most popular. The examination of mobile game customer segmentation concludes that the majority of mobile gamers do not spend money on mobile games at all and the remaining gamers are divided into enthusiastic payers and reluctant payers. The theoretical part also covered research on the 2D mobile game development tools of the Libgdx framework.

The game design was created on the basis of the research in the theoretical part. The design consists of the game's idea, features and gameplay as well as the implementation plan of the game with the Libgdx framework. Features that arose from the analysis of successful mobile games were added to the game. On the grounds of the market research, free-to-play was selected as the monetization strategy. The implementation of the game was planned according to the features that the Libgdx framework provides for 2D mobile game development. Based on the design made on the grounds of the research in the theoretical part, the procession to the actual implementation can be seamless.

### KEYWORDS:

mobile games, game design, Libgdx

# SISÄLTÖ

<b>KÄYTETYT LYHENTEET</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>7</b>
<b>2 MOBIILIPELIT</b>	<b>8</b>
2.1 Menestyvän mobiilipelin ominaisuudet	8
2.1.1 Pelattavuus	8
2.1.2 Sisältö	9
2.2 Markkinat	11
2.2.1 Monetisaatio	12
2.2.2 Asiakassegmentit	15
<b>3 LIBGDX</b>	<b>17</b>
3.1 Scene2D	18
3.2 Tekstuuriatlas	19
3.3 Ortografinen kamera	20
3.4 Audio	20
<b>4 MINERAL DIGGER -PELI</b>	<b>22</b>
4.1 Idea	22
4.2 Ominaisuudet	23
4.3 Pelaaminen	25
4.3.1 Toiminnot	25
4.3.2 Eteneminen	25
4.4 Toteutussuunnitelma	26
4.4.1 Ulkoasu	26
4.4.2 Rakenne	31
<b>5 YHTEENVETO</b>	<b>34</b>
<b>LÄHTEET</b>	<b>35</b>

## KUVAT

Kuva 1. Kuvakaappaus Cut The Rope -pelin opastenäkymästä.	9
Kuva 2. Kuvakaappaus Angry Birds -pelin tarinasta.	10
Kuva 3. Kuva Applen julkaisemasta Best of 2014 -listasta (Today's Web 2014).	11
Kuva 4. Kuvakaappaus Infinity Blade -pelistä.	13
Kuva 5. Kuvakaappaus Texturepacker-GUI:n asetusnäkyästä.	20
Kuva 6. Kuvankäsittelyohjelmalla tehty malli Mineral Digger -pelin perusnäkyästä.	28
Kuva 7. Kuvankäsittelyohjelmalla tehty malli Mineral Digger -pelin kivien valikosta.	30
Kuva 8. Kaavio pelin luokista.	32

## KÄYTETYT LYHENTEET

Cross-platfrom	Alustariippumaton. Cross-platform-sovelluskehys mahdollistaa saman pelin julkaisun useille eri alustoille
Iterointi	Silmukkarakenteen suorittaminen
Objekti	Ohjelmiston perusyksikkö, joka sisältää joukon loogisesti yhteenkuuluvaa tietoa ja toiminnallisuutta
Ohjelmointirajapinta	Määritelmä, jonka mukaan eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja eli keskustella keskenään
Renderöinti	Kuvan luonti mallista tietokoneohjelman avulla
Sovelluskehys	Ohjelmistotuote, joka muodostaa rungon sen päälle rakennettavalle tietokoneohjelmalle

# 1 JOHDANTO

Peliteollisuus on merkittävän muutoksen vaiheessa. Viimeisimpien tutkimusten mukaan vuonna 2015 peliteollisuudessa mobiilipelit ohittavat konsolipelit tuottoisimpana sektorina. Jatkuvasti tehokkaammiksi kehittyvien älypuhelimien ja tablettien sekä kasvavien sovelluskauppojen ansiosta itsenäiset pelinkehittäjät voivat helpommin kuin koskaan tulla markkinoille hyvin alhaisella budjetilla ja lähes riskittä. Tuottavan ja toimivan mobiilipelikonseptin luominen vaatii kuitenkin tarkempaa tietoa alan markkinoista ja erilaisista ominaisuuksista, joita tällä hetkellä menestyvät mobiilipelit tarjoavat.

Pelinkehittäjillä on tärkeää olla pelinkehitystyökalu, joka mahdollistaa nopean prototyyppien teon ja kuluja säästävän tavan ideoiden implementointiin. Nykyään on tyypillistä julkaista peli nopeasti usealle alustalle. Tällaisen menettelyn mahdollistavaa ominaisuutta kutsutaan cross-platform-ominaisuudeksi, ja se on syytä nostaa tärkeäksi kriteeriksi pelinkehitystyökalulle. Pitäen silmällä peliteollisuudessa tapahtuvaa muutosta kriteeriksi voidaan myös asettaa kattavat ominaisuudet mobiilipelien kehitykseen. Näillä vaatimuksilla sopivaksi pelinkehitystyökaluksi osoittautuu Libgdx. Libgdx on avoimen lähdekoodin cross-platform-sovelluskehys, joka tarjoaa yhtenäisen ohjelmointirajapinnan pelinkehitykseen Java-ohjelmointikielellä.

Tässä opinnäytetyössä tutkitaan mobiilipelejä ja Libgdx-sovelluskehystä ja suunnitellaan kaksiulotteisen mobiilipelin konsepti ja toteutus. Teoriaosuudessa tutkitaan menestyvien mobiilipelien ominaisuuksia ja mobiilipelien markkinoita monetisaation ja asiakassegmentoinnin näkökulmasta. Libgdx-sovelluskehystä tarkastellaan kaksiulotteisen mobiilipelin kehityksen työkaluna. Teoriasuuden tutkimusten pohjalta tehdään suunnitelma, joka kattaa Mineral Digger -mobiilipelin konseptin ja toteutuksen.

## 2 MOBIILIPELIT

### 2.1 Menestyvän mobiilipelin ominaisuudet

Menestyäkseen mobiilipelin on oltava paljon muutakin kuin hauska pelata. Hyvän pelin lisäksi pelistudiot tarvitsevat korkean markkinointibudjetin saadakseen pelin käyttäjille. Sovelluskaupoissa on tuhansia hyviä pelejä, joita kukaan ei löydä, koska niiden tekijöiden markkinointibudjetti ei riitä kilpailemaan EA:n, Kings.comin ja Supercellin kaltaisten yritysten kanssa. Tuloksena on ikuisesti kasvava ”Kuollut alue”, joka koostuu mobiilipelien markkinoille hukuvista hyivistä mobiilipeleistä. (Grossberg 2014.)

Olettaen että pelistudion markkinointibudjetti on korkea, voidaan keskittyä muihin ominaisuuksiin, jotka saavat pelin menestymään. Näitä ominaisuuksia voidaan havainnollistaa tarkastelemalla menestyviä pelejä.

#### 2.1.1 Pelattavuus

lältään 2–12-vuotiaat lapset käyttävät mobiililaitettaan suurimmalta osin pelaamiseen. Tämä ikäryhmä myös pelaa useampia pelejä – myös maksullisia pelejä – verrattuna muihin ikäryhmiin. 2–12 -vuotiaiden keskimääräinen rahankäyttö uusien pelien ostoon ja pelinsisäisiin ostoihin on myös verrattaen korkea. Koska mobiilipelien pelaajista suuri määrä on lapsia, yksinkertaisuus on yleinen ominaisuus menestyvissä mobiilipeleissä. Esimerkiksi Halfbrick Studiosin kehittämä Fruit Ninja -peliä pelataan liu’uttamalla sormeja näytössä pomppivien hedelmien päältä. Toinen hyvä esimerkki on Rovion kehittämä Angry Birds -peli, jossa lintuja lingotaan kohteisiin vetämällä ritsassa olevaa lintua sormella taaksepäin. (NPD Group 2015, Marks 2013.)

Tavanomainen ominaisuus menestyvälle mobiilipelille on pelaamiseen liittyvät opasteet. Opasteet ovat yleensä visuaalisessa muodossa tekstinä tai nuolina, kuten Zeptolabin Cut the Rope -pelissä (Kuva 1.). Opasteiden tarkoitus on opettaa pelaajaa pelaamaan mahdollisimman nopeasti, ennen kuin pelaajan mielenkiinto loppuu. (Marks 2013.)



Kuva 1. Kuvakaappaus Cut The Rope -pelin opastenäkymästä.

### 2.1.2 Sisältö

Pelit kertovat tarinan. Yleensä peleissä määritellään sankari ja roisto sekä sankarin tavoite. Pelaajan tulee pelaamalla saattaa sankari tavoitteeseensa. Menestyvissä mobiilipeleissä on yleensä nähty vaivaa pelin tarinaan, jotta pelaajal-

le tulee halu viedä tarinaa eteenpäin. Rovion Angry birds -pelissä sankarina ovat linnut ja roistoina siat, jotka ovat vieneet linnuilta munia (Kuva 2.). Sankarin tavoitteena on saada munat takaisin roistolta. (Marks 2013.)



Kuva 2. Kuvakaappaus Angry Birds -pelin tarinasta.

Suurella määrällä lyhyitä tasoja pidetään pelaajan mielenkiinto yllä. Tason läpäisy tuo onnistumisen tunteen, ja kun tasot ovat lyhyitä ja niitä on paljon, onnistumisen tunteita tulee usein. Esimerkiksi Rovion Angry birds -pelissä on 495 tasoa ja jokaisen tason lopussa pelaaja palkitaan tähdillä. (Marks 2013, Rovio 2015.)

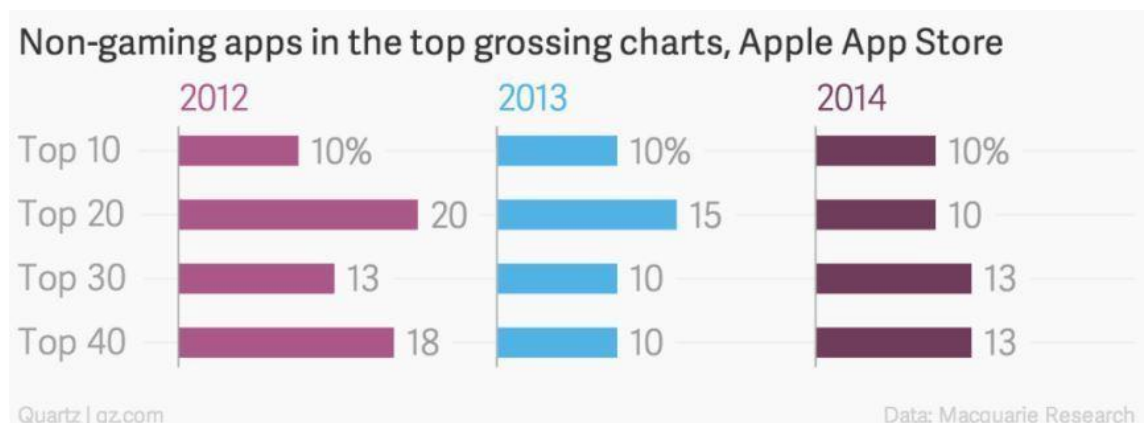
Menestyvissä mobiilipeleissä on usein mahdollisuus hankkia parannuksia, jotka auttavat pelissä etenemistä. Parannuksia voi ostaa pelinsisäisellä valuutalla, jota kerätään peliä pelaamalla tai oikealla rahalla ostamalla. Suomalainen peliyhtiö Supercell käyttää menestyneessä Clash of Clans -pelissään parannuksia. Parannukset ovat Clash of Clans -pelissä muun muassa rakennuksia, jotka puolustavat pelaajan tukikohtaa paremmin. Halfbrick Studiosin kehittämässä Jetpack Joyride -pelissä parannukset ovat muun muassa lisävarusteita, joilla pelin hahmo voi lentää helpommin pidemmälle. (Marks 2013.)

Pelin syvyydellä tarkoitetaan sen vaikeustason kasvua alusta loppuun edettäessä. Syvyyden avulla pelaaja kokee onnistumisen tunnetta heti alussa, kun peli ei ole liian vaikea. Vastaavasti pelin edetessä pelaajalle riittää haastetta, kun vaikeustaso nousee. (Marks 2013.) Esimerkiksi Zeptolabin Cut the Rope -pelin ensimmäisessä tasossa pelaajan tarvitsee ainoastaan pyyhkäistä sormella vihjeen osoittamasta paikasta päästäkseen tason läpi parhaimmilla pisteillä (Kuva 1.).

## 2.2 Markkinat

Vuonna 2013 ihmiset käyttivät mobiilipelaamiseen globaalisti noin 12,2 miljardia dollaria. Vuoteen 2012 verrattuna kasvua kertyi noin 34 %. Mobiilipelaajia on tällä hetkellä arviolta lähes 970 miljoonaa, joista noin 370 miljoonaa eli lähes 40 % käyttivät rahaa pelaamiseen. (Applift 2014.)

Pelit tuottavat eniten verrattuna muiden kategorioiden sovelluksiin sekä Applen App Storessa että Google Playssä (Today's Web 2014). Pelien suuresta osuudesta mobiilisovellusten markkinoilla kertoo myös Applen julkaisema Best of 2014 -lista, jonka mukaan yhdeksän kymmenestä eniten bruttotuloa tuottavista sovelluksista on pelejä (Kuva 3.).



Kuva 3. Kuva Applen julkaisemasta Best of 2014 -listasta (Today's Web 2014).

## 2.2.1 Monetisaatio

Useimmat mobiilisovellusten kehittäjät tekevät sovelluksia pääosalta intohimosta. Sovelluskehitys vaatii kuitenkin aikaa, työtä ja rahaa. Kehittäjän onkin tärkeää miettiä tapoja, minkä avulla sovellus alkaa tuottamaan. Monetisaatio tarkoittaa menetelmiä, joilla peli on tarkoitus tuottaa tuottamaan rahaa. Oikean monetisaatiostrategian valinta on vaikein, mutta myös tärkein askel sovelluksen menestymisen kannalta. Mobiilipelien monetisaatiostrategiat voidaan jakaa ilmaisiin peleihin, maksullisiin peleihin ja mobiilimainontaan. (Viswanathan 2015.)

### 2.2.1.1 Ilmainen peli

Ilmainen peli –strategiassa peli tarjotaan ilmaiseksi ja rahaa pyydetään pelinsisäisten ostojen kautta. Pelinsisäisillä ostoilla käyttäjät voivat ostaa uusia ominaisuuksia, päivityksiä, tasoja, tavaroita tai pelinsisäistä rahaa peliinsä. Saa-dakseen pelaajat tekemään pelinsisäisiä ostoja pelin pitää olla laadukas ja saada pelaaja sitoutuneeksi. (Francis 2013, Viswanathan 2015.)

Nykyään 91 % mobiilisovellusten markkinoiden sovelluksista on ilmaisia. Pelaajat valittavat entistä enemmän pelinkehittäjistä, jotka käyttävät pelinsisäisiä ostoja hyväkseen. Pelinkehittäjän on tärkeää löytää balanssi, jossa sovellus tuottaa rahaa ja pelaaja saa edetä pelissä mahdollisimman vähin häiriötekijöin. Tämän balanssin voi saavuttaa siten, että pelaajalle annetaan mahdollisuus pelin läpäisyyn käyttämättä lainkaan pelinsisäisiä ostoja – kuitenkin siten, että pelissä eteneminen on vaikeaa laittamatta siihen ollenkaan rahaa. (DevsBuild.it 2014.)

### 2.2.1.2 Maksullinen peli

Maksullinen peli -strategiassa pelille määritetään hinta. Peli voi tuottaa hyvin rahaa, mikäli se menestyy ja saavuttaa korkeita sijoituksia ranking-listoilla. Yleensä käyttäjät ovat kuitenkin valmiita maksamaan vain suosittujen kehittäjien peleistä. Lisäksi maksullinen peli -strategiassa kohdataan mobiilialustaan liittyvään ongelmaan: Android-käyttäjät eivät ole yhtä valmiita maksamaan peleistä kuin iOS-käyttäjät. On myös pidettävä mielessä, että sovelluskaupat vievät osuuden pelin tuottamista tuloista. (Francis 2013, Viswanathan 2015.)

Maksullinen peli -strategialla kehittäjät saavat kuitenkin rahaa jokaisesta saavuttamastaan pelaajasta. Kehittäjät voivat myös sisällyttää strategiaan pelinsisäisiä ostoja. Hankalinta maksullisten pelien monetisaatiossa on oikean hinnan määrittäminen. Hinta tulee määrittää pelin laadun perusteella. Konsolipelien tasoista grafiikkaa tarjoavat mobiilipelit voidaankin hinnoitella korkeammalle pelinsisäisiä ostoja myöten. (DevsBuild.it 2014.) Tästä esimerkkinä Infinity Blade -sarja (Kuva 4.).



Kuva 4. Kuvakaappaus Infinity Blade -pelistä.

### 2.2.1.3 Mobiilimainonta

Mobiilipeleissä mainonnalla on hyvät ja huonot puolensa. On kuitenkin selvää, että mainonta on yksi suosituimmista ja laajimmalti käytetyistä mobiilisovellusten monetisaatiostrategioista. Nykyään mobiilimainonnalle on tarjolla paljon erilaisia alustoja, joilla on omat ominaisuutensa ja etunsa. Useimmat kehittäjät kokeilevat erilaisia yhdistelmiä näistä alustoista ja valitsevat sitten ne, jotka toimivat parhaiten heidän peleissään. (Francis 2013, Viswanathan 2015.)

Bannerimainokset ovat suosituin mainosten muoto. Muihin mainostyyppeihin verrattuna bannerimainokset kuitenkin tekevät vähiten tuottoa. Niitä voidaan asettaa pelin käyttöliittymän sisään omavalintaiseen paikkaan, ja niitä on saatavilla eri kokoja. Välimainokset puolestaan sijoitetaan pelin käynnistykseen tai sammutukseen, ja ne ovat yleensä koko ruudun kokoisia ikkunoita. Välimainokset voivat sisältää kuvia, videoita tai linkkejä nettisivuille. Tarjousseinät ovat myös suosittu mainonnan muoto. Ne lähettävät käyttäjille koko ruudun kokoisia ikkunoita, jotka sisältävät tarjouksia, alennuksia ja uusien tasojen avausmahdollisuuksia ja pyytävät vastineeksi muiden sovellusten lataamista tai muihin uutiskirjeisiin rekisteröitymistä. (Francis 2013, Viswanathan 2015.)

Tutkimukset ovat nopeasti yleistynyt mobiilimainonnan muoto. Tutkimukset renderöidään omaksi kerroksekseen pelin sisään, minkä vuoksi pelin käyttöliittymää ei tarvitse muuttaa. Suorittamalla tutkimuksen loppuun käyttäjä osallistuu arvontaan ja sovelluskehittäjä saa maksun. (Viswanathan 2015.)

Ilmoitusmainokset lähettävät käyttäjälle mainoksia, vaikka käyttäjä ei pelaisi peliä. Tällaiset mainokset tuovat kehittäjälle enemmän rahaa, mutta ne ovat myös tunkeilevia ja voivat häiritä käyttäjää. (Viswanathan 2015.)

## 2.2.2 Asiakassegmentit

Segmentointi on markkinointistrategia, jossa laajat kohdemarkkinat jaetaan useampiin segmentteihin erilaisten asiakkaiden erityyppisten tarpeiden tyydyttämiseksi (e-conomic 2015). Markkinoijat ovat jo kauan käyttäneet asiakassegmentointia jakaakseen asiakkaat hallinnoitaviin lokeroihin ja analysoidakseen heidän käytöstään. Asiakassegmentoinnin avulla johdettuja käsityksiä käytetään kohdistettujen kampanjoiden pitämiseen ja tällä tavalla suurempien tulojen saavuttamiseen. (Lele 2015.) Mobiilipelin asiakkaat voidaan segmentoida asiakkaan kuluttamiskäyttäytymisen perusteella. Tällaista segmentointitapaa käyttäen asiakkaat jakautuvat kolmeen ryhmään: innokkaat maksajat, vastahakoiset maksajat ja ei-maksajat. (Seufert 2012.)

### 2.2.2.1 Innokkaat maksajat

Mobiilipelin pelaajista pienin osa kuuluu innokkaisiin maksajiin. Tähän ryhmään kuuluvat käyttäjät tekevät pelinsisäisiä ostoja lähes välittömästi pelin hankittuaan. Tällaisille asiakkaille pitää jatkuvasti antaa mahdollisuus uusien ostojen tekemiseen esimerkiksi tarjoamalla tavaroita pelihahmon ulkoasun paranteluun. Kun uutta ostettavaa on tarjolla, innokkaat maksajat pysyvät tyytyväisinä ja jatkavat ostojen tekemistä. (Seufert 2012.)

### 2.2.2.2 Vastahakoiset maksajat

Toiseksi pienin osa pelaajista kuuluu vastahakoisiin maksajiin. Tällaiset asiakkaat tekevät ostoja ainoastaan silloin, kun niitä tarjotaan alennuksen tai ainutlaatuisen tilaisuuden kanssa. Vastahakoiset maksajat tarttuvat yleensä alennusmyynteihin ja yhdistelmäpaketteihin. Tämän segmentin pelaajat tekevät yleensä ensimmäisen ostonsa pelin puolella välissä läpäistyään jo useamman hankalan kohdan. (Seufert 2012.)

### 2.2.2.3 Ei-maksajat

Suurin osa mobiilipelin pelaajista kuuluu ei-maksajiin, jotka eivät maksa pelaamisesta missään vaiheessa. Tämä ei kuitenkaan tarkoita sitä, että ryhmän asiakkaat eivät pidä pelistä. Esimerkiksi lapsilla ei välttämättä ole mahdollisuutta rahan käyttöön. Jotkut myös haluavat välttää virtuaalisten tavaroiden ostajan leimaa. Tällaisista asiakkaista voidaan kuitenkin saada tuottoa mainosten avulla: jos asiakas pitää pelistä riittävästi, hänen pitäisi sietää pelaamisen yhteydessä ilmestyvät mainokset. (Seufert 2012.)

### 3 LIBGDX

Libgdx on avoimen lähdekoodin cross-platform-sovelluskehys, joka tarjoaa yhtenäisen ohjelmointirajapinnan pelinkehitykseen Java-ohjelmointikielellä. Libgdx käyttää paljon myös C-ohjelmointikieltä suorituskykykriittisiin tehtäviin, yhdistääkseen muita C-pohjaisia kirjastoja ja mahdollistaakseen cross-platform-ominaisuuden. Cross-platform-ominaisuudellaan Libgdx tukee seuraavia alustoja: Windows, Linux, Mac OS X, Android (2.2+), BlackBerry, iOS, Java Applet, Javascript/WebGL (Chrome, Safari, Opera, Firefox, IE via Google Chrome Frame). (Libgdx 2014, Oehlke 2013.)

Libgdx tarjoaa ympäristön nopeaan prototyyppien tekoon ja iterointiin. Sen sijaan, että ohjelmakoodi pitäisi aina erikseen kääntää alustakohtaiseen muotoon, se voidaan ajaa suoraan tietokoneen työpöydällä. Ohjelmakoodin ajaminen työpöydällä mahdollistaa Java Virtual Machinen tarjoamien helpottavien funktioiden käytön, kuten Code Hot Swapping. Code Hot Swapping antaa käyttäjän tehdä muutoksia koodiin, vaikka peli on käynnissä. Tämä ominaisuus vähentää huomattavasti ideoiden kokeiluun ja virheiden korjaamiseen kuluvaa aikaa. (Oehlke 2013.)

Libgdx ei kuitenkaan ole pelimoottori. Se ei tarjoa pelimoottorille tavanomaisia työkaluja, kuten tasomuokkainta. Päinvastoin kuin pelimoottorit yleensä, Libgdx ei myöskään edellytä ennalta määritetyn työskentelytavan noudattamista, mikä vapauttaa käyttäjän määrittämään oman työskentelytavan eri projekteihin. (Oehlke 2013.)

Libgdx sisältää laajan valikoiman ominaisuuksia erilaisten pelien tekoon. Tarkoitus on, että käyttäjä valitsee näistä ominaisuuksista peliinsä sopivimmat. Seuraavaksi tarkastellaan opinnäytetyössä suunniteltavan kaksiulotteisen mobiilipeilin kannalta oleellisia ominaisuuksia.

### 3.1 Scene2D

Libgdx-sovelluskehiksen mukana tulee Scene2D-ohjelmointirajapinta skene-kaavioiden luomiseen. Skene-kaavio on hierarkisesti järjestelty rakennelma, joka koostuu samankaltaisista objekteista kuin tiedostot ja kansiot tietokoneen kovalevyllä. Libgdx:ssä tällaisia objekteja kutsutaan Actor-objekteiksi. Actor-objektit voidaan laittaa sisäkkäin, jolloin syntyy loogisia ryhmiä. Ryhmien luonti on kätevä ominaisuus, sillä Actor-objektiin tehdyt modifikaatiot vaikuttavat myös objektin sisään laitettuihin Actor-objekteihin. Lisäksi jokaisella Actor-objektilla on oma paikallinen koordinaatistosysteemi, minkä avulla on helppo määrittää suhteellisia siirtymiä, kuten sijaintia ja kokoa, Actor-objektiryhmän sisällä. Actor-objektit tunnistavat myös kosketuksen toisiin Actor-objekteihin ja käyttäjän tekemät interaktiot. (Libgdx 2014, Oehlke 2013.)

Actor-objekteja voidaan manipuloida ajan suhteen sisäänrakennetun action-systeemin avulla. Actor-objekteihin voidaan luoda komplekseja efektejä, jotka alkavat jaksoissa, samanaikaisesti tai molempien yhdistelmänä. Kaikki edellämainittu toiminnallisuus on kapseloitu Stage-luokkaan, joka sisältää objektien hierarkian ja jakaa käyttäjän tekemät interaktiot. Actor-objekteja voidaan lisätä tai poistaa Stage-objektista milloin tahansa. Stage-luokka ja Actor-luokka sisältävät molemmat `act()`-metodin, joka suorittaa jonkin aikaan perustuvan action-systeemin toiminnon. Kutsumalla Stage-objektin `act()`-metodia kutsutaan samalla kaikkien Stage-objektin skene-kaaviossa olevien Actor-objektien `act()`-metodia. (Libgdx 2014, Oehlke 2013.)

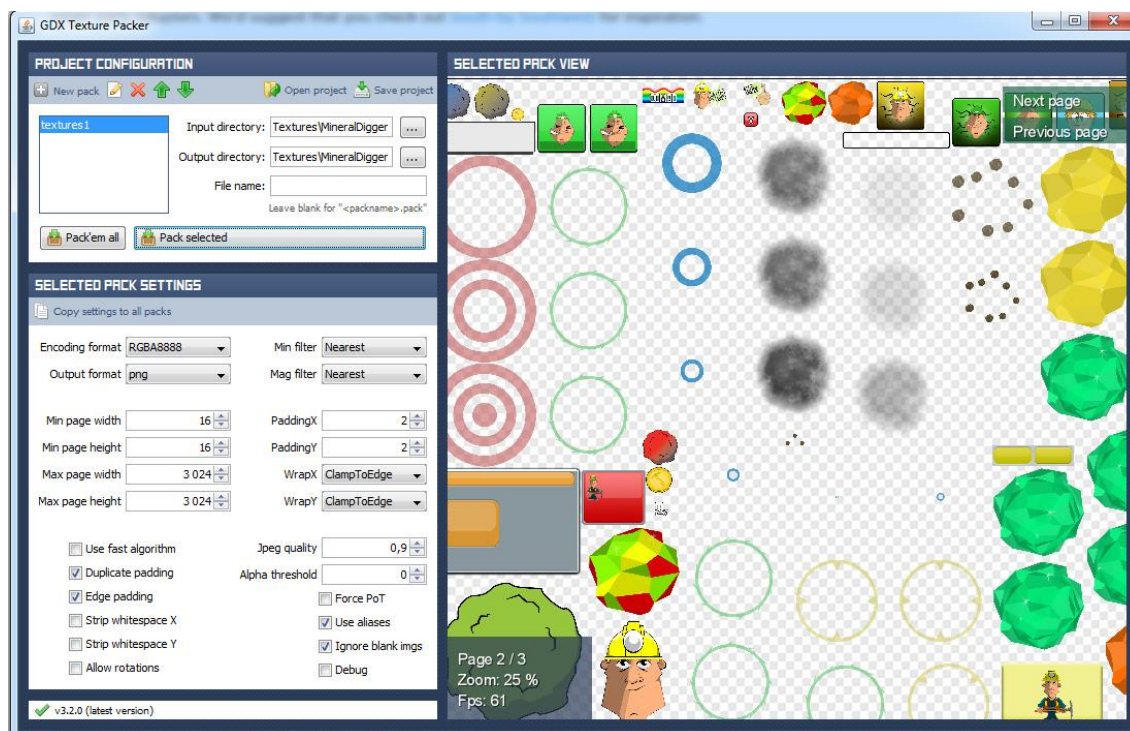
Scene2D on tehokas työkalu pelinkehitykseen. Stage-objekti vastaa peliruutua ja Actor-objektit pelin hahmoja. Hahmoja voidaan helposti liikuttaa koordinaattisysteemin avulla ja niille voidaan luoda monimutkaisia efektejä action-systeemin avulla. Scene2D sisältää myös kätevän ominaisuuden pelin valikoiden luomiseen. Tämän ominaisuus on Scene2D UI, ja se toimii jatkeena Sce-

ne2D ohjelmointirajapinnalle tarjoamalla laajan valikoiman UI-elementtejä, kuten nappeja, kuvia, valintalaatikoita, leimoja, slider-elementtejä ja ikkunoita. Kaikki UI-elementit ovat myös Actor-objekteja, joten niille voi määrittää erilaisia efektejä tai koota ryhmiksi. (Libgdx 2014, Oehlke 2013.)

### 3.2 Tekstuuriatlas

Tekstuuriatlas (texture atlas) on tavallinen kuvatiedosto, joka voidaan renderöidä peliruudulle samalla tavalla kuin muutkin kuvat. Sitä käytetään kuitenkin kuvasäiliönä, joka sisältää useista pienempiä kuvia järjestettynä niin, että ne eivät ole päällekkäin. Tekstuuriatlaksen avulla vähennetään grafiikkaprosessorille lähetettävien kuvien määrää, mikä parantaa huomattavasti renderöintitehokkuutta. Tekstuuriatlakset ovat erityisen käytännöllisiä peleihin, joissa renderöidään paljon pieniä ja erilaisia kuvia samanaikaisesti, koska erilaisten tekstuurien välillä vaihtelu on hyvin raskas prosessi. Aina kun tekstuuria vaihdetaan renderöitäessä, uutta dataa täytyy lähettää videomuistiin. Tältä vältytään, kun kaikkeen käytetään samaa tekstuuria eli tekstuuriatlasta. (Libgdx 2014, Oehlke 2013.)

Libgdx mahdollistaa tekstuuriatlaksen luomisen ohjelmallisesti, jolloin peliprojektiin lisätyt kuvat pakkautuvat atlakseen automaattisesti, kun peli ajetaan. Tekstuuriatlaksen luominen ohjelmallisesti toimii hyvin, mutta käyttäjä ei näe suoraan atlaksen asetuksiin tehtyjen muutosten tulosta. Libgdx:n mukana tulee kuitenkin Aurélien Ribonin kehittämä TexturePacker-GUI, joka on erillinen ohjelma tekstuuriatlaksen luomiseen. TexturePacker-GUI näyttää juuri luodun tekstuuriatlaksen kuvatiedostona ja tarjoaa graafisen käyttöliittymän asetusten säätämiseen (Kuva 5.). (Libgdx 2014, Oehlke 2013.)



Kuva 5. Kuvakaappaus Texturepacker-GUI:n asetuskäytännöstä.

### 3.3 Ortografinen kamera

Libgdx:ssä kaikki mitä peliin piirretään, tehdään näkyväksi kameran avulla. Kameran sijaintia ja lähennystä voidaan muuttaa. Kaksiulotteiseen projektointiin sopii Libgdx:n tarjoama ortografinen kamera. Ortografinen kamera on pelaajan 2D-näkymä pelistä, ja sen koko määritellään leveytenä ja korkeutena ohjelmallisesti. (Libgdx 2014, Oehlke 2013.)

### 3.4 Audio

Musiikin ja äänien toistoon Libgdx tarjoaa kaksi rajapintaa, Music ja Sound, joista kummallakin on oma käyttötarkoituksensa. Libgdx:ssä äänet ovat noin se-

kunnin mittaisia tai lyhyempiä äänitiedostoja, esimerkiksi aseiden laukauksen ääni. Ääninä käytettävät äänitiedostot dekodataan jo silloin, kun äänitiedostot ladataan peliin. (Libgdx 2014, Oehlke 2013.)

Kun äänitiedostoja käytetään puolestaan musiikkina, ne dekodataan musiikin toiston yhteydessä, ja niistä pidetään muistissa ainoastaan tarvittava määrä. Äänitiedostoja tulee käyttää musiikkina silloin, kun toistetaan pitkiä äänitiedostoja, kuten taustamusiikkia. Vastaavasti äänitiedostoja tulee käyttää ääninä silloin, kun kyseessä on lyhyt äänitehoste, koska äänitehosteita saatetaan käyttää toistuvasti ja musiikkina käytettäessä ne jouduttaisiin dekodamaan aina toistettaessa. (Libgdx 2014, Oehlke 2013.)

## 4 MINERAL DIGGER -PELI

Opinnäytetyössä suunniteltiin mobiilialustoille suunnattu peli, Mineral Digger. Mineral Digger -peliä suunniteltaessa pidettiin mielessä opinnäytetyön teorioosuudessa tutkittuja menestyvän mobiilipelin ominaisuuksia. Pelin pelaamisesta haluttiin tehdä mahdollisimman yksinkertaista ja alkuun suunniteltiin tutoriaali, joka opastaa pelaajaa pelaamaan. Monen lyhyen tason sijaan pelissä pelataan yhtä loputonta tasoa, joka muuttuu tasaisesti vaikeammaksi pelin edetessä. Pelaajan onnistumisen tunteita haetaan tason aikana edettyjen vaiheiden ja uuden sisällön saavuttamisen avulla. Lisäksi peli tarjoaa paljon erilaisia parannuksia eri muodoissa.

Pelin monetisaatioon valittiin ilmainen peli -strategia. Pelin voi ladata ilmaiseksi, ja siinä voi edetä maksamatta mitään. Etenemistä helpottamaan pelaajalle tarjotaan erilaisia ominaisuuksia, joita pelaaja voi hankkia pelinsisäisillä ostoilla. Ilmainen peli -strategiaan päädyttiin, koska vain pieni osa mobiilipelien pelaajista on valmis maksamaan pelistä ja tässä peliprojektissa tärkeimpänä tavoitteena oli saada asiakkaat vähintäänkin lataamaan peli. Ilmainen peli -strategia oli sopivin vaihtoehto myös pelin ominaisuuksien kannalta: jos kaikki pelin ominaisuudet tarjottaisiin heti pelin oston yhteydessä, pelaaja pääsisi pelin läpi lähes välittömästi.

### 4.1 Idea

Pelin pohjana oli Orteil-yhtiön julkaisema yksinkertainen, mutta koukuttava Cookie Clicker -selainpeli. Cookie Clicker julkaistiin 10. elokuuta 2013 ja se on toteutettu JavaScript-ohjelmointikielellä. Cookie Clickerin idea on valmistaa keksejä klikkaamalla suurta keksin kuvaa kunnes keksejä on tarpeeksi parannuksien ostamiseen. Pelissä voi edistyä loputtomasti, ja se kestää niin kauan kuin pelaajan laitteisto jaksaa sen kuormitusta. (Cookie Clicker Wiki 2014.)

Mineral Digger -pelin perusidea otettiin Cookie Clickeristä: loputon klikkailu ja sitä kautta edistyminen. Keksin sijaan Mineral Diggerissä klikataan kiveä ja tietokoneen hiiren sijaan käytetään omaa sormeaa. Pelaajan tavoitteena on mobiililaitteen näyttöä painamalla hakata pelissä näkyvää isoa kivenlohkareta ja tienata tällä tavalla pelinsisäistä rahaa. Ruudun alareunassa näkyy kiven kestävyys palkkina, joka pienenee aina kun kiveä hakataan. Kun palkki katoaa näkyvistä, kivi tuhoutuu ja kiven sisältä paljastuu mineraali. Mineraaleista pelaaja saa lisää pelinsisäistä rahaa. Kun kivi on tuhoutunut ja mineraali on kerätty, ruutuun ilmestyy uusi kivi, jota pelaaja saa taas alkaa tuhoamaan. Hankkimillaan pelinsisäisillä rahoilla pelaaja voi ostaa työläisiä, uusia kiviä ja parannuksia.

## 4.2 Ominaisuudet

Samasta perusideasta huolimatta Mineral Digger eroaa innoittajastaan, Cookie Clickeristä huomattavasti. Mineral Diggerissä klikkailun päämääränä on pelinsisäisen rahan keräämisen lisäksi mineraalien löytäminen. Peliin tehtiin myös ominaisuuksia, jotka erottavat sen Cookie Clicker -pelistä ja vastaavanlaisista kilpailijoista sekä mahdollistavat pelin monetisaation.

Omaperäisempään suuntaan peliä vie rahaa vastaan hankittavat työläiset, jotka nopeuttavat pelin edistymistä hakkaamalla omatoimisesti kiveä. Pelissä on painike, jota painamalla tulee esiin valikko työläisistä. Valikossa työläiset esitetään painikkeina, joissa näkyy kuva työläisestä, ostettujen työläisten määrä ja työläisen hinta. Mikäli pelaaja on tienannut tarpeeksi pelinsisäistä rahaa, painamalla työläisen painiketta hän saa kyseisen työläisen avukseen kiven hakkaamiseen. Kullakin työläisellä on oma nopeutensa ja voimansa. Nopeus määrittää, kuinka tiheään tahtiin työläinen lyö kiveä. Voima määrittää, kuinka paljon yksittäinen lyönti vie hakattavan kiven kestävydestä. Nopeus-voimasuhteeltaan tehokkaammat työläiset maksavat vastaavasti enemmän.

Pelissä hakattavalla kivellä on ominaisuutena kestävyys sekä lista erilaisista mineraaleista, joita pelaaja voi kiven sisältä löytää. Kestävyys määrittää, kuinka kauan kivi kestää hakkaamista ennen kuin se tuhoutuu. Pelissä on valikko, josta

voi ostaa kestävämpiä – ja vastaavaasti kalliimpia – kiviä. Kestävämpien kivien sisältä voi löytyä harvinaisempia ja arvokkaampia mineraaleja.

Pelissä on 100 toinen toistaan arvokkaampaa mineraalia, joilla on ominaisuuksina harvinaisuus, arvo ja koko. Kiven sisältä löytyvä mineraali määritty hakattavan kiven, mineraalin harvinaisuuden ja satunnaismuuttujan perusteella. Mineraalin paljastuttua kiven sisältä pelaaja saa pelinsisäistä rahaa mineraalin arvon ja koon perusteella.

Parannukset ovat monista muistakin peleistä tuttu ominaisuus, joka auttaa pelaajaa etenemään pelissä nopeammin. Parannuksille on Mineral Diggerissä oma valikko, johon on listattu kaikki pelin parannukset painikkeiden muodossa. Kullakin parannuksella on oma kuva ja nimi, jotka selittävät parannuksen vaikutusta. Parannuksen hinta määräytyy sen vaikutusten mukaan. Esimerkki parannuksesta on tiettyjen työläisten tehokkuuden kaksinkertaistuminen.

Kaikki pelin ominaisuudet, joita pelaaja voi hankkia pelinsisäisellä rahalla, kuuluvat pelin monetisaatiostrategiaan. Mikäli pelaaja yrittää hankkia ominaisuutta, johon pelinsisäiset rahat eivät vielä riitä, peliruutuun avautuu ikkuna, jossa tarjotaan mahdollisuus pelinsisäisen rahan ostoon oikealla rahalla. Ikkunassa on lista erilaisista summista pelinsisäistä rahaa, ja niitä vastaavat hinnat oikeassa rahassa määritettynä. Pelaajan tulee antaa luottokorttitietonsa puhelimensa käyttöjärjestelmän sovelluskauppaan, jos hän haluaa käyttää oikeaa rahaa pelinsisäisen rahan hankkimiseen. Kun pelaaja on valinnut haluamansa summan pelinsisäistä rahaa, summaa vastaavan oikean rahan veloitus tapahtuu automaattisesti luottokortin tietojen mukaiselta tililtä ja pelinsisäinen rahasumma lisätään peliin.

### 4.3 Pelaaminen

Pelaamisessa pääpainona on yksinkertaisuus. Pelissä kaikki tapahtuu samalla peliruudulla, eli pelinsisäisiä siirtymiä näkymästä toiseen tai latausvaiheita ei ole. Peliä voi pelata yhtä sormea käyttäen naputtamalla sitä mobiililaitteen ruudussa samaan paikkaan. Tarkoitus onkin, että peliä voi pelata lähes ajattelematta, ja silti edistystä tapahtuu jatkuvasti. Koska edistystä tapahtuu jokaisella sormen painalluksella ja pelaaminen on helppoa, pelaajan on vaikeaa lopettaa pelaamista napauttamatta ruutua vielä kerran.

#### 4.3.1 Toiminnot

Mineral Digger -peliä pelataan koskettamalla ruudussa olevaa kivenlohkarettä toistuvasti sormella. Jokainen sormen kosketus tuhoaa kiveä ja tuo pelaajalle pelinsisäistä rahaa. Ruudussa on kiven lisäksi alareunassa kolme painiketta: Units, Rocks ja Upgrades. Units-painike avaa valikon, josta voi ostaa työläisiä, Rocks-painike avaa valikon, josta voi ostaa kiviä ja Upgrades-painike avaa valikon, josta voi ostaa parannuksia. Kun yhtä näistä kolmesta painikkeista painetaan, muut painikkeet siirtyvät pois näkyvistä. Kun painettua painiketta painetaan uudelleen, auennut valikko katoaa ja muut painikkeet tulevat näkyviin. Valikot aukeavat peliruudun alareunasta siten, että ruudussa olevaa kivelohkarettä voi tuhota, vaikka jokin valikoista olisi auki.

#### 4.3.2 Eteneminen

Peli alkaa tutoriaalista, jossa ruutuun putoaa kivi ja kiven yläpuolelle ilmestyy pää ja puhekupla. Puhekuplassa on tekstiä, joka neuvoo koskettamaan ruudussa olevaa kiveä muutaman kerran sormella. Tutoriaalin kivi on todella heikko, ja jo muutaman kosketuksen jälkeen kivi tuhoutuu. Kiven tuhouduttua näkyviin

tulee mineraali, josta pelaaja saa tarpeeksi rahaa työläisen ostamiseen. Kiven tuhoututtua myös puhekuplassa teksti muuttuu, ja tällä kertaa se ohjeistaa pelaajaa ostamaan työläisen. Tuhoutuneen kiven paikalle ilmestyy uusi kivi, joka ei ole enää tutoriaalinen heikko kivi. Kun ensimmäinen työläinen hankitaan, tutoriaali loppuu ja pää puhekuplineen katoaa.

Tutoriaalinen jälkeen pelaaja tietää pelin perusrakenteen ja voi alkaa omatoimisesti tuhoamaan ruudussa näkyvää uutta kiveä. Tutoriaalissa hankittu työläinen auttaa kiven hakkaamisessa tietyin väliajoin. Pelaajan seuraavat tavoitteet ovat kiven tuhoaminen, pelinsisäisen rahan kerääminen ja uusien työläisten hankinta.

Tutoriaali ei opasta pelaajaa vielä pelin alussa uusien kivien tai parannusten hankintaan, koska liian pitkä tutoriaali voi käydä pelaajalle ärsyttäväksi. Vasta kun pelaaja painaa ensimmäistä kertaa Rocks- tai Upgrades-painiketta, tutoriaalista tuttu pää ja puhekupla ilmestyvät opastamaan kivien tai parannusten hankinnassa. Kun viimeistäkin painiketta on painettu, pelaaja tietää koko pelin kulun. Pelaajalle jää päätettäväksi, missä järjestyksessä hän hankkii uusia kiviä, työläisiä ja parannuksia.

#### 4.4 Toteutussuunnitelma

Mineral Digger -pelin toteutussuunnitelmassa yhdistettiin Libgdx-sovelluskehityksen tarjoamat kaksiulotteisen mobiilipelin kehitykseen sopivat työkalut ja huolellisesti laadittu pelisuunnitelma. Toteutussuunnitelma koostuu ulkoasullisen ja rakenteellisen toteutuksen suunnittelusta.

##### 4.4.1 Ulkoasu

Pelin toteutuksen pohjaksi luotiin malli sen ulkoasusta Adobe Photoshop -kuvankäsittelyohjelmalla. Kuvankäsittelyohjelmasta ulkoasun eri elementit tallennetaan yksittäisiksi kuvatietostoiksi samaan kansioon. Kuvista luodaan

Libgdx-sovelluskehiksen mukana tulevalla TexturePacker-GUI-ohjelmalla tekstuuriatlas, joka lisätään peliprojektiin.

Kuvankäsittelyohjelmalla tehty malli toteutetaan pelissä käyttämällä Libgdx:n Scene2D-ohjelmointirajapintaa. Aluksi peliruutuun luodaan pohjalle Stage-objekti, jonka sisään laitetaan kaikki pelin perusnäkyssä olevat elementit (Kuva 7.). Stage-objektiin lisätään elementtien jäsentelyä helpottamaan koko peliruudun kattava Table-elementti eli taulukko, jonka taustaksi määritetään tekstuuriatlaksesta pelin taustakuva.



Kuva 6. Kuvankäsittelyohjelmalla tehty malli Mineral Digger -pelin perusnäkömängstä.

Kuvankäsittelyohjelmalla tehdyn mallin toteutus aloitetaan graafisesta käyttöliittymästä, joka pitää sisällään pelin painikkeet, valikot, pelinsisäisen rahan, kiven kestävyyspalkin ja tutoriaalın elementit. Graafinen käyttöliittymä toteutetaan

hyödyntämällä Scene2D UI:n tarjoamia UI-elementtejä. UI-elementit ovat Actor-objekteja, joten niillä on oma kosketuksen tunnistus, koordinaattisysteemi ja action-systeemi. Pelin alareunan kolme painiketta ovat ImageTextButton-elementtejä. ImageTextButton-elementille voi määrittää taustakuvan, ja sen sisään voi lisätä kuvia ja tekstejä. Alareunan painikkeille määritetään tekstuuriatlaksesta sama painikkeen tausta. Kuhunkin painikkeeseen lisätään painikkeen avaaman valikon sisältöä kuvaava ikoni Image-elementtinä ja teksti Label-elementtinä.

Kukin alareunan painike avaa oman valikkonsa, mutta jokainen valikko toteutetaan samalla tavalla. Valikon pohjaksi luodaan Stage-objekti, jonka sisään laitetaan Table-elementti. Table-elementtiin lisätään ScrollPane-elementti, mikä tekee sisällöstä vieritettävän, mikäli se ei mahdu ruutuun. ScrollPane-elementtiin lisätään valikon sisällöstä riippuen painikkeet kullekin työläiselle, kivelle tai parannukselle. (Kuva 8.) Kaikkien kolmen valikon painikkeet toteutetaan samalla tavalla: painikkeen pohjaksi laitetaan Table-elementti, jonka taustaksi valitaan kuva tekstuuriatlaksesta. Table-elementtiin lisätään tekstiä Label-elementin avulla. Label-elementeillä ilmoitetaan esimerkiksi parannuspainikkeen tapauksessa parannuksen hinta.



Kuva 7. Kuvankäsitteluohjelmalla tehty malli Mineral Digger -pelin kivien valikosta.

Label-elementtiä käytetään myös graafisen käyttöliittymän yläreunassa olevan pelinsisäisen rahan määrän näyttämiseen. Label-elementti laitetaan Table-elementtiin, jonka taustaksi määritetään harmaa palkki ja sijainniksi ruudun ylä-

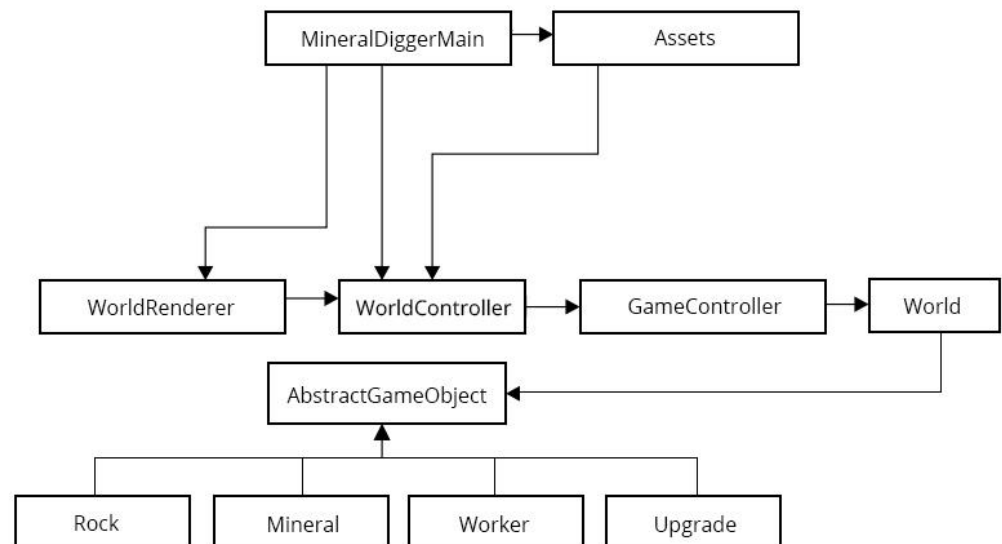
reuna. Table-elementtiin lisätään vielä rahan määrän näyttävän Label-elementin oikealle puolelle Image-elementtinä kuva pelinsisäisestä rahasta.

Pelin tutoriaali toteutetaan valikoiden tapaan myös omalle Stage-objektille. Stage-objektiin laitetaan Table-elementti, johon lisätään Image-elementtinä tutoriaalissa esiintyvä pää. Table-elementtiin lisätään myös toinen Table-elementti, jonka taustaksi määritetään puhekupla. Tekstiä puhekuplaan laitetaan Label-elementin avulla.

Graafisen käyttöliittymän lisäksi pelissä hakattava kivi toteutetaan Scene2D UI:n avulla. Kivelle luodaan oma luokkansa pelaamiseen liittyviä funktiota ja arvoja varten, mutta ulkoasullisesti kivi toteutetaan Image-elementtinä. Koska Image-elementti on Actor-objekti, kivi tunnistaa käyttäjän sormen kosketuksen. Actor-objektin action-systeemin avulla kivelle luodaan efekti, joka kosketettaessa ensin suurentaa ja sitten pienentää kiveä. Sen lisäksi että kiven kosketusefekti tuo peliin eloa, se myös indikoi käyttäjän kosketusta. Action-systeemin efektejä käytetään myös valikoiden avautumisessa. Valikon avaavaa painiketta painettaessa käynnistyy efekti, jossa piilossa olevan valikon y-koordinaattia muutetaan niin, että valikko tulee ruudun alareunasta näkyviin. Kun samaa painiketta painetaan uudelleen, efekti on päinvastainen. Valikon efektiin lisätään vielä Libgdx:n tarjoama interpolaatio-algoritmi, joka muokkaa efektin suoritus-  
tahtia tehden efektistä eloisamman.

#### 4.4.2 Rakenne

Pelin rakenteellisen toteutuksen pohjaksi tehtiin kaavio, joka havainnollistaa pelin luokkia ja niiden hierarkiaa. Kaavion ylimpänä luokkana on MineralDiggerMain, jonka Libgdx tekee automaattisesti, kun peliprojekti luodaan. Libgdx aloittaa pelikoodin lukemisen MineralDiggerMain-luokasta. MineralDiggerMain-luokka käsittelee käyttäjän interaktiot ja luo objektit Assets-, WorldRenderer-, WorldController-, GameController- ja World-luokista. (Kuva 9.)



Kuva 8. Kaavio pelin luokista.

Assets-luokka sisältää kaikki pelissä tarvittavat visuaaliset ja auditiiviset materiaalit. Visuaalinen materiaali on Assets-luokassa pakattuna tekstuuriatlakseeseen, jonka kautta peliin haetaan kuvat. Auditiivinen materiaali on Assets-luokassa määriteltynä joko Music-luokan objekteiksi tai Sound-luokan objekteiksi riippuen äänen käyttötarkoituksesta.

WorldRenderer-luokassa tapahtuu pelin renderöinti. Luokassa luodaan pelissä käytettävä ortografinen kamera, joka saa pelin näkymään kaksiulotteisesti. Lisäksi WorldRenderer-luokassa luodaan ja päivitetään kaikki pelissä käytettävät Scene2D-objektit.

WorldController-luokka sisältää pelin logiikat. Luokassa suoritetaan kiven hakkaamisesta seuraava kestävyuden pienentäminen ja pelinsisäisen rahamäärän

kasvattaminen sekä päivitetään pelissä hankittuja työläisiä. WorldController-luokassa myös tarkistetaan, onko kiven sisältä paljastuva mineraali jo löydetty.

GameController-luokassa suoritetaan pelin etenemisen kannalta merkittäviä tehtäviä. GameController-luokassa suoritetaan funktio, joka arpoo pelissä hakattavan kiven mineraalilistasta seuraavaksi ilmestyvän mineraalin. Lisäksi luokassa on listat jo löydetyistä mineraaleista ja hankituista parannuksista.

World-luokassa luodaan pelin maailma. Pelin maailma koostuu Rock-, Mineral-, Worker- ja Upgrade-luokan objekteista. Rock-luokka vastaa pelin kiveä, Mineral-luokka mineraalia, Worker-luokka työläistä ja Upgrade-luokka parannusta. World-luokassa on lista peliin hankituista työläisistä ja objektina pelissä hakettava kivi. Kun pelin valikoista hankitaan uusi kivi tai työläinen, World-luokan työläisten listaan lisätään uusi työläinen tai kivi-objekti päivitetään uuteen kiveen.

Pelin maailman objektit eli kivet, mineraalit, työläiset ja parannukset periytyvät kaikki AbstractGameObject-luokasta, joka on abstrakti. Tämä mahdollistaa kaikkien pelin maailman objektien säilyttämisen samassa listassa World-luokassa. Kun objektit ovat samassa listassa, niitä on helpompi hallinnoida silmukoissa. Esimerkiksi jokaisella pelin maailman objektilla on funktio, joka piirtää objektin näkyviin. WorldRenderer-luokassa haetaan World-luokasta lista, joka sisältää kaikki pelin maailman objektit ja kutsutaan listan objektien piirto-funktiota for-silmukassa.

Kaikilla pelin maailman objekteilla on muuttujina koko ja sijainti sekä Image-elementti, joka näkyy pelin ruudussa. Rock-luokassa on lisäksi muuttujina kiven hinta, kestävyys ja lista kaikista mahdollisista kiven sisältä paljastuvista mineraaleista. Mineral-luokassa muuttujia ovat arvo ja harvinaisuus ja Worker-luokassa hinta, voimakkuus ja nopeus. Upgrade-luokassa muuttujia ovat hinta, nimi ja kuvaus. Lisäksi Upgrade-luokassa on funktio, joka suoritetaan, kun parannus ostetaan.

## 5 YHTEENVETO

Opinnäytetyössä tehdyn tutkimuksen avulla saatiin selkeästi eriteltyä menestyvien mobiilipelien ominaisuuksia niin pelattavuudeltaan kuin sisällöltäänkin. Myös mobiilipelien markkinoihin paneuduttiin tutkimuksella mobiilipelien moneti-  
saatiosta ja asiakassegmentoinnista. Työssä tutkittiin lisäksi Libgdx-sovelluskehystä ja sen soveltuvuutta kaksiulotteisen mobiilipelin kehitykseen erittelemällä sen ominaisuuksia.

Opinnäytetyön teoriaosuuden tutkimusten avulla luotiin kattava ja yksityiskoh-  
tainen suunnitelma kaksiulotteisesta Mineral Digger -mobiilipelistä ja sen toteu-  
tuksesta. Suunnitelmassa käydään läpi pelin idea, ominaisuudet ja pelaaminen  
ja suunnitellaan pelin toteutus sekä ulkoasullisesti että rakenteellisesti. Mobiili-  
peleistä tehdyn tutkimuksen avulla peliin suunniteltiin sopivat ominaisuudet ja  
markkinointistrategia. Libgdx-sovelluskehystä tehty tutkimus puolestaan sel-  
vensi pelin käytännön toteutuksen suunnittelua. Seuraava vaihe Mineral Dig-  
ger -pelin kehityksessä on käytännön toteutus, johon voidaan nyt siirtyä helpos-  
ti, koska peli on suunniteltu huolellisesti sekä konseptiltaan että toteutukseltaan.

## LÄHTEET

Applift 2014. The Global Mobile Games Landscape 2014. Viitattu 1.3.2015 <http://www.applift.com/The-Global-Mobile-Games-Market.pdf>.

Cocos2D 2015. About. Viitattu 10.04.2015 <http://www.cocos2d-x.org/wiki/Cocos2d-JS>.

Cookie Clicker Wiki 2014. What is Cookie Clicker? Viitattu 1.3.2015 [http://cookieclicker.wikia.com/wiki/Cookie\\_Clicker\\_Wiki](http://cookieclicker.wikia.com/wiki/Cookie_Clicker_Wiki).

Corona Labs 2015. Corona SDK. Viitattu 10.4.2015 <https://coronalabs.com/products/corona-sdk>.

DevsBuild.it 2014. How to Monetize Your Game (Without Upsetting Your Users). Viitattu 10.03.2015 <http://devsbuild.it/content/How-Monetize-Your-Game-Without-Upsetting-Your-Users>.

e-conomic 2015. Segmentointi – Mitä tarkoittaa Segmentointi? Viitattu 21.5.2015 <https://www.e-conomic.fi/kirjanpito-ohjelma/sanakirja/segmentointi>

Francis, B. 2013. 40 iOS and Android mobile monetization strategies for app developers. Viitattu 1.3.2015 <http://appflood.com/blog/40-ios-and-android-mobile-monetization-strategies-for-app-developers>

Grossberg, R. 2014. The Anatomy of a Successful Mobile Game. Viitattu 10.3.2015 [http://insights.wired.com/profiles/blogs/the-anatomy-of-a-successful-mobile-game?xg\\_source=msg\\_appr\\_blogpost#axzz3Uewol4tw](http://insights.wired.com/profiles/blogs/the-anatomy-of-a-successful-mobile-game?xg_source=msg_appr_blogpost#axzz3Uewol4tw).

Lele, S. 2014. Why is mobile user segmentation crucial for app developers? Viitattu 21.5.2015 <http://blog.personagraph.com/mobile-user-segmentation-crucial-app-developers>

Libgdx 2014. Goals and Features. Viitattu 10.3.2015 <http://libgdx.badlogicgames.com/features.html>.

Marks, T. 2013. The Essential Steps For Mobile Games Success. Viitattu 10.3.2015 <http://www.mindgrub.com/sites/default/files/downloads/The%20Essential%20Steps%20For%20Mobile%20Games%20Success.pdf>.

NPD Group 2015. Average Time Spent Playing Games on Mobile Devices Has Increased 57 Percent Since 2012. Viitattu 10.3.2015 <https://www.npd.com/wps/portal/npd/us/news/press-releases/2015/average-time-spent-playing-games-on-mobile-devices-has-increased-57-percent-since-2012>.

Oehlke, A. 2013. Learning Libgdx Game Development. Packt Publishing.

Rovio 2015. Angry Birds. Viitattu 3.3.2015 <http://www.rovio.com/en/our-work/games/view/1/angry-birds>.

Seufert, E. 2012. Mobile game user segmentation through monetization behavior. Viitattu 10.3.2015 <http://mobiledevmemo.com/mobile-game-user-segmentation-through-monetization-behavior>.

Today's Web 2014. Hot Mobile Games Industry Trends For 2015. Viitattu 10.3.2015 <http://todaysweb.net/hot-mobile-game-dev-industry-trends-2015>.

Traeg, P. 2014. Four Ways To Build A Mobile Application, Part 4: Appcelerator Titanium. Viitattu 14.3.2015 <http://www.smashingmagazine.com/2014/03/10/4-ways-build-mobile-application-part4-appcelerator-titanium>.

Unity 2014. Get Unity. Viitattu 14.3.2015 <http://unity3d.com/get-unity>.

Unity documentation 2015. Creating and Using Scripts. Viitattu 14.3.2015. <http://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>.

Viswanathan, P. 2015. Mobile App Monetization Models. Viitattu 10.3.2015 <http://mobiledevices.about.com/od/marketingapps/tp/Mobile-App-Monetization-Models.htm>.

Weiss, N. 2014. Selecting a Cross-platform Game Engine. Viitattu 14.3.2015 <http://www.binpress.com/blog/2014/05/14/selecting-cross-platform-game-engine>.