

# **Puheterapiaa tukevan mobiilipelin kehittäminen**



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeen ammattikorkeakoulu, tietojenkäsittely

Syksy, 2018

Juho Ukkonen

Tietojenkäsittely  
Visamäki

---

<b>Tekijä</b>	Juho Ukkonen	<b>Vuosi</b> 2018
<b>Työn nimi</b>	Mobiilipelin kehitys puheterapian avuksi	
<b>Työn ohjaaja/t</b>	Tommi Saksa	

---

## TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli kehittää mobiilipeliä puheterapian avuksi. Keskeisiin tavoitteisiin kuului prototyypin kehittäminen ja jatkokehityksen suunnittelu sekä pelin hyödyllisyyden arviointi puheterapian apuna.

Ensiksi taustatyönä tutkittiin lasten kehitystä ja pelejä opetuksen apuna. Taustatyötä tehtiin tutkimalla eri lähteistä miten pelien vaikutusta opetukseen. Taustatyötä tehtiin pelin suunnittelussa apuna olleen puheterapeutin kanssa, mm. käymällä läpi erilaisia puheterapian tukena käytettäviä pelejä ja leikkejä. Taustatyöksi tutkittiin myös pelin kehityksen periaatteita ja tärkeimpiä huomioitavia asioita pelin kehityksessä.

Taustatöiden jälkeen valittiin keskeinen peli, jota hyödynnetään puheterapiassa usein. Puheterapeutin kanssa päädyttiin luokittelupeliin, jossa luokitellaan erilaisia asioita kuten eläimiä, hedelmiä ja huonekaluja. Valittu peli päätettiin toteuttaa Android-mobiilialustalle. Huolellisen suunnittelun jälkeen pelin prototyyppi toteutettiin opinnäytetyön käytännönsuudessa.

Valmis prototyyppi testattiin sekä pelin toiminnallisuus puheterapian apuna arvioitiin. Positiivisten arviointi tuloksien jälkeen pelille kehitettiin jatkokehityssuunnitelma.

**Avainsanat** Mobiilipeli, Android, puheterapia, pelinkehitys

**Sivut** 25 sivua, joista liitteitä 1 sivu

Business information technology

Visamäki

---

**Author**

Juho Ukkonen

**Year** 2018

**Subject**

Development of mobile game to be use as part of speech therapy

**Supervisors**

Tommi Saksa

---

ABSTRACT

Goal of this thesis was to develop a prototype of mobile game that would be used as part of speech therapy. Essential targets were functional prototype and evaluating the functionality of the mobile game use as part of speech therapy and planning further development.

First there was done research of children development and use of games as part of teaching. Background work included searching information on how games influence learning. Research was partly done together with speech therapist, whom later helped planning the game. With the therapist we looked and tried different games that are often used as part of speech therapy. Lastly in the research part was going through key principles and most important things to keep in mind when developing a mobile game.

After research was done and backgrounds for development were known, we started to plan the game together with the speech therapist. We ended up deciding to develop a simple classification game, where user drags things such as animals, fruits and things from other simple classes to their class boxes. The mobile game was decided to be made for Android platform. After good planning the prototype was developed.

Ready prototype was tested and evaluated for use as part of speech therapy. After positive feedback we made a roadmap and plans for further development.

**Keywords** Mobile game, Android, speech therapy, game development

**Pages** 25 pages including appendices 1 page

# SISÄLLYS

1	JOHDANTO.....	1
2	PELIT OPETUKSEN APUNA .....	2
2.1	Lapsen oppiminen .....	2
2.2	Opetuspelit.....	2
3	ANDROID.....	4
3.1	Taulutietokoneet.....	4
3.2	Google Play.....	5
4	PELINKEHITYKSEN TYÖNKULKU: PRE-PRODUCTION .....	6
5	PELINKEHITYKSEN TYÖNKULKU: TUOTANTO .....	10
6	PROJEKTIN ALOITUS.....	13
6.1	Suunnittelun aloittaminen .....	14
6.2	Konsepti ja rautalankamallit .....	14
6.2.1	Latausruutu.....	14
6.2.2	Alkuvalikon suunnittelu .....	15
7	PROTOTYYPIN TOTEUTUS.....	18
7.1	Aloitus.....	18
7.1.1	Kansioittaminen ja organisointi.....	19
7.2	Päävalikko.....	20
7.3	Ensimmäinen prototyyppi.....	22
7.3.1	Latausruudun toteutus .....	22
7.3.2	Pelaamisen toteutus.....	22
7.3.3	Proton testaaminen.....	23
8	JATKOKEHITYS.....	23
	LÄHTEET .....	24

## Liitteet

Liite 1 Knights of Unity:n suosittama kansiotus



## 1 JOHDANTO

Tämän opinnäytetyön tavoitteena on kehittää mobiilisovellus Android tabletti -alustalle. Applikaatio on yksinkertainen mobiilipeli, jonka tavoitteena on helpottaa puheterapiaa etenkin pienten lasten kanssa. Applikaatiolla ei ole varsinaista tilaajaa, mutta se suunnitellaan yhteistyössä puheterapeutin kanssa, joka toimii projektin aikana niin sanottuna tilaajana.

Opinnäytetyön teoriaosuudessa käydään läpi pelin tai applikaation kehityksessä huomioitavia asioita ja kehityksen eri vaiheet. Teoriaosuuden tavoitteena on avata lukijalle kehityksen vaiheet ja vaatimukset ja antaa laaja kokonaiskuva applikaation kehityksestä, niin pienemmässä kuin isomassa useamman tiimin projektissa. Käydään myös läpi työvälineitä, ohjelmia ja työskentelytapoja.

Käytännön osuudessa toteutetaan mobiiliapplikaatio noudattaen ja käyttäen teoriaosuudessa läpikäytyjä vaiheita ja tapoja ja ohjelmia. Käytännön osuuden tavoitteena on dokumentoida mobiili applikaation kehitys niin sanotun asiakkaan ideasta melkein julkaisukelpoiseksi sovellukseksi. Vaikkei sovelluksella ole varsinaista tilaajaa, käytämme tässä osuudessa niin sanottua tilaajaa, jotta projekti olisi aidomman oloinen. Tilaaja toimii myös suunnitteluapuna antamaan kohdeyleisön näkökulma ja vaatimukset, jotta sovelluksesta saataisiin mahdollisimman toimiva.

## 2 PELIT OPETUKSEN APUNA

### 2.1 Lapsen oppiminen

Lapsen oppimisen on huomattu olevan kokonaisvaltaista. Lapsi oppii kaiken aikaa tehdessään aktiivisia asioita ja saadessaan kokemuksia sekä elämyksiä ja ollessaan vuorovaikutuksessa ympäristön kanssa. Kuin aikuisen, myös lapsen oppimista edistää, mikäli henkilö on motivoitunut oppimaan. Kasvattaja pystyy toiminnallaan vaikuttamaan lapsen motivaatioon tarjoamalla lapselle kiinnostavia ja lapsen kehitystason mukaista oppimismateriaalia sekä oppimistoimintaa. Oppimisen kannalta myös hyvä itseluottamus on tärkeää, itseluottamuksen ja hallinnantunteen avulla lapsi pystyy helpommin mieltämään haasteet oppimistilanteiksi. Tällöin lapsi suuntautuu tehtävään ja onnistuminen on todennäköisempää, kuin jos lapsen itseluottamus olisi huonompi. Mikäli lapsi ei luota itseensä hän voi jättää tehtävät tekemättä ja käyttäytyä passiivisesti. (Kajan, Karjalainen, Kauppinen 2014.)

Opetuksen sisältö ei ole keskeistä lapsen oppimisen kannalta. Tärkeämpää on, miten oppimiskokemukset liittyvät lapsen ympäristöön ja miten ne rakentavat lapsen elämää, sekä minkälaisia merkitysisältöjä lapsi antaa opetukselle. Oppimisen perusta on lapsen omassa toiminnassa, lapsi rakentaa uutta tietoa olemassa olevan tiedon ja omien kokemuksien pohjalta. Lapsen oma kokemusmaailma on pohjana lapsen oppimalle uudelle tiedolle. Vertaisryhmät, leikit sekä aktivoiva oppimisympäristö toimivat tukirakenteina oppimiselle. Näillä voidaan rakentaa lapselle uusi näkökulma oppimiseen. Vapaan leikin lisäksi lapselle ovat tärkeitä erilaiset oppimisen muodot kuten pelit ja sääntöleikit. (Kajan, Karjalainen, Kauppinen 2014.)

### 2.2 Opetuspelit

Pelit ja leikit ovat olleet oppimisen tukena aikojen alusta lähtien. Digitaalisessa muodossa niitä on nähty jo 1950-luvulla, ja jo 70-luvulla ensimmäiset digitaaliset oppimispelit tulivat kouluihin. Mutta vielä 80-luvulla sana *pele* oli negatiivinen käsite opetuksessa, ja vasta myöhemmin 90-luvulla digitaalisten pelien hyöty alettiin huomata ja niistä alkoi tulla osa yleistä opetusta. (Saarenpää 2009.)

Leikkiminen on ihmisille luonnollinen tapa oppia. Leikkiessään lapsi käyttää mielikuvitusta ja luomaan kuvitteellisia tilanteita ja näin laajentaa ymmärrystään sekä oppii sosiaalisuutta. Peleillä voidaan tarjota samankaltainen kuvitteellinen tila, jossa on mahdollista opetella asioita, joiden harjoittelu käytännössä on hankalaa. Tällaisia pelejä ovat esimerkiksi erilaiset simulaatiopelit, joita käytetään kouluttamaan

erilaisten työkoneiden ja kulkuneuvojen käyttöä tai esimerkiksi armeijassa sotatilanteiden harjoitteluun. (Saarenpää 2009.)

Pelit tukevat oppimista tehokkaasti. Oppiminen on aktiivista, mikä tarkoittaa, ettei pelkkä tiedon saaminen riitä. Oppiakseen ihmisen tulee itse prosessoida tieto. Pelit tukevat oppimista aktivoimalla oppilaan antamalla oppilaalle mahdollisuuden käyttää oppimaansa tietoa, jolloin se jää paremmin mieleen. (Saarenpää 2009.)

Opetuspelit eivät ole oma genrensä, sillä ne perustuvat aina johonkin omaan genreensä. Opetuspelit eroavat muista peleistä sisällöltään. Ensimmäiset opetuspelit olivat harjaannuttamispelejä, joissa opetettavaa asiaa toistetaan uudelleen ja uudelleen, kunnes opittu asia iskostuu oppilaan toimintaan ja ajatusmaailmaan. Tällaiset pelit on helppoa yhdistää perinteiseen opetukseen jo kertausvaiheessa. Yleinen keino motivoida oppilaita on käyttää toiminnallisia opetuspelejä. Harjaannuttamispelit ovat yleisesti yksinkertaisia, lyhyitä ja keskittyvät yhteen asiaan. Laajempia kokonaisuuksia käsiteltäessä käytetään yleensä strategia tai simulaatio genreihin liitettyjä oppimispeliejä. Tällaiset pelit ovat yleisesti kiinnostavampia kuin harjaannuttamispelit ja ne tarjoavat opetuksen lisäksi enemmän haasteita ja ajanvietettä. Toiminta ja strategia pelit ovat kuitenkin haastavampia ottaa käyttöön opetuksessa. Roolipelit taas auttavat opetettavan asian ymmärtämisessä ja niiden avulla voidaan opettaa asioita jotka ovat muutoin haastavia opettaa, esimerkiksi etiikkaa ja moraalialia. Roolipelien käyttäminen opetuksessa vaatii kuitenkin eniten työtä opettajalta. (Saarenpää 2009.)

Pedagogisten ja viihteelliseen tarkoitukseen suunniteltujen pelien suurin ero on pedagogisten pelien kytkeytyminen suoraan opetussuunnitelmaan ja sen sisältöön ja tavoitteisiin. Oppimispelit pyritään toteuttamaan pelaajan taitavuutta vastaavaksi, jotta ne hyödyttäisivät mahdollisimman paljon oppimistavoitteisiin pääsyssä sekä pelaajan motivoinnissa. Tämä on yleensä kuitenkin melko haastavaa. Pelinkehittäjät sekä kasvattajat etsivät jatkuvasti uusia tapoja hyödyntää toistensa osaamista. Pelinkehittäjien ja kasvattajien yhteistyöllä saadaan peleihin enemmän realismia sekä toimintamahdollisuuksia. (Kajan, Karjalainen, Kauppinen 2014.)



### 3 ANDROID

Android OS on Linux käyttöjärjestelmään pohjautuva ohjelmistopino, jota käytetään erilaisten mobiililaitteiden, kuten älypuhelimien ja tablettietoneiden hallintaan. Vuonna 2005 Google keskittyi valtaamaan osansa matkapuhelinmarkkinoista ja osti Androidia kehittävän Andoid Inc. yhtiön kokonaisuudessaan. Ensimmäinen Androidiin perustuva matkapuhelin myytiin vuonna 2008. Nykyään Android on Applen iOS:n rinnalla toinen suosituimmista käyttöjärjestelmistä. Android käyttöjärjestelmä perustuu avoimeen lähdekoodiin, jonka Google julkistaa Apache-lisenssillä. Tämä tarkoittaa, että laitevalmistajat ja kolmannet osapuolet voivat vapaasti muokata ja levittää omaa versiotaan Android -ohjelmistosta. Androidin parhaita puolia ovatkin sen muokattavuus ja yksilöitävyys, ja ne ovat nostaneet Androidin suosiota valmistajien, kehittäjien sekä kuluttajien keskuudessa. Androidiin on mahdollista tehdä kustomoituja käyttöliittymiä ja pienohjelmia. Käyttäjät voivat myös muokata käyttöliittymän ulkoasua ja personalisoida sen omanlaisekseen.

#### 3.1 Taulutietokoneet

Taulutietokone, lyhyemmin tabletti, on yksiosainen, kosketusnäyttöinen kannettava tietokone. Taulutietokoneissa ei yleensä ole fyysistä näppäimistöä, vaan se on usein korvattu virtuaalisella. Tabletteihin kuitenkin on saatavilla erillisiä näppäimistöjä, jotka voi liittää niihin joko fyysisen liitännän tai Bluetooth-yhteyden kautta. Taulutietokoneita on usean kokoisia, mutta tavallisesti ne ovat normaaleita kannettavia tietokoneita pienempiä ja niihin yhdistyy älypuhelimien ominaisuuksia. 2010-luvun alulla tablettien myynti kasvoi ennätysmaisesti Applen iPad -tabletin julkaisun myötä. Nykyään tabletit ovat arkipäiväisessä käytössä niin vapaa-ajalla kuin työ- ja opetuskäytössäkin.

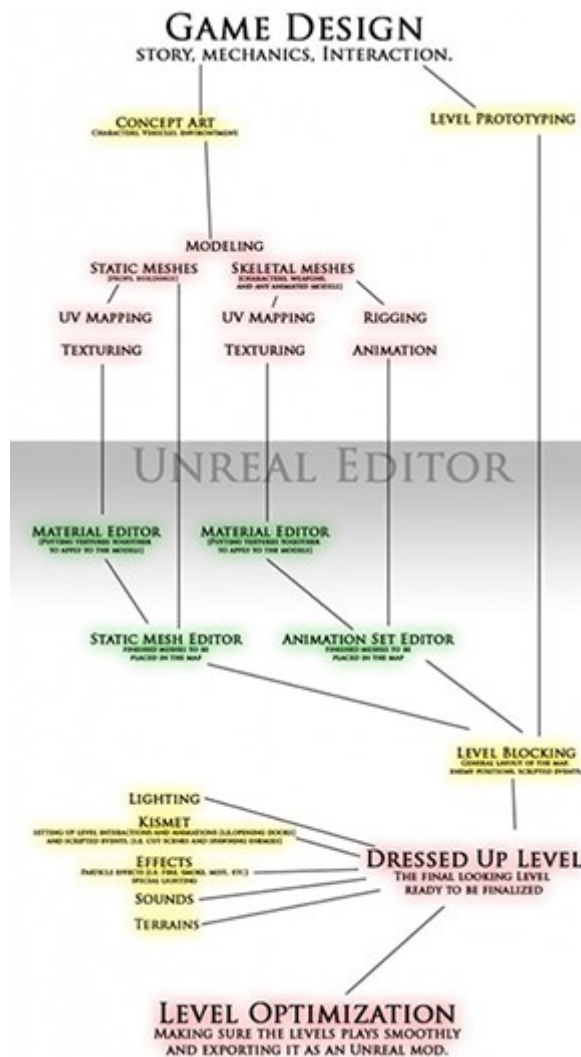
Huolimatta Androidin suosiosta älypuhelimilla, oli sen siirtyminen verrattain hidasta tableteille. Tähän hitauteen vaikutti paljolti kehittäjien ja kuluttajien välinen ”kumpi tulee ensin, muna vai kana” tilanne. Suurin osa Android appseista oli kehitetty pienille älypuhelimien näytöille, eivätkä siksi toimineet hyvin tableteilla, tämän vuoksi käyttäjät välttelivät Android tabletteja. Tämän seurauksena kehittäjät välttelivät Android tableteille appsien kehittämistä, koska käyttäjiä ei ollut riittävästi. Applen iOS tableteille oli jo kehitetty huomasti enemmän appseja, ja täten veivät Androidin markkinoita tablettien alkuvuosina.

### 3.2 Google Play

Androidin vahvimpia puolia on sen sovelluskehityksen vapaus. Kuka tahansa osaava, ohjelmointiin kykenevä voi suunnitella ja julkaista Androidille kehitettyjä sovelluksia ilmaiseksi tai halutessaan myydä niitä Androidin omassa Google Play -kaupassa. Varjopuolena kehityksen vapaudelle pidetään tosiasiaa, että kun kuka tahansa voi jakaa kehittämiään sovelluksia, syntyy hurja määrä roskasovelluksia. Roskasovellukset ovat heikkolaatuisia ja monet käyttökelvottomia. Nykyään Google on pyrkinyt tiukentamaan kriteereitään, joiden mukaan sovellukset sallitaan jaettavaksi Google Play:n. Aluksi Google Play tunnettiin nimellä Android Market, se julkaistiin vuonna 2008 ja on kehittynyt huimasti kymmenessä vuodessa. Aluksi kaupasta sai ostettua ainoastaan Android-sovelluksia, mutta nykyään sieltä voi ostaa sovelluksien lisäksi muun muassa musiikkia ja elokuvia tai kirjoja.

#### 4 PELINKEHITYKSEN TYÖNKULKU: PRE-PRODUCTION

Pelinkehitys voi olla työlästä ja aikaa vievää työtä, varsinkin jos ei ole paljoa aikaisempaa kokemusta. Ja jopa pitkällä kokemuksella isojen AAA-pelien tekeminen ei ole mahdollista yhdeltä henkilöltä vaan niihin tarvitaan useita kokeneita tiimejä suunnittelemaan ja työstämään peliä. Nopean kehityksen ansiosta pelien tekemisestä on kuitenkin tullut helpompaa koska työkalut ja palvelut ovat parantuneet ja ovat helpommin saatavilla. Nykyään myös pienillä indie-studioilla ja jopa yksittäisillä henkilöillä on mahdollisuus tehdä menestyneitä pelejä. Vaikka työkalut ovat helposti saatavilla, ja ideaa on helppo lähteä työstämään peliksi, pitäisi etenkin kokemattomien pelinkehittäjien muistaa olla haukkaamatta liian isoa palaa purtavaksi. Pienen aloittelevan tiimin on hyvä aloittaa pienestä. Kun tiimillä on yksinkertainen idea, johon he luulevat pystyvänsä laittamaan aikaa ja vaivaa voi tiimi siirtyä seuraavaan vaiheeseen pelinkehityksessä. (Kinematic Soup 2016.) Tässä kappaleessa käydään läpi pelin suunnittelua ja pelinkehityksen ideaalia työnkulkua.



Kuva 1. Level building production pipeline

Pelinkehitys vaatii useita eri taitoja, ei pelkästään teknisiä, vaan myös artistisia taitoja. Videopeleissä yhdistyy ääni ja visuaalinen taide, ohjelmointi ja pelin suunnittelu. Ensimmäistä peliä voi olla haastavaa lähteä toteuttamaan, mutta onneksi on olemassa avain vaiheita ja työtapoja joita seuraamalla voi vähentää mahdollisia ongelmia kehityksen edetessä. Näitä vaiheita ja tapoja kutsutaan työnkuluksi, jota tulisi noudattaa ja soveltaa omaan projektiin sopivaksi. (Kinematic Soup 2016.)

Ensimmäinen vaihe pelinkehityksessä ideoinnin jälkeen on konseptointi- ja esivalmisteluvaiheet, joista käytetään myös käsitteitä suunnittelu ja prototyypaaminen. Suunnittelun tarkoituksena on määrittellä pelin säännöt, pelattavuus, teema, grafiikan tyyli ja pelin kontrollit. Myös asiat kuten konseptitaide, temmon määrittely ja mekaniikat sekä kuvakäsikirjoitus ja tasojen kartoitus ovat osa suunnittelua. Suunnittelun tavoitteena on antaa tiimille selkeät tavoitteet ja auttaa ymmärtämään projektin kokonaisuutta. (Kinematic Soup 2016.)

Prototyypaaminen tarkoittaa ensimmäisen version tekemistä. Prototyypin ei ole tarkoitus olla viimeistely tai edes sisältää kaikkia toimintoja, vaan sen tarkoituksena on olla alkeellinen versio joka sisältää pelin avainmekaniikat ja säännöt, jotta nähdään ovatko ne toimivia. Prototyypaaminen on erittäin tärkeä vaihe pelinkehityksessä, sillä se säästää aikaa, jos idea ei ole sittenkään toimiva tai sitä täytyy muuttaa radikaalisti. Myös isot pelistudiot tekevät prototyyppejä ideoista testatakseen niiden toimivuuden ja varmistuakseen niiden olevan tarpeeksi hyviä toteutettaviksi. (Kinematic Soup 2016.)

Prototyypeissa käytetään hyvin yksinkertaisia malleja ja grafiikan tekstuuri sekä äänet jätetään yleensä toteuttamatta tai toteutetaan väliaikaisilla placeholder-tekstuureilla tai äänillä. Prototyypaaminen vie yllättävän paljon aikaa, ja varsinkin pienelle tiimille sen tekeminen saattaa näyttää aluksi turhalta. Mutta kun viimein prototyyppi on saatu valmiiksi ja testattua voidaan löytää useita uusia tapoja kehittää suunnitelmaa. Prototyypillä voidaan varmistaa, että ideat jotka ovat syntyneet suunnittelussa ovat oikeasti hauskoja ja niistä voidaan toteuttaa pelattava peli. On huomattavasti helpompaa muuttaa suunnitelmia alussa, kuin huomata myöhemmin pelin vaativan isoja muutoksia alkuperäiseen ideaan. (Kinematic Soup 2016.)

Prototyypaamista varten on kehitetty siihen soveltuvia työkaluja, mutta ne eivät ole välttämättömiä varsinkaan pienille studioille. Ja koska prototyypin idea ei ole olla viimeistely ja sen toteutukseen riittää yksinkertaiset muodot, kuten ympyrät ja kuutiot, jotka löytyvät jo valmiiksi melkein kaikista pelimoottoreista, voidaan prototyyppi hyvin toteuttaa kehitykseen valitulla moottorilla. Vaikkei prototyypin visuaaliset palat tai äänet ole tärkeitä säästettäväksi myöhempää käyttöä varten, osa koodista voi olla hyvin tärkeää prototyypaamisen jälkeen. Kuten minkä tahansa ohjelman toteutuksessa, on tärkeää säästää kaikki muutokset, joita kehityksen aikana tehdään itse ohjelman koodiin. *Git* on erittäin hyvä työkalu tähän tarkoitukseen. Sillä voi säilyttää ja pitää silmällä lähdekoodin muutos historiaa. Se auttaa muun muassa kokeilemaan erilaisia algoritmeja ja toteutuksia, sillä sen avulla voi aina siirtyä askeleen taaksepäin ja ladata vanhan toimivan koodin, mikäli uusi tapa ei jostain syystä toimi. Tämänkaltaista työkalua tulisi aloittaa käyttää heti kehityksen alussa, kun ensimmäiset rivit lähdekoodia kirjoitetaan. (Kinematic Soup 2016.)

Prototyypaaminen on pääasiassa suunnittelijoiden ja ohjelmoijien tehtävä, mutta jo tässä vaiheessa, kommunikointi muiden tiimien kanssa on tärkeää. Prototyypin edetessä suunnittelijat kokoavat pelin tärkeimmät toiminnot ja kehittävät niitä paremman pelattavuuden ja tasosuunnittelun saavuttamiseksi. Tässä kohtaa kehitystä pitäisi viimeistään tuoda ääni ja visuaaliset artistit mukaan projektiin. Vaikkei ääniä tai tekstuureita vielä tarvita itse prototyyppiin, kestää niiden toteuttamisessa aikaa, ja mitä aikaisemmin ääni ja grafiikka artistit pääsevät sisään pelin teemaan ja

ulkoasuun, sitä nopeammin he voivat aloittaa oman osuutensa pelin kehityksestä. Suunnittelijoiden tehtävä tässä kohtaa on auttaa artisteja ymmärtämään pelin visuaalinen vaikutelma ja millainen tunne pelaajalle halutaan luoda. (Kinematic Soup 2016.)

Ohjelmoijille suunniteltu *Git* auttaa ylläpitämään järjestystä lähdekoodin kehityksessä. Mutta myös artistit joskus tarvitsevat samantyyllisiä työkaluja pitääkseen konseptikuvat, tekstuurit ja muut pelissä käytettävät kuvat ja äänet järjestyksessä pelin kehityksessä. Näitä työkaluja kutsutaan *Digital Asset Management (DAM)* työkaluiksi. Näiden työkalujen avulla artistit voivat kokeilla useita erilaisia ilmeitä, sommitteluita ja värimaailmoja ja tallentaa ne suunnittelijoiden ja johtajien tarkasteltaviksi. Johtajat voivat näiden työkalujen avulla helposti arvioida artistien tekemiä tuotoksia ja antaa niistä palautetta, mitkä säilytetään, mitä tarvitsee kehittää lisää tai hylätä kokonaan. (Kinematic Soup 2016.)

Prototyypaamisen aikana tiimi voi valita pienen osan pelistä, joka kehitetään nopeammalla tahdilla. Tätä osaa kutsutaan vertikaaliseksi palaksi (vertical slice). Tämän osuuden kehittämisen ideana on antaa kehittäjille parempi ymmärrys kaikesta mitä tarvitsee ottaa huomioon kehityksen edetessä. Se antaa paremman kuvan teknisistä ongelmista joita tulee vastaan ja tekee deadlinejen arvioimisesta helpompaa. Lisäksi se toimii esittelykappaleena osakkaille, jotta osakkaat saisivat paremman käsityksen miltä peli tulee näyttämään valmiina. Tämän palasen tekeminen ei ole välttämätöntä, mutta se voi helpottaa, jos työnalla on iso ja monimutkainen peli. Vertikaalinen pala tulisi tehdä jostain pienestä osasta peliä joka täytyy tehdä useaan kertaan pelin kehityksen aikana. Tällainen voi olla esimerkiksi pieni taso tai alue joka toistuu pelissä moneen kertaan. Pienemmissä peleissä tämän tekeminen ei ole välttämättä kannattavaa ja saattaa kuluttaa aikaa turhaan. (Kinematic Soup 2016.)

Jossain vaiheessa pelin kehitys on pisteessä, jossa tiimi on tyytyväinen pelattavuuteen ja sen hetkiseen suunnitelmaan. Peli ei välttämättä ole lähellä valmista mutta pienet osat kuten ensimmäiset tasot ja mekaniikat ovat lähes valmiita. Tässä vaiheessa voidaan sanoa prototyypaamisen olevan tehty ja tiimi voi siirtyä seuraavaan vaiheeseen, tuotantoon. (Kinematic Soup 2016.)

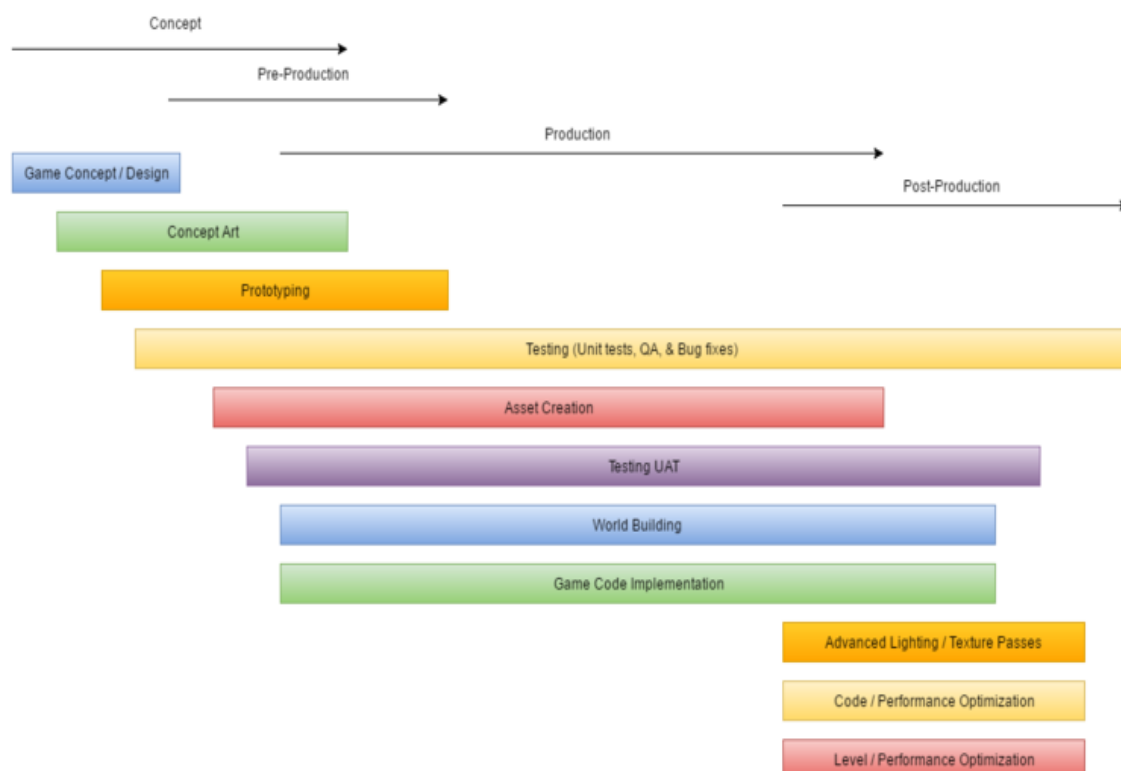
## 5 PELINKEHITYKSEN TYÖNKULKU: TUOTANTO

Tuotanto vaiheessa työskentely muuttuu vakaammaksi ja toistavaksi. Vaikka prototyypaamista ja suunnittelua tehdään edelleen pienissä määrin, suurin osa pääkohdista on jo prototyypattu ja testattu toimiviksi. Tuotanto vaihe on sisällön tuottamista tarpeeksi julkaisukelpoista peliä varten. Artistit tuottavat digitaalisia asetteja samalla kun ohjelmoijat työstävät koodia joka voidaan lisätä lopulliseen peliin, ja taso suunnittelijat muuttavat *greybox* alueet niiden lopullisiin muotoihin. (Kinematic Soup 2016.)

Tuotanto vaiheessa tuotetaan hyvin paljon digitaalisia asetteja ja ne valmistuvat käyttöä varten, voi niitä olla hankala hallita ilman oikeanlaisia työkaluja. Vaikkei digitaalisia asetteja niiden muokattavuuden takia voida säilyttää ja arkistoida kuten lähdekoodia, on digitaalisten asettien organisointiin hyviä työkaluja. Esituotannossa asettien järjestämiseen käytettiin *digital asset management (DAM)* työkaluja antamaan artisteille parempia arvioita minkä näköistä taidetta jatkaa kehittämään. Source control management (SCM) työkaluja käytetään tuotanto vaiheessa, niillä on helppo hallita asetteja ja tehdä ne helpommin saataville ohjelmoijille ja taso suunnittelijoille lisättäväksi peliin. DAM työkalut toimivat yleensä hyvin SCM työkalujen kanssa, jotta projektin johto voi edelleen tarkastella ja kommunikoida artistien kanssa helposti. (Kinematic Soup 2016.)

Kun taide asetit valmistuvat ja niillä aletaan korvata väliaikaisia versioita prototyypistä alkaa uusi erittäin aikaa vievä tehtävä, maailmanrakennus. Maailman rakennus on vaihe, jossa tasot tuodaan henkiin. Tiimi työskentelee tasojen rakennuksen parissa ja testaa jatkuvasti tasoja löytääkseen ääri tapauksia. Testauksen avulla selvitetään, miten suorituskyky saadaan pidettyä haluttuna ja että kaikki toimii kuten tarkoitettu eikä tasot aiheuta epätoivottuja sivuvaikutuksia. Maailmanrakennuksessa käytetään useita menetelmiä yhdisteltynä, jotka kaikki vievät paljon aikaa. Tässä vaiheessa on tärkeää pitää menetelmät hallinnassa, jotta tehtäviin ei kulu liikaa aikaa. Välillä tiimi saattaa löytää itsensä tilanteesta, jossa pelkällä organisoinnilla ei saada pidettyä tuottavuutta halutulla tasolla. Tämä saattaa aiheuttaa muun muassa useiden tarvittavien työkalujen ja työtapojen huonosta yhteensopivuudesta, jotka saattavat aiheuttaa pullonkaula vaikutuksen työn etenemiseen. Usein yritykset saada useampi työntekijä työskentelemään samojen asioiden, kuten tason rakentamisen kanssa saattaa aiheuttaa enemmän haittaa kuin hyötyä. Tasojen ja muiden samanlaisten asettien yhdistely saattaa aiheuttaa tietojen korruptoitumista, mikä hidastaa työn tahtia. Jotkut pelimoottorit kuten

Unity ja Unreal antavat mahdollisuuden jakaa tasot pienemmiksi osioiksi, joita voidaan työstää tehokkaasti yksilöllisesti ja minimoida tietojen yhdistämisestä aiheutuva korruptio. Vaikkei tämä ole täydellinen ratkaisu, eikä poista ongelmaa täysin ja saattaa haitata artistien luomiskykyä, nopeuttaa se hieman työn etenemistä. Jotkut kolmannen osapuolen ohjelmat kuten Scene Fusion auttavat ratkaisemaan tämän ongelman paremmin. Niiden avulla monta artistia voi rakentaa samaa tasoa samanaikaisesti, nopeamman työnkulun saamiseksi. (Kinematic Soup 2016.)



Kuva 2. Kehitys aikajana.

Tuotanto vaiheessa usein liian vähälle huomiolle jää tuotteen testaus. Testaus ei ole pelkästään valmiille applikaatiolle vaan testausta pitäisi tehdä jatkuvasti jo alkuvaiheista asti. Testaamisen liika lykkääminen voi johtaa tilanteeseen, jossa tarvitaan isoja muutoksia ja pahimmassa tapauksessa jättää valmiiseen peliin bugeja. Testaaminen ei ole pelkästään pelaustestausta, vaikka se on tärkeää ja tulee ensimmäisenä mieleen. Pelaustestausta koostuu kahdesta osasta, laadunvarmistuksesta (quality assurance QA) ja käyttökokemus testauksesta (user acceptance testing UAT). QA-testaus on pääpiirteittäin sisäinen prosessi, jossa testataan sen hetkistä versiota pelistä bugien, ongelmien ja väärin toimivien funktioiden löytämiseksi. (Kinematic Soup 2016.)

UA testausta tehdään, jotta löydettäisiin mistä pelaaja tykkää pelissä ja miten pelaaja vastaanottaa pelin. UAT koostuu sekä sisäisestä testauksesta että ulkoisesta testauksesta. Vaikka testauksesta tulee ensimmäisenä



mieleen pelaustestaus, testaukseen kuuluu myös koodi tason testejä, jotka ovat yhtä tärkeitä kuin pelaustestaus. Yksikkötestaus on näistä ensimmäinen, joka ohjelmoijan pitäisi tietää. Yksikkötestaus on erittäin hyödyllinen ja säästää paljon aikaa projektin edetessä. Sen tarkoituksena on testata lähde koodia, että se toimii kuten tarkoitettu. Lähdekoodin testaus vie paljon aikaa varsinkin manuaalisesti, joten siinä yleensä käytetään valmiiksi kirjoitettua ohjelmaa, jolla voidaan testata koko koodi ja löytää bugeja ja ei-funktioita, jotka eivät toimi kuten haluttu. (Kinematic Soup 2016.)

## 6 PROJEKTIN ALOITUS

Projektin työstäminen alkaa aina idean kehittelystä. Tässä opinnäytetyössä tehtävään peliin idea tuli ohimennen jutellen puheterapeutin kanssa. Vaikkei pelistä itsessään vielä silloin puhuttu, vaan yleisesti keskustelimme puheterapiasta, mitä se todellisuudessa on ja miten digitalisaatio vaikuttaa puheterapian kehitykseen. Puheterapiassa on pitkään käytetty pelejä avustamaan puheen kehitystä ja kommunikoinnin ymmärrystä. Yleinen käsitys on, että puheterapia on eri sanojen oikean ääntämisen opettelu ja ärrä- ja ässävikojen korjaamista, mutta todellisuudessa tämä on kuitenkin kuulemma pienin osuus itse terapiasta. Puheterapian tärkein tehtävä on oppia ymmärtämään puhetta. Tapaamani puheterapeutin, työskentelee paljon kehitysvammaisten lasten kanssa, joilla on vaikeampia ongelmia puheen kanssa kuin pelkkä ärrä- tai ässävika. He eivät esimerkiksi välttämättä ymmärrä asioiden luokittelua.

Tästä pääsemmekin pelin idean lähtökohtaan. Puheterapiassa käytetään paljon kuvakortteja, joissa on kuvia muun muassa eläimistä, huonekaluista ja erilaisista ruuista. Näillä kuvilla on paljon syvempi tarkoitus kuin pelkän kuvassa olevan sanan opettelu, kaikki nämä sanat kuuluvat johonkin helppoon ja yleiseen luokkaan. Ilman luokkia puhe ja kommunikointi olisi hyvin vaikeaa, esimerkiksi jo ruokakaupassa käyminen tuottaisi haasteita, kun kävisi ostamassa välipalaksi pari hedelmää; omenan ja banaanin. Tämän takia puheterapiassa samalla kun harjoitellaan ääntämään kuvakorttien sanoja, pelataan yleensä myös luokittelupeliä. Tuolit ja pöydät laitetaan huonekaluihin, housut ja paidat vaatteisiin ja banaanit ja omenat ruokiin. Puheterapeutti valitteli myös, miten ulkomailla käytetään paljon mobiililaitteita kuten tabletteja puheterapian apuna. Apple Storesta ja Google Playsta löytyy joitain englanninkielellä toteutettuja versioita puheterapiassa käytettävistä peleistä, mutta suomenkielisiä versioita siellä ei ole. Ulkomailla puheterapiaa avustavat mobiilipelit ovat kuulemma olleet tuloksellisia. Lapset pitävät enemmän tietokonepeleistä kuin vanhanaikaisista kuvakorteista, joten heitä on helpompaa saada mukaan pelaamaan, lisäksi on helpompaa kuljettaa tablettia kuin kymmentä eri peliä, jos asiakkaita tarvitsee käydä tapaamassa useissa eri paikoissa saman päivän aikana.

## 6.1 Suunnittelun aloittaminen

Seuraavalla tapaamisella puheterapeutin kanssa keskustelimme itse peleistä. Testailimme muutamia englanninkielisiä pelejä ja pohdimme mikä niistä tekee hyviä juuri puheterapiaan tarkoitukseen ja mitkä ovat niiden heikkouksia.

Kävimme myös läpi toteutuksessa käytettäviä laitteita ja ohjelmistoja. Pelimoottoria valittaessa kävimme läpi useita eri vaihtoehtoja, mutta koska useat jo olemassa olevat pelit oli tehty käyttäen Unity -pelimoottoria, ja koska sen käytöstä on jo kokemusta, päätimme toteuttaa pelin Unity:lla. Konseptointiin ja grafiikan toteuttamiseen valitsimme Adoben ohjelmia kuten Experience Design ja Photoshop. Toteutus alustaksi ajattelimme ensin Applen iOS -tabletti laitteita, mutta koska kehitys Applen laitteille on tehty hankalaksi ilman Mac -tietokonetta ja iOS mobiililaitetta, päätimme toteuttaa mobiilipelin Androidille.

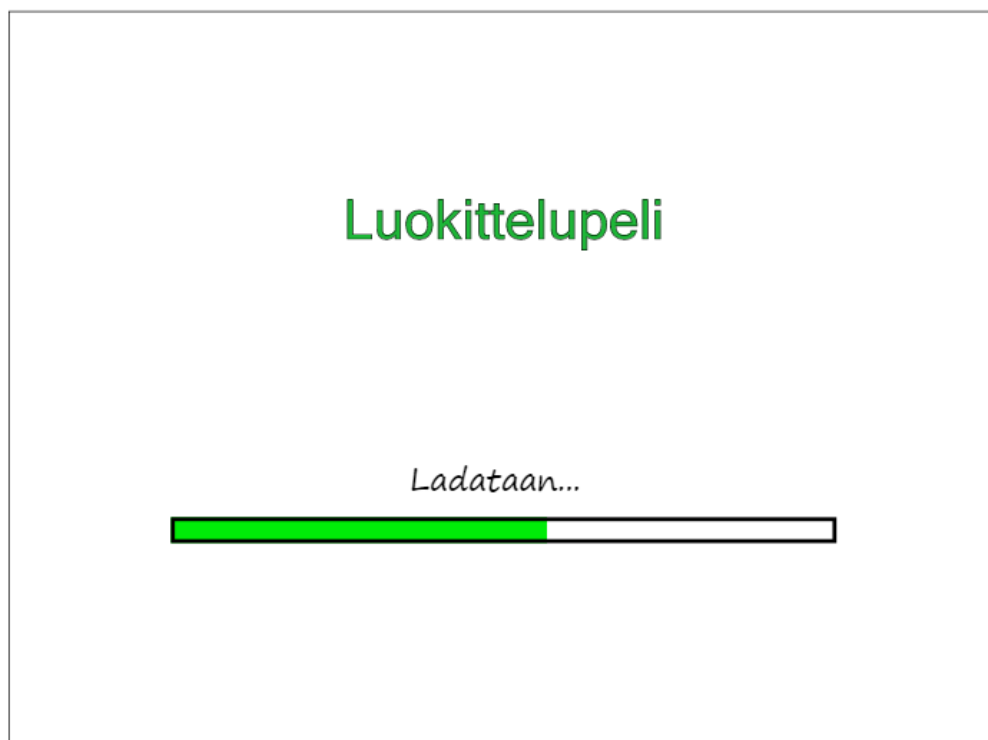
## 6.2 Konsepti ja rautalankamallit

Tämän jälkeen alkoi konseptin hahmottaminen Adoben Experience Design-ohjelmalla, joka on uusi mutta jo lyhyellä ajalla suosituksi tullut ohjelma rautalankamallien tekemisessä. Experience Design ohjelma valittiin hahmottelua varten, koska se on ilmainen ja sen käytöstä on jo jonkin verran kokemusta. Vaikka Experience Design on vasta beta vaiheessa, siitä löytyy tarpeeksi valmiita muotoja mobiili sovelluksien käyttöliittymän suunnittelua varten ja siihen on helppoa tuoda omia kuvia käytettäväksi.

### 6.2.1 Latausruutu

Jossain kohtaa jokaisen pelin kehitystä tullaan pisteeseen, jossa latausruutu on välttämätön. Yleensä latausruutu nähdään peliä käynnistettäessä tai tasosta toiseen siirryttäessä. Yleisesti käytössä olevia latausruudun animaatioita ovat latauspalkit tai pyörivät ikonit, tällaisen nähdessään käyttäjä ymmärtää, että on odotettava hetki. (Pettit 2015).

Pelin avatessa ensimmäiseksi tulee näkyviin latausruutu (kuva 1). Latausruudussa pelinlatautumista hahmottaa viiva joka kertoo pelaajalle latautumisen etenemisestä.



Kuva 3. Latausruudun rautalankamalli

### 6.2.2 Alkuvalikon suunnittelu

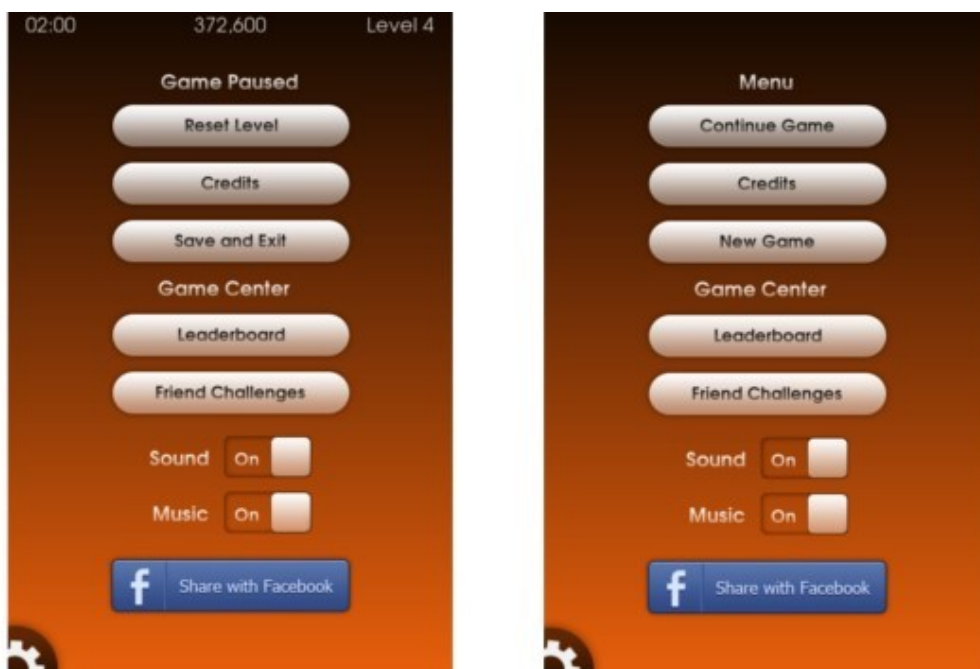
Peliä suunniteltaessa on jo alku vaiheessa hyvä ottaa huomioon alkuvalikon tärkeys. Liian usein päävalikko on toteutettu kehnosti, kiireellä ja ilman syvempää ajatusta valikkoa suunniteltaessa. Vaikka päävalikko vaikuttaa hyvin pieneltä osalta peliä, on se silti merkittävässä osassa pelin kokonaisuutta ajatellen. Kehnosti toteutettu valikko saattaa vähentää pelaajan kiinnostusta peliä kohtaan huomattavasti tai jopa saada pelaaja lopettamaan pelin pelaaminen kokonaan. (Walden 2014).

Päävalikosta olisi löydyttävä valinnat vähintään uuden pelin aloittamiseksi, asetusten säätämiseen, extroihiin ja pelin lopettamiseen. Jotta valikko olisi käyttäjälle miellyttävä, tulisi sen olla yksinkertainen ja seurata yleisesti valikoille annettuja ohjenuoria, jotka pätevät myös nettisivujen navigointi valikkoihin. Valikossa kulkemisen pitäisi viedä mahdollisimman vähän aikaa, pelaajan peliajan maksimoimiseksi. Valikossa liikkuesssa haluttuun paikkaa päästäkseen ei tulisi tarvita kuin pari painallusta. (Walden 2014).

Pelinkehittäjät yleensä haluavat näyttää mainosvideoita ja trailereita pelinkäynnistyksen jälkeen ennen alkuvalikkoa. Silloin kannattaa myös lisätä varmistus painallus ("paina jotain näppäintä jatkaaksesi") ennen alkuvalikkoon pääsemistä. Pelaajan yrittäessä ohittaa mainosvideoita, pelaaja saattaa vahingossa tehdä valintoja valikossa, mikäli mainosvideot loppuvat yllättäen ja pelaaja joutuu suoraan valikkoon. Tästä pääsemmekin seuraavaan kohtaan mikä tulisi ottaa huomioon valikkoa

suunniteltaessa. Valikon ylin oletus näppäin tulisi suunnitella niin, ettei sen painaminen vahingossa aiheuta harmia pelaajalle. Usein oletuksena valittu nappula alkuvalikossa on ”jatka peliä” minkä painaminen yleensä heittää pelaajan kysymättä viimeisimpään tallennukseensa. Tämä saattaa aiheuttaa ongelmia, jos pelaaja olisikin halunnut aloittaa uuden pelin tai säätää pelin asetuksia ennen peliin palaamista. Voisi ollakin hyvä, jos uuden pelin aloitus (uusi peli, new game) olisi ensimmäinen vaihtoehto valikossa ja vasta sen alapuolella vanhan tallennuksen jatkaminen, sillä uutta peliä aloittaessa pelaaja yleensä joutuu uuteen valikkoon. (Walden 2014).

Valikkoa suunnitellessa kannattaa miettiä mitä muita valikoita pelissä tarvitaan ja suunnitella ne samantyyllisiksi alkuvalikon kanssa. Peliä pelatessa useasti toistuvat tapahtumat jäävät nopeasti pelaajan lihasmuistiin. Valikoita suunnitellessa tämä kannattaa huomioida, sillä pelaaja hämääntyy, jos esimerkiksi alkuvalikko ja taukovalikko pitävät sisällään samoja nappeja, mutta eri järjestyksessä (Walden 2014). Tästä antaa hyvän esimerkin Walden valikoita käsittelevässä artikkelissaan. Puzzix-pelissä tämä on tehty aivan päinvastoin suosituksia.



Kuva 4. Puzzix pelin valikko

Yllä olevasta kuvasta voi helposti löytää käyttökokemuksta heikentävän nappula sijoittelun samalta näyttävistä valikoissa. Tauko valikossa kolmas nappi on tallennusta ja lopetusta varten, kun taas alkuvalikkoon jouduttaessa kolmas nappi aloittaa uuden pelin ja tuhoaa aikaisemman tallennuksen.

Tässä työssä toteutettavaan peliin suunnitellaan ja rakennetaan valikot Waldenin ohjeistuksen mukaisesti. Valikko pyritään pitämään

yksinkertaisena ja helppokäyttöisenä, ja valikossa liikkumisen tulee olla selkeää.

Alla on Waldenin ohjeistuksen mukaan tehty rautalankamalli peliin tulevasta valikosta. Tässä ei ole vielä käytetty lopullisia tekstuureita ja mm. tausta puuttuu.



Kuva 5. Alkuvalikon rautalankamalli

Alkuvalikossa pelaaja voi edetä ohjeistukseen, katsoa pelin tekijät, siirtyä pelaamaan, vaihtaa asetuksia.

Ohjeistus sisältää selkeät ohjeet pelaamiseen sekä tekstinä että animaatioina. Tekijät sivulla on listattu tekijät kuten piirtäjät ja pelin toteuttaja.

Pelaa-napin painaminen vie pelaajan valikkoon jossa pelaaja voi valita eri pelimoodista haluamansa. Eri pelimoodien vaikeudet vaihtelevat muun muassa pelissä olevien luokkien määrän mukaan.

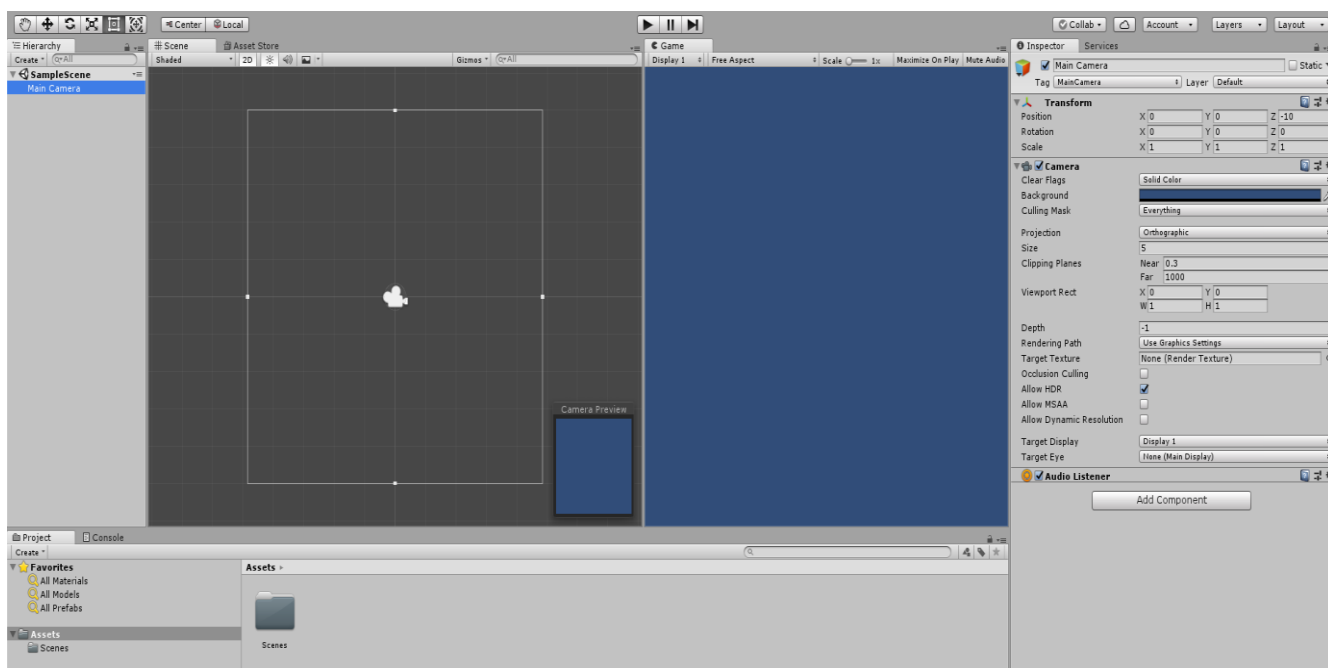
Asetuksissa voi muun muassa vaihtaa käyttöliittymän värimaailmaa. Värimaailma on tärkeässä osassa, koska kohde käyttäjinä olevilla lapsilla saattaa olla värisokeutta tai poikkeavaa värinäköä. Tämän vuoksi on tärkeää lisätä asetuksiin mahdollisuus helpottaa myös näiden käyttäjien käyttö kokemusta. Asetuksista voi myös säätää pelin äänet päälle tai pois päältä.

## 7 PROTOTYYPIN TOTEUTUS

### 7.1 Aloitus

Tässä vaiheessa, kun idean hahmottelu on pääpiirteittäin valmis ja rautalankakuvat on tehty, voidaan aloittaa itse pelin työstäminen. Ensimmäiseksi tehdään uusi Unity projekti. Kun Unity avataan ensimmäistä kertaa, se ehdottaa uuden projektin luomista.

Aloitetaan projektin luominen valitsemalla ”new project”. Tässä opinnäytetyössä toteutettava peli on kaksiulotteinen, joten luodaan 2D template. Projektissa ei käytetä erillisiä Asset paketteja, joten voidaan luoda projekti.



Kuva 6. Tyhjä Unity projekti

Projektin luomisen jälkeen Unity avautuu perus asetuksilla ja käyttäjä näkee kaikki tarvittavat työkalut heti ruudulla. Hierarchy valikossa näkyvät scenessä käytettävät objektit. Uudessa projektissa tässä valikossa ei pitäisi olla muuta kuin ”Main Camera”, joka näyttää mitä pelaaja näkee pelin käynnistyttyä. Main Cameraa painettaessa inspector ikkunaan aukeaa kameraan koodatut asetukset. Inspector ikkunasta on helppo säätää objektien asetuksia ilman että joutuu koskemaan itse koodiin. Scene ruudulla voi tarkastella muokattavana olevaa sceneä ja valita objekteja. Game ruudulla voi testata peliä. Projekti valikossa näkyy projektin assetit, jotka on hyvä pitää järjestyksessä kansioissa.

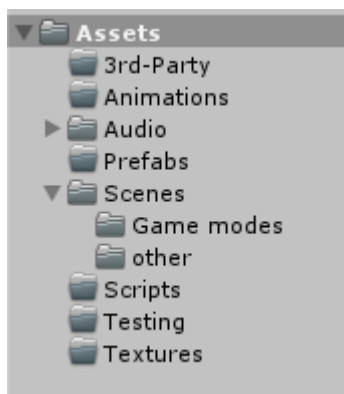
### 7.1.1 Kansioittaminen ja organisointi

Nyt kun tärkeimmät työkalut ja valikot ovat selvillä, voidaan alkaa työstää projektia. Tässä vaiheessa tehdään ensimmäinen projektin tallennus. Säännöllinen tallentaminen on tärkeää, sillä koskaan ei voi olla tietää milloin tietokone tai ohjelma kaatuu. Tällöin on hyvä olla mahdollisimman uusi versio tallessa.

Seuraavaksi tehdään projektin assettien kansioittaminen, jotta projektissa käytetyt assetit pysyvät järjestyksessä. Assettien kansioittaminen ja siistin järjestyksen ylläpitäminen on tärkeää projektissa, jossa käytetään paljon eri asetteja. Siisti ja looginen järjestys helpottaa ja nopeuttaa työskentelyä ja auttaa ulkopuolisia löytämään halutut asiat.

Pelinkehittäjät työskentelevät projektien parissa, jotka voivat koostua jopa kymmenistä tuhansista koodiriveistä ja muista aseteista. Mitä suuremmaksi toteutettava asia kasvaa, sitä hankalampaa on pitää sen rakennetta kasassa. Mikäli projektin hallinta on mennyt heti alussa pieleen, myöhemmin voi huomata käyttävänsä aikaa enemmän haluttujen asioiden löytämiseen, kuin uusien asioiden luomiseen. (The Knights of Unity 2016.)

Kansioittaminen tehdään soveltaen The Knights of Unity:n suosittelemaa kansio puuta (liite 1.). Kansiot luodaan ja nimetään tarkoituksellisesti. On hyvä olla tarpeeksi kansioita, mutta ylimääräisten kansioiden luominen vain hankaloittaa projektin hallintaa.



Kuva 7. Unity projektin kansiotus

3rd-Party kansioon lisätään kaikki kolmannen osapuolen assetit. Näitä ovat esimerkiksi Unity Storesta ostetut assetit. Näillä on hyvä olla oma kansionsa, sillä ne tulevat usein oman rakenteen kanssa, jota ei saa muuttaa.

Testing kansioon laitetaan kaikki testaustarkoituksessa luodut assetit, joiden käytöstä ei olla täysin varmoja. Tällaisia asetteja testatessa ei haluta käyttää turhaa aikaa täydelliseen organisointiin, vaan halutaan nähdä



niiden toiminta projektissa nopeasti. Mikäli testattava assetti halutaan ottaa käyttöön projektiin, voidaan se myöhemmin järjestää halutusti.

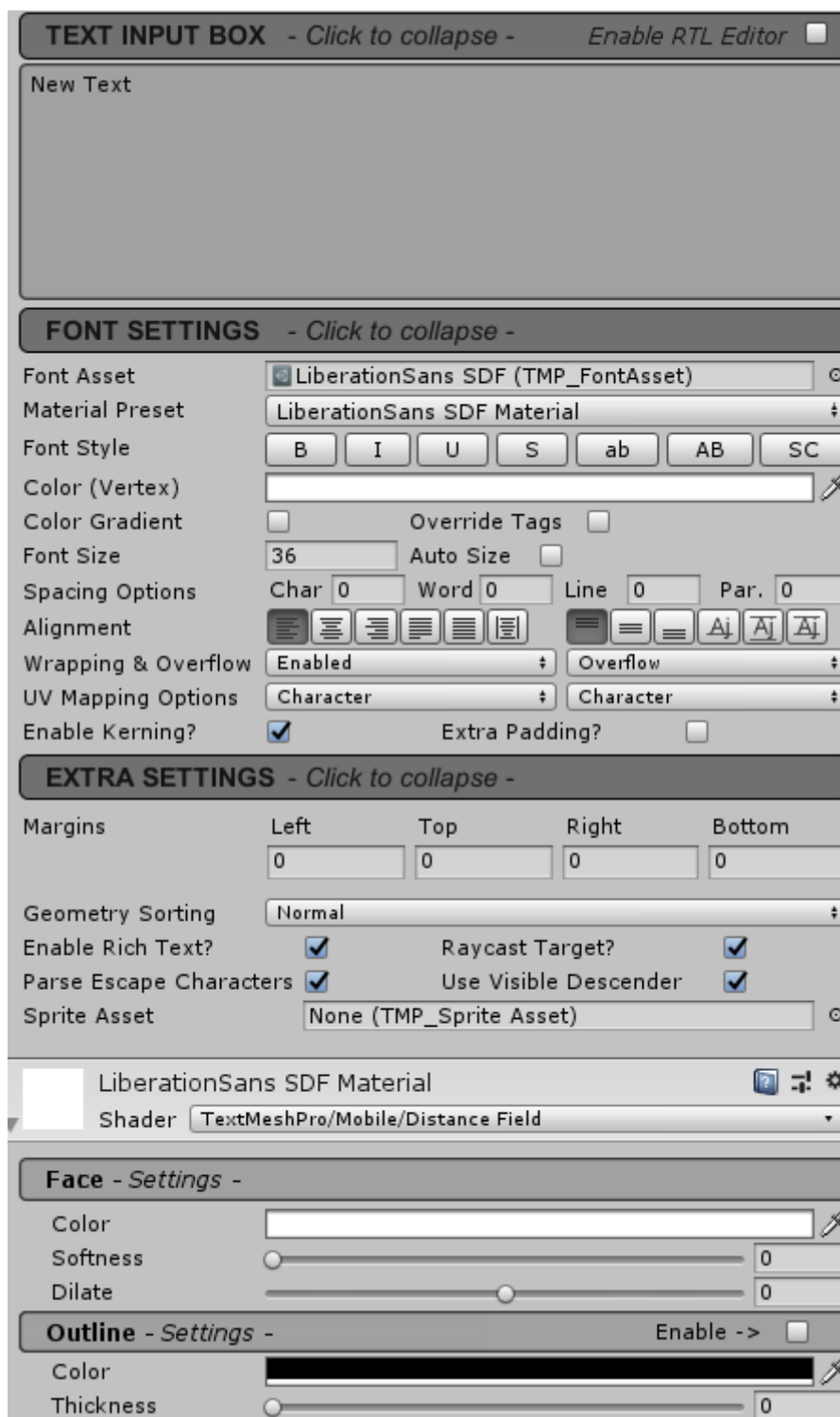
Prefabs kansioon tallennetaan prefab assetit. Prefab assetit ovat oletusobjekteja, joita käytetään useita kertoja, esimerkiksi napit voidaan luoda prefab asseteina. Prefabeja käytetään ajan säästämiseksi. On järkevää lisätä peliobjektit komponentteina ja säätää niiden arvot halutuiksi. Toistuvat peliobjektit voitaisiin tehdä myös kopioimalla olemassa olevaa objektia ja näin saataisiin duplikaatteja, mutta mikäli näitä halutaan myöhemmin muokata, jouduttaisiin jokainen objekti muokkaamaan yksitellen. Tätä varten Unityssa on prefab assetti tyyppi, joka on tarkoitettu luomaan oletusobjekteja, joita voidaan sitten lisätä peliin toistuvasti ja muokata kaikkia samaan aikaan.

## 7.2 Päävalikko

Ensimmäisenä kokonaisuutena projektiin luodaan päävalikko. Kun päävalikko on suunniteltu hyvin, on sen toteuttaminen helppoa ja nopeaa. Päävalikko toteuttamista varten tarvitaan muutama scriptti, taustakuva, tekstuurit ja yksi kolmannen osapuolen assetti tekstin hallintaan.

Päävalikon napit toteutetaan laatikkoina, jotta niiden erottaminen on helpompaa. Napeista tehdään oma prefab objekti, joka sisältää laatikon ja tekstin tekstuurit ja tyylittelyt. Näin kaikkia nappeja voidaan muokata myöhemmin yhdestä paikasta. Mallinappi koostuu kokonaisuudessaan Unityn button- ja TextMesh Pro -objekteista, sekä lyhyestä scriptistä, jolla päästään päävalikosta pelaamaan. Pelimuotoa valittaessa napit toteutettiin hieman eri tavalla, sillä koko pelimuoto objektista haluttiin tehdä yksi iso nappi. Tämä onnistuttiin tekemään lisäämällä button -komponentti itse objektiin.

Tekstit toteutetaan käyttäen Unityn Asset Storesta ladattavaa ilmaista TextMesh Pro assetia. TextMesh Pro on tehokas ja helppokäyttöinen työkalu, jolla voidaan luoda ja hallinnoida teksti objekteja huomattavasti Unityn oletus teksti työkalua paremmin.



Kuva 8. TextMesh Pro:lla luodun tekstin oletus muotoilusetukset.

Mikäli prefab objektien skriptejä ei lasketa mukaan, prototyypin päävalikko toimii yhden hyvin yksinkertaisen skriptin voimin. Ainoastaan eri skeneihin siirtymiseen tarvitaan koodia. Kaikki muu valikossa liikkuminen on pystytty toteuttamaan Unityn graafisen käyttöliittymän kautta, piilottamalla ja tuomalla esiin eri näkymiä.

### 7.3 Ensimmäinen prototyyppi

Ensimmäisessä protossa eli prototyypissä toimii ainoastaan yksi pelimuoto ja käytössä olevat tekstuurit ovat placeholdereita. Tämän tarkoituksena on testata pelin toimivuutta ja tutkia ollaanko toteutuksessa menossa oikeaan suuntaan.

#### 7.3.1 Latausruudun toteutus

Prototyyppi on hyvin pieni, ja siksi nykylaitteilla peli latautuu nopeasti. Latausruudun hahmottamiseksi on se pakotettu lyhyeksi aikaa. Tässä tapauksessa 5 sekunniksi. Latausruutuun on myös lisätty muutamia pelin latautumiseen liittyviä lauseita, kuten ”istutetaan kukkia” ja ”lämmitetään ateriaa”. Pelin latautuessa näitä lauseita näytetään satunnaisesti latauspalkin alapuolella.

#### 7.3.2 Pelaamisen toteutus

Prototyyppiä varten toteutettiin yksi pelimuoto, jossa on kaksi luokkaa. Siinä on käytetty placeholder tekstuureita, jotka muodostuvat laatikoista, joissa lukee luokiteltavien eläinten ja hedelmien nimi. Pelattavuus on kohtuullisen toimiva, eikä drag and drop scriptin tekeminen tuottanut paljoa hankaluuksia. Luokiteltavia asioita voi vetää luokkalaatikoiden päälle. Mikäli luokiteltava asia vedetään oikeaan laatikkoon, se katoaa ja oikein luokiteltujen asioiden laskuri kasvaa yhdellä, väärään laatikkoon vedettäessä luokiteltava asia pomppaa takaisin lähtökohtaansa.

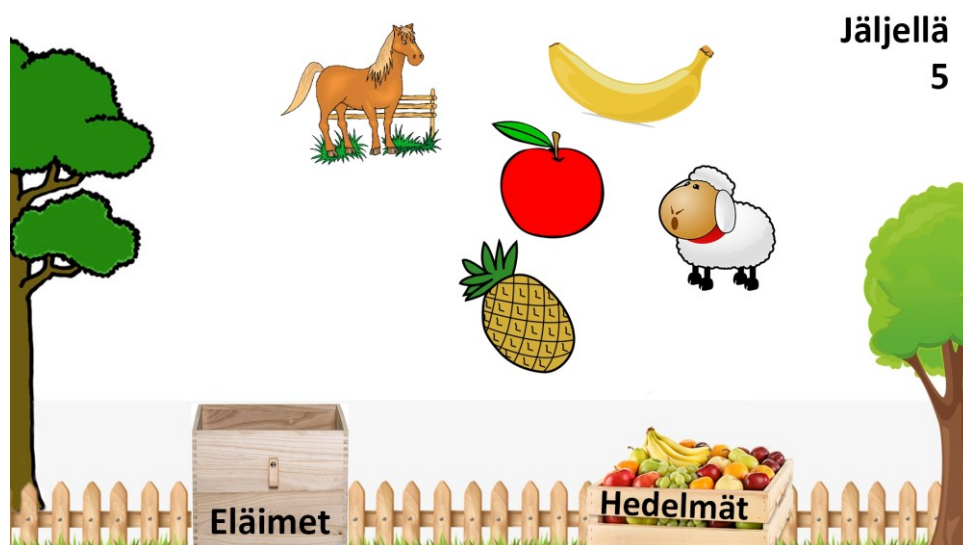
Myöhemmin pelimuotojen lisäämisen ei pitäisi tuottaa hankaluuksia, sillä pelaaminen toteutettiin jatkokehitystä ajatellen. Uusien luokkien lisääminen peliin onnistuu helposti kopioimalla olemassaolevia prefabbeja ja itse peliä pyörittävät scriptit ottavat lisätyt luokat automaattisesti huomioon.

### 7.3.3 Proton testaaminen

Prototyypin testaamiseen käytettiin Huawei P9 puhelinta ja Android versio 7.0. Unityllä testaamiseen tarvitaan uusin versio Unitystä sekä Androidista. Testaamisen valmistelu oli helppoa noudattamalla ohjeita Unityn sivuilta. Testaamista varten tarvitsi päivittää puhelimen Android versio sekä sallia USB debugging puhelimen kehitys asetuksista. Testaaminen sujui ongelmitta ja pelaaminen toimi kuten pitää.

## 8 JATKOKEHITYS

Prototyyppi on todettu toimivaksi ja sen avulla on saatu uusia kehitysideoita. Pelin jatkokehitys tästä eteenpäin tulee olemaan toistuvaa uusien pelimuotojen luontia ja uusien ominaisuuksien lisäämistä. Samalla peliin kehitetään useampien luokkien pelimuotoja sekä uusia luokiteltavia asioita. Pian peliä voidaan jo alpha testata. Jatkokehityksen isoimmiksi ongelmiksi tulevat muodostumaan tarvittavien tekstuuriin ja äänien lisääminen.



Kuva 9. Jatkokehitys demo kuva

Kehityksen edetessä peliin lisättäisiin muunmuassa kuva 9. tyylistä grafiikkaa. Tekstuurit ostettaisiin työnä kahdelta ulkopuoliselta graafikolta. Valmis peli lisättäisiin Google Play -kauppaan ilmaiseksi, jotta jokaisella olisi mahdollisuus käyttää peliä. Peliin luultavasti lisättäisiin jotain mainoksia rahoittamaan grafiikan työstäminen ja mahdollinen jatkokehitys eteenpäin.

## 9 YHTEENVETO

Opinnäytetyötä tehdessä huomattiin, ettei pelin kehittäminen ole niin yksinkertaista kuin voisi luulla. Idea voi vaikuttaa yksinkertaiselta ja helpolta ajatuksena, mutta kun sitä lähdetään työstämään huomataan ettei se olekaan niin yksinkertainen kuin aluksi luultiin. Tämä saattaa yllättää aloittelevan kehittäjän.

Käytetyt työkalut olivat päteviä ja riittäviä tässä opinnäytetyössä tehtyyn toteutukseen. Vaikka parempia mobiilikehitys alustoja olisi tarjolla, tarjoaa Unity pelimoottori kaiken tarpeellisen niin 2D- kuin 3D-sovelluksien kehitykseen mobiilialustalle. Alkuvaiheessa suunnitteluun ja myöhemmin tekstuuriin luomiseen käytettävät Adoben sovellukset olivat erittäin hyviä. Adoben Xd:llä oli helppoa ja nopeaa toteuttaa rautalankamalleja suunnitteluvaiheessa. Toteutuslupien Android puolestaan tarjosi helpon kehityksen ja testauksen kulun. Androidille voi kukaan ilmaiseksi kehittää sovelluksia, ja niiden testaaminen on helppoa jos vain omistaa Android-käyttöjärjestelmää käyttävän mobiililaitteen.

Itse tuotos oli onnistunut ja tyydytti kaikkia projektissa mukanaolleita osapuolia. Pelin prototyyppi saatiin toteutettua ja testattua, sekä saatiin alulle jatkokehitys-suunnitelma. Siitä mihin tätä opinnäytetyötä varten tehty prototyyppi jäi, on helppo jatkaa kehitystä kohti beta-vaihetta ja sen jälkeen julkaisukelpoista peliä.

## LÄHTEET

Christopher Walden 2014. A bit of design theory for mobile games. Haettu 5.6.2018 osoitteesta [https://www.gamasutra.com/blogs/ChristopherWalden/20141029/228917/A\\_bit\\_of\\_design\\_theory\\_for\\_mobile\\_games.php](https://www.gamasutra.com/blogs/ChristopherWalden/20141029/228917/A_bit_of_design_theory_for_mobile_games.php)

Kajan Katariina, Karjalainen Antti, Kauppinen Jenni. 2014. Höpsistä pussiin -peli kielen kehityksen tukena Haettu 5.3.2018 osoitteesta <https://www.theseus.fi/bitstream/handle/10024/81883/hopsista%20pussiin%20-peli%20kielen%20kehityksen%20tukena.pdf?sequence=1>

Kinematic Soup 2016. Game Development Workflow. Haettu 5.4.2018 osoitteesta <http://www.kinematicsoup.com/news/2016/10/26/game-development-workflow>

Kinematic Soup 2016. Game Development Workflow Part 2. Haettu 5.4.2018 osoitteesta <http://www.kinematicsoup.com/news/2016/11/10/game-development-workflow-part2>

Nick Pettit 2015. How to Make a Loading Screen in Unity. Haettu 17.6.2018 osoitteesta <http://blog.teamtreehouse.com/make-loading-screen-unity>

Knights of Unity 2016. 7 Ways to Keep Your Unity Project Organized. Haettu 5.6.2018 osoitteesta <http://blog.theknightsofunity.com/7-ways-keep-unity-project-organized/>

Saarenpää, H. 2009. Johdatusta oppimispelien ja pelaamalla oppimisen maailmoin. Haettu 3.3.2018 osoitteesta <http://pelitieto.net/oppimis-pelit-ja-hyotypelaaminen/>

Unity 2018. Unity Documentation. Haettu 17.6.2018 osoitteesta <https://docs.unity3d.com/Manual/Prefabs.html>

## Knights of Unity:n suosittelen kansiotus

- 3rd-Party
- Animations
- Audio
  - Music
  - SFX
- Materials
- Models
- Plugins
- Prefabs
- Resources
- Textures
- Sandbox
- Scenes
  - Levels
  - Other
- Scripts
  - Editor
- Shaders

1. **Do not store any asset files in the root directory.** Use subdirectories whenever possible.
2. **Do not create any additional directories in the root directory,** unless you really need to.
3. **Be consistent with naming.** If you decide to use camel case for directory names and low letters for assets, stick to that convention.
4. **Don't try to move context-specific assets to the general directories.** For instance, if there are materials generated from the model, don't move them to Materials directory because later you won't know where these come from.
5. Use 3rd-Party to store assets imported from the Asset Store. They usually have their own structure that shouldn't be altered.
6. Use *Sandbox* directory for any experiments you're not entirely sure about. While working on this kind of things, the last thing that you want to care about is a proper organization. Do what you want, then remove it or organize when you're certain that you want to include it in your project. When you're working on a project with other people, create your personal Sandbox subdirectory like: *Sandbox/JohnnyC*.

