

Juho Isokoski

## **NÄKYMÄTÖN OPETTAJA**

Miten kenttäsuunnittelulla voidaan opettaa pelaajaa?

## **NÄKYMÄTÖN OPETTAJA**

Miten kenttäsuunnittelulla voidaan opettaa pelaajaa?

Juho Isokoski  
Opinnäytetyö  
Syksy 2022  
Viestinnän tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Viestinnän tutkinto-ohjelma, Visuaalisen suunnittelun suuntautumisvaihtoehto

---

Tekijä(t): Juho Isokoski

Opinnäytetyön nimi: Näkymätön opettaja - Miten kenttäsuunnittelulla voidaan opettaa pelaajaa?

Työn ohjaaja(t): Tuukka Uusitalo

Työn valmistuslukukausi ja -vuosi: Syksy 2022

Sivumäärä: 31

---

Tutkielmani on katsaus videopelien kenttäsuunnitteluun ja eri keinoihin, joilla opettaa pelaajaa pelaamaan. Tutkimuskysymyksenä pohditaan, miten pelaajaa voidaan opettaa pelaamaan sanattomasti kenttäsuunnittelun avulla. Tutkielmassa keskitytään pääasiassa yksinpelattaviin tietokone- ja konsolitoimintapeleihin.

Tietoperustaosiossa perehdytään aluksi kenttäsuunnitteluun, sen historiaan, nykypäivään ja käytäntöön. Toisessa osassa tarkastellaan pelaajan opettamista tutoriaalien avulla ja perehdytään niin sanottuun orgaaniseen tutoriaaliin, jolla tarkoitetaan pelaajan opettamista kenttäsuunnittelun avulla. Lähteinä käytin pelisuunnittelun kirjallisuutta, alan ammattilaisten nettiartikkeleita, videoita ja omia kokemuksia pelaamisesta ja pelien tekemisestä.

Analyysiosiossa otin syvempään tarkasteluun 2014 julkaistun indietasoloikkapelin Shovel Knight. Analyysissä käydään läpi Shovel Knightin ensimmäistä kenttää samalla poimien merkittäviä kohtia, jotka opettavat pelaajaa. Keinoja pelaajan opettamiseen on monia, mutta tiivistetysti hyvä orgaaninen tutoriaali johdattaa pelaajan päämääräänsä iskostaen pelin toiminnallisuudet jatkuvalla toistolla ja samalla antaen pelaajalle turvallisia alueita, joissa harjoitella uusia kykyjään.

---

Asiasanat: Pelisuunnittelu, Pelituotanto, tutoriaali, kenttäsuunnittelu

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Communication, Option of Visual Communication

---

Author(s): Juho Isokoski

Title of thesis: Invisible Teacher – How to teach player through level design?

Supervisor(s): Tuukka Uusitalo

Term and year when the thesis was submitted: Fall 2022

Number of pages: 31

---

This thesis is an overview of videogame level design and the different ways to teach player how to play the game. The goal was to study how games can teach player through level design. The thesis mainly focuses on single player action games on PC and console platforms.

The first chapter explains the meaning of level design, its history, and the practice of making a level. The second chapter explores different ways of teaching player through tutorials and finally through level design. Teaching through level design is also known as organic tutorial. The sources for this thesis were gathered from game design literature, internet articles, professional video essays and from author's own experiences as a game developer.

In the third chapter I analyze the 2014 indie platformer called Shovel Knight. In the analysis I showcase how the first level of Shovel Knight teaches its player how to play the game. There are many ways a level can teach a player, but in a nutshell good organic tutorial guides player indirectly through levels, teaches mechanics through repetition and prevents frustration by giving player safe areas to practice their new skills.

---

Keywords: Video games, Game production, Game design, Level design, Tutorials

# SISÄLLYS

1	JOHDANTO .....	6
2	KÄSITTEET .....	7
3	KENTTÄSUUNNITTELU .....	9
3.1	Kenttä ja sen suunnittelu .....	9
3.2	Kenttäsuunnittelun historia .....	10
3.3	Kenttäsuunnittelu nykyään .....	11
3.4	Kenttäsuunnittelu käytännössä .....	13
4	PELAAJAN OPETTAMINEN.....	15
4.1	Tutoriaalit.....	15
4.1.1	Selostava tutoriaali.....	15
4.1.2	Kontekstuaalinen tutoriaali.....	16
4.1.3	Tutoriaalihuoneet .....	16
4.2	Tutoriaalien ongelma ja orgaaninen tutoriaali.....	17
4.2.1	Pelaajan johdattelu .....	19
4.2.2	Epäsuora opettaminen .....	19
4.2.3	Kuinka välttää pelaajan turhautuminen? .....	20
5	ANALYYSI .....	22
5.1	Shovel Knight .....	22
5.2	Kuinka Shovel Knight opettaa? .....	23
5.3	Yhteenveto .....	25
6	POHDINTA .....	27
	LÄHTEET.....	29
	KUVALÄHTEET .....	31

# 1 JOHDANTO

Videopelien kentät ovat kiinnostaneet minua jo nuoruudesta asti. Aikuisiällä olen alkanut kiinnittämään enemmän huomiota kenttäsuunnitteluun omien peliprojektieni johdosta. Teknologian kehityessä ihmisten kiinnostus videopeleihin kasvoi, ja tämän myötä peleistä on tullut saavutettavampia. Ennen pitkää laitteiston teho ja pelien jakeluformaattien tallennustilat kasvoivat. Tämä tietenkin mahdollisti audiovisuaalisesti näyttävämpiä ja mekaniikoiltaan kehittyneempiä pelejä, jotka pystyivät myös opettamaan pelaajaa pelin sisällä. Kaikki pelit eivät kuitenkaan ole onnistuneet opettamaan pelaajaa, ja niin sanotuille tutoriaaleille on muodostunut hieman negatiivinen konnotaatio.

Tässä tutkielmassa käsitellään videopelien kenttäsuunnittelua sekä perehdytään siihen, kuinka pelit voivat sanattomasti viestiä pelaajalle pelin säännöt ja mekaniikat – Toisin sanoen, miten kenttäsuunnittelulla voidaan opettaa pelaajaa pelaamaan. Tutkielmassa keskitytään pääasiassa lineaarisiin yksinpelattaviin PC- ja konsolialustojen toimintapeleihin.

Perinteisestä kenttäsuunnittelusta puhutaan nykyään ikään kuin kuolevana taiteenmuotona harrastepiireissä. Muuttuvat pelialan trendit ja suunnitteluroolien jatkuva erikoistuminen ovat siirtäneet pelinkehittäjien prioriteetit pelikokemuksiin, joissa ei ole perinteistä kenttäsuunnittelua. Voi olla, että tulevaisuudessa tätä tutkielmaa voidaan pitää ikään kuin muistopuheena. Joka tapauksessa toivon tämän tutkielman herättävän lukijassaan lisää arvostusta kenttäsuunnittelua kohtaan.

## 2 KÄSITTEET

### **AAA-pelit**

Tyypillisesti tarkoitetaan isojen ja tunnettujen julkaisijoiden tuottamia pelejä. Näiden pelien budjetit ovat nykyään miljoonien luokkaa ja usein termillä tarkoitetaan ison budjetin peliä. (Karthikeyan 2021.)

### **Art Asset**

Assetiksi kutsutaan kaikkea peliin sisälle meneviä osia. Art asset tässä tapauksessa tarkoittaa kuvallisia osia. (Unity 2022)

### **Capture the Flag**

Suoraan suomennettuna Capture the Flag tarkoittaa 'lipunryöstöä'. Tyypillinen pelimuoto räiskintäpeleissä, jossa tarkoituksena on varastaa vihollisjoukkueen lippu ja samalla suojella omaa lippua.

### **Combat Designer**

Suunnittelija, joka vastaa pelin taisteluista. Combat Designer pitää huolta, että pelin taistelu pysyy pelin vision mukaisena (Dreo 2021).

### **Encounter Designer**

Suunnittelija, joka suunnittelee pelaajan kohtaamisia NPC-hahmojen kanssa ja pelaajan vuorovaikutuksia ympäristön ja esineiden kanssa (CD PROJECT RED 2022).

### **Ensimmäisen persoonan räiskintä**

Englanniksi First Person Shooter, lyhennettynä FPS. Ammuntapelien tyyliä, jossa pelin näkymä on pelihahmon perspektiivistä (Karthikeyan 2022).

### **Environment Artist**

Environment artistit antavat kentän raakaversiolle visuaalisen ilmeen (Unreal Engine 2021).

### **Games-as-a-service**

Ansaintamalli, jossa peli jatkaa ansaitsemistaan julkaisun jälkeenkin. Esimerkiksi pelin sisäisillä ostoilla tai jatkuvalla kuukausimaksulla (Zaiets 2020).

### **Indie-pelit**

Indie on peräisin englannin kielen sanasta Independent. Itsenäisesti, ilman julkaisijaa tuotettuja ja julkaistuja pelejä. Usein myös tarkoitetaan pienellä budjetilla tuotettuja pelejä (Bramble 2022).

### **Mission Designer**

Suunnittelija, joka kirjoittaa ja luo tehtäviä kenttiin (England 2014).

**Modi**

Modit ovat käyttäjien itsenäisesti tuotettua sisältöä peleihin. Modit ovat tyypillisesti epävirallisia, eikä niihin lasketa pelin sisäisiä luomuksia (De 2021).

**NPC**

Lyhennetty englannin kielen sanoista Non-playable character eli pelin hahmo, jota pelaaja ei voi ohjata (Unity 2022).

**Procedural Generation**

Sisällöntuottamista puolisatunnaisesti algoritmien avulla. Joissain peleissä kenttiä ei ole varsinaisesti rakennettu, vaan generoitu (Extra Credits 2015).

**Prop**

Esineitä, joiden kanssa pelaaja voi olla vuorovaikutuksessa (Unity 2022).



## 3 KENTTÄSUUNNITTELU

Tämä luku käsittelee kenttäsuunnittelua eli Level Designia. Luvussa avataan mitä kenttä tarkoittaa ja perehdymme hieman kenttäsuunnittelija -ammattinimikkeen historiaan. Lopuksi katsotaan vielä tiivistetysti, mitä kenttäsuunnittelija käytännössä tekee, eli myös hieman kentän luomisprosessista.

### 3.1 Kenttä ja sen suunnittelu

Pelink kehitys on monivaiheinen työ ja vaatii monenlaista osaamista. Pelisuunnittelijaideoi pelin konseptin ja luo pelille säännöt, koodarit kirjoittavat pelin toiminnallisuuden, artistit ja graafiset suunnittelijat luovat pelille visualisen ilmeen ja pelin *art assetit*. Äänisuunnittelijat ja säveltäjät luovat pelin äänimaailman ja musiikit ja niin edelleen. (Rogers 2010, 12–15.) Rooleja voi olla lukuisia ja pienemmissä tiimeissä yksi jäsen voi joutua täyttämään useamman roolin. Luonnollisesti yksin työskentelevä joutuu huolehtimaan kaikista osa-alueista. Kuitenkin pelin kehityksessä tulee ennen pitkää vaihe, jossa kaikki nämä osat kohtaavat. Tätä vaihetta kutsutaan kenttäsuunnitteluksi.

Lähes kaikissa videopeleissä on kenttiä. Pelistä riippuen kenttiä saatetaan kutsua kartoiksi, tasoiksi tai tehtäviksi. Jo pitkään peleissä on esiintynyt laaja skaala eri muotoisia ja laajuisia kenttiä. Olkoon se *Capture the Flag* kartta räiskintäpelissä, kilparata rallipelissä tai yksinkertaisesti Pac-Manin labyrinti (Byrne 2005, 7).

Nimestä ja muodosta huolimatta kenttä on kaikessa yksinkertaisuudessaan tila, jossa peli tapahtuu. Tämä tila asettaa myös rajat, joiden puitteissa pelaaja on vuorovaikutuksessa pelin kanssa. Kenttäsuunnittelulla tarkoitetaan näiden tilojen suunnittelua ja rakentamista. (The Level Design Book 2022a.)

### 3.2 Kenttäsuunnittelun historia

Kenttäsuunnittelija-ammattinimike on oikeastaan aika uusi. Ammattinimikkeen katsotaan syntyneen 1990-luvulla ensimmäisten kolmiulotteisten räiskintäpelien ilmestyessä markkinoille ja mullistaen koko pelialan. ("Hourences" 2008, 15.) Tämä ei tarkoita, etteikö kenttäsuunnittelua olisi ollut peleissä ennen 1990-lukua, vaan siitä vastasi joku toinen rooli, esimerkiksi ohjelmoija tai pelisuunnittelija.

Kenttäsuunnittelun katsotaan alkaneen ruohonjuuritason toimintana modikulttuurin syntyessä. Ensimmäisen persoonan räiskintäpelien esi-isä Id Softwaren Wolfenstein 3D sai fanitekoisen epävirallisen kenttäeditorin, jolla pelaajat olivat arviolta tehneet vuoteen 1993 mennessä satoja karttoja. Tämän innoittamana Id Softwaren John Carmack pyrki luomaan vuoden päästä julkaistavasta legendasta Doomista yhä avoimempaa alustaa modaajille ja kenttäsuunnittelijoille. Vielä tänäkin päivänä pelinkehittäjät saattavat käyttää Doomia alustana oman pelin rakentamiseen. (The Level Design Book 2022b.)

Kenttäsuunnittelusta kasvoi nopeasti suosittu harrastus peliharrastajien keskuudessa. Suosio jatkoi kasvuaan 1990-luvun puolivälissä, kun Id Softwaren Quake ja 3D Realmsin Duke Nukem 3D julkaistiin kenttäeditorin kera. Samoihin aikoihin internetyhteyksistä tuli saavutettavampia ja näin harrastajien luomukset pääsivät helpommin pelaajien koettavaksi. Jotkut harrastekenttäsuunnittelijat saattoivat ansaita kohtuullisen seuraajakunnan pelaajien keskuudessa. Pelistudiot olivat tietoisia harrastelijapiireistä ja saattoivat palkata taitavimpia karttojen tekijöitä kenttäsuunnittelijoiksi. Tämä tuli tarpeeseen, sillä kolmannen ulottuvuuden saapuminen, sekä pelien kokojen kasvaminen lisäsivät pelisuunnittelijoiden jo ennalta suurta taakkaa. Oli tarvetta erikoistuneemmalle roolille. ("Hourences" 2008, 15.)

Kenttäsuunnittelijan rooli olikin merkittävä. Tällä niin sanotulla kenttäsuunnittelun kultaisella aikakaudella ammattilaiskenttäsuunnittelija saattoi vastata lähes kaikista kentän osa-alueista. Kenttäsuunnittelija oli vastuussa muun muassa visuaalisesta ilmeestä, pelattavuudesta, käsikirjoituksesta ja kohtausten ohjaamisesta. Isoissa kehitystiimeissä kenttäsuunnittelija saattoi muun muassa luoda konsepteja, pohjapiirroksia, rakentaa 3D-raakaversioita kentästä, tarkkailla pelitestaajia sekä

sijoittaa kentälle *proppeja* ja *NPC*-hahmoja. Kenttäsuunnittelija oli kuin kentän omistaja, "auteur", ja hänellä oli viimeinen sana siihen, mitä sisältöä kenttään pääsee. (The Level Design Book 2022b.)

Teknologia jatkoi kehittymistä huimaa vauhtia. Laitteistojen suorituskyky kehittyi ja peleiltä odotettiin yhä näyttävämpiä ja suurempia kenttiä. Ennen pitkää myös kenttäsuunnittelijan taakka alkoi käydä liian suureksi ja tämän vuoksi kenttäsuunnittelijan lisäksi tarvittiin erikoistuneempia rooleja. Toki 1990-luvun lopulla kenttäsuunnittelijan lisäksi studiossa saattoi olla myös *environment artist*, mutta 2000-luvun alussa useampi prosessi sai oman erikoistuneen ammattinimikkeen. Ammattikenttäsuunnittelija ei välttämättä enää rakentanut raakaversioita kentästä, luonut *proppeja* ja suunnitellut *NPC*-kohtaamisia. (The Level Design Book 2022b.) Pelin laajuudesta riippuen pelistudioissa saattoi nyt esimerkiksi olla kenttäarkkitehti, joka suunnittelee kentän pohjapiirustukset ja erikseen kentänrakentaja. Taisteluja sisältävän pelin vihollisista vastasivat nyt combat designer ja encounter designer, kun taas pelaajan tavoitteista vastasi nyt mission designer.

### 3.3 Kenttäsuunnittelu nykyään

Muuttuneet pelitrendit, alan jatkuva erikoistuminen ja kenttäsuunnittelutyökalujen hidas katoaminen ovat heikentäneet perinteisen kenttäsuunnittelun merkitystä nykypäivänä. Yhdeksi kuuluisaksi trendiksi nousivat pelit, jotka nojaavat kenttissään satunnaisuuteen eli käyttävät niin kutsuttua procedural generation -tapaa luoda kenttiä. Tämän trendin popularisoi muun muassa Minecraft (2011, Mojang), jossa kenttää syntyy satunnaisesti lisää pelaajan edetessä maailmassa. Yhä useammat AAA-pelit puolestaan siirtyvät games-as-a-service-ansaintamalliin isoine maailmoineen ja kausittain muuttuvin sisältöineen staattisten tarkoin suunniteltujen kenttien sijasta. Esimerkkejä näistä ovat muun muassa Genshin Impact (2020, miHoYo) ja Grand Theft Auto V (2013, Rockstar Games). Tällaisten maailmojen luominen on paljon vaativampaa kuin yksittäisten kenttien ja vaatiikin hieman erilaista osaamista kuin perinteinen kenttäsuunnittelu. (The Level Design Book 2022b.)

Vaikkakin perinteisten yksinpelattavien kenttien kenttäsuunnittelun merkitys on vähentynyt AAA-rintamalla, viime vuosina on tullut ison profiilin pelejä perinteisemmällä kenttäsuunnittelulla. Vuoden 2016 Doom (Bethesda Softworks) ja tämän jatko-osa Doom Eternal (2020, Bethesda Softworks) sekä Dark Souls (2011–2016, FromSoftware) -pelisarja näyttivät, että perinteisellä kenttäsuunnittelulla on vielä paikka AAA-peleissä. Retrohenkiset indie-räiskintäpelit tai jos hieman ivallinen kutsumanimi sallitaan ”boomer shooterit”, ovat kokeneet kuin uuden renessanssin viime vuosina. Pelit kuten Dusk (2018, New Blood Interactive) ja HROT (2021, Spytihnêv) omaksuvat 1990-luvun estetiikan, nopeatempoisen pelattavuuden ja perinteisen 3D-kenttäsuunnittelun.

### 3.4 Kenttäsuunnittelu käytännössä

Kenttäsuunnittelu on yhdistävä tekijä, joka tuo yhteen pelisuunnittelun, tekniikan, kuvan ja äänen. Kenttäsuunnittelija pyrkii muovaamaan näiden avulla pelaajan käytöstä ja tuntemuksia. Se mitä kenttäsuunnittelija käytännössä tekee, riippuu pelin tarpeista ja tiimin koosta. The Level Design Book -sivuston mukaan suurin osa 3D-kenttäsuunnitteluprosesseista käyvät läpi nämä seitsemän vaihetta:

1. Pre-production
2. Combat (optional)
3. Layout
4. Blockout
5. Scripting
6. Environment Art
7. Release

(The Level Design Book 2022c.)

Ensimmäisessä vaiheessa eli esituotannossa luodaan ideoita ja selvitetään projektin laajuus, tavoitteet ja rajoitteet. Esimerkiksi mitä mekaniikkoja pelaajalla on käytössä? Mitkä ovat pelaajan tavoitteet? Mikä on pelin toivottu tahditus? Toisessa vaiheessa, eli combat-vaiheessa käsitellään sitä, miten pelaajat ja/tai viholliset ovat vuorovaikutuksessa toisiinsa ja mitä se vaatii kenttäsuunnittelulta. Layoutilla tarkoitetaan yksinkertaisesti pohjapiirrosta. Tähän ei ole yhtä oikeaa tapaa, vaan se riippuu pelin luonteesta. Blockout tarkoittaa pelattavaa, raakaa prototyypikenttää, joka on rakennettu yksinkertaisista muodoista (kuva 1). Tämän avulla voidaan testata esimerkiksi, osaako pelaaja liikkua kentässä, ja tarpeen tullen kenttää voidaan muuttaa ilman suurempaa vaivaa. Scripting-vaiheessa varsinainen peli alkaa muodostua kenttään. Tässä vaiheessa kenttään ohjelmoidaan logiikka, tapahtumat ja käyttäytymiset, esimerkiksi kentän alku ja loppu, poimittavia esineitä ja vihollisen käyttäytymismallit. Environment art -vaiheessa kenttä alkaa viimein ottamaan lopullista muotoaan. Prototyypikentän muodot korvataan ympäristö artistien rakentamallaan elementeillä, kuten erilaisilla mallinuksilla, tekstuureilla ja materiaaleilla (kuva 1). Viimeisenä vaiheena on julkaisu, olkoon se sitten koko pelin julkaisu tai yksittäisen kentän julkaisu harrastepiireissä.



*KUVA 1. Yläpuolella on kentän blockout ja alla valmis kenttä. Kuva Call of Duty: Modern Warfare (2019, Activision) kentänluomisprosessista (Baker 2019.)*

Pelin tarpeitten mukaan vaihteita saattaa olla vähemmän tai ne ilmenevät eri muodoissa. Esimerkiksi projektissa, jossa ei ole fyysisiä konflikteja vihollisia vastaan, ei tarvita Combat vaihetta. 2D-peliprojektissa prosessi on tietenkin erilainen, mutta samankaltaisia vaihteita voi myös löytyä. Esimerkiksi raakakenttien rakentaminen on yleistä projektissa kuin projektissa. Nykyään kenttäsuunnittelu on erikoistunut useammaksi eri ammattinimikkeeksi. Isoissa AAA-pelistudioissa on myös mahdollista, että jokaisesta vaiheesta vastaa oma erikoistunut ammattinimike. Esimerkiksi kentän combat-vaiheesta vastaa combat designer.

## 4 PELAAJAN OPETTAMINEN

Kaikille peleille yhteistä on se, että niissä on säännöt. Peleissä ei voi tehdä ihan mitä haluaa, sillä säännöt rajaavat sitä, miten peli toimii. Esimerkiksi, milloin pelissä voi voittaa, milloin pelissä häviää ja millä tavoin pelaaja on vuorovaikutuksessa pelin tai mahdollisesti muiden pelaajien kanssa. Jos pelaaja ei ymmärrä näitä sääntöjä, voi pelin pelaaminen olla jopa mahdotonta. (Stout 2015.)

### 4.1 Tutoriaalit

1980-luvulla pelien ohjekirjat opettivat perusteellisesti, miten peliä pelataan. Ajan myötä pelien jakeluformaattit alkoivat kasvaa kokoa ja ennen pitkää tämä mahdollisti pelin sisäiset ohjeistukset. Niin kuin pelejäkin, näitä niin kutsuttuja tutoriaaleja on valtavasti. Kaikkien näiden erittely olisi erittäin aikaa vievää, eikä tutoriaaleista ole yhteisesti hyväksyttyä jaottelua. Tässä osiossa käydään läpi esimerkkejä yleisimmistä tutoriaalityypeistä.

#### 4.1.1 Selostava tutoriaali

Selostavaa tutoriaalia voisi verrata digitaaliseen ohjekirjaan. Tällaiset tutoriaalit opastavat pelaajalle pelin toiminnot, säännöt tai tavoitteet pelin ulkopuolella (Suddaby 2012). Tämä tarkoittaa, että nämä tutoriaalit löytyvät esimerkiksi pelin lataus- ja paussiruuduilta tai valikoista. Varhaisimpia selostavia tutoriaaleja ovat ruudut, jotka näyttävät pelin näppäinkomennot. Kuvassa 2 on esimerkki selostavasta tutoriaalista pelissä Heretic (1994, Raven Software/Id Software). Selostaviin tutoriaaleihin saatetaan luokitella myös pelin sisäiset encyclopediat eli tietosanakirjat.



KUVA 2. Pelin kontrollit listattuna näytöllä (Heretic 1994).

#### 4.1.2 Kontekstuaalinen tutoriaali

Kontekstuaaliset tutoriaalit ovat pelin aikana tapahtuvia ohjeistuksia. Ideaalisti kontekstuaalinen tutoriaali näyttää pelaajalle, mitä tehdä tilanteen tullen. Nämä voivat olla esimerkiksi ohjeistuksia pelikuvan päällä, pop-up-ikkunoita, kamera-ajoja tai hahmojen dialogia. Ensimmäisen God of War -pelin (2005, SCEA) alussa, ruudun alaosaan ilmestyy ohjeet, mistä painikkeista hahmolla hyökätään. Klonoa Phantasy Reverie Series (Namco, 2022) -pelin alussa uusien mekaniikkojen esiintyessä peli pysähtyy ja ruudulle ponnahtaa pop-up-ikkuna viestimään esimerkiksi, mistä painikkeesta hypätään.

#### 4.1.3 Tutoriaalihuoneet

Tutoriaalihuoneet ovat erillisiä kenttiä, jotka opettavat pelaajalle päämekaniikat. Tutoriaalihuoneet ovat usein lyhyitä ja opettavat pelaajalle pelin mekaniikat vaiheittain. Huoneet saattavat myös olla kuin esteratoja, jossa jokainen este opettaa yhden hahmon toiminnoista, kuten pelissä Cuphead (2017, StudioMDHR Entertainment). Jotkut tutoriaalihuoneet antavat pelaajalle vaarattoman tilan, jossa pelaaja voi vapaasti testata liikekomentojaan. Tällaisen voi löytää esimerkiksi tappelupelistä tai peleistä, joissa pelaajalla on suuri määrä hyökkäyksiä. Esimerkiksi pelistä Bayonetta (2009, Sega) kuvassa 3.





KUVA 3. Kentän latautuessa pelaaja voi vapaasti kokeilla eri hyökkäysyhdistelmiä (Bayonetta 2009).

## 4.2 Tutoriaalien ongelma ja orgaaninen tutoriaali

Tutoriaaleille on vuosien saatossa muodostunut melkein negatiivinen konnotaatio. Hakemalla sanan "Tutorial" kliseiden tietokannasta Tvtropes -sivustolta voi löytää yli 10 erilaista osumaa, jotka ovat negatiivisävytteisiä. Tutoriaaleja saatetaan pitää ajan haaskauksena, pitkästyttävänä, alentavina ja pahimmillaan ne eivät opeta pelaamaan tarpeeksi hyvin. (Stout 2015.) Jotkut pelit jopa parodioivat tutoriaaleja, kuten kuvassa 4, Far Cry 3: Blood Dragon (2013, Ubisoft) pilkkaa pop-up-tutoriaaleja. Tutoriaaleissa itsessään ei välttämättä ole mitään vikaa, mutta peliin sopimaton tutoriaali voi pilata pelin ensivaikutelman. Toimintapeliin sopimaton selostava tutoriaali voi olla hyvä tutoriaali esimerkiksi monimutkaisemmassa strategiapelissä.

Mikään ei tietenkään estä kehittäjiä tekemästä peliä kokonaan ilman tutoriaalia, mutta tässä on riskinä pelaajan turhautuminen. Pelin ensivaikutelma on erittäin tärkeä etenkin nykyään, kun pelejä on saatavilla niin paljon ja kynnys vaihtaa toiseen mielekkäämpään peliin on pienempi kuin koskaan. (Suddaby 2012.)



*KUVA 4. Far Cry 3: Blood Dragon parodioi tutoriaaleja, jotka pysäyttävät pelin neuvoakseen pelaajalleen jotain ilmeistä (Far Cry 3: Blood Dragon 2013).*

On kuitenkin mahdollista opettaa pelaajaa ilman suoraa opastusta. Orgaaniset tutoriaalit ovat tutoriaaleja, jotka välttävät pelaajan opettamista tekstin avulla, vaan sen sijaan opettavat pelaajaa tarkoin harkitulla kenttäsuunnittelulla. Orgaanisesti opettavan pelin ympäristö on hallittu ja rajattu siten, että pelaaja oppii pelin säännöt, mekaniikat ja tavoitteet luonnostaan peliä pelaamalla. Parhaimmassa tapauksessa pelaaja ei välttämättä edes tiedosta, että näin tapahtuu (Mora ym. 2017, 4–5). Tällaista opetusta saattoi löytyä jo 1980-luvun peleissäkin. Termi ei tosin silloin ollut vakiintunut, eikä periaatteessa vieläkään.

Orgaanisten tutoriaalien -analyysin mukaan on mahdollista havaita kolme tavoitetta, joihin hyvin suunniteltu orgaaninen tutoriaali pyrkii. Orgaaninen tutoriaali Johdatellee pelaajaa subliminaalisesti, opettaa pelaajaa kertomatta suoraan ja lopuksi pyrkii lieventämään pelaajan turhautumista (Mora ym. 2017, 11).

#### 4.2.1 Pelaajan johdattelu

Pelaajan subliminaalinen johdatus voidaan jakaa kahteen tapaan: suora- ja epäsuora johdatus. Suoraan johdatteleva kenttäsuunnittelu tarjoaa pelaajalle vain yhden suunnan, eikä eksymiseen ole mahdollisuuksia, esimerkiksi käytävät ja huoneet, jossa on vain yksi uloskäynti. Epäsuora johdattelu tunnetaan myös nimellä wayfinding. Wayfinding on hienovaraisempaa ja sillä pyritään sijoittamaan kenttiin erilaisia suunnistusapuvälineitä. Näitä hyödyntäen pelaaja löytää, missä on ja mihin mennä. Näitä voi olla muun muassa maamerkit, valon ja varjojen erot, värien kontrastit, ympäristön muotokieli tai äänilähteet. Epäsuora johdattelu on erittäin tärkeää 3D-peleissä, varsinkin jos pelaaja voi itse ohjata näkymää. Esimerkiksi pelissä Journey (2012, thatgamecompany) pelaajan ainoa vihje minne edetä, ovat mäen huipulla liehuvat liput, kuten nähdään kuvassa 5.



KUVA 5. Mäen huipulla liehuvat liput ohjaavat pelaajan oikeaan suuntaan (Journey 2012).

#### 4.2.2 Epäsuora opettaminen

Pelaaja kohtaa peleissä uusia mekaniikkoja ja haasteita. Nämä voidaan opettaa sanattomasti pelaajalle kahdella eri keinolla. Ensimmäinen niistä on kontekstuaalinen harjoitus. Kontekstuaalisella harjoituksella tarkoitetaan mekaniikkojen iskostamista pelaajalle, eikä tätä tule sekoittaa kontekstuaaliseen tutoriaaliin. Pelaajan oppiessa uuden mekaniikan tulisi mekaniikan käyttö välittömästi iskostaa pelaajan mieleen. Esimerkiksi jos kohtausta opettaa pelaajan hyppäämään, on pelaajalle

välittömästi annettava lisää tilanteita, joissa tulee hyödyntää hyppäämistä. Toinen keinoista on nimeltään antepiece. Antepiece on pelaajan kohtaama yksinkertainen haaste, joka ei rankaise pelaajaa, vaikka tässä epäonnistuisi. Antepiecen tarkoituksena on valmistella pelaaja vaativampia haasteita eli setpieceja varten. Setpiece puolestaan voi olla samankaltainen haaste kuin antepiece, mutta tällä kertaa epäonnistumisesta rangaistaan. (Mora ym. 2017, 12.)

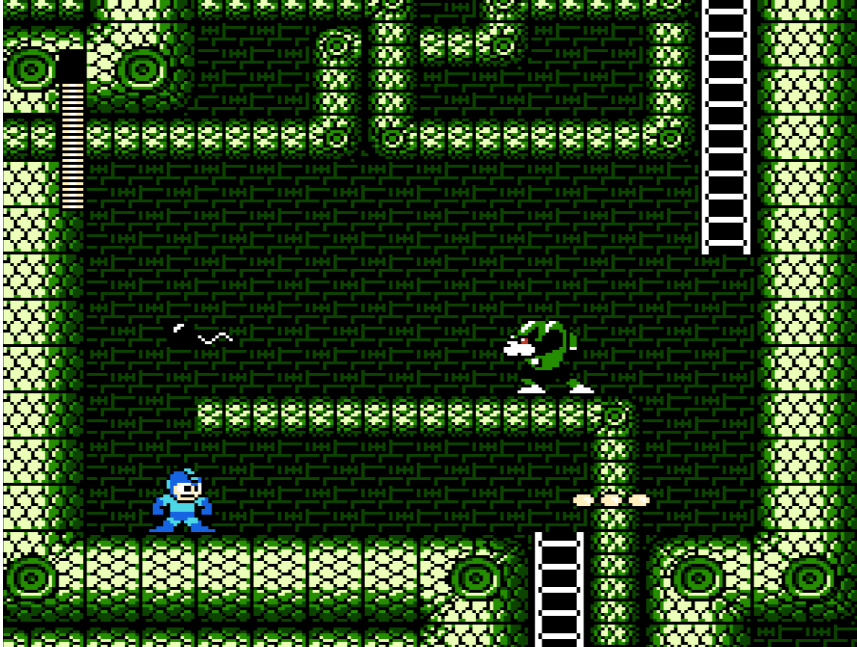
Epäonnistuminen tai häviö voivat myös olla tilaisuuksia opettaa pelaajaa. Epäonnistuessaan pelaaja pyrkii välttämään tilannetta, joka alun perin sai hänet epäonnistumaan. Virheestä oppineena pelaaja pelaa paremmin. Olettaen siis, ettei häviön syy ollut pelaajalle epäselvää. On myös mahdollista saada pelaaja vahingossa huomaamaan uusia mekaniikkoja. Esimerkiksi *Doomin* (Id Software, 1993) ensimmäisissä kentissä räjähtävät tynnyrit on usein sijoitettu vihollisten lähetyville. Voikin käydä niin, että pelaaja vahingossa ampuu vihollisten sijasta tynnyriä ja räjäyttää viholliset kuoliaaksi. Yllättyneenä ja viisaampana pelaaja oppii hyödyntämään räjähtäviä tynnyreitä tulitaiseluissa. (Mora ym. 2017, 6–7.)

### **4.2.3 Kuinka välttää pelaajan turhautuminen?**

Epäonnistuminen tai häviö ovat osa pelejä. Näin voi käydä jopa pelin alkumetreillä. Tämä ei välttämättä tarkoita, että pelin tulisi olla helpompi, vaan pelin tulisi lieventää häviöstä koituvaa turhautuneisuutta. Turhautumista voidaan lieventää esimerkiksi antamalla pelaajalle riskittömiä harjoittelu-tiloja.

Niin sanotut turva-alueet tarjoavat pelaajalle riskittömän tilan, jossa tämä voi omalla tahdillaan kokeilla pelin mekaniikkoja, esimerkiksi miten hahmolla liikutaan, mistä painikkeesta hypätään, kuinka näkymää ohjataan tai muuta vastaavaa (Mora ym. 2017, 7). Uusia vihollisia ja vaaroja voidaan esittää pelaajalle käyttäen turva-alueita. Esimerkiksi kuvassa 6 *Mega Man 3* (Capcom, 1990) esittelee ”Hammer Joe” vihollistyyppin siten, ettei pelaaja ole tämän tulilinjalla. Tämä antaa pelaajalle tilaa tarkkailla ja valmistautua vaarojen kohtaamiseen ja lieventää pelaajan turhautumista, jos tämä epäonnistuu kohtaamaan kyseisen vaaran.

Turhautumista voidaan lieventää myös nopeilla iteraatioilla. Iteraatiolla tarkoitetaan tässä tapauksessa uudelleen yritystä. Usein epäonnistuminen tarkoittaa kohtauksen uudelleen pelaamista, joskus jopa pidemmän matkan takaa. Turhautumista voidaan lieventää päästämällä pelaaja yrittämään uudestaan mahdollisimman pian.



KUVA 6. Pelaajalla on tilaa tarkkailla vihollisen hyökkäyksiä, ennen kuin loikkaa tämän tullinjalle (Mega Man 3, 1990).

## 5 ANALYYSI

Tässä luvussa syvennytään tutkimaan, miten pelit käytännössä opettavat pelaajansa pelaamaan kenttäsuunnittelun avulla. Analysoinnin kohteeksi päätyi indie-klassikko Shovel Knight, joka tunnetaan harrastepiireissä esimerkillisestä kenttäsuunnittelusta. Peli oli itselleni myös ennestään tuttu. Uudelleen pelatessani kirjasin ylös keinoja, joilla Shovel Knight johdattelee pelaajaa, opettaa pelaajaa ja kuinka peli välttää pelaajan turhautumisen.

### 5.1 Shovel Knight

Shovel Knight on Yacht Club Gamesin kehittämä ja julkaisema 2D-tasoloikkapeli, joka julkaistiin 2014 pelikonsolille ja tietokoneelle. Peli tuli pelaajien tietoisuuteen vuonna 2013 tämän ilmestyessä joukkorahoituspalvelu Kickstarteriin, jossa peliä markkinoitiin retrohenkisenä toimintaseikkailuna. Pelille oli selvästi kysyntää, sillä peli keräsi kampanjan loppuun mennessä yli 300 000 dollaria. Joukkorahoituksen menestyksen ansiosta peli sai julkaisun jälkeen runsaasti lisäsisältöä, kuten lisähaasteita, lisää pelattavia hahmoja ja jopa neljän pelaajan taistelupelimoodi.

Shovel Knight on 2D-sivuperspektiivistä kuvattu tasoloikkapeli, jossa ohjataankin nimikkoritaria, joka kantaa aseenaan lapiota. Shovel Knightin pelattavuus on yksinkertaista. Pelissä kuljetaan monien eri teemaisten kenttien halki, joiden päätteeksi kohdataan kentän päävastus. Päävastukset ovat myös ritareita omine oikkuineen, joiden mukaan kenttien teemat on luotu. Kenttien varrelta löytyy lukuisia vihollisia, sorablockeja ja erilaisia kenttäkohtaisia mekaniikkoja. Esimerkiksi liukkaat lattiat Polar Knightin jäisessä kentässä tai pimeät huoneet Spectre Knightin kummitusteemaisessa kentässä.

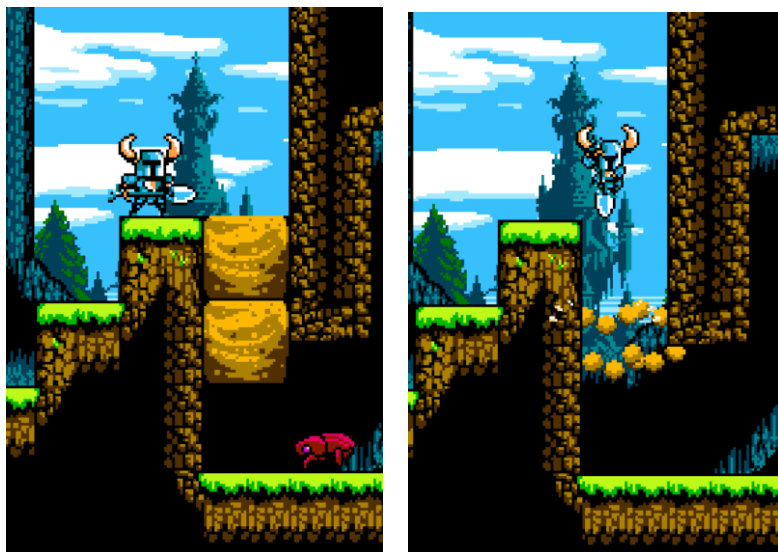
Pelissä voi liikkua vasemmalle ja oikealle, hypätä, lyödä lapiolla, ja hypyn aikana painamalla alas pelaaja tekee niin sanotun shovel dropin, jolla lapiosta tulee ikään kuin hyppykeppi. Tämän avulla voi ponnahtaa ilmaan sorablockien ja vihollisten päältä samalla tuhoten ne. Pelin edetessä pelaaja voi löytää niin kutsuttuja toissijaisia aseita, joita käytetään painamalla yhtä aikaa ylöspäin sekä lyöntipainiketta. Peli ei varsinaisesti kerro, mitä mikäkin painike tekee, mutta toimintanäppäimiä on vain kaksi. Pienellä kokeilulla nämä varmasti löytyvät. Ainoana poikkeuksena ovat toissijaiset aseet

eli relicit, joiden löytyessä ruudulle tulee tekstilaatikko, jossa kerrotaan millä näppäinyhdistelmällä asetta käytetään.

## 5.2 Kuinka Shovel Knight opettaa?

Shovel Knightin kentät ovat rakenteeltaan lineaarisia, eikä niissä ole mahdollisuutta eksyä. Pääasiassa peli siis johdattelee pelaajaa suoraan, mutta pelistä löytyy myös esimerkkejä epäsuorasta johdattelusta. Kenttien varrella pelaaja saattaa löytää esimerkiksi timantteja ja kultakolikoita, joita keräämällä pelaaja voi myöhemmissä kentissä ostaa päivityksiä, kuten lisää elämää. Nämä kerättävät esineet ovat paikoitellen sijoitettu pelaajan kulkemalle reitille kuin suuntaa antaen.

Shovel drop -mekaniikka havainnollistaa täydellisesti, miten Shovel Knightin kentät opettavat pelaajalleen sen mekaniikat ja haasteet, käyttäen antepieceä ja kontekstuaalista harjoitusta. Pelaaja väistämättä tutustuu Shovel dropiin pelin ensimmäisissä ruuduissa. Kentässä tulee kohta, jossa pelaajan tulisi pudottautua alas, mutta pelaajan tiellä on aiemmasta ruudusta tuttuja sorablockeja. Pelaaja pystyy lyömään ainoastaan vaakasuunnassa, joten pelaajan tulee keksiä käyttää shovel dropia tuhotakseen blockit. Kun blockit on tuhottu, pelaaja pääsee pudottautumaan alas ja jatkaamaan kentässä eteenpäin (kuva 7).



KUVA 7. Kohtaus pelin alkupuolella, jossa sorablockit estävät pelaajan etenemisen (Shovel Knight 2014).

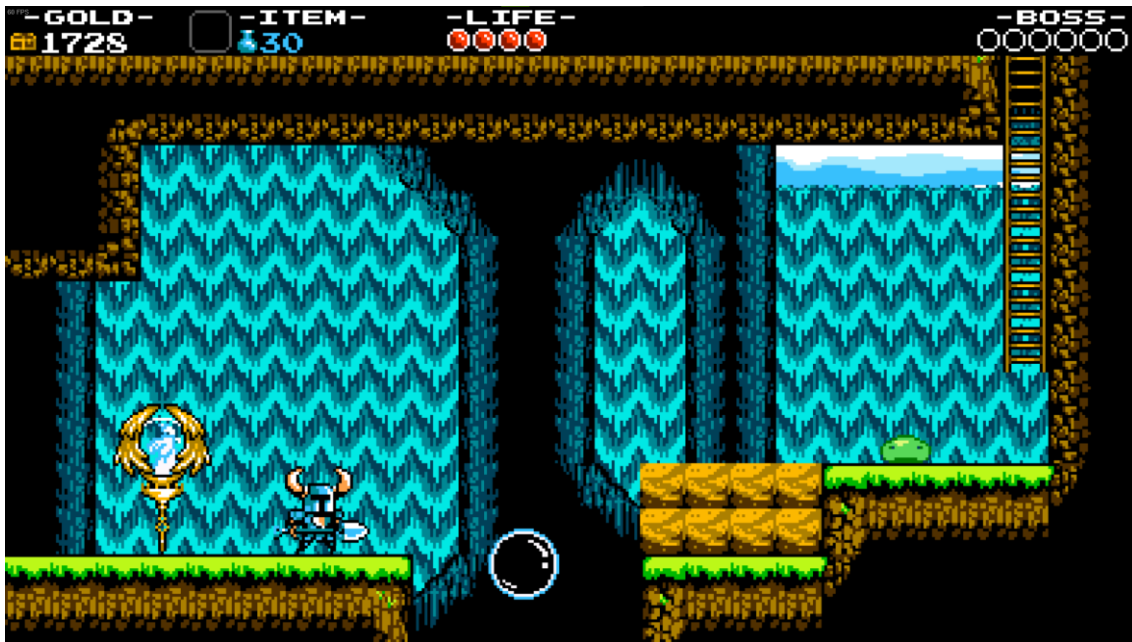
Peli laajentaa shovel dropin käyttömahdollisuuksia seuraavassa huoneessa. Pelaaja kohtaa kielekkeen, joka on liian kaukana ja liian korkealla hypättäväksi (kuva 8). Pelaajan ja kielekkeen välillä on kuoppa, jonka ylle leijailee ruudun alalaidasta kuplia. Pelaajan on tarkoitus ponnahtaa shovel dropin avulla kuplan päältä korkealle kielekkeelle, joka samalla tuhoaa kuplan. Ruudulla ei ole välitöntä vaaraa. Jos pelaaja sattuukin epäonnistumaan hypyssä, saattaa hän korkeintaan ottaa pienen määrän vahinkoa kuplasta, ja kuplan tuhoutuessa paikalle leijuu aina uusi. Kuolemaan ei voi pudota, vaan kuoppaan pudotessaan voi vain hypätä takaisin aloitus kielekkeelle ja yrittää uudelleen.



KUVA 8. Ruudulla ei ole välitöntä vaaraa (Shovel Knight 2014).

Kuplat esiintyvät uudestaan useamman kerran kentän varrella, joka kerta hieman haastavampana. Seuraavan kerran kun kuplat esiintyvät on tilanne melkein sama, mutta tällä kertaa kuplan alapuolella ammottaa kuilu ja toisella puolella odottaa sorablokkeja sekä vihollinen (kuva 9). Siltä varalta, että pelaaja sattuukin putoamaan rotkoon, on huoneen alkuun sijoitettu jatkopiste, josta pelaaja voi kuoltuaan yrittää uudestaan.





KUVA 9. Kuplat palaavat uusien haasteiden muodossa (Shovel Knight 2014).

Kentän varrella esiintyy myös välivastuksena samoja kuplia sylkeviä lohikäärmeitä. Kuplien päältä ponnahtaen pelaaja voi kätevämmiin hyökätä lohikäärmeen kimppuun ylhäältä päin. Myöhemmät kentän osat koettelevat pelaajan juuri oppimia taitoja, kun pelaaja kohtaa isompia kUILUJA, joiden ylittämiseksi on ponnahtettava useamman kuplan päältä. Shovel drop ei kuitenkaan ole näin tilannekohtainen, vaan mekaniikalle löytyy käyttöä lähes joka ruudussa, ja näin mekaniikka iskostuu pelaajalle. Peli opettaa lähes kaiken näillä periaatteilla, kuten pelaajan perusliikkeet, vihollisten käyttäytymisen ja kenttäkohtaiset haasteet.

### 5.3 Yhteenveto

Shovel Knight johdattelee pelaajaa suoraan lineaarisuutensa ansiosta, mutta kerättävät esineet johdattelevat pelaajaa myös epäsuorasti. Peli opettaa uudet mekaniikat ja haasteet riskittömässä tilassa, jossa pelaajalla on mahdollisuus omalla tahdillaan kokeilla, miten peli toimii. Pelin edetessä pelaajan taitoja koetellaan uusien haasteiden muodossa, jotka laajentavat pelin mekaniikkoja. Tämä ei päde ainoastaan ensimmäiseen kenttään, vaan jokaisessa kentässä pelaaja kohtaa uusia kenttäkohtaisia haasteita. Esimerkiksi Spectre Knightin kentässä esitellään pimeät huoneet, joissa pelaaja voi nähdä ainoastaan salaman iskun hetkellä. Lisäksi kentässä tulee vastaan tasoja, jotka uppoavat kuolettavaan veteen, jos kaksi hahmoa seisovat sen päällä yhtä aikaa. Peli esittelee nämä haasteet erikseen luoden uusia haasteita joka kohtaamisella, ja kentän loppupuolella nämä

mekaniikat näyttäytyvät yhtä aikaa. Vaikka pelaaja kohtaa pelin aikana lukuisia eri haasteita, mikään niistä ei tunnu epärealistiselta lukuisten "turvaverkkojen" ansiosta. Epäonnistuessaan pelaaja ymmärtää virheensä, mikä kannustaa yrittämään uudelleen.

## 6 POHDINTA

Kuinka pelit opettavat pelaajaa? Tiesin vastauksen olevan kenttäsuunnittelussa, mutta halusin selvittää, kuinka se tapahtuu käytännössä. Tietoperustaa kerätessäni kohtasin termin ”orgaaninen tutoriaali”, jolla tarkoitetaan pelaajan opettamista kenttäsuunnittelun avulla. Keinoja pelaajan opettamiseen löytyi monia. Tiivistetysti hyvä orgaaninen tutoriaali johdattaa pelaajan päämääräänsä, iskostaen pelin toiminnallisuudet jatkuvalla toistolla ja samalla antaen pelaajalle turvallisia alueita, joissa harjoitella uusia kykyjään. Ihanteellisesti tämä tapahtuu huomaamattomasti. Orgaanisten tutoriaalien tavoitteista turhautumisen lievitys lienee kaikista tärkein. Käytännössä orgaaniset tutoriaalit opettavat pelaajaa yrityksen ja erehdyksen kautta, on siis pidettävä huolta, etteivät erehdykset saa pelaajaa luovuttamaan.

Tietoperustan löytäminen tutoriaaleista oli yllättävän vaikeaa. Iso osa tutoriaaleja käsittelevistä lähteistä keskittyi moittimaan tutoriaaleja tai vertailemaan hyviä ja huonoja tutoriaaleja, vaikka keskityin vain alan ammattilaisten artikkeleihin ja luentoihin. Neljännessä luvussa yritin parhaani mukaan luokitella yleiset tutoriaalit järkeviin kategorioihin, koska niistä ei löytynyt yhteisesti hyväksyttyä luokittelua. On varmaankin mahdollista kategorisoida ne toisellakin tavalla. Asiasta teki vaikeampaa myös se, ettei orgaaninen tutoriaali ole yleisesti käytössä oleva termi. Kenttäsuunnittelusta löytyi materiaalia helpommin, vaikkakin merkittävä osa siitä on melko vanhaa tai yksipuolista. Esimerkiksi iso osa materiaalista keskittyi pääasiassa räiskintäpelien kenttäsuunnitteluun.

Miksi sitten kaikki pelit eivät opeta kenttäsuunnittelun avulla ja miksi yhä useammat pelien kehittäjät turvautuvat ennemmin tutoriaaleihin, vaikka niillä on negatiivinen maine? Vastaus kysymyksiin on niinkin tylsä, ettei tällainen opetustyyli sovellu kaikkiin peleihin. Näin on esimerkiksi vaativissa strategiapeleissä tai managerointi peleissä, joissa on kerroksittain järjestelmiä ja suurin osa pelattavuudesta tapahtuu valikoissa. Mitä monimutkaisempi peli, sitä vaikeampaa sanaton opettaminen on. Orgaanisen tutoriaalın omaava kenttäsuunnittelu vaikuttaa yleensä sopivan yksinkertaisempiin peleihin, kuten tasoloikkapeleihin tai ammutapeleihin. Eikä aina edes niihinkään, sillä joskus pelin rakenne voi tehdä kenttäsuunnittelulla opettamisen mahdottomaksi.

Kuten jo aiemmin sanoin, tutoriaaleissa ei itsessään ole mitään vikaa, mutta peliin sopimaton tutoriaali voi tehdä pelin opettelusta todella ärsyttävää. Uskon tutoriaalien negatiivisen maineen johtuvan juuri tästä. Esimerkiksi pelin pysäyttävä pop-up-tutoriaali soveltuu huonosti nopeatempoiseen tasoloikkapeliin.

Henkilökohtaisesti toivoisin näkeväni orgaanisen tutoriaalimateriaalin omaavaa kenttäsuunnittelua lisää. Vaikkakin peleistä löytyy nykyään hyviä perinteisiä tutoriaaleja, uskoisin opettavasta kenttäsuunnittelusta edelleen löytyvän käyttämätöntä potentiaalia. Kenttäsuunnittelu tekee peleistä uniikkeja verrattuna muihin medioihin, ja on ilo nähdä, kun siitä otetaan kaikki hyöty irti.

## LÄHTEET

Bramble, Ross 2022, What are Indie Games?. Game Maker. Hakupäivä 7.10.2022. <https://game-maker.io/en/blog/what-are-indie-games#what-is-an-indie-game>.

CD PROJEKT RED. Encounter Designer. Hakupäivä 4.10.2022. <https://jobs.smartrecruiters.com/CDPROJEKTRED/743999715095457-encounter-designer>.

De, Soham 2021. What Are Mods in Video Games?. Make Use of. Hakupäivä 7.10.2022. <https://www.makeuseof.com/mods-video-games/>.

Dreo, Marion 2021. Combat Game Designer: What Do Combat Designers Do? What Unique Skills Are Required for Combat Design? Animation Career Review. Hakupäivä 30.9.2022. <https://www.animationcareerreview.com/articles/combat-game-designer-what-do-combat-designers-do-what-unique-skills-are-required-combat-des>.

Byrne, Edward 2005. Game level design. Hingham, Mass. Charles River Media cop.

England, Liz 2014. Types of Designers. Game Developer. Hakupäivä 7.10.2022. <https://www.gamedeveloper.com/design/types-of-designers>.

Extra Credits 2015. Procedural Generation – How Games Create Infinite Worlds – Extra Credits. Hakupäivä: 7.10.2022. <https://www.youtube.com/watch?v=TgbuWfGeG2o>.

Karthikeyan, Kaavya 2021. AAA Games: A Comprehensive Guide. Gameopedia. Hakupäivä 7.10.2022 <https://www.gameopedia.com/comprehensive-guide-to-aaa-games/>.

Karthikeyan, Kaavya 2022. The Evolution of First Person Shooter (FPS) Games. Gameopedia. Hakupäivä 7.10.2022 <https://www.gameopedia.com/evolution-of-first-person-shooter-fps-games/>.

Mora, Vicente & Durán, Francisco José & Molina-Carmona, Rafael & Llorens, Faraón. 2017 Subliminal Learning. What Do Games Teach Us?. Researchgate. Hakupäivä 1.9.2022. [https://www.researchgate.net/publication/318148854\\_Subliminal\\_Learning\\_What\\_Do\\_Games\\_Teach\\_Us](https://www.researchgate.net/publication/318148854_Subliminal_Learning_What_Do_Games_Teach_Us).

Rogers, Scott 2014. Level Up! The Guide to Great Video Game Design. 2. painos. Chichester, Wiley.

De Jong, Sojerd "Hourences" 2008. Hows and Whys of Level Design. Second Edition. Sojerd De Jong. <http://www.hourences.com/product/the-hows-and-whys-of-level-design-2/>.

The Level Design Book 2022a. What is level design. Hakupäivä 11.5.2022. <https://book.leveldesignbook.com/introduction>.

The Level Design Book 2022b. History of the level designer. Hakupäivä 11.5.2022. <https://book.leveldesignbook.com/culture/history-level-designer>.

The Level Design Book 2022c. How to make a level. Hakupäivä 11.5.2022. <https://book.leveldesignbook.com/process/overview>.

Unity. General game development terms. Hakupäivä 30.9.2022. <https://unity.com/how-to/beginner/game-development-terms#general-game-development-terms>.

Unreal Engine 2021. What is an environment artist? Hakupäivä 30.9.2022. <https://www.unrealengine.com/en-US/blog/what-is-an-environment-artist>.

Zaiets, Slava 2020. Why AAA Studios Shift to Games-as-a-Service (GaaS) Model. Gridly. Hakupäivä 30.9.2022. <https://www.gridly.com/blog/games-as-a-service/>.

## KUVALÄHTEET

KUVA 1. Baker, Brian 2019. Kuvakaappaus. Why build a blockout? Blockout. The Level Design Book. Hakupäivä 1.11.2022. <https://book.leveldesignbook.com/process/blockout>

KUVA 2. Heretic. 1994. Id Software. Kuvakaappaus pelistä. Viitattu 11.8.2022.

KUVA 3. Bayonetta. 2009. Sega. Kuvakaappaus pelistä. Viitattu 22.9.2022.

KUVA 4. Far Cry 3: Blood Dragon. 2013. Ubisoft. Kuvakaappaus käyttäjältä iampanax. Viitattu 25.7.2022. <https://youtu.be/f9slerS21qg?t=552>

KUVA 5. Journey. 2012. Thatgamecompany. Kuvakaappaus käyttäjältä IAmSp00n. Viitattu 4.10.2022. <https://youtu.be/bkL94nKSd2M?t=130>

KUVA 6. Mega Man 3. 1990. Capcom. Kuvakaappaus pelistä. Viitattu 9.9.2022.

KUVA 7. Shovel Knight. 2014. Yacht Club Games. Kuvakaappaus pelistä. Viitattu 9.6.2022.

KUVA 8. Shovel Knight. 2014. Yacht Club Games. Kuvakaappaus pelistä. Viitattu 9.6.2022.

KUVA 9. Shovel Knight. 2014. Yacht Club Games. Kuvakaappaus pelistä. Viitattu 9.6.2022.