

Saska Sinkkonen

**Unity-pelimoottorin tarkastelu aloittelevan  
indie-kehittäjän näkökulmasta**

Tietotekniikan kandidaatintutkielma

19. joulukuuta 2017

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Saska Sinkkonen

**Yhteystiedot:** saansink@student.jyu.fi

**Työn nimi:** Unity-pelimoottorin tarkastelu aloittelevan indie-kehittäjän näkökulmasta

**Title in English:** Otsikko englanniksi

**Työ:** Kandidaatintutkielma

**Sivumäärä:** 30+0

**Tiivistelmä:** Tässä tutkielmassa tarkastellaan Unity-pelimoottoria aloittelevan indie-kehittäjän näkökulmasta. Tutkielmassa selvitetään, miksi Unity on niin suosittu aloittelevien indie-kehittäjien käytössä ja mitkä sen ominaisuudet tukevat heitä. Aluksi esitellään Unitya yleisesti. Sen jälkeen esitetään lähdekirjallisuuden pohjalta tärkeimmät kriteerit pelimoottorin valinnassa ja pohditaan, mitkä niistä ovat olennaimpia tutkielman näkökulman kannalta. Unitya arvioidaan näillä kriteereillä ja saadaan selville, että erityisesti sen yhteisö, tutoriaalit ja hinta ovat tärkeitä syitä sen suosiolle aloittelevien indie-kehittäjien käytössä.

**Avainsanat:** Unity, pelimoottorit

**Abstract:** This thesis reviews the Unity-game engine from the perspective of a beginner indie developer. The aim is to find out, why Unity is so popular with beginner indie developers and which features support them. First Unity is introduced generally. After that the most important criteria for selecting a game engine are presented and whether or not they are relevant from this thesis' perspective is discussed. Unity is evaluated with these criteria and it is found that especially the community, tutorials and price are important reasons for Unity's popularity in beginner indie developer community.

**Keywords:** Unity, game engines

## **Kuviot**

Kuvio 1. Pelimoottoreiden uudelleenkäytettävyyden kirjo (Gregory 2009). .....	3
Kuvio 2. Unityn käyttöliittymä. ....	9

## Sisältö

1	JOHDANTO .....	1
2	UNITY-PELIMOOTTORIN ESITTELY.....	3
	2.1 Mikä on pelimoottori? .....	3
	2.2 Unity lyhyesti .....	4
	2.3 Unity indie-kehittäjien työkaluna.....	5
3	PELIMOOTTOREIDEN ARVIOINTI .....	7
	3.1 Pelimoottorin vaatimukset.....	7
	3.2 Arviointikriteerit .....	10
4	UNITYN ARVIOINTI .....	15
	4.1 Pelimoottorin vaatimukset.....	15
	4.2 Ohjelmiston hinta .....	15
	4.3 Oppimisen helppous.....	16
	4.4 Työkalujen ja kirjastojen monipuolisuus.....	18
	4.5 Julkaisu- ja kehitysalustat .....	19
5	YHTEENVETO .....	21
	KIRJALLISUUTTA .....	23

# 1 Johdanto

Digitaaliset pelit kasvattavat nykyaikana alati suosiotaan ja viime vuosien aikana ne ovat tulleet osaksi ihmisten päivittäistä elämää. Yhdysvalloissa kuluttajat käyttivät yhteensä yli 30 miljardia dollaria videopeliteollisuuden vuonna 2016 (ESA 2017). Pelien suosion noustessa myös uusia pelikehitykseen tarkoitettuja ohjelmistoja ilmestyy koko ajan ja pelikehittäjiä on valittava lukemattomien eri tuotteiden joukosta itselleen sopivimmat. Devmaster-tietokanta listaa tällä hetkellä 376 eri pelimoottoria (Devmaster.net 2017).

Pelimoottoreista on myös tullut entistä helppokäyttöisempiä ja helpommin saatavia myös harrastelijoille. Nykyään kuka tahansa voi tehdä suhteellisen vähällä vaivalla hienojakin 3D-pelejä kotonaan, mikä tekee aiheesta erityisen ajankohtaisen. Siihen liittyvä tutkimustieto on kuitenkin vielä vähäistä ja nopean kehityksen vuoksi sekin on usein vanhentunutta. Siksi onkin tärkeää paitsi pelikehittäjiä myös pelimoottoreiden kehittäjiä kannalta selvittää, mikä tekee ohjelmistosta hyvän ja kannustaa uusia käyttäjiä sen pariin. Tässä tutkielmassa keskitytään yhden pelimoottorin, Unityn, tarkasteluun.

Unity on aloittelijaystävällisenä pidetty pelimoottori, joka soveltuu kuitenkin lähes kaikenikäisten pelien kehittämiseen. Tässä tutkielmassa tarkastellaan ja arvioidaan Unitya aloittelevan indie-kehittäjän näkökulmasta. Tarkoituksena on selvittää, miksi se on niin suosittu aloittelijoiden ja indie-kehittäjiä keskuudessa ja mikä tekee siitä hyvän nimenomaan heille. Unitya ei kuitenkaan vertailla kattavasti muihin pelimoottoreihin, joten tutkielma ei selvitä, miksi se on suositumpi kuin jokin toinen pelimoottori tai kuinka hyvä se on suhteessa muihin samanlaisiin ohjelmistoihin.

Tutkielma toteutaan kirjallisuuskatsauksena. Luvussa 2 selitetään pelimoottorin käsite ja esitellään Unity. Siinä käydään lävitse Unityn perustietoja ja kerrotaan sen nykytilasta ja menestyksestä. Luvussa 3 tutkitaan pelimoottoreiden vaatimuksia sekä kriteereitä, joilla pelimoottoreita voidaan arvioida ja pohditaan, mitkä näistä ovat tärkeimpiä tutkielman näkökulmasta. Luvussa 4 tarkastellaan Unityä löydettyjen

arvointikriteerien pohjalta. Siinä arvoidaan, kuinka hyvin Unity toteuttaa pelimootorilta halutut ominaisuudet ja mitä parannettavaa sillä kenties vielä olisi. Luvussa 5 on yhteenveto tutkielman tuloksista ja mahdollisia käyttötarkoituksia niille.

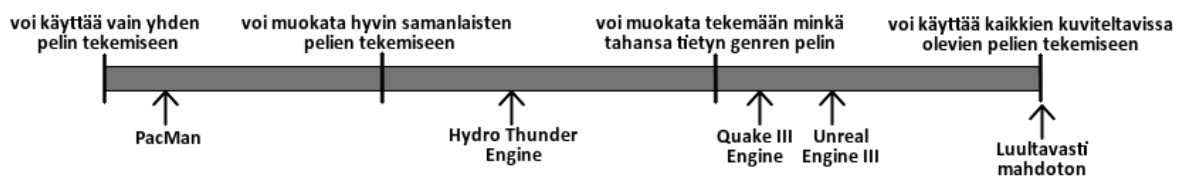
## 2 Unity-pelimoottorin esittely

Tässä luvussa kerrotaan, mikä pelimoottori on ja esitellään Unityn perustietoja. Ensimmäisessä alaluvussa määritellään pelimoottorin käsite ja listataan niiden yleisiä ominaisuuksia. Toisessa alaluvussa käydään lävitse Unityn historiaa, perustietoja sekä tilastoja. Kolmannessa alaluvussa esitellään indie-käsite ja tarkastellaan lyhyesti Unitya indie-näkökulmasta.

### 2.1 Mikä on pelimoottori?

Pelimoottorin tarkka määrittelyminen on hieman ongelmallinen tehtävä, sillä sille ei ole olemassa yhtä tiettyä määritelmää (Thorn 2011) ja pelimoottorin ja itse pelin raja voi joissain tapauksissa olla hämärä (Gregory 2009). Etenkin monissa vanhemmissa peleissä pelimoottoria ja pelin sisältöä on usein vaikea erotella.

Gregory (2009) määrittelee pelimoottorin laajennettavissa olevana ohjelmistona, jota voidaan käyttää pohjana monille eri peleille ilman merkittäviä muutoksia. Käytännön kannalta pelimoottori on siis ohjelmisto, joka helpottaa ja nopeuttaa videopelien kehittämistä. Se sisältää pelin yleiset ja uudelleenkäytettävät komponentit, mutta ei itse pelin sisältöä. Pelimoottorista riippuen, komponentit voivat olla hyvin uudelleenkäytettäviä, jolloin moottorilla voi tehdä monia erilaisia pelejä tai toisaalta pelimoottori voi olla tarkoitettu vain yhteen tiettyyn peliin. Tällä perusteella pelimoottoreita voidaan sijoittaa uudelleenkäytettävyyden kirjolle (Kuvio 1).



Kuvio 1. Pelimoottoreiden uudelleenkäytettävyyden kirjo (Gregory 2009).

Pelimoottoriin kuuluu yleensä ainakin fysiikkamoottori, jonka avulla kappaleet saadaan liikkumaan ja törmäämään sekä piirtomoottori, joka huolehtii pelin grafiikoista (Oliveira, Oliveira & Pedrini 2016). Lisäksi pelimoottorista löytyy usein työkalut, jotka auttavat muun muassa äänien, animaatioiden ja tekoälyn toteuttamisessa.

Tällä hetkellä tunnetuimpia pelimoottoreita ovat esimerkiksi Epic Gamesin Unreal Engine, Crytekin CryEngine, Valven peleistä tuttu Source sekä tässä tarkasteltava Unity.

## 2.2 Unity lyhyesti

Unity on Unity Technologiesin kehittämä pelimoottori. Se on tunnettu varsinkin indie-kehittäjien ja harrastelijoiden suosimana ohjelmistona, vaikkakin sitä ovat käyttäneet myös jotkin suuremman budjetin pelistudiot.

Unityn historia alkaa vuodesta 2002, kun kaksi pelikehityksestä kiinnostunutta tanskalaista tapasivat keskustelufoorumilla ja päätyivät yhdessä tekemään uudenlaisen pelimoottorin (Haas 2014). Kolmen vuoden päästä, vuonna 2005, julkaistiin Unity 1.0. Suuremman yleisön huomion se saavutti kuitenkin vasta myöhemmin, kun selainpelien suosio räjähti ja Unity oli ensimmäisiä ohjelmistoja, joka tarjosi hyvät työkalut niiden kehittämiseen (Fear 2009).

Alun perin Unity oli tarkoitettu vain 3D-pelien kehitykseen, mutta se on tukenut versiosta 4.3 lähtien myös kaksiulotteisten pelien kehitystä siihen erikseen suunnitelluilla työkaluilla (Juhl 2013). Vaikka Unity itsessään on toteutettu C:llä ja C++:lla, kirjoitetaan skriptit C#:lla. Myös JavaScriptin kaltaista UnityScriptiä voi käyttää, mutta sen virallinen tuki lopetettiin elokuussa 2017 (Fine 2017).

Unityn tämän hetkinen versio 2017.2 tukee 29 eri julkaisualustaa ja onkin erityisen käytetty esimerkiksi Android- ja Oculus Rift -pelien kehityksessä (Unity 2017a). Perusversio on ilmainen, mutta maksullisilla versioilla saa joitakin lisäominaisuuksia. Unity myös vaatii ostetun version mikäli yrityksen vuosituotto ylittää 100 000 dollaria. Tämän tutkielman näkökulman kannalta ilmaisversio ei kuitenkaan merkit-



tävästi eroa maksullisesta ja tutkielmassa tarkastellaan nimenomaan Unityn ilmaisversiota.

Nykyään Unity on noussut kahden kaverin kyhäelmästä maailman suosituimpien pelimoottoreiden joukkoon. Unity kertoo verkkosivuillaan, että sillä tehdään enemmän pelejä kuin millään muulla teknologialla. Tuhannesta suosituimmasta ilmais-mobiilipelistä 34% on kehitetty Unitylla. Unitylla tehtyjä tuotteita löytyy yli kahdesta miljardista mobiililaitteesta ja vuonna 2016 sillä kehitettyjä pelejä ladattiin miljardeja kertoja. (Unity 2017a).

Varmastikin tunnetuin Unity-peli on Blizzard Entertainmentin suuren suosion saavuttanut korttipeli *Hearthstone*. *Hearthstone* tuokin hyvin esille Unityn monialustatuen, sillä sitä voi pelata niin PC:llä kuin tableteilla ja kännyköilläkin (Blizzard Entertainment 2017). Muita tunnettuja Unitylla kehitettyjä pelejä ovat muun muassa Squadin avaruuslentosimulaattori *Kerbal Space Program* sekä syksyllä 2017 julkaistu, suureksi ilmiöksi noussut *Cuphead* (Unity 2017b).

### 2.3 Unity indie-kehittäjien työkaluna

Indie-peleillä tarkoitetaan itsenäisesti kehitettyjä pelejä. Van Best (2011) määrittelee pro gradussaan, että indie-pelit ovat pelejä, jotka ovat kehitetty pienissä tiimeissä suhteellisen vähällä henkilötyötunneilla ja joiden levitys tapahtuu digitaalisesti. Ne luottavat vähemmän graafiisiin ominaisuuksiin ja realismiin kuin AAA-pelit. Määritelmä ei ole tietenkään yksimielisesti sovittu. Kogel (2012) kuvaa indie-pelejä ja lomman periaatteen mukaisesti: ”Ollakseen indie, pelin kehittäjien täytyy osoittaa tiettyä asennetta pelejä ja peliteollisuutta kohtaan – kehittäjien täytyy osoittaa jotain piittaamattomuutta rahasta.” Onkin ehkä turhaa miettiä virallisia määritelmiä termille, sillä pelien luokittelu ei ole koskaan niin mustavalkoista (Simon 2013). Tämän tutkielmaan tarkoituksiin riittää ajatella indie-pelejä peleinä, jotka on tehty vähällä rahalla ja pienellä kehitystiimillä.

Jo Unityn ensimmäisen version julkaisusta lähtien suuri osa sen käyttäjistä on ollut harrastelijoita ja indie-kehittäjiä (Haas 2014). Unitya kehitetään myös erityisesti

indie-yleisö huomioon ottaen, minkä osoittaa esimerkiksi vuonna 2009 tulleet hinnoittelumuutokset. Unityn perusversio tuli ilmaiseksi, mihin Unityn perustajajäsen David Helgason kertoi syyksi sen, että ohjelmiston haluttiin tavoittavan mahdollisimman suuri yleisö ja vahvistavan asemaansa indie-yhteisössä (Remo 2009).

Unityn aseman indie-kehittäjien työkaluna huomaa nopeasti katsomalla listaa sillä tehdyistä peleistä. Hearthstonea lukuunottamatta on hyvin vähän suurien pelistudioiden kehittämiä Unity-pohjaisia pelejä. Monet tunnetuimmistakin peleistä, kuten aiemmin mainitut Kerbal Space Program ja Cuphead ovat indie-kehittäjien tuotoksia. Vertailuna voi esittää esimerkiksi listaa Unityn kilpailevaa moottoria, Unreal Engineä, käyttävistä peleistä. Muun muassa BioWaren kehittämä ja EA:n julkaisema Mass Effect -trilogia, Warner Bros.:n julkaisemat Batman: Arkham -pelit sekä suuri osa Ubisoftin Tom Clancy's -sarjasta ovat kaikki kehitetty jollain Unreal Enginen versiolla (Unreal Engine 2017).

### 3 Pelimoottoreiden arviointi

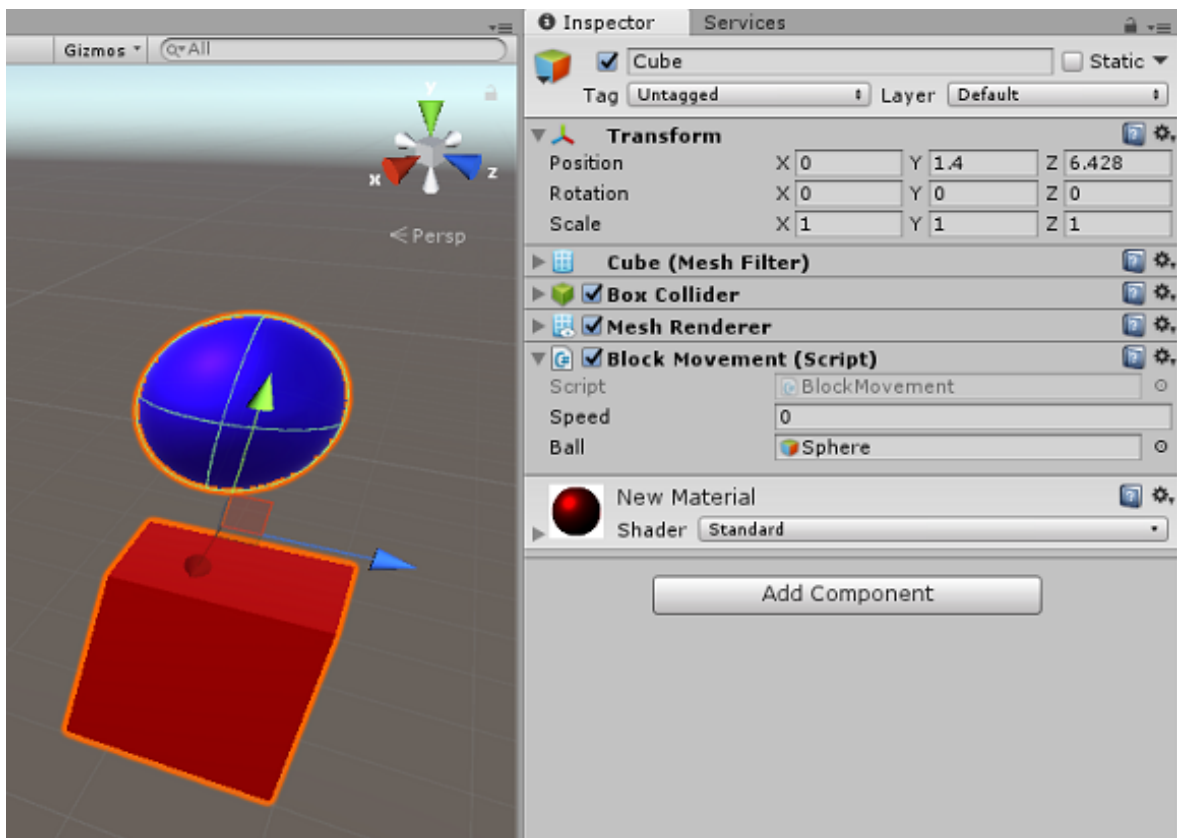
Pelimoottoreiden arviointiin ja vertailuun liittyen on tehty vain vähän tieteellistä tutkimusta. Lisäksi, koska pelimoottoreita voi käyttää moneen muuhunkin tarkoitukseen kuin pelien kehittämiseen, useat tehdyistä tutkimuksista liittyvät pelimoottoreiden arviointiin vain tietyn, täsmällisen käyttötarkoituksen kannalta. Esimerkiksi matematiikan opetus lapsille (Fu, Jensen & Hinkelman 2008) tai kirurginen koulutus (Marks, Windsor & Wünsche 2007). Monissa tutkimuksissa tarkastelu on rajattu myös pelityypin kannalta, esimerkiksi hyöty- tai mobiilipeleihin. Tulokseksi saadut arviointikriteerit voivat kuitenkin päteä myös yleisesti ja siksi aineistona on käytetty myös tutkimusta, joka ei välttämättä käsittele pelimoottoreita samasta, vain aloittelevien indie-kehittäjien näkökulmaan rajauksesta. Tästä johtuen on kuitenkin tärkeää, että on pohdittu ja perusteltu, miksi tietyt kriteerit ovat tärkeitä juuri tämän tutkielman näkökulmasta. Tässä luvussa tarkastellaan alkuun pelimoottoreilta vaadittavia ominaisuuksia. Toisessa alaluvussa tutkitaan kriteereitä, joilla pelimoottoreja voidaan arvioida ja perustellaan, mitkä näistä ovat olennaisimpia tämän tutkielman näkökulmasta.

#### 3.1 Pelimoottorin vaatimukset

Gregory (2009) luettelee, että kaikki 3D-pelit vaativat käyttäjän syötteitä, 3D-kappaleiden ja tekstin piirtämistä sekä äänisysteemin. Pelimoottorin tarkoitus on nopeuttaa pelinkehitystä, joten on tärkeää, että ne tarjoavat mahdollisuuden kehittää pelejä, ilman yksityiskohtaista tietoa matalan tason ääni- tai grafiikkaohjelmoinnista (Bishop, Eberly, Whitted, Finch & Shantz 1998). Tämä on vielä erityisen oleellista tutkielman näkökulmasta, koska pienten kehitysryhmien takia indie-projekteissa ajan säästäminen on välttämätöntä. Koska kehittäjiä on vähän, on tärkeää että sama henkilö pystyy huolehtimaan esimerkiksi grafiikkaan ja ääniin liittyvästä ohjelmoinnista sen sijaan, että jokaiselle osa-alueelle tarvittaisiin oma asiantuntijansa. Pelimoottorilta voidaan siis vaatia jonkinlainen piirtomoottori, äänisysteemi, sekä tapa käsitellä käyttäjän syötteitä. Nimenomaan 3D-kappaleiden piirtämistä taas ei voi pi-

tää pelimoottorilta välttämättömänä ominaisuutena, sillä on myös moottoreita, jotka ovat tarkoitettu vain 2D-pelien kehittämiseen, kuten Torque2D tai Löve.

Voidaan myös perustella, että aloittelevien kehittäjien kannalta välttämätön osa pelimoottoria on kehitystyökalut. Kehitystyökaluilla tarkoitetaan peliohjelmiston osia, jotka eivät ole käytössä itse pelin suoritusajana, mutta joita käytetään peliä tehtäessä. Siis esimerkiksi käyttöliittymä, jossa muodostetaan varsinainen pelimaailma yhdistelemällä grafiikkamalleja, ääniä, animaatioita ja skriptejä. Hyvillä kehitystyökaluilla on mahdollista nopeuttaa pelinkehitysprosessia huomattavasti. (Tukiainen 2013). Hyvin suunniteltu käyttöliittymä on todella tärkeää aloitteleville kehittäjille, sillä sen avulla pelin kokonaisuus on helpompi hahmottaa. Sillä voidaan myös merkittävästi vähentää vaadittavaa ohjelmointia. Kuviossa 2 näkyy rajattu kuvakaappaus Unityn käyttöliittymästä. Oikealla on valittuun objektiin ja siihen kiinnitettyihin skripteihin liittyviä muuttujia. Muun muassa objektille asetettua liikkumisnopeutta, materiaalia tai sen skripteissä määriteltyjä muita objekteja voidaan muuttaa suoraan käyttöliittymän kautta ilman, että kehittäjän täytyy kirjoittaa merkkiäkään koodia.



Kuvio 2. Unityn käyttöliittymä.

Bishop ym. (1998) mainitsevat, että yksi pelimoottoreilta vaadittava perusominaisuus on nopeus. Tämä on melko itsestäänselvää, sillä pelimoottorin pitää pystyä piirtämään ruudulle kymmeniä kuvia sekunnissa, ja jos toteutus pelimoottorin puolelta on hidas, ei pelin kehittäjä voi asialle enää mitään. Nopeuden lisäksi moottorilta vaaditaan myös vakautta. Pelien pelaaminen on viihdettä ja jos sitä täytyy esimerkiksi uudelleenkäynnistää koko ajan, on se pelaajalle todella turhauttavaa. (Kallio-Kujala 2016). Nopeutta ja vakautta voisi arvioida tarkemminkin, mutta indie-pelien vähäisempien graafisten ja muiden vaatimusten vuoksi kumpikaan on harvoin ongelma tämän tutkielman näkökulmasta.

Muita yleisiä ja tärkeitä, mutta ei välttämättä vaadittavia pelimoottorin ominaisuuksia ovat esimerkiksi fysiikkamoottori, skriptirajapinta, tekoäly, verkkokomponentti ja animaattori (Tukiainen 2013)

## 3.2 Arviointikriteerit

Petridis, Dunwell, de Freitas & Panzoli (2010) tarkastelevat artikkelissaan perusteita pelimoottorin valinnalle kehitettäessä hyötypelejä. Hyötypeleillä tarkoitetaan pelejä, joiden pääasiallinen tarkoitus ei ole viihdyttää pelaajaa (Ritterfeld, Cody & Vorderer 2009) vaan esimerkiksi opettaa tai välittää jotain sanomaa. Vaikkei tässä tutkielmassa erityisesti hyötypelien kehitystä tarkastellakaan, voi niissä käytettävien pelimoottoreiden halutuilla ominaisuuksilla olla yhtäläisyyksiä muiden tyyppisten pelien kehitykseen käytettävien moottoreiden kanssa. Petridis ym. (2010) esittävät viisi vaatimusta hyötypeleihin käytettäville pelimoottoreille: todenmukaisuus, saavutettavuus, verkostoituminen, heterogeenisyys ja koostettavuus.

Grafiikkojen ja mekaniikkojen todenmukaisuus on tärkeää, jos kyseessä on esimerkiksi jokin opetuskäyttöön tarkoitettu simulaattoripeli. Indie-kehittäjälle viimeisen päälle todenmukaiset grafiikat eivät kuitenkaan useimmiten ole ensimmäinen prioriteetti, mikä käy ilmi esimerkiksi aiemmin pohditusta indie-pelin määrittelystä, jossa todettiin niiden grafiikkaominaisuuksien olevan yleensä yksinkertaisia.

Saavutettavuus taas on selvästikin hyvin tärkeä asia tämänkin tutkielman näkökulmasta. Petridis ym. (2010) perustelevat helppouden tärkeyttä sillä, että hyötypelien kehittäjät saattavat olla muiden alojen osaajia, eivätkä välttämättä ole kokeneita pelikehittäjiä, mikä sopii perusteeksi myös tämän tutkielman kannalta. Myös Söbke & Streicher (2016) perustelee helppouden tärkeyttä tällä. Koska saavutettavuus on hyvin laaja käsite, jakavat Petridis ym. (2010) sen vielä neljään osaan. Ohjelmiston dokumentointiin ja tukeen, oppimisen helppouteen, lisensointiin ja hintaan. Siis kaikki asiat, jotka helpottavat pelimoottorin oppimista ja uusien käyttäjien saamista tuotteen pariin.

Verkostoitumiseen ja pelin sisäisten yhteisöjen saavuttamiseen peliltä ja siten pelimoottorilta vaaditaan jonkinlainen verkkokomponentti. Vaikka suuret yhteisöt ja sosiaaliset elementit ovatkin koko ajan kasvavassa suosiossa nykyajan peleissä (Petridis ym. 2010), on ehkä turha antaa liikaa painoarvoa tällaisille peleille aloittelevien indie-kehittäjien näkökulmasta. Verkkomoninpelimahdollisuus tuo lisävaikeuksia

pelin kehitykseen ja voi johtaa lisäkustannuksiin pelin julkaisunkin jälkeen esimerkiksi tarvittavien palvelimien takia. Steam-kaupan haulla näkee, että alle kymmenesosassa siellä olevista indie-peleistä on verkkomoninpeli (Steam 2017).

Heterogeenisyydellä tarkoitetaan tässä tapauksessa esimerkiksi monialustatukea. Sen tärkeyttä perustellaan sillä, että hyötypelien kohdeyleisö vaihtelee paljon eikä yleensä koostu kokeneista pelaajista, ja siksi peli pitäisi pystyä julkaisemaan monelle eri alustalle. Sama ei päde indie-kehittäjien näkökulmasta, mutta monialustatukea voi silti pitää tärkeänä myös tämän tutkielman näkökulmasta. Indie-pelien graafisten ja muiden rajoitusten vuoksi on ehkä helpompaa kehittää menestyvä peli esimerkiksi mobiilialustoille kuin tietokoneelle tai suosituimmille pelikonsoleille.

Viimeiseksi Petridis ym. (2010) mainitsevat vielä koostettavuuden, eli miten pelimoottoriin pysty tuomaan materiaalia, esimerkiksi 3D-malleja tai lisäkirjastoja muista lähteistä. Tämä on yleisesti ottaen olennainen ominaisuus pelimoottorilta, koska esimerkiksi juuri 3D-mallinnukseen on harvoin samalle tasolle pääseviä työkaluja pelimoottorin sisälle rakennettuna kuin mitä erillisestä siihen tarkoitettu ohjelmistosta löytyy. Toisaalta aloittelijat voivat pelimoottorin käyttöä opitellessaan tulla ehkä toimeen myös alkeellisimmilla moottorin sisäisillä työkaluillakin.

Patrasitidecha (2014) esittelee pro gradussaan taulukon, jonka avulla hän vertailee pelimoottoreita mobiilikehityksessä. Ensimmäisenä kriteerinä hän esittää valmiit resurssit ja ohjelmointikirjastot samoin perustein kuin Gregory (2009). Laajat ohjelmointikirjastot nopeuttavat kehitystä, koska kehittäjien ei tarvitse tietää miten kaikki asiat matalla tasolla toimivat, vaan he voivat käyttää pelimoottorin valmiita kirjastoja. Valmiita resursseja voivat olla esimerkiksi UDK:n mukana tuleva FPS (First Person Shooter) peliprojekti, jonka pohjalta kehittäjä voi helposti ja nopeasti tehdä FPS-pelin (Patrasitidecha 2014). Ajan säästäminen ja kehityksen helppous ovat tärkeimpiä asioita aloittelevien indie-kehittäjien kannalta. Taulukossaan Patrasitidecha vertailee 3D-, fysiikka-, tekoäly-, verkko-, partikkeli- ja äänikirjastoja. Kleinschmidt & Haag (2016) sanovat etenkin fysiikkasimuloinnin, törmäyksen tunnistuksen ja äänen tukemisen olevan tärkeitä ominaisuuksia. Nämä sisältyykin käytännössä jokaiseen pelimoottoriin.

Toiseksi kriteeriksi Patrasitidecha (2014) esittää pelimoottorin käytettävyyden. Ohjelmiston käytettävyys koostuu oppimisen helppoudesta ja nopeudesta, käytön tehokkuudesta, muistettavuudesta, virheiden todennäköisyydestä ja siitä, kuinka paljon käyttäjät itse pitävät ohjelmiston käyttämisestä (Nielsen & Loranger 2006). Eriyisesti oppimisen helppous on tämän tutkielman kannalta tärkeä asia. Patrasitidecha jakaa sen vielä viiteen osaan, joista jotkin ovat hyvin samankaltaisia kuin Petridiksen ym. (2010) artikkelissa. Nämä osat ovat tutoriaalit, esimerkit, yhteisöt, ohjelmointikieli sekä tekninen tuki. Vaikka näissä kahdessa tutkimuksessa jako onkin tehty hieman eri tavoilla, voidaan niistä yhdistää aloittelevan indie-kehittäjän kannalta yhdeksi tärkeimmistä ominaisuuksista pelimoottorin oppimisen helppous, joka koostuu dokumentaatiosta, tuesta, tutoriaaleista, yhteisöstä sekä skriptauskielistä. Tässä käytetään ohjelmointikielen sijasta termiä skriptauskieli erottamaan kielet, jolla pelimoottori itse on ohjelmoitu ja kielet, joilla kehittäjä kirjoittaa peliin vaikuttavat skriptit. Myös Rocha, Rocha & Araújo (2010) esittävät pelimoottoriin liittyvän yhteisön tärkeänä tekijänä ja käyttää yhteisön foorumin käyttäjämäärää suorana arviointikriteerinä.

Käytettävyyden toiseen osaan eli käytön tehokkuuteen Patrasitidecha (2014) sisällyttää pelimoottorin työkalut kuten kenttä-, animaatio- ja skriptieditorin. Kaikkia pelimoottorin resursseja pitäisi pystyä muokkaan työkaluilla niin, että kehittäjä voi nähdä muutokset suoraan pelimoottorissa. Näin vähennetään esimerkiksi tarvetta tuoda 3D-malleja projektiin useita kertoja, ja siten taas nopeutetaan kehitystä. Patrasitidecha ei kuitenkaan arvioi työkaluja muuten kuin kyllä/ei-tasolla eli sisältääkö pelimoottori kyseisen työkalun vai ei. Työkalun hyvyteen ei oteta kantaa hänen esittämässään taulukossa. Samantyyppistä jakoa esittävät Rocha, Rocha & Araújo (2010). He arvioivat pelimoottorin ominaisuuksia, jotka on jaettu graafisiin ja ei-graafisiin ja joissa on pitkälti samoja kriteereitä kuin Patrasitidechan esittämässä taulukossa. Myös Rocha, Rocha & Araújo (2010) tarkastelevat näitä kyllä/ei-tasolla.

Työkaluihin Patrasitidecha (2014) laskee myös pelimoottorin tukemat 3D-mallinnuksen työkalut. Tämä liittyy aiemmin mainittuun koostettavuuteen (Petridis ym. 2010) ja onkin vain yksi osa-alue siitä. Se on kuitenkin kiinnostavin osa-alue arvioida, koska



olisi turhaa tarkastella esimerkiksi ulkopuolisilla työkaluilla tehtyjen kuva- tai äänitiedostojen tukea pelimoottoreissa, sillä sitä voidaan pitää itsestäänselvänä. Se, mitä 3D-työkaluja pelimoottorin pitäisi tukea riippuu taas käyttäjän mieltymyksistä. Patrasitidecha on tarkastellut MAYA-, 3D Studio Max (nykyään 3ds Max)-, Blender- ja Softimage-ohjelmistojen tukea. Softimagea ei enää kehitetä eikä siihen tarjota tukipalveluja, joten se voidaan jättää ulkopuolelle arvioinnista.

Petridis ym. (2010) sisällyttivät hinnan ja lisensoinnin saatavuuteen, mutta hinnan voi pitää omana kriteerinään, kuten Patrasitidecha (2014) esittää. Lisäksi sen voisi vielä jakaa kahteen osaan, eli tuotteen varsinaiseen ostohintaan ja tekijänpalkkioon eli osinkoon, jonka pelimoottorin kehittäjä ottaa sillä tehtyjen pelien voitoista. Ostohinta on tutkielman näkökulmasta todella suuri tekijä. Indie-kehittäjä, joka ei tiedä, tuleeko ikinä pelistään mitään ansaitsemaan ei varmastikaan halua maksaa pelimoottorista ja koska ilmaisia ohjelmistoja on saatavilla suuri määrä, on vaikea saada aloittelijoita käyttämään tuotetta, jos se on maksullinen. Toisaalta maksullisistakin ohjelmistoista on usein saatavilla jonkinlainen ilmainen kokeiluversio, jonka avulla voidaan yrittää houkutella uusia käyttäjiä. Myös Kleinschmidt & Haag (2016) painottavat ohjelmiston hinnan tärkeyttä hyötypelien kehitystä tarkastelevassa artikkelissaan.

Monet ilmaisista pelimoottoreista, esimerkiksi Unreal Engine, ottavat tuotteillaan tehdyistä peleistä tekijänpalkkiota. Tämä ei kuitenkaan ole aloittelevalla kehittäjälle yhtä tärkeä asia kuin varsinainen ostohinta, koska ensimmäisistä pelikehityksen kokeiluista on vielä pitkä matka julkaistuu peliin, jolla kehittäjä voi ansaita rahaa-kin.

Kuten Petridis ym. (2010), myös Patrasitidecha (2014) esittää mahdollisia julkaisualustoja arvointikriteeriksi. Hänen tutkimuksensa keskittyi mobiilikehitykseen, joten alustavaatimukset painottuvat hänellä mobiilipuolelle. Petridis taas mainitsi yleensä ottaen monialustatuen tärkeyden, tosin perusteista, jotka eivät vahvasti päde tämän tutkielman näkökulmasta. Myös monet muut, kuten Tukiainen (2013) ovat pitäneet useita julkaisualustoja tärkeänä ominaisuutena.

Julkaisualustan lisäksi Patrasitidecha mainitsee myös kehitysalustan ja tarkastelee tukevatko moottorit kehitystä Windows-, OS X- ja Linux-käyttöjärjestelmillä. Myös Westhoven & Alexander (2015) esittävät järjestelmävaatimukset eräänä kriteerinä. Etenkin indie-kehittäjien kannalta tämä tarkoittaa pitkälti nimenomaan kehitysalustoja, koska laitteistopuoli harvoin rajoittaa pelimoottorin käyttöä nykypöydillä, varsinkaan pienemmän mittakaavan indie-peleissä.

Tässä tutkielmassa tarkasteltaviksi pelimoottorin ominaisuuksiksi voidaan siis koota näillä aineistoilla ja perusteilla neljä eri kohtaa. Nämä kriteerit toistuvat useissa lähteissä ja ne ovat myös olennaisia tämän tutkielman näkökulman kannalta. Arvioitavat kriteerit ovat ohjelmiston hinta, oppimisen helppous, työkalujen ja valmiiden kirjastojen monipuolisuus sekä julkaisualustojen ja kehitysalustojen tuki. Ohjelmiston hinta jaetaan vielä ostohintaan ja tekijänpalkkioon ja oppimisen helppous dokumentaatioon, tukeen, tutoriaaleihin, yhteisöihin ja skriptauskieliin. Työkalujen monipuolisuuteen kuuluvat myös tuetut 3D-mallinnuksen työkalut. Näistä kriteereistä aloittelevan indie-kehittäjän näkökulmasta tärkeimpinä pidetään ostohintaa sekä oppimisen helppoutta. Myös muut kriteerit ovat tärkeitä tutkielman näkökulmasta, mutta ne ovat lisäksi yleisesti ottaen oleellisia ominaisuuksia kaikille kehittäjille.

Valittuja kriteereitä tukee vielä esimerkiksi Westhoven & Alexanderin (2015) artikkelissaan pelimoottoreiden arvointiin esittämä taulukko, jossa mainitaan kaikki näistä kriteereistä. Westhoven & Alexander (2015) myös huomioivat lähdekoodin saatavuuden, eli onko pelimoottori avoimen lähdekoodin ohjelmisto. Tämä ei ole käytännön kannalta kovin tärkeää aloitteleville indie-kehittäjille, koska täysin avoin lähdekoodi olisi hyödyksi lähinnä, jos kehittäjä haluaisi muokata pelimoottorin toteutusta matalalla tasolla. Unity myös tarjoaa kehittäjille pääsyn matalankin tason grafiikkarajapintoihin vaikkei kaikki koodi olekaan avointa (Unity 2017e). Toisaalta monet suosivat avoimen lähdekoodin ohjelmistoja periaatteenkin vuoksi.

## 4 Unityn arviointi

Tässä luvussa arvioidaan Unitya aiemmin päätetyillä kriteereillä niin, että jokaisessa alaluvussa tarkastellaan yhtä kriteeriä. Arvioinnissa käytetään apuna aiemmissa tutkimuksissa saatuja tuloksia ja Unitysta esitettyjä arvioita. Unitya myös verrataan lyhyesti joihinkin muihin pelimoottoreihin.

### 4.1 Pelimoottorin vaatimukset

Unity täyttää kaikki aiemmin todetut pelimoottorin vaatimukset. Siinä on järjestelmä, joka käsittelee käyttäjän antamia syötteitä, piirtomoottori, joka pystyy piirtämään kappaleita ja tekstiä sekä äänien käytön mahdollistava äänisysteemi. Nopeuden ja vakauden arviointi on vaikeampaa, mutta käytetyssä lähdemateriaalissa ei esiintynyt mainintoja Unityn epävakaudesta tai hitaudesta, joten ei ole syytä pitää niitä ongelmana.

### 4.2 Ohjelmiston hinta

Luvussa kolme todettiin, että ohjelmiston hinta on aloittevalle indie-kehittäjälle tärkeimpiä kriteereitä pelimoottoria valittaessa. Unityn perusversio on ollut vuodesta 2009 saakka kaikille ilmaiseksi saatavilla. Unity tarjoaa myös maksullista Pro-versiota, jonka hinta on 125 euroa kuukaudessa ja mikäli yrityksen vuosituotto ylittää 100 000 euroa vuodessa, vaatii Unity päivittämistä joko Pro-versioon tai 35 euroa kuukaudessa maksavaan Plus-versioon. Plus-versio on halvempi muoto Pron ja ilmaisen välissä, joka sisältää osan Pron tarjoamista ominaisuuksista ja jolla saa ansaita 200 000 euroa vuodessa. (Unity 2017c). Aloittelevan indie-kehittäjän kannalta nämä summat ovat varmasti niin suuria, että tarvetta maksullisille versioille ei ainakaan sen takia ole (Pluralsight 2015).

Pro-versio sisältää myös joitakin ominaisuuksia joita ilmaisessa versiossa ei ole, mutta näistäkin suurin osa liittyy asioihin, jotka eivät ole tämän tutkielman nä-

kökulmasta kovin oleellisia. Pro-versio tarjoaa muun muassa parempia työkaluja grafiikka- ja äänipuolelle, joilla pelit saadaan entistä realistisemmaksi. Mutta kuten mainittu, indie-peleissä harvoin tavoitellaan äärimmäistä realistisuutta. Toinen merkittävä osa-alue, johon Pro tarjoaa ominaisuuksia, on pelin optimointi. Se antaa käyttäjälle profilointityökaluja, joiden avulla on helpompi selvittää, mitkä pelin osat vaativat eniten suorituskykyä ja näin optimoida peli toimivammaksi. Pro-versio myös pienentää ohjelmiston kokoa poistamalla pelimoottorin osat, joita peli ei käytä. Nämäkin ovat tärkeämpiä laajoihin peliprojekteihin, jotka vaativat paljon suorituskykyä eikä niinkään indieprojekteihin. Pro-versiolla pystyy lisäksi poistamaan Unityn vesileiman, joka näkyy ilmaisversiolla tehtyjä pelejä käynnistettäessä. (Unity 2017d).

Ostohinnan lisäksi otetaan huomioon tekijänpalkkio ja muut lisäkustannukset, joita moottorin käyttämisestä voi tulla, vaikkeivat ne tutkielman kannalta olekaan yhtä tärkeitä kuin varsinainen ostohinta. Unity ei peri minkäänlaista osinkoa sillä tehtyjen pelien tuotoista, eikä vaadi maksua pelien julkaisusta. Kaikki Unitylla tehty sisältö on kehittäjän itsensä omistamaa. (Unity 2017f).

Voidaan siis sanoa, että Unity täyttää hintaan liittyvät vaatimukset aloittelevan indiekehittäjän kannalta varsin hyvin. Maksullisissa versioissa on joitain ominaisuuksia, jotka voisivat olla hyödyllisiä myös indie-kehittäjille, mutta suurin osa pystyy tekemään kaiken tarvittavan ilmaisversiollakin.

### **4.3 Oppimisen helppous**

Ohjelmiston oppimisen helppouden huomattiin olevan toinen erityisen tärkeä kriteeri tämän tutkielman näkökulmasta. Se jaetaan dokumentaatioon, tutoriaaleihin, tukeen, yhteisöön sekä skriptauskieliin. Unitya pidetään suhteellisen helppokäyttöisenä pelimoottorina ja myös Smíd (2017) sanoo Unityn tärkeimmäksi ominaisuudeksi sen helppouden.

Unity tarjoaa laajan verkkosivuillaan dokumentaation skriptaukseen Unityssa sekä manuaalin, joka kertoo moottorin ja editorin ominaisuuksista. Skriptausdoku-

mentaatioissa kuvaillaan eri funktioiden käyttötarkoitukset ja joidenkin funktioiden käytöstä on annettu myös esimerkkejä. Kleinschmidt & Haag (2016) mainitsevat artikkelissaan, että Unity tarjoaa useita dokumentaatioita ja tutoriaaleja, mutta sanovat myös eri näkymien ja funktioiden ymmärtämisen vaatineen aikaa, koska dokumentaatio ei ollut täysin selkeää. Patrasitidecha (2014) taas arvioi pro gradussaan moottoreiden tutoriaaleja ja esimerkkejä asteikolla ja antaa Unitylle arvosanan 5 molemmissa kategorioissa. Myös Murphy (2016) kehuu arvostelussaan Unityn tutoriaaleja ja sanoo, että niiden avulla kokematonkin kehittäjä saa ensimmäisen pelinsä valmiiksi ja jaettavaksi muutamassa tunnissa. Unityn dokumentaation on todettu myös auttavan hyvin muun muassa hyödyllisyyden tunnistettavuudessa (Tukiainen 2013).

Pelimoottorin kehittäjien lisäksi tutoriaaleja ja esimerkkejä voivat tehdä myös tavalliset käyttäjät. Siksi pelimoottoriin liittyvä yhteisö on tärkeä tekijä oppimisen helpottamisessa. (Patrasitidecha 2014). Unitylla on suuri yhteisö ja foorumi on täynnä vastauksia, jotka auttavat käyttäjiä ongelmiansa ratkaisussa (Smíd 2017). Kuten tutoriaaleille ja esimerkeille, Patrasitidecha antaa myös Unityn yhteisölle arvosanan 5 ja sanoo, että sillä on hänen vertailemistaan pelimoottoreista parhaat yhteisöt. Myös Unityn kauppapaikasta, jossa yhteisön jäsenet voivat jakaa omia tuotoksiaan muille kehittäjille ilmaiseksi tai rahaa vastaa, on suuri apu kaikille kehittäjille (Patrasitidecha 2014; Smíd 2017). Unityn yhteisöä pidetään melko yksimielisesti yhtenä parhaista pelimoottoreiden yhteisöistä ja myös yhtenä tärkeimmistä Unityn ominaisuuksista.

Unityn verkkosivuilta löytyy asiakaspalvelu, mutta erillistä teknistä tukea ei ole ja se on käytännössä kokonaan foorumeiden varassa. Huomattavaa on myös, että Premium-käyttäjille on kuitenkin oma tukensa, jossa Unityn omat asiantuntijat auttavat käyttäjiä. Tämä on kuitenkin maksullista eikä siksi voida katsoa olevan hyödyksi tutkielman näkökannalta.

Skriptauskielen vaikutus oppimisen helppouteen riippuu vahvasti käyttäjän omista mieltymyksistä ja aiemmasta kokemuksesta eikä ole välttämättä mahdollista sanoa, mikä tai mitkä kielet ovat parhaita aloittevalle indie-kehittäjälle. Monipuolista

skriptauskielten tukea voidaan kuitenkin pitää hyvänä asiana. Unity tukee virallisesti vain C#:a, mutta myös UnityScriptin käyttö on vielä mahdollista. Käytännössä uuden käyttäjän kannalta ainut vaihtoehto on kuitenkin käyttää C#:a, sillä vielä UnityScriptin tuen aikaankin yli 85% Unityn projekteista käyttivät C#:a (Fine 2017) ja siksi myös suurin osa tutoriaaleista ja yhteisön vastauksista on kirjoitettu C#-käyttäjät mielessä. Smíd (2017) sanoo C#-skriptauksen olevan nopeaa ja tehokasta. Se onko C#-skriptaus helpompaa oppia kuin vaikkapa C++, jota iso osa muista pelimoottoreista käyttää on kehittäjästäkin kiinni. C#:a pidetään kuitenkin korkeamman tason kielenä myös helpompana oppia. Fourment & Gillings (2008) mainitsee muun muassa C:n ja C++:n muistinhallinnan ja osoittimien käytön hankalana aloittelijoille.

Näin tullaan tulokseen, että Unity suoriutuu myös oppimisen helppouden arvioinnissa hyvin, mikä on varmasti yksi suurimmista syistä, miksi se on niin suosittu aloittelijoiden keskuudessa.

#### **4.4 Työkalujen ja kirjastojen monipuolisuus**

Työkalujen ja kirjastojen monipuolisuus ei ole erityisen tärkeää nimenomaan aloitteleven indie-kehittäjän kannalta, mutta se tuli esille lähes kaikessa aineistossa ja se on yleisesti ottaen niin olennainen pelimoottorin ominaisuus, että se pitää ottaa huomioon arvioinnissa. Patrasitidecha (2014) tarkastelee yli 20:ta eri työkalua ja kirjastoa ja huomaa Unityssa puutteita vain kahdessa. Hän sanoo, ettei Unityssa ole ohjelmointikirjastoa tekoälyyn liittyen. Unityn PC-versiossa on kuitenkin jo pitkään ollut hänen kuvailemansa navigaatiosysteemi, jolla voidaan asettaa esimerkiksi liikkumisreittejä tietokoneen ohjaamille hahmoille (Unity 2017g; Smíd 2017). Toisena puutteena hän esittää Unityn UI-editorin. Unity toi kuitenkin marraskuussa 2014 uuden UI-systeemin version 4.6 mukana. Nykyään Unity siis täyttää kaikki Patrasitidechan vaatimukset työkaluista ja kirjastoista.

Unitystä myös löytyy kaikki työkalut, jotka Smíd (2017) esittää. Hän kuitenkin mainitsee, että Unity on graafisilta ominaisuuksiltaan huonompi kuin Unreal Engine

eikä se tue yhtä hyvin esimerkiksi lehvistön ja maaton grafiikoita. Monet muutkin lähteet mainitsevat Unityn olevan graafisesti muita moottoreita kuten Unreal Engineä ja CryEngineä jäljessä (Hannuksela 2016; Pluralsight 2015).

Unity tukee myös useita ulkopuolisia työkaluja, joista etenkin 3D-mallinnuksen työkalut ovat tärkeitä. Patrasitidecha (2014) arvioi 3D-työkalujen tukea neljällä eri ohjelmistolla, joista kolmea kehitetään vielä nykyään. Unity tukee näistä kaikkia. Eriytyisen tärkeäksi voi nostaa Blender-tuen, sillä se on myös ilmainen ja indie-kehittäjien suosiossa oleva ohjelmisto, kun taas 3ds Max ja Maya ovat maksullisia. Näiden kolmen lisäksi Unityyn voi tuoda tiedostoja Cinema4D-, Modo-, Lightwave- ja Cheetah3D-ohjelmistoista. Myös yleismuotoisia kuten .fbx- ja .obj-tiedostoja voi käsitellä Unityssä. (Unity 2017g).

Unity pärjää hyvin työkalujen ja kirjastojen monipuolisuuden tarkastelussa, mutta sen graafisissa ominaisuuksissa on vielä parantamisen varaa.

## 4.5 Julkaisu- ja kehitysalustat

Unity mainostaa itseään monialustaisena pelimoottorina ja sitä pidetään yhtenä sen tavaramerkeistä (Pluralsight 2015). Unity tukee tällä hetkellä 29 eri julkaisualustaa (Unity 2017a) ja pelien julkaisu eri alustoille on onnistuttu tekemään melko helpoksi (Smíd 2017).

Patrasitidecha (2014) arvioi pelimoottoreita mobiilikehityksessä ja siksi hänen esittämät alustavaatimukset painottuvat mobiilipuolelle. Unity ei tue useita näistä ja moni muu pelimoottori suoriutuikin paremmin monialustatuen tarkastelussa. Tosin tarkastelluista käyttöjärjestelmistä kaksi, Symbian ja Bada, ovat vanhentuneita eikä niitä enää kehitetä. Windows-puhelimille kehitystä Unity taas tukee jo nykyään. Unity ei kuitenkaan edelleenkään tue julkaisua WebOS-käyttöjärjestelmälle. WebOS on alun perin kännyköille suunniteltu käyttöjärjestelmä, jota nykyään käytetään lähinnä älytelevisioissa, joten sen tukemattomuus ei ole välttämättä suuri puute. Myös monissa muissa tutkimuksissa jotkin pelimoottorit ovat suoriutuneet alustavertailussa Unityä paremmin. Shiva3D (Tukiainen 2013) ja Marmalade (Kleinsch-

midt & Haag 2016) ovat todettu paremmaksi monialustatuen kannalta. Tukiaisen tutkimus on tosin vanhentunut, sillä sen mukaan Unity tukee vain kahdeksaa eri alustaa.

Kehitysalustana Unity tukee Windowsia, OS X:ää ja Linuxia. Nämä ovat yleisimmät käyttöjärjestelmät ja yhteensä yli 95%:ssa tietokoneista on käytössä jokin näistä kolmesta (Statcounter 2017). Myös Patrasitidecha (2014) tarkastelee näiden kolmen käyttöjärjestelmän tukea, tosin hänen tutkimuksessaan arvioima Unityn versio ei vielä tukenut Linuxia.

Kehitysalustojen kannalta Unity täyttää vaatimukset hyvin. Myös julkaisualustojen tuki on laaja, mutta on myös käytössä olevia alustoja, joita Unity ei vielä tue.



## 5 Yhteenveto

Tutkielmassa selitettiin pelimoottorin käsite ja käytiin lävitse pelimoottorin vaadittavat ominaisuudet sekä muut yleiset osat. Pelimoottorilta vaaditaan ainakin tapa käsitellä käyttäjän syötteitä, piirtomoottori sekä äänisysteemi. Siltä myös vaaditaan nopeutta ja vakautta. Muita yleisiä komponentteja ovat fysiikkamoottori, skriptirajapinta, tekoäly, verkkokomponentti sekä animaattori.

Tutkielmassa saatiin koottua lista sen näkökulman kannalta tärkeimmiksi pelimoottorin arvointikriteereiksi nostettavia asioita. Ne ovat ohjelmiston hinta, oppimisen helppous, työkalujen ja kirjastojen monipuolisuus sekä julkaisu- ja kehitysalustat. Oppimisen helppoutta arvioidessa tarkastellaan pelimoottorin dokumentaatiota, tukea, tutoriaaleja, yhteisöjä sekä skriptauskieliä. Hinta jaetaan ohjelmiston ostohintaan ja tekijänpalkkioihin ja työkaluihin lasketaan myös pelimoottorin tukemat ulkopuoliset 3D-mallinnuksen työkalut. Näistä erityisen tärkeitä aloittelevalle indie-kehittäjälle ovat ohjelmiston hinta ja oppimisen helppous.

Kriteereiden avulla arvioitiin Unitya ja selvitettiin syitä sen asemaan aloittelevien indie-kehittäjien pelimoottorina. Erityisen hyväksi huomattiin Unityn yhteisö ja osittain myös sen ansiosta tutoriaalit. Myös Unityn työkalut ja kirjastot todettiin monipuolisiksi vaikka graafisissa ominaisuuksissa onkin parannettavaa. Kehitysalustojen suhteen on Unity on korkealla tasolla ja myös julkaisualustojen valikoima on suuri. On kuitenkin tutkimuksia, joissa Unity on suoriutunut monia muita pelimoottoreita huonommin alustavertailussa. Hinnan arvioinnissa Unity pärjäsi hyvin ilmaisversionsa ansiosta, vaikka maksullisessa Pro-versiossa on myös joillekin indie-kehittäjille hyödyllisiä ominaisuuksia. Lisäksi Unity ei veloita tekijänpalkkioita.

Voidaan todeta, että Unity on suosittu aloittelevien indie-kehittäjien pelimoottorina etenkin sen laajan yhteisön ja helpon opittavuuden vuoksi. Myös kattava ilmainen versio on välttämätön syy Unityn suosiolle. Sen työkalut, kirjastot ja julkaisualustat ovat riittävät aloittelevan indie-kehittäjän kannalta. Tutkielmassa saatuja tuloksia

voi käyttää apuna pelimoottorin valitsemisessa ja toisaalta pelimoottoreiden kehittäjät voivat huomioida nämä ominaisuudet, jos tavoitteena on hyvä asema aloittelevien indie-kehittäjien keskuudessa.

## Kirjallisuutta

- Bishop, L., Eberly, D., Whitted, T., Finch, M. & Shantz, M. 1998. *Designing a PC game engine*. IEEE Computer Graphics and Applications 18:1, s. 46–53.
- Entertainment Software Association. 2017. *Essential facts about the computer and video game industry*. Saatavilla WWW-muodossa <URL:<http://www.theesa.com/about-esa/essential-facts-computer-video-game-industry/>>. Viitattu 18.12.2017.
- Fourment, M. & Gillings, M.R. (2008). *A comparison of common programming languages used in bioinformatics*. BMC Bioinformatics 9:82.
- Fu, D., Jensen, R. & Hinkelman, E. 2008. *Evaluating Game Technologies for Training*. Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, Montana. IEEE. s.1–10.
- Gregory, J. 2009. *Game Engine Architecture*. Boca Raton: A K Peters/CRC Press
- Haas, J. 2014. *A History of the Unity Game Engine* (opinnäytetyö, Worcester Polytechnic Institute). Saatavilla WWW-muodossa <URL:<https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/>>. Viitattu 18.12.2017.
- Hannuksela, S. 2016. *Unity 5- ja Unreal Engine 4-pelimoottorien vertailu* (opinnäytetyö, Kajaanin ammattikorkeakoulu). <http://urn.fi/URN:NBN:fi:amk-2016112517421>.
- Kallio-Kujala, J. 2016. *Pelimoottori opetuskäyttöön* (opinnäytetyö, Kajaanin ammattikorkeakoulu). <http://urn.fi/URN:NBN:fi:amk-2016120419018>.
- Kleinschmidt, C. & Haag, M. 2016. *Evaluation of Game Engines for Cross-Platform Development of Mobile Serious Games for Health*. Studies in Health Technology and Informatics 223, s. 207–214.
- Marks, S., Windsor J. & Wünsche, B. 2007. *Evaluation of game engines for simulated surgical training*. Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia. New York: ACM. s.273–280.
- Nielsen, J. & Loranger, H. (2006). *Prioritizing Web Usability*. California: New Riders

Publishing Thousand Oaks.

- Oliveira, C.V., Oliveira, F.E. & Pedrini, H. 2016 *Game Engine with 3D Graphics* (pro gradu- tutkielma, University of Campinas). Saatavilla WWW-muodossa <URL:<http://www.ic.unicamp.br/~reltech/PFG/2016/PFG-16-13.pdf>>. Viitattu 18.12.2017.
- Patrasitidecha, A. 2014. *Comparison and evaluation of 3D mobile game engines* (pro gradu -tutkielma, University of Gothenburg). Saatavilla WWW-muodossa <URL:<http://studentarbeten.chalmers.se/publication/193979-comparison-and-evaluation-of-3d-mobile-game-engines>>. Viitattu 18.12.2017.
- Petridis, P., Dunwell, I., de Freitas, S. & Panzoli, D. 2010. *An engine selection methodology for high fidelity serious games*. Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference. IEEE. s. 27–34.
- Ritterfeld, U., Cody, M. & Vorderer, P. 2009. *Serious Games: Mechanisms and Effects*. New York: Routledge.
- Rocha, R.V., Rocha R.V. & Araújo, R.B. 2010. *Selecting the Best Open Source 3D Games Engines*. 2010 Brazilian Symposium on Games and Digital Entertainment SB-Games 2010 : proceedings : 8-10 November 2010 Florianópolis, Santa Catarina, Brazil. California: IEEE. s. 333–336.
- Simon, B. 2013. *Indie Eh? Some kind of Game Studies*. The Journal of the Canadian Game Studies Association 7:11, s.1–7.
- Smíd, A. 2017. *Comparison of Unity and Unreal Engine*. Sudbury: Jones and Bartlett Publishers, Inc.
- Söbke, H. & Streicher, A. 2016. *Serious Games Architectures and Engines*. Entertainment Computing and Serious Games International GI-Dagstuhl Seminar 15283, Dagstuhl Castle, Germany, July 5-10, 2015, Revised Selected Papers. Cham: Springer International Publishing AG. s. 148–173.
- Thorn, A. 2011. *Game Engine Design and Implementation*. Sudbury: Jones and Bartlett Publishers, Inc.
- Tukiainen, T. 2013. *Mobiilipelit ja pelimoottorit* (pro gradu -tutkielma, Helsingin yliopisto). <http://urn.fi/URN:NBN:fi-fe2017112252270>.
- van Best, M. 2011. *Participatory gaming culture: Indie game de-*

- sign as dialogue between player & creator* (pro gradu-tutkielma, Utrecht University). Saatavilla WWW-muodossa <URL: <https://dspace.library.uu.nl/handle/1874/204635>>. Viitattu 18.12.2017.
- Westhoven, M. & Alexander, T. 2015. *Towards a Structured Selection of Game Engines for Virtual Environments*. Virtual, Augmented and Mixed Reality 7th International Conference, VAMR 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings. Cham: Springer International Publishing AG. s.142–152.
- Blizzard Entertainment. 2017. *Hearthstone FAQ*. Saatavilla WWW-muodossa <URL: <https://us.battle.net/hearthstone/en/faq/>>. Viitattu 16.11.2017.
- Devmaster.net. 2017. *Devmaster engines*. Saatavilla WWW-muodossa <URL: <http://devmaster.net/devdb/engines>. Viitattu 16.11.2017
- Fear, E. 2009. *United They Stand*. Saatavilla WWW-muodossa <URL: <http://www.develop-online.net/analysis/united-they-stand/0116643>. Viitattu 16.11.2017
- Fine, R. 2017. *UnityScript's long ride off into the sunset*. Saatavilla WWW-muodossa <URL: <https://blogs.unity3d.com/2017/08/11/unityscripts-long-ride-off-into-the-sunset/>. Viitattu 16.11.2017
- Juhl, E. 2013. *Releasing Unity 4.3*. Saatavilla WWW-muodossa <URL: <https://blogs.unity3d.com/2013/11/12/releasing-unity-4-3/>> Viitattu 16.11.2017
- Kogel, D. 2012. *Siri, define Indie!*. Saatavilla WWW-muodossa <URL: <https://superlevel.de/spiele/siri-define-indie/>> Viitattu 16.11.2017
- Murphy, K. 2016. *Unity Game Engine Review*. Saatavilla WWW-muodossa <URL: <https://www.gamesparks.com/blog/unity-game-engine-review/>>. Viitattu 19.10.2017.
- Pluralsight. 2015. *Unity, Source 2, Unreal Engine 4, or CryENGINE - Which Game Engine Should I Choose?*. Saatavilla WWW-muodossa <URL: <https://www.pluralsight.com/blog/film-games/unity-udk-cryengine->

- game-engine-choose>. Viitattu 19.10.2017.
- Remo, C. 2009. *In-Depth: Unity Launches Free Option, Announces Xbox 360 Support*. Saatavilla WWW-muodossa <URL: In-Depth: Unity Launches Free Option, Announces Xbox 360 Support>. Viitattu 16.11.2017.
- Statcounter 2017. *Statcounter - Desktop Operating System Market Share World Wide*. Saatavilla WWW-muodossa <URL: <http://gs.statcounter.com/os-market-share/desktop/worldwide>>. Viitattu 6.12.2017.
- Steam 2017. *Steam Store*. Saatavilla WWW-muodossa <URL:<http://store.steampowered.com/search/>>. Viitattu 19.10.2017
- Unity. 2017. *Unity Public Relations*. Saatavilla WWW-muodossa <URL:<https://unity3d.com/public-relations>>. Viitattu 19.10.2017.
- Unity. 2017. *Made With Unity*. Saatavilla WWW-muodossa <URL: <https://unity3d.com/showcase/gallery>>. Viitattu 16.11.2017.
- Unity. 2017. *Compare Plans: Plus vs Pro*. Saatavilla WWW-muodossa <URL: <https://store.unity.com/compare-plus-and-pro>>. Viitattu 6.12.2017.
- Unity. 2017. *Unity License Comparisons*. Saatavilla WWW-muodossa <URL: <http://download.unity3d.com/unity/licenses>>. Viitattu 6.12.2017.
- Unity. 2017. *Unity FAQ*. Saatavilla WWW-muodossa <URL: <https://unity3d.com/unity/faq>>. Viitattu 6.12.2017.
- Unity. 2017. *Unity graphics rendering*. Saatavilla WWW-muodossa <URL: <https://unity3d.com/unity/features/graphics-rendering>>. Viitattu 6.12.2017.
- Unity. 2017. *Unity Documentation*. Saatavilla WWW-muodossa <URL: <https://docs.unity3d.com>>. Viitattu 6.12.2017.
- Unreal Engine. 2017. *Unreal Engine Showcase*. Saatavilla WWW-muodossa <URL: <https://www.unrealengine.com/en-US/showcase/>>. Viitattu 5.12.2017.