# Pelimoottorien vertailu ja valinta

Suunnitellessa oppimispeliä ja sen toteuttamista, yksi tärkeimmistä elementeistä on oikean ja pelin toteutukseen sopivan pelimoottorin valitseminen. Pelin suunnitteluvaiheessa on hyvä tehdä vertailua eri pelimoottorien välillä ja käydä läpi niiden rajallisuuden sekä ominaisuudet. Tässä kappaleessa käyn lyhyesti läpi omat vertailukohteet sekä kerron miksi ja mihin pelimoottoriin päädyin.

Monet pelimoottorit tarjoavat useita eri vaihtoehtoja pelien ohjelmoimiselle, joka itsessään tekee vaikeaksi sopivan pelimoottorin löytämisen. Tärkeimpiä ominaisuuksia ja ehtoja, joita opinnäytetyössäni hyödynnettävän pelimoottorin pitää täyttää ovat mm. hyvä dokumentointi, riittävä ohjeistus ja avunsaanti, selaintuki, käyttöönotto ja lisenssit. Näitä lähtökohtia pidin kriteereinä valitessani seitsemää vartailunkohdetta, jotka kaikki ovat omalla tavallaan hyviä pelimoottoreita.

Usemman pelimoottorin takana toimii joko suurempi pelitalo tai pienempi ryhmä, jotka edesauttavat pelimoottorien jatkuvaa kehittämistä ja parantamista. Niiden versiot ovat useimmiten aina ajan tasalla, joka mahdollistaa tietotekniikan nopeassa ja vaihtelevassa virrassa mukana pysymisen.

Dokumentaatio  
Pelimoottorien dokumentaatio on yksi tärkeimmistä kohdista pelimoottorin valinnassa. Monen pelimoottorin dokumentointi vaihtelee suuresti ja niiden laajuus riippuu täysin pelimottoorin käyttötarkoituksista sekä tarjolla olevista ominaisuuksista. Dokumentoinnista sanotaan mm. ettei ”dokumentoinnin tarkoituksena ole listata kaikkia pelimoottorin funktioita vaan tutustuttaa käyttäjä pelimoottorin eri osa-alueisiin ja näin ollen auttaa käyttäjää pääsemään eteenpäin” (Crafty 2013). Harva pelimoottoreista pohjustaa dokumentoinnin pelkästään API –dokumenttiin, jossa kaikki funktiot ovat vain listattuna, sen sijaan että pelimoottorin toimivuuksista ja käytettävyydestä olisi kerrottu vielä erillisessä dokumentissa. Esimerkiksi Pixi.js, Crafty ja CreateJS eivät sisällä erillistä dokumentaatiota pelimoottorin ominaisuuksista vaan dokumentointi rajoittuu API-dokumenttiin kun taas Phaser, CanvasEngine, COCO S2D ja PlayCanvas sisältävät joko käyttäjälle suunnatun ohjekirjan pelimoottoriin ja sen hyödyntämiseen liittyen tai yhdistelmän niin ohjekirjasta kuin API-dokumentista. Tärkeintä kuitenkin pelimottorien kohdalla on että funktionaalinen toiminta on pystytty dokumentoimaan ja selittämään käyttäjälle niin että sen avulla voidaan helposti löytää ja hyödyntää pelimottorin ominaisuuksia ja käyttää niitä ilman komplikaatioita. Käyttäjäystävällisyys ja riittävä dokumentoini takaa pelimoottorin käyttäjäkunnan. On hyvä musitaa että mitä laajempi dokumentointi on, sitä enemmän varsinaisesta pelimoottorista saa irti.

## Aloitusoppaat ja esimerkit

Yksi tärkeä aspekti pelimoottorien hyödyntämisessä ja käyttöönotossa on saatavilla olevat esimerkit ja käyttöoppaat eli tutoriaalit. Nämä liittyvät myös käyttäjäystävällisyyteen ja takaavat pelimoottorin paremman ymmärettävyyden ja kokonaisvaltaisen hyödyntämisen. Vertailunkohteiden kohdalla tärkein esimerkki ja opas on ehdottomasti niin sanottu ”Näin pääset alkuun” –ohje, jossa annettaan moottorin käyttöönotosta yksinkertainen esimerkki. Tämän avulla käyttäjä pääsee heti kiinni pelimoottorin toimivuuteen ja oppii kuinka esimerkiksi moottorin alustus-funktiota kutsutaan ja käytetään. Laajin esimerkkien käyttö ja hyödyntäminen näkyi Phaser:llä, jonka esimerkki-sivustolta löytyy satoja eri toimintaa, ominaisuuksia ja käytettävyyttä kuvaavia esimerkkejä. Pelimoottorien käytettävyys luo kovan kilpailun pelimoottorien välille ja siksi on tärkeää että käyttäjälle selvennetään pelimoottorin ja sen ominaisuuksien käyttömahdollisuuksia. Jokainen pelimoottori toimii eri tavalla, joten jokaisen pelimoottorin kohdalla on selvitettävä sen toimivuus. Hyvänä esimerkkinä laajasta käyttäjän avustamisesta toimii PlayCanvas, joka hyödyntää normaalien esimerkkien ohella Youtubea luoden esimerkkivideoita mitä muut pelimoottorit eivät omilla sivuillaan virallisesti tee. Tämän etuna on että käyttäjä näkee konkreettisesti miten funktioita tai ominaisuuksia käytetään lisäten ainakin omalla kohdalla oppimista. Ei-virallisia video-esimerkkejä löytyy muiltakin pelimoottoreilta runsaasti käyttäjien luomina, mutta niitä ei ole hyödynnetty pelimoottorien virallisilla sivuilla.

## Ominaisuudet

Pelimoottorien ominaisuuksien tarjonta on yleisesti hyvin laaja, sillä harvoin pelimoottoreita luodaan ja kehtitetään vain yhtä pelityyppiä tai tarkoitusta varten. Vertailukohteiden tarjoamat ominaisuudet eivät poikkeaa toisistaan kovinkaan paljon, vaikka osa pelimoottoreista käyttää hyväkseen muita kirjastoja täydentämään omaansa. Pelimoottorien ominaisuuksia vertaillessa on hyvä muistaa omat tarpeet ja vaatimukset peliä varten. Jos kevyt pelimoottori täyttää käyttäjän tarpeet, ei kannata valita raskainta pelimoottoria, vaikka se olisi arvosteluissa paras. Pelimoottori on paras vain silloin, kun itse pystyy hyödyntämään kaikkia sen tarjoamia ominaisuuksia. Pelimoottorin ominaisuuksista tärkeinpänä on moottorien selaintuki, joka on miltein sama kaikilla eli modernit selaimet ovat tuettuina, jos renderöitäessä Canvas ja WebGL toimivat. Vertailukohteista ainoastaan PlayCanvas ja Phaser sisältävät integroidun fysiikkamoottorin ja näistä kahdesta Phaser hyödyntää jopa kolmea fysiikkamoottoria. Muut vertailukohteet eivät sisällä fysiikkamoottoria, mutta tukevat tämän päivän käytetyimpiä ulkopuolisia fysiikkamoottoreita, kuten Box2D, Chipmunk tai PhysicsJS, erinomaisesti. Graafisesti pelimoottorit on suunniteltu pääasiassa kaksiulotteisille peleille joskin PlayCanvas, COCO S2D ja CreateJS tukevat myös kolmiulotteista grafiikkaa. Sprite-grafiikka on laajasti tuettuna kaikilla pelimoottoreilla, joka mahdollistaa nostalgisten grafiikkojen käytön mm. pelihahmojen animaatioissa. Opetuspelissä Sprite-grafiikalla ei suurta roolia ole, mutta sen hyödyntäminen esimerkiksi animoinnissa tuo oman vivahteen pelimaailmaan etenkin kun 80-luvun pikselimäinen peligrafiikka kasvattaa jälleen suosiota. Äänet ovat myös oleellinen osa pelimaailmaa ja suurin osa pelimoottoreista hyödyntää sisäänrakennettuja ääniominaisuuksia, kuten HTML5 audiota tai audiomoottoria. Muutamat vertailunkohteista käyttävät erillistä äänikirjastoa audion toistamiseen kuten esimerkiksi CreateJS ja sen oma audion kirjasto SoundJS sekä CanvasEngine ja sen tukema SoundManager 2 –kirjasto. Oppimispelissä äänimailmaa on hyvä hyödyntää sen luoman mielenkiinnon ja tunnelman vuoksi. Etenkin tunnelmien luonti musiikin avulla peliin saa pelaajan helpommin uppoutumaan ja keskittymään itse pelaamiseen.

## Asennus ja käyttöönotto

Hyvin moni pelimoottori on luotu avoimen lähdekoodin periaatteella. Käytännössä se tarkoittaa käyttäjän mahdollisuutta muokata ohjelman lähdekoodia omiin tarpeisiinsa ja käyttää sitä vapaasti omien tarkoitusten mukaan niin kopioimiseen kuin levitykseen (COSS 2014). Avoimen lähdekoodin määrittelyssä myös lisenssit ovat tärkeässä osassa, siksi kaikki vertailukohteet toimivat MIT-lisenssien alasuudessa, joka tarkoittaa vapaata ohjelmistolisenssiä. Sen käyttäjällä on vapaat oikeudet muokata, kopioida ja käyttää teosta. MIT-lisenssi sallii myös teoksen käytön kaupallisissa suljetun lähdekoodin ohjelmistoissa (Sofokus 2014). Kaikki vertailukohteet ovat avoimena lähdekoodina toteutettuja ja hyödyntävät GitHub – versiohallintaa, josta kuka tahansa saa hakea pelimoottorin lähdekoodit sekä viimeisimmän julkaistun version ja hyödyntää niitä omassa projektissaan. Näin käyttäjät pystyvät vaikuttamaan omaan pelimoottoriinsa ja edesauttamaan niiden kehitystä. GitHub:a hyödynnetään laajasti nykypäivän avoimien lähdekoodien ohjelmistoissa luoden ohjelmien ympärille laajan kehittäjäympäristön, joka auttaa ohjelman parantamisessa sekä vikojen etsinnässä ja korjauksessa. Useampi pelimoottori tarjoaa myös suoraa linkkiä kotisivuilta tarvittavien tiedostojen lataamiseksi. Tämä helpottaa suunnattomasti asennusta ja pelimoottorin kirjaston käyttöönottoa, kun kirjaston viimeisintä versiota ei tarvitse etsiä erikseen. Varsinaisessa käyttöönotossa pelimoottorit hyödyntävät alotusoppaita runsaasti. Monet vertailukohteista ohjeistavat lisäämään JavaScript -tiedostot omaan projektikirjastoon ja tarvittavan lähdeviittauksen html tiedostoon <script> -tagin sisään, jonka jälkeen pelimoottori on jo käyttövalmis. Vertailukohteista PlayCanvas ja COCO S2D sisältävät myös omat työkalut pelimoottorin käyttöönottamiseksi kuten PlayCanvas Designer tai CodeIDE ja CocoStudio, joiden avulla pelimoottorin hyödyntäminen ja pelin teko on helppoa. Käyttöönotossa ja asennuksessa avustavat myös laajat foorumi-yhteisöt. Vertailukohteista ainoastaan CanvasEnginellä ei ole virallisia foorumeita, mutta muut vertailukohteet ylläpitävät yhteisöjä, joista osa on hyvin aktiivisessa käytössä.

## Pelimoottorin valinta

VERTAILU VIELÄ KESKEN

## Viitteet

Crafty 2011-2013. Documentation for Crafty.js. Viitattu 23.10.2014 [www.craftyjs.com](http://www.craftyjs.com) > Documentation.

COSS 2014. Avoin lähdekoodi. Viitattu 23.10.2014 [www.](http://www.wikipedia.org)coss.fi > Tietoa avoimuudesta > Avoin Lähdekoodi.

Sofokus 2000-2014. Avoin lähdekoodi – Tiedätkö mitä rajoituksia se asettaa ohjelmistosi käytölle?. Viitattu 23.10.2014 [www.sofokus.com](http://www.sofokus.com) > Blogi > Avoin lähdekoodi