

Amazon Game Tech -palveluiden hyödyntäminen Ticorporatessa

Joakim Piippo

Sami Verronen

Opinnäytetyö

Toukokuu 2019

Liiketalouden ala

Tietojenkäsittelyn tutkinto-ohjelma

Jyväskylän ammattikorkeakoulu

JAMK University of Applied Sciences

Tekijä(t) Piippo, Joakim Verronen, Sami	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 61	Päivämäärä Toukokuu 2019 Julkaisun kieli Suomi Verkojulkaisulupa myönnetty: x
Työn nimi Amazon Game Tech -palveluiden hyödyntäminen Ticorporatessa		
Tutkinto-ohjelma Tietojenkäsittelyn tutkinto-ohjelma		
Työn ohjaaja(t) Immonen, Jarkko		
Toimeksiantaja(t) Tietojenkäsittelyn tutkinto-ohjelma, Jyväskylän ammattikorkeakoulu		
Tiivistelmä <p>Tutkimuksen tavoitteena oli tutkia Amazon Game Tech -palveluiden soveltuvuutta Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opiskelijoiden peliprojektien käyttöön. Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opiskelijat osallistuvat opintojensa toisena vuonna tuotekehitysprojektiin, jossa opiskelijat tekevät pienryhmissä joko mobiilipelejä tai hyötysovelluksia.</p> <p>Tutkimus toteutettiin kvalitatiivisena tutkimuksena. Tutkimuksessa selvitettiin soveltuvatko Amazon Game Tech-palvelut opiskelijoiden peliprojektien käyttöön, kuinka Game Technin backend-palvelut otetaan käyttöön peliprojektissa, sekä kuinka Game Technin backend-palvelut eroavat muista vastaavista palveluista. Tutkimusosuus koostuu kahdesta osasta. Ensimmäisessä osuudessa tutkittiin palvelun soveltuvuutta. Palvelun soveltuvuudelle asetettiin kriteerit, jotka saatiin selville haastatteluiden pohjalta. Haastattelimme tuotekehitysprojektien toimintaa ohjaavia opettajia, jotta saimme selville, mitä kriteereitä palvelun tulisi täyttää.</p> <p>Toisessa tutkimusosassa on luotu käyttöopas Amazonin GameSparks-palvelulle. Käyttöopasta varten loimme Unityn graafisella editorilla yksinkertaisen peliprojektin, jossa palvelu otettiin vaihe vaiheelta käyttöön ja esiteltiin sen ominaisuuksia. Käyttöoppaan tarkoituksena oli, että opiskelijat voivat hyödyntää opasta omissa peliprojekteissaan. Gameparks-palvelua vertailtiin ennen tätä vielä muihin vastaaviin BaaS-palveluihin, jotta saatiin selville, onko markkinoilla muita samanlaisia palveluita, joita opiskelijat voisivat projekteissaan hyödyntää. Opinnäytetyön toimeksiantajana toimi Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelma.</p>		
Avainsanat (asiasanat) Pilvipalvelut, pelinkehitys, pelimoottori, backend, AWS, GameSparks, Unity		
Muut tiedot (salassa pidettävät liitteet)		

Author(s) Piippo, Joakim Verronen, Sami	Type of publication Bachelor's thesis Number of pages 61	Date May 2019 Language of publication: Finnish Permission for web publication: x
Title of publication The Utilization of Amazon Game Tech-services in Ticorporate		
Degree programme Business Information Technology		
Supervisor(s) Immonen, Jarkko		
Assigned by Degree programme in Business Information Technology, JAMK University of Applied Sciences		
Abstract <p>The goal of this research was to find out if Amazon Game Tech services are suitable to be used in the game development projects of the students in the degree programme Business Information Technology at JAMK. During their second-year studies, the students take part in a development project, where the goal is to work in small groups and create a mobile app or a mobile game. The study was assigned by JAMK's Business Information Technology degree programme.</p> <p>The research was carried out as a qualitative study. The study explored the applicability of Amazon Game Tech services to the student's projects, how Game Tech's backend services are deployed in a game development project, and how Game Tech's backend services differ from other similar services. The research part consists of two parts. The first part examined the suitability of the service. The criteria were set for the suitability of the service and then identified based on interviews. The game development course instructors were interviewed to find out what criteria the service should meet.</p> <p>In the second part of the research, a user guide was written for Amazon's GameSparks BaaS service. For the user guide, a simple game project was created with Unity's graphical editor introducing the service step-by-step and presenting most of its main features. The purpose of the user guide was for students to take advantage of the guide in their own projects. Before writing the user guide, GameSparks was compared to other BaaS-services to see if there exist other kinds of similar services the students could use in their projects.</p>		
Keywords/tags (subjects) cloud services, game development, game engine, backend, AWS, GameSparks, Unity		
Miscellaneous (Confidential information)		

Sisältö

1	Johdanto	6
2	Tutkimusasetelma	7
2.1	Opinnäytetyön tavoitteet ja rajaukset	7
2.2	Tutkimuskysymykset	7
2.3	Tutkimusmenetelmät	7
3	Pelien tekniset komponentit.....	9
3.1	Pelit.....	9
3.2	Pelimoottorit	10
4	Pilvipalvelut yleisesti	11
4.1	Pilvipalveluiden käyttöönottomallit	12
4.2	Pilvipalvelumallit	13
5	Amazon Web Services.....	16
5.1	Amazon Game Tech.....	16
5.2	Game Techin backend -palvelut.....	17
5.3	AWS:n analytiikkapalvelut.....	19
5.4	Game Techin sisällöntuotantotyökalut	20
5.5	AWS:n liiketoimintatyökalut	21
6	Tutkimustulokset ja niiden analysointi.....	21
6.1	Tutkimuksen toteutus	21
6.2	Tarpeen määrittely	23
6.3	Teemahaastattelut ja niiden tulokset	24
6.4	BaaS -palveluiden vertailu	26

7	Johtopäätökset.....	28
8	Pohdinta.....	30
	Lähteet	32
	Liitteet.....	36
	Liite 1. Teemahaastattelun runko	36
	Liite 2. Opas GameSparks-palvelun käyttöönottoon Unityn graafisessa editorissa.	37

Kuviot

Kuvio 1. Kvalitatiivisen tutkimuksen syklinen prosessi.	8
Kuvio 2. Pilvipalvelumallit, palveluiden ominaisuudet, sekä käyttöönottomallit.....	15
Kuvio 3. Amazonin tarjoamat Game Tech-ratkaisut	17
Kuvio 4. Tutkimuksen toteutus	23
Kuvio 5. Tulosten analysoinnista saadut kriteerit.	25
Kuvio 6. Taulukko eri BaaS-palveluiden vertailusta	27
Kuvio 7. Käyttöoppaan alustava sisällysluettelo	28

Käsitteet

Pelimoottori	<p>Pelimoottori on kehys, jonka päälle peli rakennetaan.</p> <p>Tyypillisesti pelimoottori sisältää useita erilaisia sisäänrakennettuja työkaluja, joita pelinkehittäjät voivat hyödyntää pelinkehityksessä, kuten fysiikkamoottorin, animointityökaluja, äänityökaluja sekä tekoälyn.</p>
AWS	<p>AWS on lyhenne Amazon Web Services-palveluille, joihin sisältyy Amazonin tarjoamat lukuisat erilaiset pilvipalvelut.</p>
GameSparks	<p>GameSparks on Amazonin omistama peleille tarkoitettu Backend-as-a-Service pilvipalvelu. Palvelun avulla voidaan rakentaa peleille palvelinpuolen komponentteja.</p>
MAU	<p>Lyhenne sanoista Monthly Active User, eli kuinka monta aktiivista kuukausittaista käyttäjää palvelulla on.</p> <p>Käytetään usein internet-palveluiden käytön mittauksessa.</p>
Pilvipalvelu	<p>Verkon yli käytettävä palvelu kuten sovellus, ohjelmistopalvelu tai sovelluskehitysalusta, johon päästään käsiksi internetin välityksellä.</p>
BaaS	<p>Lyhenne sanoista Backend-as-a-Service, eli pilvipalveluna toimiva backend-palvelu.</p>
Event	<p>GameSparksissa käytettävä toiminto, jolla voidaan määrittää mukautettuja tietorakenteita palvelussa.</p>
Backend	<p>Sovelluksen tai pelin palvelinpuolen logiikka.</p>
Asset	<p>Pelinkehityksessä käytettävä termi, joka kuvaa esimerkiksi peliin tuotavia ääniä, peliobjekteja tai grafiikkaa.</p>

Leaderboard	Tulostaulukko, johon tallennetaan peleissä erilaista tietoa, kuten esimerkiksi pelaajien saamia pisteitä.
Unity	Suosittu pelimoottori ja graafinen editori, jolla voidaan kehittää 2d sekä 3d-pelejä useille alustoille.
Unity Scene	Unityssä käytettäviä ympäristöjä, joihin peli itsessään rakennetaan. Sceneihin voidaan rakentaa esimerkiksi pelin päävalikot ja tasot.
Amazon Game Tech	Amazonin tarjoamat pelinkehitykseen käytettävät työkalut.
API	Ohjelmointirajapinta

1 Johdanto

Pelinkehitys kasvattaa jatkuvasti suosiotaan ja uusia pelinkehityksen työkaluja tulee markkinoille jatkuvasti lisää. Uusilla palveluilla voidaan helpottaa pelinkehittäjien työtaakkaa ja luoda entistä innovatiivisempia pelejä.

Ticorporate on Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opintoihin sisältyvä tuotekehitysprojekti, jossa kehitetään pienryhmissä joko mobiili-pelejä tai hyötysovelluksia. Kurssin tavoitteena on simuloida oikeaa tuotetta kehittävää yritystä, jonka toiminnasta opiskelijat vastaavat itse. Peliprojektit ovat kestoltaan noin kuuden kuukauden mittaisia ja yleensä peliprojekteissa hyödynnetään Unityn graafista editoria. Toimeksiantajana toimii Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma, ja toimeksiantona oli tutkia Amazon Game Tech palveluiden soveltuvuutta Ticorporaten käyttöön. Tarpeena oli löytää helposti ja nopeasti käyttöönotettava backend-ratkaisu, jota opiskelijat voivat hyödyntää peliprojekteissaan. Amazon Game Tech tarjoaa erilaisia pelinkehitykseen tarkoitettuja työkaluja, joista osa on pilven kautta käytettäviä palveluita, analytiikkatyökaluja, sekä erilaisia sisällöntuotantotyökaluja.

Opinnäytetyön teoriaosuudessa käydään läpi yleisesti pelejä, pelien tekniset komponentit, pilvipalvelut yleisesti, sekä Amazonin tarjoamat pilvipalvelut ja tarkemmin vielä Amazon Game Techin tarjoamat palvelut. Tutkimusosuudessa selvitetään haastattelujen pohjalta kriteerit palvelun soveltuvuudelle ja tämän pohjalta tehdään päätös siitä, että soveltuuko palvelu toimeksiantajan tarpeisiin. Mikäli palvelulle asetetut kriteerit täyttyvät, luodaan Game Techin GameSparks backend-palvelun käyttöön otolle kirjallinen opas, jossa käydään läpi palvelun ominaisuudet ja opastetaan sen käyttöönotossa Unity-projektissa. Kirjallista opasta varten luodaan yksinkertainen peliprojekti Unityssä, jotta palvelun käyttöönottoa voidaan paremmin kuvata.

2 Tutkimusasetelma

2.1 Opinnäytetyön tavoitteet ja rajaukset

Tutkimusosuus koostuu kahdesta osasta. Ensimmäisessä osassa on tarkoitus selvittää, täyttääkö palvelu sille ennalta määritellyt kriteerit ja vastaako se toimeksiantajan tarpeita. Kriteerit saadaan selville haastattelemalla Ticorporaten toimintaa ohjaavia opettajia. Mikäli palvelu vastaa toimeksiantajan tarpeita, siirrytään tutkimusosuuden toiseen vaiheeseen. Toisessa vaiheessa teemme yksinkertaisen pelin Unityn graafisella editorilla, jossa kuvailemme palvelun eri toimintoja ja laadimme peliprojektin pohjalta kirjallisen oppaan Amazonin GameSparks-palvelun käyttöönotolle. Opasta on tarkoitus hyödyntää Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelman opiskelijoiden peliprojekteissa. Tutkimukseen kuuluu tämän lisäksi muiden samankaltaisten BaaS-palveluiden vertailu, jotta saadaan selville minkälaisia muita vaihtoehtoja on tarjolla.

2.2 Tutkimuskysymykset

Työssä haetaan vastauksia seuraaviin kysymyksiin, soveltuuko Amazon Game Tech-palvelut Ticorporaten käyttöön, kuinka Game Techin backend-palvelut otetaan käyttöön peliprojektissa, sekä kuinka Game Techin tarjoamat backend-palvelut eroavat muista vastaavista palveluista.

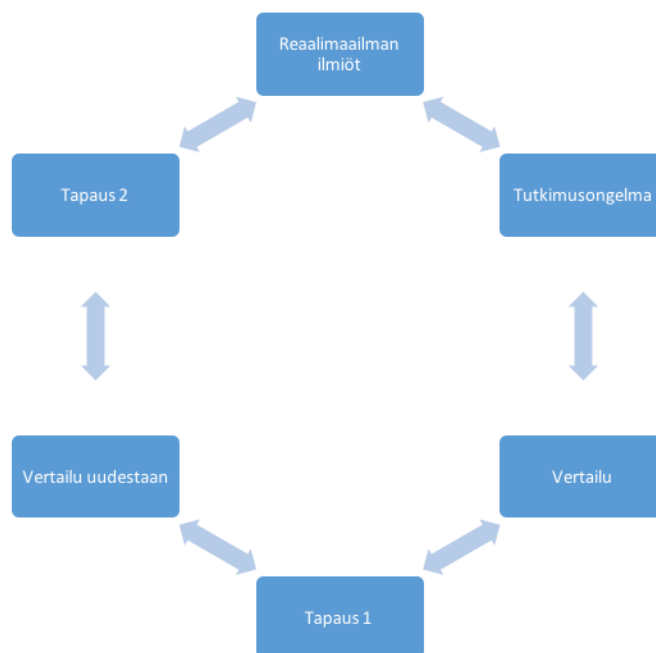
2.3 Tutkimusmenetelmät

Tutkimus toteutetaan kvalitatiivisena, eli laadullisena tutkimuksena. Tämän tutkimuksen tavoitteena on tutkia palvelun soveltuvuutta ennalta määriteltyjen kriteerien perusteella. Valitsimme tutkimusmenetelmän sen takia, että ensin on saatava selville kriteerit, joihin palvelun soveltuvuutta verrataan. Tutkimuksessa käytetään hyväksi

teemahaastatteluita, jotta saadaan selville minkälaisia kriteereitä Game Technin backend-palvelun soveltuvuudelle asetetaan.

Laadullisessa tutkimuksessa on tarkoitus saada aiheesta mahdollisimman syvälinen käsitys, ja on tarkoitus vastata kysymykseen miksi. Laadullisessa tutkimuksessa on suora kontakti tutkimuskohteen ja tutkijan välillä. Käytännön menetelmiä aineistonkeruussa ovat esimerkiksi haastattelut ja havainnointit. (Kananen 2008, 24-25.)

Työn tutkimusosuudessa on käytetty hyödyksi kvalitatiivisiin menetelmiin kuuluvaa teemahaastattelua aineistonkeruussa. Teemahaastattelussa käydään aiheet, eli teemat läpi haastateltavan kanssa. Ennakkovalmisteluiden avulla pyritään käymään koko aihealue perusteellisesti läpi. Teemahaastattelut ovat yleisiä, koska niissä pystytään rajamaan aihealue tarkkaan. (Kananen 2008, 73-74.) Alla kuvattu kvalitatiivisen tutkimuksen syklinen prosessi.



Kuvio 1. Kvalitatiivisen tutkimuksen syklinen prosessi. (Tiedot: Kananen, J 2008. s. 26)

3 Pelit ja pelimoottorit

3.1 Pelit

Videopelaaminen yleistyi 70-luvun loppupuolella peliohjelmoinnin ja pelaamisen kasvatessa suosiotaan. Ensimmäiset pelit laitettiin yleensä levykkeelle tai C-kasetille, ja niitä pelattiin joko kasettiaseman kautta tai levykeaseman kautta. Myöhemmin tulivat erilaiset pelimoduulit ja pelikonsolit. Nykyään suurin osa pelaamisesta tapahtuu älylaitteella, kuten tabletilla tai älypuhelimella. Ennen pelaaminen tapahtui suurimaksi osaksi kotona tietokoneen, television sekä ohjaimen ääressä, mutta nykyään älylaitteiden kehityksen johdosta koko pelikirjasto voi kantaa mukanaan mihin vain. Älylaitteiden kehityksen myötä on syntynyt täysin uusia peligenrejä sekä uusia tapoja pelata pelejä hyödyntäen älylaitteiden kosketusnäyttöjä. Mobiilipelaaminen on muuttanut myös tapoja, joilla pelejä tehdään. Suurin osa mobiilipeleistä ovat yksinkertaisia, itseään toistavia, sekä pelin pituudeltaan lyhyitä. Tämän vuoksi on yleistä, että mobiilipelejä kehittävät tiimit ovat kooltaan pieniä, tai peli voi olla kokonaan yhden henkilön kehittämä. Toinen asia, jolla on ollut suuri merkitys pelaamisen kehittämiseen, on pelien digitaalinen jakelu. Pelejä voidaan ladata suoraan netistä koska tahansa, mistä tahansa. Mobiililaitteilla pelejä voidaan ladata suoraan sovelluskaupasta. Digitaalisen pelaamisen alustoja ovat esimerkiksi Steam ja Origin. Konsoleilla on myös omat kauppansa, kuten esimerkiksi Playstation Store, sekä Nintendo Store. (Rogers 2014, 14-15.)

Pelien suosio voidaan osittain selittää sillä, että ne täyttävät tiettyjä henkilökohtaisia ja yhteiskunnallisia tarpeita, kuten tavoitteellisuutta ja reaaliaikaista kanssakäymistä muiden pelaajien kanssa. Pelaajia miellyttävät erilaiset pelien ja pelaamisen osatekijät ja ominaisuudet. Tämän vaihtelun vuoksi kulttuurilla ja yhteiskunnalla on erityisen suuri merkitys pelien menestyksen kannalta. (Rabin 2010, Luku 2 - Games and Society.)

3.2 Pelimoottorit

Käsite pelimoottori yleistyi 1990-luvulla id Softwaren DOOM-videopelin myötä. Kyseisessä pelissä oli selkeästi erotettu pelin ydinkomponentit toisistaan. 3d-grafiikan renderöintijärjestelmä, äänijärjestelmä, sekä peliobjektien törmäyksen havainnointi (collision detection) oli selkeästi erotettu pelimaailmasta, pelin säännöistä, tai pelin taiteesta ja grafiikoista. Tämän erittelyn ansiosta pelinkehittäjät alkoivat lisensoimaan uusia pelejä käyttäen DOOMia pelimoottorina luoden täysin uudet grafiikat, pelimaailmat, sekä muut komponentit, kuitenkin tehden vain minimaalisia muutoksia pelimoottoriin itsessään. Nykyään on yleistä, että pelimoottoreita lisensoidaan ja suuria osia pelien keskeisistä ohjelmistokomponenteista uudelleen käytetään. Tällä tavalla pelinkehittäjät voivat säästää aikaa ja rahaa, koska kaikkea ei tarvitse tehdä alusta asti. Joskus peliä ja pelimoottoria on vaikea erotella toisistaan, kun taas joissain tapauksissa ne pystytään erottamaan toisistaan suhteellisen helposti. Usein pelimoottorit ovat peligenreistä riippuvaisia, mutta nykyään monet pelimoottorit taipuvat useille eri peligenreille. Tyypillisesti nykyajan pelimoottori koostuu renderöintimoottorista (partikkelit, valaistus ja varjot yms.), fysiikkamoottorista (törmäyksen tunnistus, fysiikat, painovoima), hahmojen animointityökaluista, äänityökaluista, sekä tekoälystä. (Gregory 2009, 11-37.)

Unreal Engine tunnetaan parhaiten sen helppokäyttöisistä ja yhtenäisistä työkaluista, sekä laajasta kirjosta ominaisuuksia Pelimoottoria voidaan käyttää genrestä riippumatta, sekä sillä voidaan luoda pelejä useille eri alustoille. Pelimoottorista on saatavilla laajasti dokumentaatiota ja tutoriaaleja, mutta osa dokumentaatiosta on saatavilla ainoastaan kehittäjille, jotka ovat lisensoituja käyttämään Unreal Engineä. Unreal Engine on tehokas ja yleisesti paljon käytetty pelimoottori. (Gregory 2009, 26.)

Unity on pelimoottori ja graafinen editori, jonka avulla voidaan luoda objekteja, sekä tuoda peliin ulkoisia asetteja myös peliprojektin ulkopuolelta. Unityssä on mm. sisäänrakennettu ohjelmointiympäristö, sisäänrakennetut verkkotoiminnot ja mahdollisuus rakentaa pelejä useille eri alustoille. Unity on yksinkertainen, helppokäyttöinen, intuitiivinen ja mukautettava työympäristö. Unityn käyttö perustason lisenssillä on ilmaista, mutta saatavilla on myös maksullisia Pro- sekä Plus-tason lisenssejä. (Menard 2012. Luku 3 - An Overview of the Unity Engine.)

Unity on yksi tämän hetken suosituimmista pelimoottoreista. Unity on saatavilla sekä Windowsille että MacOS-käyttöjärjestelmille. Unity sisältää laajasti työkaluja mm. pelimaailmojen ja käyttäjäkokemuksen suunnitteluun, sekä kehittäjätyökaluja pelilogiikan suunnitteluun ja ohjelmointiin. Unity tukee sekä 2D- että 3D-pelien kehitystä, sekä pelinkehitystä useille eri genreille. Unityyn on saatavilla mukautettuja laajennuksia, resursseja, sekä työkaluja Unityn omasta Asset Storesta. Asset Storessa käyttäjät voivat esimerkiksi myydä tekemäänsä peligrafiikkaa, 3D-mallinnuksia, työkaluja tai tutoriaaleja muiden käyttäjien ladattavaksi. Unityllä voidaan tehdä pelejä yli 25:lle eri alustalle mukaan lukien konsolit, mobiililaitteet, Windows, MacOS, sekä älytelevisiot. Unityllä voidaan tehdä myös pelejä, jotka hyödyntävät virtuaalitodellisuutta tai lisättyä todellisuutta. Unity tarjoaa myös sisäänrakennettuja työkaluja liikevaihdon tekemiseen peleissä esimerkiksi lisäämällä peliin mainoksia tai pelinsisäisiä ostoksia. (Unity Technologies 2019.)

4 Pilvipalvelut yleisesti

Pilvipalveluiden avulla voidaan hankkia palveluna ulkopuoliselta palveluntarjoajalta mm. laskentatehoa, infrastruktuuria, tallennustilaa, liiketoimintaprosesseja, sovelluksia tai muita palveluita joko yksittäin tai palvelukokonaisuuksina. Pilvipalveluiden yksi suurimmista hyödyistä ovat sen skaalautuvuus ja joustavuus. Mikäli lisää resursseja tarvitaan yllättäen, ne ovat usein heti käytettävissä. Kuluttajan ei myöskään tarvitse investoida esimerkiksi palvelimiin, tai muuhun laitteistoon, vaan kaikki tarvittava saadaan palveluntarjoajalta pilven välityksellä. (Hill, Hirsch, Lake & Moshiri, 2013, Luku 1.1.)

NIST (National Institute of Standards and Technology) määrittelee pilvilaskennan malliksi, joka mahdollistaa verkon välityksellä pääsyn jaettuihin, mukautettaviin laskentaresursseihin, jotka voidaan hankkia ja vapauttaa käyttöön vähäisellä palvelun hallinnalla ja ilman vuorovaikutusta palveluntarjoajan kanssa. Pilvilaskentamalli koostuu viidestä perusominaisuudesta, kolmesta palvelumallista ja neljästä käyttöönottomallista. (Grance & Mell 2011.)

Kaikkia verkon yli käytettäviä palveluita ei automaattisesti määritellä pilvipalveluiksi, vaan palvelun tulee täyttää tietyt kriteerit. Jotta palvelu voidaan määritellä pilvipalveluiksi, kuluttajan tulee pystyä käyttämään sitä verkon yli omalla päätelaitteellaan. Kuluttajan on yksipuolisesti pystyttävä määrittelemään ja jakamaan käytettävissä olevia resursseja ilman vuorovaikutusta palveluntarjoajan kanssa. Palveluntarjoajan laskentaresurssit on jaettu palvelua käyttävien kuluttajien kanssa, käyttäen dynaamisesti sekä virtuaalisia että fyysisiä resursseja pilven kautta. Kuluttajan on pystyttävä vapauttamaan ja varaamaan pilvilaskennan resursseja joustavasti aina halutessaan, sekä pystyttävä kasvattamaan pilven kautta käytettäviä resursseja tarpeen mukaan. (Grance & Mell 2011.)

Pilvipalveluiden suurimmat haasteet kuluttajan näkökulmasta liittyvät lähinnä turvallisuuteen, siirrettävyyteen sekä palveluntarjoajan luotettavuuteen. Yrityskäyttäjien, jotka aikovat ulkoistaa infrastruktuurinsa ulkoiselle palveluntarjoajalle kannattaa kiinnittää huomiota palveluntarjoajan valintaan. Mikäli palveluntarjoajaa joudutaan syystä tai toisesta vaihtamaan, yrityksen palvelua varten kehitettyjä alustoja tai sovelluksia ei välttämättä voida sellaisenaan siirtää toiselle palveluntarjoajalle. Pilvipalveluihin siirtymisestä voi koitua huomattavia parannuksia tietoturvaan varsinkin pienyrityksille ja yksityishenkilöille, joille tietoturva asiat eivät välttämättä ole päivittäisessä toiminnassa yhtä oleellisia kuin suuremmille organisaatioille. Monelle pienyritykselle tai yksityishenkilölle tietoturva-asiat eivät välttämättä ole itsestäänselvyys, eikä niihin kiinnitetä päivittäisessä toiminnassa tarpeeksi huomiota. (Hill, Hirsch, Lake & Moshiri, 2013. Luku 1.13.)

4.1 Pilvipalveluiden käyttöönottomallit

NIST on määritellyt pilvipalveluille 4 eri käyttöönottomallia. Pilvipalveluiden käyttö vaihtelee tarpeen mukaan, ja palvelut ovat räätälöitävissä yrityksen tarpeiden mukaan. Pilvipalveluiden käyttöönottomallit ovat julkinen, yksityinen, hybridi, sekä ns. yhteisöllinen käyttöönottomalli. (Castrillo, Rountree, 2014, Luku 1 - What is the Cloud?)

Julkisessa käyttöönottomallissa kuluttaja on vastuussa ainoastaan loppukäyttäjien pääsystä palveluun. Julkisiin pilvipalveluntarjoajiin yhdistetään internetin kautta yleensä joko suoraan nettiselaimen tai asiakasovelluksen kautta. Julkisessa käyttöönottomallissa palveluntarjoaja on vastuussa pilvipalvelun järjestelmien ja laitteiden hallitsemisessa sekä valvonnassa, kuten esimerkiksi palvelimista, kovalevyistä ja verkkolaitteista. Julkisen pilven suurimmat hyödyt ovat skaalautuvuus, käytettävyys, sekä saatavuus. Suurimmalla osalla julkisen pilven palveluntarjoajista on kaikki tarvittava valmiina, jonka ansiosta palvelu on heti saatavilla. Julkinen pilvi on usein yritykselle kustannustehokkain ratkaisu verrattuna muihin käyttöönottomalleihin. Huono puoli julkisessa pilvipalvelumallissa on esimerkiksi se, että palveluntarjoaja voi päättää huoltokatkoista, jolloin palveluihin saattaa tulla käyttökatkoja. (Castrillo, Rountree, 2014. Luku 3 - Public Clouds.)

Yksityisessä käyttöönottomallissa pilvipalvelun tuottamiseen käytettävät järjestelmät ja resurssit sijaitsevat fyysisesti palvelua käyttävän yrityksen tai organisaation tiloissa. Tässä mallissa järjestelmien ja laitteiden hallinta on myös organisaation vastuulla. Yksityiseen pilveen päästään käsiksi yrityksen lähiverkosta, tai mahdollisesti etäkäytössä suojatun VPN-yhteyden kautta. (Castrillo, Rountree, 2014. Luku 3 - Private Clouds.)

Hybridipilvi on käyttöönottomalli, jossa kaksi tai useampi pilvipalvelumalli ovat linkitettyinä toisiinsa. Hybridipilvi voi tarjota yritykselle tarvittavaa joustavuutta palveluiden suunnittelussa. (Castrillo, Rountree, 2014. Luku 3 - Public Clouds.)

Yhteisöpilvi on käyttöönottomalli, jossa pilvipalvelu on jaettu kahden tai useamman organisaation kesken. Tällä tavoin saadaan enemmän yksityisyyttä kuin julkisessa käyttöönottomallissa, sekä vastuut palvelun ylläpidosta voidaan jakaa organisaatioiden välillä. (Castrillo, Rountree, 2014, Luku 3 - Community Clouds.)

4.2 Pilvipalvelumallit

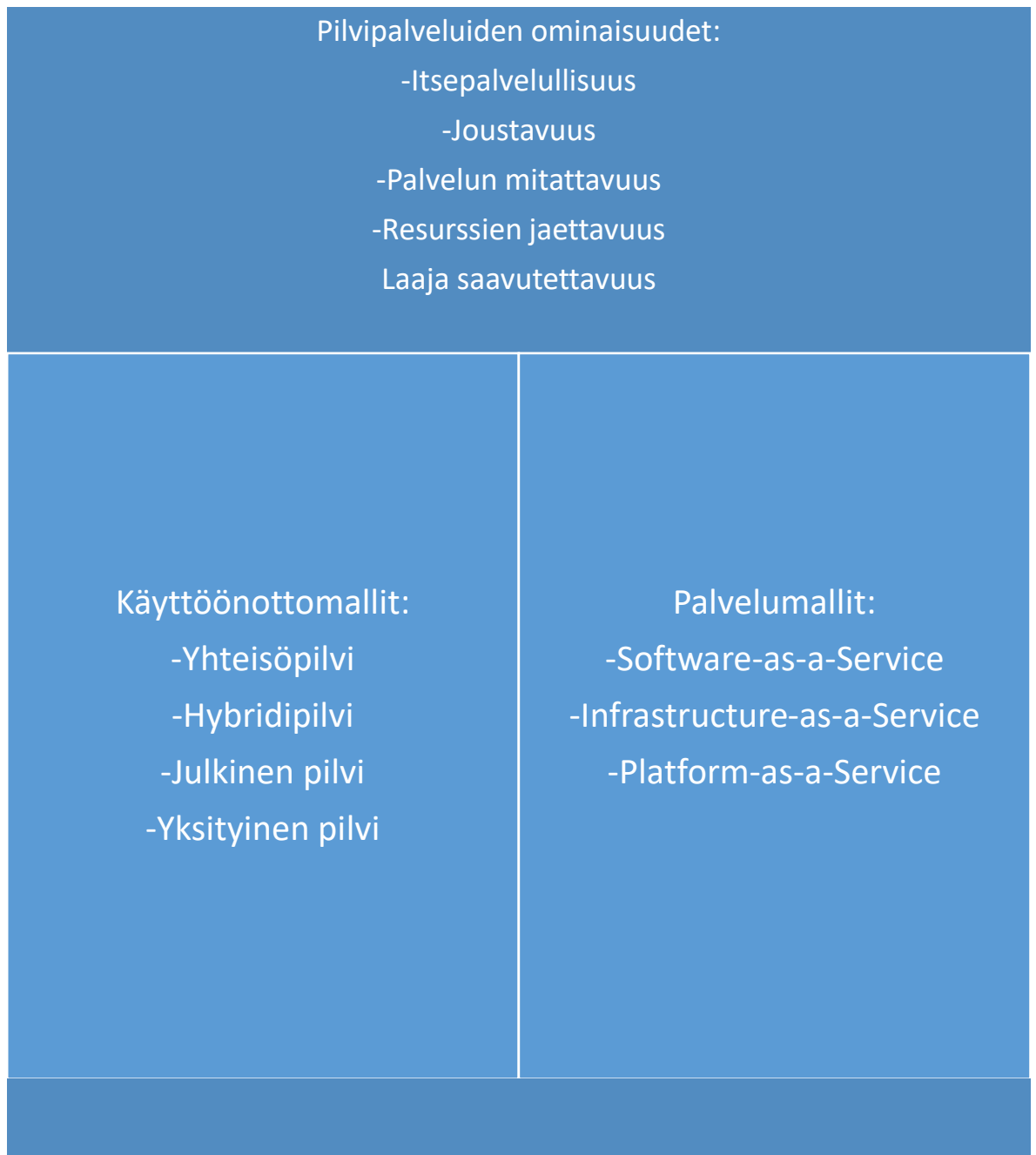
SaaS-malli mahdollistaa sovellusten käytön suoraan pilvestä. Yleensä palveluntarjoajat veloittavat joko palvelussa käytetyn ajan mukaan tai kuukausimaksulla. Saas-

mallin avulla on mahdollista säästää sekä yrityksen aikaa että kustannuksia. Sovellusta ei siis myydä asiakkaalle tai yritykselle, vaan siihen päästään käsiksi suoraan nettiselaimen kautta. Sovelluksia ei tarvitse asentaa, eikä erikseen konfiguroida. Ylläpitokustannuksia ei myöskään kerry, vaan niistä huolehtii palveluntarjoaja. (Vacca 2017. Luku 1 - Private Cloud Computing Essentials.)

Infrastruktuuri palveluna-malliin (IaaS) sisältyy esimerkiksi palvelun toimittajien hallinnoimat verkkolaitteet, palvelimet, tallennustilat, sekä muu infrastruktuuri. Palvelun kuluttajat ajavat palveluaan tai ohjelmistoaan pilvessä palveluntarjoajan infrastruktuuria hyödyntäen. Palvelusta kertyy yleensä yrityksille huomattavia aika- sekä kustannussäästöjä. Palvelusta veloitetaan usein käyttöasteen mukaan. (Vacca 2017, Luku 1 - Private Cloud Computing Essentials.)

PaaS rakentaa entisestään IaaS:n päälle käyttäen hyödykseen palveluntarjoajan tietokantoja ja väliohjelmistoja. PaaS:n avulla pystytään suorittamaan helposti skaalautuvia sovelluksia entistä kustannustehokkaammin. PaaS tarkoittaa yksinkertaisuudessaan sovellusalustaa, johon palvelun kuluttaja voi sovelluksensa siirtää. PaaS on lyhenne sanoista Platform-as-a-Service. (Vacca 2017, Luku 1 - Private Cloud Computing Essentials.)

Backend-as-a-service (BaaS) on pilvipalveluna toimiva backend, joita käytetään web- ja mobiilisovelluksissa ohjelmointirajapintojen (API) kautta. BaaS-palvelussa on mahdollisuus pilvitallennukseen, tiedostonhallintajärjestelmät, paikannuspalvelut ja käyttäjän hallintajärjestelmät. Näillä palveluilla on omat API rajapintansa, joten niiden käyttö eri sovelluksissa on helppoa API-kutsujen avulla. (What is BaaS? | Backend-as-a-Service vs. Serverless, 2019.) Alla olevassa kuvassa on havainnollistettu pilvipalveluiden ominaisuudet, yleisimmät pilvipalvelumallit, sekä niiden käyttöönottomallit.



Kuvio 2. Pilvipalvelumallit, palveluiden ominaisuudet, sekä käyttöönottomallit. (tiedot: Grance & Mell, 2011).

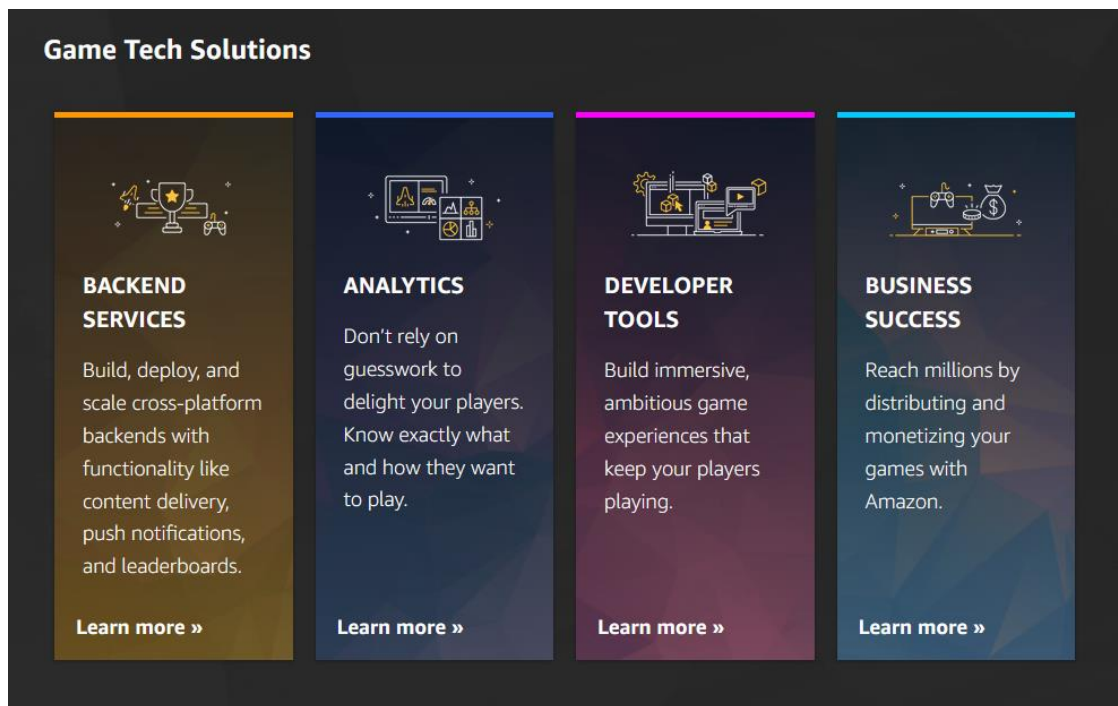
5 Amazon Web Services

Amazon Web Services tarjoaa monipuolisia pilvipalveluratkaisuja sekä yrityksille että yksityishenkilöille. Palveluista veloitetaan käyttöasteen mukaan. AWS tarjoaa kuitenkin kahdentoista kuukauden ilmaisen kokeilujakson suurelle osalle palveluista, jonka aikana palveluita voidaan käyttää rajoitettu määrä veloittamatta. Kun ilmainen kokeilujakso loppuu, palveluiden käytöstä aletaan veloittamaan automaattisesti. Jokainen palvelun uusi käyttäjä on oikeutettu ilmaiseen kokeilujaksoon, mutta tietyillä Microsoft Windows Server käyttöjärjestelmillä saattaa olla rajoituksia ilmaisen kokeilujakson käytössä. (Amazon Free Tier, 2018.)

Amazon Web Services tarjoaa oppilaitoksille AWS Educate -ohjelmaa, johon liittymällä kyseisen oppilaitoksen opiskelijat saavat vuosittain käyttöönsä 30100 dollarin arvosta krediittejä, joilla AWS:n palveluita voidaan käyttää. Krediittien määrä riippuu oppilaitoksen valitsemasta ohjelmasta. AWS Educate-palvelun kautta opiskelijat voivat lisäksi hakea töitä tai työharjoittelupaikkoja AWS:n parissa, tai esimerkiksi tallentaa palveluun oman ansioluettelonsa työnantajien nähtäville. (Teach Tomorrow's Cloud Workforce Today, 2019.)

5.1 Amazon Game Tech

Amazon Game Tech-palvelut ovat erityisesti peliprojektien käyttöön tarkoitettuja työkaluja. Amazonin Game Tech-palvelut tukevat pelinkehitystä koko pelin elämänsä ajan. Game Techin palveluihin sisältyy esimerkiksi pelin backend-palvelut (kuten GameSparks), analytiikkapalvelut, kehitys- ja sisällöntuotantotyökalut (Lumberyard, Twitch), sekä erilaiset liiketoiminta- ja mainontatyökalut (kuten Amazon Mobile Ads). (Amazon Game Tech, 2019.)



Kuvio 3. Amazonin tarjoamat Game Tech-ratkaisut (Game Tech Solutions, 2019)

5.2 Game Technin backend -palvelut

Amazon tarjoaa verkon yli pelattaville moninpeleille Gamelift-palvelua, jolla voidaan ottaa käyttöön Amazonin tarjoamia palvelimia pelissä. Palvelu sopii parhaiten sessiopohjaisille moninpeleille, ja sitä voidaan käyttää pelimoottorista riippumatta. Palvelun käyttöönottoaminen on nopeaa ja palvelun kapasiteettia voidaan skaalata tarpeen mukaan suuremmaksi. GameLift-palvelun avulla voidaan vähentää viivettä, sekä säästää kustannuksissa. (Amazon GameLift, 2018.)

GameLift toimii niin, että game server build ladataan Gameliftiin, ja Gamelift toimittaa sen hostattavaksi palvelimelle. Amazon Gamelift-palvelu hallitsee peliserverin hostaukseen tarvittavia laskentaresursseja ja mahdollistaa pelaajien yhdistämisen peliin. Gamelift säännöstelee resursseja, aloittaa ja pysäyttää pelisessiot, käsittelee pelaajien liittymispyynnöt löytämällä ja varaamalla vapaat paikat pelisessiossa. Näiden lisäksi palvelu kerää dataa esimerkiksi palvelinprosessien tilasta. (Amazon GameLift, 2018.)

GameSparks on Amazonin omistama peleille tarkoitettu pilvipohjainen backend-ratkaisu (Backend-as-a-Service), jolla voidaan rakentaa palvelinpuolen komponentteja helposti ja vaivattomasti. Palvelu tukee mm. pelaajien hallintaa (rekisteröityminen, autentikaatio), tulostaulukoita (leaderboards), pelaajien datan varastointia, ystävällistä, pelaajien välistä chattia, virtuaaliostoksia sekä analytiikka- ja raportointityökaluja. Peliprojekti täytyy rekisteröidä GameSparksin portaalissa, jonka kautta siihen voidaan luoda uusia ominaisuuksia, joita pelissä halutaan käyttää. GameSparks tarjoaa useita eri lisenssvaihtoehtoja, kuten indie-kehittäjille ja opiskelijoille suunnatun Indie & Student-lisenssin, Evaluation & Prototyping-tason lisenssin, sekä suuremmille yrityksille tarkoitettua Enterprise-tason lisenssin. Evaluation & Prototyping-lisenssi on tarkoitettu ainoastaan palvelun ominaisuuksien testaamiseen, eikä pelin julkaiseminen tällä lisenssityypillä ole mahdollista. (Game Sparks Pricing, 2018.)

GameSparksin Indie & Student -tason lisenssiin sisältyy kaikki tärkeimmät ominaisuudet, eikä sen käyttäminen maksa mitään, ellei pelillä ole yli sataatuhatta aktiivista kuukausittaista käyttäjää. Mikäli kuukausittaisten käyttäjien raja ylittyy, maksaa palvelun käyttö 2 senttiä jokaista sadantuhannen käyttäjän rajan ylittävää pelaajaa kohden. GameSparks on siis ilmainen käyttää tällä lisenssityypillä, paitsi jos peliprojekti saa huomattavan määrän suosiota. Lisenssityyppi sopii hyvin pelialaa opiskeleville, sekä pienille indie-tiimeille. Lisenssiä pitää erikseen hakea GameSparksista ja se voidaan myöntää joko opiskelijalle, pelikurssien opettajille tai muulle koulutushenkilöstölle, harrastelevalle pelinkehittäjälle tai pienille indie-kehittäjätiimeille. (Indie & Student Programme 2019.)

GameSparksin lisäksi Amazonin tarjoamien pilvipalveluiden avulla on mahdollista luoda itse alusta lähtien pelille tai sovellukselle backend käyttämällä AWS:n tarjoamia työkaluja. Tähän voidaan käyttää esimerkiksi Amazon SNS -viestintäpalvelua, Amazon Pinpoint -analytiikkapalvelua, Amazon Cognito -tunnistautumispalvelua, Amazon ElastiCache -datavarastoa, sekä Amazon Aurora ja DynamoDB -tietokantapalveluita. (Core Game Backends 2019.)

5.3 AWS:n analytiikkapalvelut

Amazonin analytiikkapalveluiden avulla voidaan kerätä ja tallentaa tärkeää dataa, riippumatta pelin koosta tai pelaajamäärästä. Analytiikkapalveluiden avulla voidaan myös mitata, kuinka pelin muutokset vaikuttavat esimerkiksi pelaajamäärään tai pelaajien käyttäytymiseen. Pelinkehitys on hyvin pitkälle päätösten tekoa ja valintoja. Analytiikkatyökalujen avulla saadaan tärkeää tietoa, joiden pohjalta pelinkehittäjät osaavat tehdä oikeita ratkaisuja. (Live Ops Analytics 2018.)

Amazon Redshift on nopea, skaalautuva datavarasto. Palvelu käyttää koneoppimista, jonka avulla saadaan datan varastoinnista ja analysoinnista nopeampaa. Palvelun avulla voidaan luoda uusi datavarasto nopeasti, tehdä sieltä kyselyitä ja analysoida dataa. (Amazon Redshift 2018.)

Amazon Kinesis -palvelun avulla voidaan kerätä, prosessoida ja analysoida suurta määrää esimerkiksi video-, ääni-, sekä lokidataa ja tehdä havaintoja datavirrasta reaaliajassa. Palvelulla voidaan käsitellä ja analysoida tietoja heti ilman että tietoja jäsennetään tai prosessoidaan ennen käsittelyn aloittamista. (Amazon Kinesis 2018.)

Amazon EMR:n avulla voidaan kerätä tietoja palvelun käyttäjien klikkaustietojen perusteella. Palvelua voidaan hyödyntää esimerkiksi mainosten kohdentamisessa oikeille asiakasryhmille, tai käyttäjien mieltymyksiä ymmärtämiseen. Palvelua voidaan käyttää myös sovellusten ja mobiilisovellusten lokitietojen prosessointiin. Palvelu auttaa ymmärtämään kuluttajakäyttäytymistä muuntamalla kerättyä dataa ymmärrettävään, hyödynnettävään muotoon. Palvelun avulla voidaan prosessoida nopeasti suuriakin määriä dataa. (Amazon EMR 2018.)

Amazon DynamoDB on ns. serverless NoSQL-tietokanta. Palveluun on sisäänrakennettuna tietoturva, sekä varmuuskopiointi ja palautus. DynamoDB skaalautuu automaattisesti tarpeen mukaan, ja se on monialueellinen tietokantaratkaisu, johon päästään tarvittaessa käsiksi useista sijainneista samanaikaisesti. Palvelussa käyttäjä voi määrittellä miltä alueilta tietokantaa käytetään, jolloin palvelu replikoi tietokannan lähimmille palvelimille. Tällä varmistetaan se, että palvelu toimii nopeasti ja luotettavasti sijainnista riippumatta. DynamoDB:tä käytetään monipuolisesti esimerkiksi peleissä, web-sovelluksissa sekä mobiilisovelluksissa. (Amazon DynamoDB 2018.)

5.4 Game Techin sisällöntuotantotyökalut

Amazon Game Tech tarjoaa lisäksi erilaisia sisällöntuotantotyökaluja. Lumberyard on Amazonin julkaisema avoimen lähdekoodin 3D-pelimoottori, joka on ladattavissa ilmaiseksi. Lumberyardin peliprojekteihin on helppo integroida AWS:n palveluita, kuten tiedon tallennusta ja varastointia, sekä pilvilaskentaa. Lisäksi AWS:n Gamelift-palvelun käyttöönottoa tuetaan Lumberyardissa tehokkaasti, tehden moninpelien kehittämisestä entistä helpompaa. Pelimoottori tukee kaikkia peligenrejä, sekä useita alustoja. Windowsin ja macOS:n lisäksi pelejä voidaan kehittää myös esimerkiksi Xboxille ja PS4-konsolille. (Lavieri, Dr. Edward 2016. Welcome to Lumberyard.)

Twitch on Amazonin omistama alusta, jonka kautta voidaan suoratoistaa itse tai katsella muiden käyttäjien suoratoistamaa pelikuvaa livenä. Twitch julkaistiin vuonna 2011 ja sen päätoimintoja ovat oman pelikuvan suoratoisto katsojille, suoratoistojen katsominen livenä, aiempien suoratoistojen katsominen jälkikäteen tallenteina, sekä lähetykseen osallistuminen chatin kautta. Twitch ei alunperin ollut Amazonin omistuksessa, mutta Amazon päätyi ostamaan sen palvelun suuren suosion ja nopean kasvun johdosta. Twitch tuo pelikokemukseen sosiaalisen komponentin, jossa sekä pelin suoratoistaja että katsojat voivat kommunikoida keskenään. Näiden toimintojen lisäksi suoratoistaja voi esimerkiksi luoda omia chat-komentoja, kyselyitä joihin katsojat voivat vastata, sekä kutsua katsojia liittymään samaan pelisessioon. (Lavieri, Dr. Edward. 2016. Engaging With Users on Twitch.)

Twitch-palvelun tarjoamat tärkeimmät komponentit, joita peliprojekteissa voidaan huödyntää ovat Twitch ChatPlay, Twitch JoinIn, sekä Twitch API. Twitch ChatPlay on Lumberyardin käyttämä sovelluskehys (framework), jolla peliin voidaan tuoda Twitchin toiminnallisuuksia. Twitch ChatPlayn avulla peliin voidaan jo kehitysvaiheessa luoda komentoja, jotka vaikuttavat suoraan pelin kulkuun. Twitchin kautta suoratoistettaessa pelin katsojat voivat halutessaan käyttää komentoja ja vaikuttaa sitä kautta suoraan pelikokemukseen. (Lavieri, Dr. Edward 2016. Engaging With Users on Twitch.)

Amazon tarjoaa työkaluja myös äänipelien tekemiseen Alexa-yhteensopiville laitteille. Tarjolla olevilla työkaluilla voidaan luoda sosiaalisia moninpelejä, kuten tietovisoja, joita voidaan pelata samassa tilassa usean pelaajan kesken. Alexa-pelien ja sovellusten kehityksessä käytetään Alexa Presentation Languagea (APL). (Alexa Multimodal, 2018.)

5.5 AWS:n liiketoimintatyökalut

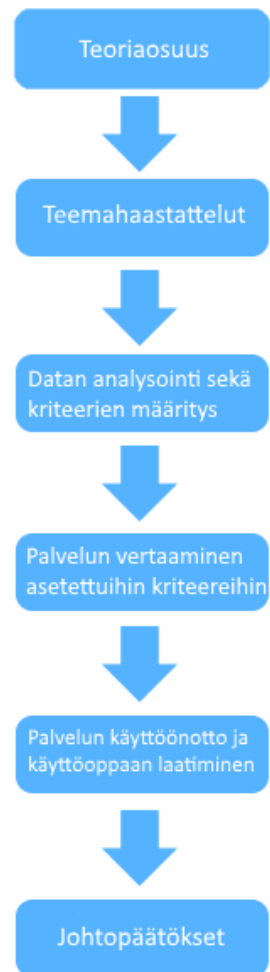
AWS Partner Network on suunniteltu auttamaan kaikenkokoisia AWS:n palveluita käyttäviä yrityksiä kasvattamaan liiketoimintaansa. Partner Network on asiakaslähtöinen palvelu, jossa alan asiantuntijat tarjoavat palveluitaan. Amazon Partner Networkin kumppanit voidaan jakaa konsultointipartnereihin, sekä teknologiapartnereihin. Konsultointipalveluihin erikoistuneet kumppanit ovat asiantuntevia palveluyrityksiä, jotka auttavat kaikenkokoisia asiakkaita suunnittelemaan, siirtämään sekä rakentamaan uusia AWS-pohjaisia sovelluksia tai pelejä. Teknologiapartnerit tarjoavat lähinnä ohjelmistoratkaisuja ja yhteistyömahdollisuuksia eri yritysten välillä. (Work with an AWS Partner 2018.)

6 Tutkimustulokset ja niiden analysointi

6.1 Tutkimuksen toteutus

Tutkimusosuus koostuu kahdesta osasta. Ensimmäisessä osuudessa on tarkoitus selvittää, soveltuuko Amazonin Game Techin backend-palvelut toimeksiantajan tarpeisiin. Tähän saadaan selvyys suorittamalla haastatteluita Ticorporaten toimintaa ohjaavilta opettajilta. Koimme saavamme ajankohtaisinta ja tarkoituksenmukaisinta tie-

toa keskittämällä haastattelut toimintaa ohjaaviin henkilöihin. Analysoimalla haastatteluista saatua aineistoa saadaan selville kriteerit, jotka palvelun tulee täyttää, jotta saadaan selville, soveltuuko palvelu käyttöön vai ei. Haastattelut suoritettiin tammi-kuussa 2018 ja aineiston hankintatapana käytimme puolistrukturoituja haastatteluja, eli teemahaastattelua. Tarkoituksena oli luoda keskustelunomainen ilmapiiri, jossa hyödynsimme ennalta määriteltyjä kysymyksiä saadaksemme mahdollisimman tarkan kuvan Ticorporaten nykytilanteesta, tulevaisuudesta ja palvelun tarpeesta. Mikäli palvelu täyttää kriteerit, eli soveltuu Ticorporaten käyttöön, tehdään toisessa tutkimusosassa palvelun käyttöönotolle kirjallinen opas, jota Ticorporateen osallistuvat opiskelijat voivat toivon mukaan hyödyntää peliprojekteissaan. Alla olevassa kuvassa on esitetty tutkimuksen eteneminen, jotta tutkimuksesta saadaan selkeämpi kuva.



Kuvio 4. Tutkimuksen toteutus

6.2 Tarpeen määrittely

Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman toisen vuoden Ticorporate-tuotekehitysprojekteissa opiskelijat luovat pienryhmissä lähinnä hyötysovelluksia ja mobiilipelejä. Ticorporaten on tarkoitus simuloida todellista tuotetta kehittävää yritystä, jonka toiminnasta ja organisoinnista vastaavat opiskelijat opettajien avustuksella. Ongelmana lyhyissä peliprojekteissa on, että pelien backendien pystyttäminen on aikaa vievää ja haastavaa, eikä projektien aikana välttämättä ke- retä tutustumaan tarkemmin tarjolla oleviin muihin ratkaisuihin. Tästä syystä oli tar-

peellista selvittää, soveltuuko Game Techin tarjoamat valmiit backend-ratkaisut Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelman opiskelijoiden käyttöön, ja vastaavatko ne toimeksiantajan tarpeita. Pohdinta-osuudessa on myöhemmin käyty läpi muita Game Techin tarjoamia palveluita ja pohdittu niiden soveltuvuutta Jyväskylän ammattikorkeakoulun tietojenkäsittelyn opiskelijoiden käyttöön.

6.3 Teemahaastattelut ja niiden tulokset

Tässä luvussa käsitellään haastatteluista saatua aineistoa. Haastattelut olivat opinnäytetyössämme tärkeässä osassa, jotta ymmärrämme paremmin tutkittavaa aihetta, toimeksiantajan tarpeita sekä heidän näkemyksiään palvelun tarpeesta ja nykytilanteesta. Aineistonkeruumenetelmänä käytimme puolistrukturoitua haastattelua. Kysymykset oli laadittu etukäteen, mutta keskustelu yritettiin pitää mahdollisimman vapaana, jotta haastatteluista saataisiin mahdollisimman paljon irti. Haastatteluja tehtiin kaksi, eli analysoitavaa dataa ei ollut määrällisesti kovin paljoa. Koimme kuitenkin saaneemme riittävästi aineistoa tutkimuksessa etenemiseen, sekä selkeän idean palvelun tarpeesta ja kriteereistä. Kysymyksiä, joita haastatteluissa kysyimme, olivat mm. minkälaisia kriteereitä backend-palvelulla tulisi olla, tarvitseeko palvelun olla ilmainen, mitä palvelun käyttöönotossa tulee huomioida, minkälaisissa projekteissa näkisit palvelusta olevan hyötyä, sekä minkälaisissa projekteissa backend-palveluista voisi olla hyötyä.

Haastattelutilanteessa hyödynnettiin muistiinpanoja, jotta pystyimme paremmin ottamaan ylös yksityiskohtaista tietoa haastateltavien mielipiteistä ja jotta voimme palata vastauksiin myöhemmin. Haastattelut on tehty parityönä, jossa toinen tekijöistä keskittyi haastattelukysymysten esittämiseen, ja toinen kirjoitti tarkat muistiinpanot haastattelutilanteesta ylös. Laadullisen aineiston analyysin aloitimme haastatteluiden jälkeen. Analysoimme haastattelukysymyksiin saatuja vastauksia ottamalla haastattelutilanteen muistiinpanoista pääpointit ylös kysymys kerrallaan. Alla pienimuotoinen analyysi haastatteluiden tärkeimmistä tuloksista. Haastattelun runko löytyy opinnäytetyön liitetiedostona.

Palvelun pakolliset kriteerit	Palvelun käyttöönotossa huomioitavaa	Muut huomiot
Palveluiden käyttö tulee olla opiskelijoille ilmaista.	Yhteensopivuus käytettävien työkalujen kanssa.	Opiskelijaprojekteissa ei tehdä verkon yli pelattavia moninpelejä.
Riittävä dokumentaation taso		Peliprojekteissa keskitytään mobiilipeleihin.
Nopea ja helppo käyttöönotto		Peliprojektien kesto huomioitava

Kuvio 5. Tulosten analysoinnista saadut kriteerit.

Haastatteluiden aikana saimme selville, että Ticorporaten kuuden kuukauden tuotekehitysprojekteissa keskitytään lähinnä mobiilipelien- ja hyötysovellusten kehittämiseen. Kuusi kuukautta on lyhyt aika, kun ottaa huomioon, että monelle opiskelijalle kyseessä on ensimmäinen peliprojekti ja opettelemista on paljon. Tästä syystä palvelun nopea ja vaivaton käyttöönotto ja palvelun riittävä dokumentaatio nähtiin välttämättömäksi ominaisuudeksi. Riittävä palvelun dokumentaatio on välttämätön, jotta opiskelijat pääsevät nopeasti tutustumaan palveluun ja sen ominaisuuksiin. Backendin rakentaminen pelille on aikaa vievää, joten tähän tarvitsi löytää nopeampi ratkaisu. Yhtenä tärkeänä ominaisuutena pidettiin sitä, että palvelun käytön tulisi olla opiskelijoille ilmaista. Yllä esille ilmenneiden kriteerien lisäksi nähtiin, että valmiin backend-palvelun käyttämisestä omissa projekteissaan opiskelijat saisivat tärkeää ohjelmointikokemusta, josta voisi olla hyötyä tulevaisuudessa myös muissa projekteissa tai työelämässä. Mikäli kriteerit täyttyvät, nähtiin hyödyllisimmäksi vaihtoehdoksi käyttää käyttöoppaan kirjoituksessa Unityn graafista editoria, koska se on Ticorporaten projekteissa eniten käytössä oleva editori.

Haastatteluista saatujen kriteerien pohjalta tarkasteltiin Game Technin GameSparks-palvelun ominaisuuksia, ja verrattiin sitä muihin vastaaviin palveluihin. Palvelun

käyttö on kirjoitushetkellä ilmaista sadan tuhannen kuukausittaisen käyttäjän rajaan asti Indie & Student-lisenssillä, joten tärkeäksi ominaisuudeksi nähty ilmaisuus täyttyy (Game Sparks Technologies, 2019). Palvelun käytölle löytyy verkosta runsaasti tutoriaaleja, ja jokaiselle toiminnolle löytyy dokumentaatio GameSparksin Learn+-palvelusta.

6.4 BaaS -palveluiden vertailu

GameSparksin kaltaisia BaaS-palveluita on olemassa useita, joten vertailimme seuraavaksi palvelua muihin vastaaviin palveluihin. Vertailtavia kriteereitä olivat palvelun hinta, sekä palvelun ominaisuudet. Vastaavia suosituimpia palveluita, joita löysimme, olivat Microsoft Azuren PlayFab, Googlen Firebase-palvelu, sekä Photon. Loimme palveluiden vertailua varten taulukon, jotta vertailtavia ominaisuuksia voidaan paremmin havainnollistaa.

	Hinta	Ominaisuudet
GameSparks	ilmainen 100 000 MAU:hun asti	Leaderboardit, pelaajan hallinta, pelin julkaisu, pelin sisäiset ostot, dokumentaatio, sekä muut tärkeimmät ominaisuudet sisältyvät ilmaiseen Student & Indie-tason lisenssiin.
Playfab	Essentials-tason lisenssillä ilmainen käyttää, ei MAU-rajoitusta.	Tärkeimmät ominaisuudet sisältyvät ilmaiseen lisenssiin, kuten leaderboardit, pelaajan hallinta, pelin sisäiset ostot, dokumentaatio, ilmainen lisenssi ei sisällä palvelutasosopimusta tai tukea.
Photon	Ilmaiseen lisenssiin sisältyy 8000 MAU:ta.	Tarkoitettu lähinnä multiplayer-peleille, ei sisällä esimerkiksi

		valmista Leaderboard-toiminnallisuutta.
Firebase	Ilmaiseen lisenssiin sisältyy rajoitettu määrä samanaikaisia yhteyksiä (100 kpl), MAU-rajoituksesta ei mainita.	Ei yhtä paljoa valmiita peleille räätälöityjä toiminnallisuuksia, kuin muissa vaihtoehdoissa. Käyttäjän tunnistus, datan varastointi, yksi ilmainen tietokanta projektia kohden. Ei esim. valmista leaderboard-toiminnallisuutta.

Kuvio 6. Taulukko eri BaaS-palveluiden vertailusta. (Pricing 2019; Free to start, simple pricing that scales with your game. 2019; Pricing Plans n.d; Select Your Photon Unity Package 2019)

Vertailun jälkeen totesimme Microsoftin Playfab-palvelun olevan lähimpänä GameSparksia ominaisuuksiltaan ja hinnoittelultaan. Molemmat palveluista ovat käytännössä ilmaisia perusominaisuuksiltaan. Koettiin kuitenkin GameSparksin dokumentaation olevan paremmalla tasolla kuin PlayFabissa, joten lähdimme tekemään käyttöopasta GameSparks-palvelun käyttöönotolle. Photonin ominaisuudet vaikuttivat painottuvat moninpelien tarpeisiin, ja se on ominaisuuksiltaan lähempänä Amazonin GameLift-palvelua. Firebase olisi mahdollisesti myös varteenotettava vaihtoehto, mutta se ei sisällä yhtä paljoa peleille räätälöityjä ominaisuuksia kuin esimerkiksi GameSparks sekä PlayFab. Tästä syystä sen käyttöönotossa kestäisi enemmän aikaa. Tutkimuksen perusteella tehty käyttöopas löytyy työn liitteenä ja alla on kuvattu sen alustava sisällysluettelo.

Sisältö

1	Johdanto	2
2	GameSparks-palvelun tärkeimmät ominaisuudet opiskelijan näkökulmasta	2
3	GameSparks-portaali ja peliprojektin rekisteröinti.....	2
4	GameSparksin käyttöönotto Unity-projektissa.....	3
	4.1 GameSparks-työkalujen tuominen Unity-projektiin ja alkuasetusten määrittäminen Unityssä.....	3
	4.2 Pelin ominaisuuksien luominen GameSparks-portaalissa.....	3
	4.3 Käyttäjän tunnistaminen ja rekisteröinti, kirjautumisikkunan luominen Unityssä	3
	4.4 Pelaajan tietojen noutaminen ja käsittely pelikoodissa.....	3
5	Yhteenveto	3

Kuvio 7. Käyttöoppaan alustava sisällysluettelo

7 Johtopäätökset

Tässä luvussa on käyty läpi, kuinka tutkimuskysymyksiin saatiin työssä vastaus ja mitä saatiin aikaiseksi. Opinnäytetyön lopputuloksena luotiin käytännönläheinen ohjeistus GameSparks-palvelun käyttöönottoon Unityn graafisessa editorissa, jota Jyväskylän ammattikorkeakoulun opiskelijat voivat toivon mukaan hyödyntää omissa peliprojekteissaan. Ohjeistus toteutettiin kirjallisena dokumenttina, jossa käytiin läpi vaihe vaiheelta palvelun perusominaisuuksien käyttöönotto. Kirjallinen ohjeistus löytyy tämän työn liitetiedostona. Ohjeistus toteutettiin Ticorporatessa eniten käytössä olevalla graafisella Unity-editorilla, jotta ohjeistus olisi kohderyhmälle mahdollisimman olemassa. Kirjallinen opas luotiin opinnäytetyön aikana itse tehdyn peliprojektin pohjalta.

Soveltuuko Amazon Game Tech-palvelut Ticorporaten käyttöön?

Saimme haastatteluiden pohjalta selville kriteerit, jotka backend-palvelun tulee täyttää, jotta se nähdään tarpeelliseksi Ticorporatessa. Keskityimme työssä enemmän backend-palvelu GameSparkiin, jonka käyttöönoton pohjalta totesimme sen olevan nopea ottaa käyttöön, helposti muokattavissa ominaisuuksiltaan, sekä hinnaltaan opiskelijoille ilmainen. Näiden ominaisuuksien pohjalta vertasimme palvelua sille asetettuihin kriteereihin, ja totesimme kriteereiden täyttyvän. Pohdinta-osuudessa on käyty läpi mitä muutamat muut Game Techin palvelut voivat tarjota opiskelijoiden käyttöön.

Kuinka Amazon Game Tech backend-palvelut otetaan käyttöön peliprojektissa?

Valmiina BaaS-palveluna Amazon tarjoaa GameSparks-palvelua, jonka otimme käyttöön peliprojektissa. Vastasimme tutkimuskysymykseen luomalla palvelun käyttöönotolle oppaan opinnäytetyötä varten tehdyn peliprojektin pohjalta. Käyttöopas löytyy opinnäytetyön liitteistä. Käyttöönoton pohjalta totesimme, että palvelun käyttöönotto Unityssä on nopeaa ja ominaisuuksien konfigurointi portaalin kautta on helppoa. Kävimme oppaassa läpi, kuinka SDK tuodaan Unity-projektiin, kuinka tulostaulukot ja eventit luodaan ja kuinka niitä käytetään, kuinka käyttäjä voi rekisteröityä ja tunnistautua, sekä muita perusominaisuuksia. Huomioimme oppaan tekemisessä kohdeyhmää ja jätimme oppaasta pois tiettyjä ominaisuuksia, joista emme kokeneet olevan peliprojekteissa hyötyä. Näitä olivat mm. moninpeliominaisuudet. Kaikkien ominaisuuksien läpi käyminen olisi vienyt liikaa aikaa, joten keskityimme niihin ominaisuuksiin, jotka olivat mielestämme tärkeimmät.

Kuinka Game Techin tarjoamat backend-palvelut eroavat muista vastaavista palveluista?

Vastataksemme tutkimuskysymykseen teimme GameSparksin kaltaisten BaaS-palveluiden vertailun, jossa vertailtavia kohteita olivat palvelun ominaisuudet ja hinta. Vertailun tavoitteena oli selvittää, onko markkinoilla muita vastaavia Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opiskelijoiden peliprojekteihin sopivia palveluita ja missä hintaluokassa. Löysimme lisäksi kolme samantyyppistä potentiaalista palvelua, joista GameSparksin ohella käyttökelpoisimmaksi todettiin Mic-

rosoftin PlayFab-palvelu. PlayFabin perusominaisuuksien käyttö myös pelin julkaisemisen jälkeen on ilmaista Essentials-tason lisenssillä, ilman MAU -rajoitusta. Microsoftin PlayFab-palvelu on opiskelijoiden käyttöön mielestämme toinen hyvä vaihtoehto, mikäli projekteissa halutaan käyttää valmiista BaaS-palvelua. Muita palveluita olivat Firebase ja Photon, jotka ominaisuuksiltaan eivät olleet Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelman opiskelijoiden tarpeisiin yhtä sopivia. Firebasessa vaikutti olevan vähemmän valmiita peleille tarkoitettuja ominaisuuksia, joten käyttöönotossa menisi enemmän aikaa. Photon taas vaikutti olevan lähinnä moninpeleille tarkoitettu palvelu, joka on ominaisuuksiltaan lähempänä GameLift-palvelua.

8 Pohdinta

Tässä luvussa käsitellään läpi työn tavoitteet, tutkimuksen tulokset, sekä tulosten käyttökelpoisuuden arviointi. Työn tavoitteena oli selvittää soveltuvatko Amazon Game Techin palvelut Jyväskylän ammattikorkeakoulun tietojenkäsittelyn opiskelijoiden käyttöön heidän peliprojekteissaan. Opinnäytetyön pääpaino oli kuitenkin Game Techin backend-palveluiden soveltuvuuden tutkimisessa. Amazon Game Tech tarjoaa valmiina BaaS-palveluna GameSparksia, mutta AWS tarjoaa myös työkaluja backendin rakentamiseen itse alusta asti. Lisäksi Game Tech tarjoaa Gamelift-palvelua, joka on lähinnä sessiopohjaisille moninpeleille tarkoitettu hallittu palvelu, jonka avulla voidaan ottaa käyttöön Amazonin tarjoamia servereitä monipeliprojektissa. Gameliftiä ei nähty tarpeelliseksi, koska haastatteluiden pohjalta saimme selville, että koulutusohjelman opiskelijaprojekteissa ei juurikaan tehdä verkon yli pelattavia moninpelejä. Amazonin tarjoamia analytiikkapalveluita emme nähneet hyödyllisiksi projekteissa, koska näiden käyttäminen ja konfigurointi vie liikaa aikaa, kun ottaa huomioon peliprojektien keston.

Game Techin tarjoamista kehittäjätyökaluista Lumberyardin pelimoottori saattaisi olla tietyissä projekteissa hyödyllinen. Emme kuitenkaan löytäneet muuta syytä käyt-

tää Lumberyardia Unityn sijasta kuin se, että sillä on parempi tuki tietyille AWS:n palveluille. Lumberyardissa on myös sisäänrakennetut työkalut Amazonin Twitch-suoraistoalustan toiminnallisuuden käyttöönotolle peliprojektissa.

Mielestämme haastatteluissa olisimme voineet laatia kysymykset hieman tarkemmin, koska osa kysymyksistä olivat liian laajoja. Osan suunnitelluista kysymyksistä jätimme kokonaan pois ja esitimme tilalla tarkentavia kysymyksiä. Olisimme saaneet haastatteluista vielä enemmän irti, jos olisimme suunnitelleet tiettyjä kysymyksiä paremmin etukäteen. Koimme kuitenkin saaneemme haastatteluista tarvittavat tiedot tutkimuksessa etenemiseen, vaikka aineistoa ei määrällisesti ollut paljoa.

Tutkimuksen aikana GameSparks-palvelun hinnoittelu muuttui. Huomasimme palveluun tulleen hinnan muutoksen liian myöhään, sillä opiskelijoiden käyttöön tarkoitettu käyttöopas oli jo lähestulkoon valmis. Aiemmin ilmaiseksi opiskelijoille tarjottu Student & Indie-tason lisenssi oli poistunut käytöstä. Palvelun hinnoittelumuutoksen takia emme voi suositella enää GameSparksia käytettäväksi Jyväskylän ammattikorkeakoulun opiskelijoiden peliprojekteissa, koska opiskelijaprojekteissa palvelua ei voida enää käyttää ilmaiseksi. Hinnoittelumuutos tuli tutkijoille yllätyksenä, eikä tutkimuksen suuntaa enää tässä vaiheessa voitu muuttaa. Palvelun hinnoittelu muuttui äkillisesti vuoden 2019 helmikuun lopussa, ja uudella hinnoittelumallilla palvelun käyttö maksaa pelin julkaisun jälkeen vähintään 299 dollaria kuukaudessa (Pricing, 2019; Self-Service Pricing FAQs, 2019). Voimme suositella GameSparksia käytettäväksi ainoastaan palvelun ominaisuuksien testausta varten, sillä uudella hinnoittelumallilla palvelua voidaan käyttää ilmaiseksi ainoastaan siihen asti, kun peli julkaistaan. Tästä syystä suosittelemme opiskelijaprojektien käyttöön mieluummin Microsoftin PlayFab-palvelua, koska tätä voidaan käyttää projekteissa edelleen ilmaiseksi, ilman MAU-rajoitusta. Mikäli olisimme huomanneet palveluun tulleen hinnoittelumuutoksen aiemmin, olisimme kerenneet vaihtaa vielä suuntaa ja tehdä esimerkiksi käyttöoppaan mieluummin Microsoftin PlayFab-palvelun käyttöönotolle. GameSparks-palvelua voi edelleen kuitenkin testimielessä käyttää ilmaiseksi uudella Development-tason lisenssillä, mutta peliä ei voida julkaista tällä lisenssityypillä.

Lähteet

Alexa Multimodal. Amazonin Alexa Multimodal-dokumentaatio. Viitattu 21.11.2018. Verkkajulkaisun osoite: <https://developer.amazon.com/alex-skills-kit/multimodal>.

Amazon DynamoDB. AWS:n verkkosivut. Viitattu 7.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/dynamodb>.

Amazon EMR. AWS:n verkkosivut. Viitattu 7.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/emr>.

Amazon Game Tech. AWS:n verkkosivut. Viitattu 11.1.2019. Verkkajulkaisun osoite: <https://aws.amazon.com/gametech/>.

Amazon GameLift. AWS:n verkkosivut. Viitattu 4.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/gamelift/>.

Amazon Kinesis. AWS:n verkkosivut. Viitattu 7.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/kinesis/>.

Amazon Partner Network. AWS: n verkkosivut. Viitattu 22.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/gametech/value-programs/>.

Amazon Redshift. AWS:n verkkosivut. Viitattu 5.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/redshift/>.

AWS Free Tier. AWS:n verkkosivut. Viitattu 4.11.2018. Verkkajulkaisun osoite: <https://aws.amazon.com/free/>.

Castrillo, I. Rountree, D. 2014. The Basics of Cloud Computing: Understanding the Fundamentals of Cloud Computing in Theory and Practice. Syngress Publishing.

Core Game Backends. AWS:n verkkosivut. Viitattu 4.4.2019. Verkkajulkaisun osoite: <https://aws.amazon.com/gaming/backend-services/>.

Events. GameSparks Learn+ verkkosivut. Viitattu 17.3.2019. Verkkójulkaisun osoite: <https://docs.gamesparks.com/documentation/configurator/events/>.

Game Tech Solutions. AWS:n verkkosivut. Viitattu 17.3.2019. Verkkójulkaisun osoite: <https://aws.amazon.com/gametech/>.

GameSparks Pricing. GameSparksin verkkosivut. Viitattu 21.12.2018. Verkkójulkaisun osoite: <https://www.gamesparks.com/pricing/>.

Grance, T.& Mell, P.2011.The NIST Definition of Cloud Computing. Viitattu 26.10.2018. Verkkójulkaisun osoite: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.

Gregory, J. 2009. Game Engine Architecture. A K Peters. Ltd. Wellesley, Massachusetts.

Hill, R. Hirsch, L. Lake, P. Moshiri, S. 2013. Guide to Cloud Computing: Principles and Practice. Springer.

Indie & Student Programme. GameSparksin verkkosivut. Viitattu 20.1.2019. Verkkójulkaisun osoite: <https://www.gamesparks.com/indie-student-programme-faq/>.

Kananen, J. 2008. Kvali. Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Lavieri, Dr. Edward. 2016. Learning AWS Lumberyard Game Development. Packt Publishing.

Live Ops Analytics. AWS:n verkkosivut. Viitattu 27.11.2018. Verkkójulkaisun osoite: <https://aws.amazon.com/gametech/live-ops-analytics/>.

Menard, M. 2012. Game Development with Unity. Cengage Learning.

Pricing Plans. Google Firebasen verkkosivut. Viitattu 14.3.2019. Verkkójulkaisun osoite: <https://firebase.google.com/pricing/>.

Pricing. GameSparksin verkkosivut. Viitattu 1.4.2019. Verkkójulkaisun osoite: <https://www.gamesparks.com/pricing/>.

Pricing. PlayFabin verkkosivut. Viitattu 14.3.2019 Verkkójulkaisun osoite: <https://playfab.com/pricing/>.

Rabin, S. 2010. Introduction to Game Development, Second Edition. Cengage Learning.

Rogers, S. 2014. Level Up! The Guide to Great Video Game Design, Second Edition. John Wiley & Sons, Incorporated.

Select Your Photon Unity Package. Photon Enginen verkkosivut. Viitattu 13.3.2019. Verkkójulkaisun osoite: <https://www.photonengine.com/en/pun/pricing/>.

Self-Service Pricing FAQs. GameSparksin verkkosivut. Viitattu 1.4.2019 Verkkójulkaisun osoite: <https://www.gamesparks.com/pricing/faqs/>.

Social & Multiplayer. GameSparksin verkkosivut. Viitattu 10.2.2019. Verkkójulkaisun osoite: <https://www.gamesparks.com/product/>.

Teach Tomorrow's Cloud Workforce Today. Amazon Web Services. AWS educaten verkkosivut. Viitattu 11.1.2019. Verkkójulkaisun osoite: <https://aws.amazon.com/education/awseducate/>.

The world's leading real-time creation platform. Unityn verkkosivut. Unity Technologies. Viitattu 6.3.2018. Verkkójulkaisun osoite: <https://unity3d.com/unity/>.

Twitch. Twitch.tv:n kotisivut. Viitattu 17.11.2018. Verkkójulkaisun osoite: <https://www.twitch.tv/p/about/>.

Vacca, J. 2017. Security in the Private Cloud. CRC Press.

What is BaaS? | Backend-as-a-Service vs. Serverless. Cloudflaren verkkosivut. Viitattu 14.4.2019. Verkkójulkaisun osoite: <https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>.

Work with an AWS Partner. AWS:n Partner Networkin verkkosivut. Viitattu 29.12.2018. Verkojulkaisun osoite: https://aws.amazon.com/partners/find-a-partner/?sc_ichannel=ha&sc_icampaign=gamedev_valueprograms&sc_isegment=en&sc_iplace=cta_ed&sc_icontent=learn_more.

Writing Cloud Code. GameSparks Learn+ verkkosivut. Viitattu 22.3.2019. Verkojulkaisun osoite: <https://docs.gamesparks.com/documentation/configurator/cloud-code/>.

Liitteet

Liite 1. Teemahaastattelun runko

Teemahaastattelun runko

Haastateltavan taustatiedot:

Jyväskylän ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Ticorporaten vastuopettajat

Haastattelun toteutus

Haastattelijat: Sami Verronen, Joakim Piippo

Teemat

Teema 1 Ticorporaten tilanne ja palvelun tarve

Esitetyt kysymykset

Mitä ominaisuuksia palvelussa tulee olla? Mitä kriteerejä palvelun tulisi ehdottomasti täyttää Ticorporatea ajatellen?

Miten Gametechin backend-palveluita voitaisiin hyödyntää Ticorporaten projekteissa?

Tarvitseeko palvelun olla ilmainen?

Mitä palvelun käyttöönotossa tulisi huomioida?

Minkälaisissa projekteissa näkisit palveluista olevan hyötyä?

Liite 2. Opas GameSparks-palvelun käyttöönottoon Unityn graafisessa editorissa.

Opas GameSparks-palvelun käyttöönottoon Unityn graafisessa editorissa

Sisältö

1	Johdanto	39
2	GameSparks-palvelun tärkeimmät ominaisuudet	40
3	GameSparks-portaali ja uuden peliprojektin rekisteröinti	41
4	GameSparksin käyttöönotto Unity-projektissa	45
	4.1 GameSparks-työkalujen tuominen Unity-projektiin ja alkuasetusten määrittäminen Unityssä	45
	4.2 Pelin ominaisuuksien luominen GameSparks-portaalissa	48
	4.3 Käyttäjän tunnistaminen ja rekisteröinti, kirjautumisikkunan luominen Unityssä	52
	4.4 Pelaajan pisteiden tallennus Leaderboardille	55
	4.5 Pelaajan tietojen noutaminen ja käsittely pelikoodissa, Leaderboard-datan hakeminen peliprojektiin	57
	4.6 Virtuaalivaluutan käyttö peliprojektissa pilvikoodia hyödyntäen.....	59

Kuviot

Kuvio 1. Uuden peliprojektin luominen GameSparks-portaalissa.	41
Kuvio 2. GameSparksissa valittavat ominaisuudet ja integraatiot.....	42
Kuvio 3. Maantieteelliset rajoitukset GameSparksissa.	43
Kuvio 4. GameSparks-portaalin etusivu palveluun rekisteröitymisen jälkeen.	44
Kuvio 5. Kirjautumispyynnön lähettäminen GameSparksin testiympäristössä. Inspector-paneelissa nähdään saatu JSON-vastaus.	44
Kuvio 6. Custom Packagen lisääminen Unity-projektiin. Tätä kautta haetaan ladattu GameSparks SDK-projektiin.	46
Kuvio 7. API Key ja API Secretin lisääminen Unity-projektissa.	47
Kuvio 8. Pelin ensimmäiseen Sceneen lisättävä GameSparksManager-objekti.	48
Kuvio 9. Event, joka tallentaa pelaajan saamat pisteet tulostaulukkoon.	49
Kuvio 10. Uuden tulostaulukon luominen portaalissa.	50
Kuvio 11. Uuden saavutuksen luominen.....	51
Kuvio 12. Sisäänkirjautumis- ja rekisteröintiruutu.....	52
Kuvio 13. Uuden pelaajan rekisteröitymispyyntö.	53
Kuvio 14. Pelaajan sisäänkirjautuminen peliin AuthenticationRequestia hyödyntäen.	54
Kuvio 15. Käyttäjän uloskirjaaminen GameSparks-palvelusta.....	55
Kuvio 16. Esimerkki, jossa on kuvattu pelaajan pisteiden kerääminen pelin aikana...56	56
Kuvio 17. Esimerkki skriptistä, joka lähettää pelaajan saamat pisteet tulostaulukkoon käyttäen aiemmin luotua eventtiä.	56
Kuvio 18. Pelaajan saavutusten hakeminen GameSparksista peliprojektiin.	58
Kuvio 19. Käyttäjän saavutusten hakeminen.	58
Kuvio 20. Leaderboard-tietojen haku GameSparksista.....	59
Kuvio 21. Uuden virtuaalivaluutan luonti, Eventin ja sen attribuutin luonti, sekä sen käyttö pelikoodissa LogEventRequestin avulla.	60
Kuvio 22. Esimerkki pilvessä ajettavasta koodista.	61

1 Johdanto

Tämän ohjeistuksen tarkoituksena on esitellä ja opastaa sen lukijaa Amazonin GameSparks-palvelun käytössä Unity-projektissa. Ohjeistus on laadittu opinnäytetyön liitteeksi, ja se on ensisijaisesti laadittu Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opiskelijoiden käyttöön. Ohjeistuksessa käydään läpi kuvallisin esimerkein GameSparks-palvelun ominaisuuksia, peliprojektin rekisteröintiä palveluun, sekä sen käyttöönottoa Unityn graafisessa editorissa. Oppaassa on esitelty pyyntöjen lähettäminen pelikoodista käsin, kuten käyttäjän rekisteröinti ja kirjautuminen, pisteiden tallentaminen tulostaulukkoon, leaderboard-tietojen haku GameSparksista Unity-projektiin, omien eventien luominen ja käyttäminen, sekä pelaajan tietojen haku peliprojektiin (kuten pelaajan saamat saavutukset). Palvelun ominaisuuksien esittelemistä varten loimme Unityssä yksinkertaisen peliprojektin, jossa otimme GameSparks-palvelun käyttöön. Peliprojektissa on tarkoitus esitellä palvelun ominaisuuksia, joten graafiseen puoleen tai pelattavuuteen ei ole tästä syystä juurikaan panostettu. Peliprojekti, joka palvelun käyttöönottoa varten luotiin, koostuu kolmesta Unity-scenestä. Ensimmäisessä Scenessä on esitelty käyttäjän kirjautuminen ja rekisteröityminen peliin. Toisessa Scenessä voidaan aloittaa uusi peli, hakea leaderboardin tietoja, tai pelaajan saamia saavutuksia tai kirjautua ulos. Kolmannessa Scenessä on itse peli, jossa pelaajan tavoitteena on kerätä mahdollisimman paljon pisteitä ennen maaliin pääsyä. Maaliin päästyään pelaajan pisteet lähetetään GameSparksiin luotuun tulostaulukkoon.

Unityn graafista editoria käytettiin oppaassa sen takia, että se on Jyväskylän ammattikorkeakoulun tietojenkäsittelyn opiskelijoilla eniten käytössä oleva pelinkehitysohjelma. Ohjelmointikielenä projektissa käytettiin samaisesta syystä C#-kieltä. Opas on kirjoitettu vaihe vaiheelta ja sisältää lukuisia havainnollistavia kuvia käyttöönoton selkeyttämiseksi ja helpottamiseksi. Peliprojekti on tehty kirjoitushetkellä uusimmalla käytössä olevalla Unityn versiolla, joka on versionumeroltaan 2018.3.5. Ennen GameSparksin backend-palveluiden käyttöönottoa peliprojektissa käydään läpi, kuinka peliprojekti rekisteröidään palveluun ja mitä ominaisuuksia peliprojektissa voidaan ottaa käyttöön. Palvelun kaikkien ominaisuuksien läpikäyminen olisi vienyt liikaa aikaa, ja

opasta varten luodun peliprojektin olisi pitänyt olla paljon laajempi. Näimme parhaaksi vaihtoehdoksi käydä oppaassa läpi tärkeimmät ominaisuudet Ticorporate-opiskelijan näkökulmasta. Esimerkiksi GameSparksin moninpeletoiminnot jätettiin tästä syystä pois oppaasta. Peliprojekti on ladattavissa osoitteesta <https://github.com/JPiippo/Testipeli>.

2 GameSparks-palvelun tärkeimmät ominaisuudet

Tässä kappaleessa on kuvattu GameSparksin mielestämme tärkeimmät ominaisuudet kohderyhmää ajatellen. GameSparks tarjoaa oppaassa käytyjen ominaisuuksien lisäksi myös useita muita ominaisuuksia, kuten esimerkiksi moninpeleille tarkoitettuja matchmaking-ominaisuuksia, pelaajien segmentointia, turnauksia, pelaajien välistä chat-palvelua, sekä pilvikoodia. (Social & Multiplayer, 2019).

GameSparksissa luoduilla Eventeillä voidaan määrittää mukautettuja tietorakenteita, jotka halutaan siirtää GameSparksin-alustaan LogEventRequest, sekä LogChallengeRequest API-kutsuilla (Events, n.d). Eventejä voidaan käyttää esimerkiksi lähettämään pelaajan pelissä saamia pisteitä GameSparksissa luotuun tulostaulukkoon. Eventit ovat siis tärkeässä osassa esimerkiksi tietojen tallentamisessa pilveen. Eventeihin voidaan myös lisätä itse luotua pilvikoodia, jonka avulla voidaan hallita peliin luotujen mukautettavien, konfiguroitavien komponenttien käyttäytymistä (Writing Cloud Code, n.d).

Oppaassa käydään läpi Unityssä itse luodun testiprojektin pohjalta mm.

- Kuinka projekti rekisteröidään ja kuinka siihen luodaan uusia ominaisuuksia
- Leaderboardin luominen ja pelaajan pelissä saamien pisteiden tallentaminen itse luodun Eventin avulla
- Leaderboard-tietojen hakeminen peliprojektiin ja tietojen käsittely
- Pelaajan rekisteröinti ja autentikointi kirjautumisikkunaa hyödyntäen
- Pelaajan tietojen hakeminen peliprojektiin, sekä pelaajan saamat saavutukset
- Virtuaalivaluutan luominen ja sen käyttö pelissä

3 GameSparks-portaali ja uuden peliprojektin rekisteröinti

Tässä luvussa käydään läpi GameSparks-portaali ja mitä se pitää sisällään. GameSparks-portaalin kautta voidaan lyhykäisyydessään muokata pelin ominaisuuksia, tarkastella analytiikkatietoja, lähettää pyyntöjä testiympäristössä, sekä ottaa käyttöön uusia ominaisuuksia. Rekisteröitymisen jälkeen voidaan valita käytettävä lisenssi-tyyppi, mutta oletuksena luodulla projektilla on palvelun ominaisuuksien testaamiseen tarkoitettu suunnitelma, joten palvelun ominaisuuksia voidaan lähteä testaamaan heti rekisteröitymisen jälkeen.

Ennen palvelun tuomista Unity-projektiin, täytyy GameSparksin portaalin kautta rekisteröityä ja luoda sitä kautta uusi peliprojekti. Peliprojektille asetetaan kuvaava nimi ja lyhyt kuvaus kyseessä olevasta projektista. "Game Settings"-kohdassa voidaan asettaa aika, kuinka monen minuutin kuluttua pelisessio päättyy automaattisesti, jos pelaaja katkaisee yhteyden. Alla kuva uuden peliprojektin lisäämisestä GameSparks-portaalissa. Uuden projektin rekisteröinti tapahtuu osoitteesta <https://portal2.gamesparks.net/login>.

Add New Game Testipeli ▾

1 — 2 — 3

Game Details Features & Integrations Geographical Setup

Use Stage 1 to enter a Name and Description for your new game and enable a session timeout for players who disconnect from your game. You'll be able to edit these after game creation from Game Overview. Click Next to proceed.

Game Details

Name Description

Game Settings

Automatically end player sessions minutes after they disconnect.

Next

Kuvio 1. Uuden peliprojektin luominen GameSparks-portaalissa.

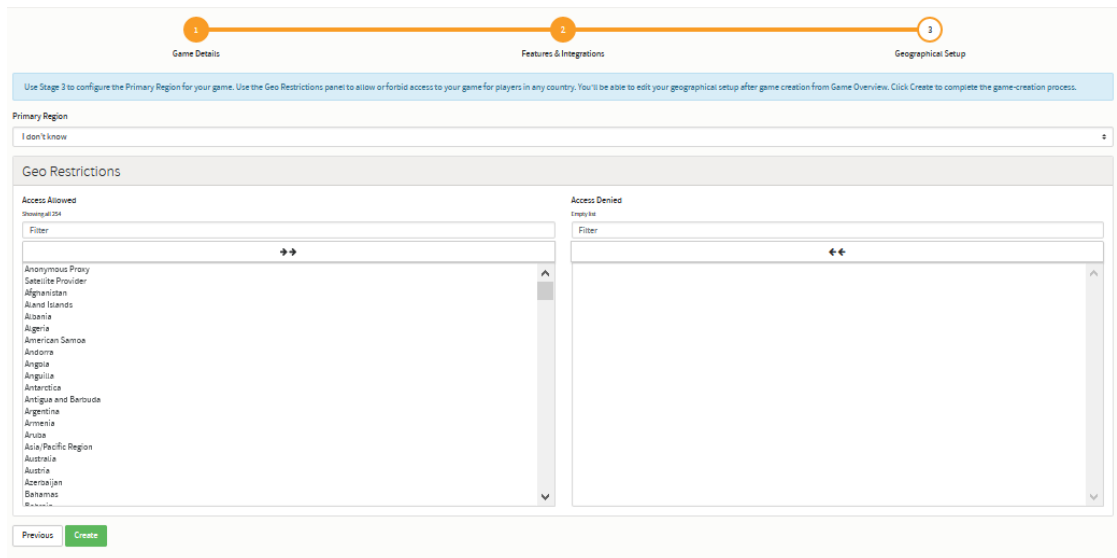
Perustietojen syöttämisen jälkeen voidaan määrittellä, mitä GameSparksin ominaisuuksia projektissa halutaan ottaa käyttöön. Vaihtoehtoja, jotka voidaan valita ovat eventit, tulostaulukot (leaderboards), tiimit tai pelaajaryhmät, virtuaalivaluutat, saavutukset (achievements), moninpeli, pelaajien välinen viestintä, ladattavat tiedostot (downloadables), muokattavat ominaisuudet, sekä oman pilvikoodin luominen. Valintoja voidaan myöhemmin muokata GameSparks-portaalin kautta. Samalla sivulla voidaan määrittellä halutut integraatiot eri palveluiden kanssa, kuten esimerkiksi Twitchin, Facebookin tai Steam-alustan kanssa. Alla olevassa kuvassa nähdään tarkemmin mitä integraatioita ja ominaisuuksia palvelussa voidaan ottaa käyttöön. Valintoja voidaan muokata myöhemmin GameSparks-portaalin kautta.

The screenshot displays two side-by-side configuration panels. The left panel, titled 'Platform Features', lists ten features with checkboxes, all of which are checked. The right panel, titled 'Integrations', lists various third-party services with checkboxes, all of which are unchecked.

Platform Features	Integrations
<input checked="" type="checkbox"/> Events Create User Events and use them to define custom data structures that you want to pass into the platform.	<input type="checkbox"/> Google
<input checked="" type="checkbox"/> Leaderboards Build traditional in-game Leaderboards to rank players or Leaderboards to control when Achievements are awarded or Challenges are won.	<input type="checkbox"/> Apple / iOS
<input checked="" type="checkbox"/> Teams Define player-based Teams and use them for team-based Leaderboards, team Achievements, and to build social features and cooperation into your game.	<input type="checkbox"/> Microsoft
<input checked="" type="checkbox"/> Virtual Goods Create Virtual Goods for any in-game assets that can be awarded, accumulated, or bought, including XP points and in-game currencies.	<input type="checkbox"/> Amazon
<input checked="" type="checkbox"/> Achievements Set up and manage Achievements for your game and use them to reward players when they trigger certain targets such as a score or particular rank.	<input type="checkbox"/> Facebook
<input checked="" type="checkbox"/> Multiplayer Set up Challenges to encourage your players to play with each other. Configure Matches to control how players are matched on customized player data.	<input type="checkbox"/> Twitter
<input checked="" type="checkbox"/> Messages Configure in-game messages sent to players through the use of socket-sent messages or push notifications.	<input type="checkbox"/> PSN
<input checked="" type="checkbox"/> Downloadables Create and manage binary data content in the platform.	<input type="checkbox"/> Steam
<input checked="" type="checkbox"/> Properties Properties are your own configuration objects you use within Cloud Code or expose to clients. Property Sets let you combine and segment Properties, delivering different configurations among players.	<input type="checkbox"/> Twitch
<input checked="" type="checkbox"/> Cloud Code Create custom logic to control the behavior of configurable components you build into your game.	<input type="checkbox"/> Kongregate
	<input type="checkbox"/> Viber
	<input type="checkbox"/> We Chat
	<input type="checkbox"/> QQ
	<input type="checkbox"/> Nintendo

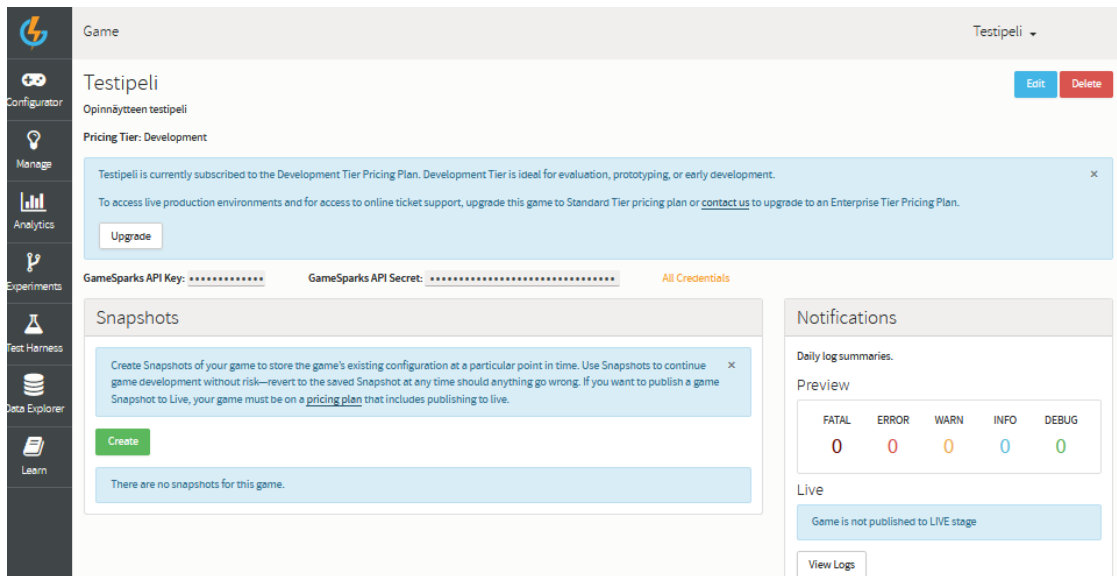
Kuvio 2. GameSparksissa valittavat ominaisuudet ja integraatiot.

Kun halutut ominaisuudet ja integraatiot on saatu valittua, tulee tämän jälkeen määrittellä mahdolliset maantieteelliset rajoitukset. Listasta voidaan valita alueet, joilla palvelun käyttö on estetty tai sallittu. Tämä estää GameSparksin toiminnan kielletyiltä alueilta yhdistäviltä pelaajilta, eikä palvelua voida näissä sijainneissa käyttää. Tämän vaiheen jälkeen voidaan rekisteröinti viedä loppuun, ja itse pelin ominaisuuksien luominen voidaan aloittaa.



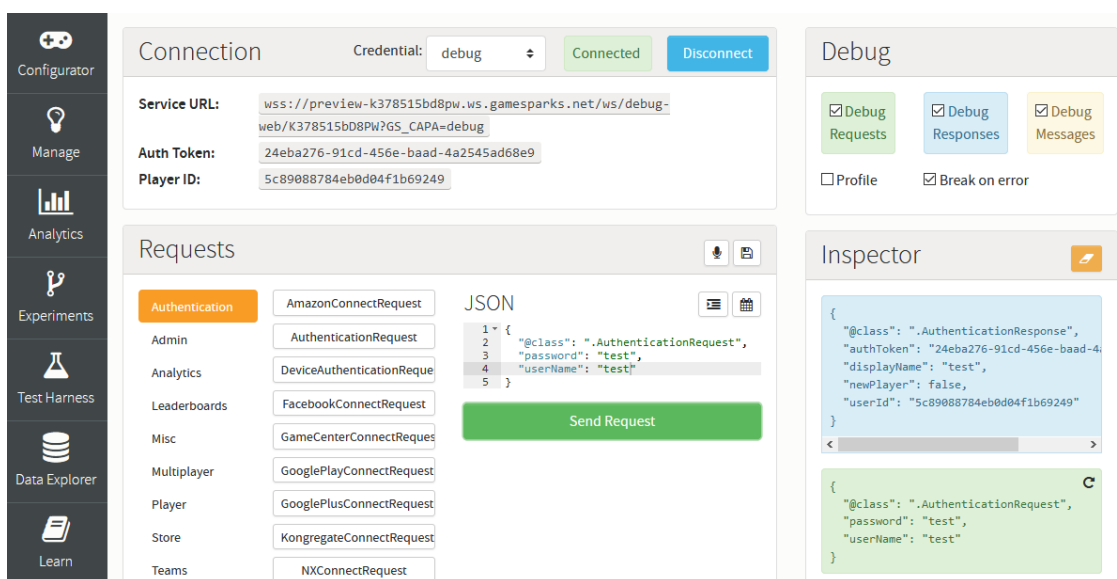
Kuvio 3. Maantieteelliset rajoitukset GameSparksissa.

Kun rekisteröinti on viety loppuun, saadaan API-key ja GameSparks API Secret, jotka lisätään myöhemmin Unity-peliprojektiin. Rekisteröinnin jälkeen portaalin kautta voidaan hallinnoida eri ominaisuuksia, luoda eventejä ja esimerkiksi tehdä kyselyitä testiympäristössä ilman että kosketaan pelikoodiin Unityn puolella. Alla olevassa kuvassa näkyy GameSparks-portaalin päänäkymä. Vasemmalla olevasta Configurator-valikosta voidaan hallinnoida pelin eri ominaisuuksia, sekä luoda esim. uusia eventejä, leaderboardoja, ja muita ominaisuuksia. Valikosta voidaan muokata rekisteröintivaiheessa määritellyjä ominaisuuksia ja halutessaan ottaa uusia ominaisuuksia käyttöön. Analytics-valikosta voidaan tarkastella API-kutsujen määrää erilaisten kriteerien perusteella.



Kuvio 4. GameSparks-portaalin etusivu palveluun rekisteröitymisen jälkeen.

Test Harness-valikon kautta päästään käsiksi testiympäristöön, jossa voidaan lähettää pyyntöjä GameSparsiin, ennen kuin sitä otetaan itse Unity-projektin puolella käyttöön. Test Harness on kätevä tapa testata esimerkiksi itse luotuja eventejä ja varmistaa että luodut ominaisuudet toimivat oikein. Pyyntöt lähetetään ja vastaanotetaan JSON-muodossa. Alla olevassa kuvassa on lähetetty AuthenticationRequest-kirjautumispyyntö luodulla testikäyttäjällä.



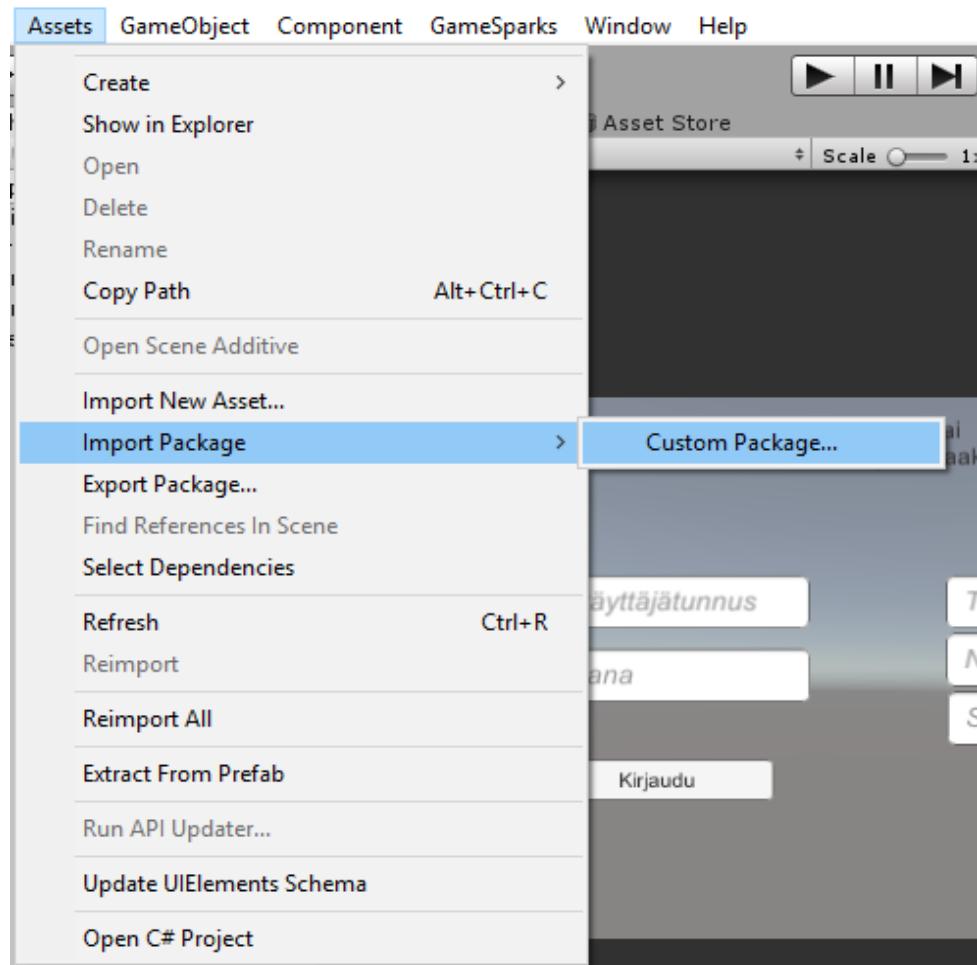
Kuvio 5. Kirjautumispyynnön lähettäminen GameSparksin testiympäristössä. Inspector-paneelissa nähdään saatu JSON-vastaus.

4 GameSparksin käyttöönotto Unity-projektissa

4.1 GameSparks-työkalujen tuominen Unity-projektiin ja alkuasetusten määrittäminen Unityssä

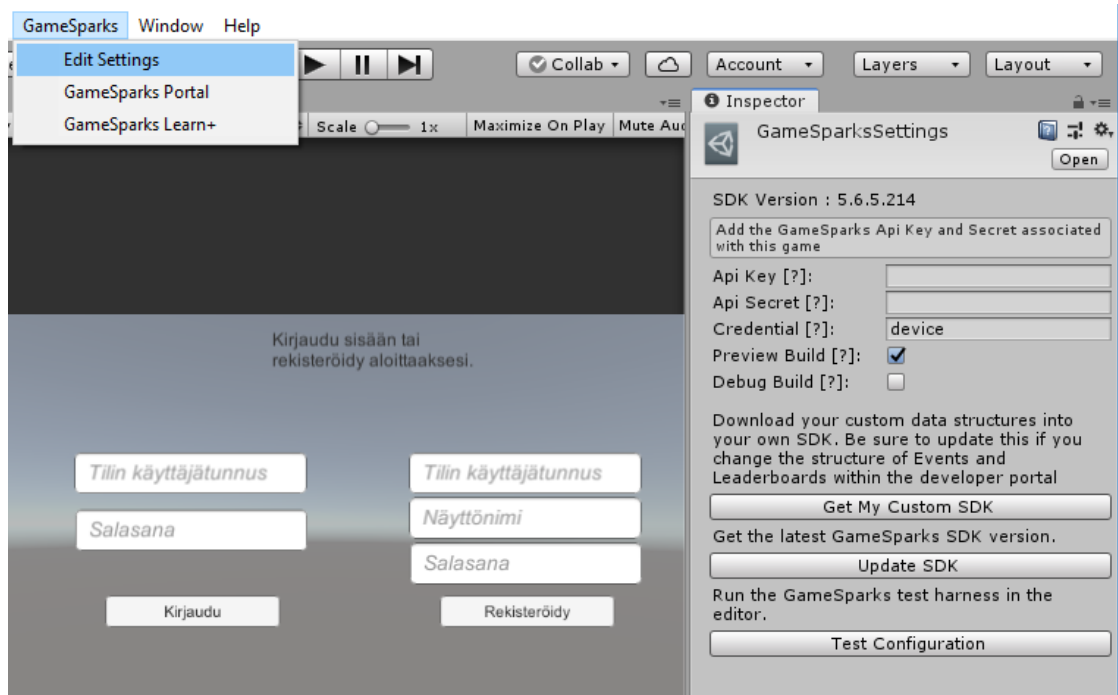
Tässä luvussa käydään läpi, kuinka GameSparksin työkalut tuodaan Unity-projektiin, ja kuinka alkuasetukset määritetään, jotta palvelu toimii normaalisti. Luvussa opastetaan myös, kuinka GameSparks-portaalin kautta voidaan luoda uusia eventejä, sekä tulostaulukoita. Kun toiminnallisuudet on saatu portaalin kautta luotua, opastetaan kuinka luotuja ominaisuuksia, voidaan käyttää pelikoodista käsin Unityn puolella. Kun SDK tuodaan projektiin, se luo GameSparks nimisen kansion projektikansioon, joka sisältää GameSparksin toiminnan kannalta oleelliset skriptit ja muut tiedostot.

1. Tuo GameSparks SDK Unity-projektiin joko lataamalla se Unityn Asset Storesta, tai GameSparksin verkkosivuilta osoitteesta <https://docs.gamesparks.com/sdk-center/unity.html>.
2. Kun paketti on ladattu, tuodaan se projektiin Assets-menusta kohdasta Import Package -> Custom Package. SDK:n tuomisen jälkeen Unityn yläpalkkiin ilmestyy GameSparks-menu, jonka kautta voidaan avata GameSparks-portaali, muokata asetuksia tai avata GameSparks Learn+, josta löytyy esimerkiksi palvelun käytön tutoriaaleja.



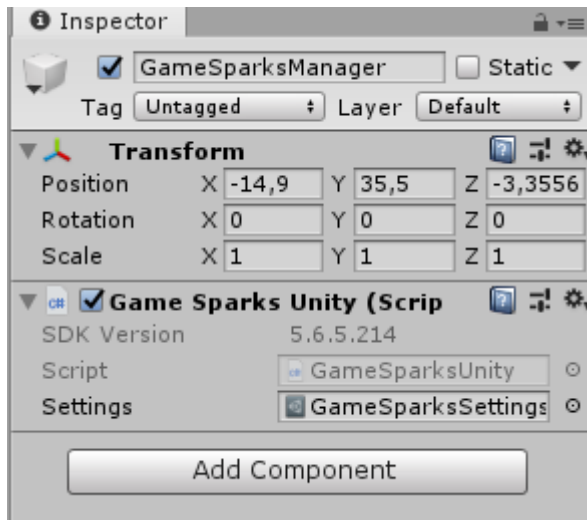
Kuvio 6. Custom Packagen lisääminen Unity-projektiin. Tätä kautta haetaan ladattu GameSparks SDK-projektiin.

- 3. Lisää rekisteröinnin yhteydessä saatu API-key ja API-secret Unity-projektissa GameSparksSettings-kohdassa. Tämä onnistuu Unityn yläpalkin GameSparks-valikon kautta navigoitaessa kohtaan "Edit Settings". Tiedoston nimi on Settings.asset.**



Kuvio 7. API Key:n ja API Secretin lisääminen Unity-projektissa.

4. Lisää pelin ensimmäiseen Sceneen GameSparksManager-niminen objekti. Objektiin tulee lisätä tämän jälkeen SDK:n mukana tullut GameSparksUnity-skripti, johon lisätään juuri konfiguroitu Settings.asset-tiedosto. Kyseessä on sama tiedosto, johon API Key ja API Secret lisättiin. GameSparksManager-objekti lisään ainoastaan peliprojektin ensimmäiseen Sceneen. GameSparks toimii tämän jälkeen pelissä normaalisti, vaikka Sceneä vaihdettaisiin.
5. Kun objekti on lisätty Sceneen, voidaan palvelua alkaa tämän jälkeen projektissa hyödyntämään. Palvelun toimivuutta voidaan testata Unityssä klikkaamalla GameSparksSettings-valikon kautta Test Configuration-painikkeesta.



Kuvio 8. Pelin ensimmäiseen Sceneen lisättävä GameSparksManager-objekti.

4.2 Pelin ominaisuuksien luominen GameSparks-portaalissa

Kuten aiemmin mainittiin, GameSparks-portaalin kautta voidaan luoda peliin haluttuja toiminnallisuksia, kuten tulostaulukoita ja saavutuksia. Tässä kappaleessa käydään tarkemmin läpi, kuinka perusominaisuuksia voidaan luoda portaalin konfiguraattorin kautta. Luodaan tulostaulukko, sekä event, jonka avulla voidaan pelikoodista lähettää pelaajan pisteet GameSparkiin luotuun tulostaulukkoon. Tämän jälkeen luodaan saavutus, joka lisätään kirjautuneen pelaajan tietoihin, kun pelaaja pääsee tulostaulukossa viiden parhaiten sijoittuneen pelaajan joukkoon. Tässä vaiheessa peliprojektin tulee olla rekisteröitynä GameSparkiin, jotta ominaisuuksia voidaan peliin lisätä. Luodaan ensimmäiseksi uusi event, sekä tulostaulukko GameSparksin konfiguraattorin kautta.

1. Luo ensin uusi event konfiguraattorilla. Luo event ennen tulostaulukkoa, koska sitä tarvitaan tulostaulukon luomisessa.
2. Valitse Configurator valikosta Events ja klikkaa Add-painiketta.
3. Eventille ja muille ominaisuuksille tulee määrittellä "Short Code"-tunniste, jotta siihen voidaan kohdistaa kutsuja pelikoodista ja portaalissa.

4. Event ei itsessään vielä tee mitään, vaan sille täytyy määritellä itse toiminnallisuudet lisäämällä siihen attribuutteja. Klikkaa eventin luomisen jälkeen Add-painikkeesta Attributes-paneelissa lisätäksesi uuden attribuutin. Lisätään tämän jälkeen vielä attribuutille aiempaan tapaan tunniste ja nimi. Attribuutti vaatii tämän lisäksi myös oletusarvon, sekä datan tallennustyyppin (Default Aggregation Type).
5. Asetetaan tässä esimerkissä oletusarvoksi 0, ja datan tallennustyyppi Maximum. Tämä mahdollistaa sen, että tulostaulukkoon tallennetaan aina aiempaa suurempi arvo, eli mikäli pelaaja saa vähemmän pisteitä kuin edellisellä kerralla, ei tallenneta pienempää pistemäärää suuremman tuloksen päälle. Alla kuva pelaajan pisteitä tallentavan eventin luomisesta ja attribuutin lisäämisestä.

Game / Configurator / Events / Pelaaja pääsee leaderboardille Testipeli

PLAYER_TO_LEAD...

Short Code

Name

Description

Save and Close Save Close

Attributes Add

Add the Attributes you want to your Event below. Attributes allow you to pass data in as part of the Event.

Short Code	Name	Data Type	Default Value	Default Aggregation Type	Actions
PLAYER_SCORES	Pelaajan pisteet tall	Number	0	Maximum	

Delete

Kuvio 9. Event, joka tallentaa pelaajan saamat pisteet tulostaulukkoon.

1. Luodaan eventin luomisen jälkeen vielä tulostaulukko, johon pisteet tallennetaan. Valitse Configurator-valikosta Leaderboards.
2. Klikkaa Add-painikkeesta luodaksesi uuden tulostaulukon. Leaderboard vaatii "Short Code"-kohtaan tunnisteen, jotta siihen voidaan kohdistaa kutsuja pelikoodissa ja portaalissa.
3. Running Total-kohtaan valitaan aiemmin luotu event, jonka kautta tiedot voidaan tallentaa. Collector-kohtaan valitaan eventiin lisätty attribuutti, joka tallentaa aina aiempaa suuremman arvon. Alla havainnollistava kuva

uuden tulostaulukon luomisesta. Näiden toimenpiteiden jälkeen tulostaulukko on valmis käytettäväksi. Tulostaulukon ja eventin toimintaa pelikoodista käsin Unityn puolella käydään läpi oppaassa vielä myöhemmin.

The screenshot shows the Unity Leaderboards configuration interface. The left panel is titled 'SCORE_BOARD' and contains the following fields:

- Short Code: SCORE_BOARD
- Name: Leaderboard
- Description: Leaderboard jossa näytetään eniten pisteitä
- Team: (None)
- Update Frequency: Real Time
- Reset Frequency: Never
- Property Set: (None)
- High Score Notifications: ON
- Social Notifications: OFF
- Top N Notifications: OFF
- Top N Threshold: (empty)

The right panel is titled 'Fields' and contains a table of fields to be added to the leaderboard:


Running Total	Collector	Filter Type	Filter Value	Sort	Group	Actions
Pelaaja pääsee leaderboardille	PLAYER_TO_LEADERBOARD.PLAYER_SCORES	*		DESC	-	Delete

Kuvio 10. Uuden tulostaulukon luominen portaalissa.


Luodaan eventin ja tulostaulukon luomisen jälkeen vielä saavutus, jonka pelaaja saa, kun pääsee tulostaulukossa viiden parhaan pelaajan joukkoon. GameSparksissa on useita valmiita tapoja avata saavutus pelaajan tietoihin, kuten esimerkiksi tulostaulukon sijoittumisen johdosta. Alla olevassa esimerkissä käydään läpi, kuinka uusi saavutus luodaan portaalissa.


1. Mene kohtaan Configurator -> Achievements.
2. Luodaan Add-painikkeella uusi saavutus. Saavutus vaatii samaan tapaan tunnistetiedon, nimen ja kuvauksen syöttämisen. Trigger Data-kohdassa voidaan valita, kuinka pelaaja saa kyseisen saavutuksen.


3. Valitaan tässä esimerkissä Trigger Data-kohtaan New Global Rank. Filter Type ja Filter Value määrittävät kuinka Achievement avataan. Valitsemalla New Global Rank ≥ 5 , voidaan esimerkiksi määrittää, että pelaajaa saa saavutuksen, kun pääsee viiden parhaan joukkoon tulostaulukossa. Saavutuksen ohessa pelaaja voidaan palkita halutessaan virtuaalivaluutalla. Ompaassa demonstroidaan myöhemmin, kuinka pelaajan avaamat saavutukset voidaan hakea GameSparksista peliprojektiin. Alla olevassa kuvassa esitetty uuden saavutuksen luominen portaalissa.


Game / Configurator / Achievements / Top 5 Leaderboardilla Testipeli 


Leaderboard

Short Code 


Name 


Description 

Virtual Good Award


Leaderboard


Repeatable

 OFF

Property Set




 

Currency Awards

Triggers

Add the Triggers you want to your Achievement below. Triggers are required for non-scripted Achievements.

Trigger Data	Filter Type	Filter Value	Actions
New Global Rank 	\geq 	5	

Kuvio 11. Uuden saavutuksen luominen.

4.3 Käyttäjän tunnistaminen ja rekisteröinti, kirjautumisikkunan luominen Unityssä

Tässä kappaleessa käydään läpi, kuinka käyttäjä kirjautuu ja rekisteröityy palveluun Unity-projektissa. Kappaleessa luodaan pelin ensimmäinen Scene, johon sisältyy sisäänkirjautumis- sekä rekisteröitymislomakkeet. Käyttäjä rekisteröityy palveluun antamalla käyttäjätunnuksen ja salasanan, jotka lähetetään `RegistrationRequest`-pyynnön mukana `GameSparks`in Unityn `onClick()`-eventiä hyödyntäen. `GameSparks` tarjoaa kolmea eri vaihtoehtoa sisäänkirjautumiselle, mutta oppaassa on käytetty perinteistä sisäänkirjautumista salasanalla ja käyttäjätunnuksella. Sisäänkirjautumisessa käytetään `AuthenticationRequest`-pyyntöä, joka lähetetään käyttäjän painaessa sisäänkirjautumispainiketta. Sisäänkirjautuminen pysyy voimassa, vaikka Sceneä vaihdetaan. Vaadittavia kenttiä rekisteröitymisessä ovat käyttäjänimi, näyttönimi, sekä salasana. Alla olevassa kuvassa nähtävissä luotava sisäänkirjautumisruutu, sekä luodut skriptit.



Kuvio 12. Sisäänkirjautumis- ja rekisteröintiruutu.

Alla kuvatut skriptit ovat käytössä kirjautumis- sekä rekisteröitymispainikkeissa. Skripti suoritetaan Unityyn sisäänrakennetulla `onClick()`-eventillä, joka suorittaa koodin kun painiketta klikataan. Käyttäjän syöttämät tiedot tallennetaan muuttujiin,

jotka lähetetään GameSparksin RegistrationRequest()-pyynnön mukana GameSparksiin. GameSparksilta saadaan vastaus ja mikäli vastauksessa ilmenee virheitä, ilmoitetaan siitä Unityn lokiin.

```

///Tätä skriptiä käytetään rekisteröitymispainikkeessa. Kun painiketta klikataan
///suoritetaan RegisterPlayer() ja lähetetään GameSparksille RegistrationRequest-pyyntö.
///Mikäli virheitä ei tapahdu, suoritetaan rekisteröinti loppuun ja käyttäjä voi kirjautua heti palveluun. Muussa tapauksessa
///lokiin tulostetaan virheilmoitus.
0 references
public class RegisterPlayerButton : MonoBehaviour
{
    1 reference | 1 reference | 1 reference
    public Text displayNameInput, userNameInput, passwordInput;

    0 references
    public void RegisterPlayer()
    {
        Debug.Log("Tarkastetaan syötettyjä tietoja...");
        new GameSparks.Api.Requests.RegistrationRequest()
            .SetDisplayName(displayNameInput.text)
            .SetUserName(userNameInput.text)
            .SetPassword(passwordInput.text)
            .Send((vastaus =>
            {
                if (!vastaus.HasErrors)
                {
                    Debug.Log("Uusi pelaaja rekisteröity tunnuksella: " + vastaus.DisplayName);
                }
                else
                {
                    Debug.Log("Rekisteröitymisessä tapahtui virhe.");
                }
            }));
    }
}

```

Kuvio 13. Uuden pelaajan rekisteröitymispyyntö.

```

6  ///Tämä skripti on käytössä kirjautumis-painikkeessa. AuthorizePlayer lähettää
7  ///GameSparksille AuthenticationRequest-pyynnön, jossa se tarkastaa käyttäjän syöttämät
8  ///kirjautumistiedot. Mikäli virheitä ei ole, siirrytään seuraavaan Sceneen ja kirjautuminen pysyy voimassa.
9  ///
10
11 0 references
12 public class AuthorizePlayerButton : MonoBehaviour
13 {
14     0 references | 1 reference | 1 reference
15     public Text displayNameInput, userNameInput, passwordInput;
16
17     0 references
18     public void AuthorizePlayer()
19     {
20         Debug.Log("GameSparks tarkastaa syötettyjä tietoja...");
21         new GameSparks.Api.Requests.AuthenticationRequest()
22             .SetUserName(userNameInput.text)
23             .SetPassword(passwordInput.text)
24             .Send((vastaus) =>
25             {
26                 ///Jos GameSparkista saadussa vastauksessa ei ole virheitä ja kirjautuminen onnistuu
27                 ///siirrytään seuraavaan sceneen.
28                 if (!vastaus.HasErrors)
29                 {
30                     Debug.Log("Pelaaja tunnistettu... Tunnus: " + vastaus.DisplayName);
31                 }
32                 ///Kirjautumisen jälkeen ladataan seuraava scene
33                 Application.LoadLevel("MenuScene");
34             }
35         );
36
37         ///Mikäli AuthenticationRequest epäonnistuu, ilmoitetaan lokiin kirjautumisen epäonnistumisesta
38         else
39         {
40             Debug.Log("Tapahtui virhe, tarkista kirjautumistiedot.");
41         }
42     }
43 }
44
45 });
46
47 }

```

Kuvio 14. Pelaajan sisäänkirjautuminen peliin AuthenticationRequestia hyödyntäen.

Tehdään yllä olevien rekisteröitymis- sekä sisäänkirjautumispainikkeiden lisäksi vielä skripti uloskirjautumiselle. Uloskirjautumisessa käytetään GameSparksin GS.Disconnect-pyyntöä, joka on kuvattu alla olevassa kuvassa. Alla olevassa esimerkissä GS.Disconnect-pyyntö on lisätty uloskirjautumispainikkeeseen, jota painamalla käyttäjän sessio lopetetaan ja ladataan ensimmäinen kirjautumiseen tarkoitettu Scene uudestaan.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using GameSparks.Api;
5  using GameSparks.Api.Requests;
6  using GameSparks.Api.Responses;
7  using GameSparks.Core;
8
9  ///Tämä skripti on käytössä uloskirjautuspainikkeessa. GS.Disconnect kirjaa käyttäjän ulos
10 ///Uloskirjautumisen jälkeen pyyntöjä ei voida enää lähettää ja käyttäjän tulee kirjautua
11 ///sisään uudestaan.
12
13 0 references
14 public class LogOutButton : MonoBehaviour
15 {
16     0 references
17     public void Logout()
18     {
19         GS.Disconnect();
20         Debug.Log("Käyttäjä kirjattu ulos");
21     }
22 }
23
24
25
26
27
28 }
29

```

Kuvio 15. Käyttäjän uloskirjaaminen GameSparks-palvelusta.

4.4 Pelaajan pisteiden tallennus Leaderboardille

Kun pelaaja on rekisteröitynyt ja päässyt kirjautumaan palveluun, voi pelaaja aloittaa uuden pelin. Tässä kappaleessa käydään läpi, kuinka pelaajan saamat pisteet lähetetään GameSparksin tulostaulukkoon, joka luotiin aiemmin. Alla on kuvattu peliprojektimme pelihahmon pisteiden keräys, sekä kuinka saadut pisteet lähetetään lopuksi tulostaulukkoon.

1. Ensin on luotu pelaajan pisteiden keräämiselle skripti. Jokaisesta kerätystä pisteestä lisätään pelaajan pistetilille 1 piste lisää. Alla olevassa kuvassa esimerkki.

```
//Alla olevassa skriptissä on kuvattu pelaajan pisteiden kerääminen. Pelaajan osuessa
//timantti-nimiseen peliobjektiin, saa pelaaja yhden pisteen lisää.
if(col.gameObject.name == "timantti")
{
    score = score + 1;
    Destroy(col.gameObject);
    scoreCounter.text = "Score: " + score.ToString();
}
```

Kuvio 16. Esimerkki, jossa on kuvattu pelaajan pisteiden kerääminen pelin aikana.

- Seuraavaksi on esitetty, kuinka pelaajan keräämät pisteet voidaan lopuksi lähettää aiemmin luotuun tulostaulukoon käyttäen aiemmin portaalissa luotua tulostaulukkoa ja eventtiä. Alla olevassa esimerkissä skripti suoritetaan pelaajan päästessä maaliin.

```
if(col.gameObject.name == "maali")
{
    Destroy(col.gameObject);

    Debug.Log("Lähetetään pisteitä GameSparksin Leaderboardille...");
    new GameSparks.Api.Requests.LogEventRequest()
        .SetEventKey("PLAYER_TO_LEADERBOARD")
        .SetEventAttribute("PLAYER_SCORES", score)
        .Send((vastaus) =>
        {
            if (!vastaus.HasErrors)
            {
                Debug.Log("Pisteet tallennettu. Lopulliset pisteet: " +score);
            }

            else
            {
                Debug.Log("Tapahtui virhe pisteiden tallennuksessa.");
            }
        });
}
```

Kuvio 17. Esimerkki skriptistä, joka lähettää pelaajan saamat pisteet tulostaulukoon käyttäen aiemmin luotua eventtiä.

Pisteet lähetetään käyttäen LogEventRequest-pyyntöä. Pyyntössä SetEventKey-kohtaan syötetään aiemmin luodun eventin tunniste (Short Code), jolla pyyntö voidaan kohdistaa oikeaan eventtiin. SetEventAttribute-kohtaan syötetään käytettävän eventin attribuutti, ja sen mukana lähetetään esimerkissä score-muuttujan arvo, johon pelaajan keräämät pisteet on pelin aikana tallennettu kumulatiivisesti. Yllä olevassa esimerkissä käytetään aiemmin oppaan aikana luotua eventtiä ja attribuuttia, joka tallentaa aina entistä suuremman arvon tulostaulukkoon. Jos pelaaja saa vähemmän pisteitä kuin edellisessä pelissä, pisteitä ei tallenneta tässä tapauksessa tulostaulukkoon ollenkaan.

4.5 Pelaajan tietojen noutaminen ja käsittely pelikoodissa, Leaderboard-datan hakeminen peliprojektiin

Tässä luvussa tarkastellaan esimerkkien avulla, kuinka Unity-projektissa voidaan hakea GameSparksiin tallennettua pelaajan dataa, sekä kuinka aiemmin luodusta tulostaulukosta voidaan hakea parhaiten sijoittuneiden pelaajien tietoja peliprojektiin. Kun tiedot on saatu palvelusta haettua, voidaan sillä peliprojektissa tehdä mitä halutaan. Tulostaulukosta voidaan esimerkiksi hakea kymmenen parhaiten sijoittunutta pelaajaa ja esittää tiedot graafisesti käyttäjälle halutulla tavalla. Pelaaja voi esimerkiksi hakea palvelusta omat saavutuksensa, ja tämän jälkeen pelikoodissa voidaan dataa käsitellä halutulla tavalla. Alla olevassa esimerkissä haetaan listana pelaajan avaat saavutukset ja tulostetaan ne Unityn lokiin. Kyseisessä skriptissä käytetään AccountDetailsRequest-pyyntöä. Tulostaulukon datan hakemisessa käytetään LeaderBoardDataRequest-pyyntöä.

```

0 references
public class GetAchievements : MonoBehaviour
{
    // Tämä skripti demonstroi, kuinka GameSparksista voidaan hakea pelaajan tietoja, kuten saatuja
    //Achievementtejä.
    0 references
    public void GetAchievementData()
    {
        ///Luodaan valmiiksi uusi lista, johon achievementit haetaan.
        List<string> achievementsLista = new List<string>();

        new GameSparks.Api.Requests.AccountDetailsRequest().Send((vastaus) => {
            if (!vastaus.HasErrors) {
                Debug.Log("Noudetaan käyttäjätilin tietoja...");
                string pName = vastaus.DisplayName; // Haetaan näyttönimi
                Debug.Log("Kirjautuneen pelaajan nimi: " + pName);
                achievementsLista = vastaus.Achievements; // Haetaan listana saadut achievementit.

                foreach(string s in achievementsLista) {
                    Debug.Log("Tililläsi on saatu achievement: " + s);
                }
            }
            else {
                Debug.Log("Tapahtui virhe tietojen haussa...");
            }
        });
    }
}

```

Kuvio 18. Pelaajan saavutusten hakeminen GameSparksista peliprojektiin.

Testikäyttäjällä, jonka pelitilillä on avattu aiemmin luotu saavutus, tulostuu yllä olevan kuvan skriptillä Unityn lokiin saatujen saavutusten nimet. Kuten aiemmin mainittiin, tässä vaiheessa saadulla datalla voidaan tehdä mitä halutaan, ja tiedot voi halutessaan esitellä käyttäjälle graafisesti tai muulla haluamallaan tavalla. Alla olevassa kuvassa esimerkki testikäyttäjällä haetusta datasta.

```

[00:24:14] Noudetaan käyttäjätilin tietoja...
UnityEngine.Debug:Log(Object)
[00:24:14] Kirjautuneen pelaajan nimi: test1
UnityEngine.Debug:Log(Object)
[00:24:14] Tililläsi on saatu achievement: SCORE_ACHIEVEMENT
UnityEngine.Debug:Log(Object)

```

Kuvio 19. Käyttäjän saavutusten hakeminen.

Alla olevassa esimerkissä demonstroidaan, kuinka GameSparksissa luodusta tulostaulukosta voidaan hakea sinne tallennettua dataa. Tähän käytetään LeaderBoardData-Request-pyyntöä. Skripti hakee tulostaulukosta kymmenen parhaiten sijoittunutta pelaajaa, jonka jälkeen silmukka käy läpi haetun datan ja tulostaa Unityn lokiin pelaajan sijoituksen, käyttäjänimen ja pelaajan saamat pisteet. SetEntryCount määrittää,

kuinka monta parhaiten sijoittunutta pelaajaa tulostaulukosta haetaan. SetLeaderBoardShortCode määrittää, mistä taulukosta dataa haetaan. Tässä tapauksessa syötetään aiemmin luodun tulostaulukon tunniste. Vastaus saadaan JSON-muodossa, ja siitä voidaan erotella haluttu data.

```
{
    new GameSparks.Api.Requests.LeaderboardDataRequest()
        .SetEntryCount(10)
        .SetLeaderboardShortCode("SCORE_BOARD")
        .Send((vastaus) =>
        {
            if(!vastaus.HasErrors)
            {
                //Loop käy läpi jokaisen elementin vastaanotetusta datasta. Nämä ovat tyyppiä LeaderboardData.
                foreach(GameSparks.Api.Responses.LeaderboardDataResponse_LeaderboardData lb in vastaus.Data)
                {
                    int pRank = int.Parse(lb.Rank.ToString()); //Haetaan pelaajan sijoitus saadusta vastauksesta
                    string pName = lb.UserName; //Haetaan pelaajan nimi saadusta vastauksesta
                    string pScore = lb.JSONData["PLAYER_SCORES"].ToString(); //PLAYER_SCORES = pelaajan saamat pisteet.

                    //Alla printataan vastaanotettu data Unityn lokiin.
                    string outPut = "Rank:"+pRank+"\n "+"Name: "+pName+" "+"Score: "+pScore;
                    Debug.Log(outPut);
                }
            }
        });
}
```

Kuvio 20. Leaderboard-tietojen haku GameSparksista.

4.6 Virtuaalivaluutan käyttö peliprojektissa pilvikoodia hyödyntäen

Tässä kappaleessa on luotu uusi virtuaalivaluutta GameSparks portaalissa, sekä luotu sille Event, joka tallentaa pelaajan tietoihin pelaajan saaman virtuaalivaluutan määrän. Alla on lisäksi esitetty esimerkki siitä, kuinka Eventiä ja sen attribuuttia voidaan käyttää pelikoodista käsin tallentamaan saadun virtuaalivaluutan summan pelaajan pelitilille. Attribuutin Default Aggregation Typeksi valitaan Sum, jolloin joka kerta kun pelaaja saa virtuaalivaluutaa lisätään se yhteen aiemman summan kanssa. Itse luotujen Eventien käytössä käytetään aiempien esimerkkien mukaisesti LogEventRequestia, jolla kutsut kohdistetaan kyseiseen Eventiin. Jotta valuutaa voidaan lisätä pelaajan tilille, tulee käyttää GameSparksin pilvikoodia. Pilvikoodissa pystytään mää-

rittelemään esimerkiksi erilaisia toimintoja valitun Eventin aktivoituessa. Alla havainnollistava kuva portaalissa luodusta Eventistä, virtuaalivaluutasta, sekä Eventin käytöstä pelikoodissa.

The screenshot shows the GameSparks portal interface. On the left is a sidebar with navigation options: Configurator, Manage, Analytics, Experiments, Test Harness, and Data Explorer. The main area is divided into three panels:

- VALUUTTA (Segment):** Configuration for the virtual currency. Fields include Short Code (VALUUTTA), Name (Virtuaalivaluutta), Description (Valuutta, jota pelaajat voivat saada pelissä), and Sign-up Bonus (100). Buttons: Save and Close, Save, Close.
- ADD_VALUUTTA:** Configuration for the event. Fields include Short Code (ADD_VALUUTTA), Name (Anna virtuaalivaluuttaa), and Description (pelaajalle virtuaalivaluuttaa). Buttons: Save and Close, Save, Close.
- Attributes:** A table for defining event attributes. It includes a table with columns: Short Code, Name, Data Type, Default Value, Default Aggregation Type, and Actions. One attribute is listed: ADD_VL, :pelaajalle rahaa, Numi, 0, Sum.



```

new GameSparks.Api.Requests.LogEventRequest()
  .SetEventKey("ADD_VALUUTTA")
  .SetEventAttribute("ADD_VL", 1)
  .Send((vastaus) =>
  {
    if (!vastaus.HasErrors)
    {
      Destroy(col.gameObject);
      Debug.Log("Pelaajalle on nyt annettu +1 virtuaalivaluuttaa");
    }
  }

```

Kuvio 21. Uuden virtuaalivaluutan luonti, Eventin ja sen attribuutin luonti, sekä sen käyttö pelikoodissa LogEventRequestin avulla.

Yllä oleva Event ja sen attribuutti ei vielä tee mitään, kun Eventiä käytetään koodissa. Jotta pelaajan tietoihin voidaan tallentaa lisää valuuttaa, tulee Eventin aktivoituessa ajaa pilvessä skripti, joka mahdollistaa valuutan tallentamisen pelaajan tilille. Tämä tehdään portaalin kautta kohdasta Configurator -> Cloud Code. Tätä kautta voidaan valita haluttu Event, ja kirjoittaa avautuvassa tekstieditorissa suoritettava koodi. Alla kuvattu valuutan lisäyksessä käytetty lyhyt skripti, joka mahdollistaa valuutan antamisen pelaajalle. `Spark.getPlayer()` hakee aktiivisen pelaajan, ja `ADD_VL` on aiemmin portaalissa luodun eventin attribuutti, joka tallentaa tiedot GameSparksiin. Pelaajan tiedot voidaan hakea GameSparksista `AccountDetailsRequestia` käyttämällä, samalla tavalla kuin pelaajan saamat saavutukset.

```
ADD_VALUUTTA ✕  
1 // =====  
2 //  
3 // Cloud Code for ADD_VALUUTTA, write your code here to customize the GameSparks platform.  
4 //  
5 // For details of the GameSparks Cloud Code API see https://docs.gamesparks.com/  
6 //  
7 // =====  
8 Spark.getPlayer().credit1(Spark.getData().ADD_VL);
```

Kuvio 22. Esimerkki pilvessä ajettavasta koodista.