



Nikolas Tukiainen

Tietokilpailupeli kaupalliseen tarkoitukseen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

15.5.2022

Tiivistelmä

Tekijä: Nikolas Tukiainen
Otsikko: Tietokilpailupeli kaupalliseen tarkoitukseen
Sivumäärä: 38 sivua
Aika: 15.5.2022

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Pelisovellukset
Ohjaajat: Lehtori Antti Laiho
Lehtori Miikka Mäki-Uuro

Insinööriyön tavoitteena oli suunnitella ja toteuttaa tilaajalle kaupalliseen tarkoitukseen sopiva tietokilpailupeli Android-, iOS- ja verkkoselainalustoille hyödyntäen Unity-pelimoottoria ja MariaDB-relaatiotietokantajärjestelmää.

Työn alussa tehtiin taustatutkimusta videopeleistä ja tarkemmin tietokilpailupeleistä osana pelialaa, niiden markkinoita ja yleisimpiä ansaintamalleja. Mobiilipelit ovat lyhyessä ajassa kehittyneet reilusti suurimmaksi pelialan tuottajaksi mobiililaitteiden kehittyttyä ja isojen pelinkehittäjien alettua kehittää mobiilipelejä. Mobiilipeleissä ansaintamalleja on monia, mutta suosituin on ehdottomasti mainoksiin perustuva ansaintamalli, jossa peli on muuten ilmainen.

Työssä tutkittiin myös kahta tärkeintä projektissa käytettyä ohjelmistoa, jotka olivat Unity ja MariaDB. Unity on suosituin, tehokas ja ilmainen pelimoottori, ja se otettiin projektiin käyttöön, koska se antaa laajat mahdollisuudet pelinkehitykseen monille eri alustoille, mitä projektissa tarvittiin. MariaDB tarjoaa halvan, hyvän ja helppokäyttöisen relaatiotietokantajärjestelmän, joka toimii helposti Unityn kanssa.

Pelisovelluksen teknisessä toteutuksessa suunniteltiin ja ohjelmoitiin käyttöliittymä-pohjainen ja tilakonetta ja kysymystietokantaa hyödyntävä tietokilpailupeli kolmelle eri alustalle. Pelisovellukseen toteutettiin myös pelin eri osien tallentaminen ja lataaminen kolmella tavalla ja mainospohjainen ansaintamalli. Työn suurin ongelmakohta oli etäpalvelimella olevan tietokannan saaminen yhteyteen Unityn kanssa. Toinen iso ongelma pelisovelluksessa oli tilakoneen toiminta, koska siinä on todella paljon eri osia ja testaamista tarvittiin paljon. Ongelmat saatiin ratkaistua paljolla testaamisella.

Insinööriyön lopputuloksena saatiin aikaiseksi mobiilialustalla ja verkkoselaimessa toimiva tietokilpailupeli ja toimiva kysymystietokanta. Tietokilpailupelin kehitys jatkuu pääasiassa viimeistelyyn ja uusien ominaisuuksien tuomiseen peliin.

Avainsanat: tietokilpailupeli, tietokilpailu, Unity, MariaDB

Abstract

Author: Nikolas Tukiainen
Title: Quiz game for commercial purposes
Number of Pages: 38 pages
Date: 15 May 2022

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology
Professional Major: Game Applications
Supervisors: Antti Laiho, Senior Lecturer
Miikka Mäki-Uuro, Senior Lecturer

The aim of this thesis was to design and make a quiz game for Android, iOS and web browser suitable for commercial purpose for the customer making use of Unity game engine and MariaDB relation database management system.

At the beginning of the thesis, background research on video games and more specifically on quiz games as part of the gaming industry, their markets and the most common earning model was conducted. In a short time, mobile games have become by far the most profitable games in the gaming industry because mobile devices have evolved, and bigger game studios have started to make mobile games. With mobile games ways to produce income are many but the most popular way to go is ad-based earning model with a free base version of the game.

The thesis also examines the two most important software used in the project, which are Unity and MariaDB. Unity is the most popular, powerful and free game engine and it was used in the project because it gives extensive opportunities for game development for many different platforms, which was needed for the project. MariaDB offers a cheap, good and easy to use relation database management system, which works easily with Unity.

In the technical implementation of the game application, an user interface-based quiz game utilizing a state machine and a database was designed and programmed for three different platforms. An implementation of saving and loading different parts of the game with three different ways and an ad-based earnings model was also made to the game. The biggest problem in the project was getting the database on the remote server connected to Unity. Another big problem in the game application was the functionality of the state machine because it has a lot of different parts, and a lot of testing was needed. All the problems got solved with a lot of testing.

As the end result of the thesis was a working quiz game for mobile and web browser platforms and a working question database. The development of the quiz game continues mostly with finishing touches and to bring new features to the game.

Keywords: quiz game, quiz, Unity, MariaDB

Sisällys

Lyhenteet

1	Johdanto	1
2	Tietokilpailupelit liiketoimintana	2
2.1	Tietokilpailupelit osana pelialaa	2
2.2	Pelimarkkinat	5
2.3	Ansaintamallit peleissä	6
3	Työssä käytetyt tekniikat ja ohjelmistot	11
3.1	Unity-pelimoottori	11
3.2	MariaDB-relaatiotietokantajärjestelmä	13
4	Tietokilpailu-pelisovellus	15
4.1	Konsepti	15
4.2	Käyttöliittymän suunnittelu ja näkymät	16
4.3	Pelin toiminta	21
4.4	Tallentaminen ja lataaminen	24
4.5	Monetisaatio	28
4.6	Alustojen erot	32
4.7	Tietokanta	33
5	Pohdinta	35
5.1	Tulokset ja ongelmakohdat	35
5.2	Jatkokehitys	36
6	Yhteenveto	37
	Lähteet	39

Lyhenteet ja käsitteet

Unity: Unity on pelimoottori.

MariaDB: MariaDB on relaatiotietokantajärjestelmä.

eCPM: Effective cost per mille. eCPM on hinta-arvio siitä, kuinka paljon tuloa mainos tuottaa jokaista tuhatta mainoksen näyttökertaa kohden.

GNU GPL: GNU General Public License on vapaiden ohjelmistojen julkaisemiseen tarkoitettu lisenssi, joka antaa kaikille oikeuden käyttää, kopioida, muokata ja jakaa edelleen ohjelmia ja niiden lähdekoodeja.

KISS: Keep it simple, stupid. Suunnitteluperiaate, jossa pyritään yksinkertaisuuteen ja välttämään kaikkea tarpeetonta monimutkaisuutta.

Kohtaus: Kohtaus (engl. scene) on tiedosto, joka sisältää kokoelman objekteja, joita tyypillisesti käytetään yhtenä elementtinä lataukseen tai purkuun pelin suorittamisen aikana.

Objekti: Objekti on esine tai asia pelissä.

1 Johdanto

Insinööriyön tavoitteena oli suunnitella ja toteuttaa tilaajalle tietokilpailupeli Android-, iOS- ja verkkoselainalustoille. Tietokilpailupeli toteutettiin Unity-pelimoottorilla hyödyntäen samalla MariaDB:n tietokantapalveluita. Insinööriyön tilaaja oli suomalainen osakeyhtiö Pressmen Finland Oy. Se on erikoistunut tietokilpailujen ja tietokilpailukysymysten tekemiseen ja tuottamiseen. Yhtiö on tehnyt vuoden 2021 loppuun mennessä yli 155 000 tietokilpailukysymystä muun muassa sanoma- ja aikakauslehtiin, televisio-ohjelmiin, kirjoihin ja tapahtumiin. Tilatun työn tarkoitus oli muun muassa kehittää yhtiön tietokilpailutuotantoa entisestään.

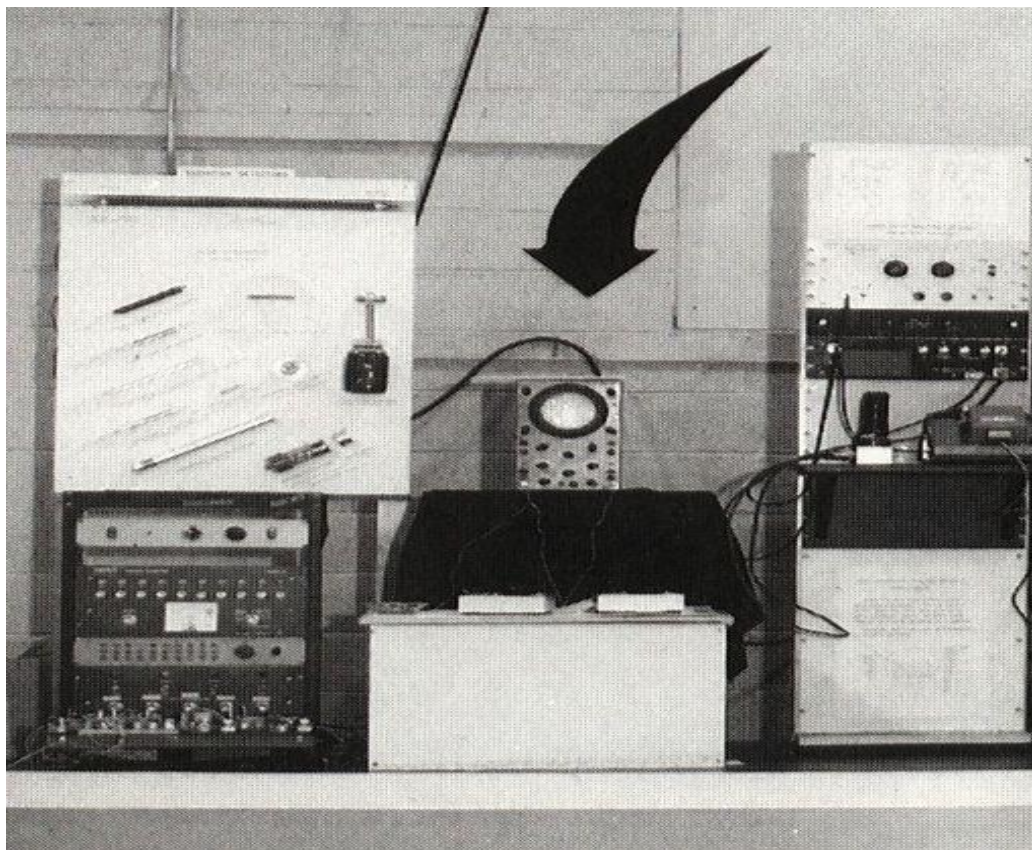
Työssä perehdytään aluksi yleisesti tietokilpailupeleihin peliteollisuudessa, yleisesti pelien, ja tarkemmin tietokilpailupelien markkinatilanteeseen, ja yleisimpiin ansaintamalleihin peleissä. Pelin suunnitteluvaiheessa on hyvä tutkia näitä asioita, jotta tietää, mitä odottaa pelin menestyksestä tai mitä tehdä menestyksen eteen ja mitkä voisivat olla parhaita tapoja saada pelillä tuottoja tehtäessä peliä kaupalliseen tarkoitukseen. Työssä perehdytään myös kahteen insinööriyössä käytettyyn teknologiaan, jotka ovat Unity-pelimoottori, jolla insinööriyön pelisovellus luotiin, ja MariaDB-tietokantapalvelu, jota käytetään pelin kysymysten varastointiin.

Työn viimeisissä osioissa käsitellään projektin tietokilpailupelin teknistä toteutusta, työn tuloksia ja pelin tulevaisuuden näkymiä. Teknisessä toteutuksessa käydään tarkemmin läpi pelin konseptia, käyttöliittymän ja näkymien suunnittelua, pelinäkömän toimintaa sekä pelin tallennustoteutuksia, monetisaatiota, eri alustojen eroja ja viimeiseksi kysymystietokantaa ja sen rakentamista sekä sitä, miten nämä toiminnallisuudet toteutettiin pelissä. Teknisten ratkaisujen esittely painottuu C#-ohjelmointikielellä ohjelmointiin Unityssa, mutta käydään myös läpi Linuxin komentokäyttöjärjestelmän kautta käytettävää MariaDB-tietokannan ohjelmointia.

2 Tietokilpailupelit liiketoimintana

2.1 Tietokilpailupelit osana pelialaa

Videopelejä on tehty 1950-luvulta asti, jolloin alettiin digitalisoida perinteisiä pelejä kuten esimerkiksi shakki, ja kuvasta 1 nähdään vuonna 1958 luotu ensimmäinen kilpailullinen moninpeli Tennis for Two. 1970-luvulta lähtien, kun pelikonsolien, esimerkiksi vuonna 1972 julkaistu maailman ensimmäinen pelikonsoli Magnavox Odyssey, ja kotitietokoneiden massatuotanto alkoi ja arcade-peleihallit alkoivat yleistyä, pelaamisesta on tullut pysyvä osa kaikenikäisten ihmisten vapaa-aikaa. Hyvin alkeellisista vähien pikselien ja muutaman toiminnon kattavista peleistä, jotka tarvitsivat valtavia laitteita edes toimiakseen ja olivat vain harvojen käytettävissä, kehittynyt viihdeteollisuuden osa on nykyään yksi tuottavimmista viihteen aloista satojen miljardien eurojen tuotoilla, ja pelit ovat saatavilla kaikkialla ja kaikille. (1.)



Kuva 1. Alkuperäinen Tennis for Two -peli (2).

Vuonna 2021 viisi pelialan yleisesti suosituinta lajityyppiä olivat Full Press Coverage -sivuston mukaan ensimmäisen persoonan ammuntopelit, nettikasinot, Battle Royale, roolipelit sekä toiminta- ja seikkailupelit (3). Mobiilipuolella nämä viisi lajityyppiä olivat pulmapelit, kasinopelit, strategiapelit, roolipelit ja simulatiopelit (4). Kuten nähdään, tietokilpailupelit eivät kuulu suosituimpiin pelien lajityyppeihin, mutta niille on silti edelleen kysyntää. Peliala on viimeisen parin vuosikymmenen aikana kasvanut räjähdysmäisesti, koska pelit ovat jatkuvasti vain paremmin saatavilla kaikille ja ohjelmistot ja laitteistot ovat parantuneet merkittävästi. Nykypäivän älypuhelimille on saatavilla kasvavassa määrin paljon erilaisia isompia ja vaativampia isojen budjettien pelejä, joista ei ollut tietoaakaan vielä kymmenen vuotta sitten. Tämä on johtanut siihen, että pelit ovat isompia ja hienompia ja puhelimillakaan ei enää ole vain pieniä ja yksinkertaisia pelejä saatavilla, mikä on vähentänyt yksinkertaisempien pelien viehättävyyttä.

Tietokilpailupelillä tarkoitetaan yleisesti peliä, jossa pelaaja yrittää vastata monein eri aihealueisiin liittyviin kysymyksiin oikein samalla kilpaillen yksin tai useamman ihmisen joukkueena. Tietokilpailupelejä on hyvin monenlaisia. Ne ovat yleinen radio- ja televisioviihdeohjelmamuoto, ja niissä pelaajalle on yleensä tarjolla erilaisia palkintoja kuten esimerkiksi rahaa. Niitä löytää myös sanoma- ja aikakauslehdistä, mihin esimerkiksi insinööriyön tilaaja Pressmen Finland Oy on tuottanut niitä, jolloin pelaaja voi yrittää kilpailla muita vastaan tai vain omaa tietämystään vastaan, koska yleensä näissä peleissä ei ole palkintoa luvassa. Tietokilpailupelejä on tehty myös lautapeleiksi, pelikonsolilla, selaimella tai mobiililaitteella pelattaviksi tietokilpailuiksi. (5.)

Tietokilpailupeli-lajityypin yksi tunnetuimmista peleistä niin Suomessa kuin maailmalla on Haluatko miljonääriksi?, joka on alun perin televisioviihdeohjelma ja myöhemmin tehty myös muun muassa lauta- ja videopeliksi, jossa pelaajat oikeilla vastauksilla ja kolmella apukeinolla tavoittelevat miljoonan arvoista rahanpottia. Haluatko miljonääriksi? -pelissä pelaajille esitetään joka kierroksella vaikeutuvia kysymyksiä, joihin pelaajalla on neljä vaihtoehtoa, ja pelin aikana voi käyttää kolme eri oljenkortta helpottaakseen kysymystä ja rajattomasti aikaa vastata kysymyksiin. Toinen, etenkin 2000-luvulla Suomessa hyvin suosittu

selainpelinä oleva tietokilpailupeli on kuvasta 2 nähtävä Älypää, jossa pelaaja vastaa kysymyksiin oikein saadakseen mahdollisimman hyvän pistetuloksen. Älypäässä esitetään sattumanvaraisia kysymyksiä, ja pelaajan pitää mahdollisimman nopeasti vastata kysymyksiin, mutta myös oikein, ettei peli pääty.

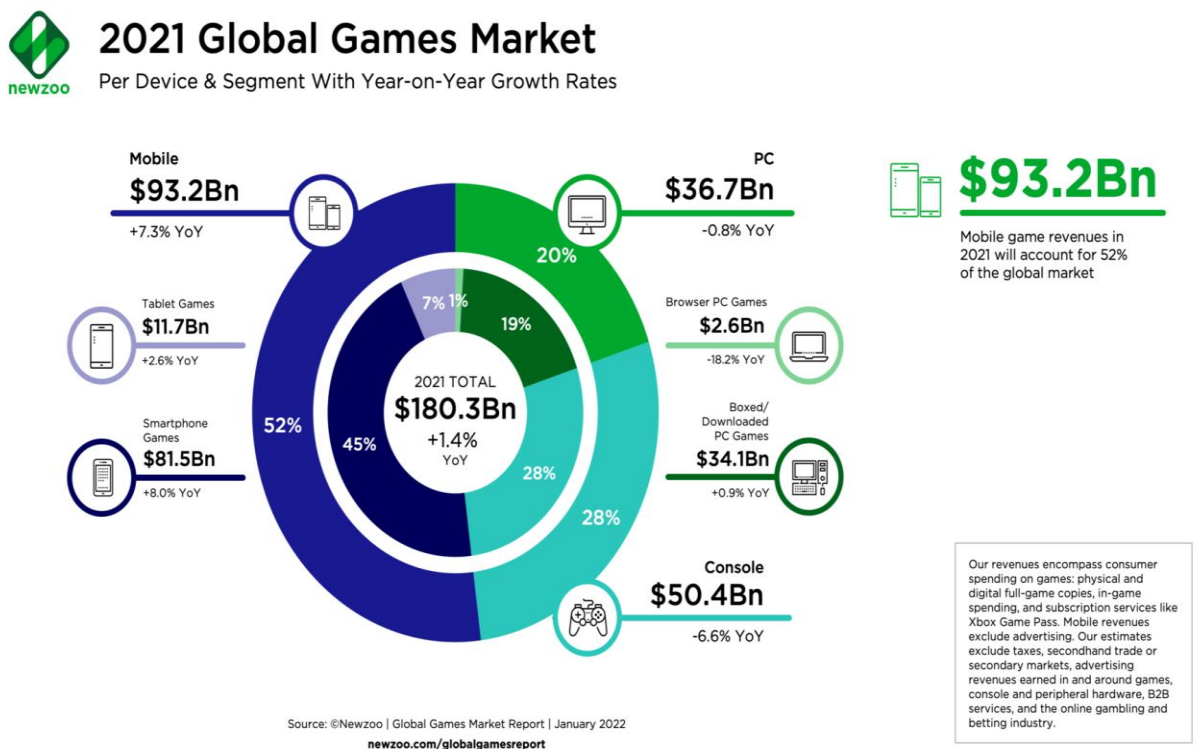


Kuva 2. Älypää-tietokilpailupeli (6).

Tietokilpailupelejä on ollut jo pidemmän aikaa monissa eri medioissa, ja ne ovat erittäin suosittuja etenkin televisio-ohjelmina. Vaikka tietokilpailupelit eivät ole tällä hetkellä suosituimpien pelien lajityyppien joukossa, niitä on aina ollut ja mobiilipuolella on tarjolla paljon erilaisia tietokilpailupelejä. Mobiilipuolella tietokilpailupeleillä on myös paljon latauksia ja hyviä arvosteluja, eli niillä ei ole isoimmat markkinat, mutta markkinat niille kuitenkin on.

2.2 Pelimarkkinat

Peliala on ollut niin pelaajien määrissä kuin tulojen osalta jatkuvassa nousussa jo todella pitkään, eikä se ole arvioiden mukaan pysähtymässä vielä pitkään aikaan. Kuvasta 3 voidaan nähdä pelimarkkinoiden tilanne vuonna 2021, jolloin peliteollisuuden yhteiset tulot olivat 180,3 miljardia dollaria ja nousu vuodesta 2020 oli 1,4 %. Kuvasta nähdään myös, että mobiilipelien tuottamat tulot ovat valtavassa nousussa 7,3 %:n nousulla ja 93,2 miljardin dollarin tuotoilla, joka on noin 51,7 % koko peliteollisuuden tuotoista. Tämä on johtanut siihen, että perinteisempien pelialustojen eli tietokone- ja konsolialustojen myynnit ovat prosentuaalisesti vähentyneet, vaikka kokonaisuudessaan näidenkin alustojen tulot kasvavat joka vuosi. Vuonna 2021 videopelien pelaajien määrä nousi myös 5,3 % ja saavutti kolmen miljardin pelaajan määrän. (7.)



Kuva 3. Vuoden 2021 pelien tulot ja vuosittainen nousu eri pelialustoilla (7).

Suomessa peliteollisuus tuotti yhteensä 195,5 miljoonaa euroa vuonna 2021, ja samalla tietokilpailupelien kerryttämät tuotot olivat Suomessa 840 000 euroa, joka on vähän yli 0,4 %. Tietokilpailupelien tuottojen uskotaan nousevan vuoden 2022 aikana 11,6 %, jolloin tuotot nousisivat 950 000 euroon, ja tästä eteenpäin tuottojen uskotaan nousevan keskimäärin 5,47 %, mikä tarkoittaisi sitä, että tietokilpailupelien tuotot nousisivat jo 1,17 miljoonaan euroon vuonna 2026. Vastaavasti koko peliteollisuuden tulojen uskotaan Suomessa nousevan 327,51 miljoonaan euroon vuonna 2026, jolloin tietokilpailupelien osuus olisi enää vähän alle 0,4 %. (8.)

Yksi isoimmista vaikuttajista tietokilpailupelien tuottojen pieneen osuuteen Suomen peliteollisuudessa on se, että Suomessa ei kovin usein julkaista uusia tietokilpailupelejä samalla, kun monien muiden eri lajityyppien pelejä julkaistaan useammin. Toinen iso vaikuttaja on, että tietokilpailupeleihin on vaikeampi implementoida erilaisia keinoja saada tuotettua rahaa, kuten mikromaksuilla, jotka ovat olleet isossa suosiossa jo vuosia niiden tuottavuuden vuoksi etenkin mobiilipeleissä. Mobiilipuolella mikromaksujen lisäksi jäljelle jää pääasiassa kaksi tapaa tuottaa pelillä rahaa, joko tehdä pelistä maksullinen tai implementoida mainoksia peliin tai molempia. Nykyään harvemmin pelejä tehdään enää maksulliseksi varsinkin mobiilialustoilla, koska tällä tavalla peleihin saadaan enemmän pelaajia ja pelintekijät suosivat yleisesti mieluummin mainoksien tai mikromaksujen käyttöä.

2.3 Ansaintamallit peleissä

Ansaintamalleilla tarkoitetaan tapoja, joilla esimerkiksi peli tai palvelu yrittää hankkia rahallisia tuloja. Ennen videopelit hyödynsivät hyvin vähän erilaisia ansaintamalleja, joista pitkään yleisin oli tehdä pelistä maksullinen kertaluontoisesti, jolloin pelaajan täytyy maksaa tietty rahasumman pelistä ja tämän jälkeen hänellä on koko peli pelattavana. Myöhemmin maksullisiin peleihin alettiin luoda lisäpakkauksia, jotka pelaaja pystyi ostamaan pienemmällä summalla saadakseen peliin lisäsisältöä. Jotkut pelit, kuten 2000-luvun alkupuolella julkaistu massiivinen monen pelaajan verkkoroolipeli World of Warcraft, eivät kuitenkaan

tehneet näin vaan peliin tehtiin kuukausittainen tilausmalli eli pelaajat maksavat kuukausittain tietyn rahasumman saadakseen pelata peliä. Monet saman lajityypin pelit ovat käyttäneet myös mallia, jonka mukaan peli on lähtökohtaisesti ilmainen, mutta pelistä voi maksaa kuukausittain saadakseen pelin kaikki alueet ja ominaisuudet avattua. (9.)

Nykyään samoja ansaintamalleja on edelleen käytössä, mutta niitä on tullut myös todella paljon lisää. Suurimpana vaikuttajana on mobiilipelien suosion kasvu, joka on muuttanut peliteollisuutta kokonaisuutena, ja perinteiset ansaintamallit eivät ole mobiilipeleissä aina ne tuottavimmat mallit. Mobiilipelien ulkopuolelle peleihin on alettu tuoda uusia ansaintamalleja, joita käytettiin ennen vain mobiilipeleissä, ja myös toisinpäin, mikä on yhdistänyt peliteollisuutta kokonaisuutena. Mobiilipeleissä yleisimmät ansaintamallit ovat erityyppiset mainokset, pelin sisäiset mikromaksut ja erilaiset rajoitetun ajan kestävät passit, joilla pelaaja saa maksua vastaan mahdollisuuden saada enemmän sisältöä peliin. Tietokone- ja konsolipeleissä yleisin ansaintamalli on edelleen kertaluontoinen maksu, jolla peli ostetaan kerralla kokonaan, ja peleihin saatetaan lisätä maksullisia lisäpakkauksia. Viime vuosina on myös tullut moniin verkkopeleihin mobiilipeleissäkin nähtäviä rajoitetun ajan passeja. (9.)

Jo pidemmän aikaa mobiilipelejä on julkaistu free-to-play-peleinä eli ilmaispeleinä, joihin on implementoitu paljon erilaisia mikromaksuja, mutta tätä ansaintamallia on alettu käyttää enemmän myös tietokone- ja konsolipeleissä. Kun peli on ilmainen, saadaan pelaajia helposti, koska heidän ei tarvitse käyttää pelaamiseen rahaa. Mobiilipeleissä kuitenkin usein, kun peliä pelataan tarpeeksi paljon, eteneminen hidastuu suhteellisen nopeasti. Esimerkiksi Supercellin kehittämässä Clash Royalessa kerätään ja parannetaan kortteja, joilla pelataan sitten toisia pelaajia vastaan. Näitä kortteja on toista sataa, ja uusia julkaistaan vähän väliä. Ne kaikki pitää saada avattua laatikoiden tai ostamisen kautta. Kaiken pystyy ostamaan tai keräämään ilmaiseksi pelaamalla peliä tarpeeksi, mutta laatikoiden avaaminen kestää pitkään ja mahdollisuus saada haluamansa on hyvin pieni ja resurssien kerääminen ostamiseen ja korttien parantamiseen on hidasta ja hidastuu sitä enemmän, mitä enemmän peliä pelaa. (9.)

Kehittyäkseen nopeammin tai saadakseen uusia kortteja pelaajat pystyvät ostamaan kaupasta pelin ehdottamia erilaisia laatikoita suoraan tai paketteja, mutta nämä laatikot ja paketit maksavat joko pelin sisäistä premium-valuuttaa, jota ostetaan oikealla rahalla tai kerätään pitkään, tai suoraan oikeata rahaa. Houkuttelevilla tarjouksilla, joilla pelaaja pääsee helposti etenemään nopeasti pelissä, saadaan omistautuneemmat pelaajat helposti käyttämään vaikka vain vähän rahaa peliin alussa päästäkseen vauhtiin, ja tämä myöhemmässä vaiheessa voi johtaa suurempaan rahan käyttöön. Tämän takia on myös tärkeää, että pelissä on saatavilla monia eri paketteja halvoista kalliimpiin. Näin saa enemmän arvoa rahalle, niin että käytännössä kuka vain pystyy käyttämään peliin rahaa. Kuvassa 4 on Clash Royal -pelin sisäinen jalokivikauppa hintoineen. (9.)



Kuva 4. Clash Royale -pelin sisäinen jalokivikauppa (10).

Mikromaksujen lisäksi mobiilipelit voivat tuottaa passiivisesti tuloja sovelluksen sisäisistä mainoksista. Pelin sisäiset mainokset ovat edelleen isoin trendi moneistoinnissa, koska pelaajat saavat rauhassa pelata ilmaiseksi peliä ja heille vain

välillä näytetään muista sovelluksista lyhyitä mainoksia, joista kehittäjät saavat pienen mainostulon. Pelaajat muistavat pelin sisäiset mainokset 47 % ajasta, ja 50 % pelaajista pitää mobiilimainoksia visuaalisesti miellyttävänä, mikä kertoo niiden tehokkuudesta. (9.)

Pelin sisäisessä mainonnassa pelin kehittäjien pitää myös miettiä pelaajien kokemusta. Mainokset eivät saa olla liian häiritseviä tai epäolennaisia, eikä mainoksia saa näyttää liikaa tai liian usein, koska pelaajan pelikokemus kärsii tästä, mikä saattaa johtaa pelaajien menetykseen. Pelinkehittäjien pitääkin löytää tasapaino mainosten määrässä ja laadussa ja valita heille oikea mainosverkosto maksimoidakseen mainostulot. Pelin sisäisiä mainoksia on eri formaateissa, joista tärkeimpiä ovat palkitsevat, väli-, pelattavat ja bannerimainokset. (9.)

Palkitsevat mainokset ovat yksi suosituimmista mainosformaateista, ja niillä on suurin eCPM (effective cost per mille) eli tuotto tuhatta näyttökertaa kohden kaikista mainosformaateista. Niiden toiminta perustuu siihen, että pelaaja vapaaehtoisesti valitsee katsoa mainoksen yleisesti nappia painamalla ja vastineeksi saa tästä itselleen jonkin palkinnon, kuten esimerkiksi pelin sisäistä valuuttaa tai muita palkintoja. Palkitsevat mainokset ovat vähemmän häiritseviä, mutta myös haluttuja pelaajien mielestä, sillä jopa 68 % pelaajista pitää tästä mainonnan formaatista ja vain 24 % pitää niitä häiritsevinä. Palkitsevat mainokset ovat myös mainostajille hyvä asia, koska pelaajat vapaaehtoisesti katsovat niitä ja näin saattavat myös paremmin keskittyä kyseisiin mainoksiin, mikä voi saada heidät painamaan mainoksia ja kokeilemaan mainostettua sovellusta. Palkitut mainokset myös useimmiten katsotaan kokonaan läpi, mikä johtaa siihen, että pelien julkaisijat saavat suurempia eCPM-tuloja ja yleisesti näin kasvattavat mainostuloja kokonaisuutena. (11.)

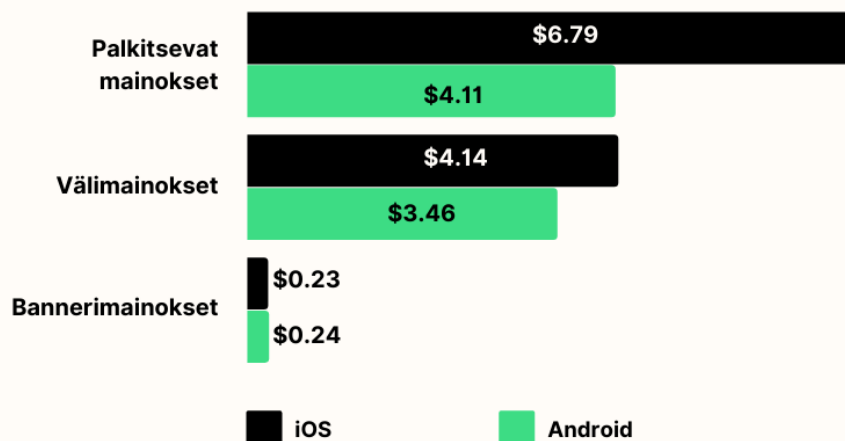
Välimainokset ovat täyden ruudun kattavia mainoksia, jotka pystyy ohittamaan tietyn ajan kuluttua, ja ne voivat olla staattisia, videoita tai myös pelattavia. Välimainokset ovat erilaisia kuin palkitsevat mainokset, koska niitä ei yleensä liitetä mihinkään nappuloihin eikä niistä pelaaja saa itselleen mitään palkintoa. Tästä syystä niiden eCPM-arvo on pienempi kuin palkitsevien mainosten.

Välimainokset implementoidaan yleisesti pelin koodissa tiettyihin tapahtumiin, kuten pelin loppumisen yhteyteen tai tietyn kysymysmäärän jälkeen. Tässä kohdassa peli on jo pysäytetty, jolloin se ei ole pelaajille niin häiritsevä eikä pelaajan pelikokemus häiriinny. Välimainoksien kanssa pelinkehittäjien pitää suunnitella tarkasti, kuinka usein ja milloin näyttää välimainoksia, niin että niillä pystyy tuottamaan tuloja menettämättä pelaajia. (11.)

Bannerimainos on perinteinen mobiilipelin mainosformaatti. Bannerimainoksia käytetään paljon, koska oikein toteutettuna ne ovat huomaamattomia sillä tavalla, että ne eivät häiritse pelaamista. Yleensä bannerimainokset sijoitetaan ylä- tai alaosaan ruutua, missä ne eivät häiritse pelikokemusta. Hyviä puolia bannerimainoksissa on mainostajille niiden pieni hinta ja tuottamisen helppous. Huono puoli niissä taas on se, että niiden suosio on laskenut, koska pelaajat ovat tottuneet niihin eivätkä enää klikkaa niitä niin usein. Mainostulojen puolesta bannerimainokset eivät pysty kilpailemaan palkitsevien mainosten tai välimainosten kanssa yksinään, mutta muihin mainosformaatteihin yhdistäessä ne ovat hyvä lisä. (11.)

Kuvassa 5 ovat eCPM-keskiarvot eri mainosformaateilla jaettuna iOS- ja Android-alustoille vuonna 2021. Tilastoista nähdään, että palkitsevien mainosten eCPM-arvot ovat paljon korkeammalla verrattuna välimainoksiin ja varsinkin bannerimainoksiin verrattuna, ja myös, että yleisesti iOS-alustalla eCPM-arvot ovat korkeammalla Androidiin verrattuna. Palkitsevien mainosten eCPM-arvoa nostaa niiden läpikatsomisprosentti ja niiden vapaaehtoisuus. Välimainokset ovat myös tehokkaita, mutta niiden eCPM-arvo on silti paljon palkitsevia mainoksia alempana, koska ne pystyy ohittamaan paljon nopeammin eivätkä ne ole vapaaehtoisia, joita pelaaja voisi samalla tavalla haluta katsoa. Bannerimainoksilla on muihin mainostyyppeihin verrattuna hyvin pieni eCPM-arvo, koska nämä mainokset näkyvät jatkuvasti ja niiden klikkauksien määrä on vähentynyt. Mainosverkostoja on myös paljon erilaisia, mutta yleisesti AdMob-verkostolla on parhaat eCPM-arvot. (12.)

eCPM keskiarvot eri mainosformaateilla 2021



Kuva 5. eCPM-keskiarvot eri mainosformaateilla vuonna 2021 (12).

Ansaintamalleja on etenkin mobiilipeleissä paljon. Ilmaiset pelit ovat paljon maksullisia pelejä suosituimpia etenkin nykyään. Maksullisissa peleissä hyvää on se, että pelaajalta saadaan tietty summa tuloja heti, mutta ilmaisissa peleissä muilla ansaintamalleilla on potentiaalia tuottaa vielä enemmän tuloja pidemmällä aikavälillä. Ilmaisen pelin kehityksessä mainosverkoston ja mainosformaattien valinta ja implementointi ovat tärkeitä asioita miettiä, jotta voidaan maksimoida pelin mainostulot ja pitää pelaajat kiinnostuneena pelistä.

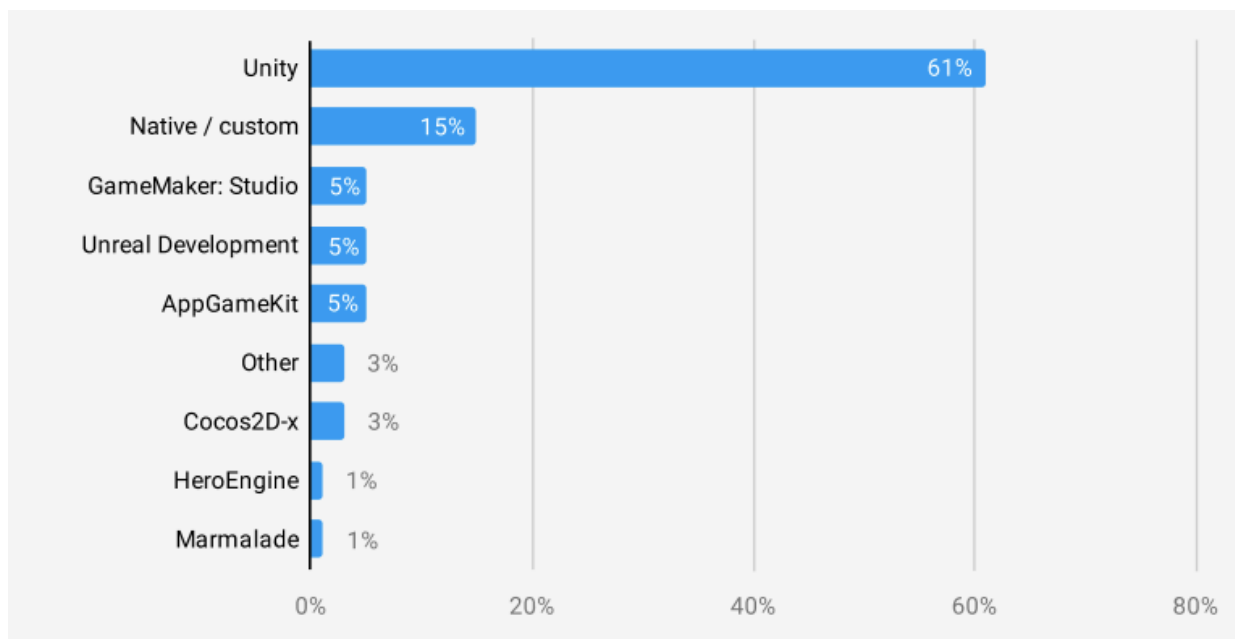
3 Työssä käytetyt tekniikat ja ohjelmistot

3.1 Unity-pelimoottori

Unity on Unity Technologies -yrityksen kehittämä järjestelmäriippumaton kaupallinen pelimoottori, joka sisältää graafisen kehitysympäristön kolmi- ja kaksikulotteisten pelien ja muiden sovellusten luomiseen. Unityn ohjelmointikielenä toimii C#, mutta vuodesta 2021 eteenpäin Unityssa pelinkehityksessä voidaan myös käyttää visuaalista ohjelmointia, jossa ei tarvitse osata tai käyttää C#-

ohjelmointia. Unityn ensimmäinen versio julkistettiin ja julkaistiin Applen vuosittaisessa Worldwide Developers Conference -konferenssissa kesäkuussa 2005, jolloin se oli Mac OS X -käyttöjärjestelmälle yksinomainen pelimoottori. Pian tämän jälkeen Unity laajentui myös Windows- ja verkkoselainpuolelle, ja nykyään se tukee jo yli 25:tä eri alustaa tietokoneissa, mobiililaitteissa, pelikonsoleissa ja virtuaalitodellisuudessa. Näistä kaksi merkittävintä mobiilikäyttöjärjestelmää ovat Android ja iOS. (13; 14; 15.)

Unitylla on saatavilla muutamia erilaisia lisenssivaihtoehtoja erilaisille pelinkehittäjille. Harrastelijat ja pienyritykset, joiden tulot ovat alle satatuhatta dollaria vuodessa, saavat käyttää Unityn ilmaista perusversiota. Tämän lisäksi Unitystä on saatavilla Plus-, Pro-, Industrial- ja Enterprise-versiot, joissa on erilaisia lisäominaisuuksia ja etuuksia perusversioon nähden. Maksullisten lisenssien vuosittaiset hinnat vaihtelevat 399 dollarista aina 4800 dollariin vuodessa organisaatiossa Unityn kanssa työskentelevää henkilöä kohden. Unity Technologiesin tutkimuksesta otetusta kuvasta 6 selviää, että vuonna 2020 mobiilipelien tekijöistä 61 % käytti peleissään Unitya ja toiseksi eniten pelintekijät käyttivät omaa pelimoottoriaan, joka muodosti 15 % pelintekijöistä. Unitylla tehdyt pelit, jos otetaan huomioon kaikki alustat eli mobiili, tietokone ja konsoli, muodostavat kokonaisuudessa yli 50 %, mikä tekee siitä pelialan käytetyimmän pelimoottorin. (14; 16; 17.)



Kuva 6. Pelintekijöiden käyttämät pelimoottorit mobiilipeleissä vuonna 2020 (17).

Unity on ominaisuuksiltaan ja mahdollisuuksiltaan erittäin laaja. Se on eniten käytetty pelimoottori etenkin mobiilipelimarkkinoilla sen vaivattomuuden, helpouden ja moninaisuuden takia luotaessa pelejä monille eri alustoille, mikä on Unitylla vaivatonta. Unitylla on myös hyvin laaja dokumentointi kaikista sen toiminnoista, ja sen yhteisöstä löytää helposti apua ongelmatilanteissa, koska niin moni käyttää juuri Unitya. Näiden syiden takia Unity valittiin tässä työssä alun perin jo tietokilpailupelin pelimoottoriksi.

3.2 MariaDB-relaatiotietokantajärjestelmä

MariaDB on yhteisön kehittämä ja kaupallisesti tuettu MySQL-relaatiotietokantaohjelmistoon pohjautuva ilmainen ja avoimen lähdekoodin relaatiotietokantajärjestelmä, joka on GNU GPL -lisenssin alla. MariaDB:n loivat ja sitä kehittävät MySQL:n alkuperäiset kehittäjät. Se luotiin, koska MySQL:n kehittäjät eivät uskoneet Oracle-yhtiön, joka osti MySQL:n vuonna 2009, olevan hyvä omistaja MySQL:lle, ja kehittäjät halusivat varmistaa MySQL-koodin pysyvän aina ilmaisenä. MariaDB on ja se on tarkoitus pitää mahdollisimman yhteensopivana

MySQL:n kanssa muun muassa kirjastojen ja komentojen osalta, mikä mahdollistaa sen käyttöönoton helposti MySQL:n tilalle, mutta vain parempana versiona. MySQL:ään verrattuna MariaDB:ssä on enemmän varastointimoottoreita, sen nopeutta on parannettu, siinä on laajennuksia ja uusia ominaisuuksia, paremmat mahdollisuudet testaukseen ja vähemmän varoituksia ja ohjelmointivirheitä. (18; 19; 20.)

MariaDB:n hinnoittelusivulta otetusta kuvasta 7 nähdään MariaDB:n kolme tärkeintä tuotetta, jotka ovat MariaDB Community Server, MariaDB Enterprise ja MariaDB SkySQL, ja mitä ne sisältävät. MariaDB Community Server on MariaDB:n ilmainen moderni relaatiotietokanta, joka on sen tärkein tuote ja yhtiön kulmakivi. MariaDB Enterprise on maksullinen täyden tuotantoluokan avoimen lähdekoodin tietokantaratkaisu. Siihen kuuluu MariaDB Enterprise Server, joka on paranneltu ja suojattu versio MariaDB Community Serveristä. Se luotiin tarjoamaan asiakkaille luotettavuutta, vakautta ja pitkäaikaista tukea etenkin suurien tietokantojen kanssa työskennellessä. MariaDB Enterprise tarjoaa myös paljon lisää erilaisia työkaluja ja laajennuksia tietokantaan ja laajan tukipalvelun. MariaDB SkySQL on ensimmäinen ja ainoa database-as-a-service eli tietokanta palveluna, joka tuo MariaDB Enterprisen täyden tehokkuuden pilveen, joka yhdistää tehokkaat ominaisuudet, laajan asiakastuen ja helppokäyttöisyyden toimivaksi kokonaisuudeksi. (21; 22; 23.)



Kuva 7. MariaDB:n tarjoamat tuotteet (24).

Tietokilpailupeliä varten tarvitaan erittäin suurten kysymysmäärien varastointiin hyvä ja tehokas tietokantatoteutus, jota olisi helppo muokata ja laajentaa ja joka olisi edullinen ylläpitää. MariaDB:n tietokantapalvelu täyttää kaikki nämä kriteerit, koska se on nopea, tehokas, lähtökohtaisesti ilmainen käyttää ja myös helppo laajentaa tarvittaessa. Tästä syystä se päätettiin ottaa käyttöön tämän työn tietokilpailuprojektiin kysymysten varastointia varten.

4 Tietokilpailu-pelisovellus

4.1 Konsepti

Insinööriyönä tehty tietokilpailupeli on pääpiirteittäin perinteinen tietokilpailupeli. Pelissä pelaajalle esitetään eri aihealueilta erilaisia kysymyksiä, joihin hänen kuuluu vastata useamman vastausvaihtoehdon turvin tietyn aikarajan sisällä. Näistä vaihtoehdoista pelaaja valitsee omasta mielestään oikean vastauksen, ja peli siirtyy kysymyksestä seuraavaan. Erona insinööriyön tietokilpailupelissä moniin muihin tietokilpailupeleihin verrattuna on, että pelissä pääpaino ei ole

siinä, että pelaaja on nopea vastaamaan. Pelissä tärkein asia on, että pelaaja voi ymmärtää kysymykset ja miettiä vastausta kohtuullisen ajan sisällä ja kehittää ja haastaa samalla itseään miellyttävällä tavalla ilman lyhyen vastausajan tuomaa kiirehdintää ja stressiä. Pelin pisteet eivät myöskään määräydy nopeuden mukaan vaan sen mukaan, kuinka paljon pelaaja saa oikeita vastauksia kullakin kierroksella.

Tietokilpailupeli sisältää kymmenen kysymyksen sarjoja tasoilla ja useita vaikeutuvia pelitasoja. Pelaajan läpäistessä pelin tulos tallennetaan tulostauluun, josta pelaaja voi tutkia omia parhaita tuloksiaan. Tästä eteenpäin pelaaja voi pelata lisää ja yrittää voittaa omat parhaat tuloksensa, mikä tuo uudelleenpelattavuutta pelille. Aloitettu peli pystytään myös tallentamalla jättämään kesken ja jatkamaan myöhemmin, mikä on myös epätavallinen ominaisuus yleisesti nopeatempoisissa tietokilpailupeleissä. Tyyliiltään peli on asiallinen, mutta pelaajalle mielenkiintoinen, mikä saavutetaan hyvillä ja mielenkiintoisilla kysymyksillä. Tietokilpailupeli luodaan mobiilille Android- ja iOS-alustalle ja verkkoselainalustalle, ja se aiotaan julkaista kaikille näille alustoille. Pelistä on pyritty tekemään mahdollisimman uudenlainen tietokilpailupeli, jollaista markkinoilla ei tällä hetkellä ole.

4.2 Käyttöliittymän suunnittelu ja näkymät

Insinööriyön tietokilpailupeli on täysin käyttöliittymäpohjainen peli, joten sen käyttöliittymän täytyy olla hyvä. Pelin käyttöliittymää ja näkymiä suunnitellessa pyrittiin KISS-periaatteeseen, joka korostaa yksinkertaisia ratkaisuja, koska käyttöliittymästä haluttiin pelaajalle mahdollisimman helppo ja miellyttävä. Käyttöliittymässä haluttiin myös mahdollistaa pelaajalle värimaailman puolesta muokattavuus. Eri pelaajilla on monesti eri mieltymykset, mistä kukin pitää ja ei pidä, tai jos joku pelaaja on värisokea, joitakin värejä voi olla vaikea erottaa, jolloin erilainen väriteema saattaa sopia paremmin ja pelattavuus paranee. Tästä syystä asetusvalikkoon luotiin kahdella nuolinäppäimellä toimiva väriteeman vaihto, joka vaihtaa pelin kaikkien taustojen ja fonttien värit teeman mukaisesti.

Pelin yksinkertaisuuden vuoksi tietokilpailupeli luotiin kokonaan yhteen kohtaukseen, ja peli jaettiin kahteen käyttöliittymäkehykseen, jotka oli tarkoitettu päävalikolle ja pelille. Päävalikko ja peli jaettiin omiin käyttöliittymäkehyksiin, koska jos Unityssa esimerkiksi näissä olevaan tekstielementtiin tulee muutos, koko kehys pitää päivittää ja hahmottaa uudelleen, mikä voisi käydä raskaaksi laitteelle. Pelissä vain tietyt osat ovat toiminnassa samanaikaisesti, kuten esimerkiksi pelin ollessa päällä päävalikko on pois käytöstä. Peli luotiin yhteen kohtaukseen pelin eri näkymien ja osien lataamisen, siirtymisen ja ylläpidon helpottamiseksi ja koska peli itsessään ei ole suorituskyvylle raskas. Tällöin pelin ei aina tarvitse ladata eri kohtauksia ja käydä läpi kaikkia ohjelmakoodeja yhä uudelleen. Tällaisessa yksinkertaisemmassa pelissä kohtauksien jatkuva lataaminen hidastaisi pelin toimintaa turhaan ja olisi rasitteena pelaajille.

Kuvasta 8 nähdään tietokilpailupelin mobiiliversion päävalikon näkymä, jonka pelaaja näkee ensimmäisenä pelin käynnistyttyä. Päävalikon tehtävänä on toimia sovelluksen navigaatorakenteen juurena, josta pelaaja pystyy liikkumaan mihin näkymään tahansa. Päävalikon toinen tehtävä on antaa pelaajalle selkeä käsitys pelin sisällöstä. Pelaajan valitseman teeman väriytykset näkyvät päävalikossa, mutta myös kaikissa muissa näkymissä. Samojen värien ja käyttöliittymäelementtien käytön tarkoitus on luoda pelille yhtenäisyyttä kaikkien eri näkymien välille. Valikossa kaikki painike-elementit haluttiin listamaisesti ja pelkinä teksteinä, jotta päävalikko olisi mahdollisimman selkeä. Valikon keskelle asetettiin oleellimmat otsakkeet, kuten pelin aloittaminen tai jatkaminen ja asetukset. Valikon alla on rivissä ei-niin-oleellisia otsakkeita, joista pääsee esimerkiksi pelin ulkopuolelle ehdottamaan kysymystä tai kertomaan virhe-epäilystä.



Kuva 8. Tietokilpailupelin päävalikon näkymä mobiiliversiossa.

Tulokset-valikko on päävalikon ensimmäinen alavalikko, ja siitä esimerkki on esitettyä kuvassa 9. Valikon tarkoitus on näyttää pelaajalle hänen parhaat paikalliset pistetuloksensa ja milloin ne on saavutettu, mikäli pelaaja on onnistunut läpäisemään pelin. Tämä kannustaa pelaajaa pelaamaan peliä uudelleen, enemmän ja kilpailemaan omaa tietämystään vastaan saadakseen parempia tuloksia. Valikossa päivämäärätekstin alle tulee tuloksien saavuttamisen päivämäärä ja pisteet-tekstin alle päivämäärän kohdalle saavutetut pisteet, ja parhaita tuloksia voi listassa olla yhteensä kymmenen. Valikko suljetaan alareunan nuolipainikkeella, joka vie pelaajan takaisin päävalikkoon.



Päivämäärä	Pisteet
14/06/2021	60
14/06/2021	10

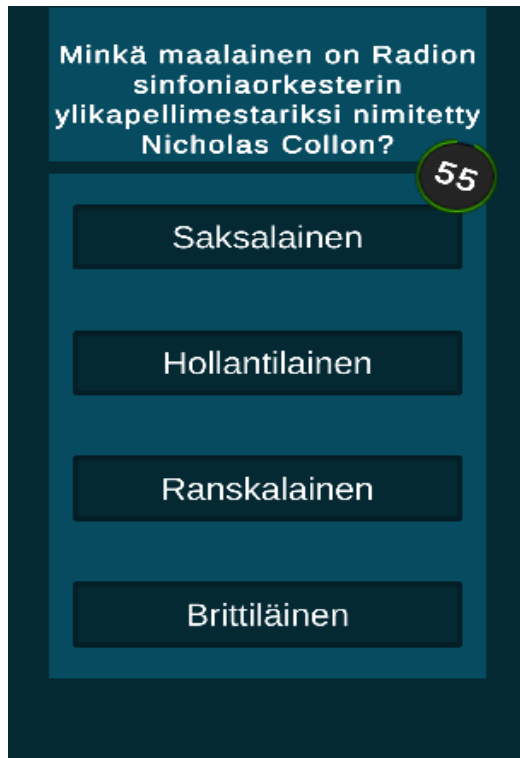
Kuva 9. Tietokilpailupelin tulokset-valikon näkymä.

Kuvassa 10 on asetukset-valikko, joka on päävalikon alavalikko. Valikossa pystyy vaihtamaan pelin väriteeman painamalla nuolinäppäimiä ja mykistämään tai ottamaan mykistyksen pois musiikista ja äänistä painamalla niiden kuvakkeita. Väriteeman nuolinäppäimiä painamalla väriteema vaihtuu automaattisesti ja myös nuolien välissä oleva teeman nimi vaihtuu osoittamaan valittua teemaa. Musiikin ja äänien painikkeita painamalla pelaaja voi valita, haluaako kuulla pelissä kyseisiä ääniä. Painikkeita painamalla niiden kuvake myös vaihtuu osoittamaan, ovatko musiikki tai äänet päällä. Valikossa on alempana myös nuolipainike, jolla pääsee valikosta takaisin päävalikkoon.



Kuva 10. Tietokilpailupelin asetukset-valikon näkymä mobiiliversiossa.

Kuvan 11 pelinäkymä on pelikokemuksen kannalta olennaisin näkymä, koska varsinainen pelaaminen tapahtuu sen kautta. Koska tietokilpailupeli on hyvin yksinkertainen ja helposti ymmärrettävä pelimekaniikkojen puolesta, pelin käyttöliittymästä haluttiin myös sellainen. Käyttöliittymään ei haluttu liikaa häiriötekijöitä, vaan pelaajalle näytetään pelkästään oleellinen pitäen tyyli asiallisena. Näkymän yläosasta pelaaja voi seurata pelin tilastoja ja etenemistä, johon kuuluvat muun muassa pisteet ja se, monesko kysymys on menossa. Näiden alapuolella on laatikko, josta pelaaja näkee senhetkisen kysymyksen, ja tämän alapuolella toinen laatikko, jossa on vastausvaihtoehtopainikkeet ja niiden välissä oikeassa reunassa ajastin, joka kertoo pelaajalle, kuinka paljon hänellä on tässä kysymyksessä aikaa vastata.



Kuva 11. Tietokilpailupelin pelinäkömä mobiiliversiossa.

Pelaajan painaessa vastauspainiketta pelinäkömän keskelle avautuu ikkuna, jossa kerrotaan, oliko vastaus oikea vai väärä. Tästä ikkunasta pelaaja pystyy poistumaan päävalikkoon tai siirtymään seuraavaan kysymykseen. Jos kysymys oli kymmenes eli viimeinen, peli avaa toisen ikkunan, jossa kerrotaan, miten pelaaja suoriutui kysymyssarjasta ja peli jatkuu tästä eteenpäin. Jos pelaaja on jo pelin lopussa, peli avaa pelin loppuikkunan, josta pelaaja näkee koko pelin tilastot ja pystyy poistumaan päävalikkoon.

4.3 Pelin toiminta

Tietokilpailupelin koko toiminnallisuus alkaa päävalikossa, jossa on MainMenu-luokka. Pelin käynnistyessä pelissä on aluksi kaikki eri näkymät päällä, jotta kaikki eri luokat saavat kaiken tarvitsemansa, esimerkiksi muut luokat ja objektit, käyttöönsä. Kaikkien eri näkymien ja objektien ei kuitenkaan tarvitse olla jatkuvasti päällä, ja tämän takia MainMenu Awake-metodissa, jota kutsutaan heti pelin ladattua luokan ilmentymän, kaikki ylimääräiset asetetaan pois päältä

SetActive-metodin avulla. Muuten MainMenu hoitaa pääasiassa valikon kaikkien painikkeiden toimintoja, joista suurin osa avaa erilaisia näkymiä, kuten esimerkiksi asetukset-valikko, ja samalla sulkee päävalikon. MainMenussa on myös StartNewButton-metodi, joka kutsuu GameManager-luokan StartNewGame-metodia.

Tietokilpailupelin tärkein luokka on GameManager, joka hoitaa kaiken pelinäkömän ollessa päällä. StartNewGame-metodista peli lähtee liikkeelle. Metodin pääasiallinen toiminto on nollata peli, niin että pisteet ja vastaukset ovat nolla, mutta myös hakea ensimmäinen kymmenen kysymyksen sarja tietokannasta. Luokassa on myös kaikki muut metodit pelin toimimiseksi, kuten seuraavan kysymyksen aloittaminen, pelin vastausvaihtoehtopainikkeiden toiminnan metodit, pelin lopettaminen ja tallentaminen.

Luokan tärkein metodi on CheckGameState, jota kutsutaan luokan Update-metodissa jokaisella ruudunpäivityksellä, ja sen rakenne on esitelty esimerkkikoodissa 1. CheckGameState toimii siis tilakoneen tavoin, koska se tarkistaa jatkuvasti, missä tilassa ollaan, ja toimii sen mukaisesti. Tilassa 1 pelaaja on vastamassa kysymykseen, joten tässä tilassa pelissä olevan ajastimen aika vähenee ja tarkistaa, onko pelaaja vastannut kysymykseen, ja kun pelaaja vastaa kysymykseen tai aika loppuu, peli siirtyy tilaan kaksi. Tilassa 2 peli tarkistaa, onko pelaajan vastaus oikein vai väärin tai, jos aika loppui, vastaus lasketaan vääräksi. Pelaaja näkee ruudulla tekstistä, menikö kysymys oikein vai väärin, ja tässä kohtaa pelaajalla on mahdollisuus painiketta painamalla jatkaa seuraavaan kysymykseen tai poistua päävalikkoon, jolloin peli myös tallentuu.

```

void CheckGameState()
{
    // Kysymys-tila
    if(gameState == 1)
    {
        // Tila minne siirrytään, kun kysymykseen on vastattu.
        if(gameState == 2)
        {
            // Tila minne siirrytään, kun kysymyssarja on käyty läpi
            if(gameState == 3)
            {
                // Pelin lopetuksen tila. Näytetään pelaajan tilastot
                if(gameState == 4)
                {
                    {
                }
            }
        }
    }
}

```

Esimerkkikoodi 1. CheckGameState-metodin rakenne.

Peli siirtyy tilaan 3, kun pelaaja on vastannut kaikkiin kysymyssarjan kysymyksiin, ja tarkistaa, kuinka paljon pelaaja on vastannut kysymyksistä oikein. Oikeiden vastauksien määrä määrää, millaisen kysymyssarjan pelaaja saa seuraavaksi eteensä. Jos peli on jo lopussa, peli siirtyy tilaan 4. Tilassa 4 pelaaja on läpäissyt pelin ja hänelle näytetään näkymässä, kuinka paljon hänelle kertyi oikeita ja vääriä vastauksia ja paljonko hänen yhteispisteensä ovat. Tästä tilasta pelaaja pystyy siirtymään painikkeen kautta joko päävalikkoon tai aloittamaan suoraan uuden pelin.

Kysymysten hakemiseen tietokannasta peliin peli käyttää DatabaseManager- ja Question-luokkaa yhdessä. DatabaseManager tekee kysymysten varsinaisen hakemisen esimerkkikoodissa 2 nähtävällä GetQuestion-metodilla. Metodi ottaa yhteyden ulkoisella palvelimella olevaan PHP-tiedostoon, joka hakee kysymykset tietokannasta tempString-lauseella. Lauseeseen tulee tietokannan hakulause SELECT-alkuisesti ja perään tiedot, mitä haetaan. Nämä tiedot ovat kysymys ja vastausvaihtoehdot, ja loppuun FROM-lauseella kerrotaan, mistä taulukosta tiedot haetaan, ja WHERE-haulla saadaan haettua oikeanlaiset kysymykset.

```

IEnumerator GetQuestion()
{
    string tempString = "";
    tempString = SQLsearch + difficulty;
    WWWForm form = new WWWForm();
    form.AddField("var1", tempString);

    using (UnityWebRequest www = UnityWebRequest.Post(osoite, form))
    {
        yield return www.SendWebRequest\(\);
        if(www.isNetworkError || www.isHttpError)
        {
            Debug.Log(www.error);
        }
        else
        {
            Debug.Log(www.downloadHandler.text);
            currentQuestion.Parse(www.downloadHandler.text);
        }
    }
}

```

Esimerkkikoodi 2. Metodi kysymyksen hakemiseen tietokannasta.

Kun kysymykset saadaan tietokannasta, DatabaseManager kutsuu Question-luokan Parse-metodia. Parse-metodi käy läpi saadusta tiedosta jokaisen kirjaimen yksitellen lisäten niitä listaan ja tarkistaen tiettyjä erikoismerkkejä, jotka voivat tarkoittaa esimerkiksi kysymyksen loppumista tai että tiettyyn kohtaan kuuluu tulla välilyönti. Näin haetut tiedot saadaan muutettua oikeaan muotoon ja lisättyä oikeaan paikkaan pelissä.

4.4 Tallentaminen ja lataaminen

Tietokilpailupeliin haluttiin sisällyttää mahdollisuus tallentaa pelin eteneminen, asetusten valinnat ja paikallisen pistetaulukon tulokset erillisiin tallennustiedostoihin pelaajan käyttämään laitteeseen. Pelin eteneminen päätettiin tallentaa, koska tässä pelissä yksi pelin läpipeluu voi kestää pitkiäkin aikoja eikä pelaajaa haluta pakottaa pelaamaan pitkiä aikoja yhtäjaksoisesti. Asetukset haluttiin tallentaa pelaajan mukavuuden vuoksi, ettei jokaisella pelikerralla pelaajan tarvitse esimerkiksi äänien mykistystä säätää erikseen. Paikallisen pistetaulukon tallennus päätettiin implementoida, jotta pelaaja voisi vertailla omia tuloksiaan ja pelata peliä enemmän voittaakseen nämä tulokset. Tallennukset ohjelmoitiin käyttäen hyödyksi eri tallennustekniikoita, koska esimerkiksi pelitiedoston tallennus

on hyvä salata kunnolla huijaamisen tai pelin rikkomisen estämiseksi, kun taas esimerkiksi asetuksissa tällaiselle ei ole tarvetta, koska asetuksissa ei ole paljoa huijattavaa, jos näihin tiedostoihin pääsisi edes käsiksi.

Sovelluksen käynnistyessä MainMenu-luokka kutsuu FindSavedGame-metodia, joka tarkistaa laitteesta, löytyykö pelin tallennustiedostoa. Jos tiedosto löytyy, päävalikossa on Jatka-painike, josta tallennettua peliä voi jatkaa, ja jos tiedostoa ei löydy, Jatka-painiketta ei näytetä. Pelaajan ei ole kuitenkaan pakko jatkaa peliä, vaikka tallennustiedosto löytyisi, vaan pelaaja voi aloittaa aina uuden pelin Uusi peli -painikkeella. Uuden pelin aloittaminen tyhjentää tallennustiedoston tiedoista ja kirjoittaa aloitusarvot tyhjään tiedostoon. Jos pelaaja kuitenkin haluaa jatkaa aikaisemmin pelattua peliä ja siirtyy peliin Jatka-painikkeella, peli avaa pelinäkömän ja kutsuu GameManager-luokan LoadPlayer-metodia, joka toimii samalla tavalla kuin uuden pelin aloittamiseen käytetty metodi, mutta muuttujien nollaamisen sijaan lataa tallennustiedostosta kaikki tarvittavat muutujat pelin toimimiseksi ja viime kerralla olleen kysymyslistan.

Pelin tallennustiedostoon tallennetaan pelille ja pelin toiminnalle kriittisiä tietoja, minkä takia tiedostolle on hyvä tehdä salaus, ettei kuka vain pääse muuttamaan sen tietoja. Tiedoston salaamisessa käytetään tiedoston alustamista binäärimuotoon, koska se on turvallinen ja tietoja on vaikea saada muutettua ulkoisesti. Esimerkkikoodista 3 nähdään pelitiedoston tallentamiseen käytettävä metodi SaveGame, joka on omassa SaveSystem-luokassa. SaveGame ottaa parametrinä GameManager-luokan, koska siellä olevia tietoja tallennetaan. Koodi alussa alustaa binäärialustajan, seuraavaksi avataan tiedostovirta tiedoston sijaintiin, että voidaan luoda tai muokata tiedostoa. Lopuksi alustaja kirjoittaa PlayerData-luokan avulla, joka ottaa tallennettavat tiedot GameManagerista, tiedot tiedostoon binäärimuodossa ja tiedoston muokkaaminen suljetaan. Tiedoston lataaminen toimii toisinpäin: tiedosto avataan ja tiedot muutetaan normaaliiksi ja siirretään PlayerDatan kautta GameManageriin käytettäväksi.

```

public static void SaveGame(GameManager manager)
{
    BinaryFormatter formatter = new BinaryFormatter();

    string path = Path.Combine(Application.persistentDataPath,
        "quizgame.data");
    FileStream stream = new FileStream(path, FileMode.Create);

    PlayerData data = new PlayerData(manager);

    formatter.Serialize(stream, data);
    stream.Close();
}

```

Esimerkkikoodi 3. Pelin tallennustiedoston tallentamisen metodi.

Asetuksien tallentamisessa, johon kuuluu äänien ja musiikin näppäimien tilat ja väriteeman indeksi, käytettiin Unityn PlayerPrefs-luokan toimintoja. PlayerPrefs on luokka, johon pystyy tallentamaan merkkijonoja, kokonais- ja liukulukuja, jotka eivät häviä pelisessioiden välillä. PlayerPrefs otettiin käyttöön, koska asetuksista ei tarvittu tallentaa kuin se, mikä väriteema pelaajalla oli käytössä ja olivatko musiikki ja äänet päällä. Pelin alussa GameManager kutsuu SettingsMenu-luokan CheckSettingsPrefs-metodia, tarkistaa pelaajan PlayerPrefsiin tallennetut tiedot ja muuttaa pelin asetukset niiden mukaisesti.

Tietojen saaminen PlayerPrefsistä onnistuu siinä luokassa olevalla GetInt-metodilla, joka palauttaa nimenmukaisesti kokonaislukuarvoja. PlayerPrefs ei tue suoranaisesti totuusarvomuuttujia, joita ääni- ja musiikinäppäimen kanssa halutaan tallentaa, koska pelin pitää tietää, soittaako se ääniä ja musiikkia pelissä. PlayerPrefsin kanssa voi kuitenkin totuusarvojen tilalla käyttää kokonaislukuja nolla ja yksi, joista nolla tarkoittaa väärää ja yksi oikeaa. Esimerkkikoodista 4 nähdään osa koodista: jos esimerkiksi musiikki ei ole mykistetty, se mykistetään asettamalla sille ensin uusi arvo SetInt-metodilla, joka toimii PlayerPrefsin tallennusmetodina, ja luvun asettamisen jälkeen tarkistetaan luvun arvo ja mykistetään. Samaa hakemista käytetään myös äänien ja väriteeman indeksin kanssa, mutta väriteeman indeksi ei ole totuusarvo vaan kokonaisluku, joka sitten sijoitetaan omaan luokkaansa, joka valitsee väriteeman.

```
if(PlayerPrefs.GetInt("MusicIsMuted", 0) == 0)
{
    PlayerPrefs.SetInt("MusicIsMuted", 1);
}
```

Esimerkkikoodi 4. PlayerPrefs tarkistaa, että jos musiikki ei ole mykistetty, se mykistetään.

Paikallisen pistetaulukon tallentamisen tiedosto päätettiin tehdä JSON-muotoiseksi tekstitiedostoksi, koska sen käyttäminen ja muokkaaminen on helppoa ja se varastoi tiedot parhaista tuloksista ja päivämäärästä, jolloin tämä tulos saavutettiin. Pistetaulukolle ei tehty minkäänlaista suojaamista, koska sille ei koettu tässä tapauksessa tarvetta. Pistetaulukon tuloksia ei verrata kuin omiin tuloksiin, jolloin tuloksia muuttamalla pelaaja huijaisi vain itseään. Sovelluksen käynnistyessä peli yrittää etsiä tallennettuja tuloksia JSON-muotoisesta tiedostosta. Pelin löytäessä tiedoston se tuo tulokset pistetaulukkoon lukemalla JSON-tiedoston ja muuttamalla tiedot merkkijonoksi, mutta jos tiedostoa ei löydy, luokka luo tyhjän tiedoston.

Pistetaulukon tallentaminen aloitetaan GameManager-luokasta, kun peli on loppunut. GameManager kutsuu Leaderboard-luokassa olevaa AddEntry-metodia ja lähettää sen mukana pelin tuloksen ja päivämäärän. AddEntry ottaa tiedot ja vertaa pisteitä pistetaulukon listan tuloksiin käymällä sitä läpi korkeimmasta tuloksesta alaspäin. Jos pisteet ovat suurempia kuin jonkin listassa olevan, pisteet siirretään sen tilalle Insert-metodilla. Tämän jälkeen metodi tarkistaa, onko listan kaikki kymmenen paikkaa jo täytetty tuloksilla. Jos tässä vaiheessa listassa on 11 pistetulosta, viimeisenä oleva pistetulos poistetaan listasta, niin että pelaaja näkee edelleen vain parhaat kymmenen tulosta. Tämän jälkeen tulos taulu tallennetaan esimerkkikoodin 5 mukaisesti SaveScores-metodilla, joka ottaa tulostaulun listan, muuttaa listan JSON-muotoiseksi merkkijonoksi ja kirjoittaa tiedot tiedostoon.

```

void SaveScores(LeaderboardSaveData leaderboardSaveData)
{
    using (StreamWriter stream = new StreamWriter(localLeaderboard-
        Path))
    {
        string json = JsonUtility.ToJson(leaderboardSaveData, true);
        stream.Write(json);
    }
}

```

Esimerkkikoodi 5. Paikallisen tulostaulun tallentamisen metodi.

Pelien eri osien tallentamiseen on monia erilaisia keinoja. Tietokilpailupelissä käytettiin kolmea eri tekniikkaa eri osien tallentamiseen, koska kaikki osat toimivat eri tavoilla. Esimerkiksi asetuksissa muutaman totuusarvomuuuttujan tallentamiseen ei tarvita monimutkaisempaa binäärialustajaa, vaan Unityn PlayerPrefs riittää hyvin.

4.5 Monetisaatio

Tietokilpailupeli aiotaan julkaista Google Play -kauppaan, App Storeen ja verkkosivustolle. Tietokilpailusta päätettiin tehdä ilmainen peli, koska nykyään on monessa tilanteessa kannattavinta julkaista mobiilipelit ilmaisina ja käyttää muita ansaintamalleja tuottamaan pelillä tuloja ja saamaan pelaajia, ja verkkosivustolle julkaistaessa maksuton peli on järkevin. Tietokilpailupeliin ei päätetty tässä vaiheessa tehdä mitään pelin sisäisiä mikromaksuja, vaan pelissä keskitytään erilaisiin mainoksiin ansaintamalleina.

Unityn mainoksien implementointi peliin alkaa Unityn palvelut-ikkunasta, josta pitää laittaa mainoksien kohta päälle. Seuraavaksi peliin pitää tuoda Unityn mainostuspaketti, joka tuo projektiin käytettäväksi esimerkiksi kaikki luokat, joita koodin puolella mainosten toimimiseen tarvitsee. Viimeinen asia, joka pitää tehdä mainoksien eteen ennen ohjelmointia, on mennä projektin kojelaudalle Unityn sivuilla. Sieltä löytyy App Storeen ja Google Playta varten projektitunnisteet molemmille alustoille ja molemmille alustoille kolmen mainostyyppin tunnisteet, jotka pitää ottaa muistiin ohjelmointia varten. Kolme eri mainostyyppiä, jotka Unitylla on tarjota, ovat välimainos, palkitseva mainos ja bannerimainos, ja nämä kaikki päätettiin implementoida tietokilpailupeliin.

Pelissä on GameManager-objekti, jossa on kiinni AdManager-luokka, joka hoitaa mainoksien toiminnallisuuksiin liittyvät asiat. Luokan alussa alustetaan App Storen ja Google Playn tunnisteet ja molempien alustojen kolme mainostunnistetta, jotka otettiin projektin kojelaudalta muistiin. Tunnisteiden lisäksi luokkaan pitää alustaa totuusarvomuuttuja tarkistamaan, onko mainoksien testitila päällä. Totuusarvo pitää olla, koska ennen virallista julkaisua testitilan totuusarvomuuttujaa ei saa ottaa pois päältä, koska Unity saattaisi vahingossa näyttää oikeita mainoksia testivaiheessa, mikä voisi johtaa huomautuksiin Unitylta. Luokan Start-metodissa mainonnalle pitää lisätä kuuntelija mainoksien toimimiseksi, ja metodissa kutsutaan myös esimerkkikoodissa 6 nähtävää InitializeAdvertisement-metodia, joka alustaa mainonnan joko Google Playhin tai App Storeen riippuen halutusta kohteesta, joka on helppo tarkistaa totuusarvomuuttujalla. Metodi lisäksi tarkistaa, onko mainosten testitila päällä.

```
void InitializeAdvertisement()
{
    if (isTargetPlayStore)
    {
        Advertisement.Initialize(playStoreID, testMode);
        return;
    }
    Advertisement.Initialize(appStoreID, testMode);
}
```

Esimerkkikoodi 6. Mainoksien alustamiseen käytetty metodi.

Luokassa on hyödynnetty metodeja, jotka saa käyttöön luokan käyttäessä IUnityAdsListener-rajapintaa. Näihin metodeihin kuuluu OnUnityAdsReady, OnUnityAdsDidStart, OnUnityAdsDidFinish ja OnUnityAdsDidError. OnUnityAdsReady kutsutaan mainoksien alustamisen jälkeen, kun ne ovat käytävissä, ja sitä käytetään pelissä bannerimainoksen alustamiseen. OnUnityAdsDidStart kutsutaan mainoksen alussa, ja sillä voidaan esimerkiksi mykistää pelin äänet mainoksen ajaksi. OnUnityAdsDidFinish kutsutaan, kun mainos loppuu, ja sen sisälle on hyvä tehdä tarkistus, epäonnistuiko, ohitettiin tai katsottiin mainos loppuun. Epäonnistuessa mainos voidaan yrittää näyttää uudelleen, ohitetussa voidaan vain muuttaa totuusarvomuuttujaa, että mainos on näytetty, ja jos mainos meni loppuun asti ja se on palkitseva mainos,

pelaajalle voidaan antaa palkinto. OnUnityAdsDidError kutsutaan, kun mainoksessa tapahtuu virhe, ja sillä voi lähettää kehittäjälle virhelokin.

Mainoksien näyttämiseen luokassa pitää olla käytössä UnityEngine.Advertisement-nimiavaruus, mutta metodit niiden näyttämiseksi on hyvä tehdä itse, jotta niitä pystytään kutsumaan toisista skripteistä helposti. Tietokilpailupeliin haluttiin bannerimainos näkyviin, ja sen paikaksi päätettiin ruudun alareuna, koska siellä se häiritsee pelaajaa vähiten. Tämä toteutettiin mainoksien valmiilla Show-metodilla, jolla mainokset pystytään näyttämään, ja SetPosition-metodilla, joka asettaa bannerin haluttuun paikkaan ruudulla. Välimainoksien ja palkitsevien mainoksien näyttäminen toimivat melkein identtisesti lukuun ottamatta sitä, että Show-metodin parametreihin tulee sen mainoksen tunnus, mikä halutaan näyttää. Esimerkkikoodissa 7 on palkitsevan mainoksen näyttämiseksi metodi, jossa tarkistetaan alusta ja se, onko mainos valmiina. Jos mainos on valmis, pelissä näytetään Androidin palkitseva mainos.

```
public void ShowRewardedVideo()
{
    if(isTargetPlayStore)
    {
        if(!Advertisement.IsReady(rewardedAndroidAd))
        {
            return;
        }
        Advertisement.Show(rewardedAndroidAd);
    }
    if(!isTargetPlayStore)
    {
        if(!Advertisement.IsReady(rewardedIOSAd))
        {
            return;
        }
        Advertisement.Show(rewardedIOSAd);
    }
}
```

Esimerkkikoodi 7. Palkitsevan mainoksen näyttämisen metodi.

Koska GameManager-luokka hoitaa melkein kaikki pelissä olevat toiminnot, myös mainoksien näyttämiseen tulevat kutsut tehdään siellä. Tietokilpailupelissä pelaajaa ei haluta häiritä mainoksilla kysymysten välissä, joten välimainokset näytetään aina, kun kysymyssarja on saatu päätökseen. Viimeisen

kysymyksen jälkeen peli siirtyy tilaan 3, jonka alussa koodi tarkistaa, onko pelaajalle näytetty mainos AdManagerissa olevalla wasAdPlayed-totuusarvomutujalla, ja jos ei ole, kutsutaan ShowInterstitialAd-metodia, joka näyttää välimainoksen. Välimainos näytettäessä wasAdPlayed muutetaan todeksi, jotta koodi jatkaa tästä eteenpäin.

Tietokilpailupeliin on myös implementoitu palkitseva mainos kysymyssarjojen väliin esimerkkikoodin 8 mukaisesti. Jos pelaaja on vastannut liikaa väärin, näkymään ilmennetään painike palkitsevalle mainokselle ja lisätietoja suorituksesta. Pelaaja voi tästä vain jatkaa peliä putoamalla alemmalle tasolle, mutta painiketta painamalla kutsua AdManagerin ShowRewardedVideo-metodia, joka tarkistaa alustan ja näyttää palkitsevan mainoksen. Jos mainos katsotaan kokonaan, AdManager kutsuu GiveReward-metodia, joka muuttaa playerIsRewarded-muuttujan todeksi. Tämä siirtää koodin esimerkin if-lauseesta else-kohtaan, mikä muuttaa näytettävää tekstiä, kadottaa mainospainikkeen ja peli jatkuu samasta tilanteesta.

```

if (level > 1)
{
    if (AdManager.playerIsRewarded == false)
    {
        roundEndText.text = "Vastasit liikaa väärin.";
        stayInRoundButton.SetActive(true);
        if (continuedToNextRound)
        {
            Level--;
            NewRound();
        }
    }
    else
    {
        roundEndText.text = "Kokeile uudelleen.";
        stayInRoundButton.SetActive(false);
        if (continuedToNextRound)
        {
            NewRound();
        }
    }
}

```

Esimerkkikoodi 8. Palkitsevan mainoksen näyttämisen tarkistus ja muutos.

Unityn mainoksien implementoiminen on suhteellisen yksinkertaista, vaikka monia asioita pitääkin monissa eri paikoissa asettaa ja tiettyjä asioita muistaa.

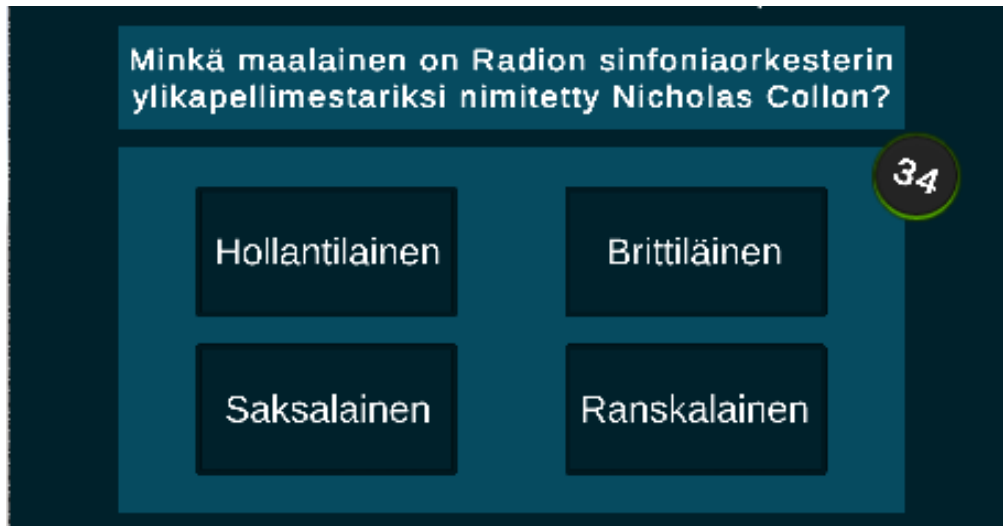
Unity tarjoaa kaikki perinteiset ansaintamallit mainoksien osalta, ja haluttaessa

peleihin olisi mahdollista myös lisätä Unityn kautta pelin sisäisiä ostoksia. Unityn mainoksia pystyy myös helposti muokkaamaan sellaiseksi kuin haluaa, mikä on pelinkehittäjälle hyvä asia. Esimerkiksi välimainoksissa voi valita sekunteina, milloin mainoksen ohittaminen on mahdollista ja millaisia mainoksia pelissä näytetään.

4.6 Alustojen erot

Tietokilpailupeli toteutettiin kolmelle alustalle, jotka ovat Android, iOS ja WebGL. Android ja iOS ovat mobiilialustoja, jotka toimivat pääosin samoilla tavoilla pelin rakentamista lukuun ottamatta. WebGL on verkkoselainalusta, joka ei kaikissa asioissa toimi samalla tavalla tai ole samanlainen kuin mobiiliversio.

Käyttöliittymän osalta WebGL- ja mobiilialustojen suurin ero on ulkonäössä, koska mobiiliversio on lukittu pystyasentoon ja WebGL on tavallisen näytön tavoin vaakasuunnassa. Pelin WebGL-versio rakennettiin suoraan mobiiliversion pohjalta, mutta tiettyjen elementtien paikkoja vähän muutettiin ruutujen ollessa eri orientaatioissa. Isoin ero käyttöliittymän osalta alustoissa on pelin kysymysnäkyssä, ja kuvasta 12 on nähtävissä verkkoselainversion kysymysnäky. Vastausvaihtoehtopainikkeet on aseteltu ruudukkomallisesti eikä mobiiliin tavoin pystymallisesti, koska WebGL-versiossa leveyttä ja tilaa on enemmän.



Kuva 12. WebGL-alustan kysymysnäkyvä.

Tietokilpailupelin toiminnallisuudet on suunniteltu pääasiallisesti toimimaan kaikilla alustoilla yhtenäisesti, koska tällä tavalla saadaan peli pidettyä helposti muokattavana ja peliä päivittäessä kaikki versiot päivittyvät samalla kertaa. Tallennustekniikat toimivat myös samoilla koodeilla mobiili- ja verkkoalustoilla, vaikka tiedostojen tallennuspaikat tietenkin eroavat. Monetisaatio ei kuitenkaan toimi verkkoselaimella samalla tavalla kuin mobiiliversiossa. Syynä tähän on se, että Unityssa implementoidut mainokset eivät toimi kuin pelkästään mobiilialustoilla. Tästä syystä pelissä kysymyssarjojen välissä olevaa apupainiketta, jolla pelaaja voi mainoksen katsomalla jatkaa peliä samasta tilanteesta, ei voida käyttää, vaan sen pitää verkkoselainpelissä olla poissa. Verkkoselaimeen julkaistun tietokilpailupelin kanssa ei voi siis pelillä itsellään saada tuottoa vaan tuotot pitäisi saada jotain muuta kautta, esimerkiksi sivulla mainostamisesta.

4.7 Tietokanta

Tietokilpailupelille asennettiin, alustettiin ja toteutettiin etäpalvelimelle MariaDB:n tarjoama relaatiotietokantajärjestelmä varastoimaan kaikki pelissä käytettävät kysymykset. MariaDB:n toimintaan saaminen aloitettiin etäpalvelimella MariaDB-palvelimen asentamisella komentoikkunan kautta. Asentamisen aikana MariaDB asentaa kaikki tarvitsemansa tiedostot. CREATE DATABASE -

komennolla saatiin luotua kysymystietokanta palvelimelle, ja CREATE USER -komennolla saadaan luotua tietokannalle käyttäjä. Use-komennolla saadaan haluttu tietokanta käyttöön, minkä jälkeen CREATE TABLE -komennolla saadaan luotua taulukko kysymyksiä varten. Taulukon luomisen yhteydessä asetetaan taulukon sarakkeiden nimet ja sisällön tyyppi, ja muita tarkentavia tietoja pystyy antamaan, kuten id:n arvolle auto_increment, joka luo kysymyksille automaattisesti tunnustearvon. Kuvassa 13 on pelin kysymystietokannan taulukon rakenne.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
kysymys	varchar(300)	NO		NULL	
vaihtoehto1	varchar(50)	NO		NULL	
vaihtoehto2	varchar(50)	NO		NULL	
vaihtoehto3	varchar(50)	YES		NULL	
vaihtoehto4	varchar(50)	YES		NULL	
vaihtoehto5	varchar(50)	YES		NULL	
vaihtoehto6	varchar(50)	YES		NULL	
vaihtoehto7	varchar(50)	YES		NULL	
vaihtoehto8	varchar(50)	YES		NULL	
vaihtoehto9	varchar(50)	YES		NULL	
vaihtoehto10	varchar(50)	YES		NULL	
oikea_vastaus	int(11)	NO		NULL	
vaikeusaste	int(11)	NO		NULL	
aihepiiri	varchar(40)	NO		NULL	
info	varchar(300)	YES		NULL	
kommentti	varchar(300)	YES		NULL	
ei_kaytossa	tinyint(1)	NO		0	
tarkkailussa	tinyint(1)	NO		0	
pvm	datetime	NO		NULL	

Kuva 13. Kysymystietokannan taulukon rakenne.

Tietokannan kysymyksien toimimiseksi pelissä niille pitää olla annettu kysymys, vähintään kaksi vastausvaihtoehtoa, oikean vastauksen numero, vaikeusaste ja aihepiiri. Muut arvot ovat joko vapaaehtoisia tai tulevat automaattisesti. Vaikka vain kaksi vastausvaihtoehtoa on minimimäärä ja kymmenen maksimi, yleensä kysymyksillä on neljä vastausvaihtoehtoa. Kysymyksien lisääminen tietokantaan hoidetaan etäpalvelimellä olevan tietokantaeditoriohjelman kautta, jossa kysymyksiä voi lisätä ja muokata, eikä suoraan tietokannassa. Taulukossa olevat kysymykset saadaan valittua SELECT-komennolla, jonka perään voidaan kirjoittaa, mitä tietoja taulukosta halutaan, ja tämän jälkeen FROM-komennolla kerrotaan, mistä taulukosta tietoja halutaan. Taulukko on tässä tapauksessa

kysymysten taulukko. Tästä syystä myös Unityn puolella pelissä kysymysten hakemisen lauseke toimii samalla periaatteella.

5 Pohdinta

5.1 Tulokset ja ongelmakohdat

Insinööriyön tavoitteena oli suunnitella ja luoda toimeksiantajalle tietokilpailupeli kaupalliseen tarkoitukseen kolmelle eri alustalle. Lopputuloksena tietokilpailupelistä saatiin luotua toimiva kokonaisuus. Tietokilpailupelissä on kaikki tarvittavat ja halutut näkymät pelin toimimiseksi, ja käyttöliittymä on halutun yksinkertainen ja helppokäyttöinen. Pelin pelaaminen ja sen toiminnot kysymyksiin vastaamisesta kysymysten hakemiseen tietokannasta toimivat toivotusti. Tallentaminen toimii kaikilla osa-alueilla ja myös eri alustoilla, pelin mainonnan toiminnot toimivat odotetusti eli pelillä voi tuottaa tuottoja, ja pelin kysymystietokannasta saatiin rakennettua yksinkertainen, mutta toimiva.

Tietokilpailupelin kehityksessä selvästi isoimmaksi ongelmaksi muodostui etäpalvelimella olevan kysymystietokannan saaminen yhteyteen Unityn kanssa. Etäpalvelimella olevat tietokannan yhteyteen liittyvät tiedostot piti saada julkisesti nähtäväksi, jotta Unity pystyy verkkoselaimen kautta ottamaan niihin yhteyttä. Tämän ratkaisemisessa meni pidemmän aikaa, mutta onneksi se lopulta saatiin toimintaan ja tämän jälkeen yhteyden ottaminen tietokantaan ja sen tietoihin oli helppoa.

Tietokilpailu-pelisovelluksen luomisessa oli myös omat vaikeutensa. Pelinäkömässä toimiva tilakone ja sen toimintojen luominen ja hiominen toimivaksi vaati paljon yrityksiä ja erehdyksiä. Vaikka tilakone ajatuksena on yksinkertainen, tietokilpailupelin tilakoneessa on todella monia eri osia, koska se hoitaa koko pelin toiminnallisuudet ja paljon erilaisia tarkistuksia, joita kaikkia piti kokeilla useita kertoja virheiden löytämiseksi. Pelin edistyessä peliin lisättiin toiminnallisuuksia, mikä usein saattaa tuoda lisää virheitä. Se tarkoitti lisää kokeilemistä.

Tietokilpailupelissä omat ongelmansa toivat myös eri näkymien hallitseminen ja käyttöliittymän suunnittelu. Pelissä on paljon erilaisia näkymiä, koska se on käyttöliittymäpohjainen, mikä välillä pelin kehityksen aikana koitui ongelmaksi. Jos koodissa tai Unityn editorissa unohtui laittaa tiettyjä näkymiä päälle tai pois päältä, pelin kokeileminen piti aina aloittaa uudelleen, mikä oli välillä turhauttavaa, koska esimerkiksi pelin loppunäkymää muokatessa piti uudelleen käydä koko peli läpi päästäkseen siihen pisteeseen pelissä, että toimintoja pystytään kokeilemaan. Tämä ratkaistiin pääasiassa tekemällä tietyissä paikoissa koodiin oikoreittejä ja helpotuksia nopeuttamaan testaamista. Käyttöliittymän suunnittelussa ja tekemisessä selkeästi isoin ongelma oli miettiä, miten kaikki tarvittavat tiedot saadaan pelaajalle mobiiliversiossa ruudun ollessa suhteellisen pieni. Aluksi mietittiin lyhenteitä, mutta se olisi jäänyt liian epäselväksi, joten päädyttiin pienentämään kysymys- ja vastausvaihtoehtolaatikoita tilan tekemiseksi.

5.2 Jatkokehitys

Insinööriyössä tehdystä pelistä saatiin luotua toimiva kokonaisuus, mutta sen kehitystä ei vielä lopetettu, vaan jatketaan insinööriyön ulkopuolella. Tulevaisuudessa tärkeimpänä tavoitteena on saada tietokilpailupeli julkaistua Google Play -kauppaan, App Storeen ja verkkosivulle. Ennen julkaisua kuitenkin peliä on vielä tarkoitus kehittää pidemmälle ja paremmaksi.

Tietokilpailupelille on tarkoitus luoda kysymystietokannan rinnalle myös pelaaja- ja tulostietokanta. Pelaajatietokannan tarkoituksena on antaa pelaajalle mahdollisuus pelata peliä rekisteröityneenä. Pelaajatietokantaa hyödyntäen pelaaja voisi pystyä jatkamaan peliään riippumatta laitteesta, jolla hän pelaa, koska tallennustiedostot voisi tallentaa pelaajatunnuksen avulla tietokantaan. Toinen pelaajatietokantaan liittyvä asia olisi tulostietokanta, joka voitaisiin liittää pelaajatietokannan yhteyteen. Parhaiden tuloksien tallentamista voitaisiin tehdä tietokantaan tietyillä pelaajatunnuksilla, jolloin pelaajat pystyisivät paikallisen tulostaulun lisäksi vertailemaan itseään muihin pelaajiin.

Tietokilpailupeliin on myös tarkoitus tehdä yleisiä parannuksia Unityn puolella. Parannuksiin kuuluisi esimerkiksi parempien äänitehosteiden ja musiikin etsiminen. Peliin olisi myös tarkoitus lisätä animaatioita eri näkymien välille, koska animaatioiden avulla pelistä saadaan helposti mukavampi pelata ja pelistä tulee viimeistellympi olo, kun eri ruutuja tai näkymiä ei vain katoa ja ilmesty pelaajan edessä välähtämällä. Vaikka tietokilpailupeliin tehtiin toimivat mainokset, pelille mietitään vielä mahdollisia lisäyksiä niihin ja myös mahdollisuutta saada peliin luotua pelin sisäisiä mikromaksuja tuottojen kasvattamiseksi.

Viimeisenä ajatuksena pelin jatkolle on pelin julkaiseminen. Pelin jakamiseksi Google Play -kauppaan pitää Google Playhin hankkia kehittäjätili olemassa olevan Google-tilin avulla. Kehittäjätilin avaaminen maksaa 25 dollaria, minkä jälkeen tilillä pystytään julkaisemaan ja hallinnoimaan sovelluksia kehittäjäkonso- lissa. Pelin julkaisemiseksi App Storeen pitää rekisteröityä Apple-kehittäjäksi, samalla tavalla kuin Google Play -kauppaan. Applen kehittäjätilin avaaminen maksaa 99 dollaria, ja se pitää uusia vuosittain. Applen kehittäjätilin avaaminen antaa myös mahdollisuuden testata sovelluksia Applen laitteilla. WebGL-sovel- luksen julkaisemiseen tarvitaan vain verkkosivu, joka tukee JavaScriptiä. WebGL-sovellukset toimivat kaikilla yleisimmillä verkkoselaimilla, kuten Google Chrome, Firefox ja Safari.

6 Yhteenveto

Insinööriyössä luotiin toimeksiantajalle kaupalliseen tarkoitukseen sopiva tietokilpailupeli mobiili- ja verkkoselainaluustoille hyödyntäen Unity-pelimootoria ja MariaDB-relaatiotietokantajärjestelmää. Tämän lisäksi työssä tutustuttiin videopelien historiaan sekä tietokilpailupeleihin osana peliteollisuutta, niiden markkinoihin ja ansaintamalleihin. Projektin alkuvaiheessa mietittiin parasta tapaa pelin luomiseen ja päädyttiin käyttämään Unitya, koska se mahdollistaa pelin luomisen monelle eri alustalle ilman suurempia ongelmia ja se oli tekijälle muista opiskeluprojekteista jo valmiiksi tuttu.

Tietokilpailupeli on peli, jossa pelaaja yrittää vastata monen eri aihealueen kysymyksiin oikein samalla kilpaillen. Tietokilpailupeli tyylilajina ei ole tällä hetkellä suosituimpia, mutta niille on silti aina kysyntää ja omat markkinat. Peliteollisuus on kasvanut yhdeksi isoimmaksi viihdeteollisuuden osaksi ja jatkaa vuosittain kasvuaan. Mobiilipeleistä etenkin on muodostunut lyhyessä ajassa pelimarkkinoiden reilusti tuottoisin osa, koska mobiililaitteet ovat kehittyneet todella nopeasti tehokkaiksi laitteiksi, ja koko ajan enemmän isoja yrityksiä lähtee kehittämään mobiilipelejä. Mobiilipeleissä käytetään nykyään ansaintamallia, jonka mukaan pelit julkaistaan ilmaisena ja tuotot tuotetaan pääasiallisesti mainonnalla ja myös mikromaksuilla. Unity-pelimoottori on suosituin pelimoottori, ja se antaa laajat mahdollisuudet pelinkehitykseen etenkin monille eri alustoille. MariaDB tarjoaa halvan, hyvän ja helppokäyttöisen relaatiotietokantajärjestelmän, joka toimii hyvin Unityn kanssa yhdessä.

Projekti onnistui tavoitteiden mukaisesti, koska tietokilpailupelistä saatiin yleisesti toimiva kokonaisuus. Tietokilpailupelin kehittämistä kuitenkin jatketaan vielä ennen tarkoitettua julkaisua lisäämällä uusia ominaisuuksia ja viimeistelemällä tiettyjä osia. Työ tarjoaa pelinkehityksestä kiinnostuneille tietoa pelien markkinoista, hyvistä ansaintamalleista, nykyaikaisesta peliohjelmoinnista Unitylla ja MariaDB-tietokannan hyödyntämisestä Unityn kanssa.

Lähteet

- 1 Video Game History. Verkkoaineisto. Smithsonian. <<https://www.si.edu/spotlight/the-father-of-the-video-game-the-ralph-baer-prototypes-and-electronic-games/video-game-history>>. Luettu 14.4.2022.
- 2 Tennis For Two at Brookhaven NL, 1958. Verkkoaineisto. Pong-museum.com. <<https://commons.wikimedia.org/wiki/File:TennisForTwo-Machine.jpg>>. Luettu 14.4.2022.
- 3 What are the Most Popular Online Game Genres in 2021? 2021. Verkkoaineisto. Full Press Coverage. <<https://fullpresscoverage.com/2021/06/24/what-are-the-most-popular-online-game-genres-in-2021/>>. 25.6.2021. Luettu 15.4.2022.
- 4 Knezovic, Andrea. 2022. 147 Mobile Gaming Statistics for 2022 That Will Blow Your Mind. Verkkoaineisto. Udonis. <<https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics>>. 24.3.2022. Luettu 15.4.2022.
- 5 Pai Raikar, Sanat. Quiz. Verkkoaineisto. Britannica. <<https://www.britannica.com/topic/quiz>>. Luettu 17.4.2022.
- 6 Älypää. 1998. Sanoma Media Finland Oy.
- 7 Wijman, Tom. 2021. The Games Market and Beyond in 2021: The Year in Numbers. Verkkoaineisto. Newzoo. <<https://newzoo.com/insights/articles/the-games-market-in-2021-the-year-in-numbers-esports-cloud-gaming/>>. 22.12.2021. Luettu 17.4.2022.
- 8 Trivia Games. Verkkoaineisto. Statista. <<https://www.statista.com/outlook/dmo/app/games/trivia-games/finland?currency=EUR>>. Luettu 18.4.2022.
- 9 Knezovic, Andrea. 2022. Mobile Game Monetization Trends: Best Strategies to Monetize Your Game in 2022. Verkkoaineisto. Udonis. <<https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-game-monetization-trends>>. 12.1.2022. Luettu 21.4.2022.
- 10 Clash Royale. 2016. Supercell.
- 11 Knezovic, Andrea. 2022. Increase Your In-Game Advertising Revenue with These Mobile Game Ad Formats. Verkkoaineisto. Udonis. <<https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-game-ad-formats>>. 25.1.2022. Luettu 21.4.2022.

- 12 Knezovic, Andrea. 2022. eCPMs Revealed! Check out Top Ad Monetization Networks for Apps & Games in 2022. Verkkoaineisto. Udonis. <<https://www.blog.udonis.co/mobile-marketing/mobile-apps/ecpms>>. 16.1.2022. Luettu 22.4.2022.
- 13 Create and grow more with Unity. Verkkoaineisto. Unity Technologies. <<https://unity.com/>>. Luettu 13.10.2021.
- 14 Welcome to Unity. Verkkoaineisto. Unity Technologies. <<https://unity.com/our-company>>. Luettu 13.10.2021.
- 15 Brodtkin, Jon. 2013. How Unity3D Became a Game-Development Beast. Verkkoaineisto. Dice. <<https://insights.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/>>. 3.6.2013. Luettu 13.10.2021.
- 16 Plans and pricing. Verkkoaineisto. Unity Technologies. <<https://store.unity.com/>>. Luettu 13.10.2021.
- 17 Irpan, Emory; Gohil, Arjun & TenBoer, Nathan. 2021. 2021 Gaming Report. Verkkoaineisto. Unity Technologies. <https://images.response.unity3d.com/Web/Unity/%7B4eb56531-e6aa-492f-8fda-c68ae20af950%7D_2021_Gaming_Report_-_Operate_Solutions.pdf>. Luettu 15.10.2021.
- 18 Pearce, Rohan. 2013. Dead database walking: MySQL's creator on why the future belongs to MariaDB. Verkkoaineisto. Computerworld. <https://www2.computerworld.com.au/article/457551/dead_database_walking_mysql_creator_why_future_belongs_mariadb/>. 28.3.2013. Luettu 7.4.2022.
- 19 About MariaDB Server. Verkkoaineisto. MariaDB Foundation. <<https://mariadb.org/about/>>. Luettu 7.4.2022.
- 20 MariaDB versus MySQL – Features. Verkkoaineisto. MariaDB. <<https://mariadb.com/kb/en/mariadb-vs-mysql-features/>>. Luettu 7.4.2022.
- 21 MariaDB Community Server. Verkkoaineisto. MariaDB. <<https://mariadb.com/products/community-server/>>. Luettu 7.4.2022.
- 22 MariaDB Enterprise. Verkkoaineisto. MariaDB. <<https://mariadb.com/products/enterprise/>>. Luettu 7.4.2022.
- 23 SkySQL. Verkkoaineisto. MariaDB. <<https://mariadb.com/products/skysql/>>. Luettu 10.4.2022.

- 24 Pricing. Verkkoaineisto. MariaDB. <<https://mariadb.com/pricing/>>. Luettu 7.4.2022.

