

Lisätty todellisuus

Reality Check -peli

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2016
Leo Lastikka

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

LASTIKKA, LEO:

Lisätty todellisuus
Reality Check -peli

Ohjelmistotekniikan opinnäytetyö, 40 sivua

Kevät 2016

TIIVISTELMÄ

Opinnäytetyössä käsitellään lisättyä todellisuutta ja sitä hyödyntävän sovelluksen kehittämistä Unity3D-pelimootorilla Vuforia-kirjaston avulla. Tarkoituksena oli kehittää Lahden kaupunginkirjastolle lisättyä todellisuutta hyödyntävä sovellus, jonka avulla koululaisryhmien kirjastovierailuista saataisiin mielenkiintoisempia. Teknisestä näkökulmasta sovelluksen haasteena oli tehdä siitä sellainen, että sen sisältöä voisi ylläpitää ja muokata verkkopalvelimella sijaitsevasta hallintapaneelistä.

Lahden kaupunginkirjastolla on käynnissä hanke, jonka tarkoituksena on tutkia ja testata lisätyn todellisuuden käyttämistä kirjaston palveluissa sekä tuoda sitä yleisön tietoisuuteen. Hanke on tarkoitettu kaikkien Päijät-Hämeen kirjastojen eli LASTU-kirjastojen käyttöön. Tarkoituksena on kehittää sovelluksia, jotka toimivat Android- ja iOS-laitteilla sekä tarjota niitä ilmaiseksi asiakkaille. Hankkeeseen kuuluu myös kirjaston oman henkilöstön koulutus lisätyn todellisuuden kanssa toimimiseen.

Opinnäytetyössä käsitellään myös lisätyn todellisuuden historiaa ja nykypäivän sovelluksia. Lisätyn todellisuuden mahdollistavaa teknologiaa on ollut olemassa jo suhteellisen pitkään, mutta etenkin viime aikoina, tekniikan kehityksen myötä, sovellusten määrä on lisääntynyt paljon. Entistä suurempi sovellusten määrä kannustaa myös uusia kehittäjiä tutustumaan lisättyyn todellisuuteen, ja sen tulevaisuus näyttääkin lupaavalta.

Projektiryhmässä oli Lahden ammattikorkeakoulusta kolme ohjelmoijaa ja kuusi graafikkoa sekä kirjaston työntekijöitä. Ohjelmoijat olivat vastuussa sovelluksen ja hallintapaneelin suunnittelusta sekä toteutuksesta. Graafikot suunnittelivat sovelluksen käyttöliittymän ja sisällön. Kirjaston edustajat tarjosivat tätä projektia varten hankkimansa materiaalit sovellukseen kuuluvia tarinoita varten. Projektista saatiin tehtyä testauskelpoinen pilottiversio, mutta kehitystyö jatkuu.

Asiasanat: lisätty todellisuus, Vuforia, Unity3D

Lahti University of Applied Sciences
Degree Programme in Information Technology

LASTIKKA, LEO:

Augmented Reality
Reality Check Game

Bachelor's Thesis in Software Engineering, 40 pages

Spring 2016

ABSTRACT

This thesis deals with augmented reality and the implementation of an application that uses the Unity3D game engine and the Vuforia library. The purpose of the work was to create an augmented reality application for the Lahti City Library to make schoolchildren's visits more interesting. From a technical viewpoint the challenge was to create an application that could be maintained and modified by a management console on a web server.

Lahti City Library is working on a project with the purpose of studying and testing the usage of augmented reality in library services and bringing it to the awareness of the public. The project is meant to be used by all of the Lahti region libraries. The goal is to create applications that work on Android and iOS devices and provide them to the clients for free. The project also aims to train the personnel of the libraries to work with augmented reality.

This thesis also studies the history and modern applications of augmented reality. The technology that makes augmented reality possible has existed for relatively long time but especially recently, because of the advancement in technology, the number of these applications has increased greatly. The increasing number of these applications also encourages new developers to study augmented reality and its future seems promising.

The project group consisted of three programmers and six graphic designers from Lahti University of Applied Sciences and some personnel from the Lahti City Library. The programmers were responsible for the design and implementation of the application as well as the management console. Graphic designers were responsible for the user interface and the content of the application. The contacts in the library offered material that they had gathered for the stories in the application. The project resulted in a testable pilot version but the development continues.

Key words: augmented reality, Vuforia, Unity3D

SISÄLLYS

1	JOHDANTO	1
2	KIRJASTON TOIMINTAYMPÄRISTÖ	3
3	LISÄTTY TODELLISUUS	5
3.1	Määritelmä	5
3.2	Historia	5
3.2.1	Sensorama	6
3.2.2	Ensimmäinen päässä pidettävä virtuaalitodellisuuslaite	7
3.2.3	Videoplace	8
3.2.4	ARToolkit	9
3.3	Teknologia	9
3.3.1	Päässä pidettävät näytöt	10
3.3.2	Kädessä pidettävät näytöt	11
3.3.3	Projektionäytöt	11
3.4	Käyttökohteet	12
3.4.1	Mainonta	12
3.4.2	Arkkitehtuuri	13
3.4.3	Navigaatio	13
3.4.4	Opetus	14
3.4.5	Lääketeiede	14
3.4.6	Peliteollisuus	14
4	UNITY	15
4.1	Lisenssit	15
4.2	WWW ja WWWForm	16
4.3	IEnumerator ja yield	17
4.4	JsonUtility	18
4.5	UI	19
5	VUFORIA	23
5.1	Lisenssit	23
5.2	Kohteiden hallinta	24
5.3	Web-rajapinta	25
5.4	Erilaiset tunnistuskohteet	26
5.5	Tekstintunnistus	29

6	SOVELLUKSEN TOTEUTUS	31
6.1	Kehitysympäristön alustaminen	31
6.2	Palvelin ja hallintapaneeli	32
6.3	Tietokanta	33
6.4	Sovelluksen ja palvelimen rajapinta	34
6.5	Sessiot	36
6.6	Polut ja rastit	36
6.7	Rastien lataaminen sovellukseen	37
6.8	Rastien sisällön näyttäminen	37
7	YHTEENVETO	40
	LÄHTEET	41
	LIITTEET	46

LYHENTEET

GET	Metodi, jota käytetään usein tiedon hakemiseen web-sivustolta.
HTTP	<i>Hypertext Transfer Protocol</i> on protokolla, jota web-palvelimet käyttävät tiedonsiirtoon.
JSON	Tekstimuotoinen esitys olio- tai taulukkorakenteesta.
Luokka	Luokka tarkoittaa olio-ohjelmoinnissa mallia, josta voidaan luoda sen kaltaisia ilmentymiä.
Olio	Luokan ilmentymä.
POST	Metodi, jota käytetään usein tiedon lähettämiseen web-palvelimelle.
QR-koodi	<i>Quick Response</i> -koodi on kaksiuotteinen viivakoodin kaltainen kuva, jossa mustavalkoiseen ruudukkoon voidaan tallentaa informaatiota.
UI	<i>User Interface</i> eli käyttöliittymä
URL	<i>Uniform Resource Locator</i> eli merkkijono, jolla kerrotaan tietyn tiedon yksikäsitteinen paikka.
XML	<i>Extensible Markup Language</i> on merkitsemiskieli, jolla voidaan tuottaa standardimuotoiltu tietorakenne.

1 JOHDANTO

Lisätty todellisuus on saanut viime aikoina näkyvyyttä erityisesti mobiililaitteiden yleistymisen sekä niiden teknologian nopean kehittymisen ansiosta. Uusi tekniikka mahdollistaa todellisemman tuntuisten sovellusten kehittämisen kuin aikaisemmillä tekniikoilla on ollut mahdollista. Tämä on saanut myös sovelluskehittäjät kiinnostumaan aihepiiristä aiempaa enemmän, koska nykyisen teknologian avulla voidaan kehittää jo hyvin monipuolisia sovelluksia.

Lahden kaupunginkirjaston toiminnasta vastaa Lahden kaupungin sivistystoimiala. Kirjastotoiminta kuuluu osaamisen ja tiedon vastuualueeseen, ja sen avulla edistetään esimerkiksi tietoyhteiskunnan vaatimia kansalaistaitoja. Lahden pääkirjastossa käy tällä hetkellä noin 3000 asiakasta päivässä ja sivukirjastoissa yhteensä noin 1500. Lahden kaupunginkirjaston lainausmäärät ovat laskeneet kahdesta miljoonasta reiluun puoleentoista miljoonaan neljäntoista viime vuoden aikana, mutta kirjasto yrittää säilyttää asiakaskuntansa uudistumalla.

Tavoitteena on kehittää Lahden kaupunginkirjastolle sovellus, jonka avulla siellä vierailevien kouluryhmien tutustumiskäynneistä saataisiin mielenkiintoinen kokemus eri-ikäisille oppilaille. Sovelluksen erityispiirteenä on sen sisällön tuottaminen ja hallinta verkkopalvelimelta käsin. Tavoitteena on myös selvittää sovelluksessa käytettävien tekniikoiden lisenssejä.

Opinnäytetyössä tutustutaan ensin lisättyyn todellisuuteen käsitteenä. Lisättyä todellisuutta esitellään lisää tutustumalla sen historiaan ja teknologioihin, joita siinä hyödynnetään. Tämän jälkeen käsitellään vielä erilaisia sovellusalueita, jotka hyödyntävät lisättyä todellisuutta.

Seuraavaksi opinnäytetyössä käsitellään kirjastolle tehdyssä sovelluksessa käytettyjä teknologioita. Näihin lukeutuvat Unity3D-pelimootori sekä Vuforia, joka on lisätyn todellisuuden toiminnallisuuksia sisältävä kirjasto. Unitystä esitellään vain sovelluksen kehityksen kannalta olennaisimmat asiat. Vuforiasta käsitellään sovelluksessa vaadittujen asioiden lisäksi myös sen muita ominaisuuksia, koska Vuforia on saanut lisää ominaisuuksia.

sia viime vuosien aikana ja on nyt yksi käytetyimpiä lisätyn todellisuuden kirjastoja.

Tämän jälkeen opinnäytetyössä käsitellään itse sovelluksen kehitystyö. Kehitystyöhön kuului kehitysympäristön alustaminen, suunnittelua ja itse toteutus. Kehitystyön lisäksi esitellään sovelluksen toimintaperiaatetta.

Viimeisessä luvussa on opinnäytetyön yhteenveto, jossa tiivistetään sovelluksen onnistuminen ja pohditaan lisätyn todellisuuden merkitystä nykyään ja sen tulevaisuuden mahdollisuuksia.

2 KIRJASTON TOIMINTAYMPÄRISTÖ

Lahden koulujen opetussuunnitelmaan ovat kuuluneet jo pitkään vierailukäynnit kirjastoihin. Tällä hetkellä tutustuminen tapahtuu perinteisesti tiedon etsintään perustuvien kirjallisten tehtävien avulla. Koululaiset kiertävät kirjaston tiloihin asetelluilla rasteilla ja ratkaisevat siellä olevia tehtäviä. Tehtävissä pitää esimerkiksi etsiä teoksiin liittyviä tietoja kirjaston omasta aineistohausta.

Tällä hetkellä koululaisryhmän ohjaaja ei tiedä, missä vaiheessa tehtäviä kukin opiskelija on menossa tai mitkä tehtävät ovat hankalia tai vaikeita löytää. Koululaisten tekemien vastausten läpi käyminen on työlästä, ja sille ei ole aina tarpeeksi aikaa, koska jokaiselle ryhmälle varattu aika on rajallinen. Osa koululaisista saattaa myös kokea vanhanaikaiset tehtävät ikävyyttäväiksi, mikä johtaa siihen, että kirjastossa käyminen ei tunnu luontevalta.

Lahden kaupunginkirjastolla on työn alla hanke, jonka tavoitteena on tutkia lisätyn todellisuuden käyttämistä kirjaston palveluissa sekä opettaa sitä työntekijöille ja tuoda sitä tunnetuksi yleisölle. Yksi hankkeen konkreettisista toteutuksista on Lahden ammattikorkeakoulun kanssa tehty lisätyn todellisuuden sovellus Lahden kaupunginkirjastolle.

Lisätyn todellisuuden avulla tavallisen tuntuista asioista on mahdollista tehdä kiinnostavampia. Tämän ajatuksen myötä myös kouluryhmien toivotaan innostuvan uuden tekniikan käyttämisestä, ja kirjaston kiinnostavuus lisääntyy.

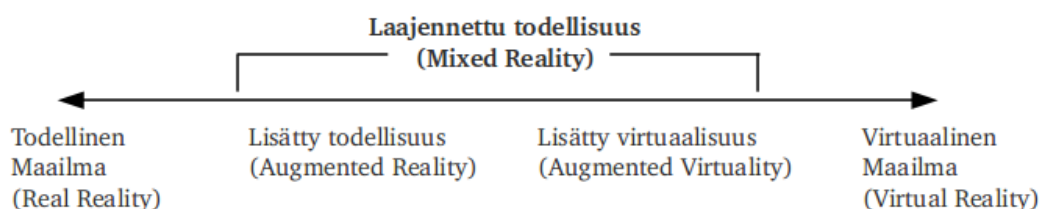
Sovelluksen kohderyhminä olivat erityisesti ala- ja yläkoululaiset, ja kirjasto halusi myös ottaa eri ikäryhmät huomioon sovelluksen kehityksessä. Kirjasto käytti teemojen ja tarinoiden suunnittelussa apuna Lahden Renkomäen koulun kolmannen luokan oppilaita sekä Lahden yläkoulujen seitsemännen luokan oppilaita. Oppilailta saatujen ideoiden ja ehdotuksien avulla saatiin muodostettua molemmille ikäryhmille sopivat tarinat peliä varten.

Sovelluksesta oli kehitettävä sellainen, että sitä olisi mahdollista käyttää erilaisissa ympäristöissä. Esimerkiksi kirjaston koon ei pitäisi estää sovelluksen käyttämistä. Sovelluksen sisällön oli myös oltava sellaista, että kirjaston henkilöstön olisi helppo tuottaa ja hallita sitä.

3 LISÄTTY TODELLISUUS

3.1 Määritelmä

Lisätty todellisuus on yksi laajennetun todellisuuden alakäsitteistä. Siinä käyttäjä näkee todelliseen maailmaan lisättyjä virtuaalisia elementtejä. Lisätty todellisuus eroaa virtuaalisesta todellisuudesta siten, että virtuaalinen todellisuus koostuu pelkästään virtuaalisista elementeistä ja lisätyssä todellisuudessa virtuaalisia elementtejä käytetään vain apukeinona. (Milgram & Kishino 1994, 2-3.)



KUVIO 1. Laajennetun todellisuuden jatkumo (Milgram & Kishino 1994, 3)

Sekä lisätty todellisuus että virtuaalinen todellisuus voidaan luokitella laajennetun todellisuuden jatkumon alle (KUVIO 1). Tämä jatkumo on yksinkertainen tapa kuvata siirtymää todellisen maailman ja virtuaalisen maailman välillä. (Milgram & Kishino 1994, 2-3.)

3.2 Historia

Lisätyn todellisuuden teknologian kehitys voidaan katsoa alkaneeksi jo vuonna 1962, kun Morton Heilig rakensi prototyypin tulevaisuuden elokuvateatteristaan, joka sai nimekseen Sensorama (Payatagool 2008). Käsitteenä lisätty todellisuus on paljon uudempi. Nykyisen tekniikan myötä lisättyä todellisuutta on alettu käyttää yhä useammilla aloilla.

Myös laitteiden koko on pienentynyt huomattavasti vuosien saatossa. Esimerkiksi Sensorama oli ison jääkaapin kokoinen, ja nykyään vastaava teknologia saadaan älypuhelimien kokoiseen laitteeseen. Lisättyä todelli-

suutta hyödyntävä teknologia on nykyään myös kaikkien saatavilla, mikä on vaikuttanut sen suosioon.

3.2.1 Sensorama

Sensorama oli ensimmäinen mekaaninen laite, joka yhdisti useita aisteja realistisen elokuvakokemuksen aikaansaamiseksi (KUVA 1). Sensoraman keksijä Morton Heilig oli kiinnostunut kolmiulotteisuuden käsitteestä kuvan ja äänen suhteen, mutta muut eivät innostuneet hänen ideastaan. Heilig rakensi tämän takia ensimmäisen prototyypin laitteestaan itse vuonna 1962. Kiinnostavuudestaan huolimatta keksintö ei saanut rahoitusta, ja sen kehitys loppui siihen. Sensoraman toimintoihin kuului esimerkiksi tärkevä tuoli, tietyissä kohdissa elokuvaa aktivoituvat tuoksut ja tuulta simuloivat tuulettimet. (Payatagool 2008.)

Introducing . . .

sensorama

The Revolutionary Motion Picture System
that takes you into another world
with

- 3-D
- WIDE VISION
- MOTION
- COLOR
- STEREO-SOUND
- AROMAS
- WIND
- VIBRATIONS



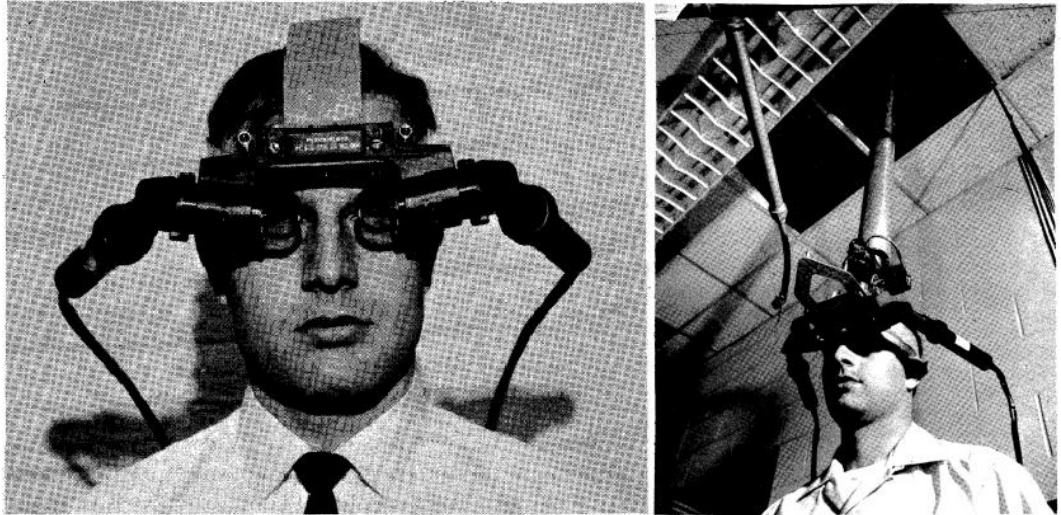
○ PATENTED

SENSORAMA, INC., 855 GALLOWAY ST., PACIFIC PALISADES, CALIF. 90272
TEL. (213) 459-2162

KUVA 1. Sensoraman mainos (Payatagool 2008)

3.2.2 Ensimmäinen päässä pidettävä virtuaalitodellisuuslaite

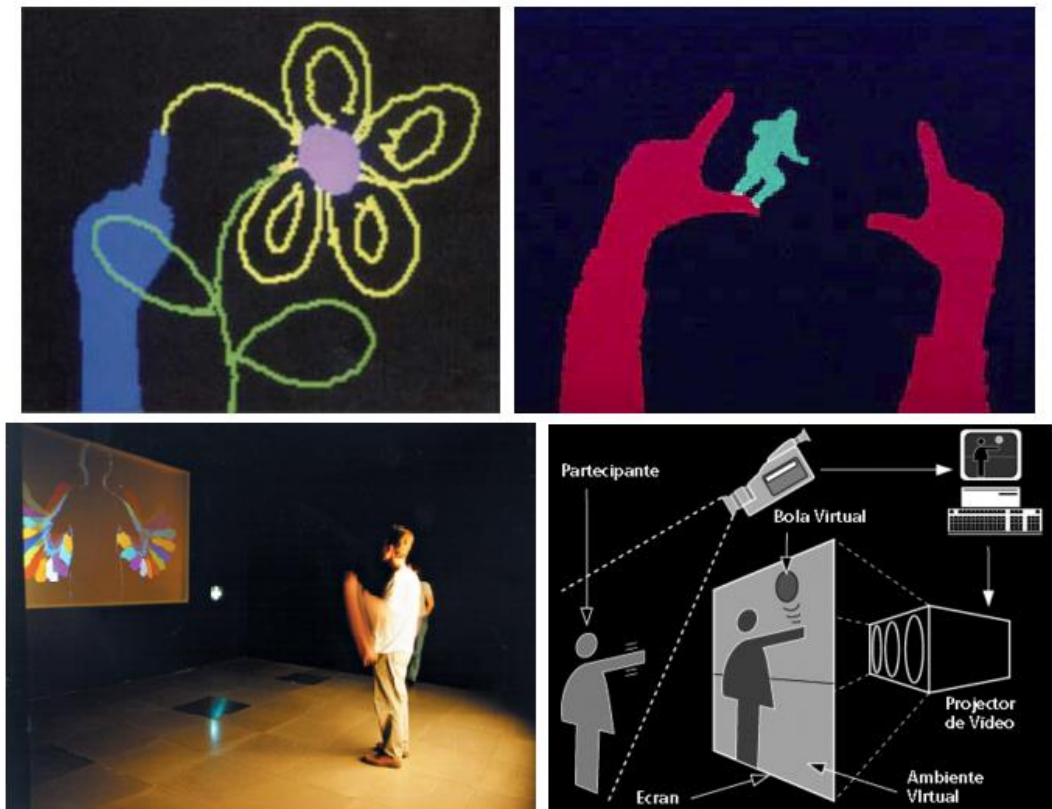
Ivan Sutherland kehitti vuonna 1968 ensimmäisen päässä pidettävän virtuaalitodellisuuslaitteen. Järjestelmä näytti käyttäjälle tietokoneen piirtämää yksinkertaista kuvaa, joka muodostui rautalankamalleista. Käyttäjän näkemän kuvan perspektiiviin vaikutti käyttäjän pään asento, jonka mittaamiseksi laite oli kiinni katossa roikkuvassa mittausturissa (KUVA 2). (Sutherland 1968, 1-8.)



KUVA 2. Ivan Sutherlandin virtuaalitodellisuuslaite (Sutherland 1968)

3.2.3 Videoplace

Videoplace oli ensimmäinen lisätyn todellisuuden sovellus, jossa käyttäjä pystyi liikkeillään olemaan vuorovaikutuksessa virtuaalisen maailman kanssa reaaliajassa. Myron Krueger kehitti Videoplacen vuonna 1975. Se koostui kahdesta tai useammasta tilasta, joissa oli samanlainen järjestelmä. Järjestelmä toimi siten, että kun käyttäjä astui kameran eteen, hän näki ruudulla sekä itsensä että mahdolliset muut käyttäjät (KUVIO 2). Käyttäjä pystyi olemaan vuorovaikutuksessa muiden käyttäjien kuvien kanssa esimerkiksi kääntämällä niitä tai muuttamalla niiden kokoa. Käyttäjä pystyi tekemään saman myös tietokoneen luomille muodoille. (Bryon 2010.)



KUVIO 2. Videoplacen toimintaa (Bryon 2010)

3.2.4 ARToolkit

ARToolkit on avoimen lähdekoodin kirjasto, joka on tarkoitettu lisätyn todellisuuden sovellusten kehittämiseen. Hirokazu Kato aloitti kirjaston kehittämisen vuonna 1999, ja sen kehittäminen jatkuu edelleen (Kato & Billinghurst 1999, 1). Kirjasto toimii nykyään kaikilla yleisimmillä alustoilla mukaan lukien verkkoselaimet. (ARToolKit 2016.)

3.3 Teknologia

Lisätyn todellisuuden toteuttaminen vaatii laitteistolta tiettyjä ominaisuuksia. Ensinnäkin lisätyn todellisuuden aikaan saaminen vaatii erilaisia syötteitä reaali maailmasta, jotta virtuaaliset elementit voidaan asettaa oikeisiin paikkoihin tai jotta käyttäjälle voidaan näyttää oikeaa tietoa.

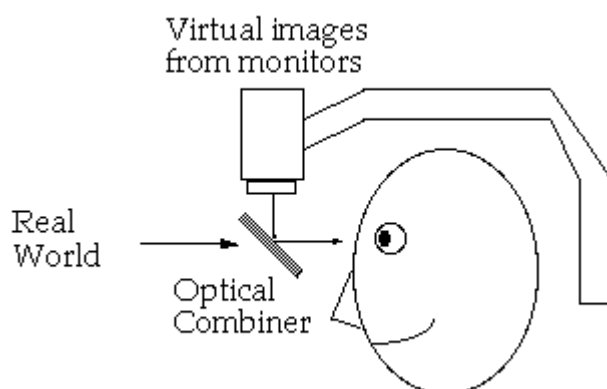
Tarpeellisin syöte on useimmissa sovelluksissa kuvasyöte, jota saadaan esimerkiksi mobiililaitteen kamerasta. Myös erilaisten antureiden tietoja

voidaan käyttää hyödyksi. Esimerkiksi mobiililaitteiden kiihtyvyyssanturi antaa tiedon laitteen asennosta.

Syötteen saatuaan sovelluksen on prosessoitava se ja tuotettava käyttäjälle näytettävä tieto. Useimmat lisätyn todellisuuden sovellukset perustuvat kuvantunnistukseen, joka vaatii laitteistolta kohtuullisen suurta laskentatehoa. Lisätyn todellisuuden sovellukset voidaan jakaa kolmeen näyttötekнологiaan: päässä pidettäviin näyttöihin, kädessä pidettäviin näyttöihin ja projektionäyttöihin.

3.3.1 Päässä pidettävät näytöt

Päässä pidettävät näytöt voidaan jakaa vielä kahteen ryhmään: optisiin näyttöihin ja videonäyttöihin. Optisissa näyttöissä käyttäjä näkee lisätyn todellisuuden puoliläpäisevälle pinnalle heijastettuna (KUVIO 3). Näin käyttäjä näkee virtuaaliset elementit todellisen maailman päällä. Optisella näytöllä toteutettu lisätty todellisuus on paljon videonäyttöä rajoitetumpaa ja soveltuu hyvin esimerkiksi kertomaan lisätietoa katseltavasta kohteesta.

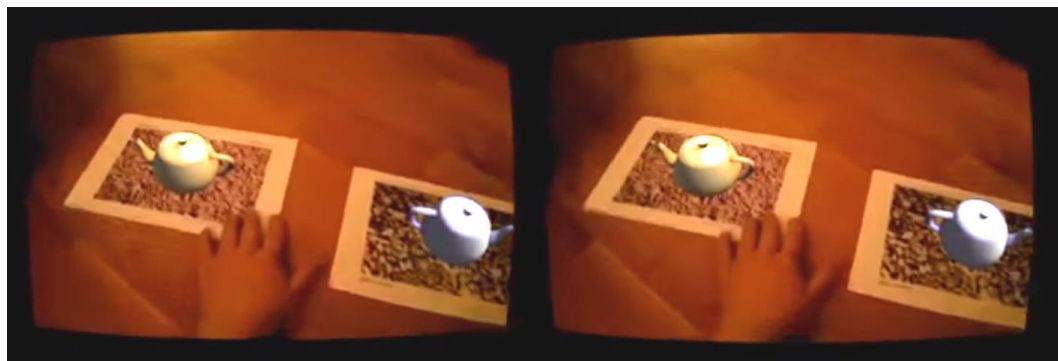


KUVIO 3. Optisen näytön toimintaperiaate (Azuma 2005)

Videonäyttö mahdollistaa todellisemman tuntuksen kokemuksen, koska toisin kuin optisessa näyttössä, videonäytön kuvaan voidaan lisätä kaikki virtuaaliset elementit ennen kuvan näyttämistä katsojalle (KUVIO 4).

Päässä pidettävät videonäytöt sisältävät useimmiten kaksi erillistä näyttöä.

Tällöin molemmille silmille on omat näyttönsä, joiden kuvat eroavat toisistaan hieman kolmiulotteisuuden aikaansaamiseksi. (Thor 2015.)



KUVIO 4. Jatkettua todellisuutta virtuaalilasien läpi katsottuna (Thor 2015)

3.3.2 Kädessä pidettävät näytöt

Kädessä pidettäviä laitteita ovat esimerkiksi älypuhelimet ja tabletit. Näillä laitteilla lisätty todellisuus toteutetaan siten, että laitteen kamerasta tulevaan kuvaan lisätään virtuaaliset elementit, ja se esitetään käyttäjälle laitteen näytöltä. Mobiililaitteiden rajoittavin tekijä on niiden lyhyt akunkesto ja pieni laskentateho, mutta laitteiston kehittyessä tämä ongelma vähenee.

3.3.3 Projektionäytöt

Projektioteknologia perustuu projektoriin, joka heijastaa jollekin pinnalle kuvaa tai videota. Tällä tavalla voidaan näyttää pinnalla lisätietoa, joka auttaa esimerkiksi suunnittelutyössä tai tuotteen toiminnan mallinnuksessa (KUVIO 5). Projektionäyttö eroaa muista lisätyn todellisuuden näyttötekniologioista sillä, että käyttäjä ei tarvitse erillistä laitetta sen katseluun. Tämä teknologia vaatii kuitenkin näyttöpinnalta tiettyjä ominaisuuksia. Paras tulos saadaan, jos pinta on vaalea, heijastamaton ja tasainen. (Raskar, Welch & Fuchs 1998, 1-2, 5.)



KUVIO 5. Auton sisärakenne heijastettuna sen kylkeen (Volkswagen AG 2016)

3.4 Käyttökohteet

Lisätyn todellisuuden käyttö on saanut suosiota melko hitaasti, vaikka sen vaatimat teknologiat ovat olleet olemassa jo pitkään. Vasta viime aikoina elektronisten komponenttien koko on pienentynyt niin, että kaikki tarpeellinen saadaan mahtumaan esimerkiksi puhelimen kuorien sisään. Lisätyn todellisuuden nykykäytöstä on olemassa erilaisia esimerkkejä monilta aloilta.

3.4.1 Mainonta

Mainonnassa lisättyä todellisuutta voidaan hyödyntää vaikkapa tuote-esittelyissä. Esimerkiksi, kun käyttäjä katsoo tuotteen mainosta älylaitteen kanssa, hän voi nähdä tuotteen esittelyvideon tai itse tuotteen kolmiulotteisena. Leluyhtiö Legokin on käyttänyt lisättyä todellisuutta tuotteidensa mainonnassa (KUVIO 6).



KUVIO 6. Virtuaalinen Legotalo paketin päällä (Horsey 2011)

3.4.2 Arkkitehtuuri

Lisätyn todellisuuden avulla voidaan esimerkiksi katsoa oikeassa maailmassa sijaitsevaa paikkaa älylaitteen läpi ja nähdä suunniteltu rakennus oikeassa paikassa. Lisättyä todellisuutta voidaan myös käyttää huoneiden sisustuksen suunnittelun apuna. Esimerkiksi tyhjäan huoneeseen voidaan lisätä virtuaalisia huonekaluja ja katsoa etukäteen, miltä huone näyttäisi niin sisustettuna. (Busta 2015.)

3.4.3 Navigaatio

Lisätyllä todellisuudella voidaan tehdä navigoinnista tehokkaampaa. Esimerkiksi auton tuulilasissa voidaan näyttää ajo-ohjeita ja muita tietoja liikenteestä, jolloin kuljettajan ei tarvitse kääntää katsettaan pois tiestä. (Burns 2010.)

Toinen alue, jossa lisättyä todellisuutta on käytetty navigoinnissa, on ilmailu- ja avaruusteknologia. Esimerkiksi sotilaskoneissa ja -helikoptereissa käytetään usein heijastusnäyttöä ja kypärään asennettua näyttöä, jotka

antavat lisätietoa lennon tilasta ja ympäristöstä. (Foyle, Andre & Hooey 2005, 3-4.)

3.4.4 Opetus

Myös opetuskäytössä voidaan hyödyntää lisättyä todellisuutta. Esimerkiksi oppikirjoihin tai monisteisiin voidaan lisätä kuvia, jotka aktivoivat lisätyn todellisuuden elementtejä mobiililaitteella katsottuna. Etenkin pienten oppilaiden voi olla vaikea ymmärtää monia asioita, joten esimerkiksi kolmiulotteinen malli tai animaatio opetettavasta asiasta saattaa auttaa oppimista suuresti. (Maier, Klinker & Tonnis 2009, 1-2, 6.)

3.4.5 Lääketiede

Lääketieteen alalla lisätty todellisuus mahdollistaa esimerkiksi kirurgeille ja lääkäreille lisäinformaation näkemisen suoraan potilaaseen heijastettuna. Toinen mahdollisuus on nähdä suoraan potilaan sisälle näytön ja esimerkiksi ultraäänidatan avulla. (Ackerman 2000.)

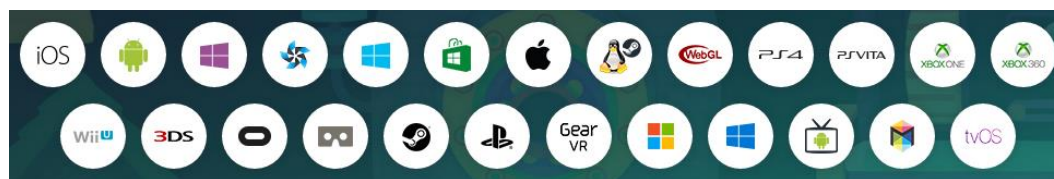
Yksi lääketieteen alan keksintö on laite, joka ottaa lämpökuvan potilaan pintaverisuonista, käsittelee kuvan niin, että suonet korostuvat siitä, ja heijastaa kuvan potilaan päälle, jolloin suonet näkyvät lääkärille selvemmin (Miyake, Zeman, Duarte, Kikuchi, Ramacciotti, Lovhoiden & Vrancken 2006).

3.4.6 Peliteollisuus

Lisätyn todellisuuden avulla pelaajat voivat kokea pelin virtuaaliset ominaisuudet oikeassa maailmassa. Esimerkiksi LyteShot on lisättyä todellisuutta hyödyntävä peli. Se vaatii älypuhelimien, joka on yhteydessä internetiin, sekä itse pelissä käytettäviä langattomia ohjaimia ja sensoreita, jotka ovat yhteydessä pelaajan puhelimeen. Pelaaja pysyy tilanteen tasalla pelipalvelimelta tulevien viestien avulla, ja pelipalvelin saa uutta tietoa pelaajien sensoreilta. (Lyteshot 2015.)

4 UNITY

Unity on pelimoottori, joka pyrkii mahdollistamaan alustasta riippumattoman kehityksen. Unity sisältää oman editorin, jonka avulla kehittäminen on nopeaa ja helposti opittavaa. Unity tukee useimpia tällä hetkellä käytettyjä alustoja, suosituimpina iOS, Android ja Windows (KUVIO 7), sekä mahdollistaa saman tuotteen julkaisun usealle eri alustalle. Tällä hetkellä Unityn editori on saatavilla Windowsille ja Macille. (Unity Technologies 2016c; Unity Technologies 2016f.)



KUVIO 7. Unityn tukemat alustat (Unity Technologies 2016f)

4.1 Lisenssit

Unity on päätenyt lisensseissään ratkaisuun, jossa se tarjoaa pääasiassa kahta lisenssiä (KUVIO 8). Personal Edition -lisenssi vaatii käyttäjältä pelkästään rekisteröitymisen, ja tämän jälkeen kehitystyössä voi käyttää kaikkia pelimoottorin ominaisuuksia. Professional Edition -lisenssi tarjoaa tämän lisäksi tiimityöskentelyyn tarkoitettuja työkaluja sekä muita etuja kehitykseen, mutta on myös maksullinen. (Unity Technologies 2016c.)

Tavallisten lisenssien lisäksi Unityllä on mahdollista kehittää pelejä esimerkiksi uhkapeliteollisuuteen, joka vaatii erillisen lisenssin. Unityn kehittäjiin voi ottaa yhteyttä myös muissa erikoistapauksissa, ja heiltä on mahdollista kysyä itselle sopivista ratkaisuista. (Unity Technologies 2016i.)

Unity haluaa myös auttaa oppimisessa ja tarjoaa tätä ajatellen kouluille ja oppilaitoksille erikoistarjouksia lisensseihin ja sisältöön. Unityn verkkosivuilta löytyy myös esimerkkiprojekteja ja videoharjoituksia, joiden tarkoituksena on opettaa pelien tekemistä esimerkkien avulla. (Unity Technologies 2016c.)

UNITY 5 What's included		PERSONAL EDITION	PROFESSIONAL EDITION
Engine with all features	?	✓	✓
Royalty-free	?	✓	✓
All platforms (limitations apply)	?	✓	✓
Customizable Splash Screen		✗	✓
Unity Cloud Build Pro - 12 Months	?	✗	✓
Unity Analytics Pro	?	✗	✓
Team License	?	✗	✓
Prioritized bug handling	?	✗	✓
Game Performance Reporting	?	✗	✓
Beta access	?	✗	✓
Unlimited Revenue and Funding	?	✗	✓
Future platforms included	?	✗	✓
Professional editor skin		✗	✓
Asset Store Level 11	?	✗	✓
Professional Community Features	?	✗	✓
Source code access	?	✗	\$
Premium Support	?	\$	\$

KUVIO 8. Unityn lisenssien sisältö (Unity Technologies 2016c)

4.2 WWW ja WWWForm

Unity tarjoaa helpon työkalun verkkosivuille pääsyyn. WWW-luokan avulla voidaan tehdä GET- ja POST-hakuja palvelimelle. Oletuksena WWW-luokka käyttää GET-metodia, jolloin hakuparametrit syötetään suoraan verkko-osoitteen perään. WWW-luokan avulla voidaan myös ladata tiedostoja laitteen omasta muistista tiedoston URL-osoitteen perusteella. (Unity Technologies 2016j.)

```

IEnumerator HaeSessiot()
{
    // osoite
    string url = "http://niisku.lamk.fi/~kirjastoar/rajapinta.php";

    // tehdään uusi lomake lähetettävälle tiedolle
    WWWForm form = new WWWForm();

    // lisätään lähetettävä tieto
    form.AddField("toiminto","haeSessiot");

    // tehdään haku palvelimelle
    WWW www = new WWW(url, form.data);

    // funktio jää odottamaan palvelimen vastausta ja jatkaa suoritusta sen jälkeen
    yield return www;
}

```

KUVIO 9. POST-haku Unityllä

POST-hakuja voi tehdä, mikäli WWW-luokan konstruktorille annetaan lähetettävä data erillisenä parametrina. POST-kutsussa lähetettävät tiedot on muodostettava WWWForm-olion avulla. WWWForm-olioon voidaan säilöä avain–arvo-pareja AddField-metodilla. Arvona voidaan antaa merkijono tai binääridataa, joka mahdollistaa tiedostojen lähettämisen. (Unity Technologies 2016k.)

WWW-haku on asynkroninen operaatio, eli vastauksen saapumiseen kuluva aikaa ei tiedetä. Unityssä asynkronisiin toimintoihin voidaan käyttää IEnumeratorin palauttavaa funktiota, jonka suorittaminen voidaan pysäyttää tarvittavaksi ajaksi. Esimerkiksi kuvan funktio lopettaa suorituksensa palvelimelle tehtävän haun ajaksi ja jatkaa suoritusta, kun palvelimen vastaus on saapunut kokonaan (KUVIO 9). (Unity Technologies 2016j.)

4.3 IEnumerator ja yield

IEnumerator on C#-kielen rajapinta, jonka avulla luokasta voidaan toteuttaa yksinkertainen lista. Kun IEnumerator-rajapinnan toteuttavaa listaa iteroidaan läpi, listan elementti voidaan palauttaa ”yield return <value>;”-komennolla. Listan iterointi voidaan lopettaa poistumalla funktiosta komennolla ”yield break;”. (Microsoft 2016a.)

Varattua sanaa `yield` voidaan käyttää funktiosta poistumiseen siten, että funktio muistaa seuraavalla kutsukerralla, mihin se jäi viimeksi. Tämä asettaa funktiolle tiettyjä vaatimuksia. Esimerkiksi funktion palautustyyppin on oltava `IEnumerator` tai `IEnumerable`. (Microsoft 2016b.)

4.4 JsonUtility

JSON on tekstimuotoinen esitys olion tai taulukon rakenteesta. Tiedon välittäminen JSON-muodossa on tehokkaampaa kuin esimerkiksi XML-muodossa, johon sisältyvät elementtien tunnisteet lisäävät lähetettävän tiedoston kokoa (KUVIO 10).

```
<!-- XML -->
<Pelaaja>
  <id>0</id>
  <nimi>Pelaaja1</nimi>
</Pelaaja>

// JSON
{ "Pelaaja": { id:0, nimi:"Pelaaja1" } }
```

KUVIO 10. Sama tieto esitettyinä XML- ja JSON-muodoissa

Koska JSON on käytännössä pelkkää tekstiä, se pitää kääntää myös ohjelman ymmärtämään muotoon. Tämä onnistuu Unityn sisältämällä `JsonUtility`-luokalla. Luokalla on funktiot sekä JSON:n enkoodaamiseen että de-koodaamiseen. (Unity Technologies 2016e.)

Luokka on toiminnoiltaan hyvin karsittu, mutta se on myös pienikokoisempi kuin useat muut JSON-kirjastot. Tämän vuoksi dataa ei ole mahdollista käsitellä puuna, jonka avain–arvo-pareja voisi lisätä ja poistaa vapaasti. Sen sijaan tulevalle ja lähtevälle datalle on kirjoitettava omat kiinteät luokat sen perusteella, millaista dataa niiden avulla välitetään (KUVIO 11). Nämä luokat voivat sisältää muita luokkia tai struct-rakenteita sekä taulukoita. (Unity Technologies 2016e.)

```

//JSON
{
  "id": 0,
  "pelaajat": [
    {
      "id": 0,
      "nimi": "Pelaaja 1"
    },
    {
      "id": 1,
      "nimi": "Pelaaja 2"
    },
    {
      "id": 2,
      "nimi": "Pelaaja 3"
    }
  ]
}
}

public class Pelaaja
{
    public int id;
    public string nimi;
}

public class Peli
{
    public int id;
    public Pelaaja[] pelaajat;
}

public class JsonTest
{
    public void LuePelaajat(string jsonString)
    {
        // muunnetaan tekstimuotoinen json C#-olioksi
        Peli peli = JsonUtility.FromJson<Peli>(jsonString);

        // tulostetaan kaikki pelaajat
        foreach(Pelaaja pelaaja in peli.pelaajat)
        {
            Console.WriteLine("Pelaaja, id:{0} nimi:{1}", pelaaja.id, pelaaja.nimi);
        }
    }
}
}

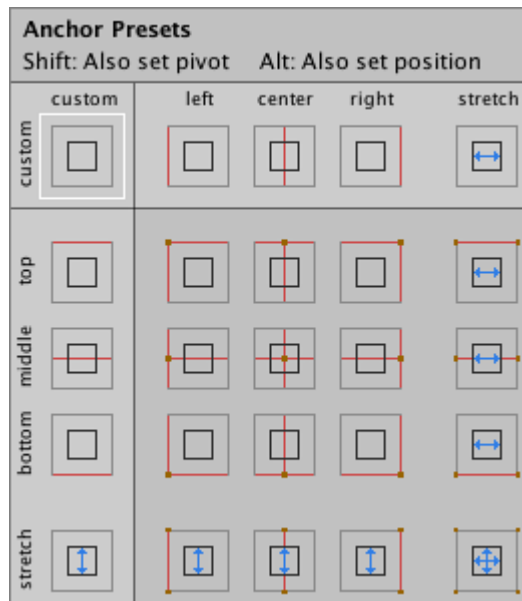
```

KUVIO 11. Json-tekstin muuntaminen C#-olioksi

4.5 UI

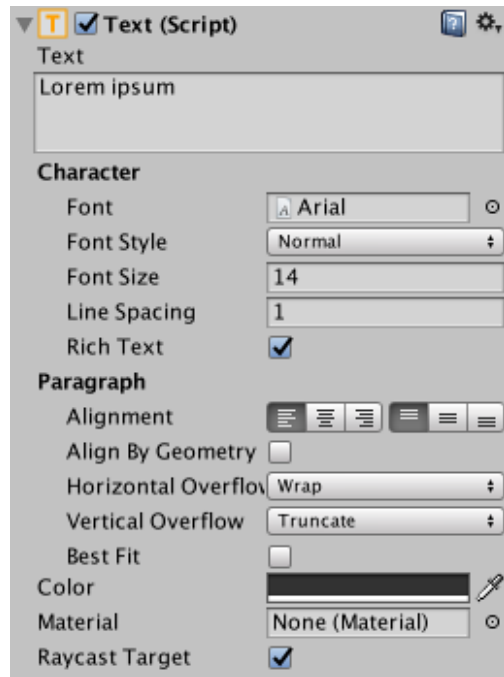
Käyttöliittymän toteuttaminen Unityllä uudistui versiossa 4.6 tulleilla uusilla käyttöliittymäkomponenteilla. Uusien komponenttien ansiosta käyttöliittymän suunnittelu onnistuu nopeasti Unityn editorilla. Komponenteille on helppo määritellä ankkuripisteet, joiden avulla niitä voi liikuttaa ja venyttää esimerkiksi näytön koon muuttuessa (KUVIO 12). Tämän ansiosta Unityllä

on helppo tehdä käyttöliittymä, joka toimii useilla resoluutioilla ja kuvasuh-
teilla. (Unity Technologies 2016a.)



KUVIO 12. Komponentin ankkurointiasetukset (Unity Technologies 2016a)

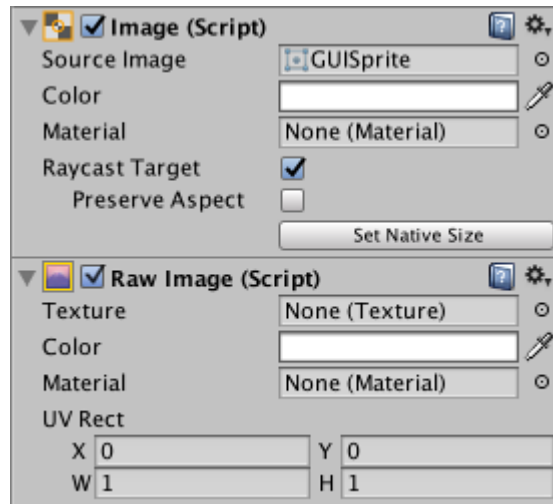
Uusi käyttöliittymä perustuu Canvas-komponenttiin, joka voidaan piirtää
näytölle kameran päälle, tai se voidaan asettaa pelimaailmaan tavallisena
peliobjektina. Canvas-komponentin sisään voi lisätä tavallisimpia käyttöliit-
tymäelementtejä, kuten tekstiä, kuvia ja painikkeita. (Unity Technologies
2016b.)



KUVIO 13. Tekstikomponentti editorissa (Unity Technologies 2016h)

Tekstikomponentti sisältää kaikki tavallisimmat tekstin ominaisuudet, joita on helppo muokata. Näihin lukeutuvat esimerkiksi tekstin koko, fontti ja väri, kuten kuvassa näkyy (KUVIO 13). Tekstikomponentti sisältää myös ominaisuuden ”Best Fit”, jolla on mahdollista vaihtaa tekstin kokoa automaattisesti niin, että se mahtuu täysin sille annetulle alueelle. Tekstille voidaan antaa tällöin maksimi- ja minimifonttikoot, joiden perusteella tekstin koko määrittyy. (Unity Technologies 2016h.)

Tekstikomponentin lisänä voi käyttää komponenttia ”Outline” eli ääriiviiva, jolla voidaan lisätä tekstille värilliset reunat. Tällä lisäyksellä voidaan tehdä tekstistä paremmin erottuvaa asettamalla esimerkiksi valkoiselle tekstille mustat reunat. Näin teksti pysyy helposti luettavana, vaikka taustan väri vaihtuisi. (Unity Technologies 2016h.)



KUVIO 14. Kuvakomponentit editorissa (Unity Technologies 2016)

Uudessa käyttöliitymässä on kaksi erilaista kuvakomponenttia (KUVIO 14). Tavallinen kuvakomponentti (Image) on kuvista monipuolisempi. Se antaa enemmän mahdollisuuksia kuvan animointiin ja efektien tekemiseen, mutta vaatii kuvaksi Sprite-tyyppisen kuvan. Yksi mahdollisuus on käyttää kuvan täyttötoimintoa. Se mahdollistaa kuvan osittaisen näyttämisen, jonka avulla voi esimerkiksi tuoda kuvan esiin liukuvasti. Komponentti tukee myös interaktiivisia toimintoja, eli käyttäjä voi esimerkiksi klikata kuvaa ja käynnistää jonkin tapahtuman. (Unity Technologies 2016d.)

Raakakuvakomponentti (Raw Image) on tavallista kuvaa yksinkertaisempi, mutta se ei ole interaktiivinen. Tätä komponenttia voi siis käyttää pelkästään koristeena, mutta se soveltuukin hyvin esimerkiksi taustakuvaksi tai ikoniksi. Raakakuva sisältää vähemmän toimintoja, mutta kuvaksi voi asettaa minkä tahansa tekstuurin (Texture). (Unity Technologies 2016g.)

5 VUFORIA

Vuforia on lisätyn todellisuuden kirjasto, joka sisältää useita kuvantunnistukseen perustuvia tekniikoita lisätyn todellisuuden aikaansaamiseksi. Vuforia sisältää myös oman palvelun, jonka kautta kehittäjä voi hallita näitä tunnistettavia kohteita, joita kehitettävä sovellus käyttää toiminnassaan. (PTC 2016c.)

Vuforiaa voidaan käyttää lisätyn todellisuuden sovellusten kehittämiseen mobiilialustoille. Näihin alustoihin kuuluvat sekä älypuhelimet ja tabletit että useat virtuaalilasit. Vuforialla voi tehdä sovelluksia Android-, iOS- ja Windows 10 -laitteille. (PTC 2016b.)

5.1 Lisenssit

Vuforia tarjoaa eritasoisia lisenssejä erilaisia käyttötarpeita varten (KUVIO 15). Starter-lisenssin käyttämiseen vaaditaan rekisteröityminen, jonka jälkeen sovellusta voi kehittää maksutta. Starter-lisenssillä tehty sovellus näyttää kerran päivässä Vuforian vesileiman näytön kulmassa, kun sovellusta käytetään. Tällä lisenssillä voidaan myös tehdä tuhat pilvitunnistusta kuukaudessa. (PTC 2016i.)

Classic-lisenssillä tehdystä sovelluksesta voi piilottaa vesileiman, mutta lisenssi ei tue pilvitunnistuspalvelua. Tämä lisenssi toimii kertamaksulla, jonka jälkeen ei ole kuukausittaisia maksuja. (PTC 2016i.)

Cloud-lisenssi sopii sellaisille sovelluksille, jotka tarvitsevat useita tunnistuskohteita, tai mikäli kohteet päivittyvät usein. Tämä lisenssi sisältää omat alatasonsa, joista kukin taso sisältää eri määrän pilvitunnistuksia kuukaudessa. (PTC 2016i.)

Jokainen lisenssityyppi sisältää rajattoman määrän sovelluksen sisäisesti tekemiä kohdetunnistuksia. Juuri mainituilla lisensseillä voidaan tehdä sovelluksia Android-, iOS- ja Windows 10 -alustoille, mutta virtuaalilasit vaativat erillisen lisenssin. (PTC 2016i.)

	Starter	Classic	Cloud		
			Bronze	Silver	Gold
Features					
Image, Object, Cylinder, User-defined Target, Frame Marker, Text & Smart Terrain	✓	✓	✓	✓	✓
Cloud Recognition Service	✓	--	✓	✓	✓
Remove Watermark	--	✓	✓	✓	✓
Pricing & Usage					
Device Recos	unlimited				
Cloud Recos	1,000 per month	--	10,000 per month	50,000 per month	150,000* per month
Cloud Targets	1,000	--	100,000*		
Price per App	\$0 Sign Up	\$499 one time fee Sign Up	\$99 per month Sign Up	\$399 per month Sign Up	\$999 per month Sign Up

KUVIO 15. Yleiskatsaus Vuforiaan lisensseistä (PTC 2016i)

5.2 Kohteiden hallinta

Vuforiolla on kehittäjiä varten tarkoitettut verkkosivut, joiden kautta voi hallita lisenssien lisäksi tunnistuskohteita (Target). Tunnistuskohteille tehdään omat tietokannat (Database), joita sovellus voi hyödyntää (KUVIO 16). Pelkästään laitteen sisäisiä tunnistuksia hyödyntävä sovellus voi käyttää samanaikaisesti montaa eri tietokantaa, kun taas pilvitunnistuksia hyödyntävä sovellus voi käyttää vain yhtä tietokantaa. (PTC 2016j.)

Jokaiselle tietokannalle on määriteltävä sen tyyppi joko laitetietokanta tai pilvitietokanta. Laitetietokannan käyttäminen onnistuu sellaisenaan. Pilvitietokannan käyttäminen on muuten samanlaista kuin laitetietokannan, mutta pilvitietokannan käyttöön vaaditaan kaksi avainta sovelluksen puolelle ja kaksi avainta hallintapalvelimen puolelle. (PTC 2016j.)

License Manager Target Manager

Target Manager

Use the Target Manager to create and manage databases and targets.

Add Database

Database	Type	Targets	Date Modified
KirjastoAR	Device	3	Feb 16, 2016 10:26

License Manager Target Manager




Target Manager > KirjastoAR

KirjastoAR [Edit Name](#)

Type: Device

Targets (3)

Add Target Download Dataset (All)

<input type="checkbox"/>	Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/>	 scorpion	Single Image	★★★★☆	Active	Feb 16, 2016 10:14
<input type="checkbox"/>	 Untitled	Single Image	★★★★☆	Active	Feb 04, 2016 10:35
<input type="checkbox"/>	 Marker-00	Single Image	★★★★★	Active	Feb 02, 2016 12:18

KUVIO 16. Vuforia tunnistuskohteiden hallinta (PTC 2016j)

Vuforia kuvakohteiden hallinta toimii siten, että käyttäjä lataa palveluun kuvia, joita hän haluaa käyttää sovelluksessaan. Tämän jälkeen palvelu antaa kuvalle arvosanan, joka kertoo, kuinka helposti kuva on tunnistettavissa. Tunnistusta helpottavat kuvassa olevat kulmikkaat alueet ja suuret värien kontrastierot (PTC 2016g). Kuvista on myös mahdollista nähdä harmaasävyinen versio, jossa näkyy rastein merkittynä, mitä alueita Vuforia kuvasta tunnistaa. Haluamansa tunnistuskohteet voi ladata palvelusta pakettina (Unity Package), jonka voi tuoda suoraan Unityn editoriin. (PTC 2016j.)

5.3 Web-rajapinta

Vuforia mahdollistaa oman julkaisujärjestelmän käyttämisen pilvitunnistuksen hallintaan tarjoamalla palveluihinsa helppokäyttöisen rajapinnan. Tä-

män rajapinnan kautta on mahdollista lisätä, muokata ja poistaa tunnistuskohteita. (PTC 2016n.)

Web-rajapinnan käyttämiseen tarvitaan kaksi erilaista avainta. Palvelimen avainta käytetään kohdetietokannan ylläpitoon, ja avaimella on tätä varten luku- ja kirjoitusoikeudet. Asiakkaan avainta käytetään, kun sovellus tekee kyselyitä kohdetietokantaan. Tällä avaimella on pelkkä lukuoikeus, ja avain syötetään sovellukseen kehityksen yhteydessä. (PTC 2016n.)

```
StringToSign =  
  HTTP-Verb + "\n" +  
  Content-MD5 + "\n" +  
  Content-Type + "\n" +  
  Date + "\n" +  
  Request-Path;  
  
Signature = Base64(HMAC-SHA1(server_secret_key, StringToSign ) ) ;  
  
Authorization: VWS {provision_access_key}:{Signature}
```

KUVIO 17. Rajapintakutsussa vaadittu Authorization-kenttä (PTC 2016n)

Kaikissa rajapintakutsuissa käytetään salattua yhteyttä. Tämän lisäksi rajapintaan tehtävät haut muotoillaan Vuforia standardin mukaisiksi ja salaataan (KUVIO 17). Hakuihin sisältyy tieto käytettävästä protokollasta, haun sisältö, haun tyyppi, aikaleima ja haun osoite. (PTC 2016n.)

5.4 Erilaiset tunnistuskohteet

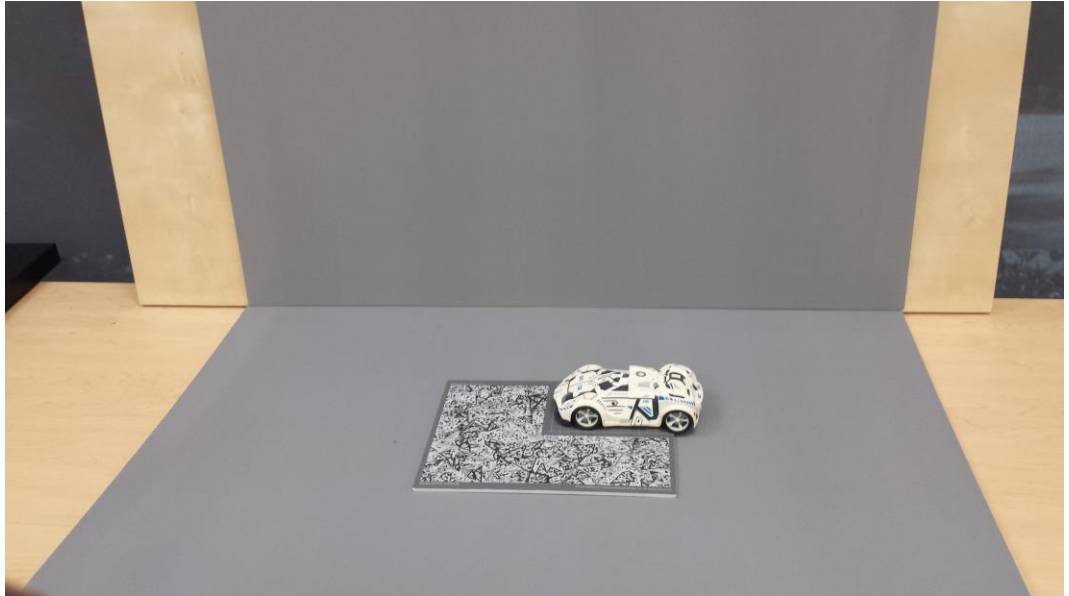
Useissa lisätyn todellisuuden sovelluksissa lisätyn todellisuuden esittäminen vaatii jonkin kuvan tai merkin tunnistamisen. Tunnistuksen jälkeen sovellus näyttää esimerkiksi kohteeseen liitetyn 3d-mallin tai videon. Kuviossa (KUVIO 18) näkyy soraa esittävän kuvan päällä virtuaalinen kahvipannu. (PTC 2016c.)



KUVIO 18. Esimerkki 3d-mallista kuvakohteen päällä (PTC 2016e)

Kuvakohteen tunnistus on yksi lisätyn todellisuuden perustoiminnoista. Vuforia kuvantunnistus ei vaadi kuvaan QR-koodia tai muuta vastaavaa mustavalkoista aluetta, jonka ohjelma tunnistaisi, vaan kuvan tunnistaminen perustuu sen luonnollisiin piirteisiin. Kun sovellus tunnistaa kuvan, se jatkaa kuvan tunnistusta, kunnes kuva katoaa kokonaan näkyvistä. Kuvan tunnistessaan sovellus voi myös käynnistää tapahtuman ja lopettaa kuvantunnistuksen tapahtuman ajaksi. (PTC 2016e.)

Vuforia osaa myös tunnistaa erilaisia kolmiulotteisia muotoja. Uusimpana mahdollisuutena on minkä tahansa kolmiulotteisen muodon skannaaminen. Tämä on monimutkaisempi toiminto ja vaatii erillisen Vuforia kehitystyökaluista löytyvän sovelluksen sekä hyvin valmistellun kuvausympäristön (KUVIO 19). Tällaista kohdetyppiä voi käyttää esimerkiksi lelujen tai pienten tuotteiden yhteydessä. (PTC 2016h.)



KUVIO 19. Tasavärinen kuvaustausta 3d-mallin kuvausta varten (PTC 2016n)

Tavallisempia muotoja edustavat laatikko ja sylinteri. Nämä muodot perustuvat muodon tunnistuksen ohella myös vahvasti edellä mainittuun kuvantunnistukseen. Muodoista monimutkaisempi on laatikko, joka muodostuu yhteensä kuudesta kuvasta. Nämä kuvat muodostavat laatikon tahkot, ja jokaiselle kuvalle on määriteltävä niiden oikea paikka laatikossa. Laatikkokohdetta voi käyttää esimerkiksi tuote-esittelyissä. (PTC 2016f.)



KUVIO 20. Sylinterikohde ja virtuaalinen jalkapallo (PTC 2016a)

Sylinterikohde on kolmiulotteisista kohteista yksinkertaisin. Sellaisen tekeminen Vuforiaan kohdehallinnassa vaatii pelkästään kuvan ja sylinterin säteen määrittelyn. Sylinterikohdetta voi käyttää esimerkiksi tölkkien, mukien ja pullojen yhteydessä (KUVIO 20). (PTC 2016a.)

5.5 Tekstintunnistus

Vuforiolla on mahdollista tunnistaa tekstiä valmiista sanojen listasta. Kehitystyökalujen mukana saatava lista sisältää satatuhatta yleistä englanninkielistä sanaa, joita on mahdollista käyttää sovelluksissa. Kehittäjä voi myös luoda omia sanalistoja ja tehdä suodattimia, joilla voidaan estää tiettyjen sanojen tunnistaminen. (PTC 2016k.)

Tekstintunnistus on hyvä ratkaisu sellaisiin tapauksiin, joissa on tunnistettava yksittäisiä sanoja tai useita sanoja peräkkäin. Tekstintunnistusta voidaan käyttää sovelluksessa sellaisenaan tai sitä voidaan käyttää muiden

lisätyn todellisuuden ominaisuuksien kanssa yhdessä. Myös tekstintunnistuksen yhteydessä on mahdollista esimerkiksi näyttää 3d-malli, kun sovellus näkee oikean sanan. Koodissa sovelluksen seuraamien sanojen tarkastelu näyttää kuvion mukaiselta (KUVIO 21). (PTC 2016k.)

```
StateManager stateManager = TrackerManager.Instance.GetStateManager();  
// get all currently tracked words from the WordManager:  
IEnumerable<WordResult> wordResults =  
    stateManager.GetWordManager().GetActiveWordResults();  
// log a the string value of each word:  
foreach (WordResult wordResult in wordResults)  
{  
    Word word = wordResult.Word;  
    Debug.Log("Currently tracked word: " + word.StringValue);  
}
```

KUVIO 2. Seurattavana olevien sanojen tulostaminen (PTC 2016d)

6 SOVELLUKSEN TOTEUTUS

Sovellus toteutettiin Lahden ammattikorkeakoulun projektikurssilla. Projektitiimiin kuului yhdeksän jäsentä: kolme ohjelmoijaa ja kuusi graafikkoa. Sovelluksen toteutuksessa pidettiin yhteyttä Lahden kaupunginkirjaston kontaktihenkilöihin.

6.1 Kehitysympäristön alustaminen

Vuforia on lisätyn todellisuuden kirjasto, ja sen ominaisuuksien käyttämiseen vaaditaan jokin kehitysympäristö. Vuforian lisääminen Unityyn oli tehty helpoksi. Vuforian verkkosivuilta pystyi lataamaan Unityn kanssa käytettävän kirjaston sekä omat kirjastot Android- ja iOS-käännöksiä varten (KUVIO 22).

Vuforia 5 SDK

Use the Vuforia SDK to build Android and iOS applications for mobile devices and digital eyewear. Apps can be built with Android Studio (Java/C++), XCode (C++), and Unity, the cross-platform game engine.



[Download for Android](#)

vuforia-sdk-android-5-0-10.zip (6.38 MB)



[Download for iOS](#)

vuforia-sdk-ios-5-0-10.zip (16.69 MB)



[Download for Unity](#)

vuforia-unity-5-0-10.unitypackage (33.17 MB)

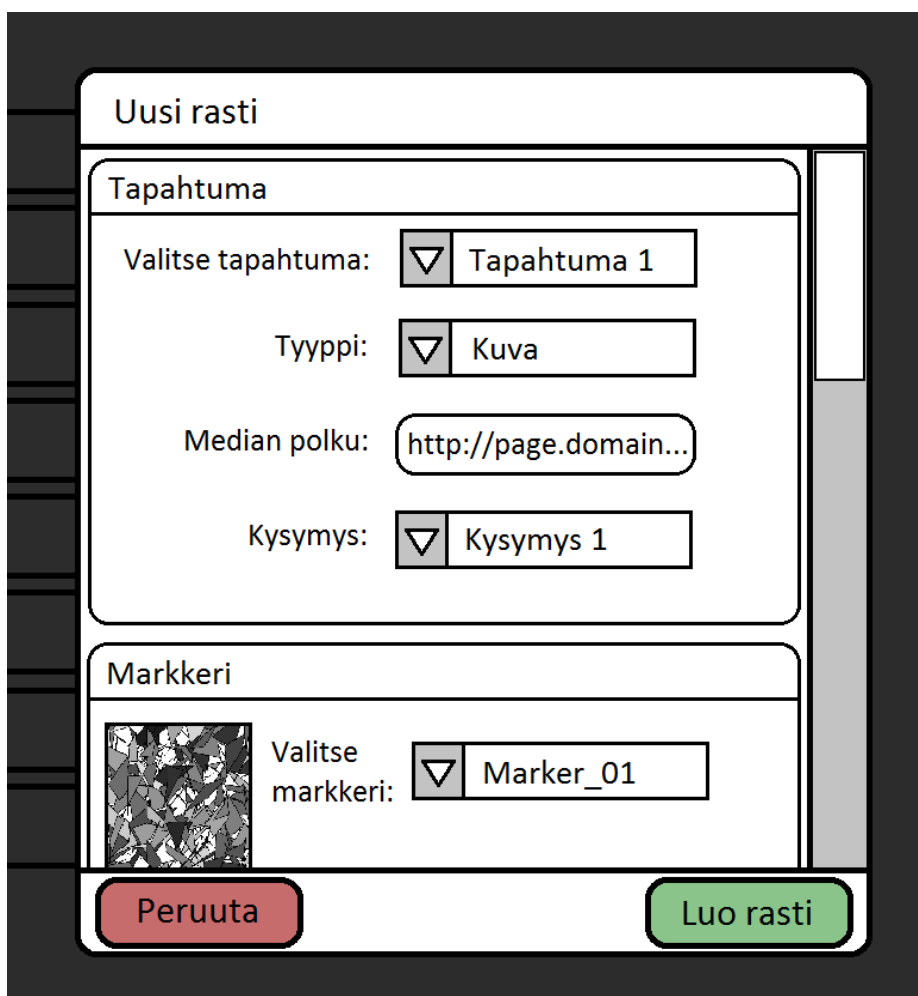
KUVIO 22. Vuforian kehitystyökalujen latausvaihtoehdot (PTC 2016I)

Unityä varten tarkoitettu kirjasto oli pakattu Unityn käyttämään pakettimuotoon (Unity Package), joka oli helppo tuoda Unityn editoriin valikon kautta. Paketti sisälsi kaikki koodit ja useita valmiita peliobjekteja, jotka vaadittiin lisätyn todellisuuden aikaansaamiseksi.

Vuforian tarjoamat valmiit osat eivät yksin riittäneet vaadittujen toimintojen toteuttamiseen. Tämä vaati uusien skriptien kirjoittamista Vuforian ohjelmointirajapintaa hyödyntäen. Vuforian dokumentaatiosta löytyi helposti ohjeita vaadittujen muutosten tekemiseen, joten Vuforian toiminnallisuuksien hyödyntäminen ohjelmoinnissa ei osoittautunut ongelmaksi.

6.2 Palvelin ja hallintapaneeli

Yksi sovelluksen haasteista oli toteuttaa palvelin, jonka hallintapaneelin kautta voidaan luoda ja muokata sovelluksen esittämää sisältöä. Hallintapaneelista tehtiin verkkoselaimella toimiva, koska se oli helposti omaksuttava ympäristö myös kirjaston työntekijöille.



Uusi rasti

Tapahtuma

Valitse tapahtuma: ▼ Tapahtuma 1

Tyyppi: ▼ Kuva

Median polku: http://page.domain...

Kysymys: ▼ Kysymys 1

Markkeri

Valitse markkeri: ▼ Marker_01

Peruuta Luo rasti

KUVIO 3. Luonnos hallintapaneelin rastinluonti-ikkunasta

Hallintapaneelin oli tarjottava monipuoliset hallintamahdollisuudet kaikille sovelluksen vaatimille sisällöille. Opinnäytetyön kirjoitushetkellä hallintapaneelista oli olemassa käytännössä vain luonnoksia (KUVIO 23). Hallintapaneelin kehitys jatkui kuitenkin tämän jälkeen.

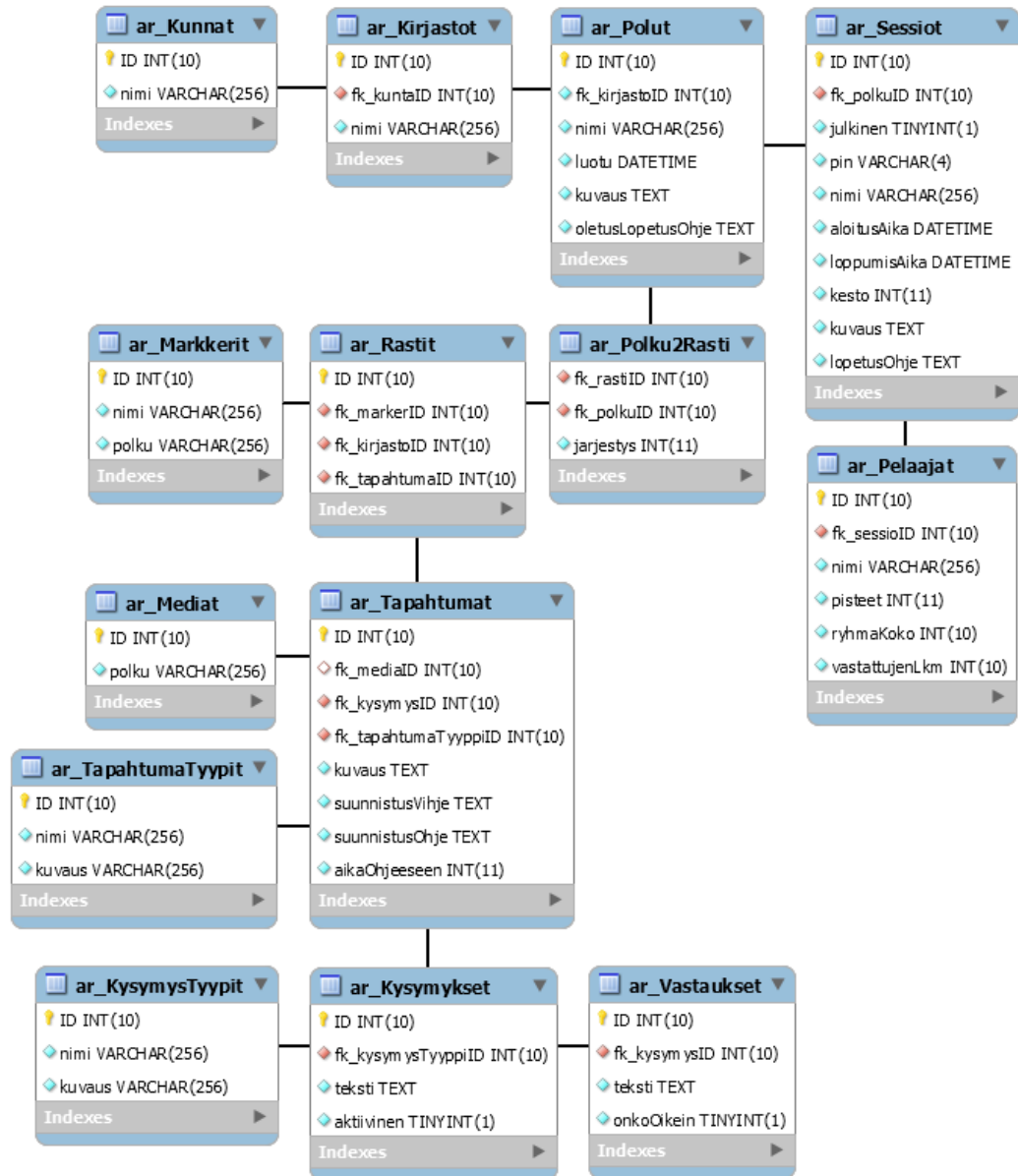
6.3 Tietokanta

Tietokannan suunnittelu aloitettiin heti projektin alussa. Suunnittelussa oli mietittävä tietokannalle toimiva perusrakenne ja järkevä tapa tallentaa pelaajien tilastoja. Suunnittelussa oli myös otettava huomioon erilaisten kysymys- ja vastaustyyppien toteuttaminen rasteille.

Tietokannaksi valittiin yleinen MYSQL-tietokanta. Tietokantamoottorina käytettiin InnoDB:tä, joka tukee esimerkiksi viiteavaimia. Niitä tarvitaan relaatioiden luomiseksi eri tietokantataulujen välille.

Tietokannan keskeisimpiä tauluja olivat Sessiot, Polut ja Rastit. Näihin tauluihin on talletettu sovelluksen toiminnan kannalta oleellisimpia tietoja. Näiden taulujen vieressä näkyy niihin liittyvät muut taulut (KUVIO 24). Esimerkiksi Rastit-tilusta näkee, mikä markkeri aktivoi rastiin liitetyn tapahtuman.

Tietokannan kehitystyö oli vielä keskeneräinen opinnäytetyön kirjoitusvaiheessa, joten tietokantakuvauksessa ei ole esimerkiksi tauluja erilaisten tilastojen ja pelaajien vastauksien tallentamiseen.



KUVIO 4. Tietokannan rakenne

6.4 Sovelluksen ja palvelimen rajapinta

Sovellus keskustelelee palvelimen kanssa aina hakiessaan uusia rasteja tai ladatessaan kuvia. Yhteys päätettiin toteuttaa palvelimella sijaitsevan rajapinnan avulla.

Sessioden hakeminen

Hakee kaikki julkiset sessiot, jotka näytetään mobiililaitteessa, kun pelaaja haluaa pelata yksin.

Parametrit

Nimi	Tyyppi	Kuvaus/arvo
toiminto	string	haeSessiot

Palautusarvot

Nimi	Tyyppi	Kuvaus/arvo
onnistuiiko	int	1/0
viesti	string	esim. virheviesti
sessiot	array	lista sessioista
Session rakenne		
id	int	session ID
polkulD	int	polun ID
julkinen	tinyint	1/0
pin	string	pin-koodi
nimi	string	session nimi
alkamisAika	datetime	
loppumisAika	datetime	
kesto	int	polun kesto minuuteissa
kuvaus	string	session kuvaus
lopetusOhje	string	polun lopetusohje

KUVIO 5. Esimerkki sovelluksen ja palvelimen välisestä rajapinnasta

Web-palvelimella sijaitseva rajapinta on kirjoitettu php-ohjelmointikielellä. Sovellus tekee haut rajapintaan HTTP-protokollalla POST-metodilla ja saa vastauksen JSON-muotoisena tekstinä. Kuvat ja muut suuremmat tiedostot sovellus hakee suoraan palvelimelta niiden URL-osoitteen perusteella. Rajapinnasta tehtiin kuvaus, jonka perusteella sovellusta ja hallintapaneelia pystyttiin kehittämään samanaikaisesti (KUVIO 25).

6.5 Sessiot

Palvelimella sijaitsevasta hallintapaneelista voidaan luoda pelisessioita sekä vierailleville kouluryhmille että yksittäisille kirjastossa kävijöille. Jokaiselle sessiolle on mahdollista määritellä sen aukeamis- ja päättymisaika sekä sen kesto. Näin on mahdollista luoda sessioita etukäteen ja varmistaa, että ne ovat auki oikeaan aikaan. Sessioiden luonti tapahtuu verkkoselaimella toimivasta hallintapaneelista.

Sessioiden päätarkoituksena on mahdollistaa yksittäisten koululaisryhmien pääseminen samaan peliin. Sessiot voidaan suojata pin-koodilla, joka käyttäjän on syötettävä ennen kuin hän voi liittyä sessioon.

Sessioita on myös mahdollista luoda julkisiksi, jolloin kuka tahansa kirjastossa kävijä voi liittyä niihin ja pelata polun läpi. Sovelluksen käyttäjä näkee listan sessioista ja voi liittyä julkisiin sessioihin ilman pin-koodia.

Kun session aika on kulunut loppuun, pelaajien tekemistä vastauksista saadaan yhteenveto. Sovelluksen käyttäjät näkevät yhteenvedosta yksinkertaistetun version, jossa käyttäjä näkee omat pisteensä ja sijoittumisensa muihin nähden. Hallintapaneelin kautta nähdään kaikki pelaajien tekemät vastukset, joihin voi lukeutua esimerkiksi rasteilla otetut valokuvat. Hallintapaneelista nähdään myös tilastoina, kuinka kauan rastien ratkaisemiseen ja rasteilla siirtymiseen on mennyt aikaa.

6.6 Polut ja rastit

Jokaiseen sessioon kuuluu yksi polku. Polut sisältävät rasteja, joita sovelluksen käyttäjän ratkaisevat järjestyksessä. Rastit sisältävät erityyppisiä kysymyksiä ja vastauksia niihin. Kysymyksiin lukeutuu tavallisten monivalintakysymyksien lisäksi kuvia ja videoita, joiden esittämiin haasteisiin käyttäjien on vastattava.

Myös vastaustyyppejä on useita erilaisia. Käyttäjän on esimerkiksi otettava kuva tai nauhoitettava lyhyt äänitiedosto (KUVIO 26). Näitä tiedostoja voidaan tarkastella session loputtua hallintapaneelin kautta.

TapahtumaTyypit	KysymysTyypit
nimi	nimi
Teksti	Yksi oikein
Kuva	Monta oikein
Video	Vapaa teksti
Aani	Kuva
	Aaniklippi

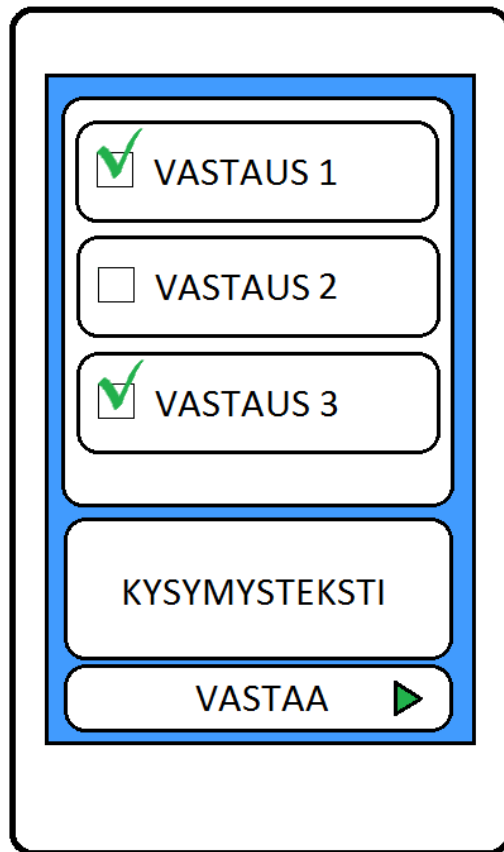
KUVIO 6. Tapahtumien ja kysymysten tyypit

6.7 Rastien lataaminen sovellukseen

Kun pelaaja on valinnut session, johon hän aikoo liittyä, sovellus alkaa ladata kyseisen session tietoja. Yksi tärkeimmistä tiedoista on session polku, joka sisältää tiedot kaikista polulle kuuluvista rasteista. Sovellus lataa heti ensimmäisen rastin, jotta se olisi valmiina, kun pelaaja on syöttänyt nimimerkinsä ja aloittanut pelin. Tämän jälkeen sovellus lataa rastit ja niiden sisällöt järjestyksessä taustalla.

6.8 Rastien sisällön näyttäminen

Kun pelaaja löytää oikean tunnistuskohteen, sovellus aktivoi siihen liittyvän rastin. Sovellus luo rastin tyyppin mukaan uuden peliobjektin, jonka avulla kysymykset ja vastaukset voidaan näyttää.



KUVIO 7. Yksinkertainen monivalintakysymys sovelluksessa

Rastipohjat hyödyntävät uuden käyttöliittymän tarjoamia komponentteja, joiden avulla on helppo näyttää usean kokoisia ja määrältään vaihtelevia sisältöjä. Esimerkiksi kysymyksellä saattaa olla mikä tahansa määrä vastausvaihtoehtoja, ja sovelluksen on pystyttävä näyttämään ne kaikki. (KUVIO 27.)

Videoiden näyttämisessä käytettiin Vuforia tarjoamaa ominaisuutta videoiden toistamiseen. Tämä oli nopein ratkaisu videoiden näyttämiseen Android-käyttöjärjestelmällä, koska oletuksena Unity avaa videot Androidin tarjoamalla erillisellä sovelluksella. Tämä ei sopinut suunniteltuun sovelluksen toimintalogiikkaan, koska tarkoituksena oli toistaa videot Unityllä tehdyn sovelluksen sisällä.

Yksinkertaisin rastipohja sisälsi pelkän ruudulle piirrettävän käyttöliittymän, jossa rastin sisältö näkyi (KUVIO 27). Käyttöliittymä oli jaettu kahteen osaan, joista toisessa näytettiin kysymys ja toisessa vastausvaihtoehdot.

Rasteille tehtiin myös kolmiulotteisuutta hyödyntäviä pohjia. Näitä olivat esimerkiksi kirja ja kääre, jotka sisälsivät myös aukeamisanimaation, jonka jälkeen rastin sisältö näkyi niiden sisällä.

7 YHTEENVETO

Opinnäytetyön tavoitteena oli tutustua lisättyyn todellisuuteen ja toteuttaa sitä hyödyntävä sovellus Lahden kaupunginkirjastolle. Tämän lisäksi tavoitteena oli tutustua lisättyyn todellisuuteen ja sen historiaan.

Sovelluksen toteutus vaati tutustumista Unity3D-pelimoottoriin ja lisätyn todellisuuden Vuforia-kirjastoon. Vuforian verkkosivuilla olevien ohjeiden avulla kaikkien perustoimintojen testaus onnistui nopeasti. Itse sovelluksen tekeminen vaati alussa paljon suunnittelua etenkin palvelimella olevan tietokannan osalta. Seuraavaksi toteutettiin sovelluksen ja palvelimen välinen yhteys, joka mahdollisti sovelluksen testauksen oikealla palvelimelta tulevalla datalla. Tämän jälkeen sovelluksesta tehtiin älypuhelimella testattava versio, joka sisälsi suurimman osan perustoiminnoista.

Toimivan testiversion jälkeen mobiilisovelluksen kehitys jatkui ja hallintapaneelin toteutus aloitettiin. Hallintapaneeli oli vielä keskeneräinen opinnäytetyön kirjoitusaikaan, mutta itse kehitysprojekti jatkui sen jälkeenkkin. Aikataulun puolesta myös hallintapaneelin valmistumiselle jäi aikaa.

Vuforia tarjosi hyvin toimivat työkalut lisätyn todellisuuden aikaansaamiseksi. Teknologia oli kehittynyt jo siihen vaiheeseen, että kuvantunnistus toimi nopeasti ja esimerkiksi tunnistuskohteen päällä oleva virtuaalinen 3d-malli pysyi oikeassa asennossa, vaikka tunnistuskohdetta liikuteltiin. Lisättyllä todellisuudella oli siis mahdollista saada aikaan vakuuttavan näköisiä virtuaalisia elementtejä.

Tämän tasoista lisättyä todellisuutta voidaan käyttää jo useilla eri aloilla niin, että siitä on oikeaa hyötyä. Esimerkiksi lääketieteen alalla lisätty todellisuus ei ole vielä kovin yleistä, mutta sen tarjoamat mahdollisuudet ovat suuret. Lisätyn todellisuuden sovellusten näkyvyys edistää muidenkin tietoisuutta siitä, jolloin sovelluksien määrä kasvaa ja uusia innovaatioita syntyy.

LÄHTEET

Ackerman, J. 2000. UNC Ultrasound/Medical Augmented Reality Research [viitattu 20.3.2016]. Saatavissa:

<http://www.cs.unc.edu/Research/us/>

ARToolKit. 2016. About ARToolKit [viitattu 18.3.2016]. Saatavissa:

<http://artoolkit.org/about-artoolkit>

Azuma, R. 2005. Registration Errors in Augmented Reality [viitattu

18.3.2016]. Saatavissa: https://www.cs.unc.edu/~azuma/azuma_AR.html

Bryon, F. 2010. Myron Krueger [viitattu 17.3.2016]. Saatavissa:

<http://thedigitalage.pbworks.com/w/page/22039083/Myron%20Krueger>

Burns, M. 2010. GM's Enhanced Vision System brings augmented reality to vehicle HUDs [viitattu 20.3.2016]. Saatavissa:

<http://techcrunch.com/2010/03/17/gms-enhanced-vision-system-brings-augmented-reality-to-vehicle-huds/>

Busta, H. 2015. Three Augmented and Virtual Reality Apps for Design and Construction [viitattu 20.3.2016]. Saatavissa:

http://www.architectmagazine.com/technology/products/three-augmented-and-virtual-reality-apps-for-design-and-construction_o

Foyle, D., Andre, A. & Hooey, B. 2005. Situation Awareness in an Augmented Reality Cockpit: Design, Viewpoints and Cognitive Glue [viitattu 20.3.2016]. Saatavissa:

https://www.researchgate.net/profile/Becky_Hooey/publication/228367987_Situation_awareness_in_an_augmented_reality_cockpit_Design_viewpoints_and_cognitive_glue/links/00b7d53063a13c7a83000000.pdf

Horse, J. 2011. Lego Augmented Packaging Let You See The Finished Product (video) [viitattu 18.3.2016]. Saatavissa: <http://www.geekygadgets.com/lego-augmented-packaging-let-you-see-the-finish-product-video-19-09-2011/>

<http://www.geekygadgets.com/lego-augmented-packaging-let-you-see-the-finish-product-video-19-09-2011/>

Kato, H. & Billinghamurst, M. 1999. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System [viitattu 18.3.2016]. Saatavissa:

https://www.researchgate.net/profile/Hirokazu_Kato2/publication/3824580_Marker_tracking_and_HMD_calibration_for_a_video-based_augmentedreality_conferencing_system/links/02e7e5239b0cba61e000000.pdf

Lahden kaupunginkirjasto. 2015. Reality check! - Lisättyä todellisuutta kirjastoihin [viitattu 25.2.2016]. Saatavissa:

<http://hankkeet.kirjastot.fi/hanke/reality-check-lis%C3%A4tty%C3%A4-todellisuutta-kirjastoihin>

Lyteshot. 2015. LyteShot Live Action Gaming [viitattu 20.3.2016]. Saatavissa: <http://www.lyteshot.com/>

Maier, P., Klinker, G. & Tonnis, M. 2009. Augmented Reality for teaching spatial relations [viitattu 20.3.2016]. Saatavissa:

<http://far.in.tum.de/pub/maierp2009ijas/maierp2009ijas.pdf>

Microsoft. 2016a. IEnumerator Interface [viitattu 13.4.2016]. Saatavissa:

<https://msdn.microsoft.com/en-us/library/system.collections.ienumerator.aspx>

Microsoft. 2016b. Yield (C# Reference) [viitattu 13.4.2016]. Saatavissa:

<https://msdn.microsoft.com/en-us/library/9k7k7cf0.aspx>

Milgram, P. & Kishino, F. 1994. A Taxonomy Of Mixed Reality Displays [viitattu 14.3.2016]. Saatavissa:

http://web.cs.wpi.edu/~gogo/hive/papers/Milgram_IEICE_1994.pdf

Miyake, R., Zeman, H., Duarte, F., Kikuchi, R., Ramacciotti, E., Lovhoiden, G. & Vrancken, C. 2006. Vein imaging: a new method of near infrared imaging, where a processed image is projected onto the skin for the enhancement of vein treatment [viitattu 20.3.2016]. Saatavissa:

<http://www.ncbi.nlm.nih.gov/pubmed/16918565>

Payatagool, C. 2008. Theory and Research in HCI: Morton Heilig, Pioneer in Virtual Reality Research [viitattu 17.3.2016]. Saatavissa: http://www.telepresenceoptions.com/2008/09/theory_and_research_in_hci_mor/

PTC. 2016a. Cylinder Targets [viitattu 20.3.2016]. Saatavissa: <https://developer.vuforia.com/library//articles/Training/Cylinder-Target-Guide>

PTC. 2016b. Devices [viitattu 20.3.2016]. Saatavissa: <https://www.vuforia.com/Devices>

PTC. 2016c. Features [viitattu 20.3.2016]. Saatavissa: <https://www.vuforia.com/Features>

PTC. 2016d. How To Implement Text Recognition In Unity [viitattu 9.4.2016]. Saatavissa: <https://developer.vuforia.com/library//articles/Solution/How-To-Implement-Text-Recognition-in-Unity>

PTC. 2016e. Image Targets [viitattu 20.3.2016]. Saatavissa: <https://developer.vuforia.com/library//articles/Training/Image-Target-Guide>

PTC. 2016f. Multi-Targets [viitattu 20.3.2016]. Saatavissa: <https://developer.vuforia.com/library//articles/Training/Multi-Target-Guide>

PTC. 2016g. Natural Features and Image Ratings [viitattu 25.2.2016]. Saatavissa: <https://developer.vuforia.com/library/articles/Solution/Natural-Features-and-Ratings>

PTC. 2016h. Object Recognition [viitattu 20.3.2016]. Saatavissa: <https://developer.vuforia.com/library//articles/Training/Object-Recognition>

PTC. 2016i. Pricing Overview [viitattu 25.2.2016]. Saatavissa: <https://developer.vuforia.com/pricing>

PTC. 2016j. Target Manager [viitattu 26.2.2016]. Saatavissa: <https://developer.vuforia.com/target-manager>

- PTC. 2016k. Text Recognition [viitattu 27.2.2016]. Saatavissa: <https://developer.vuforia.com/library/articles/Training/Text-Recognition-Guide>
- PTC. 2016l. Vuforia 5 SDK [viitattu 25.2.2016]. Saatavissa: <https://developer.vuforia.com/downloads/sdk>
- PTC. 2016m. Vuforia Object Scanner [viitattu 20.3.2016]. Saatavissa: <https://developer.vuforia.com/library/articles/Training/Vuforia-Object-Scanner-Users-Guide>
- PTC. 2016n. Vuforia Web Services [viitattu 26.2.2016]. Saatavissa: <https://developer.vuforia.com/library//articles/Training/Using-the-VWS-API>
- Raskar, R., Welch, G. & Fuchs, H. 1998. Spatially Augmented Reality [viitattu 18.3.2016]. Saatavissa: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.4260&rep=rep1&type=pdf>
- Sutherland, I. 1968. A head-mounted three dimensional display [viitattu 17.3.2016]. Saatavissa: <http://design.osu.edu/carlson/history/PDFs/p757-sutherland.pdf>
- Thor, P. 2015. Prototype of Augmented Reality through the Samsung Gear VR [viitattu 18.3.2016]. Saatavissa: <https://www.youtube.com/watch?v=E-oMPq4a3R8>
- Unity Technologies. 2016a. Basic Layout [viitattu 20.3.2016]. Saatavissa: <http://docs.unity3d.com/Manual/UIBasicLayout.html>
- Unity Technologies. 2016b. Canvas [viitattu 20.3.2016]. Saatavissa: <http://docs.unity3d.com/Manual/UICanvas.html>
- Unity Technologies. 2016c. Get Unity [viitattu 27.2.2016]. Saatavissa: <https://unity3d.com/get-unity>
- Unity Technologies. 2015d. Image [viitattu 20.3.2016]. Saatavissa: <http://docs.unity3d.com/Manual/script-Image.html>

Unity Technologies. 2015e. JSON Serialization [viitattu 25.2.2016]. Saatavissa: <http://docs.unity3d.com/Manual/JSONSerialization.html>

Unity Technologies. 2015f. Multiplatform [viitattu 18.3.2016]. Saatavissa: <https://unity3d.com/unity/multiplatform>

Unity Technologies. 2015g. Raw Image [viitattu 20.3.2016]. Saatavissa: <http://docs.unity3d.com/Manual/script-RawImage.html>

Unity Technologies. 2015h. Text [viitattu 20.3.2016]. Saatavissa: <http://docs.unity3d.com/Manual/script-Text.html>

Unity Technologies. 2015i. Unity for gambling: Develop once develop anywhere [viitattu 20.3.2016]. Saatavissa: <https://unity3d.com/unity/industries/gambling>

Unity Technologies. 2015j. WWW [viitattu 25.2.2016]. Saatavissa: <http://docs.unity3d.com/ScriptReference/WWW.html>

Unity Technologies. 2015k. WWWForm [viitattu 25.2.2016]. Saatavissa: <http://docs.unity3d.com/ScriptReference/WWWForm.html>

Volkswagen AG. 2016. Applications [viitattu 18.3.2016]. Saatavissa: <http://www.tracking-challenge.de/content/tc/content/en/Object-Tracking/Applications.html>

LIITTEET