

Elizabeth Tran

Puheentunnistus virtuaalitodellisuudessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinööryö

07.11.2018

Tekijä Otsikko	Elizabeth Tran Puheentunnistus virtuaalitodellisuudessa
Sivumäärä Aika	35 sivua 07.11.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintäteknikka
Ammatillinen pääaine	Pelisovellukset
Ohjaajat	Lehtori Antti Laiho
<p>Insinööriyön tavoitteena oli toteuttaa puheentunnistusta ja virtuaalitodellisuutta hyödyntävä sovellus. Sovellus sisältää suunnitelman mukaan mahdollisuuden luoda, muokata ja poistaa erilaisia kappaleita kuten kuutioita ja palloja puheentunnistuksen avulla virtuaalimaailmassa. Työ tehtiin Unity-pelimootorilla ja ohjelmointikielenä käytettiin C#-kieltä. IBM Watson -tekoälypalveluita hyödynnettiin puheentunnistuksen implementoinnissa. Sovellus on suunniteltu Android-laitteelle, mutta sitä voidaan muokata muille alustoille sopiviksi, esimerkiksi HTC Vivelle tai iOS-laitteelle.</p> <p>Tuloksena syntyi IBM Watsonin palveluita hyödyntävä sovellus, jossa käyttäjä voi luoda ja poistaa kuutioita puheentunnistuksen avulla virtuaalitodellisuudessa. Kaikkia ominaisuuksia ei valitettavasti saatu implementoitua sovellukseen, mutta se tarjoaa hyvän pohjan laajentamista varten ja mahdollisuuden tutustua kyseiseen aiheeseen.</p> <p>Sovelluksen avulla tutkittiin sitä, miten puheentunnistusta voitaisiin hyödyntää virtuaalimaailmassa. Samalla myös tarkasteltiin erilaisten puheentunnistusta tarjoavien palveluiden toimintaa ja helppokäyttöisyyttä sekä miten palveluiden yhdistäminen pelimootoriin tapahtuu.</p>	
Avainsanat	VR, virtuaalitodellisuus, puheentunnistus, Unity, IBM Watson

Author Title	Elizabeth Tran Speech Recognition in Virtual Reality
Number of Pages Date	35 pages 7 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Game Applications
Instructors	Antti Laiho (Senior Lecturer)
<p>The purpose of this thesis was to create an application that utilizes speech recognition and virtual reality. According to the plan, the application includes the ability to create, edit and delete different 3D-models such as cubes and balls with the use of speech recognition in a virtual reality world. The project was done with Unity-game engine and the programming language was C#. IBM Watson -services were used in implementing speech recognition. The application was planned for an Android device, but it can be adjusted for other platforms such as HTC Vive or iOS.</p> <p>In the end, the final version of the application only includes the ability to create and delete cubes. Unfortunately, other functionalities were not implemented, but this application can be used as a base project for creating similar applications and also offers possibilities to expand to different use cases of speech recognition or virtual reality.</p> <p>The benefits and possibilities of using speech recognition in virtual reality were explored with the help of the application. At the same time the functionalities of different services offering speech recognition were researched, e.g. how easily it could be integrated to a game engine and the user-friendliness of using the services.</p>	
Keywords	VR, Virtual Reality, Speech Recognition, Unity, IBM Watson

Sisällys

Lyhenteet

1	Johdanto	1
2	Virtuaalitodellisuus ja puheentunnistus	2
2.1	Virtuaalitodellisuuden historia	2
2.2	Puheentunnistuksen periaate	7
2.3	Puheentunnistuksen yleisyys laitteissa ja sovelluksissa	8
2.4	Tilannekatsaus virtuaalitodellisuuteen	10
3	Sovelluksen suunnittelu virtuaalilaseille	12
3.1	Sovelluksen vaatimukset	12
3.2	Ohjelmointirajapinnan valitseminen puheentunnistukselle	14
3.3	Virtuaalitodellisuuden yleisimpien ongelmien minimointi	19
4	Puheentunnistusovelluksen toteutus	20
4.1	Komentojen luominen ja sen toiminta	20
4.2	Puheentunnistuksen soveltaminen Unityyn	23
4.3	Sovelluksen testaus virtuaalilaseilla	27
5	Tulokset ja sovelluksen mahdollisuudet	28
5.1	Sovelluksen analysointi	28
5.2	Sovelluksen hyödyt ja käyttömahdollisuudet tulevaisuudessa	29
6	Yhteenveto	31
	Lähteet	32

Lyhenteet

3D	Three-dimensional space. Kolmiulotteisuus.
API	Application Programming Interface. Ohjelmointirajapinta, jonka avulla eri ohjelmat voivat keskustella ja vaihtaa tietoja keskenään.
HMD	Head Mounted Display. Päähän laitettava laite, jossa on näyttö. Virtuaalitodellisuuslaitteet sisältävät HMD-laitteen.
HTTP	Hypertext Transfer Protocol. Protokolla, jonka avulla erilaiset selaimet ja palvelimet käyttävät tiedonsiirtoa.
IDE	Integrated Development Environment. Ohjelmointiympäristö, jolla voidaan suunnitella ja toteuttaa ohjelmistoja.
JSON	JavaScript Object Notation. Tiedostomuoto tiedonvälitykseen.
REST	Representational State Transfer. HTTP-protokollaan perustuva arkkitehtuuri erilaisten rajapintojen toteuttamiseen.
SDK	Software Development Kit. Paketti, joka sisältää monia aputyökaluja esimerkiksi erilaisten sovelluksien tai ohjelmistokehysten luomista varten.
VR	Virtual Reality. Virtuaalitodellisuus.

1 Johdanto

Virtuaalitodellisuus on ollut yksi kuumimpia trendejä teknologian maailmassa, sillä se toi ennennäkemätöntä teknologiaa kuluttajien käsille. Vuosien varrella laatu on selvästi parantunut sekä lasien grafiikkalaatu epäilemättä kehittynyt tarkemmaksi. Se tarjoaa uuden tavan olla vuorovaikutuksessa virtuaalimaailman kanssa ja olla kosketuksissa virtuaalisen datan kanssa.

Samaan aikaan myös puheentunnistuksen laatu on parantunut merkittävästi vuosien varrella ja sen käyttö erilaisissa laitteissa on yleistynyt. Hyviä esimerkkejä näistä laitteista ovat älypuhelimet, joissa virtuaaliset avustajat kuten Siri ja Google Now antavat nykyään tarkempia tuloksia ja ymmärtävät luonnollista kieltä paremmin kuin koskaan aiemmin.

Tämä insinööriyö yhdistää yllä olevat asiat tutkimalla, miten puheentunnistusta voidaan hyödyntää virtuaalitodellisuudessa. Puheentunnistuksen avulla voidaan suorittaa toimintoja pelkästään puhumalla, jolloin VR-sovellusten käyttäminen helpottuu ja mahdollisesti myös nopeutuu. Insinööriyö tutkii nykypäivän teknologian valmiuksia ja samalla myös etsii mahdollisia käyttötapoja tulevaisuutta varten.

Työn aihe muovautui tekijän kiinnostuksesta virtuaalitodellisuuteen. Tekijä on työskennellyt parin VR-kouluprojektien parissa, josta kokemusta osittain tuli. Puheentunnistuksen lisääminen VR:ään oli ideana mielenkiintoinen, sillä kyseiset osa-alueet ovat jatkuvasti kehittymässä parempaan suuntaan ja ovat teknologia-alan lupaavimpia aiheita.

Insinööriyö esittelee puheentunnistuksen historiaa ja sen jälkeen periaatetta siitä, miten puhe tunnistetaan ja käsitellään. Samalla myös tarkastellaan, mistä virtuaalitodellisuus sai alkunsa ja missä sitä nykyään käytetään. Sen jälkeen löytyvät eri vaiheet vastaavanlaisen sovelluksen tekemiseen sekä kerrotaan hieman sovellusta varten harkituista pilvi- ja tekoälypalveluista.

2 Virtuaalitodellisuus ja puheentunnistus

2.1 Virtuaalitodellisuuden historia

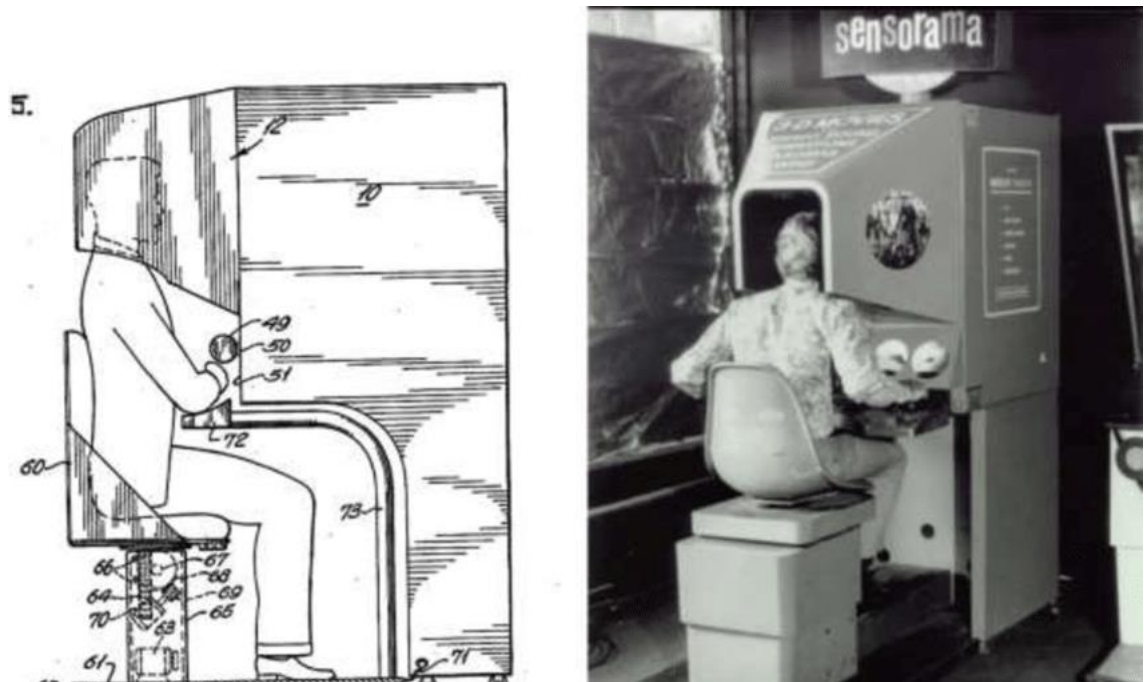
Ensimmäinen askel kohti virtuaalitodellisuutta tapahtui jo 1800-luvulla. Brittiläinen tiedemies ja keksijä Charles Wheatstone kehitti stereoskoopin, joka käytti peilejä hyväksi luodakseen 3D-immersion käyttäjälle. Stereoskooppi tarvitsee neljä peiliä, jotka asetetaan vastakkain pareittain 45 asteen kulmaan. Katsojan pitää katsoa suoraan peileihin, ja tuloksena peilit synnyttivät kolmiulotteisen efektin. Stereoskooppi oli yksi aikansa mullistavimpia keksintöjä, koska se oli ensimmäinen laite, joka esitteli kaksikulotteisten kuvien katsomisen syvyydellä ja immersioilla manipuloimalla syvyytnäköä. Sillä on myös merkittävä rooli nykyajan teknologiassa, sillä hieman identtistä stereoskoopin tekniikkaa käytetään muun muassa Googlen Cardboard-laseissa [Dybsky 2017]. Stereoskoopeista kehitettiin monia eri versioita, kuten kuvassa 1 näkyy: jokainen versio alkoi vähitellen muistuttaa nykypäivän virtuaalitodellisuuslaseja.



Kuva 1. Eri versioita stereoskoopista vanhimmasta uusimpaan: Wheatstone (1838), Brewster (1849) ja Holmes (1861).

Kirjailija Stanley G. Weinbaum kirjoitti tieteisfiktiivisen tarinan, *Pygmalion's Spectacles*, vuonna 1935. Tarinassa yksi henkilöistä on keksinyt lasit, joiden avulla käyttäjä voi kokea fiktiivisen maailman hologrammien avulla. Siinä voi puhua varjoille (eli fiktiivisen maailman hahmoille), jotka vastaavat takaisin ja maailmaan voi tutustua koskettamalla, haistamalla tai maistamalla. Vaikka Weinbaumin tarkka kuvaus mahdollisista virtuaalilaseista oli fiktiota, se oli kuitenkin ensimmäinen vaikuttava ja suuntaa antava idea siitä, minkälaiset virtuaalitodellisuuslasit voisivat olla [Dybsky 2017; Brown 2017]. 20 vuoden jälkeen kehitettiin laite, joka käytti kaikkia tuntoaisteja jollain tavalla hyödyksi. Sensorama oli iso ja arcade-pelikoneiden muotoinen laite, joka tarjosi käyttäjilleen immersiiivisen kokemuksen monien pienlaitteiden kuten tuulettimien ja hajugeneraattoreiden avulla [Dybsky 2017; Turi 2014]. Kuvassa 2 vasemmalla näkyy Heiligen luonnos kyseisestä laitteesta, ja oikealla laite on koehenkilön testattavana. Laite

ei jäänyt kehittäjän Morton Heiligin ainoaksi keksinnöksi, sillä hänen toinen laitteensa oli virtuaalitodellisuuslasit.



Kuva 2. Vasemmalla Heiligin kehittämä luonnos Sensorama-laitteesta ja oikealla itse kyseinen laite testauksessa.

Heiligin virtuaalitodellisuuslasien muotoilu oli melko samanlainen verrattuna nykypäivän lasiin. Lasit laitettiin päähän ja niissä oli 3D-stereoskooppinen näyttö, leveä näkökenttä sekä stereoäännet, mutta siitä puuttui liikkeentunnistin. Uudesta laitteesta ei kuitenkaan tullut suurta hittiä, mutta tästä huolimatta laitetta pidetään maailman ensimmäisinä virtuaalilaseina. Vasta vuonna 1968 tietokonegraafikon Ivan Sutherlandin lasista tuli suosittuja: ”*Sword of Damocles*”-lasit voitiin yhdistää tietokoneeseen, jossa näytettiin lasien videokuvaa yksinkertaisilla kappaleilla. Samalla jos käyttäjä käänsi päätään, kuva liikkui sen mukana. Rajoittavana tekijänä oli kuitenkin sen pelottava muotoilu ja suuri paino, kuten kuvassa 3 näkyy, minkä takia se jäi pelkäksi laboratorioprojektiksi [Dormehl 2017].

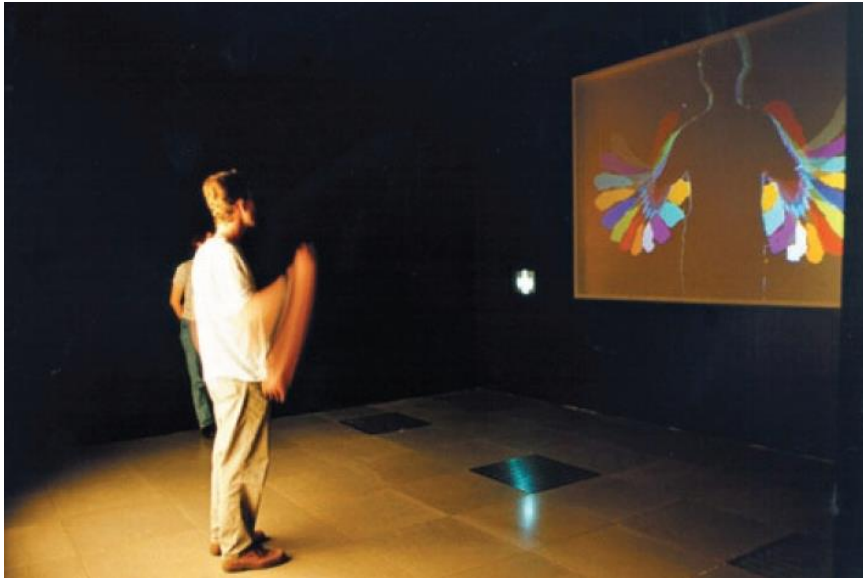


Kuva 3. "Sword of Damocles" testissä.

Armeija oli jo pitkään kiinnostunut virtuaalitodellisuuteen liittyvistä laitteista, koska niillä oli suuri potentiaali sotilaiden tai lentäjien kouluttamisessa. 70-luvun aikana armeijalla oli ongelmia kouluttaa lentäjiä käyttämään uusia hävittäjiä, koska hävittäjiä kehitettiin jatkuvasti käyttämään uusinta teknologiaa, jolloin niiden ohjaaminen ja käyttäminen muuttui vaikeammaksi. Armeija halusi jonkinlaisen ratkaisun suureen ongelmaan, joten armeijan oma tutkija Thomas Furness halusi kehittää ratkaisun rakentamalla toimivan ja hyödyllisen lentosimulaattorin armeijalle. Hän kehitti projektiaan monta vuosikymmentä, ja ensimmäiset versiot laitteesta olivat päähän laitettavat virtuaalilasit. Myöhemmin hän lisäsi simulaattorin laitteeseen, jonka nimeksi tuli "*Super Cockpit*". Laite toimi niin, että päähän laitettiin virtuaalilasit ja käyttäjä istui ohjaamossa. Käyttäjälle näytettiin 3D-generoitu kartta kaikkine mittareineen ja tietoineen. Laitetta oli mahdollista kontrolloida eleillä, puheella ja silmillä. [Dybsky 2017.]

Käsitettä "virtuaalitodellisuus" ei ollut vielä olemassa ennen 80-lukua, vaikka virtuaalitodellisuuteen liittyviä laitteita oli jo kehitetty. Tietokonetaiteilija Myron W. Krueger kollegoidensa kanssa teki kokeilun 70-luvun puolivälissä, jossa heidän tavoitteenaan oli luoda interaktiivinen virtuaalimaailma. Kokeilun aikana syntyi projekti *Videoplace*, joka käytti projektoria koneellisesti luodun maailman ja oikean maailman yhdistämiseen. Kuvassa 4 näkyy projektin käyttötapa, eli käyttäjä seisoo seinän edessä ja hänen eteensä projisoidaan videokuvaa. Videokuvassa käyttäjä on "liitetty" virtuaalimaailmaan, jolloin hän pystyy erilaisin elein olla vuorovaikutuksessa videokuvan kanssa. Krueger esitteli käsitteen "Artificial Reality" vuonna 1983 ilmestyneessä kirjassaan *Artificial Reality*, joka oli mahdollisesti ensimmäisiä käsitteitä kyseiselle

aiheelle [Krueger ym. 1985]. "Virtuaalitodellisuus" käsitteenä tuli ilmi vasta 80-luvulla, kun tietojenkäsittelytieteilijä Jaron Lanier käytti käsitettä tutkiessaan kyseistä teknologiaa. Hänen perustamansa yrityksen VPL Research tekivät ensimmäisiä VR-kulutuslaitteita, joista tuli merkittäviä esimerkkejä nykypäivän VR-laitteille. [Rubin 2017].



Kuva 4. Videoplace käytti projektoria, tietokonetta ja kuvaa hyväksi luodakseen interaktiivisen kokemuksen.

Virtuaalitodellisuuslaitteet rantautuivat pelimarkkinoille 90-luvun alussa, sillä yritys Virtuality Group esitteli virtuaalitodellisuutta käyttäviä pelikoneita. Laitteita oli kahdenlaisia: laitteita, joissa istutaan ja toisessa seistään. Molemmissa pelaajalle laitettiin virtuaalilasit, jotka reagoivat pään liikehdintään. Mukana olivat myös ohjaimet, jotka toimivat pelin sisällä pelaajan virtuaalisina käsinä. Pian myös peliyritys Sega tuli mukaan kilpailuun tuomalla omalle Sega Genesis -laitteelleen VR-lisäosan, kuten kuvassa 5 näkyy. Sega VR oli visiirin muotoinen virtuaalilasi, joka muistuttaa hyvin paljon nykypäivän virtuaalilaseja: päähän laitettava laite, jossa silmien kohdalla on "pieni, suljettu näyttö" [Dybsky 2017]. Valitettavasti lasit eivät koskaan tulleet markkinoille, koska Sega koki niiden olevan liian realistisia ja olivat huolissaan liiallisesta immersioista, joka saattaisi aiheuttaa mahdollisia tapaturmia käyttäjille (Dormehl 2017).



Kuva 5. Sega Genesis oli taloudellisesti epäonnistunut kokeilu.

Vuonna 2010 pieni startup-yritys Oculus VR (aiemmin *Oculus*) julkaisi Kickstarter-rahoituskampanjan tulevalle hankkeelleen, Rift-virtuaalitodellisuuslaseille. Yrityksen tavoitteena oli vain saada rahoitusta laitteen osille, jotta he saisivat rakennettua jonkinlaiset virtuaalilasit. Pian alle viikossa heidän tavoitteensa ylittyi 10-kertaisesti, mikä kertoi siitä, että kuluttajat sekä yritykset olivat kiinnostuneita laseista. Kuvassa 6 näkyvät Rift-lasit, jotka olivat edeltäjiään kompaktimmat, hienommat ja kevyemmät. [Dormehl 2017; Kumparak 2014]. Oculus oli kehittänyt sellaisen laitteen, joka antoi merkittävän jalansijan virtuaalitodellisuudelle asettua teknologian maailmaan, sillä suuryritykset alkoivat kiinnostua kyseisestä teknologiasta. Suuryritys Facebook päätyikin ostamaan Oculuksen vuonna 2014.



Kuva 6. Ensimmäinen kehittäjäversio Oculus Rift -laitteesta vuodelta 2013.

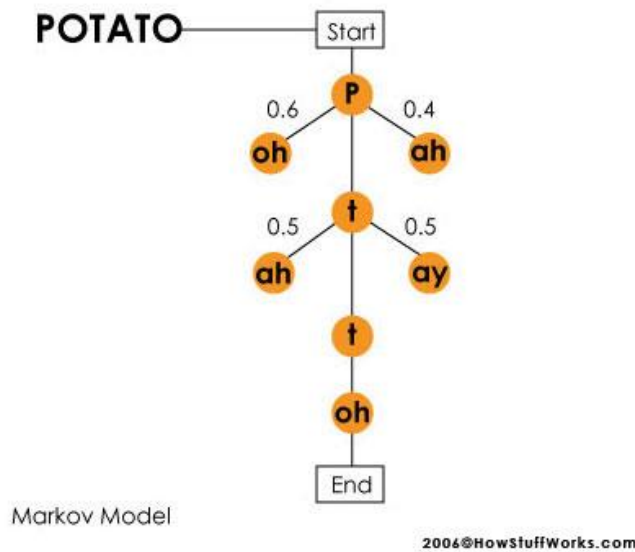
Virtuaalitodellisuuden suosio on ollut kasvavaa, sillä monet yritykset kehittävät omia laitteitaan ja sovelluksiaan markkinoille. Teknologialta odotettiin paljon siinä toivossa, että siitä syntyisi positiivisia hyötyjä monille aloille.

2.2 Puheentunnistuksen periaate

Kun jokin kappale tärisee tai värähtelee, se synnyttää ääntä ilmaan ja luo samalla tiheitä ääniaaltoja [Korpinen 2005]. Sama periaate pätee myös ihmisiin: kun ihminen puhuu, äänihuulet värähtelevät ja siten tuottavat puhetta. Puheentunnistusta käyttävät laitteet hyödyntävät tätä tunnistamalla ääniaallot ja tallentavat sen laitteeseen. Tallentamisen jälkeen puhesignaali digitoidaan eli muutetaan koneille ymmärrettäviksi tiedoiksi mittaamalla ääniaaltojen pituuksia ja erottelemalla turhat taustäänen pois signaalista. Muutettu puhesignaali pilkotaan sitten pieniin osiin, joista erotellaan klusiilit eli äänneet, jotka syntyvät, kun ilmavirta pysäytetään hetkellisesti ihmisen ääntöväylässä ja purkautuvat hetken kuluessa päästäen ilmavirran taas kulkemaan. Suomen kielessä klusiileja ovat esimerkiksi konsonantit k, t ja p. Erottelun aikana nämä osat vertaillaan foneemeihin eli äänneisiin kielen tunnistamiseksi. (Korpinen 2001.)

Vertailun jälkeen foneemeja tutkitaan tarkemmin kontekstin ymmärtämiseksi. Foneemeja analysoidaan tilastollisten mallien avulla, joissa käytetään todennäköisyyslaskentaa ja matemaattisia funktioita päättämään puheen tulos. Käytetyin malli on Markovin piilomalli, joka soveltuu erinomaisesti puheentunnistuksen tarkasteluun sen yksinkertaisuuden ja tehokkuuden vuoksi. Markovin piilomallissa sanat ovat kuin ketjuja, jotka koostuvat ketjun osista eli foneemeista. Sanaa muodostaessa ketjun osa haarautuu moniin mahdollisiin foneemeihin, joiden tavoitteena on täsmätä seuraavan foneemin kanssa. Näille osille annetaan sen jälkeen todennäköisyyspistemäärä, joka muodostuu siten, että koneelle annetaan paljon dataa, jonka avulla kone oppii ja osaa arvioida paremmin. Mitä korkeampi pistemäärä, sitä todennäköisemmin se muistuttaa eniten seuraavaa foneemia.

How Speech Recognition Works



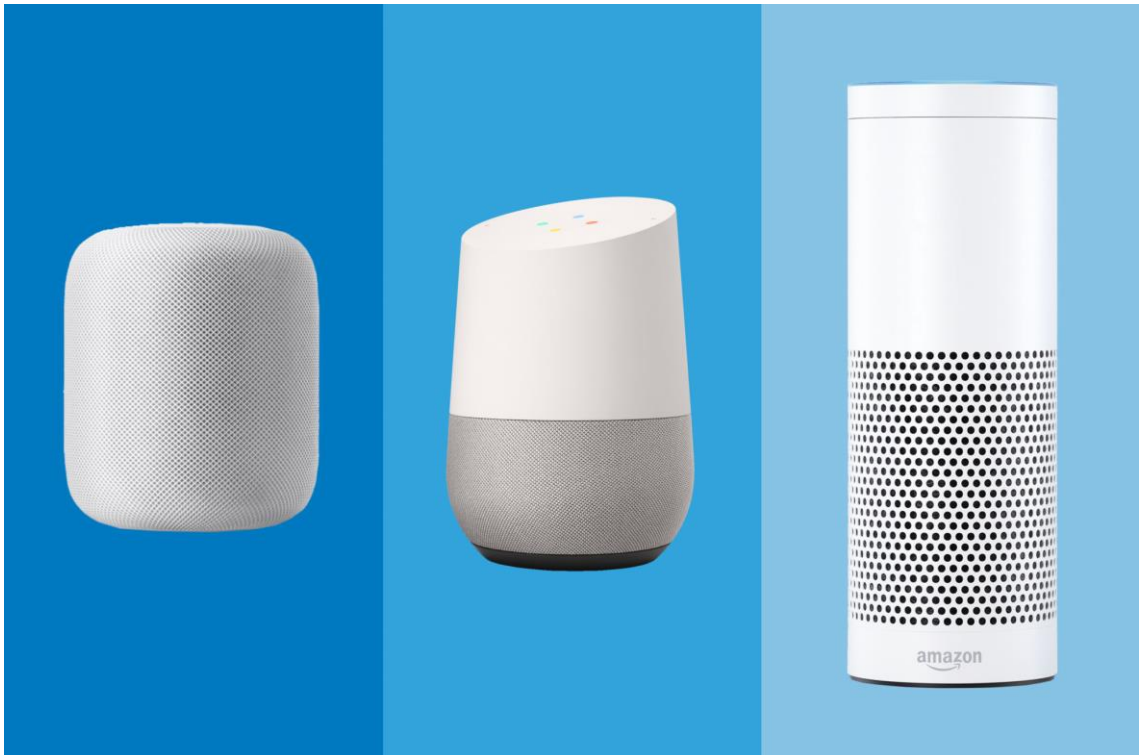
Kuva 7. Foneemien vertailua ja sanan muodostamista Markovin piilomallin avulla.

Kuva 7 havainnollistaa Markovin piilomallin toteutusta, jossa muodostetaan sana *Potato*. Ensimmäinen foneemi (*P*) haarautuu kahteen osaan, jossa toisella on pienempi pistemäärä kuin toisella vaihtoehdolla. Suuremman pistemäärän omaava osa valitaan ja sitten jatketaan seuraavaan foneemiin. Tätä tehdään niin kauan, kunnes päästään sanan loppuun. Tämän jälkeen kerrotaan tulos käyttäjälle (Grabianowski 2006).

2.3 Puheentunnistuksen yleisyys laitteissa ja sovelluksissa

Puheentunnistuksen suosio on kasvanut huomattavasti vuosien varrella teknologian kehittyessä. Kehityksen ansiosta puheentunnistusta käytetään monipuolisesti eri aloilla kuten lääketieteessä, markkinoinnissa ja taloudessa [van der Velde 2018]. Puheentunnistuksen suosituin käyttösovellus on virtuaalinen avustaja, joka asennetaan tiettyyn laitteeseen ja toimii nimensä mukaisesti käyttäjän avustajana. Virtuaalinen avustaja aktivoidaan ennalta määritetyn komennon avulla, jonka jälkeen käyttäjä voi pyytää avustajaa suorittamaan yksinkertaisia toimintoja, kuten viestien kirjoittamista tai yhteystietojen lisäämistä. Virtuaalisia avustajia löytyy erilaisista kulutuselektronikoista kuten älypuhelimista, älykelloista ja älykaiuttimista. Monet suuret teknologiayritykset ovat lähteneet kehittämään omia avustajiaan, joista tunnetuimmat ovat esimerkiksi teknologiayritysten Applen Siri ja Googlen Google Now.

Virtuaalisten avustajien määrä kulutuselektronikoissa on kasvanut huomattavasti käsitteen ”Esineiden internet” (*Internet of Things*) myötä, mikä tarkoittaa internetin laajentumista erilaisiin laitteisiin ja koneisiin. Kyseisiä tavaroita voidaan ohjata ja monitoroida verkon yli. Esimerkiksi Amazonin älykaiutin Echo, joka sisältää avustajan Alexan, ottaa ensin käskyjä, lähettää ne Amazonin omille palvelimille käsiteltäviksi ja lähettää sitten vastauksen takaisin käyttäjälle. Laitteilta voidaan kysyä ja pyytää samoja asioita kuin puhelimista löytyviltä virtuaaliavustajilta. Kuvassa 8 näkyvät markkinoiden suosituimmat älykaiuttimet, joista Amazon Echo on kolmesta laitteesta myydyin (Perez 2018). Älykaiuttimien suurimmat tuottajat ovat raportoineet puheentunnistuksen virhetason olevan nykyään alle 5 %, mikä on ollut hyväksyttävällä tasolla monille ihmisille. Vuoteen 2020 arvioidaan monien tavallisten kotilaitteiden tai -tavaroiden kuten jääkaappien, palovaroittimien ja peilien liittyvän esineiden internetiin, mikä tarkoittaa puheentunnistuksen suosion nousua (Boyd 2018).



Kuva 8. Vuonna 2018 älykaiuttimien suosio räjähti nousuun. Kuvassa markkinoiden suosituimmat laitteet vasemmalta oikealle: Apple Homepod, Google Home ja Amazon Echo.

Puheentunnistusta hyödynnetään myös jokapäiväisessä elämässä esimerkiksi eri asiakaspalvelutilanteissa tai älypuhelimien sovelluksissa. Pankit ovat jo pitkään käyttäneet puheentunnistusta 24 tunnin puhelupalveluissa, joihin asiakkaat voivat soittaa ja kertoa asiansa. Suosituimmissa sovelluksissa kuten musiikkipalvelussa

Spotifyssä voidaan esimerkiksi pyytää sovellusta soittamaan tietty kappale tai kysyä suositusta. Tämä on yksi esimerkki siitä, miten sovelluksessa voidaan puheentunnistusta hyödyntää. Tapa tulee yleistymään, sillä puheentunnistuksen käyttäminen on melko vaivatonta, ja se voidaan aktivoida yksinkertaiselta äänikomennolla. Tulevaisuudessa saattaa olla mahdollista jopa automatisoida erilaiset asiakaspalvelutilanteet voidaan jopa automatisoida yhdistämällä puheentunnistus ja tekoäly, jolloin ihmisiä ei tarvita lainkaan.

2.4 Tilannekatsaus virtuaalitodellisuuteen

Vaikka virtuaalitodellisuus on ollut olemassa jo 50-luvulta asti, se ei kuitenkaan saanut suurta suosiota kuluttajien keskuudessa kömpelön laitteiston takia. Muutos tapahtui vasta 2010-luvulla. Kun Facebook osti Oculus-yrityksen vuonna 2014, tämä osoitti sen, että suuret yritykset alkoivat kiinnostua virtuaalitodellisuudesta [Brown 2017]. Pian suuret teknologiayritykset kuten HTC, Sony ja Google liittyivät mukaan kehityskilpailuun ja samalla julkaisivat omat laitteensa markkinoille. Virtuaalitodellisuuden kohdeyleisö oli aluksi melko rajoittunutta, koska laitteet olivat liian kalliita monille kuluttajille, ja sen lisäksi ne tarvitsevat tehokkaan koneen tai konsolin käyttääkseen laitetta. Hintojen takia kuluttajina olivat ensin pelaajat ja teknologiaintoilijat, jotka olivat valmiita maksamaan uutuudesta. Tähän ongelmaan Samsung julkaisi oman laitteensa Gear VR:n, joka erosi muista kilpailijoistaan niin, että se on langaton ja toimii puhelimen kanssa. Laitteen hinta ei myös ollut päätä huimaava, jolloin lopuilla kuluttajillakin oli mahdollisuus kokeilla uutta teknologiaa. Monet yritykset alkoivat seurata Samsungin jalanjalkia julkaisemalla halpoja virtuaalilaseja markkinoille. Googlen yksinkertaiset, mutta käytännölliset pahvista muodostettavat lasit olivat suurmenestys sen edullisen hinnan ja helppokäyttöisyyden takia.

Ensimmäisen VR-laitteen julkaisun jälkeen vuodesta 2016 myynti on ollut joka vuosi nousussa. VR-laitteiden myynti ylitti miljoonarajan vuonna 2017: myyntiluvut koostuivat suurimmaksi osaksi kalliista VR-laitteista kuten Sonyn Playstation VR:stä ja HTC Vivestä, mutta kännyköitä käyttävät VR-lasit myivät näiden ohella myös melko hyvin [Lamkin 2017]. 2018 ei kuitenkaan ollut virtuaalitodellisuuden suurin menestys, vaikka nousua arviottiin jopa vuoteen 2022 asti. Kuten kuvassa 9 näkyy, datan mukaan esimerkiksi HTC Viven myynti on hidastunut merkittävästi ja HTC on jopa irtisanoneet 1500 työntekijää (Fruhlinger 2018). Osasyyn myynnin hidastumiseen ja laskuun on se,

että laitteiden kehittäminen on melko pitkä prosessi eikä uutta voida heti julkaista markkinoille. Monet laitteet ovat melko isoja, langallisia ja kömpelöitä, jolloin nämä ominaisuudet ovat myös vaikuttaneet myyntilukuihin. Langattomia laitteita on kuitenkin julkaistu hiljattain keväällä 2018, kuten Oculus Go ja HTC Vive Focus, mutta näiden myynti on myös tasaantunut alkuhuuman jälkeen. Ostamisen suhteen kuluttajat yleensä omistavat vain yhden VR-laitteen, koska markkinoiden suosituimmat ovat kalliita, ja halvimmat ovat grafiikoiltaan melko tyydyttävät esimerkiksi pelaamista varten.



Kuva 9. HTC Viven myyntimäärä on laskenut huomattavasti vuoden 2018 kesän aikana.

Vaikka virtuaalitodellisuuden kasvu on hieman hidastunut, monet alan asiantuntijat uskovat virtuaalitodellisuuden olevan standardi myöhemmin tulevaisuudessa eri aloilla. Muutos tulee tapahtumaan melko hitaasti, sillä sen hintava valmistus on vieläkin este monille kuluttajille. Tulevaisuudessa saatetaan nähdä esimerkiksi maantiedon tunnilla opiskelijat VR-laseineen seikkailemassa ympäri maailmaa tai kuntoutuksessa potilas katselemassa kauniita ja rauhoittavia maisemia.

3 Sovelluksen suunnittelu virtuaalilaseille

Seuraava osio käy läpi eri asioita, joita kannattaa ottaa huomioon, kun suunnitellaan sovellusta virtuaalitodellisuudelle.

3.1 Sovelluksen vaatimukset

Virtuaalitodellisuudelle suunniteltaessa kannattaa muistaa pitää vaatimuslista lyhyenä ja yksinkertaisena. Tietokoneelle tai mobiilille suunniteltu sovellus ei välttämättä toimi oikein virtuaalitodellisuudessa, jos sovellusta yritettäisiin suoraan siirtää virtuaalitodellisuuteen ilman minkäänlaisia muutoksia.

Sovelluksessa käyttäjä pystyisi käyttämään puheentunnistusta luomaan ja muokkaamaan erilaisia 3D-objekteja virtuaalimaailmassa. Kun käyttäjä avaa sovelluksen, käyttäjän eteen tuodaan virtuaalimaailma, jota voi katsella ympäriinsä. Käyttäjä aktivoi puheentunnistuksen sanomalla tietyn sanan, jonka jälkeen voidaan komento lausua. Jos puheentunnistus tunnistaa komennon, niin suoritetaan vaadittu komento, ja jos ei, niin ilmaistaan käyttäjälle jonkinlaisena virheilmoituksena esimerkiksi äänimerkkinä tai tekstinä, jos käyttäjä ei laita ääniä päälle.

Virtuaalimaailmassa käyttäjällä olisi mahdollisuus luoda 3D-kappaleita, kuten kuutioita, palloja ja lieriöitä. Kappaleiden kokoja on mahdollista muokata isommaksi tai pienemmäksi ja värejä voi myös muuttaa. Niitä pystyy siirtämään lähemmäs, kauemmas, vasemmalle ja oikealle. Objekteja on mahdollista myös poistaa. Ruudun keskiosaan käyttöliittymään lisätään osoitin, jonka avulla voidaan valita tietty objekti. Osoittimen avulla käyttäjä pystyisi osoittamaan tiettyyn kappaleeseen, johon haluaisi komennon kohdistuvan. Suunnitelmana on pitää sovellus melko yksinkertaisena, koska virtuaalitodellisuudelle tehdessä sovellukseen ei kannata laittaa paljon asioita tai ominaisuuksia.

Sovelluksessa käytettävien komentojen tulee olla yksinkertaisia ja ytimekkäitä. Liian monimutkainen ja pitkä komento altistaa helpommin virheiden syntymiseen. Komentojen tulee sisältää vähintään verbi (mitä halutaan tehdä), väri (minkä värinen luotava kappale on) ja kappaleen muoto (minkä muotoinen kappale on).

Jos sovellusta haluaa kehittää Android- tai iOS-laitteelle, Googlen VR SDK:n lataaminen on välttämätöntä. Googlen SDK sisältää kaikki tarpeelliset työkalut VR-kokemuksen luomiseen. Muille VR-laitteille ei tarvitse ladata kyseistä SDK:ta.

Laitteistoksi kannattaa valita sellainen kännykkä, joka tukee virtuaalitodellisuutta. Jos VR:stä haluaa saada kaiken mahdollisen hivin irti, niin puhelimesta pitäisi olla ainakin jonkinlainen liikesensori kuten gyroskooppi VR-maailman katselua varten. Jos sitä ei ole, niin vähintään kompassi, jota voidaan hyödyntää ulkoisten sovellusten avulla ja toimimaan melko samalla tavalla kuin gyroskooppi. Ilman liikesensoria VR-sovelluksia ei voisi katsella ympäriinsä, jolloin minkäänlaista immersiota ei synny.

Kuvassa 10 näkyy Unityn omat minimivaatimukset Googlen SDK:ta käyttäville laitteistoille. Android-laitteiksi suositellaan sellaisia, joissa on järjestelmänä vähintään Android 5.0 ja iOS-laitteille minimissään iOS 8. Minimilaitteissa saattaa ongelmana kuitenkin olla grafiikka, sillä VR-sovellusten kuvataajuudeksi suositellaan jopa yli 90 FPS. Tutkimusten mukaan alle 90 FPS:llä pyörivät sovellukset luovat todennäköisemmin käyttäjilleen jonkinlaisia oireita kuten lievää pahoinvointia [IrisVR]. Pahoinvoinnista ja muista mahdollisista oireista kerrotaan lisää osiossa 3.3. Insinööritöön projektissa tavoitellaan parhaita mahdollista kuvataajuutta, joten 50-60 FPS olisi ideaalisin ainakin kännyköille. Tavoitteen pitäisi toteutua moitteettomasti, sillä projektin ei pitäisi vaatia paljoa tehoja vähäisen sisällön takia.

Device	Hardware	Software
Cardboard	<ul style="list-style-type: none"> - Devices running Android 4.1 or later with a gyroscope. - Devices running iOS 8 or later with a gyroscope. - A viewing device compatible with Cardboard. 	<ul style="list-style-type: none"> - Unity Cardboard integration requires Android Lollipop or greater. - iOS devices with the Google Cardboard application installed.

Kuva 10. Unityn ja Googlen minimivaatimukset puhelimille.

Resoluutiolla ei ole paljoa väliä tämän sovelluksen käyttämisen kannalta. Nykyisten VR-laitteiden resoluutiot ovat välillä 960x1080 – 1440x1600 (per silmä). Jos resoluutio on liian matala, se saattaa synnyttää ns. ”Screen-door”-efektin (SDE) kuten kuvassa 11 näkyy. Efektillä tarkoitetaan sitä, kun ruudusta voidaan erottaa pikselit selkeästi [Lagage 2016]. Kyseiseen efektiin välillä törmää myös tämän hetkisten markkinoiden VR-laitteisiin ja myös insinööriyön projektissa saattaa kyseinen efekti hieman näkyä.



Kuva 11. Screen-door-efektissä näkyvät pikselit selkeästi.

Jotta komentoja voitaisiin tulkita reaaliajassa mahdollisimman nopeasti, latenssin eli datan lähettämisen keskimääräinen aika palvelimelle ja takaisin pitäisi olla mahdollisimman matala. Peleissä ihanteellisin arvo olisi alle 20 ms (0.02s), mutta puheentunnistuksen kanssa hyväksyttävä arvo on noin 300 ms (0.3s). [Presser 2018.] Projektissa ei kuitenkaan keskitytä paljoa latenssiin, vaan tärkeämpää on se, että palvelu toimii ja suorittaa hommansa.

3.2 Ohjelmointirajapinnan valitseminen puheentunnistukselle

Puheentunnistusta varten on olemassa monenlaisia ohjelmointirajapintoja ja muita mahdollisuuksia tarjolla Unity-pelimoottorille. Seuraavana käydään läpi erilaisia vaihtoehtoja, joita harkittiin opinnäytetyöprojektiin.

Rajapinta tarjoaa työkalut ohjelmistojen rakentamiseen. Se yleensä sisältää listan erilaisista operaatioista kuten funktioista, joita kehittäjä voi käyttää vapaasti. Rajapintojen avulla kehittäjän ei tarvitse tehdä kaikkia toiminnallisuuksia tyhjästä, jolloin säästyy myös paljon aikaa ja rahaa. Jos kehittäjän pitää esimerkiksi ohjelmoida popup-ikkuna

sovellukseen tai kirjata käyttäjä johonkin sivustoon sisälle, niin näihin on jo rajapinnat olemassa, sillä rajapintoja on moneen lähtöön, kuten kuvassa 12 näkyy.



Kuva 12. Rajapintoja on monenlaisia, joista kehittäjä voi hyötyä ajallisesti sekä rahallisesti.

Rajapintoja on erilaisia, joista tunnetuin on REST API. Kyseiseen rajapintaan lähetetään pyyntöjä, jotka riippuvat siitä, miten tietoa haluaa käsitellä. Jos palvelimelta halutaan saada tietoa, voidaan sinne lähettää GET-pyyntö, tai jos tietoa halutaan lisätä, niin lähetetään POST-pyyntö. Joissakin tapauksissa rajapintoja voidaan ladata yhtenä pakettina eli SDK:na (Software Development Kit), jonka avulla voidaan hyödyntää rajapinnan tarjoamia operaatioita. Tämä tapahtuisi käytännössä niin, että ohjelmoidessa kutsutaan esimerkiksi kirjasto "Random", joka sisältäisi operaatioita liittyen satunnaisuuteen.

IBM Watson

IBM Watson tarjoaa monia erilaisia tuotteita käyttäen tekoälyä apunaan, kuten kielten kääntämistä ja visuaalista tunnistusta (esimerkiksi objektit ja sen ominaisuuksien tunnistaminen). Palvelulla on kolme erilaista maksusuunnitelmaa: Lite, Standard ja Premium. Ensimmäinen suunnitelma eli Lite on maksuton siihen asti, kunnes ylitetään palveluiden käyttöraja, joka on 10 000 API-kutsua kuukaudessa. Tili poistetaan automaattisesti, jos sitä ei käytä kuukauteen. Kaksi viimeistä suunnitelmaa eli Standard ja Premium ovat maksullisia ja sisältävät loputtoman määrän API-kutsuja sekä kaikki data tallennetaan. Watson voidaan helposti yhdistää Unityyn, sillä Unitylle on tarjolla Watson-paketti, jonka voi ladata suoraan pelimoottoriin ja on heti käytettävissä. Palvelu

sisältää Speech-to-Text- ja Watson Assistant -ominaisuudet, jotka ovat hyödyllisiä insinööriyöhön liittyvän projektin tekemiseen. Seuraavana on linkki Watsonin kotisivuille:

<https://www.ibm.com/watson/>

Google Cloud

Google Cloud on pilvipalvelu, jonka avulla käyttäjä voi esimerkiksi säilöä dataa suorittaa analytiikkaa tai hyödyntää koneoppia. Palvelua on mahdollista käyttää ilmaiseksi rajojen puitteissa ja alkuun saa myös 300 dollaria ilmaista krediittiä vuodeksi. Jos käyttö ylittää palvelujen rajat, niin hinnat muodostuvat käytön mukaan riippuen palveluista. Cloud on mahdollista yhdistää Unityyn, sillä se tarjoaa myös Watsonin tavoin paketin, jonka voi ladata pelimoottorille. Google Cloud itsessään sisältää Speech-to-Text (Speech Recognition) -ominaisuuden ja monia erilaisia tekoälypalveluita, mutta vain osa palveluista on Unityn käytössä vähäisten liitännäisten takia. Unity ja Google Cloud keskittyvät tällä hetkellä enemmän datan analysointiin ja backend-puolen toteuttamiseen. Seuraavana on linkki Googlen omille sivuille:

<https://cloud.google.com/>

Microsoft API & Microsoft Azure

Microsoftilla on oma puheentunnistus API, joka on jo integroituna Unityyn. Puheentunnistuksen voi helposti lisätä omaan projektiin muutamalla koodinpätkällä. Kyseinen API vaatii Windows-pohjaisen laitteen, joten puheentunnistus ei toimi esimerkiksi Android- tai iOS-laitteilla. Microsoftilla on myös oma pilvipalvelu Azure, joka sisältää puheentunnistuksen ja luonnollisen kielen tunnistuksen. Jälkimmäinen palvelu muistuttaa IBM:n Watson-palvelua, sillä molemmissa voidaan määrittää tunnistettavat sanat ja määritellä niille toiminnot. Azuren palveluhinnat määrittyvät käytön mukaan eli koko palvelu on ilmainen tiettyihin rajoihin asti. Azurella ei ole virallista integrointimahdollisuutta Unityn kanssa, mutta on olemassa epävirallisia liitännäisiä, joiden avulla voidaan Azuren palveluja hyödyntää Unityssä. Alla ovat palveluiden kotisivuille, joista ensimmäinen linkki on Microsoftin omaan API:in ja toinen Azureen:

<https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input-in-unity>

<https://azure.microsoft.com/en-us/>

Android

Androidilla on puheentunnistuksen API, joka sisältää muun muassa tunnistuksen ja kielituen. API:n hyödyntäminen ei maksa mitään ja sitä voidaan käyttää rajattomasti. Vaikka Unityllä voidaan kehittää sovelluksia Androidille, natiivikirjastoja ei voida kuitenkaan suoraan käyttää Unityn sisällä. Jos API:a haluaa käyttää, niin pitää joko ostaa lisäosa tai käyttää epävirallista liitännäistä, joka esimerkiksi löytyy sivuilta https://github.com/PingAK9/SpeechAndText_Unity_iOS_Android. Seuraavana löytyy linkki Androidin puheentunnistuksen dokumentaatioon:

<https://developer.android.com/reference/android/speech/package-summary>

Lopullinen valinta

Kaikista ylläolevista palveluista ja ohjelmointirajapinnoista insinööriyön projektiin valittiin IBM Watson. Watson tarjoaa puheentunnistusta ja Speech-to-Text-palveluja, joita voidaan vaivattomasti implementoida Unity-projektiin. Palvelun ilmaisversio sopii hyvin tähän käyttöön, koska kutsuja ei tarvitse tehdä kovin paljoa. Puheentunnistus myös toimii kaikilla laitteilla esimerkiksi PC:llä tai kännyköillä, joten laitevalikoima ja niiden tuki on laajempi kuin muilla vaihtoehdoilla. Watsonin dokumentaatio on selkeä ja hyvien ohjeiden avulla on helppo päästä alkuun. Taulukko 1 sisältää vertailun mainittujen palvelujen ominaisuuksien kesken.

Taulukko 1. Palvelujen välinen vertailu

Rajapinta	Speech to Text	Komentojen määrittely (Intentiot & entiteetit)	Unity-tuki	Hinta	Soveltuvat alustat
IBM Watson	x	x (Assistant)	x	Ilmainen tiettyyn rajaan asti. Lite plan (Assistant: 10k API kutsua, Speech to Text: 100min/kk)	Kaikki (PC, Mac, Linux, Android, iOS)
Google Cloud	x	x (DialogFlow)	x (Eri liitännäisiä (plugin) eri palveluille)	Ilmainen tiettyyn rajaan asti. Free Tier (DialogFlow: 1000 kutsua/kk, Speech to Text: 60min/kk)	Kaikki (PC, Mac, Linux, Android, iOS)
Microsoft API & Microsoft Azure	x	x (Azure; Language Understanding, LUIS)	Azurelle ei virallista tukea. Epävirallisia liitännäisiä löytyy Internetistä. Microsoft API valmiina Unityssä.	Azure ilmainen tiettyyn rajaan asti. Free plan (LUIS: 5 transactions per second (TPS, transaktio per sekunti), Speech to Text: 5h/kk). Microsoft API aina ilmainen.	Azure: kaikki (PC, Mac, Linux, Android, iOS). Microsoft API: Windows-laitteet.
Android	x	Ei ole.	x	Aina ilmainen	Android

3.3 Virtuaalitodellisuuden yleisimpien ongelmien minimointi

Virtuaalitodellisuus toi paljon uusia mahdollisuuksia teknologian maailmaan, mutta toi samalla monia uusia ongelmia ja mahdollisia haittoja, joiden vuoksi kaikki eivät pysty laitteita käyttämään. Tällä hetkellä virtuaalitodellisuuden yleisimmät ongelmat liittyvät ohjattavuuteen ja terveyteen (Nielsen 2017). Yksi tunnetuimmista haitoista on virtuaalitodellisuuden aiheuttama pahoinvointi, joka syntyy hitaan ruudunpäivitysnopeuden takia. Matala kuvataajuus ja sen hitaus aiheuttavat *sensory conflict* -tilan käyttäjälle, mikä tarkoittaa sitä, että näköhavainto ja fyysinen havainto aiheuttavat toisilleen konfliktin. Tällainen voisi tapahtua esimerkiksi käyttäjän käyttäessä laitetta istuen samalla, kun liikkuu virtuaalimaailmassa (Caddy 2016).

Pahoinvoinnin estämiseksi on keksitty erilaisia ratkaisuja. Yksi mahdollinen keino olisi kuvataajuuden (*FPS, Frames Per Second*) pitäminen korkeana [Crytek 2016]. Kun kuva päivittyy tarpeeksi nopeasti käyttäjälle, virtuaalitodellisuuden immersio on syvempi ja käyttäjän liikkua virtuaalitodellisuudessa ei ole hidastuksia, jotka saattaisivat ”erottaa” käyttäjän virtuaalikokemuksesta [Patrão ym. 2015]. Opinnäytetyötä varten tehtävästä mobiilisovelluksesta pitää tehdä mahdollisimman kevyt ja nopea, jotta sovellus ei kuormittaisi laitteistoa liikaa eikä sisältäisi minkäänlaisia viiveitä. Tämä saavutetaan sillä, että sovellus optimoidaan eri keinoin kuten poistamalla ylimääräiset objektit ja alentamalla sovelluksen grafiikkatasoa. (Porter 2017.)

Oculus on listannut kaikista mahdollisista terveyteen liittyvistä oireista, jotka saattavat syntyä pelaamisen aikana. Näistä vakavimpia ovat esimerkiksi uupumus ja kohtaukset, mutta yleisimpiä ovat kuitenkin silmien väsyminen, huimaus ja päänsärky. Laittevalmistajat ja pelinkehittäjät usein suosittelevat käyttäjiä pitämään taukoja puolen tunnin välein, jotta pelaamisesta tai käyttämisestä ei tulisi minkäänlaisia oireita. Pitkällä aikavälillä syntyvät haitat eivät ole tiedossa, mutta vaikutuksia tutkitaan jatkuvasti.

Jotta VR:ää voisi käyttää, ensin pitäisi hankkia sitä tukeva laite. VR-laitteiden hinnat ovat tähän mennessä olleet aika korkeita, jolloin yleensä vain teknologiainfoilijat ovat olleet ensimmäisiä ostajia ja kokeilijoita. Koneisiin ja konsoleihin yhdistettävien VR-laitteiden ohelle on markkinoille myös tullut halvempia vaihtoehtoja, kuten Googlen pahviset Cardboard-lasit, joihin voidaan asettaa kännykkä lasien näytöksi. Nykyään kännyköiden teho on sellaista luokkaa, että useimmat niistä jaksavat pyörittää tehoja syöviä sovelluksia, jotka usein keskittyvät tarkkaan grafiikkaan. Kännyköitä ei luultavammin

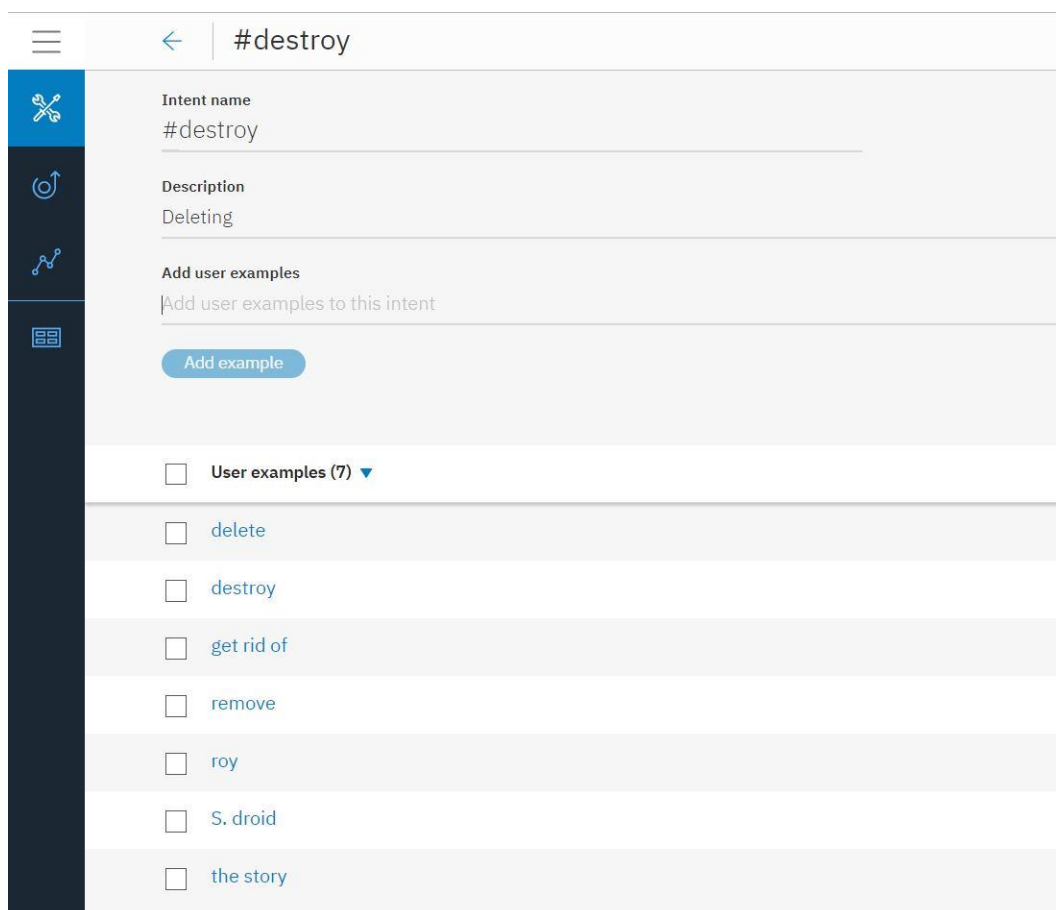
tarvita myöhemmin tulevaisuudessa, sillä Oculus julkaisi vuonna 2018 ensimmäiset langattomat virtuaalilasit Oculus Go:n, joita ei tarvitse yhdistää mihinkään laitteeseen eikä kännykkää tarvita. Markkinoiden halvimpien laitteiden hinnat liikkuvat 5-100 euron välillä.

4 Puheentunnistusovelluksen toteutus

4.1 Komentojen luominen ja sen toiminta

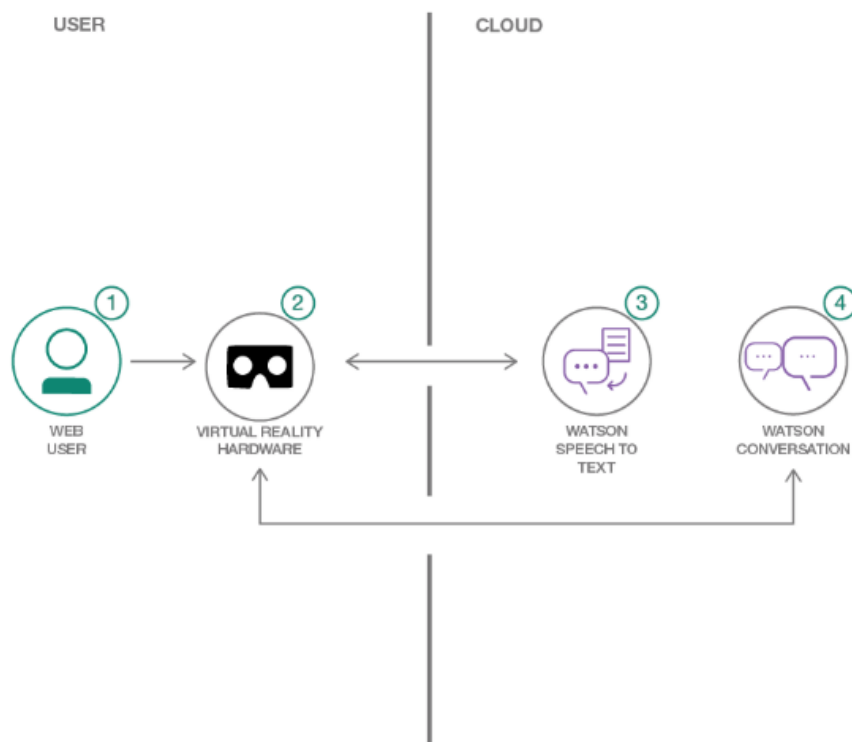
Puheentunnistusovellusta luodessa käytetään Watsonin tarjoamaa Assistant- ja Speech-To-Text-palvelua. Assistenttiin voidaan lisätä omia komentoja ja käyttää niitä missä tahansa sovelluksessa ja Speech-To-Text hoitaa käyttäjän komentojen muuttamisen tekstiksi.

Assistant-palvelu toimii niin, että tekijä luo listan komennoista eli intentioista/aikomuksista (*Intents*) ja sanoista (*Entities*), joita haluaa mukaan sovellukseen. Komentojen kannattaa olla yksinkertaisia ja ytimekkäitä, jotta niiden käyttö olisi mahdollisimman sujuvaa ja helppo ymmärtää. Listaa luodessa sille annetaan ensin pääsana (esimerkiksi "Destroy"), jota kutsumalla aktivoidaan komento. Kyseiseen listaan lisätään synonyymeja, kuten "Remove" tai "Delete" monipuolisuuden vuoksi. Synonyymeja lisättäessä on suositeltavaa, että laitetaan mukaan myös ns. "virheellisiä" komentoja, joita kone saattaisi kuulla väärin. Kone saattaa ymmärtää esimerkiksi sanan "Destroy" toisenlaisena sanana "The Story", koska molemmat saatetaan sanoa melko samalla tavalla. Jos komentoja halutaan tunnistaa myös "virheellisistä" komennoista, voidaan yllä olevat sanat lisätä. Kuvassa 13 on näkymä Watson Assistant -työtilasta, jonne on omat komennot luotu ja lisätty.



Kuva 13. IBM Watsonin tarjoamaan Assistant-palveluun voidaan lisätä listoja komennoista. Kuvassa näkyy esimerkki luokan sisällöstä, jonne voi lisätä synonyymeja ja mahdollisia "väärinkultuja" sanoja.

Kun käyttäjä sanoo komennon mikrofonin, komento muutetaan tekstiksi Speech-to-Text-palvelun avulla. Palvelu palauttaa tekstin sovellukselle, joka antaa sen eteenpäin Assistant-palvelulle. Sanottu komento "puretaan" sanoiksi ja niitä vertaillaan jo ennalta määrättyihin listoihin. Esimerkiksi lause "Create a big red ball" puretaan osiin, jolloin tekstistä tulee "Create", "big", "red" ja "ball". Näitä sanoja vertaillaan listoihin, joita käyttäjä on luonut Assistant-palvelun sisällä. Purettu lause palautetaan takaisin sovellukselle, jossa tietyille sanoille ohjelmoitu toiminnallisuus suoritetaan. Kuvassa 14 näytetään Watsonin ja sovelluksen toimintaprosessi.



Kuva 14. Miten paikallinen sovellus ja IBM Watson kommunikoivat keskenään.

Sovellukseen halutaan soveltaa neljä eri komentoa: objektin luominen (Create), muokkaaminen (Edit), poistaminen (Destroy) ja avun pyytäminen (Help). Kyseiset komennot lisätään Assistant-palvelun "Intents"-luokkaan. Lisättyjä pääkomentoja voidaan kutsua myös synonyymeillä, jotka pitää lisätä listoihin jo etukäteen. "Intents"-luokan lisäksi on myös "Entities"-luokka, joka edustaa objektityyppiä [Defining entities, 2018]. Sovellusta varten on luotu neljä aliluokkaa, jotka ovat objektin siirtäminen (direction), objektin värin vaihto (material), objektin muoto (object) ja objektin koon muuttaminen (scale). Näihin neljään luokkiin on listattu erilaisia sanoja, joilla käyttäjä saattaisi objekteja kuvailla tai liikutella. Esimerkiksi "Direction"-luokkaan on lisätty monia suuntaan liittyviä sanoja: "farther", "up" ja "closer". Lisäksi jokaiselle sanoille voidaan lisätä synonyymejä, kuten "farther"-sanalle voidaan lisätä synonyymit "further" ja "far".

Jos komentoja halutaan lisätä, niin ensin pitää päättää aikomus eli intentio. Kyseinen sana on se komento, jota kutsutaan sovelluksen sisällä. Luonti tehdään Watson Assistant-sivulla, jossa valitaan työtila, mihin uusia komentoja halutaan luoda. Intentioiden lisäksi samalla sivulla voidaan määrittää myös entiteettejä. Entiteetiksi kannattaa valita sellaisia asioita, joita komennosta halutaan tunnistaa. Jos olisi esimerkiksi olemassa sellainen sovellus, jossa voidaan luoda eläimiä, niin esimerkkikomento voisi olla "I want to have a dog". Tässä komennossa "want" olisi

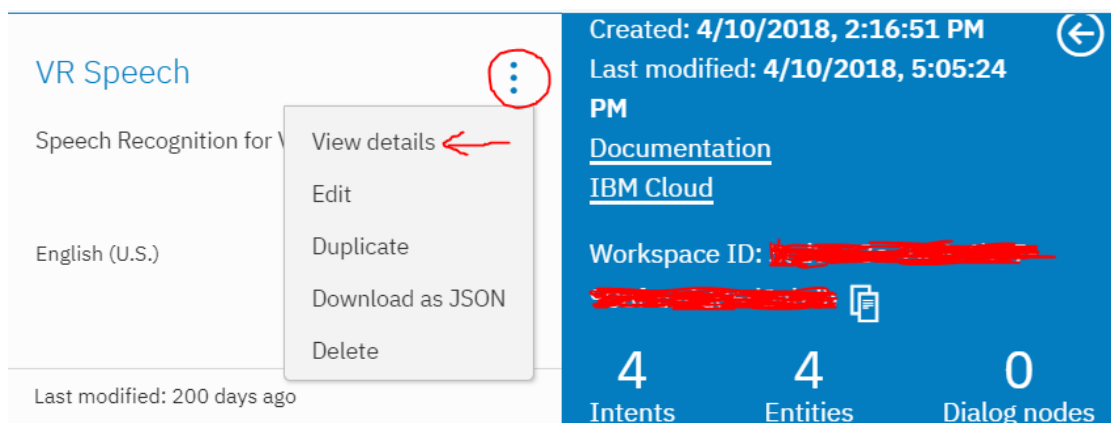
intentio ja entiteetti "animal", koska "dog"-sanan ylempi luokka on eläin. Intentioiden ja entiteettien lisäksi näille on kannattavaa lisätä synonyymisanoja, jotta komennot olisivat mahdollisimman monipuolisia.

Komennoiksi voidaan oikeastaan valita mikä tahansa sana, koska kaikkea on mahdollista ohjelmoida sovellukseen.

4.2 Puheentunnistuksen soveltaminen Unityyn

Insinöörintyön projekti käyttää pohjanaan IBM:n luomaa esimerkkiprojektia, joka hyödyntää Watson Assistant ja Speech-to-Text-palveluita. Esimerkkiprojektin on tehnyt IBM:n Developer Advocate Niklas Heidloff ja projektin voi ladata hänen github-sivuilta: <https://github.com/nheidloff/unity-watson-vr-sample>. Esimerkkiprojekti toimii niin, että käyttäjä voi kävellä, pysähtyä ja tiedustella Saksan eri kaupunkien säätiedotusta puheentunnistuksen avulla. Omaa sovellusta varten kyseisiä toimintoja ei tarvita, mutta ne voidaan jättää, jos halutaan.

Esimerkkiprojektiin pitää lisätä omat IBM-tunnukset eli käyttäjätunnus ja salasana. Sen lisäksi projektissa halutaan käyttää omia komentoja, joten tarvitaan myös oman työtilan id. Työtila on siis se, mihin omat komennot ja sanat on luotu. Kuvassa 15 näkyy paikka, mistä työtilan id löytyy Watson Assistant -sivuilta.



Kuva 15. Työtilan ID löytyy painamalla ikonia kyseisen työtilan oikeasta yläkulmasta, josta avautuu valikko. Sieltä painamalla "View details" -nappulaa löytyy työtilan ID.

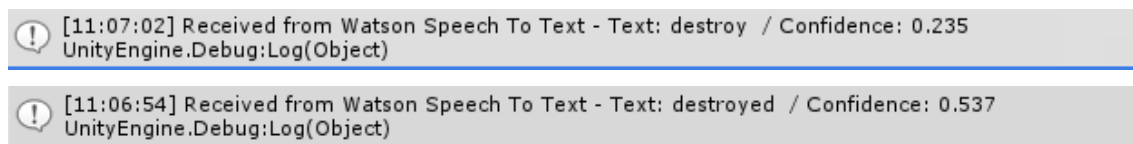
Jotta luotuja komentoja voitaisiin käyttää sovelluksessa, ne pitää implementoida Unityn sisällä. Käyttäjän komentoa on käsiteltävä ja siitä erotella sanat, jotta tunnistettaisiin

entiteetit ja intentiot. Kun sanat on tunnistettu, niille pitää vielä erikseen luoda halutut toiminnot, esimerkiksi punaisen pallon ilmestyminen maailmaan, jos sanottu komento sisälsi "create"-, "red"- ja "ball"-sanat.

Watsonilla on valmis Unity-paketti, jonka voi viedä suoraan Unityyn. Paketti sisältää kaiken tarpeellisen Assistantin ja Speech-to-Text -palveluiden käyttämistä varten. Projektiin luodaan skripti, joka hoitaa komentojen tunnistamisen ja sanojen erottelun. Sovellus toimii tällä tavalla:

1. Watsonia varten pitää ensin antaa oma käyttäjätunnus ja salasana.
2. Mikrofoni laitetaan päälle ja vastaanotetaan käyttäjän puhe.
3. Kun puhe on loppunut, mikrofoni suljetaan. Välitetään puhe Watsonin Speech-to-Text -palvelulle.
4. Speech-to-Text -palvelu palauttaa sovellukselle komennon tekstinä. Kyseinen lause käsitellään lähettämällä Watsonin Assistantille, jossa erotellaan intentiot ja entiteetit.
5. Assistant palauttaa tunnistetut sanat sovellukselle, jossa sanoja vastaavat toiminnot suoritetaan.

Kun tunnistus on tehty, voidaan sanoille luoda toiminnallisuutta. Tunnistuksen yhteydessä Watson palauttaa myös "Confidence Score" eli luottamuskertoimen, kuten kuvassa 16 näytetään. Se kertoo sen, kuinka varma tai luottavainen Watson on tuloksestaan (Murdock 2016).



Kuva 16. Komentojen kuultua Watson palauttaa käyttäjälle luottamuskertoimen, joka näkyy kuvasta. Mitä korkeampi kerroin, sitä varmempi Watson on omasta tuloksestaan.

Komentojen toiminta voidaan luoda uuteen tai olemassa olevaan skriptiin. Sitä ennen intentiot ja entiteetit pitää erotella sanotusta komennosta. Tämä on jo valmiiksi implementoitu IBM:n projektissa, jolloin sinne pitää vain lisätä oma logiikka. Esimerkkikoodi 1 havainnollistaa sanojen erottelun ja niille implementoidut toiminnot. Jos sanottu komento oli "Create", niin etsitään pelaaja ja haetaan siltä senhetkinen sijainti. Tämän jälkeen luodaan pelaajan eteen jokin kappale listasta "_shapes". Luonnin jälkeen avataan taas mikrofoni uusia komentoja varten funktiolla "RecordAgain()". Sama tehdään myös "Destroy"-komennolle eli etsitään pelaaja ja haetaan siltä sijainti. Tämän jälkeen luodaan "Raycast", joka on ns. näkymätön viiva esimerkiksi törmäyksiin tarkoitettu fysiikan operaatio. Tuhoaminen tehdään vain silloin, jos käyttäjä katsoo suoraan kappaleeseen eli ehto toteutuu, kun "Raycast" osuu käyttäjän edessä olevaan kappaleeseen. Tällöin voidaan suorittaa tuhoaminen ja sen jälkeen taas kuunnellaan käyttäjää.

```

***Conversation.cs
string intent = messageResponse.intents[0].intent;
string entity = messageResponse.entities[0].entity;
if (intent.Contains("create"))
{
    Debug.Log("Create invoked.");

    GameObject player = GameObject.Find("Player");
    Instantiate(_shapes[index], player.transform.position +
        _camera.transform.forward * 2, Quaternion.identity);

    RecordAgain();
}
else if (intent.Contains("destroy"))
{
    Debug.Log("Destroy invoked.");

    GameObject player = GameObject.Find("Player");

    RaycastHit hit;
    if (Physics.Raycast(player.transform.position,
        _camera.transform.forward, out hit, 10))
    {
        Destroy(hit.transform.gameObject);
    }

    RecordAgain();
}

```

Esimerkkikoodi 1. Intentioiden ja entiteettien tunnistus sekä niiden toiminnallisuus. "MessageResponse" sisältää sarjallistettua dataa, joka on saatu Watsonilta.

Intentioiden ja entiteettien tunnistaminen pitää tehdä käsin, sillä muuten sovelluksessa ei tapahdu mitään.

4.3 Sovelluksen testaus virtuaalilaseilla

Projektin kokoaminen riippuu siitä, millä alustalla haluaa sovellusta testata. Sovelluksen testaaminen tietokoneella on melko yksinkertaista, sillä Unityssä oletusalustana on PC, Mac & Linux. Puheentunnistusta varten koneen oma tai jokin muu erillinen mikrofoni toimii hyvin tähän tarkoitukseen. Jos sovellusta haluaa kokeilla puhelimella (Android & iOS), niin alustaa on muutettava haluamaan puhelinalustaan. Ennen sovelluksen siirtämistä puhelimeen pitää täyttää vaaditut tiedot (esim. paketin nimi, API Level) Unityssä navigoimalla Edit -> Project -> Player Settings. Näkymän pitäisi olla sama kuin kuvassa 17. Samalla sivulla pitää myös sallia VR:n käyttö etsimällä "XR Settings" -otsikko ja sieltä ruksia "Virtual Reality Supported" päälle. Tämän jälkeen sen alle tulee laatikko, jonne voidaan lisätä tuetut laitteet. Insinööriyön projektia varten lisätään ainakin "Cardboard" painamalla +-painiketta.



Kuva 17. Paketin nimeä on pitää muuttaa ja API Levelin minimivaatimus on 19 (Android 4.4) VR-sovelluksille. Kyseinen kuva on esimerkki Android-laitteelle siirrettävästä sovelluksesta.

Koko prosessi iOS-laitteelle on melko samankaltainen, mutta tietojen muuttaminen kuten kuvassa 17 on näytetty, tehdään Apple-laitteiden omassa IDE:ssä, Xcode:ssa. Tiedoissa tarvitsee muuttaa vain paketin nimi ("Package Name").

Sovelluksessa vaaditaan ympärille katsomista, jolloin puhelimella testaaminen on hieman helpompaa kuin tietokoneella. Puhelimella testattaessa sitä voi liikutella ympäriinsä, toisin kuin tietokoneella, jossa on pakko siirtää kameraa käsin. Puhelin voidaan laittaa VR-laseihin kiinni, jos VR-lasit ovat saatavilla. Immersio on tällöin syvempi ja käytettävyys mukavampaa sekä luonnollisempaa.



Kuva 18. Puhelimen näkymä sovelluksesta. Näkymässä näkyvät mikrofonin tila ja saatavilla olevat äänityslaitteet.

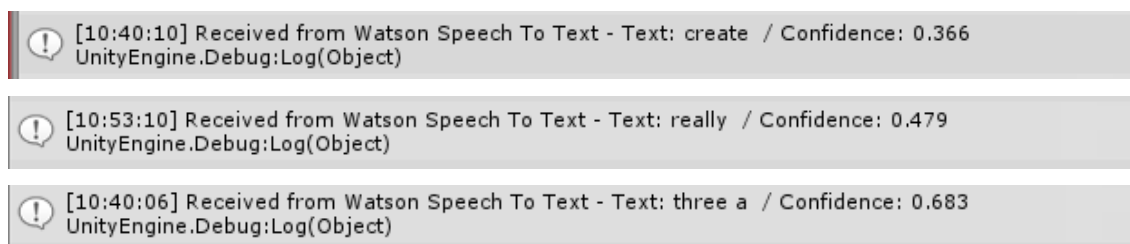
Kuvassa 18 näkyy sovelluksen viimeisin versio VR-näkymässä, jossa lasit ovat välttämättömät sen käyttämistä varten.

5 Tulokset ja sovelluksen mahdollisuudet

5.1 Sovelluksen analysointi

Sovellukseen ei ehditty tehdä kaikkia haluttuja toimintoja, mutta päätoiminnot kuitenkin saatiin mukaan. Tällä hetkellä sovelluksessa voidaan ainoastaan luoda kuutioita ja tuhota luodut kappaleet. Käyttäjä voi sanoa komennot luonnollisella kielellä, kunhan se sisältää vaaditut pääsanat kuten "create" tai "cube". Käskyjä ei siis tarvitse sanoa muodossa "Create cube", sillä esimerkiksi "Can you create a cube for me" on ymmärrettävää Watsonille. Myös pääsanojen synonyymeja voidaan käyttää, kuten "Give me a cube" tai "Make me a cube".

Sovelluksen puheentunnistuksessa on pieni viive, jota ei voida välttää, sillä datan analysointi vie luonnollisesti aikaa. Viive on suunnilleen kahden sekunnin pituinen, ja se on samanpituinen siitä huolimatta, vaikka komentona olisi yksi sana tai pitkä lause. Koodin optimointi saattaisi auttaa viiveen lyhentämiseen, mutta tätä ei ole testattu.



Kuva 19. Komennon tulkinta riippuu aika paljon siitä, miten käyttäjä sanoo komennon ja myös laitteistolla on vaikutusta. Watson tulkitse "create"-sanon mielenkiintoisin tuloksin.

Watson osasi kääntää useimmat komennot oikein, mutta se pariin otteeseen tulkitse esimerkiksi sanan "create" joksikin muuksi. Kuvassa 19 näkyy Watsonin eri tulkinnot kyseisestä sanasta. Puheentunnistus on helposti altis virheellisille tulkinnoille, joten tätä ei voi oikeastaan mitenkään välttää. Sanottaessa komento kannattaa muistaa sanoa se selkeällä puheella ja kovaan ääneen, sillä mikrofonin laadulla on myös vaikutusta lopulliseen tulokseen. Sovellusta testattaessa piti sanoa komento hieman kovalla äänellä, koska hiljaisella äänellä sanottu komento ei välttämättä käynnistänyt Watsonin tunnistuspalvelua.

Sovellusta ei pysty tällä hetkellä käyttämään puhelimessa, vaan se toimii ainoastaan koneella. Sovellukseen on laitettu kehittämistä varten näkyville saatavilla olevat mikrofonit tekstiojektiin, joka näytti puhelimen kohdalla "Android Audio Input". Se ei kuitenkaan ottanut minkäänlaista äänikomentoa vastaan, koska sovelluksessa ei tapahtunut mitään. Tämä toimii kuitenkin PC:llä. Kyseistä ongelmaa ei ehditty ratkaista ajanpuutteen takia.

5.2 Sovelluksen hyödyt ja käyttömahdollisuudet tulevaisuudessa

Puheentunnistuksen avulla laitteiden käyttö helpottuu huomattavasti, sillä pienet, fyysiset liikkeet kuten nappien painaminen vähentyy eikä käyttäjän tarvitse toistaa liikkeitä kovin usein. Tilanne on usein myös sellainen, että ohjaimissa ei ole tarpeeksi painikkeita. Tämä mahdollisuus tarjoaa uuden tavan käyttää sovelluksia, koska ohjaimia ei välttämättä enää tarvita eikä käsiä tarvitse käyttää. Puheentunnistusta voidaan myös viedä pitkälle eteenpäin kehittämällä keskustelevan tekoälyn, jonka kanssa käyttäjä voisi olla vuorovaikutuksessa virtuaalimaailmassa. Tällainen kokemus syventäisi immersiota virtuaalimaailmaan, koska nykyisten laitteiden avulla kokemus keskeytyy esimerkiksi

ulkoisten häiriötekijöiden takia. Insinööriyön sovellusta voidaan laittaa VR-laseihin ja hallita sitä pelkästään puheen ja liikkeen avulla.

IBM Watsonin palveluita voidaan hyödyntää laajemmin insinööriyön sovelluksessa. Watson tarjoaa esimerkiksi datan analysointipalveluja, jolloin koneoppimista tai syväoppimistä on mahdollista implementoida. Kielten kääntäminen on myös toteutettavissa, jolloin esimerkiksi jonkinlainen botti voisi vaikka puhua monella eri kielellä tai sovellus kääntää monikieliseksi. Mahdollisuuksia on siis monia, joten erilaisia tekoälypalveluita kuten Watsonia kannattaisi hyödyntää.

Sovellusta voidaan mahdollisesti pelillistää, sillä puheentunnistusta käyttäviä VR-pelejä on olemassa. Kyseiset pelit eivät ole olleet suurimpia menestyksiä, koska puheentunnistuksessa on omat hankaluutensa. Riippuen siitä minkälainen tempo pelissä on, nopeatempoisissa peleissä ei puheentunnistus sovi toisin kuin hieman hitaimmissa, joihin puheentunnistus saattaisi sopia. Puheentunnistus sopisi esimerkiksi opetuskäyttöön soveltuviin peleihin mainiosti, koska se tarjoaisi oppilaille interaktiivisen tavan oppia uusia asioita.

Sovelluksen käyttö ei rajoitu vain mobiililaitteisiin, sillä sitä voidaan hyödyntää myös muissa VR-laitteissa. Parilla muutoksella sovellukseen voidaan sovellusta helposti levittää muihin alustoihin.

Sovellukseen ei saatu kaikkia ominaisuuksia tehtyä ajanpuutteen takia. Tästä huolimatta tämä kuitenkin tarjoaa aloituspohjan samankaltaisille projekteille. Sovellukseen voisi kehittää esimerkiksi implementoimalla jonkinlaisen botin, jonka kanssa käyttäjä voisi keskustella erilaisten tekoälypalvelujen avulla.

Ulkoisia palveluita kannattaa hyödyntää, sillä ne tarjoavat paljon mahdollisuuksia monipuolisten sovelluksien tekemiseen. Tämän insinööriyön myötä löytyi paljon erilaisia pilvi- ja tekoälypalveluita, joita käyttämällä varmasti hyötyy ajallisesti sekä rahallisesti. Nykyään on yleistä se, että sovelluksia tehdessä ei aloiteta enää tyhjästä, vaan hyödynnetään kyseisiä palveluita ja erilaisia rajapintoja, jotka tarjoavat hyviä pohjia ohjelmistojen aloittamiseen.

6 Yhteenveto

Yhdessä puheentunnistus ja virtuaalitodellisuus ovat melko tuntemattomia toisilleen, sillä molemmat ovat kehittyneet kunnolla vasta tämän vuosikymmenen aikana. Kehitys on ollut vahvasti nousussa nykyaikaisen teknologian takia ja kuluttajien kiinnostus uuteen teknologiaan on samalla hyvin levittänyt kyseisten teknologioiden näkyvyyttä.

Tämän insinööriyön tarkoituksena oli tutkia uusia ja erilaisia käyttömahdollisuuksia kyseisistä teknologioista. Molemmista aiheista ei valitettavasti löytynyt paljoa tietoa, sillä tällaista komboa ei kovin moni ole vielä yhdistänyt. Koko insinööriyöprosessi oli opettavainen, sillä kaikki tieto minkä löysi, oli uutta. Uuden tiedon soveltaminen sovellukseen oli yksi tavoitteista, joten tekeminen oli kiehtova prosessi.

Ajanpuutteen takia sovellukseen ei ehditty implementoida kaikkia suunniteltuja ominaisuuksia. Puheentunnistuksen ja virtuaalitodellisuuden taustatyö pintaa syvemmälle vei eniten aikaa, koska aiheista oli vain pintapuolisesti. Alun perin suunnitelmana oli lähteä tekemään sovellusta kokonaan tyhjästä, mutta juuri ajanpuutteen takia päätettiin ottaa IBM:n esimerkkiprojektin tämän sovelluksen pohjaksi. Esimerkkiprojektin ymmärtämiseen meni toiseksi eniten aikaa, koska itse Watson ja sen tarjoama rajapinta olivat uusia asioita, joten tutkimiseen meni luonnollisesti aikaa. Vähiten aikaa kului oikeastaan sovelluksen tekemiseen ja ohjelmoimiseen, koska Unity-pelimoottorista on jo aiempaa käyttökokemusta.

Tuloksena saatiin kuitenkin toimiva sovellus käyttäen ulkoista tekoälypalvelua. Sovelluksessa on paljon parannettavaa, sillä esimerkiksi yksi tärkeimpiä ominaisuuksia oli saada se toimimaan puhelimella, mutta tämä tyssäsi mikrofoni-ongelmaan ja ajanpuutteeseen. Myös osa komennoista jäi tekemättä, kuten kappaleiden muokkaaminen ja liikuttaminen. Puuttuvat ominaisuudet voidaan kuitenkin tulevaisuudessa lisätä.

Lähteet

Best, Jo. 2013. IBM Watson: The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next. Verkkoaineisto.

<<https://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>>. 2013. Luettu 17.04.2018.

Boyd, Clark. 2018. The Past, Present and Future of Speech Recognition Technology.

Verkkoaineisto. <https://medium.com/swlh/the-past-present-and-future-of-speech-recognition-technology-cf13c179aaf>. 10.01.2018. Luettu 10.06.2018.

Brown, Lisa. 2017. A Brief History of Virtual Reality. Verkkoaineisto. <[https://fil-](https://fil-mora.wondershare.com/virtual-reality/history-of-vr.html)

[mora.wondershare.com/virtual-reality/history-of-vr.html](https://fil-mora.wondershare.com/virtual-reality/history-of-vr.html)>. 24.10.2017. Luettu 04.05.2018.

Caddy, Becca. 2016. Vomit Reality: Why VR makes some of us feel sick and hot to

make it stop. Verkkoaineisto. <<https://www.wearable.com/vr/vr-headset-motion-sickness-solution-777>>. 19.10.2016. Luettu 20.04.2018.

Crytek. 2016. On VR Development and Avoiding Motion Sickness. Verkkoaineisto.

<<http://www.crytek.com/blog/on-vr-development-and-avoiding-motion-sickness>>. 04.10.2016. Luettu 26.04.2018.

Fruhlinger, Joshua. 2018. Consumer interest in VR is declining according to sales data

trends. Verkkoaineisto. <https://media.thinknum.com/articles/sales-data-shows-that-consumer-interest-in-vr-is-waning/>. 07/2018. Luettu 22.07.2018.

Grabianowski, Ed. 2006. How speech recognition works. Verkkoaineisto. <[https://elec-](https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm)

[tronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm](https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm)>. 10.10.2006. Luettu 4.2.2018.

Defining entities. 2018. Verkkoaineisto.

<<https://console.bluemix.net/docs/services/conversation/entities.html#defining-entities>>. 13.03.2018. Luettu 19.04.2018.

Dormehl, Luke. 2017. 8 virtual reality milestones that took it from sci-fi to your living room. Verkkoaineisto. <<https://www.digitaltrends.com/cool-tech/history-of-virtual-reality/>>. 13.10.2017. Luettu 02.05.2018.

Dybsky, Denis. 2017. The History of Virtual Reality: Ultimate Guide. Part 1. Verkkoaineisto. <<https://teslasuit.io/blog/history-of-virtual-reality-ultimate-guide>>. 01.03.2017. Luettu 13.05.2018.

History of Virtual Reality. Verkkoaineisto. Virtual Reality Society. <<https://www.vrs.org.uk/virtual-reality/history.html>>. Luettu 20.05.2018.

IrisVR. The Importance of Frame Rates. Verkkoaineisto. <<https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates>>. Luettu 21.10.2018.

Korpinen, Pertti. 2005. Ääni syntyy. Verkkoaineisto. <http://www.aanipaa.tamk.fi/synty_1.htm>. 2005. Luettu 4.2.2018.

Krueger, Myron W.; Gionfriddo, Thomas; Hinrichsen, Katrin. 1985. Tutkielma. University of Connecticut. http://dada.compart-bremen.de/docUploads/chi85_krueger.pdf. Luettu 27.05.2018.

Kumparak, Greg. 2014. A Brief History of Oculus. Verkkoaineisto. <<https://techcrunch.com/2014/03/26/a-brief-history-of-oculus/>>. 26.03.2014. Luettu 02.05.2018.

Lagace, Marc. 2016. What is the “screen-door effect” and why does It happen? Verkkoaineisto. <<https://www.vrheads.com/what-screen-door-effect-and-why-does-it-happen>>. 20.07.2016. Luettu 23.10.2018.

Lamkin, Paul. 2017. Virtual Reality Headset Sales Hit 1 Million. Verkkoaineisto. <https://www.forbes.com/sites/paullamkin/2017/11/30/virtual-reality-headset-sales-hit-1-million/#22d116fc2b61>. 30.11.2017. Luettu 22.07.2018.

Microsoft. 2018. Microsoft Azure. Verkkoaineisto. <<https://azure.microsoft.com/en-us/>>. 2018. Luettu 27.10.2018.

Microsoft. 2018. Voice input in Unity. Verkkoaineisto. <<https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input-in-unity>>. 2018. Luettu 27.10.2018.

Murdock, Bill. 2016. How to select a threshold for acting using confidence scores. Verkkoaineisto. <<https://developer.ibm.com/watson/blog/2016/06/23/how-to-select-a-threshold-for-acting-using-confidence-scores/>>. 23.06.2016. Luettu 19.10.2018.

Nield, David. 2017. 5 problems virtual reality needs to solve to go mainstream. Verkkoaineisto. <<https://www.t3.com/news/5-problems-virtual-reality-needs-to-solve-to-go-mainstream>>. 12.04.2017. Luettu 17.04.2018.

Patrão, Bruno; Pedro, Samuel & Menezes, Paulo. 2015. Tutkielma. University of Coimbra. <<http://scitecin.isr.uc.pt/Proceedings/Papers/EPCGI/17.pdf>>.

Perez, Sarah. 2018. Google Home Mini was the best-selling smart speaker in Q2. Verkkoaineisto. <<https://techcrunch.com/2018/09/19/google-home-mini-was-the-best-selling-smart-speaker-in-q2/?guccounter=1>>. 08/2018. Luettu 21.10.2018.

Porter, Tim. 2017. VR Optimization Tips fom Underminer Studios. Verkkoaineisto. <<https://software.intel.com/en-us/articles/vr-optimization-tips-from-underminer-studios>>. 04.04.2017. Luettu 29.04.2018.

Presser, Rachel. 2018. The Importance of Latency in Online Gaming. Verkkoaineisto. <<https://www.bandwidthplace.com/the-importance-of-latency-in-online-gaming/>>. 14.02.2018. Luettu 28.10.2018.

Rubin, Peter. 2017. A Conversation with Jaron Lanier, VR Juggernaut. Verkkoaineisto. <<https://www.wired.com/story/jaron-lanier-vr-interview/>>. 21.11.2017. Luettu 27.05.2018.

Savolainen, Erkki. 2001. Mitä fonologia on? Verkkoaineisto. <<https://fi.finnlectura.fi/verkkosuomi/Fonologia/sivu112.htm>>. 2001. Luettu 4.2.2018.

Savolainen, Erkki. 2001. Klusiilit p, t, k, d, (b, g). Verkkoaineisto. <<https://fi.finnlectura.fi/verkkosuomi/Fonologia/sivu151.htm>>. 2001. Luettu 4.2.2018.

Turi, Jon. 2014. The sights and scents of the Sensorama Simulator. Verkkoaineisto. <<https://www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/>>. 16.02.2014. Luettu 21.05.2018.

van der Velde, Naomi. 2018. Innovative uses of speech recognition today. Verkkoaineisto. <https://www.globalme.net/blog/new-technology-in-speech-recognition#Voice_Technology_in_Finance>. Päivitetty 07.03.2018. Luettu 01.05.2018.