

Panu Weckman

Alustariippumaton puhetta tukeva ja korvaava kommunikointisovellus

Opinnäytetyö

Syksy 2020

SeAMK Tekniikka

Tietotekniikan koulutusohjelma

SeAMK 

SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Ohjelmistotekniikka

Tekijä: Panu Weckman

Työn nimi: Alustariippumaton puhetta tukeva ja korvaava kommunikointisovellus

Ohjaaja: Tapio Hellman

Vuosi: 2020 Sivumäärä: 36 Liitteiden lukumäärä: 0

Tässä työssä kehitettiin alustariippumaton kommunikointisovellus kommunikoinnin tueksi. Työn kehitysympäristöksi valittiin Unity-pelimoottori, jolla voi tehdä luonnostaan alustariippumattomia sovelluksia ilman ylimääräistä käännöstyötä. Työssä käydään läpi myös käyttöliittymien sekä puhetta tukevien ja korvaavien sovelluksien teoriaa.

Lopputuloksena saatiin uusi kommunikointisovellus, jonka kehitys jatkuu vielä tämän työn jälkeen. Jatkokehityksen tavoitteena on saada sovelluksesta kaupallisesti menestyvä tuote.

Avainsanat: AAC, C#, Unity, TTS, teksti puheeksi, puhetta tukeva ja korvaava kommunikointi, kommunikointisovellus

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Software Engineering

Author: Panu Weckman

Title of thesis: Augmentative and Assistive Cross Platform Communication Application

Supervisor: Tapio Hellman

Year: 2020 Number of pages: 36

In the thesis, a new cross platform communication programme was developed to support communication with speech-impaired people. The development environment was a Unity game engine, since it is a cross platform environment by nature. The work also studied the theory of software development concerning interfaces, and augmentative and alternative communication programmes.

As the result of the thesis there is a new cross platform communication programme, and its development will be continued after this thesis project. The objective of the further development is to make the programme a commercially successful product.

Keywords: AAC, C#, Unity, TTS, Text-To-Speech, augmentative and alternative communication, communication program

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	3
Kuva-, kuvio- ja taulukkoluettelo.....	5
Käytetyt termit ja lyhenteet.....	7
1 JOHDANTO.....	9
1.1 Työn tausta.....	9
1.2 Työn tavoite.....	9
1.3 Työn rakenne.....	10
2 PUHETTA TUKEVA JA KORVAAVA KOMMUNIKOINTI.....	11
2.1 Yleistä.....	11
2.2 Tietokoneavusteinen kommunikointi.....	11
2.3 Käyttäjälähtöinen suunnittelu.....	12
2.4 Saavutettavuus.....	13
3 TEKNIIKAT, KEHYKSET JA TYÖKALUT.....	14
3.1 Scrum.....	14
3.2 Unity.....	14
3.3 RT-Voice PRO.....	15
3.4 File Transfer Server.....	15
4 BLABBER.....	16
4.1 Kehitystyö.....	16
4.1.1 Sisällön luominen.....	23
4.1.2 TTS.....	27
4.1.3 Tallentaminen.....	27
4.1.4 Värivalikko.....	28
4.1.5 Siirto toiselle laitteelle.....	29
4.1.6 Kuvahaku.....	30
5 TULOKSET JA YHTEENVETO.....	32
LÄHTEET.....	34

Kuva-, kuvio- ja taulukkoluetelo

Kuva 2. Xamarin-prototyyppi Android-emulaattorissa	17
Kuva 3. Kommunikointisovellus, prototyyppi nro 1. Käyttäjätila.	18
Kuva 4. Kommunikointisovellus, Prototyyppi nro 1. Muokkaustila ja avoin sivupaneeli.	18
Kuva 5. Blabber, prototyyppi nro.. 3.....	19
Kuva 6. Blabber, prototyyppi nro 3, avoin sivupaneeli.	19
Kuva 7. Blabber, prototyyppi nro 3. Asetusvalikko.	20
Kuva 8. Blabber, versio nro 0.6. Käyttäjätila.	21
Kuva 9. Blabber, versio nro 0.6, asetusvalikko.	21
Kuva 10. Blabber esillä Sillalla-seminaarissa 17.3.2020	22
Kuva 11. Blabber. versio nro 0.9, beta.....	23
Kuva 12. Solun prefab ja sen komponentit	24
Kuva 13. Solun olioluokka.....	24
Kuva 14. Vasemman paneelin hierarkia Unity-editorissa.....	25
Kuva 15. Keskipaneelin hierarkia Unity-editorissa	25
Kuva 16. Satunnaistunnisteen luovat metodit.	26
Kuva 17. Solun koon muuttava metodi	27
Kuva 18. Värivalikko	28
Kuva 19. ColorPicker-luokka kursorin sijainnin laskentaan ja värin valintaan väriympyrästä.....	29
Kuva 20. Haku Papunetin kuvapankista	30

Kuva 21. Kuvahaku Windows-laitteen tiedostojärjestelmästä. 31

Kuva 22. Kuvahaku Android-laitteen tiedostojärjestelmästä. 31

Käytetyt termit ja lyhenteet

AAC	Augmentative and alternative communication, eli puhetta tukeva ja korvaava kommunikointi.
Alusta	Eri laitteet sekä näiden käyttöjärjestelmät, esim. tietokoneet ja älypuhelimet/tabletit ja näiden omat käyttöjärjestelmät.
Apuvälinekeskus	Sairaanhoitopiirien yksikkö, joka hoitaa keskitetysti piirin alueen apuvälinehankinnat, lainaamiset, opastukset sekä näiden huolto- ja korjauspalvelut
Bliss-symboli	Bliss-symbolit muodostuvat geometrisistä peruskuvioista, symbolin merkitys riippuu sen muodosta, koosta sekä sijainnista. Bliss symboleita käytetään Bliss-kielessä, joka on kansainvälinen käsitekirjoitusjärjestelmä.
C#	Ohjelmointikieli.
C#-objekti	C#-ohjelmointikielessä käytettävät objektit, esim. string, integer ja boolean, joita käytetään objektin tyypin määrittelyssä.
Framework	Ohjelmistokehys, runko, jonka päälle voidaan rakentaa sovelluksia.
JSON	Javascript Object Notation, kevyt avoimen standardin tiedostomuoto tiedonvälitykseen. Tiedostomuotoa on helppo lukea ja kirjoittaa.
Papunet	Osa kehitysvammaliittoa, Papunet tuottaa materiaalia ja oppaita puhevammaisuudesta, tarjoaa puhevammaisille ja kehitysvammaisille mahdollisuuden yhteyden pitoon, ja on verkkosivujen saavutettavuuden asiantuntija, tarjoten koulutusta ja ohjeita saavutettavien verkkosivujen toteutuksesta.

Prefab	Unityn uudelleen käytettävä elementti, esimerkiksi peliobjekti, jolle on määritetty valmiiksi tietyt asetukset.
SAPI 5	Speech Application Programming Interface, Microsoftin kehittämä puheentunnistus- ja puhesynteesiteknologia.
Tikoteekki	Kehitysvammaliiton alainen tietotekniikka- ja kommunikaatiokeskus, jonka päämääränä on edistää puhevammaisten henkilöiden osallisuutta, itsemääräämisoikeutta ja mahdollisuuksia kommunikoida yhteisössään ja yhteiskunnassa.
Tikoteekkiverkosto	Tikoteekin ylläpitämä verkosto erikoissairaanhoidon yksiköiden asiantuntijoista.
TTS	Text to Speech Synthesis, ohjelma, joka kääntää tekstin ääniaalloiksi, eli teksti puheeksi -syntesointi.
UDP	User Datagram Protocol, yhteydetön kuljetuskerroksen tietoliikenneprotokolla, joka ei vaadi yhteyttä laitteiden välille, mutta mahdollistaa tiedon siirron.
UNITY-Asset	Unity Asset Storesta hankittava elementti, joka voi olla esim. työkalu, ohjelmisto-, grafiikka- tai äänipaketti.
UNITY	Monialustainen pelimoottori.
UNITY-objekti	Unity-pelimoottorin oma objekti, esimerkiksi "GameObject", "Texture" tai "Sprite", joita käytetään samoin, kuin C#-kielen objekteja.

1 JOHDANTO

1.1 Työn tausta

Oikeus kommunikointiin on ihmisoikeus ja ensisijaisesti ihminen käyttää kommunikointikeinonaan puhetta, jota voidaan tehostaa monin ei-kielellisin keinoin. Puhelukyky voi kuitenkin jäädä kehittymättä tai sen voi menettää esimerkiksi etenevän sairauden tai onnettomuuden vuoksi. Tätä varten on kehitetty useita eri ratkaisuja, joilla voidaan helpottaa kommunikoimista. (Huuhtanen 2011, 8.)

Puhetta tukevasta ja korvaavasta kommunikoinnista käytetään kansainvälistä yleisnimitystä AAC (Augmentative and alternative communication) (Huuhtanen 2011, 8).

Kommunikoinnin apuvälineet on suunniteltu avuksi kommunikointivaikeuksista kärsiville ihmisille. Niihin kuuluvat erilaiset kommunikointikansiot, -taulut sekä puhelaitteet ja kommunikointisovellukset. Kommunikointisovellukset ovat kommunikointia tukevia, täydentäviä ja/tai korvaavia ohjelmistoja, joiden avulla henkilö, joka ei muutoin pysty kommunikoimaan selkeästi saa itselleen keinot kommunikointiin. (Tikoteekki 2018; Papunet 2020a.)

Suuri osa kommunikointisovelluksista keskittyy Windows- ja iOS-alustoille Androidin jäädessä vähemmälle. Niillekin olisi Tikoteekkiverkoston asiantuntijoiden mukaan tarvetta, sillä Android-laitteet ovat keskimäärin halvempia ja valikoimaa iskunkestäviin älylaitteisiin löytyy enemmän (Adepoju 2018; Higginbotham, Jacobs 2011, 2; Bullitt Group Research 2017).

1.2 Työn tavoite

Tavoitteena työssä on kehittää alustariippumaton kommunikointisovellus, tuoda hyödyllinen kommunikointiohjelma Android-alustalle ja tämän avulla mahdollisesti tulevaisuudessa alentaa apuvälinekeskuksien kustannuksia laitehankinnoissa.

1.3 Työn rakenne

Johdannon jälkeen käsitellään puhetta tukevaa ja korvaavaa kommunikointia, tietokoneavusteista kommunikointia sekä käyttöliittymiä saavutettavuuden näkökulmasta. Tämän jälkeen käsitellään tutkimuksen projektityöhön käytettyjä tekniikoita, kehyksiä sekä työkaluja. Luvussa 4 käydään läpi ohjelmiston kehitystyö ja lopuksi käsitellään tulokset ja työn yhteenveto.

2 PUHETTA TUKEVA JA KORVAAVA KOMMUNIKOINTI

2.1 Yleistä

Puhetta tukeva ja korvaava kommunikointi on viestintämenetelmä, jota käytetään, kun henkilöllä on vaikeuksia kommunikoida selkeästi normaalein keinoin. (Papunet 2020a).

Syitä kommunikointivaikeuksiin voi olla esimerkiksi synnynnäinen puhevamma tai neurologinen sairaus. Näiden henkilöiden kommunikointimahdollisuuksia voidaan edistää esimerkiksi kuntoutuksella, kommunikoinnin apuvälineillä, tulkkauspalveluilla sekä kanssaihmisten toiminnalla. (Papunet 2020b.)

Puhevammaiset voivat saada Suomen apuvälinekeskuksista kommunikointilaitteiston ja sitä tukemaan kommunikointikansion. Nämä voidaan toki hankkia myös itse, mutta on suositeltavaa, että potilas arvioidaan kunnolla. Apuväline valitaan yhteistyössä puhevammaisen henkilön, lähi-ihmisten sekä kommunikoinnin ja apuvälinealan asiantuntijoiden kanssa. Potilaan arvioinnin jälkeen hänelle sekä hänen lähipiirilleen opetetaan, kuinka kommunikointikansio, laite sekä ohjelma toimivat, tämä tehdään yleensä yhdessä puheterapeutin sekä tietotekniikan asiantuntijan kanssa. (Huuhtanen 2011, 87-89.)

2.2 Tietokoneavusteinen kommunikointi

Tietokoneavusteinen kommunikointi on tietokoneessa käytettävää kommunikointi- ja puhesynteesiohjelman synteesiä. Näistä käytetään myös nimitystä kommunikointiohjelma. (Huuhtanen 2011, 87.)

Kommunikointiohjelmien avulla saadaan puhekyvyttömälle henkilölle rakennettua yksilöllinen kommunikointisanasto. Tämä voidaan tehdä kuvilla, bliss-symboleilla tai lauselistoina ja yleensä viestit kuullaan puhesynteesin puhumana. (Papunet 2019a.)

Kommunikointiohjelmien kirjo on laaja ja monipuolinen, esimerkiksi kuvasymboleilla toimiva ohjelmisto sopii paremmin vaikeavammaiselle henkilölle, kuin pelkkään tekstiin pohjautuva viestintä. (Papunet 2019a.)

Suomessa on käytössä AAC-sovelluksina muun muassa Speaking Dynamically Pro, Go Talk Now, Communicator 5 sekä Grid 3 (Papunet 2019b).

2.3 Käyttäjälähtöinen suunnittelu

Kehitettäessä ohjelmia henkilöille, joiden kognitiiviset ja/tai motoriset kyvyt ovat heikompia kuin muilla, on kannattavaa suunnitella ohjelmisto käyttäjiä ajatellen. Human-centred design (käyttäjälähtöinen suunnittelu) on lähestymistapa sovelluskehityksessä, jonka tarkoituksena on kehittää hyödyllisiä ja käytettäviä ohjelmia keskittymällä käyttäjiin ja heidän tarpeisiinsa (Waller & Balandin & O'Mara & Judson 2005.)

Sovelluksen kehityksen jokaisessa vaiheessa on hyvä ottaa huomioon käyttäjät ja kaikki, joita sovellus tulee koskettamaan (SFS-EN ISO 9241-210:2019 2019, 6, 13.)

Käytettävyyssominaisuuksia mietittäessä on myös valittava kohderyhmä. Tätä varten on arvioitava keitä käyttäjät ovat, voidaanko heitä kouluttaa etukäteen, millaisessa ympäristössä sovellusta käytetään, mikä on sovelluksen tarkoitus ja mikä on käyttäjille tärkeintä. Kohderyhmän henkilöillä on paremmin tietoa siitä, minkälaisia ominaisuuksia tai minkälainen käyttöliittymä sovelluksessa toimisi parhaiten ja heitä täytyisikin haastatella kehityksen jokaisessa vaiheessa (Niemelä, 2020.)

Käyttäjälähtöisessä suunnittelussa on myös omat heikkoutensa. Yksi niistä on sen hitaus, sillä tutkimustyö ja käyttäjäkyselyt hidastavat itse kehitystyötä merkittävästi (Bowles 2013.)

2.4 Saavutettavuus

Saavutettavuudella tarkoitetaan sellaista tuotetta, jota kaikki voi käyttää vammoista tai toimintarajoitteista huolimatta ilman muiden ihmisten apua. Suunniteltaessa saavutettavaa tuotetta täytyy ottaa huomioon myös esim. sokeat tai neliraajahalvaantuneet, joita varten voi olla erilaisia avustavia laitteita, kuten katseohjaimet tai ruudunlukuohjelmat. (Papunet 2019c.)

3 TEKNIIKAT, KEHYKSET JA TYÖKALUT

3.1 Scrum

Scrum on runko projektinhallinnalle, jonka perusideana on olla ketterä ja nopea tapa kehittää sovelluksia, joiden eri vaiheissa voi tulla yllättäviä muutoksia päivittäin. Scrumin perustana on empiirinen prosessinhallintateoria. Tämän mukaan tieto tulee kokemuksesta ja päätökset tehdään sen mukaan, mitä tiedetään. Scrum on iteratiivinen ja lisäävä lähestymistapa, jolla optimoidaan ennalta-arvattavuutta ja hallitaan riskejä. Scrumin kolme empiirisen prosessinhallinnan pilaria ovat läpinäkyvyys, tarkastelu sekä mukautuminen. (Schwaber & Sutherland 2017, 3, 4.)

Scrum-projektinhallinnassa projekti jaetaan useisiin sprintteihin, jotka ovat enintään kuukauden pituisia jaksoja, joille on asetettu tietyt tehtävät. Sprintille asetetaan tavoite, jota ei sprintin aikana muuteta ja jokaista sprinttiä käsitelläänkin omana ”mini-projektina”. Sprintti voidaan peruuttaa, ennen kuin sprintille määritelty aikaraja on ummessa. Syynä voi olla esimerkiksi, jos sprintin tavoitteet eivät vastaa enää nykytilannetta. (Schwaber & Sutherland 2017, 9, 10.)

3.2 Unity

Unity on pelimoottori ja kehitysympäristö, jolla voidaan kehittää pelejä ja ohjelmistoja useille eri alustoille. Unity huolehtii kääntämisen eri alustoille lähes automaattisesti, jolloin kehittäjän ei tarvitse juurikaan miettiä, mille alustalle sovellusta, milloinkin kehitetään. Tämä mahdollistaa pelien ja ohjelmien kääntämisen lähes kaikille alustoille saumattomasti. (Patil & Alvares 2015.)

Unityn vahvuudet ovat sen monipuolisuudessa ja helppoudessa. Lisäksi Unityn yhteisö on laaja, ja sieltä saa nopeasti apua projektiin. Unity myös tukee useimpia tiedostomuotoja, mikä mahdollistaa mm. 3D-elementtien suunnittelun lähes kaikilla 3D-mallinnusohjelmilla. Unity onkin parhaimmillaan suunniteltaessa 2D- ja mobiilisovelluksia. (Thinkwik 2018.)

Myös Unityn "Asset Store" mahdollistaa nopeamman kehityksen. Valmiiden työkalujen ja Unity-Asettien hankkiminen ja liittäminen projekteihin helpottaa ja nopeuttaa sovellusten kehitystä (Tanant 2018).

Tämä ei kuitenkaan tarkoita, että Unity olisi kaikkivoipa. Vaikka siinä on paljon hyviä ominaisuuksia, on myös huonoja ominaisuuksia. Suuri määrä ominaisuuksia tekee ohjelmasta samalla raskaan. Tämä näkyy myös sillä Unitylla kehitettyjen sovellusten koossa. Monipuolisuuden vuoksi moottori joutuu tekemään myös kompromisseja grafiikan ja muistin käytön suhteen. Loppujen lopuksi Unityn hyvät puolet ovat merkittävästi paremmat kuin sen mukanaan tuomat huonot puolet ja tästä syystä se on mahdollisesti maailman suosituin pelimoottori. (Kumar 2019.)

3.3 RT-Voice PRO

RT-Voice Pro on Crosstales, inc. yrityksen valmistama Unity-työkalu, jolla on helppo muuntaa teksti puheeksi saumattomasti laitteesta riippumatta. Työkalu antaa laajat mahdollisuudet äänen muokkaukselle sekä synteessin valinnalle. Se tukee useita eri synteesejä mahdollistaen alustariippumattoman puhesynteesisivalinnan, tämän lisäksi työkalu on SAPI-5 yhteensopiva. (Crosstales 2020.)

3.4 File Transfer Server

File Transfer Server on Unity-työkalu, joka on riippumattoman kehittäjän kehittämä paketti, joka hallitsee tiedonsiirtoa UDP-protokollan avulla, mahdollistaen tiedostojen jakamisen laitteelta toiselle, kunhan molemmat laitteet ovat samassa paikallisverkossa (esimerkiksi. sama WLAN-verkko). Työkalu mahdollistaa myös laitteen tiedostojärjestelmän selauksen, joka palauttaa käyttäjälle tiedoston fyysisen osoitteen. (Monsuarez 2019.)

4 BLABBER

4.1 Kehitystyö

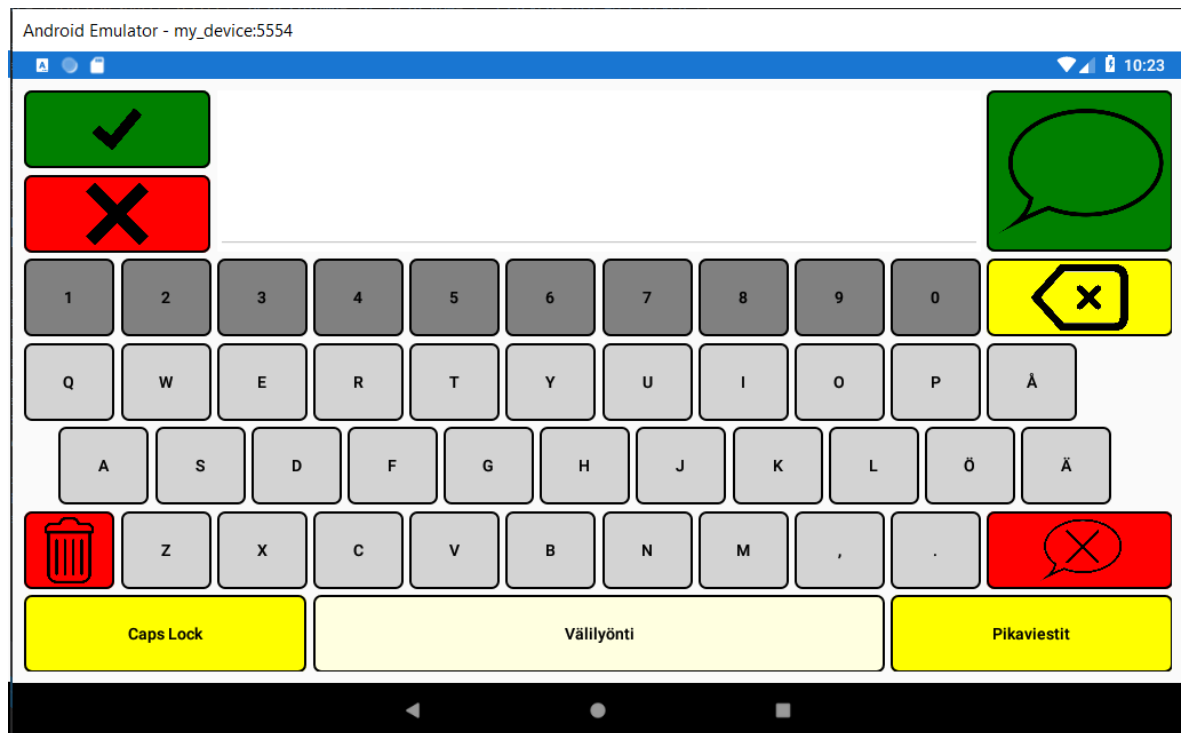
Kehitystyön tavoitteena oli suunnitella ja kehittää uusi alustariippumaton kommunikatiohjelma, jolla saadaan laajennettua apuvälinekeskuksien laitekantaa.

Blabber-ohjelmiston kehitys aloitettiin vertailemalla, mikä muissa vastaavissa ohjelmissa on hyvää ja mikä huonoa, jonka jälkeen keskusteltiin Etelä-Pohjanmaan sairaanhoitopiirin alueellisen apuvälinekeskuksen puheterapeutin sekä apuvälineasiantuntijan kanssa ohjelmiston käytettävyydestä ja mahdollisesta käyttöliittymän ulkoasusta.

Ulkoasu suunniteltiin piirtämällä raakavedoksia paperille. Vedoksista valittiin mieltyväin, ja ohjelmistoa alettiin rakentaa tämän raakavedoksen ympärille.

Alusta asti yksi suunniteltu ominaisuus sovellukseen oli piilotettava vasen ja oikea solupaneeli, jolloin esim. pienemmät näytöt voivat hyödyntää mahdollisimman paljon käytettävää tilaa puhesoluille. Lopulliseen versioon annettiin mahdollisuus valita, ovatko kyseiset paneelit piilossa vai näkyvissä.

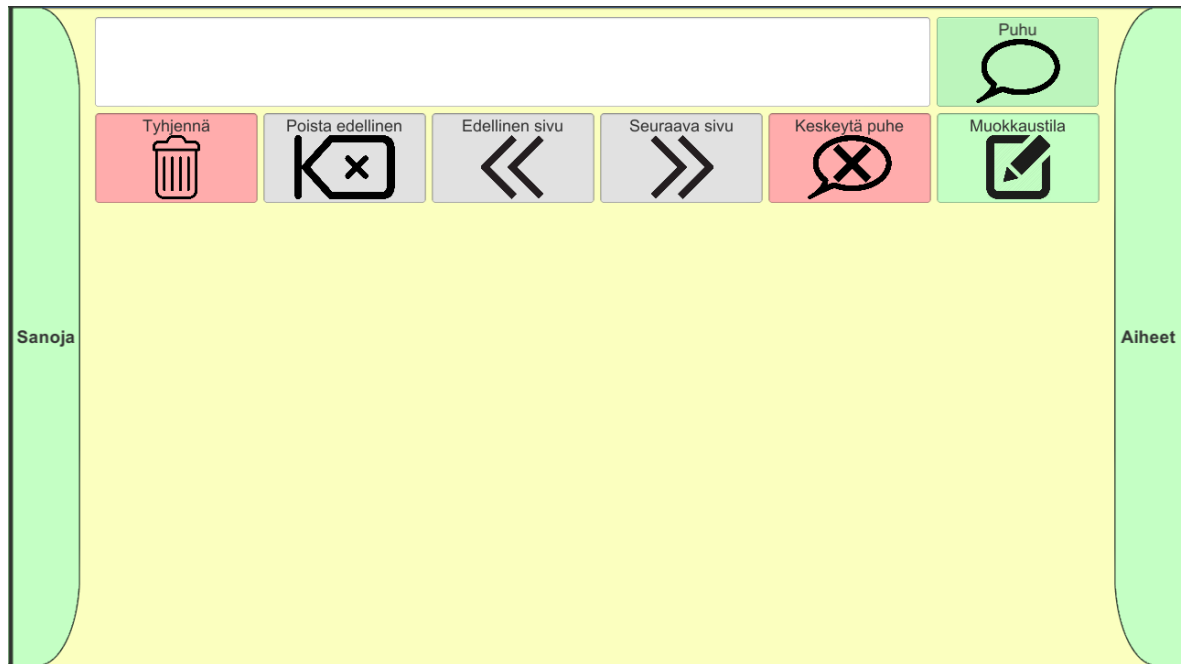
Raakasuunnitelman valmistumisen jälkeen testattiin muutamaa tekniikkaa ja kehystä, vaihtoehtoina olivat Unity-pelimoottori, Xamarin Framework sekä täysin naatiivi mobiilikehittäminen. Ensimmäinen prototyyppi tehtiin käyttämällä Xamarin frameworkia, mutta tällaisen sovelluksen kehittäminen sillä osoittautui raskaaksi prosessiksi, jonka vuoksi lopulta päädyttiin kehittämään sovellusta Unitylla.



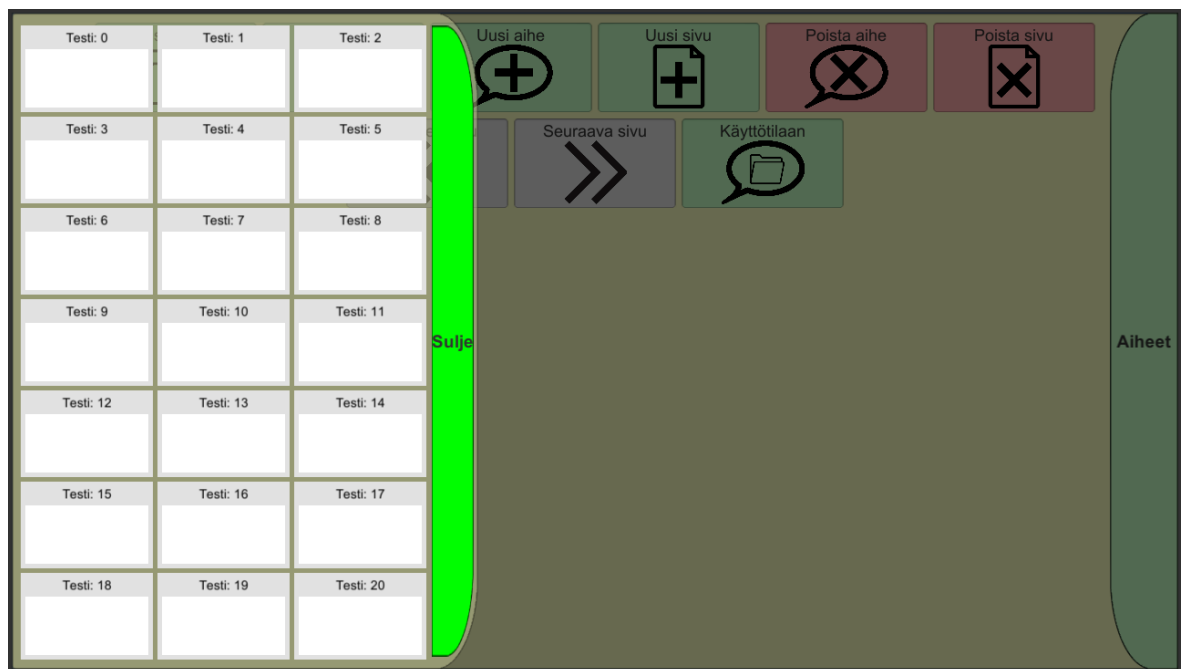
Kuva 1. Xamarin-prototyyppi Android-emulaattorissa.

Xamarin-versio oli yksinkertainen teksti puheeksi -sovellus, jonka päätoiminto oli kirjoittaminen. Sovellus oli yhteensopiva EVA Facial Mouse -sovelluksen kanssa, jolla Android-laitetta voi hallita pelkillä pään liikkeillä.

Unitylla sovelluksesta tehtiin useampi prototyyppi, joilla testattiin ominaisuuksia ja hiottiin käyttöliittymää, Suurin osa prototyyppiversioista on rakennettu alusta alkaen eri Unityn versioilla. Ensimmäinen prototyyppi tuli valmiiksi helmi-maaliskuun vaihteessa 2018. Tässä versiossa ei vielä pystynyt varsinaisesti tekemään mitään, vaan se toimi lähinnä konseptina ulkoasusta ja käyttöliittymästä. Tässä vaiheessa oli tiedossa jo suurin osa perusominaisuuksista, kuten erillinen käyttäjätila ja muokkaustila.

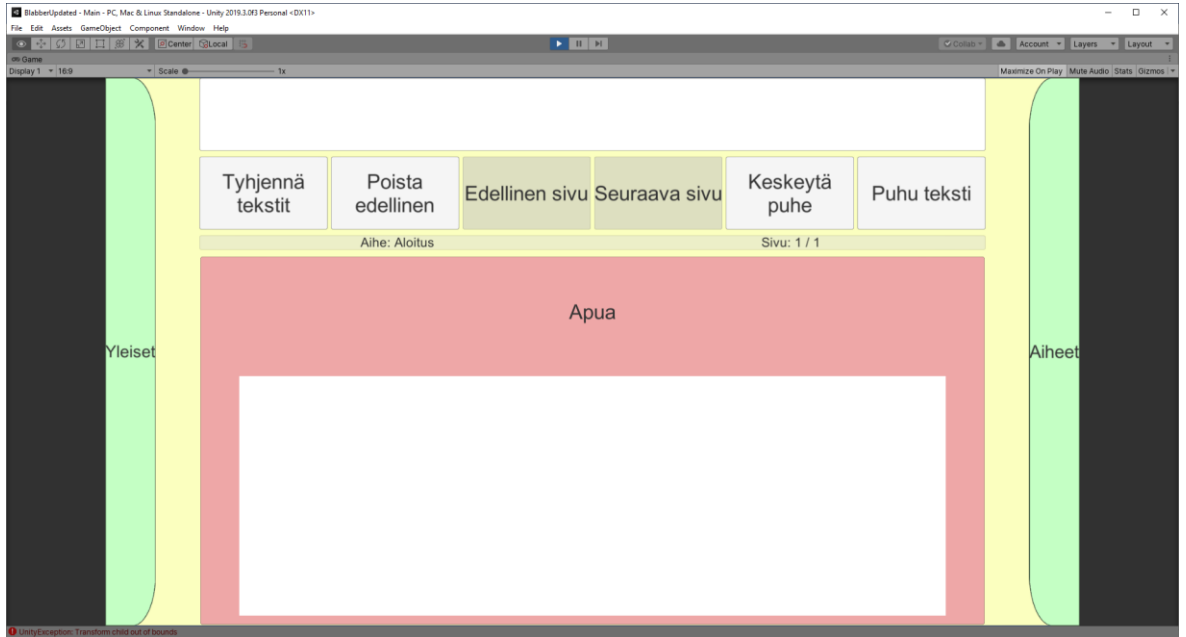


Kuva 2. Kommunikointisovellus, prototyyppi nro 1. Käyttäjätila.

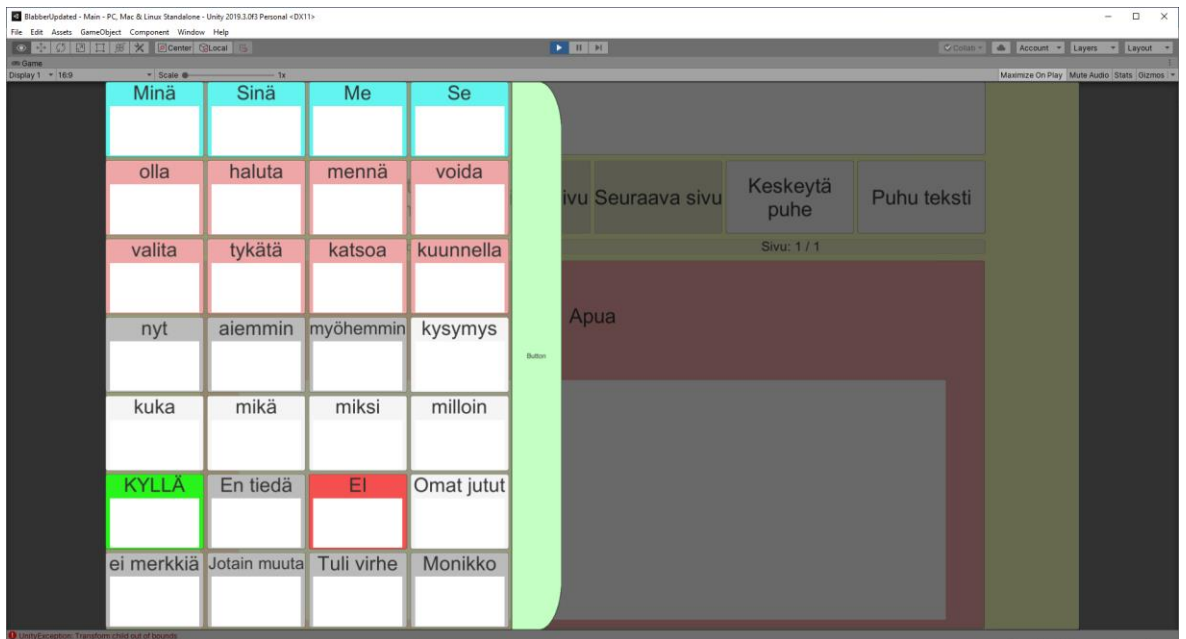


Kuva 3. Kommunikointisovellus, Prototyyppi nro 1. Muokkaustila ja avoin sivupaneeli.

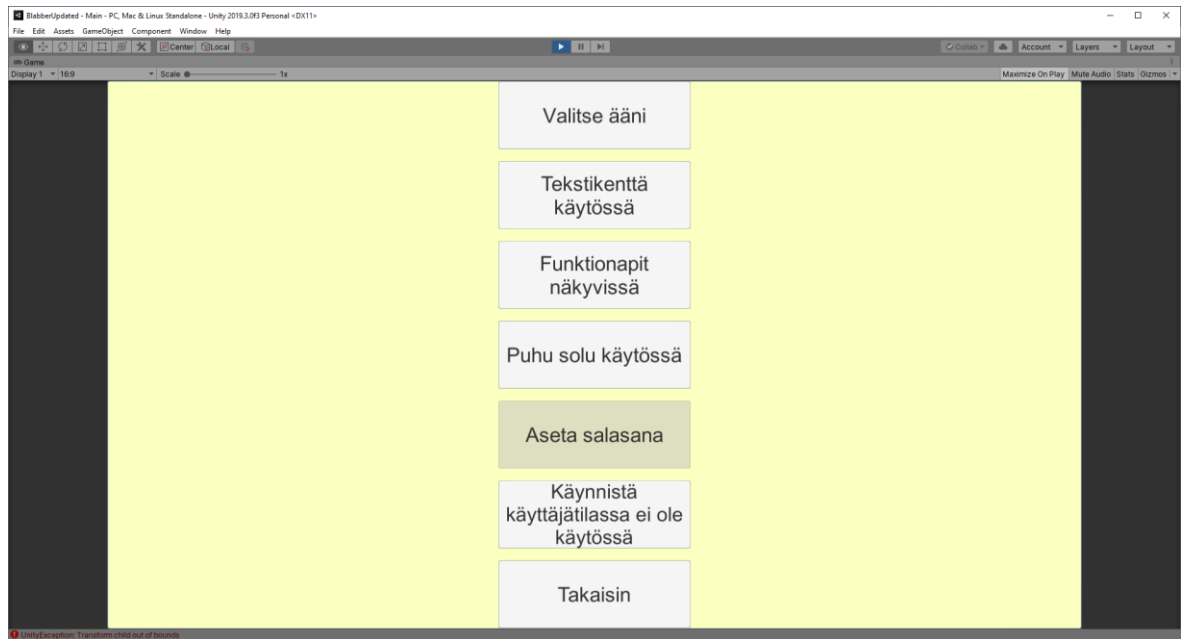
Prototyyppien kehittyessä hahmottui tarkemmin myös haluttu ulkoasu ja käyttöliittymä. Kolmas prototyyppi oli jo ominaisuuksiltaan paljon kehittyneempi aikaisempiin versioihin nähden. Sitä esiteltiin Tikoteekkiverkostolle toukokuussa 2018. Samalla sovellus sai lopullisen nimensä Blabber, eli vapaasti suomennettuna Pälättäjä. Nimi valikoitui, koska ohjelmalle haluttiin sitä kuvaileva kansainvälinen nimi. Pikaisen haun perusteella kyseinen nimi oli käytettävissä.



Kuva 4. Blabber, prototyyppi nro 3.



Kuva 5. Blabber, prototyyppi nro 3, avoin sivupaneeli.



Kuva 6. Blabber, prototyyppi nro 3. Asetusvalikko.

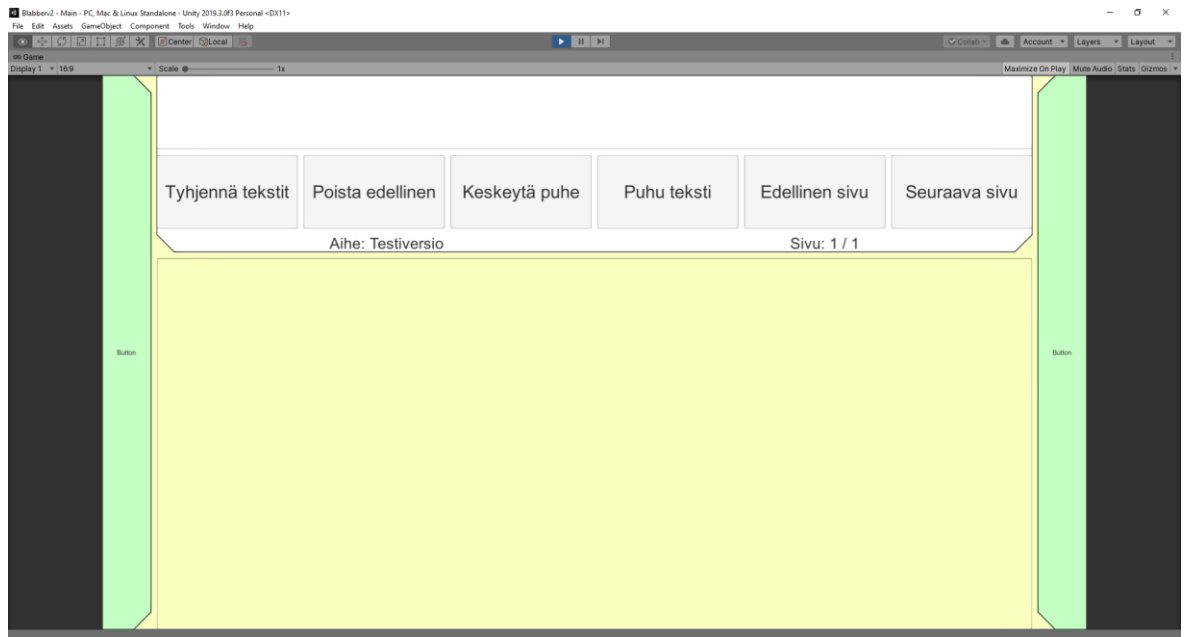
Tikoteekkiverkoston esityksen jälkeen pidettiin lyhyt kommentointituokio, jossa verkoston asiantuntijat pääsivät esittämään kehitysideoita, ehdotuksia sekä kysymään tarkentavia kysymyksiä sovelluksesta. Näiden ideoiden ja ehdotusten perusteella kehitettiin seuraavat versiot.

Kehitysideoiden pohjalta saatiin myös tarkennettua ohjelmaan sisällytettävät ominaisuudet ja toiminnot, jotka ovat:

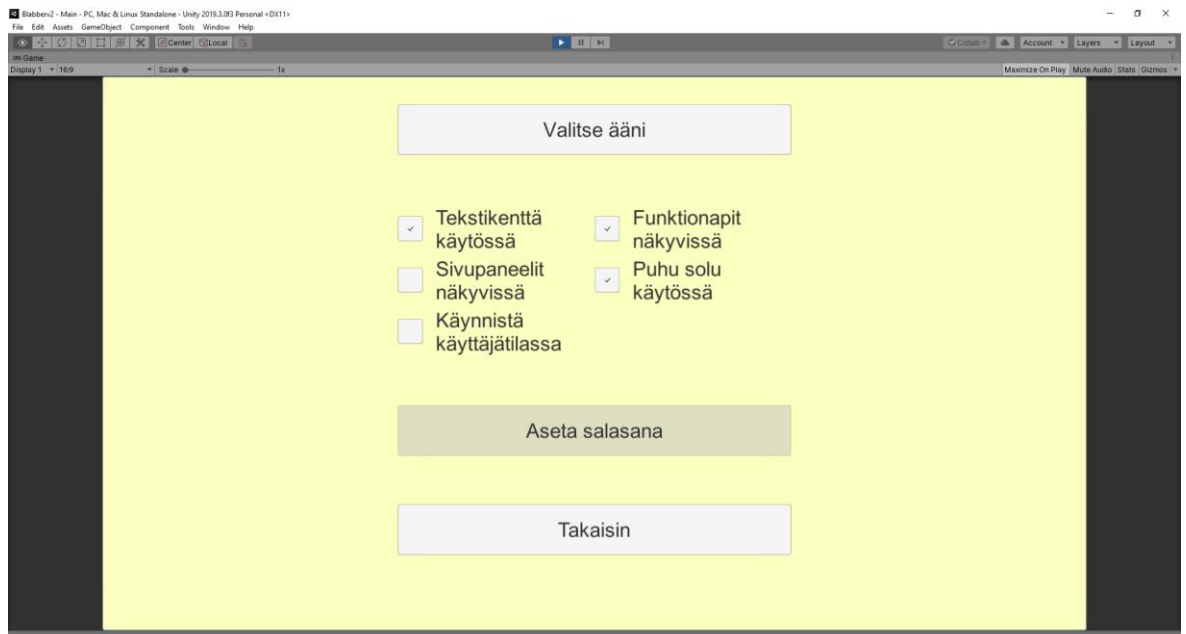
- TTS-toiminto
- aihekansioiden luominen
- sivujen luominen
- puhesolujen luominen
- aihekansioiden, sivujen sekä puhesolujen lisäys ja poisto
- solujen värien, kuvan ja tekstin muokkaus
- saumaton vaihto sivujen ja aiheiden välillä
- puhekansion tallennus ja siirto laitteelta toiselle alustasta riippumatta
- salasanasuojaus muokkaustilalle ja asetuksille
- Papunet-kuvapankin integrointi
- kuvien haku laitteen tiedostojärjestelmästä.

Seuraavassa versiossa Blabber kehittyi niin ulkonäöllisesti kuin myös käytännön osalta. Tällöin jäi myös prototyyppimäinen suunnittelu pois, ja kehityksessä otettiin

käyttöön Scrum-projektinhallinta. Samalla kehitys siirtyi iteroivaan menetelmään. Sovelluksen ensimmäiset versiot jaottelivat muokkaus-, käyttäjä- sekä asetustilan omiin ”Sceneihin”. Tämä ratkaisu hylättiin ja koko sovellus rakennettiin yhteen sceneen, mikä mahdollisti yhtenäisemmän ulkoasun, käyttöliittymän ja myös selkeämmän iteroivan kehitysmenetelmän.

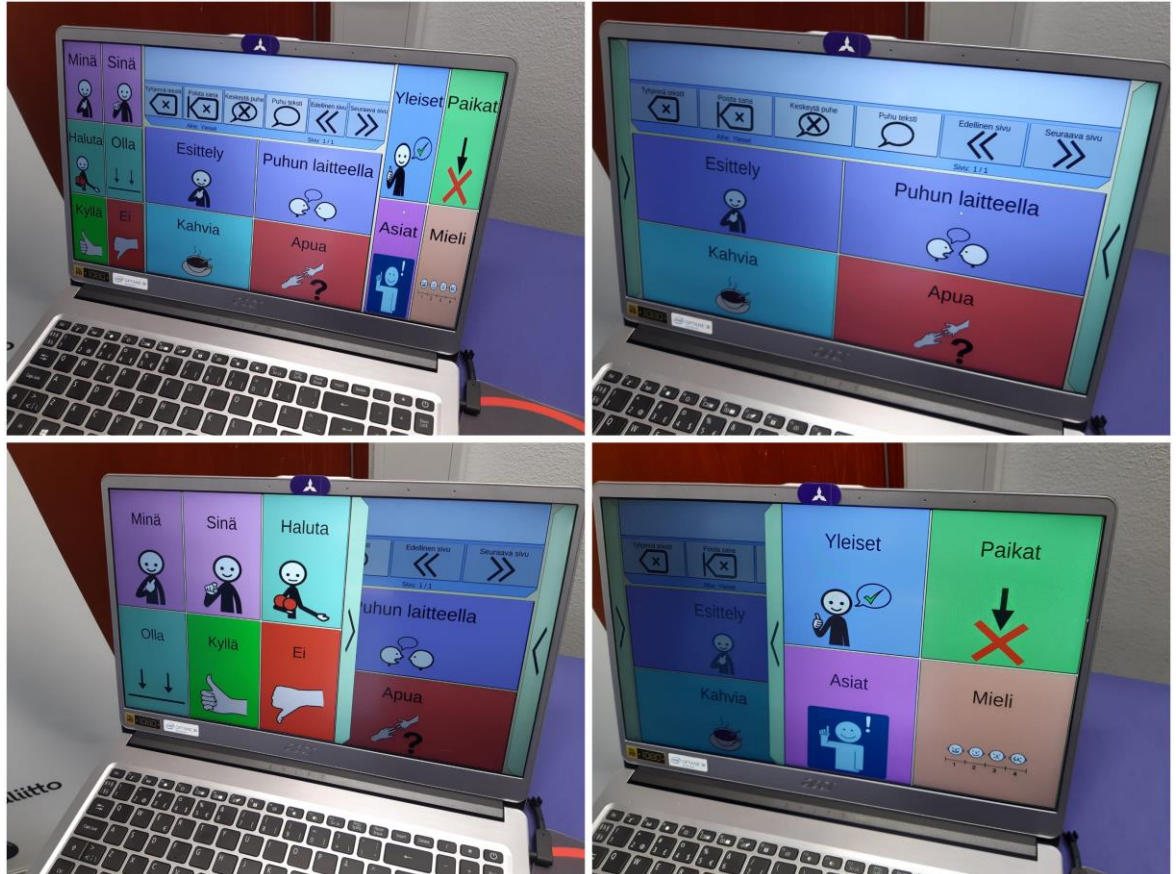


Kuva 7. Blabber, versio nro 0.6. Käyttäjätila.



Kuva 8. Blabber, versio nro 0.6, asetustilikko.

Versio 0.8 oli esillä myös Kehitysvammaliiton Sillalla-seminaarissa Papunetin esitelypisteellä yhden päivän ajan. Loput seminaarista jouduttiin valitettavasti peruuttamaan vuoden 2020 koronapandemian vuoksi.



Kuva 9. Blabber esillä Sillalla-seminaarissa 17.3.2020

Sovelluksen beta-versio saatiin jakoon huhtikuun alussa 2020. Sovellusta varten tehtiin nettisivu, josta sen voi ladata testaamista varten. Samalla julkaistiin käyttöoppaaksi 3 videota sovelluksen käytöstä ja ominaisuuksista.



Kuva 10. Blabber. versio nro 0.9, beta

Sovelluksen kehittämistä varten Etelä-Pohjanmaan Apuvälinekeskus lainasi apuvälinelaitteitaan testikäyttöä varten.

4.1.1 Sisällön luominen

Sovelluksen yksi tärkeimmistä ominaisuuksista on sisällön muokkaus, eli aihekanssioiden, sivujen, solujen lisääminen, poistaminen ja muokkaaminen.

Tämä toteutettiin luomalla Unityyn sivuille, aiheille ja soluille omat prefabit.

Näistä monimutkaisin lienee Solun prefab, jota varten tehtiin myös oma C#-luokka, jolle annettiin useita eri boolean-, string- sekä integer-arvoja sekä Unity-objektien arvoja, kuten väri ja tekstuuri.



Kuva 11. Solun prefab ja sen komponentit

```

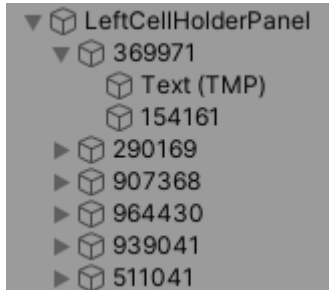
public class CellOptions : MonoBehaviour
{
    [Header("User changeable settings of Cell")]
    public bool speakCellText = true;
    public bool speakHiddenText;
    public bool noSpelling;
    public bool addToSentenceField = true;
    public bool isJumpToPage;
    public bool isEmptyCell;
    public string hiddenText;
    public int jumpToPage;
    public string jumpToSubject;
    public Color mainColor = Color.white;
}

```

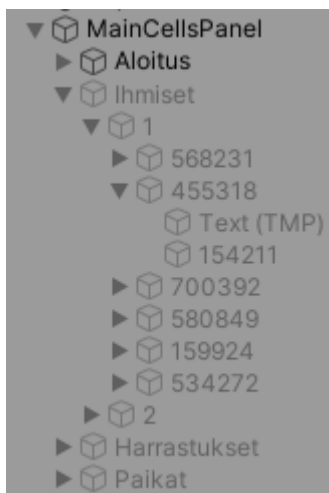
Kuva 12. Solun olioluokka

Muut prefabit ovat yksinkertaisempia.

Sisällöllä on oma hierarkiansa, jonka mukaisesti jokainen elementti asettuu ruudulle.



Kuva 13. Vasemman paneelin hierarkia Unity-editorissa



Kuva 14. Keskipaneelin hierarkia Unity-editorissa

MainCellsPanel on keskipaneelin Unity-objekti, joka hallitsee lapsien sijaintia. Lapsina tällä on tässä tapauksessa aiheet: Aloitus, Ihmiset, Harrastukset, Paikat. Niiden tilalla voi olla mitä käyttäjä itse haluaa.

Jokaisella aiheella voi olla lapsina 0-n määrä sivuja, tässä esimerkissä aiheella Ihmiset on kaksi sivua.

Sivuilla puolestaan voi olla lapsinaan 0-n määrä soluja, joista jokainen solu saa yksilöllisen satunnaisgeneraattorilla luodun tunnisteen.

```

private string CreateId(string str)
{
    string newstring = Random.Range(100000, 999999).ToString();
    if (IdExists(newstring, u.leftCells.transform) ||
        IdExists(newstring, u.rightCells.transform) ||
        (SelectedPage() != null && IdExists(newstring, SelectedPage().transform)))
    {
        CreateId(str);
    }
    return newstring;
}

private bool IdExists(string str, Transform parent)
{
    foreach (Transform item in parent)
    {
        if (item.name == str)
        {
            return true;
        }
    }
    return false;
}

```

Kuva 15. Satunnaistunnisteen luovat metodit.

Lähes jokaisella prefabilla on Layout Group -komponentti, joka määrittää tarkemmin, miten sen lapset asetellaan. Tämän lisäksi soluja sisältävien vanhempien sisällön koko muokkautuu lasten lukumäärän mukaan. Näin se mukautuu dynaamisesti käytettävissä olevan koon mukaan ruudukkomaiseen asetteluun.

```

public static Vector2 CellSizer(GameObject container, int cells, bool main, bool isSidePanelEnabled)
{
    int rows = 0;
    int columns = 0;
    switch (cells)
    {
        case 1:
            rows = 1;
            columns = 1;
            break;
        case 2:
            rows = 1;
            columns = 2;
            break;
        case 3:
            rows = 1;
            columns = 3;
            break;
        ...
        ...
        case 64:
            rows = 8;
            columns = 8;
            break;
    }
    if (container.GetComponent<RectTransform>().sizeDelta.x < container.GetComponent<RectTransform>().sizeDelta.y)
    {
        int newRows = columns;
        columns = rows;
        rows = newRows;
    }

    float x = (container.GetComponent<RectTransform>().sizeDelta.x - (5 * columns - 1)) / columns;
    float y = (container.GetComponent<RectTransform>().sizeDelta.y - (5 * rows - 1)) / rows;
    return new Vector2(x, y);
}

```

Kuva 16. Solun koon muuttava metodi

4.1.2 TTS

TTS eli Text To Speech -syntetisointi toteutettiin käyttämällä Unity-asettia nimeltä RT-Voice Pro. Tämän lisäys projektiin oli suhteellisen helppoa ja se toimi oikein lähes välittömästi.

Ainoat muutokset tähän tehtiin lisäämällä kyseisen asettin mukana tullut SAPI Unity-elementti projektiin ja aktivoimalla se hallintapaneelist.

4.1.3 Tallentaminen

Tallentaminen on tärkeä osa lähes mitä tahansa sovellusta. Blabber tallentaa sisällön JSON-muodossa. Unitylla on oma JSON-luokka, mutta tässä tapauksessa sitä ei käytetty, vaan sen sijaan JSON-tiedosto rakennettiin itse, sillä Unityn oma JSON Utility -luokka ei hallitse Array-tyypin serialisointia ja deserialisointia.

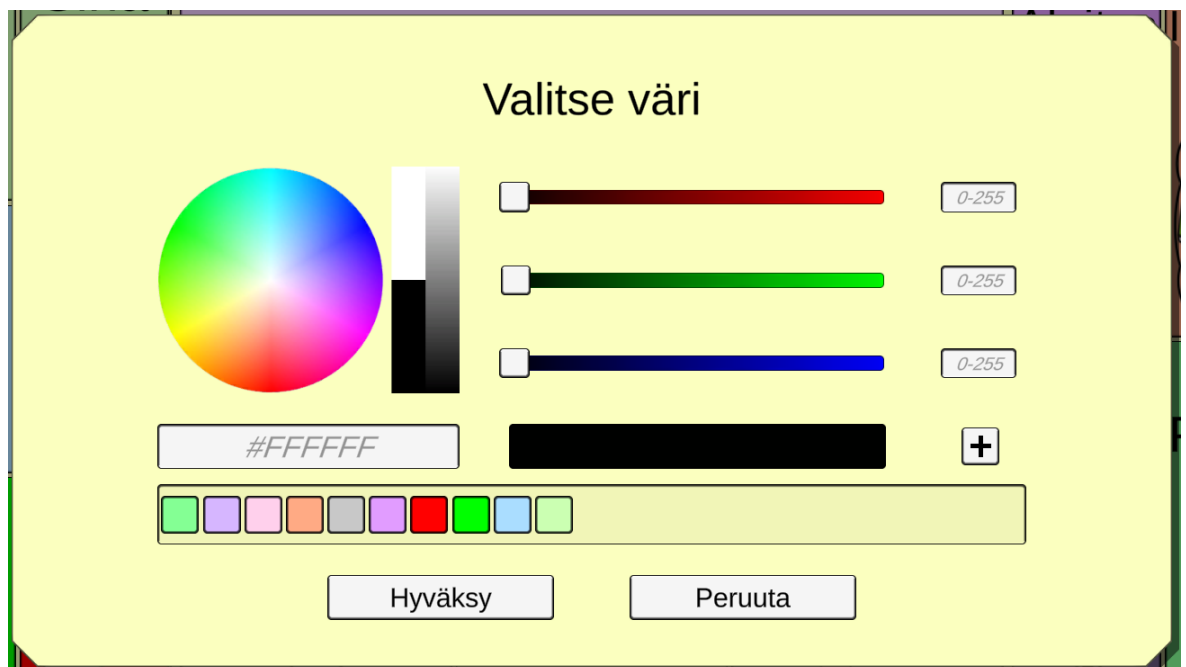
Sovellus rakentaa JSON-merkkijonon manuaalisesti ja tallennus tapahtuu binääridatana.

Kuvien tallentaminen oli haastavaa, sillä tekstuurit täytyi muuttaa JSON-muotoa varten tekstimuotoon. Tämä tapahtui kääntämällä haluttu tekstuuri ensin binääridataksi ja siitä Base64-muotoiseksi merkkijonoksi, joka tallennetaan JSON-tietueeseen.

Tämä on kuitenkin suhteellisen virheille altis ratkaisu, jolle tätä kirjoittaessa on jo parempia vaihtoehtoja mietinnässä.

4.1.4 Värivalikko

Solujen ja aiheiden värin valintaa varten täytyi tehdä erillinen värivalikko, jotta käyttäjällä olisi mahdollisimman helppo valita käytettäviä värejä. Värin voi valita joko väriympyrästä, syöttämällä RGB-arvot tekstikenttiin, säätämällä niitä liikusäätimillä tai kirjoittamalla halutun värin heksadesimaaliarvon omaan kenttäänsä. Jotta valittua väriä voisi käyttää myös myöhemmin, esimerkiksi verbejä sisältävien solujen erotukseksi, voi värin myös tallentaa ja käyttää siitä myöhemmin uudestaan.



Kuva 17. Värivalikko

Värivalikon väriympyrän koodauksessa täytyi laskea kursorin sijainnin ja väriympyräkuvan sijainnin sekä päätelaitteen ruudun koon ja väriympyräkuvan koon erotus,

kun hiiren painike on alhaalla. Tässä käytettiin apuna Unityn RectTransformUtility-luokkaa.

```
public class ColorPicker : MonoBehaviour, IPointerDownHandler, IPointerUpHandler
{
    public Canvas canvas;
    public GameObject prefab;
    bool isDown = false;
    Vector2 pos = new Vector2(0, 0);
    UserInterface u;

    private void Start()
    {
        u = FindObjectOfType<UserInterface>();
        canvas = FindObjectOfType<Canvas>();
        prefab = Instantiate(prefab, new Vector2(0,0), Quaternion.identity, transform);
    }

    private void Update()
    {
        if (isDown == true)
        {
            if (RectTransformUtility.RectangleContainsScreenPoint(GetComponent<RectTransform>(), Input.mousePosition) == true)
            {
                pos = RectTransformUtility.PixelAdjustPoint(Input.mousePosition, GetComponent<RectTransform>(), canvas);
                Rect pixelRect = RectTransformUtility.PixelAdjustRect(GetComponent<RectTransform>(), canvas);
                Vector3[] corners = new Vector3[4];
                GetComponent<Image>().rectTransform.GetWorldCorners(corners);
                Rect newRect = new Rect(corners[0], corners[2] - corners[0]);
                int x = (int)Input.mousePosition.x - (int)newRect.x;
                int y = (int)Input.mousePosition.y - (int)newRect.y;
                float xDiff = (pixelRect.width * 100 / newRect.width - 100);
                float yDiff = (pixelRect.height * 100 / newRect.height - 100);
                x = (int)(x + (xDiff * x / 100));
                y = (int)(y + (yDiff * y / 100));
                Color color = GetComponent<Image>().sprite.texture.GetPixel(x, y);
                u.hexCode.text = ColorUtility.ToHtmlStringRGB(color);
                prefab.transform.position = new Vector2(Input.mousePosition.x, Input.mousePosition.y);
            }
        }
    }

    public void OnPointerDown(PointerEventData eventData)
    {
        isDown = true;
    }

    public void OnPointerUp(PointerEventData eventData)
    {
        isDown = false;
    }
}
```

Kuva 18. ColorPicker-luokka kursorin sijainnin laskentaan ja värin valintaan väriympyrästä.

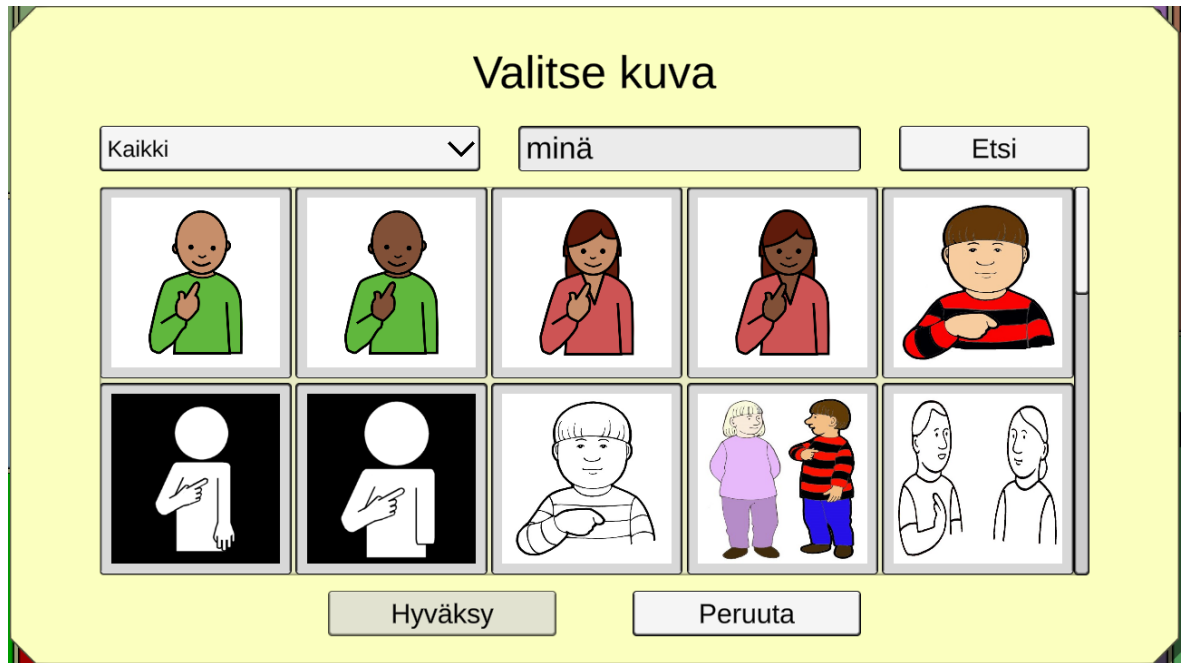
4.1.5 Siirto toiselle laitteelle

Yksi tärkeimmistä ominaisuuksista oli tallennetun datan siirto laitteelta toiselle helposti ja nopeasti. Tähän käytettiin eToilen File Transfer Serveriä, joka lähettää ja vastaanottaa dataa UDP-protokollan välityksellä.

File Transfer Serverin kehittäjä teki tätä projektia varten version työkalustaan, jolla vastaanottava laite voi havaita tulevan tiedoston siirron, ja näin joko sallia tai pysäyttää sen.

4.1.6 Kuvahaku

Kuvahakua varten integroitiin Papunetin JSON-rajapinta sovellukseen. Kuvahaulla voi siis hakea Papunetin kuvapankista kuvia eri hakusanoilla ja määritteillä.

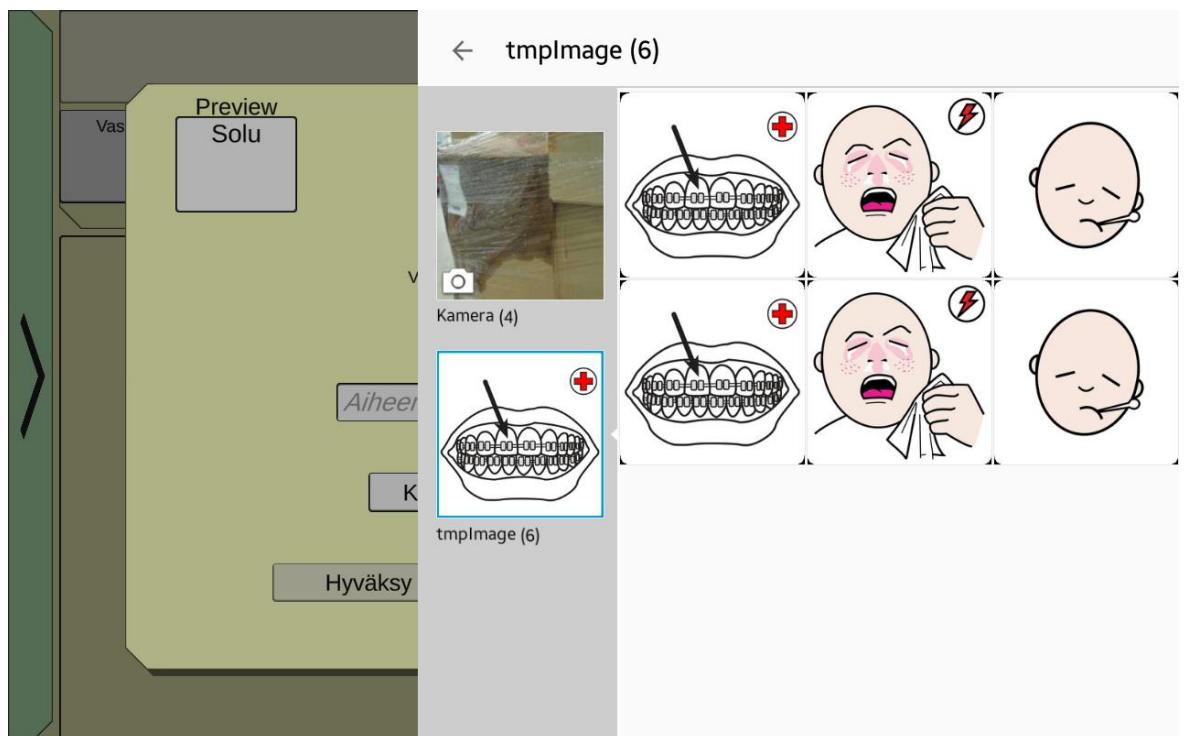


Kuva 19. Haku Papunetin kuvapankista

Kuvahakuun lisättiin myös ominaisuus, jolla voi hakea kuvia laitteen tiedostojärjestelmästä. Tätä varten käytettiin Windows-laitteiden tiedostojärjestelmälle eToilen FileBrowser Unity-asettia sekä Android-laitteille Süleyman Yasir Kulan tekemää Unity-asettia nimeltä Native Gallery.



Kuva 20. Kuvahaku Windows-laitteen tiedostojärjestelmästä.



Kuva 21. Kuvahaku Android-laitteen tiedostojärjestelmästä.

5 TULOKSET JA YHTEENVETO

Opinnäytetyön tavoitteena oli kehittää kommunikointiohjelmisto, joka toimii lähes kaikilla käyttöliittymillä ja laitealustoilla. Opinnäytetyössä päästiin pääosin haluttuun tavoitteeseen, vaikkakin sovellus vaati vielä jatkokehitystä.

Sovellukseen saatiin sisällytettyä kaikki luvussa Kehitystyö määritellyt ominaisuudet ja sen betaversio annettiin Tikoteekkiverkoston ja Papunetin käyttöön 6.4.2020.

Tämän lisäksi sovellukselle tehtiin kolme opasvideota, joissa käydään suurin osa Blabberin toiminnoista läpi. Videot julkaistiin Youtubessa ja osoitteessa <http://pwproductions.fi/portfolio-Blabber.html>

Sovelluksesta toivottiin lisää kommentteja ja jatkokehitysideoita, mutta otanta oli hyvin heikko. Näiden pohjalta saatiin kuitenkin tehtyä lyhyt ohjelmavirheluettelo:

- Android-puhelimilla ei pääse takaisin päävalikkoon
- DialoQ Taivuttimen sekä DialoQ Speech äänisynteesien kanssa puheessa on lyhyt viive
- Tallennuksen lukeminen ei lataa kaikkia kuvia oikein.

Tätä kirjoittaessa sovelluksen jatkokehityksestä käydään keskusteluja Code-Q-ohjelmistoyhtiön kanssa sovelluksen mahdollista kaupallistamista varten.

Sovelluksen jatkokehitystä varten täytyy myös miettiä mahdollisia lisäominaisuuksia, kuten tukea katseohjaimille, askellusta ja laajempia muokkausmahdollisuuksia.

Sovelluksessa on paljon hyviä puolia ja piilotettavat sivupaneelit ovat mielestäni hyvä lisäys tämänkaltaisiin sovelluksiin, sillä pienemmillä päätelaitteilla voi pienten solujen hahmottaminen olla hankalaa.

Tämänhetkinen sovellus vaatii kuitenkin vielä paljon jatkokehittämistä, ohjelmavirheiden korjausta sekä markkinatutkimusta. Suuri osa ominaisuuksista, kuten tiedoston siirto, tallentaminen ja ehkä myös TTS-toiminto joudutaan mahdollisesti rakentamaan uudelleen.

Tällaisenaan sovellus kelpaa kuitenkin esittely- ja testikäyttöön. Se on saanut runsaasti myönteisiä kommentteja Papunetin henkilökunnalta, Code-Q:lta sekä Etelä-Pohjanmaan apuvälinekeskukselta.

Yhteenvetona voisi sanoa projektin onnistuneen hyvin. Tavoite saavutettiin ja kentällä on selkeästi kiinnostusta kotimaiseen laadukkaaseen kommunikointisovellukseen. Tätä varten Blabberia täytyy kuitenkin kehittää edelleen. Lisäksi täytyy hakea rahoitusta, sillä yksityisenä sovelluskehittäjänä resurssit ovat hyvin niukat, varsinkin kun kysymykseen tulee eri ohjelmistolisenssien hankinnat.

LÄHTEET

- Adepoju, A. 2018. Why are most Android phones cheaper than iPhones. [Verkkoartikkeli]. Dignited 17.11.2018. [Viitattu 17.2.2020], Saatavissa: <https://www.dignited.com/36659/why-are-most-android-phones-cheaper-than-iphones/>
- Bowles, C. 2013. Looking beyond User-Centered Design. [Verkkoartikkeli]. A list apart 1.2.2013. [Viitattu 21.3.2020]. Saatavissa: <https://alistapart.com/column/looking-beyond-user-centered-design/>
- Bullit Group Research. 2017. Rapid growth in rugged phones. [Verkkoartikkeli]. Bullit Group 8.2017. [Viitattu 17.2.2020], Saatavissa: <https://bullitt-group.com/wp-content/uploads/2017/08/Rugged-market-position-paper-v8.pdf>
- Crosstales 2020. Rt-voice pro documentation. [Verkkoartikkeli]. Crosstales 20.4.2020. [Viitattu 25.4.2020]. Saatavissa: <https://www.crosstales.com/media/data/assets/rtvoice/RTVoice-doc.pdf>
- Higginbotham, J. & Jacobs, S. 2011. The Future of the Android Operating System for Augmentative and Alternative Communication. [Verkkoartikkeli]. American Speech-Language-Hearing Association 1.6.2011. [Viitattu 27.4.2020], Saatavissa: <https://pubs.asha.org/doi/abs/10.1044/aac20.2.52>
- Huhtanen, K. 2011. Puhetta tukevat ja korvaavat kommunikointimenetelmät Suomessa. Helsinki: Kehitysvammaliitto, Oppimateriaalikeskus Opike.
- Kumar, H. 2019. Unity 3D Game Development: Evaluating the Pros & Cons. [Verkkoartikkeli]. Sysbunny 15.11.2019. [Viitattu 25.4.2020]. Saatavissa: <https://www.sysbunny.com/blog/unity-3d-game-development-evaluating-the-pros-and-cons/>
- Monsuarez, J. 2019. File Transfer Server Cross-platform file transfer solution over UDP. [Verkkoartikkeli]. eToile 2019. [Viitattu 25.4.2020]. Saatavissa: <https://drive.google.com/file/d/1sLJQ8BEEP-KEIkPmEL68QICeOO4p3gAsM/view>
- Niemelä, H. 2020. Sovelluksen käytettävyys. [Verkkoartikkeli]. Seinäjoen ammattikorkeakoulu 31.1.2020. [Viitattu 7.3.2020]. Saatavissa: <http://urn.fi/URN:NBN:fi-fe202002115036>
- Papunet. 2019a. Kommunikointiohjelma. [Verkkoartikkeli]. Papunet 11.12.2019. [Viitattu 20.4.2020], Saatavissa: <https://papunet.net/tietoa/kommunikointiohjelma>

- Papunet. 2019b. Kommunikointiohjelmat PC-tietokoneille. [Verkkoartikkeli]. Papunet 10.12.2019. [Viitattu 28.4.2020]. Saatavissa: <https://papunet.net/tietoa/kommunikointiohjelmat-pc-tietokoneille>
- Papunet. 2019c. Mitä on saavutettavuus? [Verkkoartikkeli]. Papunet 2019. [Viitattu 23.4.2020]. Saatavissa: <http://papunet.net/saavutettavuus/mita-on-saavutettavuus>
- Papunet. 2020a. Puhetta korvaava kommunikointi eli AAC. [Verkkoartikkeli]. Papunet 5.3.2020. [Viitattu 3.4.2020]. Saatavissa: <https://papunet.net/tietoa/puhetta-korvaava-kommunikointi-eli-aac>
- Papunet. 2020b. Mitä on puhevammaisuus? [Verkkoartikkeli]. Papunet 5.3.2020. [Viitattu 3.4.2020]. Saatavissa: <https://papunet.net/tietoa/mita-on-puhevammaisuus>
- Patil, P, P. & Alvares, R. 2015. Cross-platform Application Development using Unity Game Engine. [Verkkoartikkeli]. International Journal of Advance Research in Computer Science and Management Studies 4.4.2015. [Viitattu 28.4.2020]. Saatavissa: https://www.researchgate.net/profile/Pratik_Patil21/publication/312591645_Cross-platform_Application_Development_using_Unity_Game_Engine/links/5884d39faca272b7b44a847f/Cross-platform-Application-Development-using-Unity-Game-Engine.pdf
- Schwaber, K., Sutherland, J., 2017. The Scrum Guide. [Verkkoartikkeli]. Scrum 11.2017. [Viitattu 24.4.2020]. Saatavissa: <https://www.scrum-guides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- SFS-EN ISO 9241-210:2019. 2019. Ergonomics of human-system interaction. Part 210: Human-centred design for interactive systems (ISO 9241-210:2019). Helsinki: Suomen standardisoimisliitto
- Tanant, F. 2018. Best game engines for beginners: 14 Alternatives. [Verkkoartikkeli]. Website Tool Tester 29.11.2018. [Viitattu 25.4.2020]. Saatavissa: <https://www.websitetooltester.com/en/blog/best-game-engine/>
- Thinkwik. 2018. CryEngine vs Unreal vs Unity: Select the Best Game Engine. [Verkkoartikkeli]. Medium 20.4.2018 [Viitattu 25.4.2020]. Saatavissa: <https://medium.com/@thinkwik/cryengine-vs-unreal-vs-unity-select-the-best-game-engine-eaca64c60e3e>
- Tikoteekki, 2018. Opas kommunikoinnin apuvälinepalveluihin. [Verkkoartikkeli]. Tietotekniikka- ja kommunikaatiokeskus 2020. [Viitattu 28.4.2020]. Saatavissa: https://www.kehitysvammaliitto.fi/tikoteekki/wp-content/uploads/sites/2/2018/08/Opas_kommunikoinnin_apuvälinepalveluihin2.pdf

Waller, A. & Balandin, S, A. & O'Mara, D, A. & Judson A, D. 2005. Training AAC Users in User-Centred Design. [Verkkootikkeli]. Accessible Design in the Digital World Conference 2005. [Viitattu 28.4.2020]. Saatavissa: https://www.scienceopen.com/document_file/6ebb0c48-015b-427f-aa4b-2cc70af253d4/ScienceOpen/001_Waller.pdf