

Iida Salminen

HUMANOIDIPALVELUROBOTIN KÄYTTÖÖNOTTO
TUKIVIITTOMIEN TULKIKSI

Sähkö- ja automaatiotekniikan tutkinto
2018

HUMANOIDIPALVELUROBOTIN KÄYTTÖÖNOTTO TUKIVIITTOMIEN TULKIKSI

Salminen, Iida
Satakunnan ammattikorkeakoulu
Sähkö- ja automaatiotekniikan koulutusohjelma
Huhtikuu 2018
Sivumäärä: 86
Liitteitä: 24

Asiasanat: humanoidipalvelurobotti, tukiviittomat, ohjelmointiympäristö

Opinnäytetyössä tarkasteltiin Softbank Robotics -yhtiön valmistaman Pepper-humanoidirobotin käyttöönottoa tukiviittomien tulkkina opetustilanteissa. Työn tavoitteena oli toteuttaa opetustuokio tukiviittomin Pepper-palveluhumanoidirobottia hyödyntäen. Tukiviittomien avulla Pepper opettaa kaupassakäyntiä lapsille, ja kohderyhmänä voivat toimia esimerkiksi autismihäiriöiset tai kuulovammaiset lapset. Pepper kertoo, viitto ja näyttää kuvin, miten kaupassa toimitaan. Kuvallinen kerronta tapahtuu robotin rinnassa sijaitsevan tabletin kautta. Työn tilaajana toimi Satakunnan ammattikorkeakoulun Hyvinvointia edistävien teknologioiden tutkimusryhmä. Opetustuokiota käytettiin osana projektia, joka tehtiin yhteistyössä käännöspalvelu Evantia Oy:n kanssa.

Opinnäytetyössä tarkasteltiin Pepper-humanoidirobottia ja sen ohjelmointiympäristöä. Pepper-humanoidirobotille opetettiin viittomia välipistein aikajanelle sekä yhdistettiin eri ominaisuuksia, kuten puhetta, tabletin kuvaa ja opetettuja liikkeitä. Ohjelman rakentaminen aloitettiin tukiviittomien opettamisesta, ja rakentamisessa edettiin aina vaiheittain sitä mukaa kuin informaatiota karttui ja löydettiin sopivia ratkaisuja. Lähdemateriaalina hyödynnettiin Softbank Robotics -yhtiön kattavia verkkosivuja sekä runsaasti robotiikkaan liittyviä tieteellisiä artikkeleita. Softbank Robotics -yhtiön aldehynan sivuston tuki testasi myös ohjelmaa ja oli apuna ongelmatilanteiden selvittämisessä.

Opinnäytetyön tulokset osoittavat, että tukiviittomien toteutus Pepper-humanoidirobotille on mahdollista. Kuitenkin robotin rakenne ei mahdollista kaikkien tukiviittomien toteutusta: osa viittomista ei ole ymmärrettävissä liikeratojen rajoitteiden johdosta. Johtopäätöksenä on, että Pepper-robotia voidaan käyttää tulkkina vain tiettyjen viittomien toteutuksessa. Tulevaisuudessa ohjelmaa tullaan esittämään kouluissa ja sitä voidaan laajentaa muiden arjen askareiden opettamiseen lapsille. Opinnäytetyötä voidaan käyttää myös oppaana ohjelmointiympäristön opiskeluun sekä kehittää jatkosovelluksia Pepper-humanoidirobotille tukiviittomiin liittyen.

COMMISSIONING OF HUMANOID SERVICE ROBOT TO BE A SIGN LANGUAGE INTERPRETER

Salminen, Iida

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in automation technology

April 2018

Number of pages: 86

Appendices: 24

Keywords: humanoid service robot, sign language, programming environment

The purpose of this thesis was to program a tutorial program for kids by using Softbank Robotics's humanoid service robot Pepper. In thesis, the research part was to study the possibility of humanoid service robot Pepper to make signs. The humanoid service robot Pepper speaks and gestures some words of the sentences and during the sentence, Pepper's tablet shows the picture based on the sentence. The humanoid service robot Pepper teaches how to make groceries. While using sign language, target group could allocate for autistic and children with special needs. Mandator for this study was Satakunta University of applied Sciences's Welfare enhancing technologies research group. Tutorial program was used as part of the project which was cooperated with Evantia Oy.

In this thesis the humanoid service robot Pepper and its programming environment was considered. First Pepper was taught to use sign language by teaching signs for robot with intermediate steps. Second, behaviours (speech, picture and sign language) were combined while constructing the program. The program was proceeded systematically while information for programming were searched. Softbank robotics's website was used as a reference. Program was tested by Softbank Robotics Aldebaran support when appeared problems could not be solved.

The results of study show that Pepper could execute signs but not all of them. Pepper's structure and movement limits made it impossible to execute some of the signs. Pepper could execute some of the signs but the signs wasn't very understandable. Pepper can be used for certain signs. The tutorial program will be presented at schools. The tutorial program can be also applied for other everyday's routines. The thesis can be used as a guide for tutorial documentation of Pepper and for programming.

SISÄLLYS

1	JOHDANTO.....	6
2	PALVELUROBOTIIKKA.....	8
2.1	Palvelurobotiikan määritelmä ja sovellusalueet.....	8
2.2	Hoivarobotiikka palvelurobotiikan kehityksen tuloksena	8
2.2.1	PARO-hyljerobotti	10
2.2.2	Evondos E300-lääkeannostelurobotti.....	11
2.2.3	UV-desinfiointirobotti	12
2.2.4	BEAM Telepresence -robotit	12
2.2.5	TUG-kuljetusrobotti	13
2.2.6	GIM-robottialustat.....	14
2.2.7	TASKI IntelliBot -siivousrobotti.....	15
2.2.8	Zora-robotti	16
2.2.9	”Kaveri”-robotti Robosoft Kompai	17
2.3	Palvelurobotiikan haasteena yhteiskunnallinen hyväksyttävyyys	18
3	PEPPER-HUMANOIDIROBOTTI.....	21
3.1	Toimilaitteet	22
3.2	Pepper-robotin rakenne	24
3.2.1	Moottorit ja nivelet	24
3.2.2	Hätäseis ja latausasema	25
3.2.3	Kaiuttimet ja mikrofonit	26
3.2.4	2D- ja 3D-kamerat.....	27
3.2.5	Ledit	28
3.2.6	Anturit	29
3.3	Pepper-robotin kanssa toimiminen	33
3.3.1	Pepper-robotin käynnistäminen.....	33
3.3.2	Pepper-robotin kanssa kommunikointi.....	34
3.3.3	Sleep mode - ”Nukkumistila”.....	35
3.3.4	Pepper-robotin autonomous life on- ja off -tila.....	36
3.3.5	Pepperin sammuttaminen	36
3.3.6	Pepper-robotin ilmoitukset	36
4	CHOREGRAPHE-OHJELMOINTIYMPÄRISTÖ	37
4.1	Ohjelmointiympäristön esittely.....	37
4.2	Yhdistäminen robottiin	37
4.3	Pääkomponentit.....	38
4.3.1	Pääpalkki	39
4.3.2	Työkalupalkki	40

4.3.3	Box libraries	42
4.3.4	Robotin näkymä.....	46
4.3.5	Robotin sovellukset	47
4.4	Animointi	49
4.4.1	Timeline	49
4.4.2	Liikkeiden opettaminen	51
4.5	Python	54
4.6	Korjaus ja virhetilanteet	57
5	KAUPASSAKÄYNNIN OPETUSVIDEON TOTEUTUS JA VIITTOMIEN OPETTAMINEN PEPPERILLE.....	59
5.1	Alkuperäinen ohjelma.....	59
5.2	Alkuperäinen ohjelma.....	67
5.2.1	Lauseet ilman tukiviittomia.....	68
5.2.2	Lauseet tukiviittomilla.....	70
5.3	Uusi ohjelma	73
5.4	Ongelmakohdat ja ratkaisut	78
6	LOPPUPÄÄTELMÄT	80
	LÄHTEET.....	82
	LIITTEET	

1 JOHDANTO

Opinnäytetyössäni tarkastelen Pepper-humanoidirobotin käytettävyyttä tukiviittomien tulkkina opetustuokioissa. Pepper-robotin avulla voidaan esimerkiksi erityislapsille opettaa, miten kaupassa toimitaan. Ohjelmassa yhdistetään eri oppijoille tarkoitettuja metodeja, kuten puhumista, kuvan näyttämistä sekä viittomia. Aihe on mielenkiintoinen, sillä Pepper on ensisijaisesti asiakaspalveluun ja viihdyttämiseen tarkoitettu robotti. Pepper on tähän mennessä toiminut opetusvälineenä lukiolaisille ja nuoremmille opiskelijoille. Opiskelijat esimerkiksi pääsevät ohjelmoimaan yksinkertaisia ohjelmia Pepper-robotin kanssa. Pepper-robottia on myös käytetty eri tapahtumissa ja tilaisuuksien alkuohjelmissa toivottamaan vieraat tervetulleiksi. Eri yritykset haluavatkin Pepper-robotin vierailevaksi tähdeksi tilaisuuksiinsa. Siksi onkin kiinnostavaa tämän tutkimuksen kautta selvittää, voidaanko sovellusalueita laajentaa esimerkiksi juuri kouluihin ja opetukseen.

Tämä tutkimus on toteutettu Satakunnan ammattikorkeakoulussa Hyvinvointia edistävien teknologioiden tutkimusryhmän tarpeisiin ja tutkimus on osa projektia, joka on tehty yhteistyössä käännöspalvelu Evantia Oy:n kanssa. Työssä lähdetään liikkeelle tutustumalla Pepper-robottiin Softbank Robotics -yhtiön aldebaran sivustoa hyödyntäen. Sivusto antaa laajan tietopaketin Pepper-robotista ja sen toiminnasta. Sivustossa voidaan myös tutustua ohjelmointiympäristöön ja sen rakenteeseen, ja sivusto antaa hyvät eväät siihen, miten aloitetaan ohjelman tekeminen. Pepper-robotin ohjelmointiympäristönä toimii Choregraphe, jossa käytetään graafista ohjelmointia. Graafisen ohjelmoinnin toimintolaatikot on rakennettu python-ohjelmointikielellä ja siten työssä pääsee tutustumaan graafiseen ohjelmointiin, python-kieleen ja itse Choregraphe-ohjelmaan, sen tekovaiheisiin ja lopputuloksiin. Työ antaa kattavan esittelyn palvelurobotiikasta ja myös taitoja ohjelmoida humanoidirobottia.

Työssä tutustutaan ja esitellään myös palvelurobotiikkaa sekä erilaisia palvelurobotteja, joita on jo käytössä Suomessa. Esittelyn kautta saadaan vertailupohjaa siihen, mitä palvelurobotteja on jo olemassa ja missä niitä käytetään.

Työssä esitellään myös Pepper-robottia teknillisesti ja sitä, miten Pepper-robotti toimii. Näin saadaan kokonaisvaltainen kuva robotista ja käyttöohjeet robottia varten.

Hyvinvointia edistävien teknologioiden tutkimusryhmä voi tulevaisuudessa hyödyntää ohjelmaa kouluissa ja kehittää uusia sovelluksia Pepper-robotin ominaisuuksien puitteissa.

2 PALVELUROBOTIIKKA

2.1 Palvelurobotiikan määritelmä ja sovellusalueet

Palvelurobotti on sisä- ja ulkotiloissa toimiva, palvelutehtäviin rakennettu robotti, jota käytetään ihmisten kanssa toimimiseen ja ihmisen työtä helpottamaan. Palvelurobotit ovat liikkuvia ja mahdollisesti kappaleita käsitteleviä laitteita, jotka toimivat itsenäisesti ja vuorovaikutuksessa ihmisten kanssa. Palvelurobottien teknologia on rakennettu helppokäyttöiseksi ja turvalliseksi käyttäjälle. Robottien anturit mahdollistavat robottien liikkeiden toteuttamisen turvallisesti. (Salmi 2014)

Palvelurobotiikka jaetaan palveluiden mukaan yritys- ja kuluttajakuntaan. Yritystasolla palvelurobotiikka kerää suosiota, sillä palvelurobotteja otetaan yhä enemmän käyttöön esimerkiksi puolustusteollisuudessa, terveydenhuollon sektorilla ja lypsytehtävissä. Palvelurobottien sovelluksiin kuuluvat esimerkiksi maatalouden, siivouksen, rakentamisen ja lääketieteen (kirurgia, hoiva, kuntoutus) sovellukset sekä avaruus- ja vedenalaiset sovellukset. Esimerkiksi navettojen lypsyrobotit, erilaiset sisä- ja ulkologistiikan laitteet sekä sairaaloissa leikkausrobotit ovat käytössä yritysten arjessa. Henkilökohtaisten palvelurobottien yleistyminen on hitaampaa, mutta jatkuva kehitys tuo uusia robottisovelluksia helpottamaan ihmisten työtä kodeissa sekä koulutus- ja palvelusektorilla. Erilaisia palvelurobotteja kodin huoltotehtävissä ovat esimerkiksi pölynimurit, ruohonleikkurit ja koulutusta tukevat laitteet sekä liikkumisen avustajat. Palvelurobotiikkaan kuuluu myös kaivoskoneissa ja -laitteissa käytettävät robotit. Esimerkiksi älykkäät työkoneet liikkuvat autonomisesti kaivostunneleissa ilman kuljettajaa ja niitä ohjataan valvomoista. (Miniwebin www-sivut n.d.)

2.2 Hoivarobotiikka palvelurobotiikan kehityksen tuloksena

Palvelurobotiikkaa on kehitetty myös hoiva-, avustus- ja kuntoutustehtäviin. Käyttötarkoitukset vaihtelevat avustavista terapeuttisiin ja vuorovaikutusta tukeviin robotteihin. Esimerkiksi syöttökäsivartta ja pyörätuoliin liitettävää robottikättä sovelletaan ikääntyneiden ja vammaisten hoidossa.

Myös itsenäisesti asuvien asiakkaiden tueksi on kehitetty kuntoa ylläpitäviä robotteja, jotka tekevät hälytyksen kaatumisen tai muun hädän sattuessa ja jotka avaavat videoyhteyden lääkäriin tai omaiseen. Hoitolaitoksissa hoivarobotti on avustamassa ja ohjaamassa potilaan kehon liikkeitä, ja se asettaa sopivan vastuksen laitteelle potilaan oman jaksamisen ja kunnon mukaan. Esimerkiksi alla olevassa kuvassa (kuva 1) Indego-kävelyrobotti tukee liikuntarajoitteisten kuntoutusta ja antaa lisätukea seisoma- ja kävelyharjoittelussa.



Kuva 1. Folkhälsanin kuntoutusyksikössä harjoitetaan kävelyä kävelyrobotin kanssa. (Folkhansan www-sivut n.d.)

Suomessa ovat myös yleistymässä leikkaussalien leikkausrobotit. Leikkaussalien robotteja käytetään erityisesti pehmeiden sisäelinkudosten leikkaamisessa. Terapia- ja etäläsnaolorobotit rauhoittavat muistisairaita ja antavat hoitajille mahdollisuuden suunnitella työtään ja käyttää ajan tehokkaammin muihin tehtäviin. Myös työolosuhteet kevenevät henkisesti ja fyysisesti kuormittavassa työssä. Lääkeannostelurobotit ovat myös käytössä muistisairailta.

Kuitenkin taloudellisista syistä kaikkien edellä mainittujen laitteiden soveltaminen kotihoidon asiakkaille on rajallista, koska kunnilla ei välttämättä ole resursseja investoida korkeahintaisiin hoivarobotteihin.

Hoivarobotiikkaa on kehitetty myös ei-fyysiseen kuntoutukseen. Ihmistä muistuttavat humanoidit, kuten NAO, ja eläimenkaltaiset zoonoidit, kuten Paro-hyljerobotti, on kehitetty luomaan vuorovaikutusta ihmisten välillä. Esimerkiksi Paro-hylkeen on tutkittu vähentävän muistisairaiden levottomuutta ja lisäävän yhteisöllisyyden tunnetta vanhusten palvelukodeissa. NAO-humanoidista ovat hyötäneet myös autismihäiriöiset lapset, joiden on vaikea hahmottaa sosiaalisen vuorovaikutuksen tilanteita ja sääntöjä. NAOlle haetaan käyttökohteita myös ikäihmisten arjen virkistäjänä: pieni humanoidi on lukenut *Aamulehteä* ja pitänyt jumppaa vanhuksille Tampereen vanhainkodissa. (Salmi 2014)

2.2.1 PARO-hyljerobotti

PARO on hylkeen näköinen hoivarobotti (kuva 2), jonka tehtävänä on herätellä ihmistä kommunikointiin ja ylläpitämään vuorovaikutustaitoja. PAROn on kehittänyt japanilainen professori Takanori Shibatan vuonna 1993. Kooltaan Paro on 57-senttinen ja se painaa noin 3 kiloa. PARO sisältää ääniantureita, joiden avulla se pystyy oppimaan henkilöiden ääniä ja muistamaan, miten tuttu henkilö on. Valoantureiden avulla se ylläpitää vuorokausirytmää eli päivisin se on aktiivisempi kuin yöllä. Turkissa sijaitsevien tuntoantureiden avulla PARO pystyy reagoimaan silittelyyn ja asennon muutokseen. PARO aktivoi muistisairaita ihmisiä ja antaa virikkeitä potilaille. PAROn tarkoituksena on antaa hoivaa, rauhoittaa ja lievittää ahdistuneisuutta erityisesti muistisairaiden hoidossa. PARO on käytössä ympäri Eurooppaa, ja Suomessa hyljerobottia käytetään ainakin Havukosken vanhustenkeskuksessa, Helsingissä Kustaankartanossa ja Koskelassa sekä Oulun kaupungissa. (Innohoivan www-sivut n.d.)



Kuva 2. Ikääntyneille Paro on ollut ilahduttava asukas Vantaan Koivukylän vanhustentalossa. (Iltalehden www-sivut 2013)

2.2.2 Evondos E300-lääkeannostelurobotti

Evondos-lääkeannostelurobotti (kuva 3) tarjoaa asiakkaille lääkkeiden koneellista annosjakelua. Se on suunniteltu pitkäaikaislääkitystä käyttäville. Automaattiannostelijan sisällä olevalle kiekolle syötetään noin kahden viikon lääkkeet, ja robotti muistuttaa sanallisesti, valo- ja äänimerkein tai tarvittaessa puhelinsoitolla, milloin lääkkeet pitää ottaa. Robotti käyttää internetyhteyttä tiedonsiirtoon hoitajan käyttämän etähoitojärjestelmän kanssa. Näin hoitaja saa tiedon lääkkeen ottamisesta tai ottamatta jättämisestä. (Evondoksen www-sivut n.d.)



Kuva 3. Lääkeannostelurobotit kertovat, milloin seuraava lääke on saatavilla. (Turun sanomien www-sivut 2017)

2.2.3 UV-desinfiointirobotti

UV-desinfiointirobotti on desinfiointiin tarkoitettu robotti (kuva 4). Se on käytössä erityisesti sairaaloissa, mutta sitä käytetään myös tuotantolinjoilla ja jopa hotelleissa. Robotti kehitettiin ennaltaehkäisemään sairaalainfektioita, jotka aiheuttavat suuria kustannuksia sairaaloille. Desinfiointirobotti puhdistaa käyttäjän huoneessa merkityt alueet noin 10–15 minuutissa. Desinfiointirobotti rikkoo bakteerien ja virusten DNA-rakenteen ja estää siten epidemioiden kehittymisen ja leviämisen. UV-desinfiointirobotti voi tappaa 99,9 prosenttia bakteereista 2–3 metrin alueella, kun yleinen potilashuoneen koko on alle 10 metriä. (Meditas Oy:n [www-sivut](#) n.d.)



Kuva 4. UV-desinfiointirobotti puhdistaa huoneen noin 15 minuutissa. (Meditas Oy:n [www-sivut](#) n.d.)

2.2.4 BEAM Telepresence -robotit

BEAM+, BEAM+MAX ja BEAM Pro ovat etäohjattavia palvelurobotteja (kuva 5), joita käytetään virtuaaliseen kommunikointiin esimerkiksi hoitajien ja lääkärin välillä. Ne ovat liikkuvia robotteja, niillä on pitkä runko ja pieni LCD-näyttö, josta voi valita sovelluksia ja esimerkiksi käydä keskustelua etänä. Robottia voidaan käyttää esimerkiksi opetustapahtumissa ja sairaaloissa. Käyttäjä hallitsee robottia ja se on toimiva ratkaisu esimerkiksi etätyöskentelyyn. Sen kaksi HDR-kameraa mahdollistavat navigoinnin. Kaiuttimien avulla äänentoisto toimii keskustelutilanteissa.

BEAM-robotti voidaan yhdistää verkkoon WLAN-yhteyden avulla. Siihen on myös integroitu bluetooth, jotta yhteys voidaan jakaa. BEAM-robottia käytetään esimerkiksi kokouksissa sekä lääkärin ja potilaiden välisissä tapaamisissa. Robotti suorittaa myös kiertokäyntejä potilaiden luona ja hoitaa yhteydenpidon potilaiden ja omaisten välillä. (Meditas Oy:n www-sivut n.d.)



Kuva 5. Beam telepresence -robotti voidaan ohjata mihin vain, ja robotin välityksellä voidaan käydä keskusteluja. (Eagledatan www-sivut 2016)

2.2.5 TUG-kuljetusrobotti

TUG on liikkuva robotti, joka toimii automaattisena kuljetusjärjestelmänä (kuva 6). Sitä käytetään sairaaloissa esimerkiksi lääkkeiden, ruoka-annosten, liinavaatteiden ja jätteiden kuljettamiseen.

Robottijärjestelmä tukee Teollisuus 4.0 -mallia. Siten se mahdollistaa laitteiden ja järjestelmän yhdistämisen reaaliajassa tuotantokoneisiin ja ohjelmistoihin. Sen kulkua voi myös seurata vaikka tabletilla reaaliajassa. Antureiden ja kameroiden avulla se pystyy liikkumaan turvallisesti sairaalaympäristöissä ja esimerkiksi toimittamaan aterian oikeaan paikkaan turvallisesti ja sujuvasti. TUG vastaa reaaliajassa tuotantolinjan ERP/MES-signaaleihin ja muihin äkillisiin kutsuihin, joiden avulla pystytään ehkäisemään järjestelmien yllättäviä ja ei-toivottuja katkoja.

Sitä voidaan ohjata omalta älypuhelimelta, tabletilta tai seinään kiinnitetystä näppäinpäätteestä. Se kantaa maksimissaan 544 kg painoisen kuorman. (Eagledatan www-sivut 2016)



Kuva 6. TUG-robottia voidaan käyttää erilaisiin kuljetuksiin, ja siihen on vaihdettavissa automaattisesti kiinnittyvä kuljetusvaunuosa. (Eagledatan www-sivut 2016)

Suomessa on käytössä samankaltaisia kuljetusrobotteja, kuten Omron-mobiilirobotti, joka tunnettiin ennen Adept Lynxinä. Se on pienempi kooltaan ja se navigoi itse reitinsä ja säätelee nopeutensa sopivaksi. Robottia ohjataan ensin manuaalisesti, jolloin se piirtää alueesta kartan laserskannereilla. Piirtämänsä kartan avulla robotti navigoi helpoiten saavutettavan reitin valittuun kohteeseen. Se navigoi antureiden ja kameroiden avulla, jolloin se ei törmää esineisiin ja jolloin se osaa myös väistellä ihmisiä. Robotit pystyvät kuitenkin kuljettamaan vain maksimissaan 60–90 kg kuorman ilman erillistä vaunua. (Omronin www-sivut n.d.)

2.2.6 GIM-robottialustat

G/01 on suomalaisen GIMIN kehittämä robotti, joka soveltuu myös ulkokäyttöön (kuva 7).

Se on kooltaan kottikärryjä hieman pienempi ja sen maksiminopeus on 10km/h. Se pystyy kuljettamaan maksimissaan 200 kg kuorman. Se on varustettu erilaisilla antureilla ja kameratolpalla, jossa on neljä kameraa. Laser-luotaus mahdollistaa ympäristön tarkastelun ja navigoinnin reaaliajassa. Siten se ei törmäile puihin, ihmisiin tai muihin esteisiin. Sitä käytetään materiaalin kuljettamiseen ja oikein konfiguroituna se kykenee vaikkapa luomaan lunta pihalla. (Gimltd:n www-sivut n.d.)

Esimerkiksi Helsingin Otaniemessä istuinlisäosalla varustettuna robotti kuljetti chileläistä valtiomiestä ja esitteli erilaisia kohteita. Sitä käytetään myös 3D-mallinnukseen, esimerkiksi kaivostyössä robotti kulkee kaivoksen käytävillä ja kerää antureiden avulla ympäristöstä dataa, jonka avulla myöhemmin pystytään tietokoneella muodostamaan tunnelista 3D-mallinnus. (Kähkönen 2016)



Kuva 7. G/01 robottiin on mahdollista kiinnittää vedettäviä tai työnnettäviä työkaluita, esimerkiksi pyöriä lakaisuharjoja, katupesureita, lumiauroja ja -linkoja. (Kähkönen 2016)

2.2.7 TASKI IntelliBot -siivousrobotti

TASKI on hiljainen, kosketusnäytöllä varustettu siivousrobotti (kuva 8). Se on kooltaan noin metrin pituinen ja painaa 275 kiloa. Se toimii imurina ja alueiden lakaisijana. TASKI on varusteltu IntelliTrack -seurantajärjestelmällä, joka estää sen törmäämisen esteisiin ja myös sen katoamisen.

Robotin anturit mahdollistavat 360 asteen näkymän alueesta. Se pystyy kommunikoi-
maan älypuhelimien kanssa. Sillä on suuri 38-litrainen pölypussi, ja se pystyy työ-
kentelemään 3,5 tuntia yhtäjaksoisesti. Sitä käytetään kauppakeskuksissa, lentoken-
tillä, toimistoissa, hotelleissa ja sairaaloissa. (Finncleanin www-sivut n.d.)



Kuva 8. Siivousrobotti vapauttaa siivoojan tekemään muita töitä, kun robotti hoitaa
lattiapinnat. (Finncleanin www-sivut n.d.)

Myös erilaisia imuri- ja ruohonleikkurirobotteja on kehitetty kotikäyttöön helpotta-
maan ihmisten arkea.

2.2.8 Zora-robotti

Zora-robotti on belgialaisen yhtiön, QBMT, kehittämä autonominen hoivarobotti
(kuva 9). Se on ranskalaisen Softbank Roboticsin kehittämä NAO-humanoidirobotti
ja se tuli markkinoille vuonna 2008. Uusin versio on julkaistu vuonna 2014. Nimensä
robotti on saanut Zora-ohjelmasta. NAO on kooltaan pieni, 58-senttinen ja painoltaan
4,3 kg (Wikipedia www-sivut n.d.). Zora on ihastuttanut monia, ja sitä käytetäänkin
eniten palvelu- sekä terveydenhuoltosektoreilla. Sen HD-kamerat mahdollistavat ih-
misen tunnistamisen ja ultraäänianturit ympäristön havainnoinnin. Sen nivelet, moot-
torit ja ihmismäinen muoto mahdollistavat liikkumisen, jolloin se pystyy tanssittamaan
ja viihdyttämään asiakkaita ikään katsomatta. Mikrofonit mahdollistavat keskuste-
lemisen ja kuuntelemisen. Nao-robottia ohjataan Linux-pohjaisella NAOqi-käyttöjärjes-
telmällä. NAO voidaan kytkeä myös Choregraphen graafiseen ohjelmointiympäris-
töön ja siihen voidaan ohjelmoida useita erilaisia sovelluksia.

Zoraa voidaan käyttää oppimisvälineenä erilaisille ohjelmointikielille, hoivarobotiikan sovelluksena potilaiden kuntoutukseen sekä vanhusten ja lasten jumppauttamiseen ja seuralaiseksi.

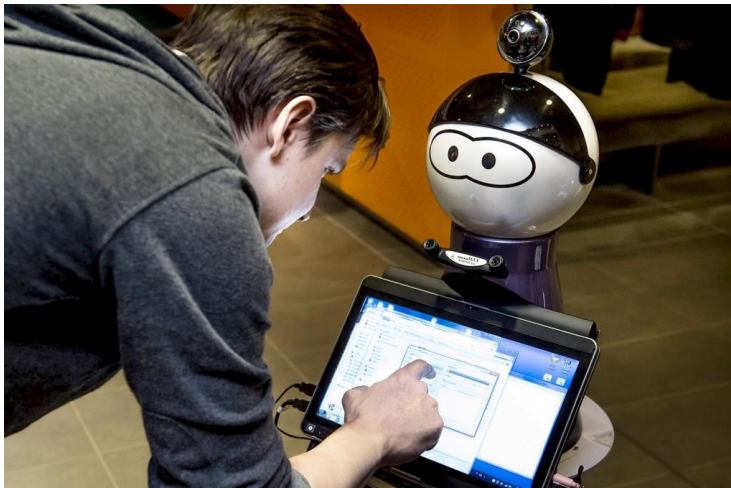


Kuva 9. Zora on ketterä ja sen liikkeet on helposti ohjelmoitavissa. (Messukeskuksen www-sivut 2016)

2.2.9 ”Kaveri”-robotti Robosoft Kompai

Ranskalaisen Robosoft -yrityksen kehittämä Kompai-palvelurobotti (kuva 10) tuli markkinoille vuonna 2010. Alun perin se on kehitetty viihdyttämään ja auttamaan vanhuksia heidän kodeissaan. Sillä on kuitenkin monia sovellusalueita: se tervehtii, juttelee ja opastaa. Robottia on käytetty myös peliteknologiateollisuudessa. Siihen on kehitetty erilaisia pelejä, joita lapset ja vanhukset voivat pelata. Robottia hyödynnetään myös terveydenhuollossa, sillä se voi nauhoittaa aktiviteetteja ja kerätä dataa potilaan terveydentilasta päiväkirjasovellukseen: esimerkiksi diabeteksen hoitoon liittyvät mittaukset, kuten verensokerin, hiilihydraattien ja insuliiniannosten määrät. Se muistuttaa potilaitaan lääkkeenotosta ja tapaamisista. Se ottaa yhteyttä lääkäriin, välittää etähoitoja ja pitää muistilistoja sekä lähettää sähköposteja. Robotti on ollut käytössä Suomessa Centria-ammattikorkeakoulussa ja Centria on myös ollut kehittämässä tokiolaisen Ochanomizun yliopiston kanssa tutkimusta robotin käytettävyydestä.

Erilaisia kehittämissideoita on runsaasti, kuten yksinkertaisen käsivarren integrointi esimerkiksi lääkerasian tai vesilasin poimimista varten. (Uus journalin www-sivut 2012)



Kuva 10. Suomea puhuva ”Kaveri”-robotti voi esimerkiksi muistuttaa palvelutalossa asuvaa vanhusta lääkkeenotosta ja tapaamisista. (Yli-Parkas 2015)

2.3 Palvelurobotiikan haasteena yhteiskunnallinen hyväksyttävyys

Yhteiskunnassa käydään kiihtyvää keskustelua robottien lisäämisen tarpeista palvelu- ja hoitoaloilla. Robottien yleistymisen terveydenhuollon sektorilla herättää niin paheksuntaa kuin hyväksyntääkin. Yleisesti voidaan kokea, että palvelurobotit olisivat uhka ihmistyön, hoivan, arviointikyvyn ja vuorovaikutuksen säilymiselle. Joidenkin näkemysten mukaan robotit veisivät ihmisten työpaikat ja näin ollen niiden lisääntymisen myötä ennustetaan korkeampaa työttömyyttä. Tutkimusten mukaan yli puolet suomalaisista ja jopa kolme viidesosaa eurooppalaisista ei sallisi robottien käyttöä lasten ja vanhusten hoidossa. Suomalaisista 84 % ei pitäisi ajatuksesta, että robotti hoitaisi omia lapsia tai omia ikääntyneitä vanhempiaan. Vain 7 % hyväksyisi robottihoitajan. (Kyrki 2015)

Kysymyksiä herättää etenkin robottien kyvykkyys samankaltaisiin vuorovaikutuksen, arviointikyvyn ja erilaisten tehtävien tekoon, joita ihminen hoitaa päivittäin potilaan kanssa.

Palvelurobotiikan haasteena ovat myös eettiset näkökulmat ja käyttöönotto: miten robotiikka sulautuu asiakkaiden ja työntekijöiden arkeen? Robotit koetaan enemmän hyväksyttävämpinä teollisuudessa, erityisesti raskaissa ja vaarallisissa töissä.

Kokemuksen ja tiedon puute, media ja muiden mielipiteet luovat ihmisille jo valmiiksi negatiivisia käsityksiä ja asenteita. Monet vierastavat ajatusta robotiikasta, sillä koneiden lisääntyminen ja nopea teknologian kehittyminen voivat tiputtaa ”monet junasta” ja luoda siten vastarintaa. Kuitenkin ihmiset tarvitsevat aikaa oppimiseen, perehdyttämiseen ja käyttöönoton suunnitteluun, sillä ei voida olettaa, että robotin käyttö työntekijän arjessa sujuisi heti täydellisesti. Tarvitaan siis enemmän pitkäaikaisia kokeiluja, jolloin saadaan tietoa robotin todellisista vaikutuksista ja hyödyistä arjessa. Esimerkiksi hoitoalalla on tehty lyhyitä kokeiluja käyttäjien parissa: Jotkut potilaista hurmaantuivat pienestä NAO-robotista ja tahtoivat jutella sen kanssa enemmän. Jotkut taas eivät pitäneet siitä, että robotti ohjasi jumpan eli käyttäjät kaipasivat aitoa ihmis-kontaktia. Hoitoalan henkilöstöllä on myös näkemyksiä robotin käyttöönotosta. Kuitenkin henkilöstön mielipiteet ja suhtautuminen ovat muuttuneet positiivisemmiksi käytön kokemusten myötä. (Stty:n www-sivut 2016)

Robotti on käyttäjien näkökulmasta koettava hyödylliseksi ja helppokäyttöiseksi. Helppokäyttöisyys auttaa teknologian vähäisiä käyttökokemuksia omaavien työntekijöiden oppimista ja suhtautumista asiaan. Robotin ulkonäöllä on myös oma merkityksensä, sillä se todennäköisesti koetaan miellyttävämpänä, jos ulkonäkö ei ole pelottava.

Työntekijöiden arjessa robotti tulisi nähdä myös työvälineenä ja siksi tulisi ennen kaikkea hyödyntää sen tuomia helpotuksia. Robotin ei ole tarkoitus olla ”ylimääräinen kapistus”, joka vaikeuttaa työtä. Robotin mukanaolo muuttaa työtapoja ja näiden tulosten esiin saaminen vaatii pitkäaikaisempaa kokeilua. Myös asiakkaiden ja potilaiden näkökulmasta on ajateltava, miten robotti nimenomaan tukisi ihmissuhteita ja vuorovaikutusta, eikä niinkään korvaisi näitä kokonaan. (Kyrki 2015)

Myös käyttäjien ja suunnittelijoiden välinen vuoropuhelu saattaisi lisätä hyväksyttävyyttä eli ei vain oleteta, että osallistetaan myös käyttäjiä. (Stty:n www-sivut 2016).

Esimerkiksi Meditas Oy:n henkilöstö koostuu sekä insinööreistä, sosiaalityöntekijöistä että sairaanhoitajista. Näin voidaan ottaa huomioon esimerkiksi sairaanhoitajien näkökulma siellä, missä palvelurobotteja erityisesti tarvitaan sairaanhoitajan työssä. Ja näin myös insinöörit suunnittelevat, minkälainen robotti kyseiseen työhön parhaiten soveltuisi.

Pitkäaikaisempien kokeilujen myötä saataisiin kokonaisvaltaisempaa ymmärrystä robotiikan merkityksestä ihmisen toiminnassa ja arjessa. Kokeilut ovatkin kasvamassa ja vähemmistö alkaa muuttua enemmistöksi kokeilujen suhteen. Siten robotit ovat enemmän pinnalla ja mukana yritysten arjessa. Kokeiluista uutisoidaan paljon, mikä herättää myös kiinnostusta, ja yhä useammat yritykset haluavat ottaa selvää palvelurobotin käyttömahdollisuuksista ja ottavat robotin koekäyttöön. Näin robottien käyttö voi huomaamatta sulautua osaksi arkipäivää, ja lopulta hitaammat omaksujat ehkä alkavat myös hyväksyä robotit.

3 PEPPER-HUMANOIDIROBOTTI

Pepper on 120-senttinen humanoidirobotti (kuva 11). Nykyinen SoftBank ja Aldebaran Robotics SAS on kehittänyt Pepper-robotin vuonna 2014. Pepper tuotiin markkinoille Euroopassa, Iso-Britanniassa vuonna 2016 (Wikipedia www-sivut n.d.). Pepper on roboteista ensimmäinen, joka pystyy määrittelemään tunteita, kuten iloa, surua ja vihaa. Pepper on kykeneväinen myös tulkitsemaan hymyn, otsan rypistelyn, äänen sävyt ja elehdinnät antureiden ja 3D-kameran avulla.

Pepper-robotin rakenne on ihmistä muistuttava, jotta se olisi helpommin lähestyttävä. Se näyttää ystävälliseltä ja moni suhtautuukin Pepper-robottiin positiivisesti sen miellyttävän ulkonäön vuoksi. Etenkin lapsille Pepper on mukava keskustelukumppani. Pepper tukee monia kieliä, kuten japania, englantia, ranskaa, italiaa, saksaa, espanjaa, kiinaa, hollantia ja arabiaa. Vain joissakin versioissa se tukee osaksi muita harvinaisempia kieliä, kuten suomea, ruotsia, venäjää, puolaa ja korean kieltä.

Pepper-robottia käytetään Japanissa eri yrityksissä toivottamaan vieraat tervetulleiksi, antamaan informaatiota ja hauskuuttamaan asiakkaita. Japanilainen pankkiyhtiö hyödyntää Pepper-robottia juuri edellä mainituissa tilanteissa. Pepper sopii myös kotikäyttöön, ja se toimiikin useissa japanilaisissa perheissä seuralaisena. Joulukuussa 2015 Nescafé palkkasi 1 000 Pepper-robottia auttamaan asiakkaita sopivan kahvikoneen löytämisessä. Ranskassa Pepper auttaa asiakkaita supermarket Carrefourissa löytämään haluamansa tuotteen. Kesäkuussa 2016 Belgiassa Pepper otettiin käyttöön sairaaloissa potilaiden kanssa kommunikoimiseen. Iso-Britanniassa Pepper-robottia käytetään toimistoissa vieraiden kasvontunnistukseen ja asiakkaiden kanssa keskusteleminen. (Reese 2016)



Kuva 11. Pepper Satakunnan ammattikorkeakoulussa. (Glad 2017)

3.1 Toimilaitteet

Seuraavaan taulukkoon (TAULUKKO 1) on merkitty Pepper-robotin toimilaitteiden tekniset tiedot.

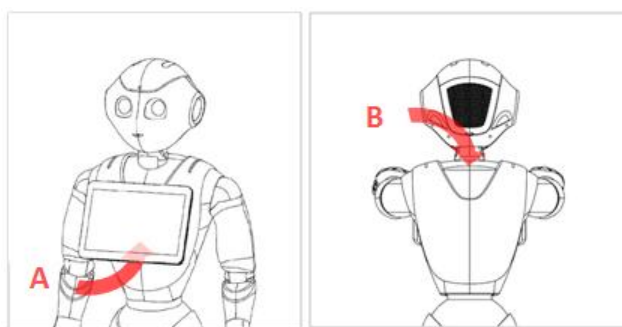
TAULUKKO 1. Pepper-robotin tekniset tiedot.

Mitat	1,210mm (korkeus) x 480mm (syvyys) x 425mm (leveys)
Paino	29kg
Akku	Litium-ioni akku Kapasiteetti: 30Ah/795Wh Käyttöaika: n. > 12 tuntia
Anturit	Pää: Mic x 4, RGB-kamera x 2, 3D-anturi, Kosketusanturi x 3 Rinta: Gyri anturi x 1 Kädet: Kosketusanturi x 2 ”Jalat”: Sonar anturi x 2, laser anturi x 6, Bumber anturi x 3, Gyro anturi x 1, infrapuna anturi x2
Liikkuvat osat	Liikkeen kulmat: Pää 2, Kädet: 2 x 2 (vasen/oikea), kyynärpää 2 x 2 (vasen/oikea), ranne 1 x 2 (vasen/oikea), käsi: 1 x 2 (vasen/oikea), rinta 2, polvi 1, pyörät 3, moottorit 20
Näyttö	10.1” kosketusnäyttö CPU: 1.3 GHz quad-core ARM Cortex-A7 Cache 512 KB L2 Wi-Fi, Bluetooth 1.6G pixel/sec @416MHz DDR3 SDRAM: 1GB Flashmemory: 32GB Typpi: IPS Resoluutio: 1280*800 Väri: 24bit true color
Alusta	NAOqi OS
Emolevy	Proessori: Atom E3845 CPU: Quad core Kellotaajuus: 1.91 Ghz RAM: 4GB DDR3 Flash memory/Micro SDHC: 8GB/ 16GB
Verkko	Wi-Fi: IEEE 802.11 a/b/g/n (2.4Ghz/5 Ghz) Ethernet portti x 1 (10/100/1000 base T)
Liikenopeus	max. 2 km/h

3.2 Pepper-robotin rakenne

Pepperin rakenne on ihmistä muistuttava. Pepper koostuu päästä, käsivarsista ja rungosta. Pepperin ”jalat” ovat osa runkoa ja ainoat liikkuvat osat ovat pää ja kädet. Pepper liikkuu myös keskirungosta, mutta se ei kierry niin kuin muut nivelet. Sormet ovat yhteinen kokonaisuus, jolloin sormet eivät liiku erikseen.

Pepper-robotin virtanappi (kuva 12) sijaitsee tabletin alla. Selässä, olkapäiden välissä sijaitsee Stop-nappula.

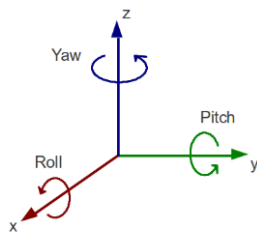


Kuva 12. Pepperin virtapainike ja Stop-painike. (Aldebaran www-sivut 2018)

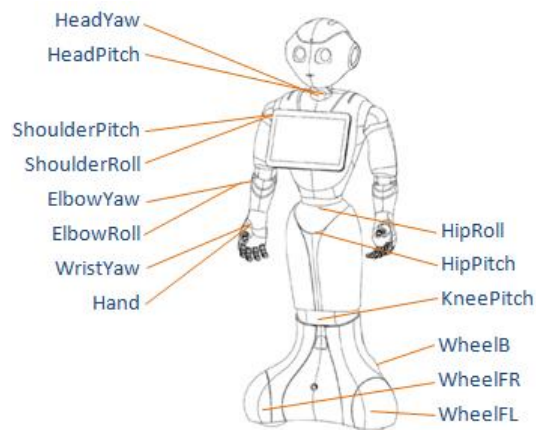
3.2.1 Moottorit ja nivelet

Kuvassa 14 nähdään moottorien sijainnit. Pepper-robotin pyörät antavat mahdollisuuden liikkua 360 asteen kulmassa, maksimissaan nopeudella 3 km/h. Pään, käsivarsien ja keskirungon 20 moottoria hallitsevat ja suorittavat Pepper-robotin liikkeitä. Moottorien tiedot ja tyypit löytyvät liitteistä 2.

Robotissa on yhdistetty kaksi runko-osaa. Kädet ja pää pyörivät nivelakselien ympäri. Runko-osien kiertoa varten sijoitetaan jokaiseen liitokseen origo. Kun robotti on nol-lapisteessä, kaikilla nivelillä on sama orientaatio. Rullauskierrat menevät X-akselin ympäri. Nivelten kaltevuuskulmat pyörähtävät Y-akselin ympäri ja kiertymä pystyakselilla Z-akselin ympäri (kuva 13). Liitteistä 3 ja 4 löytyy tarkemmat kulmat.



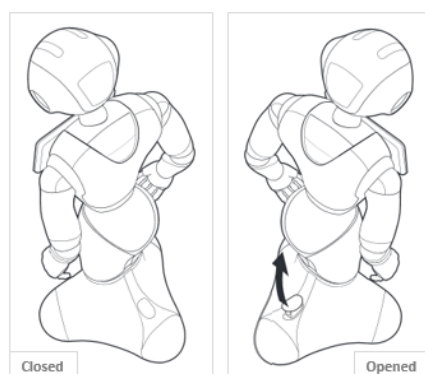
Kuva 13. Nivelten kiertymät.
(Aldebaran www-sivut 2018)



Kuva 14. Moottorien sijainti. (Aldebaran www-sivut 2018)

3.2.2 Hätäseis ja latausasema

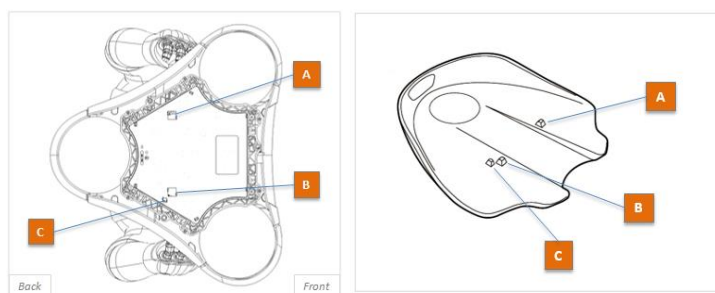
Pepper-robotin hätäseis-luukku sijaitsee jalkojen tuntumassa. Ks. kuva 15 alla.



Kuva 15. Hätäseis-painikkeen sijainti. (Aldebaran www-sivut 2018)

Virtaliittimet (kuva 16) sijaitsevat Pepper-robotin pohjassa. A on +napa ja B on –napa. C taas kuvaa varmistusta (security), jonka avulla varmistetaan, että Pepper on oikeassa kohdassa latausasemalla.

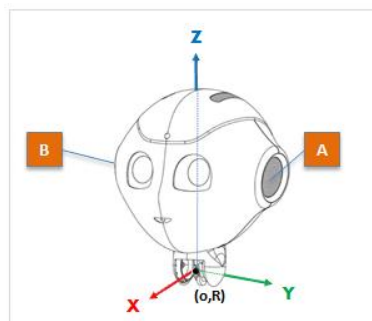
Pepper pystyy automaattisesti siirtymään latausasemaan, jos asema sijaitsee lähellä. Englanninkielisen perusohjelman ollessa käynnissä Pepper-robotille voidaan sanoa: ”Go to your station!”. Pepper tottelee ja sovittaa itsensä latausasemaan.



Kuva 16. Latausaseman liittimet. (Aldebaran www-sivut 2018)

3.2.3 Kaiuttimet ja mikrofonit

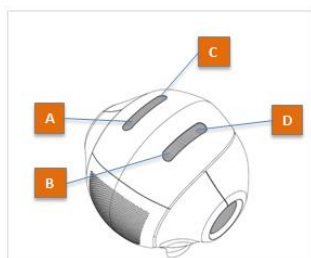
Kaiuttimet (kuva 17) sijaitsevat Pepper-robotin päässä. Kaiuttimia on kaksi kappaletta, ja niiden läpimitta on 25 mm. Kaiuttimien tarkkuus on 78db 1W/1m @ 1 kHz. Taajuusalue 70 Hz – 7.2 kHz. (Aldebaran www-sivut 2018)



Kuva 17. Vasen (A) ja oikea (B) kaiutinanturi. (Aldebaran www-sivut 2018)

Mikrofonit (kuva 18) sijaitsevat pään alueella. Mikrofoneja on neljä kappaletta. Niiden tarkkuus on 300mV/Pa +/- 3dB @ 1 kHz. Taajuusalue on 100Hz-10Khz. (Aldebaran www-sivut 2018)

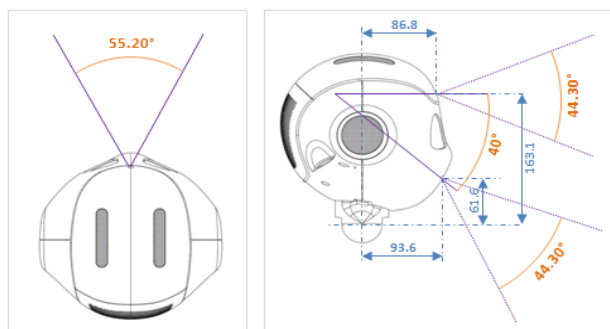
Neljäsuuntamikrofonit mahdollistavat tunteiden ja vuorovaikutuksen tulkinnan äänen sävyistä. Mikrofonit auttavat myös havaitsemaan, mistä äänet tulevat ja paikantavat niiden sijainnin. Niiden avulla Pepper myös analysoi äänenpainon ja sävyn, jolloin Pepper robotin on helpompi havainnoida tunteita.



Kuva 18. Vasen (A ja C) ja oikea (B ja D) mikrofonianturit. (Aldebaran www-sivut 2018)

3.2.4 2D- ja 3D-kamerat

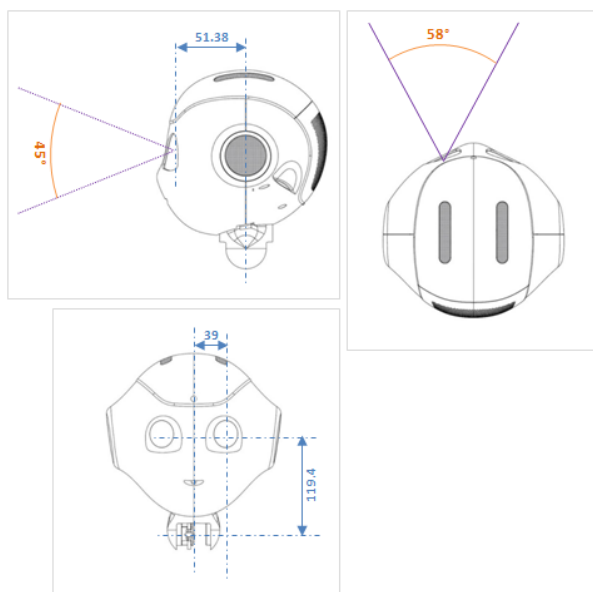
2D-videokamerat (kuva 19) sijaitsevat Pepper-robotin kasvoissa sekä otsan kohdalla ja suussa. 2D-kamera on mallia OV5640 ja tyypiltään SOC kuva-anturi. Kameroiden resoluutio on 2560*1920/kuvanottotajuus fps (frame per second). (Aldebaran www-sivut 2018)



Kuva 19. 2D-kameroiden kuvanottoalue. (Aldebaran www-sivut 2018)

3D-kamerat (kuva 20) sijaitsevat silmissä. Resoluutio 3D-kamerassa on 320x240 ja kuvanottoaajuus on 20 fps. Kamera on malliltaan ASUS Xtion ja tyypiltään SOC kuva-anturi. (Aldebaran www-sivut 2018)

Pepper pystyy toimimaan monimutkaisissa ympäristöissä 3D-kameran ja kahden HD-kameran ansiosta, jotka auttavat tunnistamaan kasvoja, niiden muotoja ja tunteet keskustelukumppanin kasvoilta. Kun käytetään tätä kamerateknologiaa, Pepper prosessoi muodot kuvina ja määrittelee sitä kautta objektin, jonka havaitsee. Kuvat käsitellään muodontunnistusohjelmistolla. Esimerkiksi jos keskustelukumppani on iloinen, Pepper havaitsee iloisuuden kasvopiirteistä ja eleistä ja reagoi iloisesti takaisin. Jos keskustelukumppani taas on surullinen, Pepper reagoi suruun lohduttavasti.



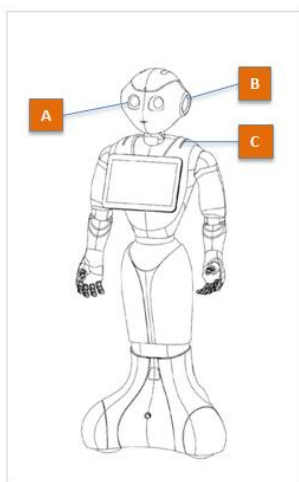
Kuva 20. 3D-kameran kuvanottoalue. (Aldebaran www-sivut 2018)

3.2.5 Ledit

Ledit (kuva 21) sijaitsevat silmissä, päässä ja olkapäillä. Silmissä on kahdeksan lediä per silmä ja ledit ovat RGB-LED:ejä. (Aldebaran www-sivut 2018)

Näin Pepper-robotin silmien värit muodostuvat punaisen, vihreän ja sinisen värisistä valolähteistä. Esimerkiksi visuaalinen palaute saadaan Pepperin tapauksessa silmien ledien väreistä ja ledivalon paikasta.

Korvien alueella ledejä on 10 kappaletta per korva. Ne ovat tyypiltään 16 Blue levels eli ledit näyttävät sinistä väriä. Käsivarsissa sijaitsevia ledejä on kaksi kappaletta. Ne ovat myös RGB-ledejä.

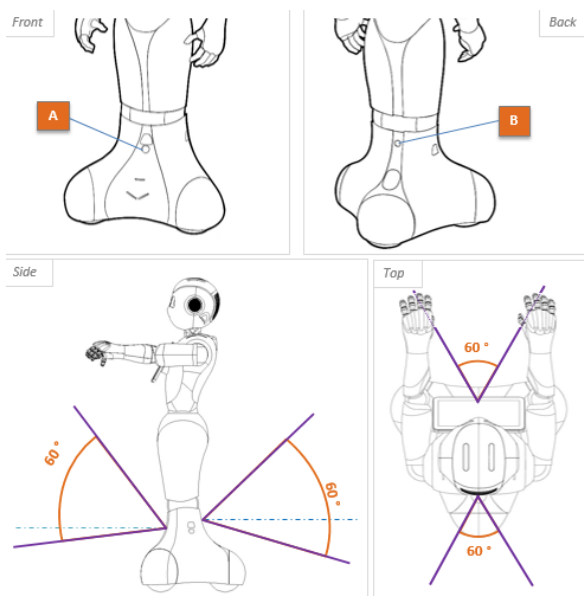


Kuva 21. Ledien sijainnit. (Aldebaran www-sivut 2018)

3.2.6 Anturit

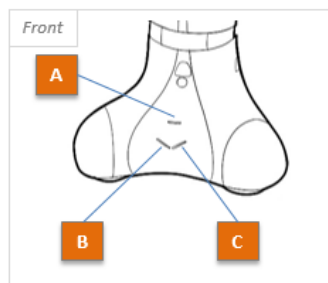
Pepper-robotilla on lukuisia antureita: kaksi ultraäänilähetintä ja -vastaanotinta, kuusilasertunnistinta ja kolme este-ilmaisinta (Aldebaran www-sivut 2018). Nämä anturit antavat sille tietoa lähistöllä olevien kohteiden (kolmen metrin etäisyydeltä) paikasta. Akun toinen anturi ilmaisee sen lataustason sekä lämpötilan. Pepper-robotilla on tunnistimia, joita käytetään, kun Pepper-robotilla pelataan pelejä tai käydään keskusteluja.

Laser-anturit sijaitsevat jalkojen alueella (kuva 22). Niitä on yhteensä kuusi. Kolme ensimmäistä sijaitsevat robotin etupuolella havainnoiden edessä olevaa aluetta. Aallonpituus on 808 nm ja kuvataajuus on 6.25Hz per laser. (Aldebaran www-sivut 2018)



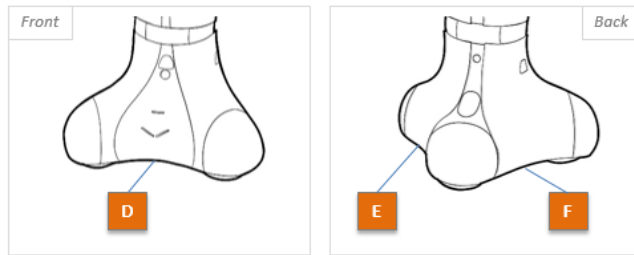
Kuva 22. Antureiden sijainti. (Aldebaran www-sivut 2018)

Laserin ohjaimet (kuva 23) sijaitsevat jalkojen edessä ja alapuolella. B rajaa vasemman alueen robotista katsoen ja C oikean puolen.



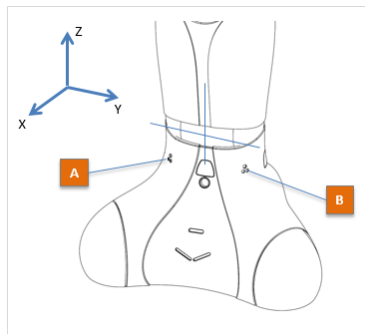
Kuva 23. Laser-anturit arvioivat edessä sijaitsevat esineet. (Aldebaran www-sivut 2018)

Laserin ohjaimet (kuva 24), jotka havainnoivat ympäröivää ympäristöä, sijaitsevat Pepper-robotin pohjassa. Kuva ohjaimien paikoista ja tekniset tiedot löytyvät seuraavasta.



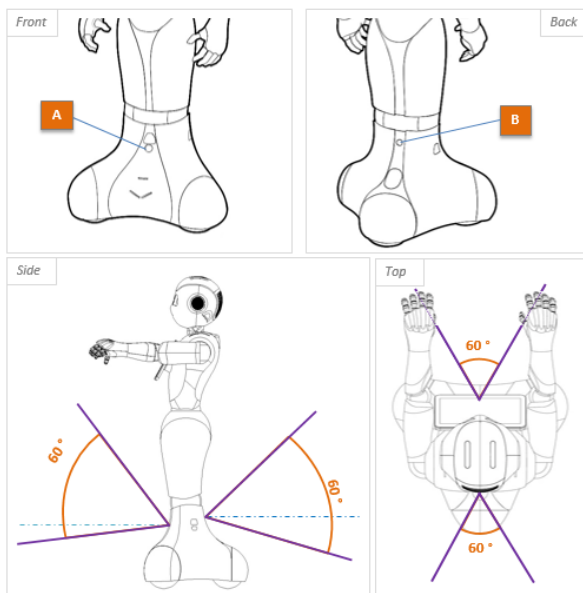
Kuva 24. Laserit, jotka havainnoivat Pepperin ympäröivää aluetta. (Aldebaran www-sivut 2018)

Pepper on myös varustettu kahdella infrapuna-anturilla (kuva 25). Ne sijaitsevat jalkojen etuosassa.



Kuva 25. Infrapuna-anturit mittaavat etäisyyttä ja auttavat seuraamaan tiettyä linjaa. (Aldebaran www-sivut 2018)

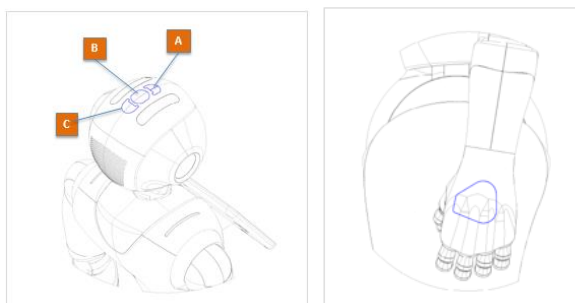
Pepper-robotin jalkojen etuosassa sijaitsevat myös ääniluotaimet (kuva 26), jotka arvioivat esteiden sijaintia Pepperin edessä. Taajuus on 42 kHz. Tarkkuus -86dB. Resoluutio 0.03m. (Aldebaran www-sivut 2018)



Kuva 26. Ultraääniantureiden sijainnit. (Aldebaran www-sivut 2018)

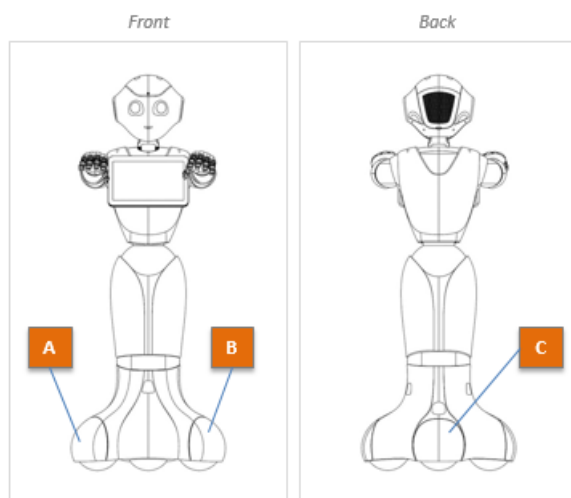
Pepper sisältää myös 29 kappaletta MRE- eli digitaalisia asentoantureita, jotka mittaavat pyörimistä magneettikentässä. Niiden tarkkuus on 12 bittiä, ts. 4096 arvoa per kääntö, 0.1° tarkkuudella. (Aldebaran www-sivut 2018)

Kontakti- ja tuntoanturit (kuva 27) sijaitsevat Pepper-robotin käsissä ja päässä. Esimerkiksi päätä painamalla voidaan resetoitava ohjelma. Jos halutaan opettaa liikkeitä animointitilassa, tulee käsivarressa sijaitsevien kosketusanturien olla aktiivisena. Pepper-robotin kädestä on pidettävä kiinni siihen asti, kunnes haluttu asento on asetettu, ja kun kädestä päästetään irti, lukkiutuvat raajat haluttuun asentoon.



Kuva 27. Pään kosketusantureiden sekä käden kosketusanturin sijainti. (Aldebaran www-sivut 2018)

Pepper-robotilla on myös puskurit (kuva 28), jotka suojaavat sitä törmäyksiltä. Ne sijaitsevat jalkojen alueella seuraavan kuvan mukaisesti.



Kuva 28. Puskureiden sijainti. (Aldebaran www-sivut 2018)

Pepper tunnistaa ihmiset ja esteet törmäyksenestojärjestelmänsä ansiosta. Pepper pysyy myös säilyttämään tasapainonsa, mikä estää sitä kaatumasta, jos joku tönäisee robotia.

3.3 Pepper-robotin kanssa toimiminen

3.3.1 Pepper-robotin käynnistäminen

Jos Pepper on latausasemassa, se voidaan käynnistää tabletin alla sijaitsevasta napista. Jos Pepper ei ole latausasemassa, on varmistettava, että tilaa riittää käynnistymiseen. Pepper-robotin ledit vilkkuvat ensin valkoisesta siniseksi. Ledit muuttuvat sitten siniseksi, nopeasti punaiseksi ja lopuksi vihreäksi. Ledit ovat valkoiset, kun käynnistys on onnistunut. Pepper pitää äänen: ”OGNAK GNOUK” ja nousee ylös. Pepper pyörittää muutaman kerran puolelta toiselle ja sen jälkeen se on valmis ohjelmoitavaksi.

3.3.2 Pepper-robotin kanssa kommunikointi

Autonomous life on -tilan on oltava päällä, jotta keskustelu onnistuu. Pepper-robotilla on kolme aluetta (zone), joiden mukaan se havainnoi ympäristöään. Kun Pepper on yhdistetty ohjelmointiympäristöön, voidaan asetuksista säätää alueen sädettä. Ensimmäisen alueen säde on noin 1,5 metriä.

Pepper seuraa ääniä, kasvoja ja liikettä. Pepper pyrkii pitämään mahdollisimman pitkään katsekontaktin. Pepper ei reagoi uuteen ihmiseen ennen kuin on lopettanut prosessin edellisen kanssa tai dialogi/toiminto on vielä kesken/käynnissä. Pepper kuuntelee, jos silmien ledit ovat siniset. Silmien ledit muuttuvat vihreäksi, kun Pepper lakkaa kuuntelemasta ja prosessoi juuri kuulemaansa. Pepper pitää "bilip"-äänen, kun se alkaa kuunnella, ja saman äänen uudestaan, kun se lopettaa kuuntelemisen.

Kun halutaan suorittaa jokin ohjelma tai kun ohjelma on käynnissä, se voidaan keskeyttää. Toiminnon voi keskeyttää, kun kämmentä pidetään Pepper-robotin pään päällä (kuva 29) ja kämmen peittää kameran. Sekunnin kuluttua kuuluu "blop" ja Pepper-robotin ledit muuttuvat vihreäksi. Käden voi ottaa pois, kun toiminto on pysähtynyt.

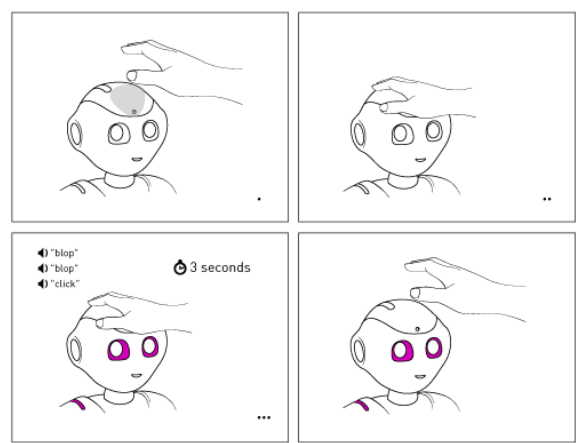


Kuva 29. Ohjelma pysäytetään asettamalla käsi pään kameran osaan ja sormet pään kosketusantureille. (Aldebaran www-sivut 2018)

3.3.3 Sleep mode - “Nukkumistila”

Kun Pepper “nukkuu”, sen ei ole mahdollista suorittaa mitään toimintoa. Pepper-robotin voi tällöin kuitenkin päivittää. Kun Pepperin jättää nukkumaan yöksi, se nopeuttaa päivitysten tekemistä. Nukkumistilassa Pepper tarkistaa, onko päivityksiä saatavilla ja se päivittyy klo 2.00–3.00 aikaan yöllä. Muussa tapauksessa päivitykset pitää hyväksyä ennen päivittämistä ja vie aikaa, ennen kuin Pepper-robotin voi ottaa taas käyttöön.

Pepper-robotin autonomous life on -tila vaaditaan, jotta Pepperin voi laittaa nukkumaan. Pepper laitetaan nukkumaan pitämällä kättä Pepperin pään päällä noin kolmen sekunnin ajan (kuva 30). Käden pitää peittää otsassa sijaitseva kamera ja pään päällä sijaitsevat anturit. Pepper pitää äänen “blop”, “blop”, “click” ja silmien ledit muuttuvat violetiksi. Kun käsi otetaan pois pään päältä, Pepper nukahtaa kolmen sekunnin kuluessa. Pepper-robotin moottorit vapautuvat ja Pepper asettuu lepoasentoon. Kun käden asettaa pään päällä sijaitsevan anturin päälle, Pepper herää.



Kuva 30. Pepperin asettaminen Sleep-tilaan. (Aldebaran www-sivut 2018)

3.3.4 Pepper-robotin autonomous life on- ja off -tila

Pepper-robotin autonomous life -tilan saa päälle painamalla kaksi kertaa rinnassa sijaitsevasta painikkeesta. Kun autonomous life -tila laitetaan päälle, Pepper pitää iloisen äänen ja Pepperin moottorit menevät päälle.

Sitten Pepper ”nousee” seisoma-asentoon. Kun autonomous life -tila laitetaan pois päältä, painetaan kaksi kertaa rintapainikkeesta. Pepper huokaisee, menee lepoasentoon ja sammuttaa moottorinsa.

3.3.5 Pepperin sammuttaminen

Pepper-robotin saa sammutettua painamalla noin kolme sekuntia tabletin alla sijaitsevasta napista. Pepper sanoo ”GNUK GNUK”, ”nukahtaa” ja moottorit vapautuvat. Ledit vilkkuvat valkoisina. Ledit lakkaavat vilkkumasta, kun sammutus on suoritettu onnistuneesti ja Pepper on lepoasennossa.

3.3.6 Pepper-robotin ilmoitukset

Kun Pepper-robotilla on ilmoitus, se pitää pienen äänen ja olkapäistä sijaitsevista ledeistä nähdään, mistä on kyse. Vihreä väri merkitsee palautetta onnistuneesta toiminnasta. Keltainen väri tarkoittaa varoitusta ja se vaatii korjausta. Punainen väri tarkoittaa häiriötä eli toiminnan suorittaminen on epäonnistunut tai on ilmennyt virhe, eikä toimintaa ole mahdollista suorittaa.

Kun Pepper-robotin tabletin alla olevaa painiketta painaa kerran, Pepper kertoo nimensä ja IP-osoitteensa. Jos olkapäissä sijaitsevat ledit vilkkuvat, Pepper kertoo ilmoituksen. Jos painat vielä kerran rintapainikkeesta, se ohittaa viestin.

Jos Pepper sanoo IP-osoitteen liian nopeasti tai epäselvästi, voidaan painaa vielä kerran, jolloin se lausuu sen hitaasti numero numerolta.

4 CHOREGRAPHE-OHJELMOINTIYMPÄRISTÖ

4.1 Ohjelmointiympäristön esittely

Choregraphe on monipuolinen ohjelmointiympäristö sovellusten toteuttamiseen. Choregraphe on pääsääntöisesti tarkoitettu robottien ohjelmointiin ja ohjelma voidaan yhdistää omaan robottiin IP-osoitteen avulla. Virtuaalirobotti on ohjelmaan asennettuna jo valmiiksi. Choregraphella voidaan ohjelmoida erilaisia robotin toiminnallisuuksia ja monitoroida niitä. Choregraphella voidaan myös luoda animaatioita, toimintoja ja dialogeja. Niitä voidaan testata ja simuloida sekä virtuaalirobotilla että fyysisellä robotilla. Choregraphella voidaan luoda erilaisia sovelluksia, kuten dialogeja, tanssia ja palveluita.

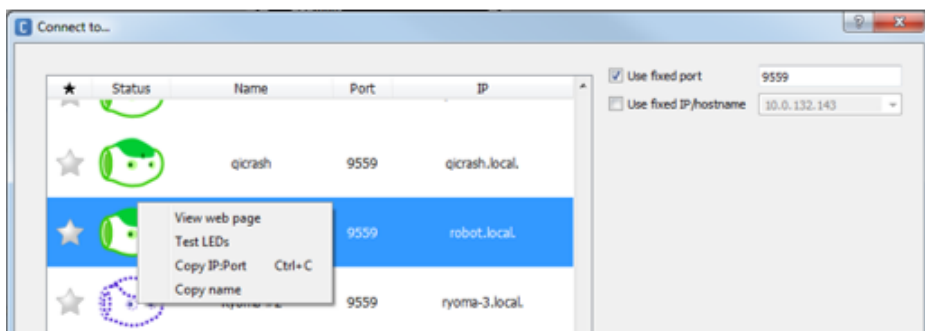
Choregraphe käyttää Python ja C++-ohjelmointikieltä. Erilaisia toiminnallisuuksia voidaan ohjelmoida käyttämällä graafista ohjelmointia tai kirjoittamalla esimerkiksi Python-koodia. Ohjelman voi ladata koneelle helposti Choregraphen sivuilta löytyvien ohjeiden mukaan. Ohjelma tukee Linuxia, Windows 7 ja 8.1 ja Mac Os 10.11 El Capitania. Choregraphe on linkitetty esimerkiksi NAOqiin ja siten sillä on mahdollista ohjelmoida NAO-robotia. (Aldebaran www-sivut 2018)

4.2 Yhdistäminen robottiin

Choregraphe voidaan yhdistää fyysiseen robottiin Connection-palkin alta Connect to -valikon alta tai painamalla vihreätä verkon kuvaa (kuva 31) vasemmassa ylälaidassa. Sen jälkeen valitaan robotti, johon yhdistetään.



Kuva 31. Vihreä painike Choregraphe-sivun ylälaidassa.



Kuva 32. Fyysisen robotin IP-osoitteen syöttäminen (Aldebaran www-sivut 2018)

Jos robottia ei löydy listasta, kirjoitetaan sen IP-osoite ja portti oikeassa laidassa sijaitseviin kenttiin ja painetaan Select (kuva 32). Jos ohjelma on jo yhdistetty toiseen fyysiseseen robottiin, ohjelma varmistaa uuden yhteyden luomisen ja edellisen katkaisemisen. Tämän jälkeen painetaan Ask For Connection. Kun yhteys on luotu, otsikkopalkissa nähdään yhdistetyn robotin nimi ja robottinäkyvässä nähdään robotti 3D-kuvana.

Kun yhdistetään virtuaalirobottiin, valitaan Connection ja valikosta Connect to tai painamalla vihreää verkon kuvaketta vasemmalta ylälaidasta. Jos Virtuaalirobotti löytyy listalta, kaksoisklikataan robottia. Jos robottia ei löydy listasta, kirjoitetaan IP-osoite kenttään 127.0.0.1 kortiksi 9559 ja painetaan Select.

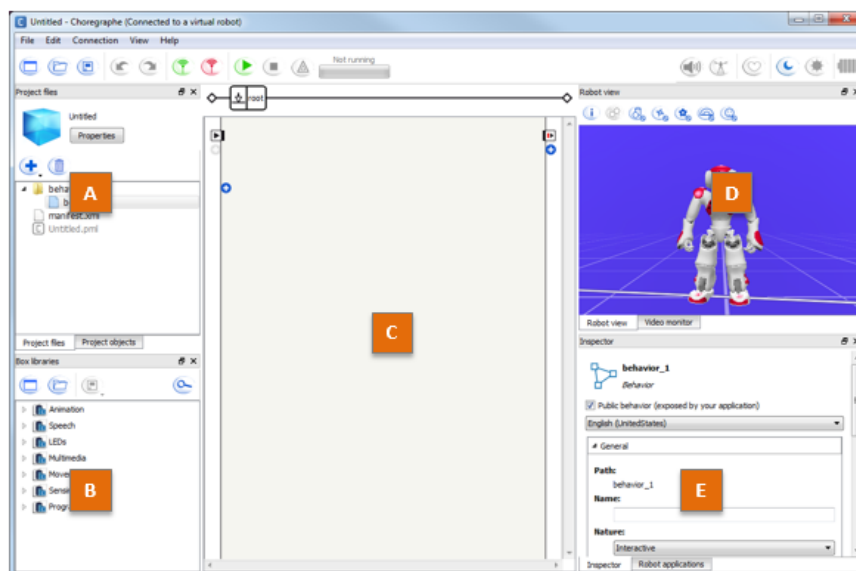
4.3 Pääkomponentit

Choregraphen aloitusnäyttö on hyvin yksinkertainen. Käyttöliittymä sisältää valikkopalkin, työkalupalkin ja eri ”aihioita”. Eri aihiot sisältävät erilaisia toimintoja robotin ohjelmointiin liittyen. Nämä ”aihiot” on jaettu viiteen alueeseen (kuva 33). Ensimmäinen alue (A) on projektin tiedoston aihio, mistä löytyvät kaikki projektin kohteet ja niiden asetukset. Asetuksista voidaan muokata projektin näkymää ja vaihtaa kieltä.

Toinen alue (B) kuvaa toimintalaatikoita, ja ne sisältävät eri toiminnallisuuksia. Eri otsakkeiden alta löytyvät eri toiminnallisuudet, kuten esimerkiksi Speech-otsakkeen alta löytyy Say-laatikko.

Kolmas alue (C) on ns. kulkukaavio, jossa on toimintalaatikoita linkitettyinä toisiinsa input/output-välityksellä. Alueelle tiputetaan raahaamalla näitä toimintalaatikoita, jotka löytyvät box libraries alta.

Seuraava alue (D) kuvaa robotin näkymää. Yhdistetty robotti esitetään 3D-kuvana. Viimeinen alue kuvaa robotin tarkastelu- ja sovellusasetuksia. Tarkasteluasetuksista näkyy valitun tai aktiivisen kohteen asetukset. Esimerkiksi kun Animated Say -laatikko on valittuna, oikealla nähdään sen asetukset, josta ominaisuuksia voi muuttaa. Sovellusasetukset (E) taas näyttävät käytettävissä olevat sovellukset yhdistetylle robotille.



Kuva 33. Ohjelman käyttöliittymä ja eri aihiot. (Aldebaran www-sivut 2018)

4.3.1 Pääpalkki

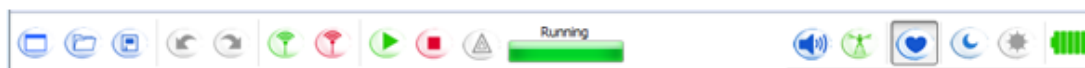
Pääpalkissa vasemmalla ylälaudassa on File, Edit, Connection, View ja Help. File-valikon alta voidaan luoda uusi, avata jo olemassa oleva sekä tallentaa nykyinen projekti. File-valikon alta voidaan myös muokata projektin asetuksia ja siirtää nykyinen projekti toiseen sovellukseen esimerkiksi CRG-tiedostoksi. Useat ohjelmat tukevat tätä tiedostomuotoa, jolloin projekti voidaan avata erilaisilla ohjelmilla.

Edit-valikon alta voidaan palata taaksepäin tai eteenpäin viimeisissä ohjelman liikkeissä. Edit-valikon alta voidaan myös muuttaa Choregraphen käyttöliittymän asetuksia.

Connection-valikon alta voidaan muodostaa/katkaista yhteys fyysiseen tai virtuaalirobottiin. Kun yhteys on luotu, voidaan suorittaa tai pysäyttää toiminto, avata tai piilottaa lokitiedot häiriöiden sattuessa, siirtää tiedostoja tietokoneen ja robotin välillä ja päivittää robottijärjestelmän uuteen versioon (in Connection-> Advanced).

View-valikon alta voidaan hallita näkyvillä olevia ”aihioita”. Help-valikon alta voidaan nähdä tilastot nykyisistä toiminnallisuuksista sekä päästään yleisiin Choregraphen dokumentteihin, josta löytyy informaatiota ongelmatilanteisiin.

4.3.2 Työkalupalkki



Kuva 34. Yläpalkin eri painikkeet. (Aldebaran www-sivut 2018)

Ensimmäiset kolme kuvaketta ovat Create/Open/Save Project (kuva 34). Nuolista pääsee taakse- ja eteenpäin viimeisistä tehdyistä muutoksista. Verkon kuvakkeet kuvaavat yhteyttä eli Connect ja Disconnect. Seuraavista painikkeista, Play- ja Stop-painikkeet, voi käynnistää ja pysäyttää ohjelman. Seuraava painike näyttää varoitukset ja häiriöt. Kun painetaan play-painiketta, prosessipalkki näyttää, missä tilassa ohjelma on. Harmaalla pohjalla nähdään, ettei ohjelmaa ole ladattu. Vihreällä pohjalla nähdään, että ohjelma on ladattu. Seuraavasta painikkeesta voidaan säätää robotin puhetta.

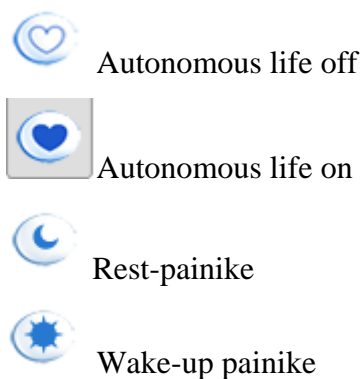
Vihreä, ”ihmisen” tapainen painike, kuvaa Animointi-tilaa (kuva 35). Painikkeen ollessa vihreä animation mode ei ole aktiivinen, jolloin ei voida animoida. Kun painike on oranssi, animointitila on lataamassa tai purkautumassa. Kun painike on punainen, animointitila on aktiivinen.

Animointitilaa käytetään esimerkiksi timeline-boksin kanssa, jotta voidaan animoida liikkeitä. Animointi on mahdollista vain Autonomous life -tilan ollessa pois päältä.



Kuva 35. Animation mode-painikkeen tilat. (Aldebaran www-sivut 2018)

Autonomous Life -tilan saa pois päältä painamalla robotin rinnassa olevaa on/off-nappia kaksi kertaa tai painamalla sydämen painiketta (kuva 36). Robotti voidaan herättää tähden muotoisesta Wake up-painikkeesta.

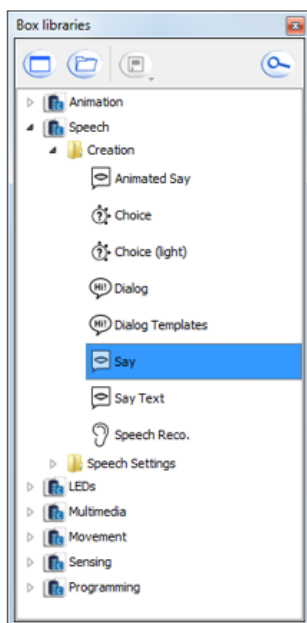


Kuva 36. Autonomouslife-tilat sekä Pepper-robotin nukahtamiseen ja herättämiseen tarvittavat painikkeet. (Aldebaran www-sivut 2018)

Kuun näköinen Rest-painike (kuva 36) kuvaa lepotilaa. Tällöin moottorit ovat vapautuneet. Robotti menee kyykky-asentoon ja lepotilaan. Akun kuva kertoo yhdistetyn robotin akun lataustason. Vihreänä akku on täynnä. Oranssi väri kertoo akun kulumisesta ja jäljellä on tiettyjä prosentteja. Punainen väri kertoo akun latauksen olevan hyvin alhainen, ja robotti sammuu muutamien minuuttien kuluttua, jollei sitä kytketä lataukseen.

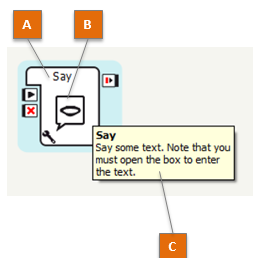
4.3.3 Box libraries

Yläpalkissa näytön kuvakkeesta voidaan luoda uusi toimintoboksi (kuva 37). Kansion kuvakkeesta voidaan avata jo voimassa oleva toimintoboksi. Kansio-kuvakkeen vierestä löytyy tallennuspainike sekä etsintätyökalu toimintolaatikoille.



Kuva 37. Toimintolaatikoiden lista speech-kansion alta.

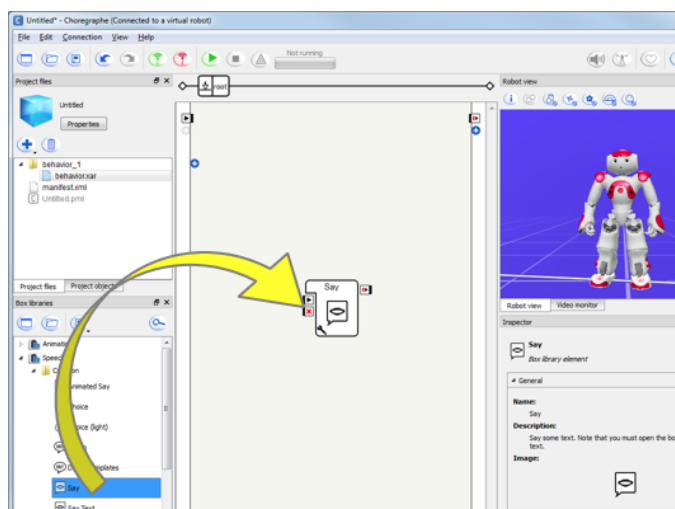
Laatikkokirjasto sisältää listan erilaisista toimintobokseista. Jokainen laatikko voi sisältää erilaisia tiedostotyyppejä (musiikkia, kuvia jne.) Kun valitaan laatikko, laatikon informaatio näkyy oikealla inspector-aihiossa.



Kuva 38. Esimerkki boksen alueista. (Aldebaran www-sivut 2018)

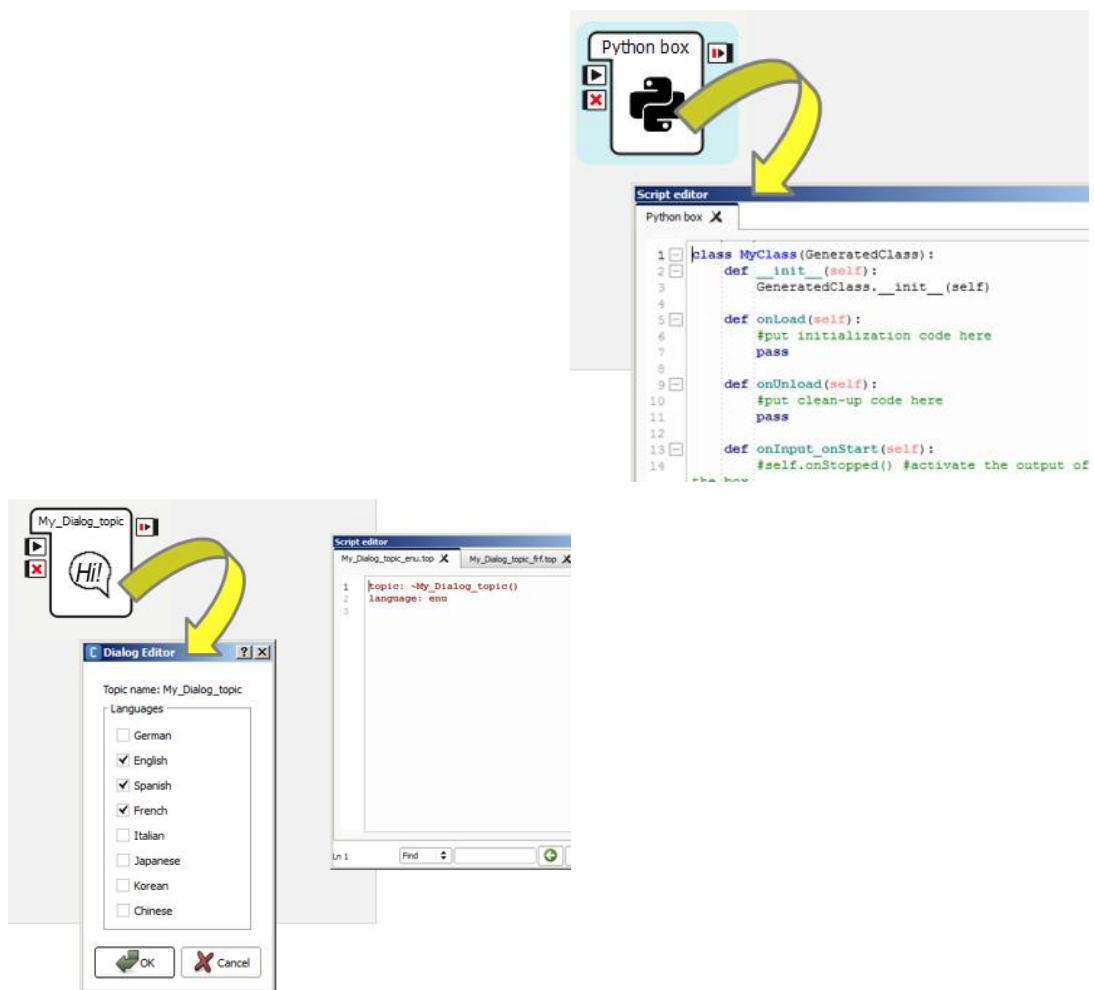
Boksi (kuva 38) koostuu eri alueista. A kuvaa boksen nimeä, josta nähdään, mihin toimintoon viitataan. B on kuva toiminnosta, mikä havainnollistaa toimintoa. C on kuvaus laatikosta ja sisältää vinkkejä sen käyttöä varten. Boksen vasen nuoli kuvaa inputtia, johon se vastaanottaa käskyjä. Nuoli oikealla kuvaa outputtia, joka lähettää datan eteenpäin, kun boksen toiminnot on suoritettu.

Laatikon voi vierittää kulkukaavioon (kuva 39). Oikeaa hiirtä painamalla ja valitsemalla Set parameters voidaan laatikolle syöttää parametrejä.

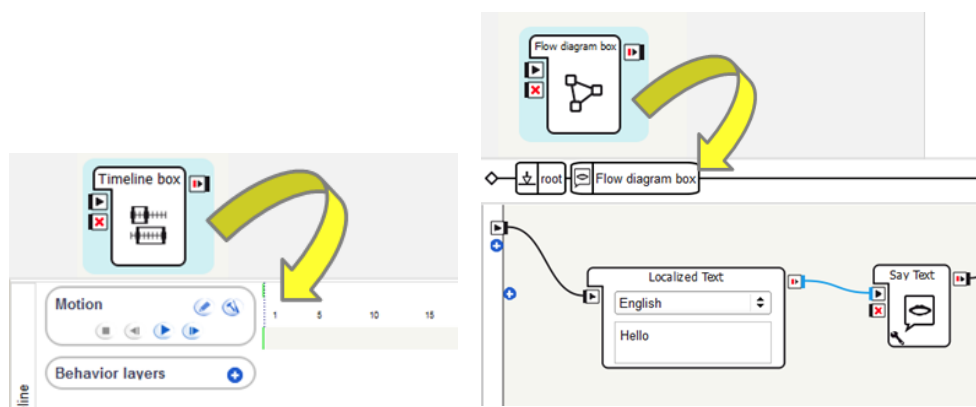


Kuva 39. Laatikon vierittäminen kulkukaavioon. (Aldebaran www-sivut 2018)

Bokseja on erilaisia. Esimerkiksi Python-boksi (kuva 40) sisältää komentorivin, johon voidaan kirjoittaa koodikielellä haluttu toiminto. Flow diagram -laatikkoa käytetään, kun halutaan tiivistää muita toimintobokseja yhdeksi toiminnoksi. Flow diagram -laatikko voi sisältää monta boksia, jotka on yhdistetty. Dialog-boksi taas mahdollistaa dialogin luomisen (kuva 40).

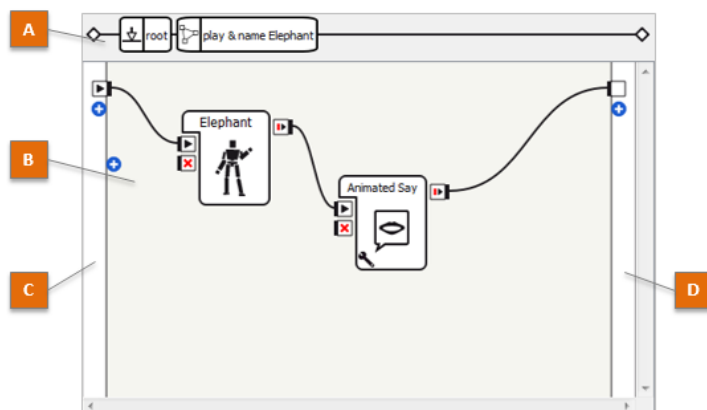


Kuva 40. Python- ja Dialog-boksin sisältö. (Aldebaran www-sivut 2018)



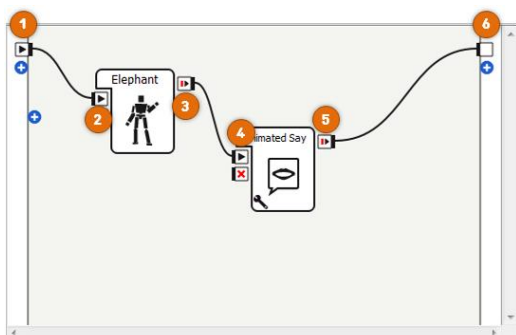
Kuva 41. Timeline- ja Flow Diagram boksin sisältö. (Aldebaran www-sivut 2018)

Timeline-laatikko (kuva 41) sisältää aikajanan eli voidaan tehdä erilaisia toiminnallisuksia tietyssä ajassa, ja aikajanaa on helppo hyödyntää esimerkiksi liikkeiden opettamisessa. Timeline-boksiin voidaan lisätä useita ominaisuuksia lisäämällä kerroksia (Behaviour layers). Flow diagram on ryhmä toimintobokseja, jotka on linkitetty toisiinsa. Voidaan käyttää eri toimintoja, lisätä kuvia, puhetta, musiikkia ja animoida robotia samanaikaisesti.



Kuva 42. Flow diagram -boksen kulkukaavioesimerkki. (Aldebaran www-sivut 2018)

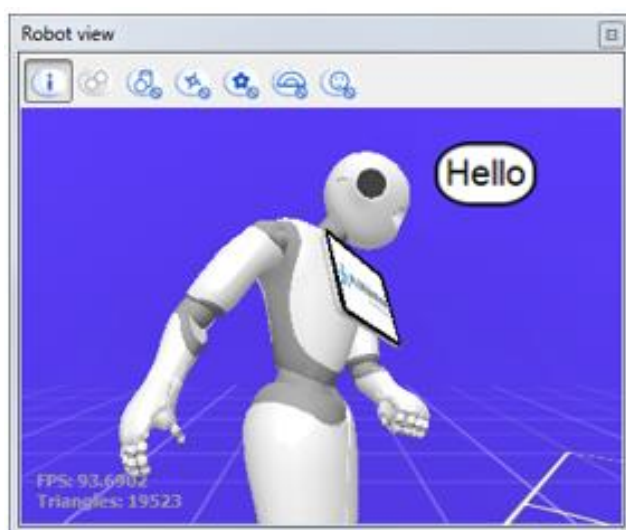
Kuvassa 42 A kuvaa boksen polkua, ja tästä nähdään eri flow diagramin tasot, jos boksi sisältää muita bokseja. B kuvaa kulkukaaviota, johon voidaan luoda, lisätä, poistaa tai muokata eri bokseja. C kuvaa input-rajapintaa, jossa voidaan lisätä tai poistaa tuloja. D kuvaa output-rajapintaa. Flow diagram voidaan luoda yhtenä laatikkona tai se voidaan sisällyttää toiseen laatikkoon.



Kuva 43. Esitellään boksien yhdistäminen toisiinsa. (Aldebaran www-sivut 2018)

Kuvassa 43 Flow diagrammin OnStart input (1) rajapinnasta yhdistetään Elephant Localized Text boksen (2) onStart inputtiin. Elephant boksen onStopped output (3) linkitetään Animated Say onSart inputtiin (4). Näin ohjelma jatkaa kulkuaan. Animated Say boksen OnStopped output (5) on yhdistetty Flow diagram OnStopped outputtiin (6), jolloin ohjelma on suoritettu.

4.3.4 Robotin näkymä



Kuva 44. Pepper-robotin näkymä. (Aldebaran www-sivut 2018)

Robotin näkymässä voidaan tarkastella ja muuttaa robotin nivelien asentoja. Siellä voidaan myös liikuttaa robottia. Robotin näkymästä nähdään myös lauseet, jotka robotti sanoo tai kuulee. Näkymä näyttää myös objektit ja alueet, joissa se havainnoi ympäristöä. Robottia voidaan navigoida 3D-kuvassa käyttämällä hiirtä. Vasenta hiirenpainiketta painamalla ja vetämällä saadaan liikutettua eteen, taakse, vasemmalle tai oikealle. Hiiren rullaa käyttämällä saadaan zoomattua lähemmäs ja kauemmas. Oikeaa hiiren painiketta käyttämällä saadaan kierrettyä kulmaa.

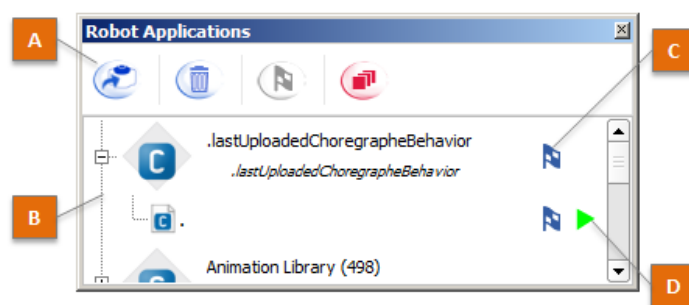
Kuvassa 44 yläkulmassa sijaitsevasta i-painikkeesta voidaan piilottaa sekä näyttää FPS-(Kuvanottotaajuus) ja Triangles-arvot. Seuraava painike oikealla kuvaa törmäyspainiketta, josta voidaan piilottaa tai näyttää törmäyksen tiedot. Sylinterin kuvakkeesta voidaan piilottaa tai näyttää muistiin varastoidut tiedot.

Sylinterin jälkeisestä painikkeesta voidaan estää tai kytkeä päälle robotin asennon seuranta. ”Mutterin” näköisestä painikkeesta voidaan estää tai kytkeä päälle moottorien arvot. Kaaren näköisestä painikkeesta voidaan näyttää tai piilottaa alueet, jotka on määriteltä. Robotin näkymässä nähdään myös ihmiset, jotka robotti havaitsee määritellyllä alueella. Hymynaama-painikkeesta voidaan piilottaa tai näyttää ihmiset, jotka ovat robotin alueella.

4.3.5 Robotin sovellukset

Robotin sovellukset voivat olla valmiita ohjelmia, jotka sisältävät useita toimintoja. Esimerkiksi ohjelma, jossa Pepper-robotti tanssii, laulaa ja keskustelee, voi olla tallennettu sovelluksena. Ohjelmoitavat ohjelmat voi tallentaa sovelluksena tai valmiita ohjelmia voi ostaa ja ladata sovelluskirjastoon.

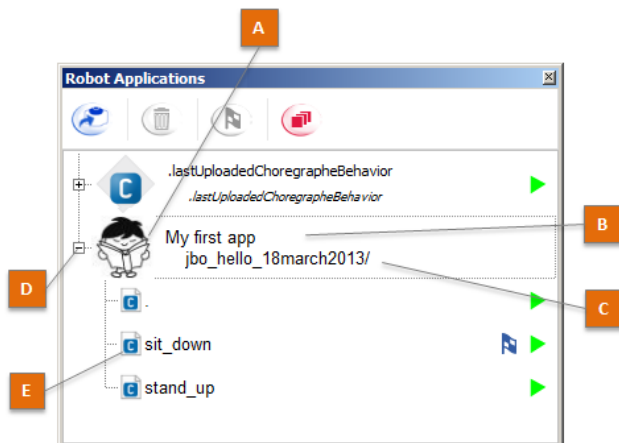
Robotin sovelluspalkki näyttää käytössä olevat sovellukset yhdistettyyn robottiin. Robotin sovelluspalkilla voit suorittaa, pysäyttää tai siirtää toiminnot ja sovellukset yhdistettyyn robottiin. Voidaan myös asettaa oletusarvot sekä pakata ja ladata nykyinen toiminta sovelluksena yhdistettyyn robottiin.



Kuva 45. Robotin näkymän alapuolella sijaitsee sovelluksien hallintaan tarkoitettu valikko. (Aldebaran www-sivut 2018)

Kuvan 45 ensimmäisellä painikkeella (A) voidaan suorittaa sovelluksia ja toimintoja. Omasta ohjelmasta voidaan tehdä sovellus myös painamalla tätä nappia, jolloin se näkyy myös Pepper-robotin tabletin sovelluslistassa. Sovelluslistasta (B) nähdään käytettävissä olevat sovellukset.

Asetuksista (C) voidaan muokata sovelluksen parametreja. Suoritustila tulee näkyviin asetusten oikealle puolelle (D). Vihreä nuoli kuvaa Play-tilaa ja punainen neliö Stop-tilaa.



Kuva 46. Sovellusvalikon eri osat. (Aldebaran www-sivut 2018)

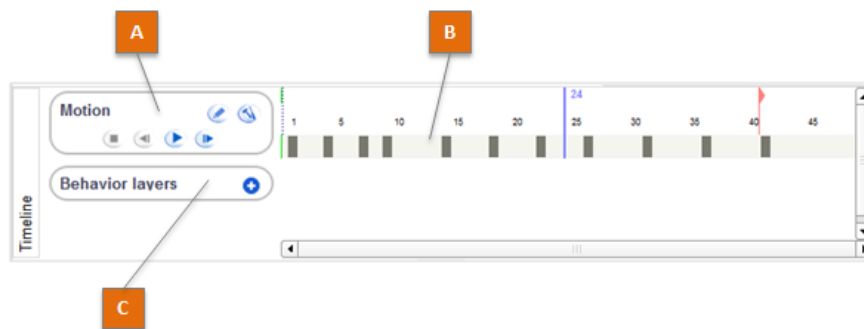
Kuvassa 46 nähdään My first app -sovellus ja sen puurakenne. A kuvaa sovelluksen ikonia, se voidaan vaihtaa asetuksista. B kuvaa sovelluksen nimeä, se näkyy myös esimerkiksi Pepper-robotin tabletilla. C kuvaa sovelluksen ID:ta. Miinus ja plus -painikkeista (D) voidaan laajentaa tai pienentää puurakenteen näkyvyyttä. Puurakenteesta nähdään sovelluksen eri toiminnot (E) ja niiden omat ID:t. Tässä kuvassa My first app sisältää istumisen ja seisomisen toiminnot.

Yläpalkin sininen painonappi kuvaa toiminnon lataamista sovelluksena robottiin. Lipun kuva on sininen, kun toiminto on valittu oletukseksi. Punainen nappi pysäyttää kaikki suoritettavat toiminnot.

4.4 Animointi

4.4.1 Timeline

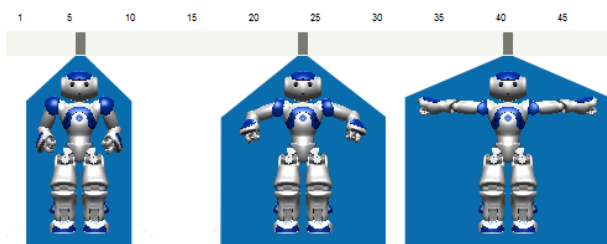
Timeline-boksiin pystytään lisäämään liikettä ja erilaisia toimintobokseja, ja toiminnot toteutetaan aikajanalla tietyssä ajassa.



Kuva 47. Aikajanana rakenne. (Aldebaran www-sivut 2018)

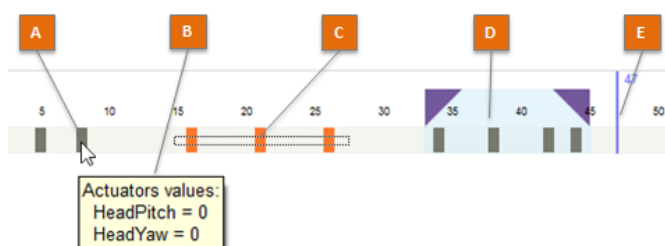
Kuvassa 47 nähdään aikajanana eri osat. Motion-laatikko kuvaa liikkeen editoria. Editorissa voidaan hallita liikkeen aikajanaa. Laatikosta löytyy play-, stop- ja kelaus-painikkeet. Yläreunassa sijaitsevat liikkeen muokkauspainikkeet. Motion-laatikon vieressä on aikajana, jossa nähdään liikkeen paikat ja ajan kulku. Aikaviivaimella (B) näkyvät kehysnumerot, johon voidaan määrittellä aloitus- ja lopetuskohta. Vihreä nuoli aikaviivaimella kuvaa aloituskehystä, josta ohjelma lähtee suorittamaan ohjelmaa aikajanalla. Sininen kursori näyttää, missä kohdassa ohjelma kulkee. Kursori liikkuu suoritetun ohjelman mukana, kunnes saavuttaa loppukehysten. Punainen nuoli kuvaa lopetuskehystä.

Harmaalla pohjalla taas nähdään liikkeen palkit, jotka kuvaavat animoituja peräkkäisiä asentoja. Asennot (kuva 48) yhdessä luovat liikkeen. Aikajanalle (C) voidaan lisätä myös eri kerroksia ja niissä voidaan luoda eri toimintoja, esimerkiksi istumisen ja seisomisen erikseen omiin kerroksiin. Ohjelman suorituksen nopeutta voidaan säätää asetuksista. Goto- ja Stop-bokseilla voidaan pysäyttää, keskeyttää ja siirtää aikakursoria toiseen kehykseen.



Kuva 48. Esimerkkinä NAO-robotin liikkeitä harmaiden palkkien kohdalla.

(Aldebaran www-sivut 2018)



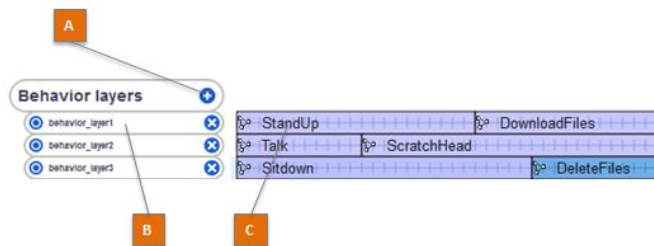
Kuva 49. Palkkien hallinta aikajanelle. (Aldebaran www-sivut 2018)

Kuvassa 49 harmaa palkki (A) kuvaa asentoa, joka on opetettu robotille. Kun hiiren asettaa palkin päälle, se näyttää asennon arvot (B). Asentoja voidaan myös kopioida ja sijoittaa aikajanelle.

Esimerkiksi liikkeiden opettamisessa, kun halutaan pitää tietty asento vähän kauemmin, kannattaa sama asento kopioida toisen perään. Kannattaa myös ottaa huomioon liikkeiden suoritukseen tarvittava aika. Liikkeiden välinen aika määrittää sen, kuinka kauan robotille annetaan aikaa suorittaa liike. Se auttaa liikkeen luonnollisuuteen ja näin vältetään töksähtelevät ja liian nopeat liikkeet vierittämällä harmaata palkkia kauemmaksi. Nopeasti suoritettavissa liikkeissä taas harmaita laatikoita voidaan sijoittaa lähemmäs toisiaan.

Oranssi väri (C) kuvaa valittuja asentoja, jolloin niitä voidaan siirtää pitämällä hiirtä pohjassa ja vierittämällä haluttuun suuntaan.

Sinisen ja violettien kehyksen (D) alla voidaan levittää harmaiden palkkien väliä ottamalla kiinni jommasta kummasta violetin laatikon reunasta ja vierittää haluttuun suuntaan. Violetti kursori (E) osoittaa, mihin seuraava asento sijoittuu.



Kuva 50. Toimintokerrosten luominen. (Aldebaran www-sivut 2018)

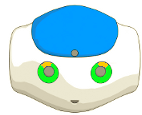
Kuvassa 50 esitetään toimintokerrosten osia. Kerroksia voidaan lisätä plus-painikkeesta (A). Kerroksille voidaan asettaa nimi. Kun kerrosta painetaan, se tulee aktiiviseksi ja avautuu uusi kulkukaavio (B). Kulkukaavioon voidaan lisätä laatikoiden avulla erilaisia toimintoja (C). Rastista voidaan poistaa kerros. Sinisestä pallosta taas nähdään, että kerros sallitaan ja suoritetaan. Jos se on valkoinen, sitä ei sallita, eikä suoriteta. Tässä tapauksessa robotti suorittaa esimerkiksi seisomisen, puhumisen ja istumisen.

4.4.2 Liikkeiden opettaminen

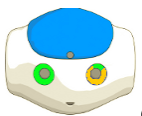
Liikkeiden opettaminen lähtee liikkeelle Timeline-boksin luomisesta kulkukaavioon. Boksi saadaan lisättyä kulkukaavioon painamalla hiiren oikeaa näppäintä ja create new box → Timeline. Robotti tulee laittaa animointitilaan, jotta liikkeiden opettaminen on mahdollista. Esimerkiksi Pepperin silmien ledeistä nähdään, mitkä osat ovat vapautettuina ja mitkä osat ovat jäykät. Alla olevassa kuvassa nähdään ledien värit, vapaana olevia raajoja voi liikuttaa. Silmät ovat oranssit, kun asento on jäykistetty.



pää ja molemmat kädet jäykistetty



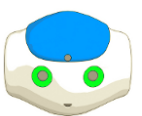
Pää jäykistettynä, kädet vapaana



oikea käsi vapaana, vasen käsi jäykistetty, pää vapaa



vasen käsi vapaana, oikea jäykistetty, pää jäykistetty



Kaikki vapaana



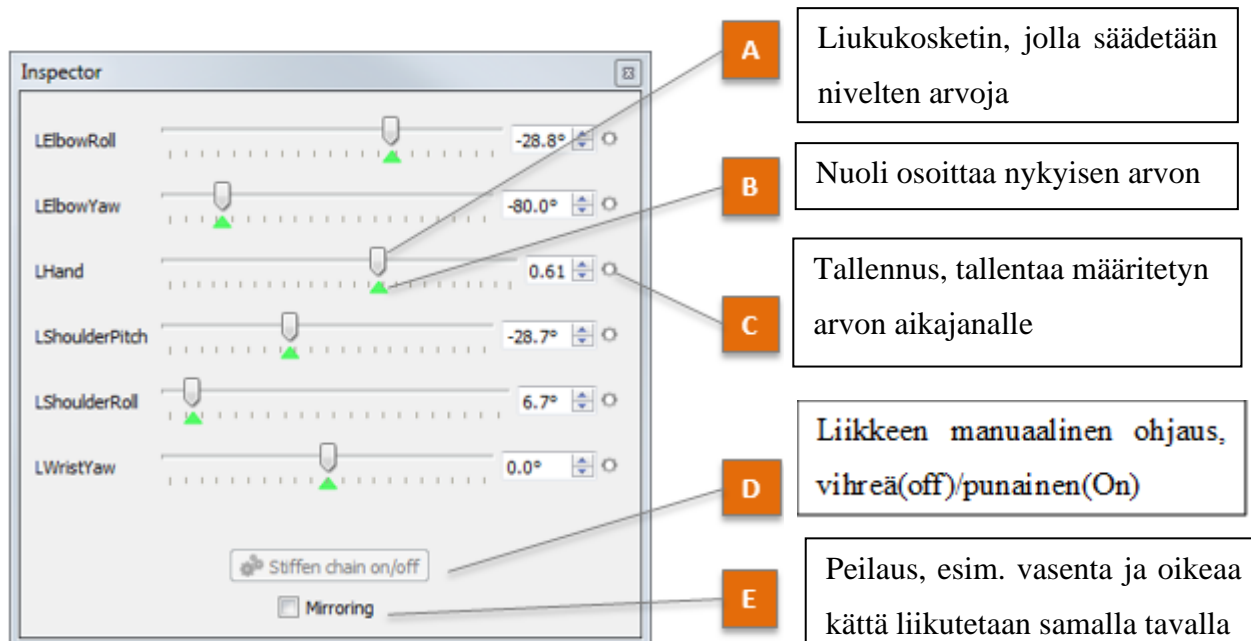
oikea käsi vapaana, vasen jäykistetty

Kuva 51. Nao-robotin silmien ledeistä nähdään, mitkä raajat ovat vapautuneet ja liikutettavissa animointitilassa. Ne ovat identtiset Pepper-robotilla. (Aldebaran www-sivut 2018)

Silmien ledien ollessa vihreät tiedetään, mitä osaa voidaan liikuttaa kämmenen anturin ollessa aktiivinen. Halutun asennon löydyttyä vapautetaan käsi kämmenen anturilta, jolloin asento lukittuu.

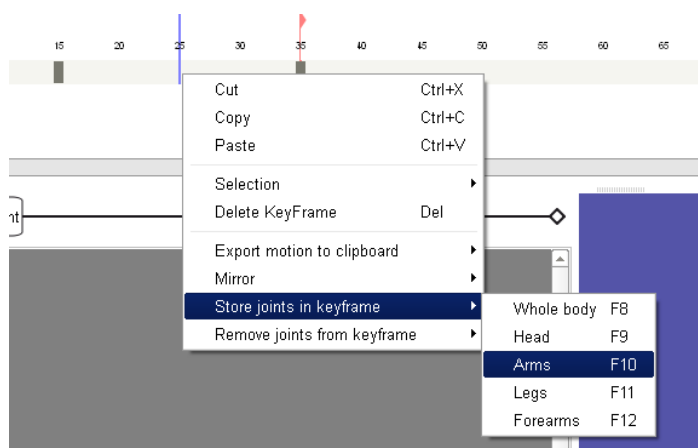
Haluttu asento tallennetaan aikajanelle laittamalla käsi päässä sijaitsevan anturin päälle. Kun korvien ledit ovat vilkahtaneet sinisenä, asento on tallennettu. Tallennettu asento näkyy aikajanelle harmaana palkkina.

Jos fyysistä robottia ei ole saatavilla, voidaan virtuaalirobotin niveliä liikuttaa ja tallentaa asentoja. 3D-virtuaalirobotin näkymässä on Inspector, josta voi muuttaa nivelten arvoja. Kun painetaan virtuaalirobotin mitä tahansa liikkuvaa osaa, saadaan nämä asetukset näkyviin. Kun virtuaalirobotin käsi on halutussa asennossa, painetaan tallennusnappia ja se muuttuu punaiseksi.



Kuva 52. Virtuaalirobotin nivelten asentoja voidaan muuttaa oheisen asetukset-ikkunan avulla. (Aldebaran www-sivut 2018)

Kun halutaan asettaa asento tiettyyn kohtaan aikajanelle, aikajanan sinisen kursorin kohdalla painetaan hiiren oikeaa näppäintä ja valitaan ”Store joints in keyframe”. Valikosta voidaan valita, mitkä nivelten asennot tallennetaan. Kuvan 53 tapauksessa tallennetaan virtuaalirobotin käsien asennot aikajanelle.



Kuva 53. Asentojen tallentaminen aikajanelle. (Aldebaran www-sivut 2018)

4.5 Python

Choregraphessa voidaan myös eri toiminnot toteuttaa koodaten python-ohjelmointikielillä. Voidaan luoda boksi ja ohjelmoida haluttavat toiminnot python-laatikkoon. Kuvasta 54 nähdään Python-boksin sisältö.

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of the box
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to call onUnload of this box in a onStop method.
        pass
```

Kuva 54. Python-boksin sisältö. (Aldebaran www-sivut 2018)

Ensin määritellään luokka MyClass. GeneratedClass on luokka, jossa toteutetaan metodit. Se sisältää kaiken tarvittavan informaation, kuten inputit, outputit ja muut parametrit.

Kun boksi on ladattu, kutsutaan onLoad-funktiota. Funktio voi sisältää toiminnon, jota kutsutaan, kun boksi on ladattu. onUnload-funktio toimii samalla tavalla, kun boksia ei ole ladattu. Se suorittaa tietyn metodin, joka on ohjelmoitu funktioon. Näitä ei ole kuitenkaan pakko käyttää.

Jos halutaan vaikuttaa Timeline-boksiin, käytetään self.getTimeline() -funktiota. Timelinen toimintoja voidaan hallita erilaisia funktiota käyttäen. Esimerkiksi Play() funktio suorittaa aikajanahan kursorin nykyisestä sijainnista loppuun asti, kun taas goTo(int) funktion avulla voit siirtää kursoria. Stop() -funktio luonnollisesti pysäyttää suorittamisen. Pause() pysäyttää suorittamisen siihen kohtaan, jossa kursori on liikunnut aikajanalla.

Alla oleva kuva (kuva 55) esittää Animated Say-boksin sisältöä.

```

1 import time
2
3 class MyClass(GeneratedClass):
4     def __init__(self):
5         GeneratedClass.__init__(self, False)
6         self.tts = ALProxy('ALAnimatedSpeech')
7         self.ttsStop = ALProxy('ALAnimatedSpeech', True) #Create another proxy as wait is
            blocking if audioout is remote
8
9     def onLoad(self):
10        self.bIsRunning = False
11        self.ids = []
12
13    def onUnload(self):
14        for id in self.ids:
15            try:
16                self.ttsStop.stop(id)
17            except:
18                pass
19        while ( self.bIsRunning ):
20            time.sleep( 0.2 )
21
22    def onInput_onStart(self):
23        self.bIsRunning = True
24        try:
25            movement = self.getParameter("Speaking movement mode")
26            textParam = self.getParameter("Text")
27            if movement == "disabled":
28                textParam = "^start({0}) {1}
29        ^wait({0})".format(self.getParameter("Animation"), textParam)
30            sentence = "\RSPD="+ str( self.getParameter("Speed (%)") ) + "\ "
31            sentence += "\VCT="+ str( self.getParameter("Voice shaping (%)") ) + "\ "
32            sentence += textParam
33            sentence += "\RST\ "
34            id = self.tts.post.say(str(sentence), {"speakingMovementMode":movement})
35            self.ids.append(id)
36            self.tts.wait(id, 0)
37        finally:
38            try:
39                self.ids.remove(id)
40            except:
41                pass
42            if ( self.ids == [] ):
43                self.onStopped() # activate output of the box
44                self.bIsRunning = False
45
46    def onInput_onStop(self):
47        self.onUnload()

```

Kuva 55. Animated Say-laatikon toteutus python-ohjelmointikielillä.

Alussa Python ajaa time-nimisen tiedoston. Se luetaan aina, kun ohjelmassa kutsutaan time-parametria. Se on tiedosto tai ”kirjasto”, josta löytyvät funktiot. Kolmannella rivillä luodaan luokka MyClass, joka esittää boksia. GeneratedClass on luokan MyClass yläluokka, joista löytyy sisäänrakennettuja metodeita. Rivillä 4 määritellään luokan metodi __init__, jonka avulla luodaan uusi instanssi, ”Olio”. Luokan jäsenfunktioissa määritellään olioiden toiminnallisuus. Ensimmäisen parametrin nimi on self, koska niin on sovittu ja se on yleisesti käytössä. Rivillä 5 asetetaan metodille tila False. Sitä ei suoriteta.

Rivillä 6 asetetaan parametrin arvo tts (Text to say) kenttään. Arvo on ALProxy ("ALAnimatedSpeech"). Alproxy antaa pääsyn moduuliin, johon olet yhdistämässä, ja sen arvo on ALAnimatedSpeech. ALAnimatedSpeech on mooduli, se vastaanottaa tekstin ja prosessoi sen, ja robotti sanoo tekstin.

Kun boksi on "ladattu" (onLoad), bIsRunning parametrin arvo asetetaan tilaan False eli ei suoriteta vielä mitään, vaan vasta sitten, kun input saa signaalin. Seuraavalla rivillä luodaan uusi tyhjä taulukko muuttujalle ids.

Kun boksi on suoritettu (Unload), ids-tilukosta haetaan id-arvo. Käytetään try – except -rakennetta, jossa käytetään stop() -funktiota, joka saa arvonsa taulukosta. Eli pysäytetään toiminto 200 ms viiveellä while-komennolla, kun stop saa arvon. Muussa tapauksessa ei tehdä mitään, jolloin käytetään syntaksia pass.

Kun input OnStart saa signaalin, suoritetaan seuraavat toiminnot riviltä 23 alkaen. bIsRunning asetetaan tilaan True. Käytetään taas try – except -rakennetta. Self.getParameter on funktio, joka tulostaa minkä tahansa parametrin arvon. Tässä tapauksessa se tulostaa Speaking movement moden arvon movement-muuttujaan. Tämä kuvaa suoritettavaa animaatiota, joka on valittu. Seuraavalla rivillä tulostuu Text-kentän arvo muuttujaan TextParam. Eli kun set parameters animated say -laatikkoon kirjoitetaan teksti, jonka robotin on tarkoitus sanoa, asetetaan lause tässä kohdassa. Jos movement arvo on disabled eli animaatio on pois käytöstä, vain TextParam arvo käynnistetään, eli robotti toistaa animated say -boksiin sijoitettu tekstin ilman animointia. Se odottaa (wait) kunnes teksti on toistettu, jolloin output voidaan asettaa päälle. Sentence muodostuu kaikista niistä ominaisuuksista, mitä set parameters ikkunasta voi muokata. Äänen nopeutta ja muotoa voi muokata liukukoskettimien avulla. id toimii kuin say boksi ja palauttaa tekstin arvon ja speaking movement mode saa arvon movement-muuttujasta. Toistetaan annettu parametrin arvo animaationa. Append siirtää ids-listaan id:n arvon. Rivillä 35 odotetaan, että id saa arvon 0, jotta tiedetään, että teksti on toistettu. Sitten ids siirretään pois listasta. Jos lista on tyhjä, voidaan siirtyä onStopped-tilaan ja aktivoida output. Sen jälkeen asetetaan bIsRunning tilaan False. Lopuksi kutsutaan unload() -funktiota ja 200ms viiveellä pysäytetään boksi.

4.6 Korjaus ja virhetilanteet

Jos robotti ei saa yhteyttä ohjelmaan tai robotti ei löydy listasta, on varmistettava, että robotti on varmasti päällä ja yhdistetty omaan verkkoon. Yhdistettäessä robottia voi käyttää fixed IP/hostname ja kirjoittaa IP-osoite ja yrittää yhdistää uudelleen. Jos tämäkään ei auta, voidaan yrittää toimia seuraavasti: Käynnistetään Choregraphe ilman NAOqia, koska NAOqi on sulautettu Choregrapheen ja voi olla viallinen. Voidaan myös yrittää testata ohjelmaa ilman palomuuria ja sen jälkeen poistaa Choregraphen viitteet valtuutetuista ohjelmista tietokoneen palomuurin asetuksista. Choregraphen voi yrittää asentaa uudestaan, jos ongelma ei poistu.

Robotin päivittäminen Choregraphella saattaa epäonnistua, ja se voi johtua useista syistä. Ohjelma antaa usein virhekoodin. Koodi ja sanoma ilmestyvät ohjelman alaosaan viestilaatikkoon. Koodi 1 tarkoittaa yhteysvirhettä, jolloin varmistetaan yhteys ja päivitetään uudelleen. Koodi 2 ilmoittaa, ettei tiedostoa löydy. Varmistetaan tiedostojen oikea sijainti, valitaan tiedosto ja päivitetään uudelleen. Koodi 3 ilmoittaa SSH-yhteyden virheestä, ja toimenpiteiksi suositellaan yrittämään uudestaan. Koodi 4 ilmoittaa yhteyden katoamisesta, jolloin yritetään uudestaan. Koodi 5 ilmoittaa todentamisvirheestä, jolloin suositellaan tarkistamaan robotin salasana ja yrittämään uudestaan. Koodi 6 ilmoittaa ei voimassa olevasta ladastusta tiedostosta, jolloin suositellaan yrittämään uudelleen ja ongelman jatkuessa kehoitetaan ottamaan yhteyttä aldebaranin tukeen. Koodi 7 ilmoittaa virheestä datan siirron aikana, ja toimenpiteiksi suositellaan yrittämään uudestaan ja tarkistaa robotin muisti, sillä virhe saattaa johtua tilan puutteesta.

Toimintojen testaamisessa voi myös ilmetä ongelmia, jos ohjelma ei lataa ohjelmaa play-painikkeesta. Ongelma voi johtua siitä, että robotti on käynnistetty uudelleen, ja näin ollen Choregraphe katkaisee automaattisesti yhteyden robottiin. Kun ilmenee ongelmia toimintolaatikoissa, suositellaan testaamaan laatikoita yksi kerrallaan, jotta löydetään virhe. Laatikoita voi testata Play-painikkeesta tai kaksoisklikkaamalla laatikon inputtia. Kun halutaan nähdä, mitä robotti suorittaa tällä hetkellä tai mitä se suoritti viimeksi, voidaan avata robotin omat IP-sivut ja avata lista suoritetuista toiminnoista. Lista voi auttaa virheen löytämisessä ohjelmassa, jolloin nähdään, missä kohtaa robotti on vielä suorittanut ohjelmaa onnistuneesti.

Jos laatikko muuttuu punaiseksi ohjelman suorituksen aikana, ohjelmassa ilmenee virhe, eikä ohjelmaa kyetä suorittamaan. Se voi johtua siitä, että moduulia ei voida ladata tai sitä ei ole olemassa. Jos Play-painikkeen jälkeen painettaessa punainen laatikko menee virheeseen, saattaa python-koodissa olla virhe. Jos laatikko menee punaiseksi, kun ohjelma on suoritettu, python-koodin metodin kutsu tai sen suorittaminen on vielä kesken, vaikka boksi on jo suoritettu. Python-koodilla on toimittava niin, että muut kutsut on suoritettu ennen onUnload kutsua.

Boksin python-koodiin voidaan lisätä self.logger-muuttuja kertomaan varoitukset ja virheet. Lisätään alla oleva koodi omaan Python-koodiin, jolloin viesti-ikkunaan ilmestyy boksin virheiden ilmoitukset.

```
def onLoad(self):  
    self.logger.debug("Loaded box %s", self.getName())
```

Kuva 56. Varoitusten ja virheiden ilmoitus python-koodissa. (Aldebaran www-sivut 2018)

5 KAUPASSAKÄYNNIN OPETUSVIDEON TOTEUTUS JA VIITTOMIEN OPETTAMINEN PEPPERILLE

Kaupassakäynnin opetusvideon tarkoituksena on hyödyntää erilaisia Pepperin ominaisuuksia ja toteuttaa opetusvideo tukiviittomien kanssa. Opetusvideossa yhdistetään erilaisia elementtejä, kuten ääni, animointi (eleet), liikkeiden opetus (tukiviittomat) ja näytöllä esitettävät kuvat. Ohjelmassa on 13 lausetta. Pepperin on tarkoitus viittoa tummennetut sanat. Ohjelmaan tuli projektin aikana muutoksia, mutta muutokset eivät kuitenkaan olleet niin merkittäviä, että tutkimus olisi niiden vuoksi muuttanut luonnettaan. Seuraavissa kappaleissa esitetään molemmat versiot ja lopputulokset.

5.1 Alkuperäinen ohjelma

Ohjelmassa on tarkoitus opettaa kaupassakäyntiä. Ohjelmaa lähdettiin rakentamaan ensin niin, että opetettiin tukiviittomat Timeline-boksia hyödyntäen. Tukiviittomien jälkeen ohjelmaa rakennettiin sitä mukaan, kun löydettiin erilaisia vaihtoehtoja toteuttaa tietty toiminto. Tässä vaiheessa ei ollut vielä ratkaisua kuvan näyttämisestä tabletilla. Seuraavalla sivulla kuvataan ensimmäisen ohjelman rakennetta.

Pepperin opetustuokio: Kaupassa käyminen

1. Tee kauppalista. Kirjoita listaan kaikki, mitä tarvitset kaupasta.

Pepper sanoo: ”Tee kauppalista. Kirjoita listaan kaikki, mitä tarvitset kaupasta.” Tämä on toteutettu Animated say -laatikolla. Lause on kirjoitettu Set parameters Text -kohtaan. Elehdintä tähän on tuotu animaatiokirjastosta. Animaation linkki kopioitiin Animation-kohtaan. Lauseessa ei ole viittomaa.

2. Ota mukaan kauppalista, kauppakassi, rahapussi ja avaimet.

Pepper sanoo: ”Ota mukaan kauppalista, kauppakassi, rahapussi ja avaimet.” Tämä toteutettiin Animated say -laatikolla. Lause on kirjoitettu Set parameters Text -kohtaan. Elehdintä tähän tuotiin animaatiokirjastosta. Animaation linkki kopioitiin Animation -kohtaan. Lauseessa ei ole viittomaa.

3. Kulje kauppaan liikennesääntöjä noudattaen.

Pepper sanoo: ”Kulje kauppaan liikennesääntöjä noudattaen.” Tämä toteutettiin Animated say -laatikolla. Lause on kirjoitettu Set parameters Text -kohtaan. Elehdintä tähän tuotiin animaatiokirjastosta. Animaation linkki kopioitiin Animation kohtaan. Lauseessa ei ole viittomaa.

4. Jos sinulla on **pienet** ostokset, ota **kori**. Jos sinulla on **suuret** ostokset, ota **kärryt**.

Pepper sanoo: ”Jos sinulla on **pienet** ostokset, ota **kori**. Jos sinulla on **suuret** ostokset, ota **kärryt**.” Tämä toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timelinen kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen Text-kohtaan.

Pienet-tukiviittoma (kuva 57) toteutettiin niin, että ensin Pepperin kädet ovat alhaalla. Tämä asento tallennettiin omaksi harmaaksi palkikseen. Sitten nostettiin kädet puoleen väliin osoittamaan ”pientä” ja tallennettiin tämä asento omaksi harmaaksi palkikseen aikajanelle.

Kori-viittoma (kuva 58) toteutettiin nostamalla toinen käsi puoleen väliin ja kääntämällä kämmen nyrkiksi ja taivuttamalla sitä niin kuin se pitäisi mielikuvituskoria kädessä. Tämä asento tallennettiin omaksi harmaaksi palkiksi aikajanelle. Laskettiin kättä hiukan alemmas ja tallennettiin asento. Sitten kopioitiin ylemmässä asennossa oleva ”kori”-asento alemman kori-asennon perään ja taas alempi ”kori” ylemmän perään. Näin saatiin luotua toistoliike.

Suuri-viittoma (kuva 57) toteutettiin samalla tavalla kuin pieni, mutta käsien ollessa puolessa välissä avattiin hiukan niitä sivulle osoittamaan ”suurta”.

Kärryt-viittomassa (kuva 58) ensimmäisessä asennossa kädet ovat puolessa välissä ja nyrkissä. Tallennettiin asento. Sitten taivutettiin käsiä vähän itseensä päin, että näyttäisi siltä, kuin Pepper työntäisi kauppakärryjä. Tallennettiin asento. Kopioitiin asennot taas perätysten kuten kori-viittomassa, ja saatiin toistoliike. Laskettiin kädet alas ja tallennettiin asento aikajanan loppuun.



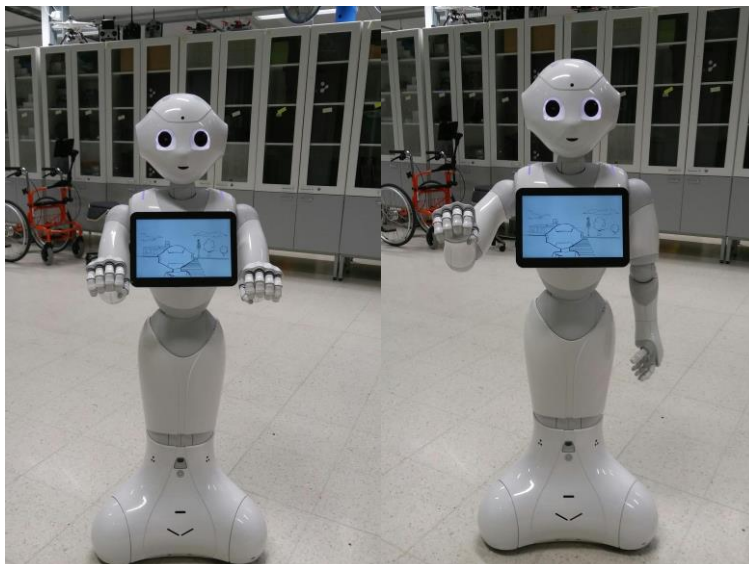
Kuva 57. Pieni- ja suuri-viittoma.

5. **Kerää kaikki** tuotteet **koriin** tai **kärriin**.

Pepper sanoo: ”**Kerää kaikki** tuotteet **koriin** tai **kärriin**.” Tämä toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timelinen kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen Text kohtaan.

Kerätä-tukiviittoma (kuva 60) toteutettiin niin, että ensin Pepperin toinen käsi taivutettiin vastaanottamaan tavaraa ja toinen asetettiin sen ”päälle” niin kuin Pepper raahaisi tavaraa toiseen käteen. Tallennettiin asento. Sitten siirrettiin raahaavaa kättä pois päin toisesta kädestä ja tallennettiin tämä asento.

Kaikki-viittoma toteutettiin nostamalla toinen käsi puoleen väliin ja viistosti vasempaan päin ja käännettiin kämmen nyrkiksi. Tallennettiin asento. Sitten siirrettiin kättä reilusti oikealle samassa linjassa ja tallennettiin asento. Näin luotiin liikerata vasemmalta oikealle. Sitten kopioitiin korin toteutus edellisestä lauseesta kaikki-viittoman harmaiden palkkien perään. Sitten kopioitiin kärriin toteutus kori-viittoman harmaiden palkkien perään. Laskettiin kädet alas ja tallennettiin asento aikajanan loppuun. Toteutus löytyy liitteestä 12.



Kuva 58. Kärry- ja kori-viittoma.

6. Jos **et** löydä **kaikkea** tai jokin tuote on **loppu**, **pyydä** apua myyjältä.

Pepper sanoo: ”Jos **et** löydä **kaikkea** tai jokin tuote on **loppu**, **pyydä** apua myyjältä.” Lause toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timeline kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen Text-kohtaan.

Ei-tukiviittoma on toteutettu samoin kuin pudistaisi päätään puolelta toiselle. Ensin Pepperin päätä siirrettiin vasemmalle ja tallennettiin asento. Sitten päätä käännettiin oikealle ja tallennettiin asento. Kopioitiin edellisen lauseen kaikki-viittoman toteutus ei-viittoman palkkien perään.

Loppu-tukiviittoma (kuva 59) toteutettiin niin, että kädet taivutettiin kyynärpästä osoittamaan eteenpäin ja kämmeniä käännettiin hiukan toisiinsa päin niin, että seuraavassa liikkeessä Pepper kykenisi laittamaan kädet yhteen päälletysten. Tallennettiin asento. Sitten siirretään käsiä yhteen ja toistensa ”ohi”. Tämä oli haasteellista, koska Pepperin kädet ovat lyhyet, eikä niitä saa toistensa ohi. Pepper lyö kädet vain yhteen päälletysten kuvaamaan ”loppua”. Tallennettiin asento.

Pyytää-viittoma (kuva 59) toteutettiin niin, että ensin Pepperin oikea käsi on alhaalla. Tallennettiin se omaksi harmaaksi palkiksi.

Sitten nostettiin käsi puoleen väliin ja tallennetaan se omaksi palkikseen. Seuraavaksi käännettiin kyynärpästä niin, että kämmen osoitti Pepperin ”poskea” kohti. Tallennettiin asento. Sitten siirrettiin kämmen kiinni ”poskeen” ja tallennettiin asento. Sen jälkeen kopioitiin palkit perätysten, jotta saatiin toistoliike. Sitten kopioitiin käsi puolessa välissä -asento toistoliikkeiden perään aikajanalla. Sen perään kopioitiin vielä käsi alhaalla -asento, jolloin Pepperin käsi tulee rauhassa alas. Toteutus löytyy liitteistä 13 ja 14.



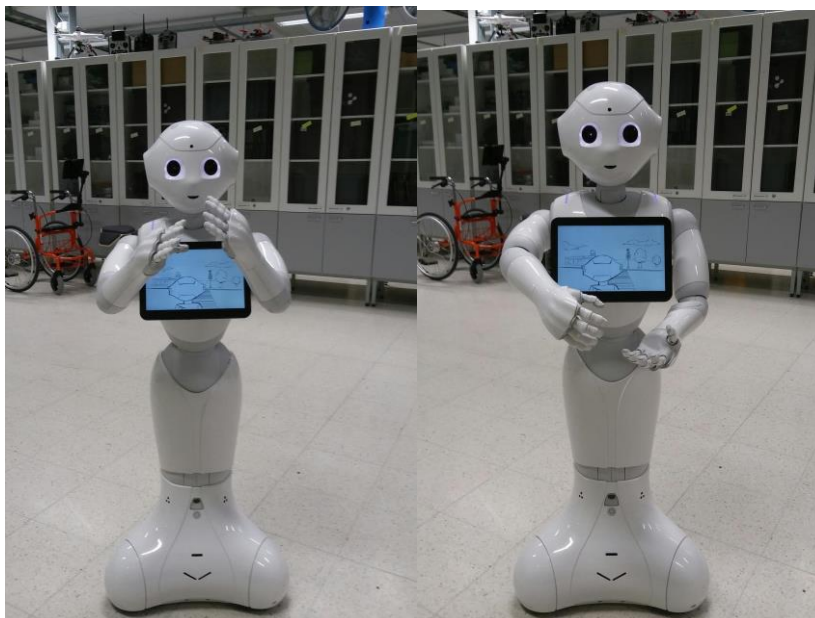
Kuva 59. Pyytää- ja loppu-viittoma.

7. Kun **kaikki** ostokset on **kerätty**, mene **kassajonoon**.

Pepper sanoo: ”Kun **kaikki** ostokset on **kerätty**, mene **kassajonoon**.” Lause toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timelinen kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen Text-kohtaan.

Kaikki-tukiviittoma kopioitiin edellisestä lauseesta aikajanalle. Kerätä-tukiviittoma kopioitiin 5. lauseesta aikajanalle kaikki-viittoman perään. Jonoviittoma (kuva 60) toteutettiin niin, että Pepperin kädet nostettiin taas puoleen väliin ja taivutettiin kyynärpäistä hiukan ylöspäin ja käsiä käännettiin vähän yhteen pystysuunnassa.

Tässä viittomassa oli tärkeää, että sormet ovat suoraan ja hiukan erillään toisistaan. Pepperin sormet eivät liiku kunnolla, mutta niitä saatiin hieman suoraan. Tallennettiin tämä asento. Sitten vietiin käsiä pois päin toisistaan pystysuunnassa. Tallennettiin asento. Laskettiin kädet välipistein alas ja tallennettiin asennot aikajanan loppuun. Toteutus löytyy liitteestä 16.



Kuva 60. Jono- ja kerätä- viittoma.

8. Kun on sinun vuorosi, **nosta** tuotteet kassalle. Jos keräsit ostokset koriin, palauta kori.

Pepper sanoo: ”Kun on sinun vuorosi, **nosta** tuotteet kassalle. Jos keräsit ostokset koriin, palauta kori.” Lause toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timelinen kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen Text-kohtaan.

Nosta-tukiviittoma toteutettiin niin, että nostettiin Pepperin kädet puoleen väliin. Kämmenet käännettiin osoittamaan toisiaan kohti ja sormet olivat mahdollisimman suorassa. Tallennettiin asento. Sitten nostettiin asentoa ja ”kuroteltiin” vähän eteenpäin, jotta se näyttäisi siltä kuin nostettaisiin jollekin tasolle tavaraa. Tallennettiin asento.

Kopioitiin vielä ensimmäinen asento toisen asennon perään, jotta kädet palautetaan, kun on ”nostettu”. Laskettiin kädet välipistein alas ja tallennettiin asennot aikajanan loppuun. Toteutus löytyy liitteestä 17.

9. Maksa ostokset.

Pepper sanoo: ”Maksa ostokset.” Tämä toteutettiin Animated Say -laatikolla. Lause on kirjoitettu Set parameters Text-kohtaan. Elehdintä tähän tuotiin animaatiokirjastosta. Animaation linkki kopioitiin Animation-kohtaan. Lauseessa ei ole viittomaa. Toteutus löytyy liitteestä 18.

10. Pakkaa ostokset kauppakassiin.

Pepper sanoo: ”Pakkaa ostokset kauppakassiin.” Tämä toteutettiin Animated Say -laatikolla. Lause on kirjoitettu Set parameters Text -kohtaan. Elehdintä tuotiin animaatiokirjastosta. Animaation linkki kopioitiin Animation-kohtaan. Lauseessa ei ole viittomaa. Toteutus löytyy liitteestä 19.

11. Jos käytit kärryjä, palauta kärryt.

Pepper sanoo: ”Jos käytit kärryjä, palauta kärryt.” Tämä toteutettiin Animated Say -laatikolla. Lause on kirjoitettu Set parameters Text -kohtaan. Elehdintä tähän tuotiin animaatiokirjastosta. Animaation linkki kopioitiin Animation-kohtaan. Lauseessa ei ole viittomaa.

12. **Mene kotiin** liikennesääntöjä noudattaen.

Pepper sanoo: ”**Mene kotiin** liikennesääntöjä noudattaen.” Lause toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timelinen kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen Text-kohtaan.

Mennä-tukiviittoma toteutettiin niin, että nostettiin Pepperin vasen käsi puoleen väliin ja taivutettiin kyynärpästä käsi pystyyn osoittamaan suoraan ylös. Tallennettiin asento. Sitten taivutettiin käsi suoraksi eteenpäin, jolloin Pepper osoittaa eteenpäin. Tallennettiin tämä asento. Laskettiin käsi alas ja tallennettiin asento.

Koti-tukiviittoma (kuva 61) toteutettiin kääntämällä Pepperin oikea käsi välipistein kohti poskea ja samaan aikaan käännettiin vasen käsi koskettamaan oikeaa kättä sivusuunnassa kyynärpäältä taivuttaen. Laskettiin kädet välipistein alas ja tallennettiin asennot aikajanan loppuun. Toteutus löytyy liitteestä 21.



Kuva 61. Koti- viittoma.

13. Pura ostoskassit heti. Laita etenkin jääkaappiin ja pakastimeen menevät ruoat heti paikalleen.

Pepper sanoo: ”Pura ostoskassit heti. Laita etenkin jääkaappiin ja pakastimeen menevät ruoat heti paikalleen.” Tämä toteutettiin Animated Say -laatikolla. Lause on kirjoitettu Set parameters Text -kohtaan. Elehdintä tuotiin animaatiokirjastosta. Animaation linkki kopioitiin Animation kohtaan. Lauseessa ei ole viittomaa. Toteutus löytyy liitteistä 22 ja 23.

14. Eikös ollutkin **helppoa? Nyt sen tiedät!**

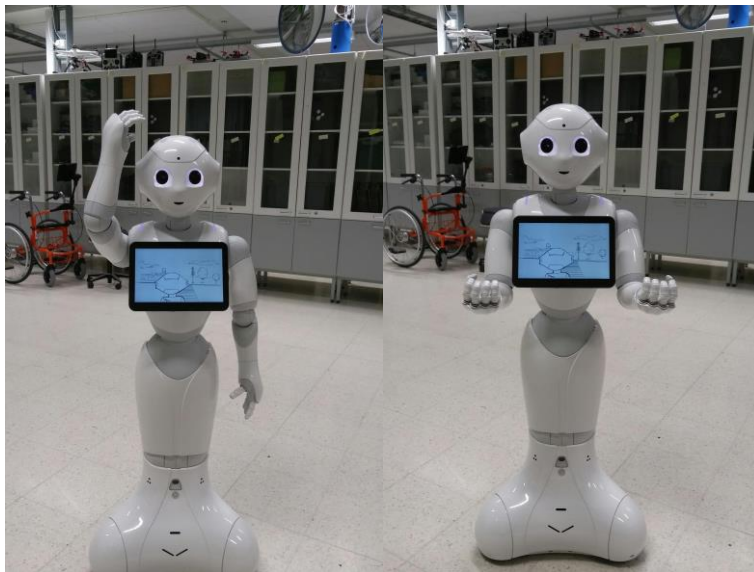
Pepper sanoo: ”Eikös ollutkin **helppoa? Nyt sen tiedät!**” Lause toteutettiin Timeline-laatikolla. Say-laatikko lisättiin Timelinen kulkukaavioon. Say-laatikon parametreihin lisättiin yllä oleva lause taukoineen.

Helppo-tukiviittoma (kuva 62) toteutettiin niin, että nostettiin Pepperin kädet puoleen väliin ja taivutettiin kämmenet osoittamaan kohti kattoa.

Tallennettiin asento. Seuraavaksi käännettiin Pepperin sormia kämmeniä kohti. Tallennettiin asento. Kopioitiin nämä asennot perätysten, jolloin saatiin toistoliike. Laskettiin kädet alas ja tallennettiin asento.

Nyt-tukiviittoma toteutettiin nostamalla kädet taas puoleen väliin ja taivutettiin kyynärpästä kättä ylemmäs ja kämmenet kasvoja päin. Tallennettiin asento. Laskettiin käsiä hiukan alas ja tallennettiin asento.

Tietää-tukiviittoma (kuva 62) toteutettiin nostamalla oikea käsi puoleen väliin. Tallennettiin asento. Nostettiin käsi Pepperin päätä kohti ja koskettamaan lähes päätä. Tallennettiin asento. Sitten siirrettiin kämmen koskettamaan Pepperin päätä. Tallennettiin asento. Asennot kopioitiin perätysten, jolloin saatiin toistoliike. Laskettiin käsi välipistein alas ja tallennettiin asento aikajanan loppuun. Toteutus löytyy liitteestä 24.



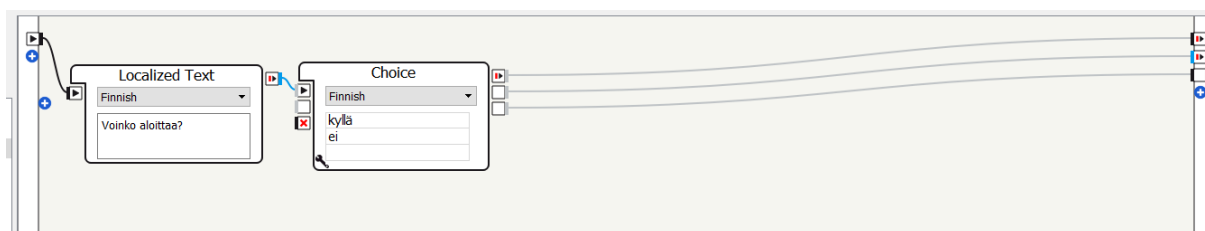
Kuva 62. Tietää- ja helppo-viittoma.

5.2 Alkuperäinen ohjelma

Alkuperäisessä opetusvideossa alkuun lisättiin Choice-laatikko. Choice-laatikolla voidaan määrittää Pepper-robotille kysymys, jonka se esittää saadessaan katsekontaktin, ja vastauksen perusteella Pepper jatkaa seuraavaan vaiheeseen. Seuraava vaihe on usein määritetty ohjelmaan.

Choice-laatikko lisättiin voice-kansion alta kirjastosta. Choice-laatikko vaatii Localized Text -laatikon, jotta saadaan määritettyä kieli ja kysymys (kuva 63).

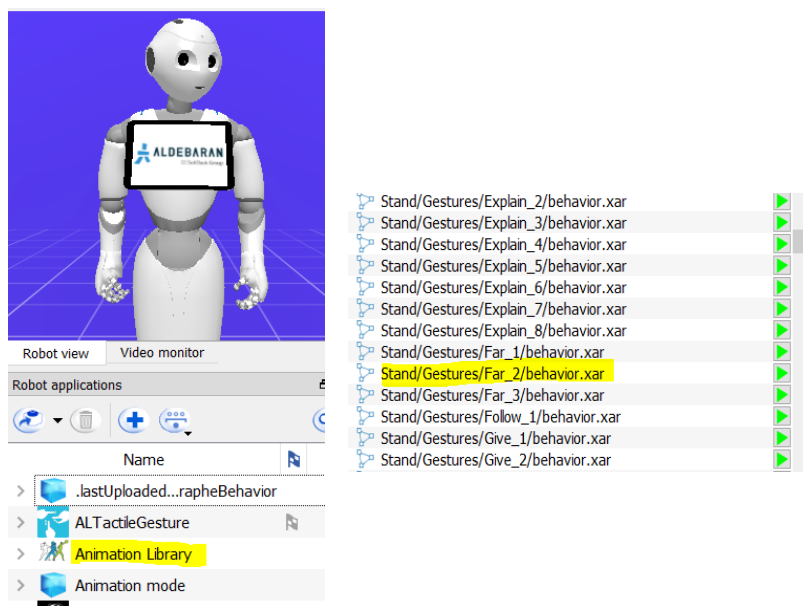
Kieli asetetaan suomeksi laatikon yläpuolella sijaitsevaan kenttään ja alapuolella sijaitsevaan tekstikenttään kirjoitetaan lause, jonka Pepper toistaa. Lisätään Localized Text -laatikko, josta löytyvät vastaukset. Asetetaan kieli suomeksi ja alapuolelle sijaitsevaan kenttään valitaan yksi tai useampi vastaus. Ohjelmassa vastauksina toimivat ”kyllä” ja ”ei”. Jos vastataan Pepperille ”kyllä”, Pepper lähtee suorittamaan ohjelmaa. Jos vastaus on ”ei”, Pepper jatkaa kysymistä. Pepper kysyy: ”Aloitetaanko?”, ja kun Pepper saa vastauksen ”kyllä”, Pepper sanoo: ”Aloitetaan!” ja lähtee suorittamaan valmista ohjelmaa.



Kuva 63. Choice-laatikon sisältö.

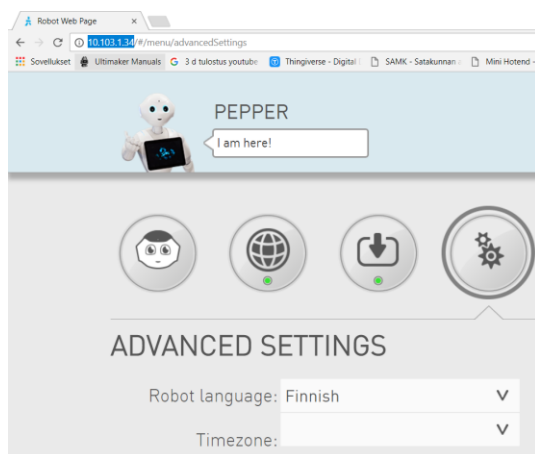
5.2.1 Lauseet ilman tukiviittomia

Lauseet ilman tukiviittomia on toteutettu Animated Say -laatikkoa käyttäen. Animated Say -laatikko valittiin, koska se sisältää puheominaisuuden ja elehdinnän. Näin puhuminen on luontevampaa. Animated Say -laatikko liu’utetaan työalustalle ja yhdistetään edelliseen choice-laatikkoon.



Kuva 64. Animation library:n sijainti ja eleen valinta.

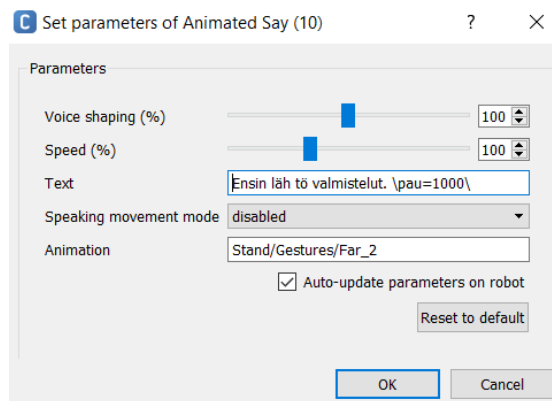
Animation-kohtaan voidaan valita oikealta alakulmasta robot view -kohdan alapuolella sijaitsevasta Animation Librarysta animaatio (kuva 64). Oikeasta laidasta löytyy erilaisia eleitä tekemään sanomasta luonnollisemman. Kirjasto on hyvin laaja ja eleen voi toistaa testaukseksi vihreästä nuolesta. Aloituksiin valitaan ele Far_2.



Kuva 65. Pepperin oma sivusto. Lause kirjoitetaan ylänurkkaan ”I am here!” -kenttään ja robotin kieli vaihdetaan Robot Language kohdasta.

Lausetta testattiin Pepperin omalla sivustolla (kuva 65). Sivustolle pääsee kirjoittamalla Pepperin IP-osoite selaimen osoitteeksi. Ensinnäkin pitää määrittellä kieliasetukset oikeaksi ennen lauseen testaamista.

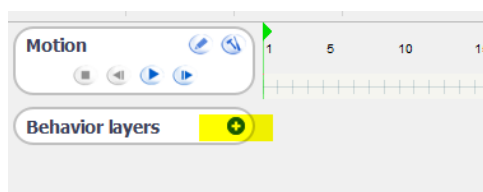
Lauseen voi kirjoittaa ”I am here! ” -tekstikenttään ylälaudassa ja Pepper toistaa lauseen Enteriä painamalla. Lauseeseen lisätään ylimääräisiä kirjaimia ja taukoja, jotta se kuulostaa aidommalta. Lausetta voidaan myös tavuttaa, jotta se saadaan kuulostamaan paremmalta. Tauot toteutetaan millisekunteinä ja kirjoitetaan muotoon `\pau=x\`, esimerkiksi 1 sekunti merkitään `\pau=1000\`. (kuva 66). Toiminallisuutta voidaan testata play-painikkeesta parametrien sijoittamisen jälkeen.



Kuva 66. Animated Say-laatikon parametrien ikkuna, johon voidaan kirjoittaa lause sekä animaatio.

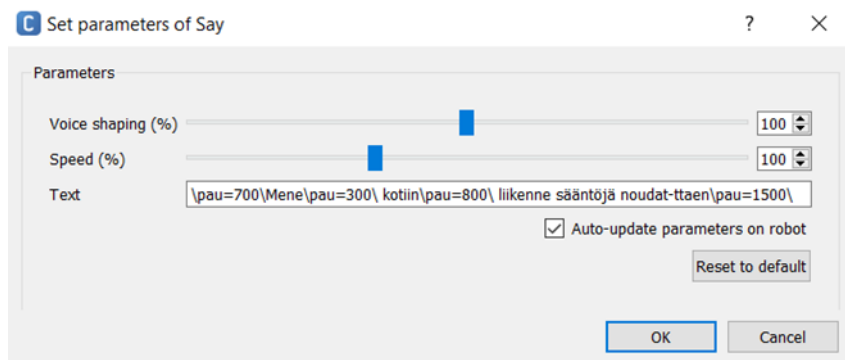
5.2.2 Lauseet tukiviittomilla

Alkuperäisessä ohjelmassa tukiviittomia sisältävät lauseet toteutettiin Timeline-boksia käyttäen. Luotiin Timeline-boksi pääkulkukaavioon. Aikajanelle opetettiin viittomat ja kulkukaavion ensimmäiseen kerrokseen (kuva 67) lisättiin Say-boksi toistamaan lauseet. Painetaan keyframe aktiiviseksi, jotta saadaan kulkukaavio aktiiviseksi muok-kausta varten.



Kuva 67. Toimintokerroksen lisäys.

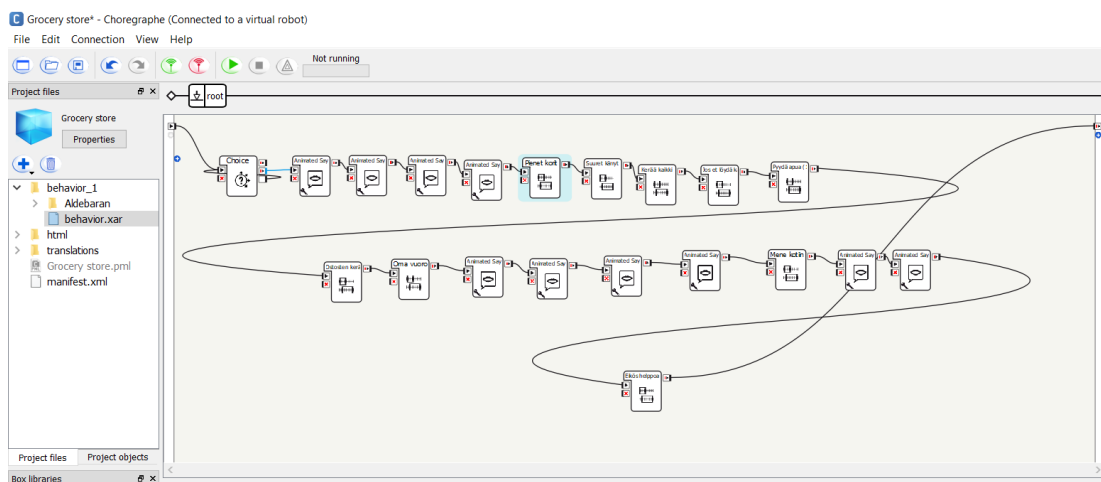
Kun kulkukaavio on aktiivinen, lisätään myös say-boksi kirjastosta ja liu'utetaan kulkukaaviolle. Say-boksi on tässä tapauksessa paras vaihtoehto puheen tuottamiseen, koska liikkeet opetetaan aikajanelle ja ylimääräistä elehdintää ei tarvita. Say-boksiin kirjoitetaan lause (kuva 68), joka on testattu Pepperin sivustolla.



Kuva 68. Say-boksin parametrien asettelu.

Kun lähdetään opettamaan liikkeitä, painetaan ensin sydäimestä autonomous life off. Pepper nukahtaa ja sen jälkeen ”herätetään” Pepper painamalla tähden muotoisesta painikkeesta. Laitetaan animation mode päälle. Opetetaan viittomat yksi kerrallaan ja välipistein.

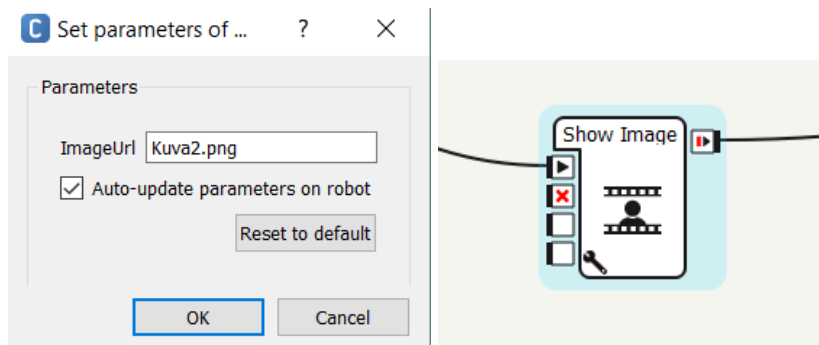
Haasteena on saada ajoitettua lause ja viittomat yhtenevästi siten, että tummennetut sanat ja viittomat toteutuisivat samaan aikaan. Ohjelmaa testaamalla, taukoja lisäämällä ja palikoiden siirtämisellä saadaan ajoittaminen onnistumaan. Melkein valmis ohjelma nähdään alhaalla ilman kuvien toteutusta (kuva 69).



Kuva 69. Ohjelma koottuna ilman kuvien toteutusta pääkulkukaavioon.

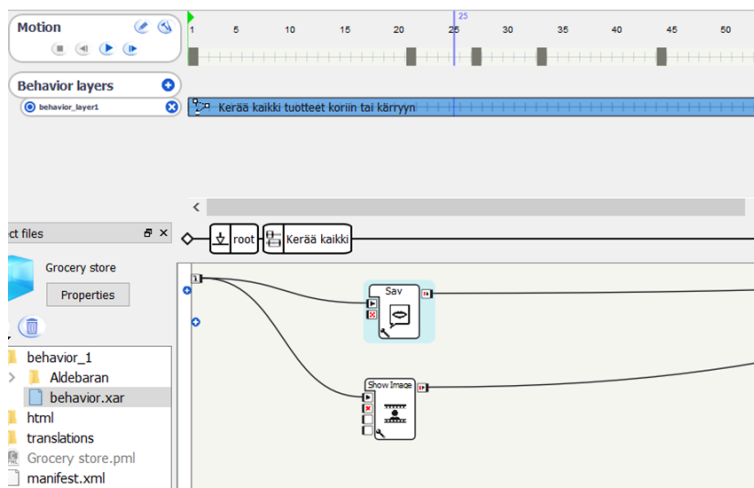
Kuvat saatiin myöhemmin käyttöön. Kuvat piti muokata oikean kokoisiksi, jotta ne mahtuisivat tabletin näytölle. Koko on noin 1280 x 800 pikseliä. Show image -laatikko tukee kaikkia kuvatiedostomuotoja, kuten png-, gif- ja jpg-tiedostotyyppejä.

Show image -laatikko (kuva 70) hakee kuvat ohjelman html-kansiosta, joten kuvat on lisättävä sinne. Polku kirjoitetaan esimerkiksi muodossa ”Kuva2.gif.”



Kuva 70. Show Image-laatikon toteutus. (Kuvakaappaus omasta ohjelmasta.)

Show Image -laatikko lisättiin samaan kerrokseen kulkukaavioon say-boksin rinnalle (kuva 71), jolloin puheen ja kuvan ajoittuminen onnistuisivat samanaikaisesti.



Kuva 71. Show-Image boksen sijoittaminen Say-boksin rinnalle.

Animated Say -boksit oli tarkoitus sijoittaa Flow diagrammeihin Show image -boksin rinnalle. Ohjelmaan tuli tässä kohtaa muutoksia ja ohjelmaa lähdettiin muokkaamaan uuden ohjelman mukaan.

5.3 Uusi ohjelma

Uusi ohjelma ei poikennut vanhasta paljoakaan. Choice-boksin korvasi aloitusspiikki, jossa kerrotaan lyhyesti, mitä seuraavaksi tapahtuu, joten lupaa aloittamiseen ei tarvita. Jokaisessa aloitusspiikissä kokonaisuus tiivistetään, kuten esimerkiksi lähtövalmistelut sisältävät pakkaamisen ja kauppaan lähtemisen. Kuulija saa kokonaiskuvan seuraavista tulevista lauseista spiikkien avulla. Spiikkien aikana näytetään myös seuraavan lauseen kuva. Molempien lauseiden kuvia ei saatu aloitusspiikkiin toteutettua ajoituksen vaikeuden vuoksi. Lauseet ovat suurin piirtein samat ja niiden välissä on aina muuttaman sekunnin tauko. Alla kuvataan uuden ohjelman rakenne.

Uusi ohjelma:

- I. LÄHTÖVALMISTELUT -> (ruudulla koko STRIPPI 1–2 kuvat) Spiikki: Ensin lähtövalmistelut. Ele esim. sormi pystyssä.

Pepper sanoo: ”Ensin lähtövalmistelut” elehtien ja tabletilla näkyy kuva Pepperistä, kun Pepper kirjoittaa listaan, mitä kaupasta tarvitaan. Toteutus löytyy liitteestä 5.

1. Spiikki: Tee kauppalista.

Pepper sanoo: ”Tee kauppalista” elehtien ja tabletilla näkyy edellinen kuva. Toteutus löytyy liitteestä 6.

2. Spiikki: Ota mukaan kauppalista, kauppakassi, rahapussi ja avaimet.

Pepper sanoo: ”Ota mukaan kauppalista, kauppakassi, rahapussi ja avaimet.” Tabletille näkyy kylä, jossa Pepper ottaa nämä mukaansa. Toteutus löytyy liitteestä 7.

- II. OSTOKSILLA -> (ruudulla koko STRIPPI 3–6 kuvat) Spiikki: Sitten ostoksille. Ele esim. sormi pystyssä.

Pepper sanoo: ”Sitten ostoksille” elehtien. Tabletille näkyy Pepperistä kuva, kun hän on odottamassa liikennevaloissa ja menossa kauppaan. Toteutus löytyy liitteestä 8.

3. Spiikki: Kulje kauppaan liikennesääntöjä noudattaen.

Pepper sanoo: ”Kulje kauppaan liikennesääntöjä noudattaen.” Tabletilla näkyy edellinen kuva. Toteutus löytyy liitteestä 9.

4. Spiikki: Jos sinulla on **pienet** ostokset, ota **kori**. Jos sinulla on **suuret** ostokset, ota **kärri**.

Pepper sanoo: ”Jos sinulla on pienet (viitotaan pienet) ostokset, ota kori (viitotaan kori). Jos sinulla on suuret (viitotaan suuret) ostokset, ota kärri (viitotaan kärri).” Tabletilla esitetään kuva Pepperistä pohtimassa, ottaisiko korin vai kärri. Toteutus löytyy liitteestä 10 ja 11.

5. Spiikki: **Kerää kaikki** tuotteet **koriin** tai **kärriin**.

Pepper sanoo: ”Kerää kaikki (viitotaan kerätä ja kaikki) tuotteet koriin (viitotaan kori) tai kärriin (viitotaan kärri).” Tabletilla esitetään kuva, missä ostokset on kerätty kärriin. Toteutus löytyy liitteestä 12.

6. Spiikki: Jos **et** löydä **kaikkea** tai jokin tuote on **loppu**, **pyydä apua** myyjältä.

Pepper sanoo: ”Jos et (viitotaan ei) löydä kaikkea (viitotaan kaikki) tai jokin tuote on loppu (viitotaan loppu), pyydä apua (viitotaan pyytää ja apu) myyjältä.” Tabletilla esitetään kuva, jossa Pepper pohtii jonkin tuotteen sijaintia ja myyjä on lähettyvillä, jos tarvitaan apua. Toteutus löytyy liitteestä 13 ja 14.

- III. KASSALLA -> (ruudulla koko STRIPPI 7-10 kuvat) Spiikki: Nyt kassalle. Ele esim. sormi pystyssä.

Pepper sanoo: ”Nyt kassalle” elehtien. Tabletilla esitetään kuva, jossa Pepper seisoo kassajonossa. Toteutus löytyy liitteestä 15.

7. Spiikki: Kun **kaikki** ostokset on **kerätty**, **mene kassajonoon**.

Pepper sanoo: ”Kun kaikki (viitotaan kaikki) ostokset on kerätty (viitotaan kerätä), mene kassajonoon (viitotaan mene ja jono).” Tabletilla esitetään edellinen kuva. Toteutus löytyy liitteestä 16.

8. Spiikki: Kun on sinun vuorosi, **nosta** tuotteet kassalle.

Pepper sanoo: ”Kun on sinun vuorosi, nosta (viitotaan nostaa) tuotteet kassalle.”

Tabletilla esitetään kuva elintarvikkeista kassahihnalla. Toteutus löytyy liitteestä 17.

9. Spiikki: Maksa ostokset.

Pepper sanoo: ”Maksa ostokset” elehtien. Tabletilla esitetään kuva maksukorttilukijasta ja Pepper maksaa lähimaksulla. Toteutus löytyy liitteestä 18.

10. Spiikki: Pakkaa ostokset kauppakassiin.

Pepper sanoo: ”Pakkaa ostokset kauppakassiin.” Tabletilla esitetään kuva Pepperistä, joka pakkaa ostoksia reppuun. Toteutus löytyy liitteestä 19.

IV. KOTIINPALUU -> (ruudulla koko STRIPPI 11–12 kuvat) Spiikki: Sitten takaisin kotiin. Ele esim. sormi pystyssä.

Pepper sanoo: ”Sitten takaisin kotiin” elehtien. Tabletilla esitetään kuva Pepperistä liikennevaloissa palaamassa kotiin kauppakassin kanssa. Toteutus löytyy liitteestä 20.

11. Spiikki: **Mene kotiin** liikennesääntöjä noudattaen.

Pepper sanoo: ”Mene kotiin (viitotaan mene ja koti) liikennesääntöjä noudattaen.”

Tabletilla esitetään edellinen kuva. Toteutus löytyy liitteestä 21.

12. Spiikki: Pura ostoskassit heti. Laita etenkin jääkaappiin ja pakastimeen menevät ruoat heti paikoilleen.

Pepper sanoo: ”Pura ostoskassit heti. Laita etenkin jääkaappiin ja pakastimeen menevät ruoat heti paikoilleen.” Tabletilla esitetään kuva Pepperistä laittamassa ruoat jääkaappiin. Toteutus löytyy liitteestä 22 ja 23.

V. LOPPUSANAT -> (ruudulla koko STRIPPI 13 kuva)

Tätä spiikkiä ei esitellä millään tavalla, vaan ohjelma hyppää tauon jälkeen suoraan lauseeseen 13.

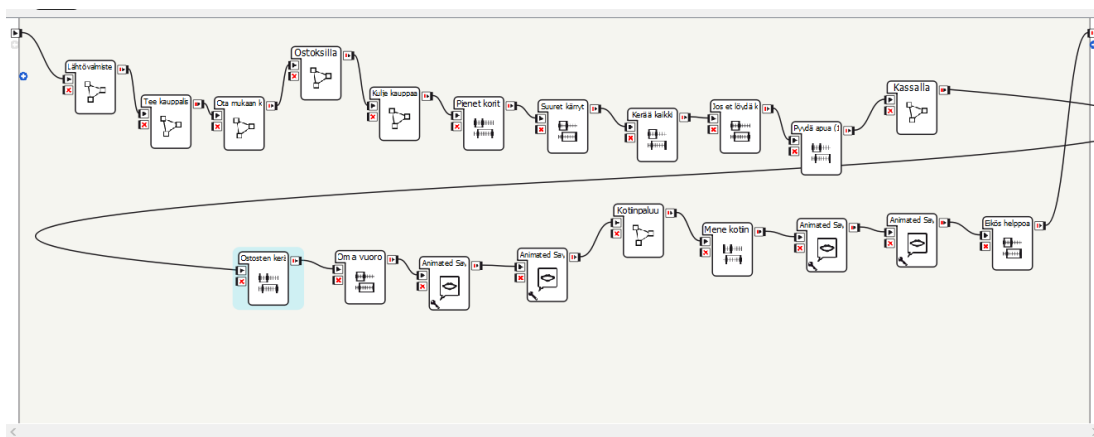
13. Spiikki: Eikös ollutkin helppoa? Nyt sen tiedät!

Pepper sanoo: ”Eikös ollutkin helppoa (viitotaan helppo)? Nyt sen tiedät! (viitotaan nyt ja tietää)”. Tabletilla esitetään kuva iloisesta Pepperistä. Liite 24.

Ohjelma on toteutettu Flow diagram -bokseja hyödyntäen, jotta saadaan useita toiminnallisuuksia toteutettua samaan aikaan. Boksi saadaan luotua kulkukaavioille hiiren oikeaa näppäintä painamalla, create new box ja valitaan Flow diagram. Animated Say -boksia käytetään lauseisiin, joissa ei ole tukiviittomia.

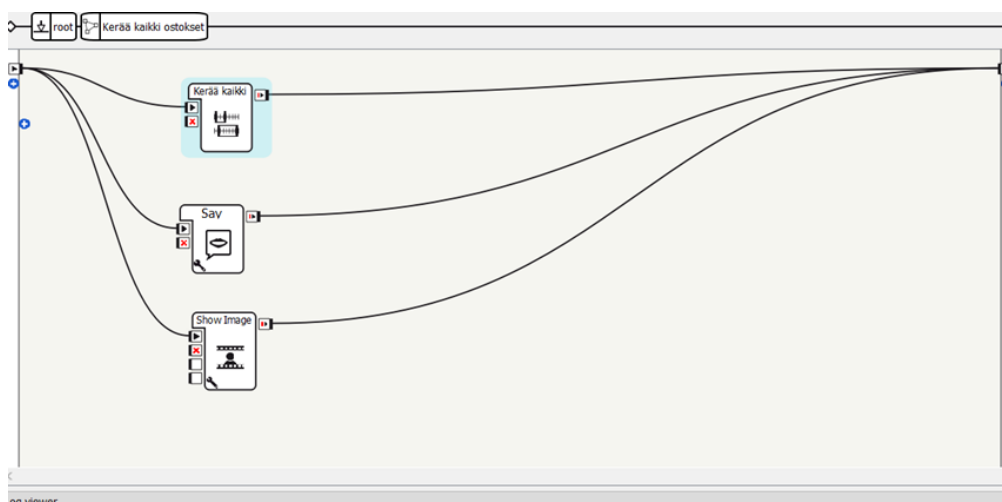
Spiikkien alussa on tarkoitus näyttää kuva siitä, mitä tulee tapahtumaan. Spiikin jälkeen kuvat esitetään erikseen lauseisiin liittyen. Lisätään show image -boksi kulkukaavioon Animated Say -boksin rinnalle ja kirjoitetaan kuvan polku parametreihin. Jotta saadaan lause ja kuva toteutettua samaan aikaan, yhdistetään molemmat boksit lähtöön ja tuloon. Lähdöstä lähtee signaali molempiin bokseihin, joten toiminnot suoritetaan samaan aikaan.

Vanhasta ohjelmasta kopioitiin Timeline-boksit, koska lauseet eivät muuttuneet merkittävästi ja ne sisälsivät samat tukiviittomat. Lauseet ilman tukiviittomia toteutettiin myös flow diagram -boksia hyödyntäen ja sijoitettiin Animated Say- ja Show Image -boksit sen sisään rinnakkain. Kaikki boksit yhdistettiin pääkulkukaaviossa (kuva 72).

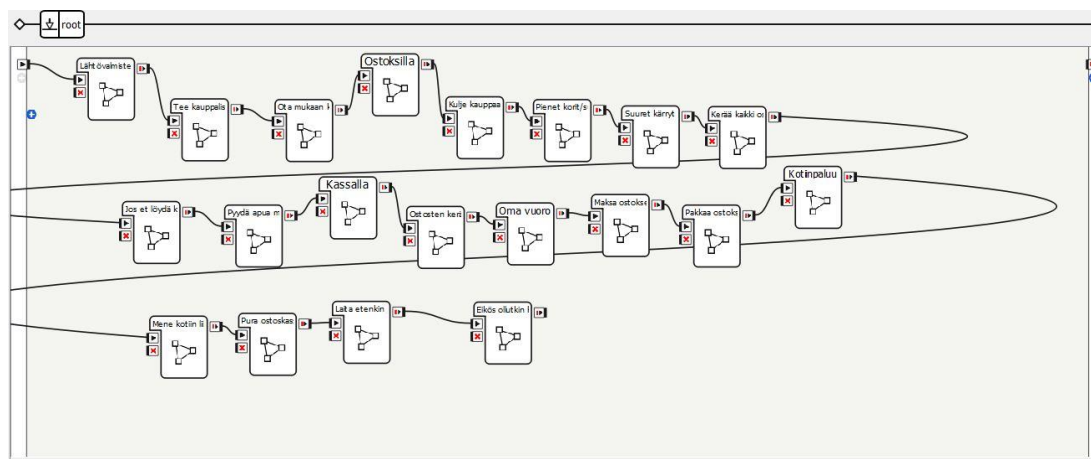


Kuva 72. Ohjelma kokonaisuudessaan.

Kesken ohjelmanteon Pepperin ohjelmisto päivittyi, mikä vaikutti timeline-boksien translation-tiedostoihin. Pepper ei suorittanut liikkeitä, puhetta eikä näyttänyt kuvaa. Softbankin tukeen otettiin yhteyttä ja henkilöstö testasi ohjelmaa, mutta ei osannut ratkaista ongelmaa. Softbankin mukaan syynä saattoi olla se, että päivityksen myötä Pepper ei tue kokonaan suomen kieltä. Toimintokerros ”rikkoi” käännökset ja siitä syystä se ei hakenut tekstiä käännöstiedostosta. Tämän seurauksena poistettiin Say- ja Show Image -laatikko Timelinen sisältä. Luotiin Flow diagram -laatikko, minkä sisään liitettiin erikseen timeline-, say- ja how image -boksi rinnatusten samalle kulkukaaviolle (kuva 73). Pääkulkukaavion lopullinen ohjelma esitetään alla (kuva 74.)



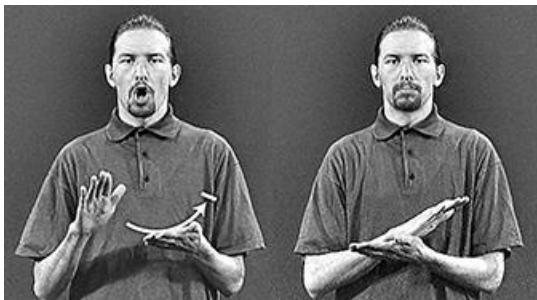
Kuva 73. Flow Diagram Timeline-laatikon yhteydessä.



Kuva 74. Lopullinen ohjelma.

5.4 Ongelmakohdat ja ratkaisut

Ohjelman tuloksista nähdään, että Pepper pystyy viittomaan vain osan tukiviittomista. Monissa viittomissa Pepperin tabletti oli tiellä, jolloin joidenkin tukiviittomien liikeradat jäivät vajavaisiksi. Esimerkiksi ”loppu”-viittomassa (kuva 77) käsiä ei saa taivutettua yhteen ja ohi toisistansa tabletin ja nivelten liikkumisen rajallisuuden vuoksi.



Kuva 77. Loppuviittoma (Suvi viittomien www-sivut 2013)

Apu-viittomassa (kuva 78) Pepperin käsiä ei myöskään saada taivutettua tarpeeksi lähelle ja ylös. Käsien nivelten liikkumisen rajallisuus tuli esiin muissakin viittomissa.



Kuva 78. Apuviittoma. (Suvi viittomien www-sivut 2013)

Esimerkiksi kerätä-viittomassa Pepperin ranne ei taivu ja se kykenee vain pyörähtämään. Siitä syystä kerätä-tukiviittoma toteutettiin niin, että Pepperin käsi on sivussa ja ”kerää” toiseen käteen. Pepper-robotin sormet eivät varsinaisesti liiku, ainoastaan peukalo on erillinen ja muut sormet yhtenevät. Se myös vaikeuttaa joidenkin tukiviittomien toteutusta. Esimerkiksi jono-viittomassa ei saada sormia erotettua.

Pepper viittoo pieni-, suuri-, kori- ja kärryt -tukiviittoman selkeästi ja ymmärrettävästi. Muut viittomat toteutettiin lähes luonnollisesti, mutta yleisesti Pepper-robotin oli ihmisiin verrattuna hankalampi toteuttaa tukiviittomia.

Tukiviittomien opetustilanteessa ilmeni myös ongelmia. Pepperin pää vapautui opetustilanteessa, vaikka animation mode oli päällä. Kun animointi-tilasta poistuttiin, pää jäi silti vapaaksi, vaikka sen pitäisi lukittua. Tästä syystä Pepper ei suostunut suorittamaan ohjelmaa, eikä se ottanut käskyjä vastaan. Tilannetta oli hankala korjata, joten Pepper laitettiin ”nukkumaan” ja se herätettiin uudestaan. Opetustilanteissa animaatiotilassa Pepper meni itsenäisesti autonomous life -tilaan, se ei ”keskittynyt” tehtävään, vaan se alkoi tarkastella ympäristöä. Tämä hankaloitti tukiviittomien opettamista, koska robotti liikkui. Pepper laitettiin ”nukkumaan” ja herätettiin uudestaan.

Choregraphe-ohjelmassa oli hankaluuksia. Ohjelma toimi ajoittain. Välillä Pepper ei lähtenyt suorittamaan ohjelmaa. Erityistä ongelmaa ei kuitenkaan löytynyt, ja ratkaisuna poistettiin input-viivat ja asetettiin ne uudestaan laatikkoon. Välillä Pepper-robotti ei puhunut ja liikkeet jäivät kesken.

Ratkaisuna kopioitiin English translation -koodi ja purettiin finnish translation -koodin päälle. Ongelmana saattoi olla, ettei Timeline-toimintokerros hakenut käännöstä oikeasta tiedostosta. Viimeisimmässä ohjelman ratkaisussa ohjelma toimi moitteettomasti.

Timeline-laatikon toteutuksessa lisättiin uusi toimintokerros, jotta kuvan näyttäminen, puhe ja tukiviittomat saataisiin suoritettua samanaikaisesti. Toisen kerroksen kulukaavioon lisättiin show image -laatikko. Tässä ratkaisussa ilmeni kuitenkin ongelma, sillä suoritettaessa ohjelmaa kuva ehti poistua ennen puhetta. Kuvien lisäämisessä ilmeni myös ongelma, sillä Show Image -laatikko ei näyttänyt muita kuin .gif-muodossa olevia kuvia. Softbankin sivujen mukaan Show Image -laatikko tukee kaikkia kuvatiedostoja, joten kuvia piti muokata, vaikka alkuperäiset kuvat olivat tuettavassa muodossa. Softbankin henkilöstö ei osannut sanoa, miksi polku ei toiminut tai miksi se ei näyttänyt kuvaa. Choregraphe-ohjelmointiympäristö päivittyy melko usein, jolloin muutokset ovat mahdollisia ja toiminnot eivät välttämättä löydy samasta paikasta ja/tai ei toimi samalla tavalla.

6 LOPPUPÄÄTELMÄT

Tuloksista nähtiin, että Pepper-humanoidirobotin käyttöönotto tukiviittomien tulkkina on mahdollista. Pepper ei kuitenkaan kyennyt kaikkia tukiviittomia suorittamaan liikeratojen rajoitusten vuoksi. Pepperin rakenne ja ominaisuudet on kehitetty kommunikointiin, tunteiden tunnistamiseen ja ihmisten viihdyttämiseen. Tukiviittomilla kommunikoinnin mahdollisuuksiin on vielä matkaa, sillä Pepperin rakennekohtaiset rajoitukset sekä ominaisuudet eivät tue tämältyyppistä kommunikointia.

Softbank Robotics on kehittämässä Pepperille sovelluksia, joissa se soveltuisi enemmän ihmisten pariin. Tarkoitus olisi kehittää Pepper-robotille empatiakykyä, käsittelemään etiikkaa, moraalisia asioita sekä muita sellaisia ominaisuuksia, joita tarvitaan ihmisten kanssa kommunikointiin. Täten Pepper pystyisi kuuntelemaan empaattisesti asiakkaita ja parantamaan heidän oloaan sekä keskusteleminen saisi uusia ulottuvuuksia eettisten näkökulmien kautta. Lisäominaisuudet toisivat uusia mahdollisuuksia ihmisten apuna toimimiseen eri ammateissa. Se kuitenkin vaatisi suurempaa algoritmia, vaativampaa koodausta ja laskentatehoa.

Muihin sovelluksiin, kuten vanhusten auttamiseen ja esineiden poimiseen, Softbank Robotics on kehittänyt uuden robotin Romeon. Romeo on kehitetty auttamaan vanhuksia, jotka ovat menettämässä kykynsä liikkua ja toimia arjessa yksin. Robotti on noin 140 senttinen, joten pituuden puolesta sen on helpompi olla tukemassa liikkumista. Sen sormet ovat liikkuvat ja se pystyy kantamaan esineitä sekä tarttumaan niihin. Sillä on myös jalat, jolloin se kykenee nousemaan portaita ja liikkumaan erilaisessa maastossa. Pepperin tapauksessa, jos se halutaan opettaa noutamaan tavaraa tai tarttumaan esineisiin, olisi kehittämistä robotin motoriikassa. Pepperin yhtenäinen alaosaa ei mahdollista liikkumista erilaisissa maastoissa tai kallistumaan alas noutamaan esinettä. Pepperin sormet ovat myös yhtenevät, eivätkä ne kykene tarttumaan. Kamerateknologiaa pitäisi kehittää, jotta se tunnistaisi esineet, sillä nyt Pepper hakee kasvoja ja tulkitsee niitä. Muutokset vaatisivat laskentatehoa ja rakennemuutoksia robottiin.

Kehitteillä on kuitenkin ajatus siitä, että Pepper kykenisi tunnistamaan viittomakieltä ja näin kommunikoimaan myös tukiviittomien avulla. Kuitenkin Pepper-robotin kohdalla laitteistorajoitukset ovat suurempi ongelma. Pepper-robotin kamerat eivät sisällä esimerkiksi Microsoftin kinect-tyyppistä liikkeen tunnistusta ja analysointijärjestelmää, jonka avulla pystyttäisiin kolmiulotteiseen liikkeentunnistukseen. Täten Pepper kykenisi tunnistamaan kaikkien käsien sekä sormien liikkeet reaaliajassa.

Viittomissa sormien eri asennot ovat myös tärkeitä ja niiden erottaminen toisistaan vaatisi edellä mainittua kamerateknologiaa sekä laskentatehoa, jota Pepperistä ei löydy. Tulevaisuudessa ohjelmaa voidaan hyödyntää kouluissa, päiväkodeissa sekä opetustilaisuuksissa. Pepperille voidaan opettaa lisää viittomia ja vaikeammin toteuttavissa olevat viittomat voidaan kuitenkin havainnollistaa kuvien avulla Pepperin tabletilla. Ohjelmaa voidaan soveltaa myös muihin arjen askareiden opettamiseen. Ajatuksena on, että ohjelmaa testataan autististen lapsien kanssa kouluissa myöhemmin syksyllä. Pepperiä voidaan käyttää apuvälineenä opetuksessa ja mahdollisuuksien mukaan lapset voisivat opettaa Pepperille tukiviittomia.

LÄHTEET

Kyrki, V. Coco, K. Hennala, L. Laitinen, A. Lehto, P. Melkas, H. Niemelä, M. Pekkarinen, S. Tilanneraportti 2015. Viitattu 6.2.2018.

http://www.aka.fi/globalassets/33stn/tilannekuvaraportit/stn2015-hankkeet/tech-kyrki-robotiikkahyvinyvointi-jaterveyspalveluissa_20160104.pdf

Aldebaran www-sivut. 2018. Viitattu 8.2.2018.

http://doc.aldebaran.com/2-5/family/pepper_technical/contact-sensors_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/joints_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/motors_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/power_entry_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/charging_station_pep.html

Aldebaran www-sivut. 2018. Viitattu 9.2.2018.

http://doc.aldebaran.com/2-5/family/pepper_technical/loudspeaker_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/microphone_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/video_2D_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/video_3D_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/leds_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/sonar_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/laser_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/laser_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/irspot_pep.html

Aldebaran www-sivut. 2018. Viitattu 12.2.2018.

http://doc.aldebaran.com/2-5/family/pepper_technical/sonar_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/contact-sensors_pep.html

http://doc.aldebaran.com/2-5/family/pepper_technical/contact-sensors_pep.html

http://doc.aldebaran.com/2-5/family/pepper_user_guide/sleep_pep.html

http://doc.aldebaran.com/2-5/family/pepper_user_guide/activity_pep.html

Aldebaran www-sivut. 2018. Viitattu 13.2.2018.

<http://doc.aldebaran.com/2-5/software/choregraphe/index.html>

<http://doc.aldebaran.com/2-5/software/choregraphe/objects/box.html>

http://doc.aldebaran.com/2-5/software/choregraphe/connection_widget.html

<http://doc.aldebaran.com/2-5/software/choregraphe/interface.html>

<http://doc.aldebaran.com/2-5/software/choregraphe/interface.html>

http://doc.aldebaran.com/2-5/software/choregraphe/panels/box_libraries_panel.html

http://doc.aldebaran.com/2-5/software/choregraphe/objects/flow_diagram_box.html

http://doc.aldebaran.com/2-5/software/choregraphe/objects/timeline_box.html

Aldebaran www-sivut. 2018. Viitattu 14.2.2018.

http://doc.aldebaran.com/2-5/software/choregraphe/objects/python_box.html

http://doc.aldebaran.com/2-5/software/choregraphe/objects/dialog_box.html

http://doc.aldebaran.com/2-5/software/choregraphe/panels/robot_3d_view.html

http://doc.aldebaran.com/2-5/software/choregraphe/panels/flow_diagram_panel.html#choregraphe-howto-convert-boxes-to-one-box

http://doc.aldebaran.com/2-5/software/choregraphe/panels/flow_diagram_panel.html#choregraphe-howto-convert-boxes-to-one-box

http://doc.aldebaran.com/2-5/software/choregraphe/panels/flow_diagram_panel.html#choregraphe-howto-convert-boxes-to-one-box

http://doc.aldebaran.com/2-5/software/choregraphe/panels/flow_diagram_panel.html#choregraphe-howto-convert-boxes-to-one-box

http://doc.aldebaran.com/2-5/software/choregraphe/panels/flow_diagram_panel.html#choregraphe-howto-convert-boxes-to-one-box

Aldebaran www-sivut. 2018. Viitattu 15.2.2018.

http://doc.aldebaran.com/25/software/choregraphe/panels/timeline_panel.html#time-line-panel

http://doc.aldebaran.com/2-5/software/choregraphe/panels/timeline_panel.html#time-line-panel

http://doc.aldebaran.com/2-5/software/choregraphe/panels/timeline_panel.html#time-line-panel

<http://doc.aldebaran.com/2-5/software/choregraphe/tutos/movements.html#choregraphe-tuto-animation-mode>

http://doc.aldebaran.com/2-5/software/choregraphe/panels/timeline_panel.html

http://doc.aldebaran.com/25/software/choregraphe/panels/robot_3d_view.html#choregraphe-detailed-interface-motion-widget

http://doc.aldebaran.com/2-5/software/choregraphe/animation_mode.html

Aldebaran www-sivut. 2018. Viitattu 14.3.2018.

(<http://doc.aldebaran.com/2-5/software/choregraphe/interface.html>)

(<http://doc.aldebaran.com/2-5/software/choregraphe/interface.html>)

(http://doc.aldebaran.com/2-5/software/choregraphe/panels/robot_applications.html)

(http://doc.aldebaran.com/2-5/software/choregraphe/panels/robot_applications.html)

Aldebaran www-sivut. 2018. Viitattu 19.3.2018.

<http://doc.aldebaran.com/2-5/software/choregraphe/troubleshooting.html>

([http://doc.aldebaran.com/2-](http://doc.aldebaran.com/2-5/software/choregraphe/objects/python_script.html#python)

[5/software/choregraphe/objects/python_script.html#python](http://doc.aldebaran.com/2-5/software/choregraphe/objects/python_script.html#python))

Eagledatan www-sivut. 2016. Viitattu 3.2.2018. (<http://eagledata.fi/beam-plus-telepresence-robotti.html>)

Eagledatan www-sivut. 2016. Viitattu 3.2.2018. (<http://eagledata.fi/tug-kuljetusrobotti.html>)

Evondoksen www-sivut. Viitattu 2.2.2018. (<https://evondos.fi/automaattinen-laakkeiden-annostelupalvelu/evondos-e300-laakeautomaatti/>)

Finncleanin www-sivut. Viitattu 5.2.2018.

(https://www.finnclean.fi/fi/product_or_service/taski-intellibot/)

Folkhansan www- sivut. Viitattu 15.3.2018.

(<https://www.folkhalsan.fi/fi/landing/indego-kavelyrobotti/artikel-om-indego/>)

Gimltd:n www-sivut. Viitattu 5.2.2018. (<http://gimltd.fi/platforms.html>)

Glad, Tomi. Samkin www-sivut. 2017. Viitattu 15.3.2018.

(<http://www.samk.fi/uutiset/avoimet-ovet-3d-tulostusta-robotteja-peleja/>)

Iltalehden www-sivut. 2013. Viitattu 31.1.2018.

(http://www.iltalehti.fi/iltvuutiset/20130131019168783_v0.shtml)

Innohoivan www- sivut. Viitattu 31.1.2018.

(<http://www.innohoiva.fi/tuote/paro-hyljerobotti/>)

Kähkönen, H. 2016. Robotit. Viitattu 5.2.2018.

(<https://www.mikrobitti.fi/2016/11/ihan-oikeita-robotteja-nama-ovat-tulevaisuuden-kumppaneitamme>)

Meditas Oy: www-sivut. Viitattu 2.2.2018.

(http://meditas.fi/data/documents/MeditasUV-DRobotti_esite.pdf)

Meditas Oy:n www-sivut. Viitattu 2.2.2018.

(<http://meditas.fi/tuotteet/uv-desinfiointirobotti/>)

Meditas Oy:n www-sivut. Viitattu 3.2.2018.

(<https://www.meditas.fi/tuotteet/beam/>)

Messukeskus www-sivut. 2016. Viitattu 5.2.2018. (<https://messukeskus.com/press-release/open-doors-robots/?lang=en>)

Miniwebin www-sivut. Viitattu 28.1.2018.

(http://miniweb.lpt.fi/automaatio/opetus/luennot/pdf_tiedostot/Robotiikka_yleinen.pdf)

Omronin www-sivut. 2018. Viitattu 22.4.2018

<https://industrial.omron.fi/fi/products/mobile-robot>

Reese, H. Robots. 2016. Viitattu 7.2.2018.

(<https://www.techrepublic.com/article/pepper-the-robot-the-smart-persons-guide/>)

Salmi, T. 2014. Robotiikka. Viitattu 28.1.2018.

(<http://www.vtt.fi/Impulssi/Pages/Robotiikka-%E2%80%93-monien-mahdollisuuksien-tekniikka.aspx>)

Stty:n www-sivut. 2016. Viitattu 6.2.2018.

(https://stty.org/images/stories/SoTeTiTe_RobotiikanHaasteista_23_05_16.pdf)

Suvi viittomien www-sivut. 2013. Viitattu 1.3.2018.

http://suvi.viittomat.net/wordsearch.php?a_id=115&word_search=loppu&offset=0&sssf=0&mpw=1

http://suvi.viittomat.net/wordsearch.php?a_id=189&word_search=apu&offset=0&sssf=0&mpw=1

Turun sanomien www-sivut. 2017. Viitattu 2.2.2018.

(<http://www.ts.fi/uutiset/paikalliset/3644706/Kotihoidon+ensimmaiselle+laakeannost+elurobotille+kehuja>)

Uas journalin www-sivut. 2012. Viitattu 6.2.2018.

(<https://uasjournal.fi/tag/palvelurobotit/>)

Wikipedian www-sivut. Nao robot. Viitattu 6.2.2018.

([https://en.wikipedia.org/wiki/Nao_\(robot\)](https://en.wikipedia.org/wiki/Nao_(robot)))

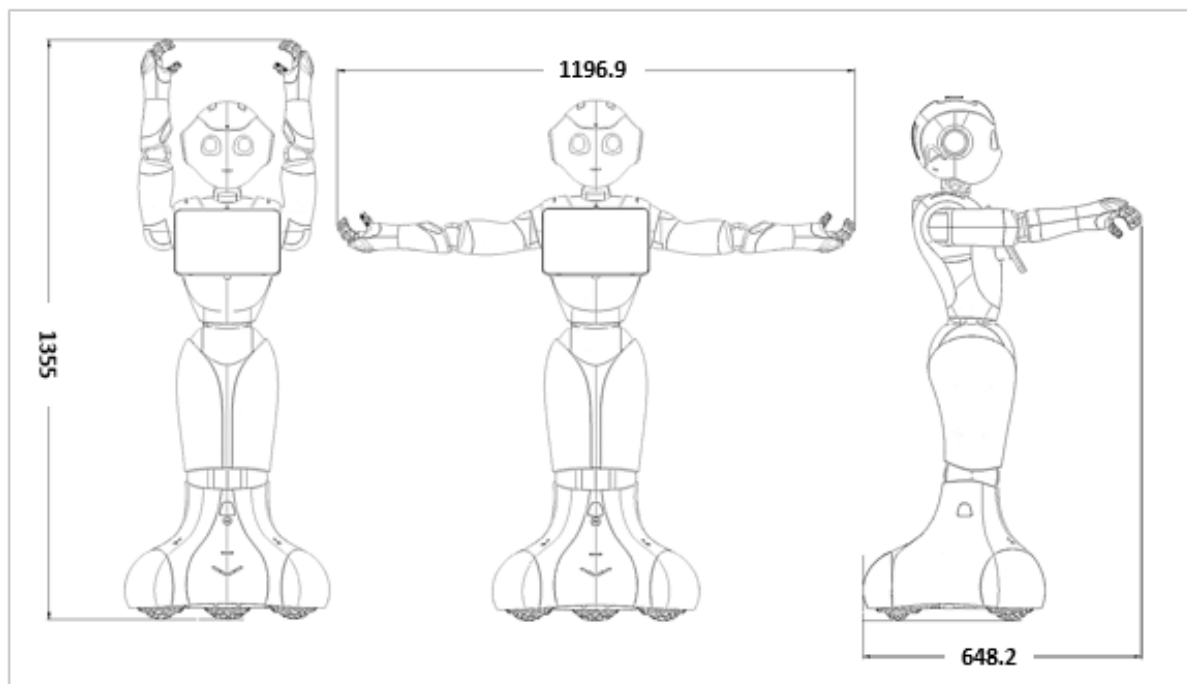
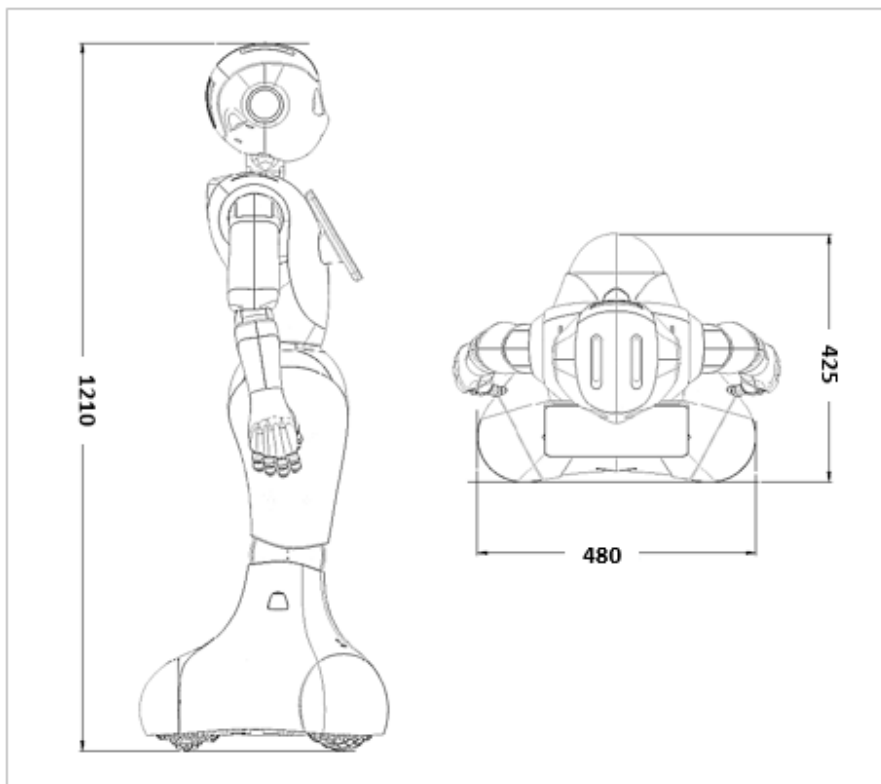
Wikipedian www-sivut. Pepper robot. Viitattu 7.2.2018.

([https://en.wikipedia.org/wiki/Pepper_\(robot\)](https://en.wikipedia.org/wiki/Pepper_(robot)))

Yli-Parkas, H .2015. Turun sanomien www-sivut. Viitattu 6.2.2018.

(<http://hyvinvointi.ts.fi/terveys/robotit-vanhusten-jumppaohjaajiksi/>)

PEPPERIN MITAT



(Aldebaran www-sivut 2018)

MOOTTORIEN TYYPIT

Motors

	Motor type 1	Motor type 2	Motor type 3	Motor type 4	Motor type 5
Family	Brush DC Coreless			BLDC (Brushless DC)	
Model	SE24P2CTCA	SE24M1RTCA	17N88208E	EC-45_50W	EC45_70W
No load speed (rpm)	5 715	8 459	8 400	4 730	6 110
Torque constant (mNm/A)	27.5	19.2	20.1	47.5	36.9
Stall torque (mNm)	122.9	61.7	9,4	402	1460
Max Continuous torque (mNm)	42.9	23.8	4.9	69.6	128
Speed Reduction ratio - Type A	101.3	118.1	50.61	108.33	108.33
Speed Reduction ratio - Type B			36.24		25

Head

Joint	Motor	Reduction ratio
HeadYaw	Type 1	Type A
HeadPitch	Type 2	Type A

Arms

Joint	Motor	Reduction ratio
ShoulderPitch	Type 1	Type A
ShoulderRoll	Type 2	Type A
ElbowYaw	Type 1	Type A
ElbowRoll	Type 2	Type A

Hands

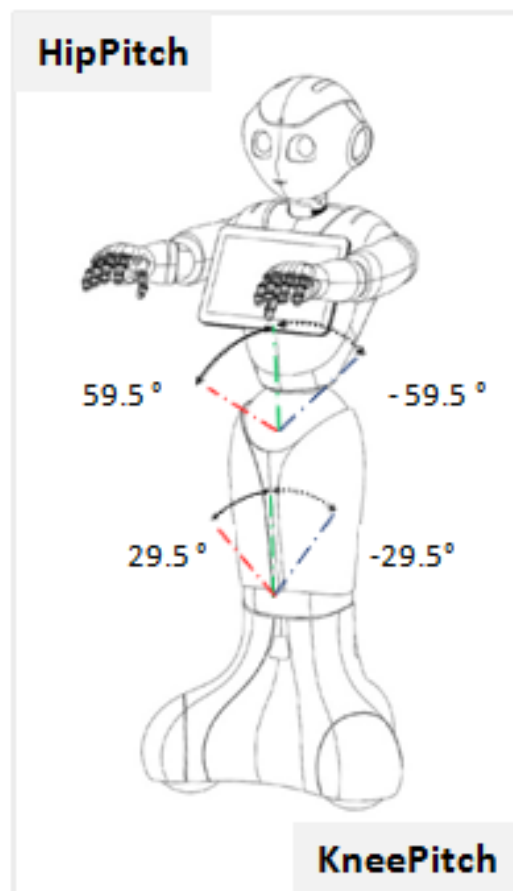
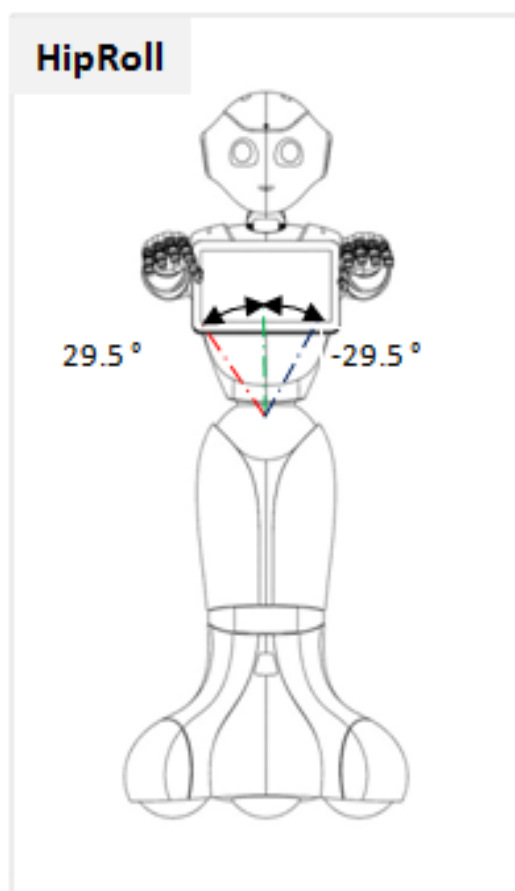
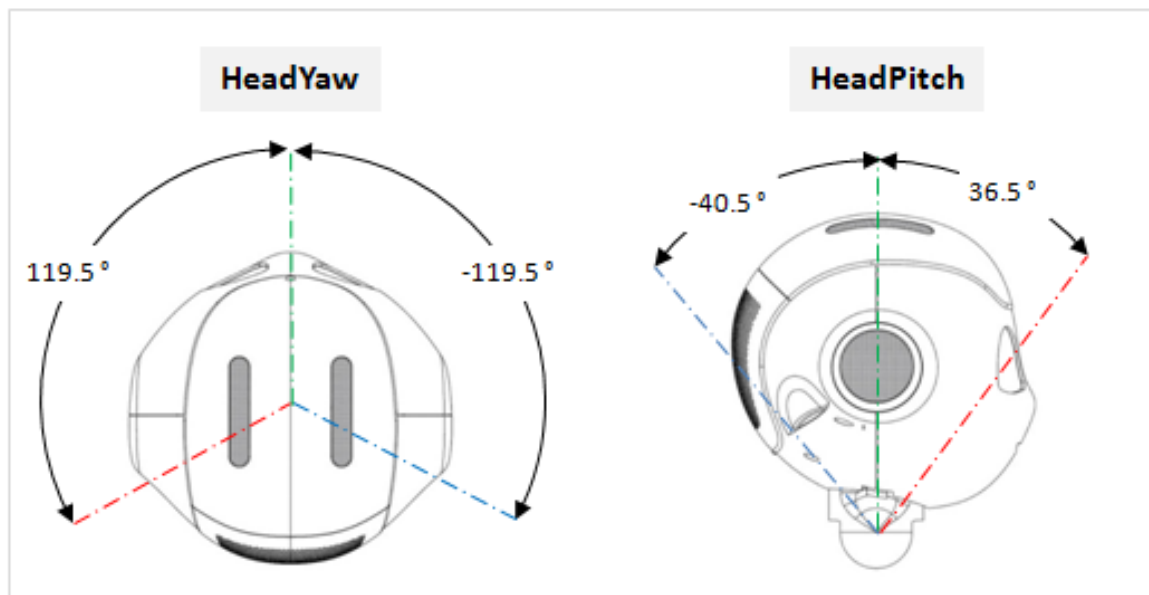
Joint / Actuator	Motor	Reduction ratio
WristYaw	Type 3	Type A
Hand	Type 3	Type B

Leg

Joints	Motor	Reduction ratio
HipRoll	Type 4	Type A
HipPitch	Type 5	Type A
KneePitch	Type 5	Type A
WheelFL	Type 5	Type B
WheelFR	Type 5	Type B
WheelB	Type 5	Type B

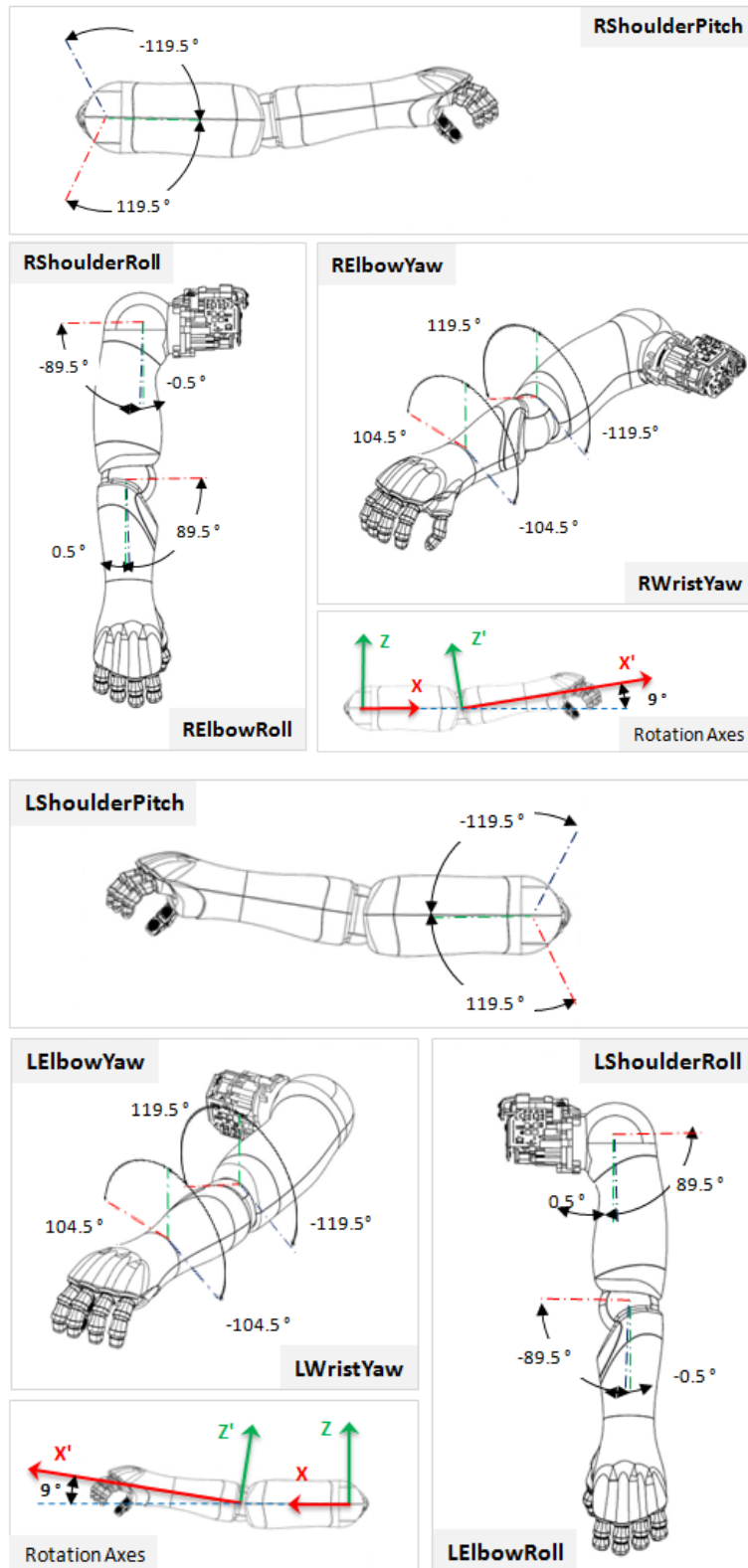
(Aldebaran www-sivut 2018)

PÄÄN JA KESKIVARTALON TAIPUMISKULMAT



(Aldebaran www-sivut 2018)

KÄSIEN TAIPUMISKULMAT



(Aldebaran www-sivut 2018)

SPIIKKI 1

C Set parameters of Animated Say ? X

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

Speaking movement mode

Animation

Auto-update parameters on robot

root Lähtövalmistelut

LAUSE 1

C Set parameters of Animated Say ? X

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

Speaking movement mode

Animation

Auto-update parameters on robot

root Tee kauppalista

The diagram below the dialog shows a sequence of two actions: 'Animated Say' and 'Show Image'. The 'Animated Say' block is highlighted with a blue selection box. Both blocks are connected to a central timeline with curved lines indicating their duration and sequence.

LAUSE 2

The image shows a software interface with two parts. The top part is a dialog box titled "Set parameters of Animated Say". It contains several settings: "Voice shaping (%)" is a slider set to 100; "Speed (%)" is a slider set to 100; "Text" is a text field containing "Ota mukaan kauppalista, kauppakassi, rahapussi ja avaimet.\pau=1500\"; "Speaking movement mode" is a dropdown menu set to "disabled"; "Animation" is a text field containing "Stand/Gestures/Explain_11". There is a checked checkbox for "Auto-update parameters on robot" and a "Reset to default" button. At the bottom are "OK" and "Cancel" buttons.

The bottom part is a timeline diagram. It shows a sequence of events on a horizontal axis. The first event is a box labeled "root" with a sub-label "Ota mukaan kauppa...". This is followed by two curved lines representing transitions. The first transition leads to an "Animated Say" block, which contains a speech bubble icon. The second transition leads to a "Show Image" block, which contains an icon of a person standing. The timeline starts and ends with a play button icon.

SPIIKKI 2

The image shows a software interface with a dialog box titled "Set parameters of Animated Say" and a timeline below it.

Dialog Box: Set parameters of Animated Say

- Parameters:**
 - Voice shaping (%): Slider set to 100.
 - Speed (%): Slider set to 100.
 - Text: Sitten ostok-sille. \pau=1000\
 - Speaking movement mode: disabled
 - Animation: Stand/Gestures/Far_2
- Auto-update parameters on robot
- Reset to default
- OK
- Cancel

Timeline:

- Timeline title: root - Ostok-silla
- Timeline contains two blocks:
 - Animated Say
 - Show Image

LAUSE 3

C Set parameters of Animated Say ? X

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

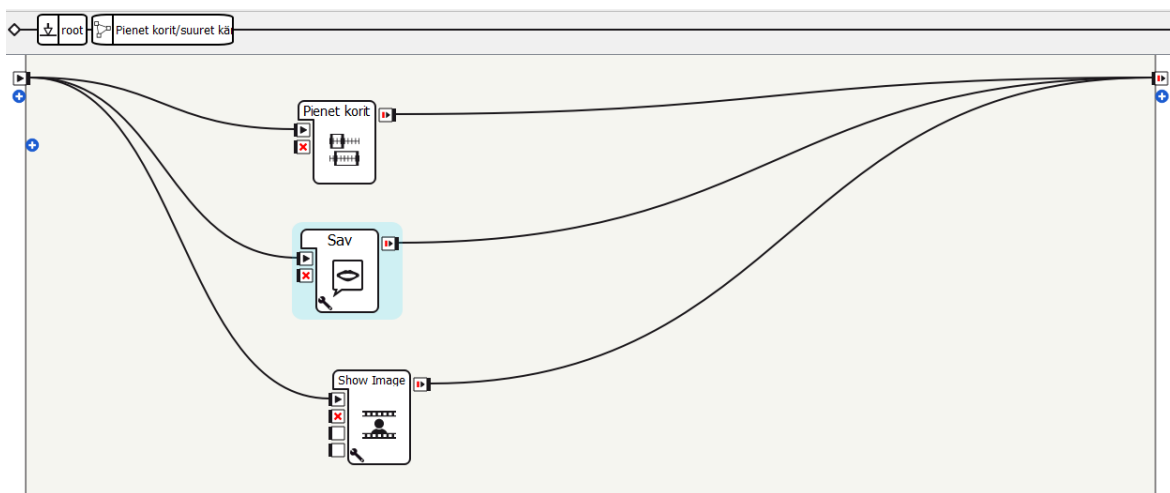
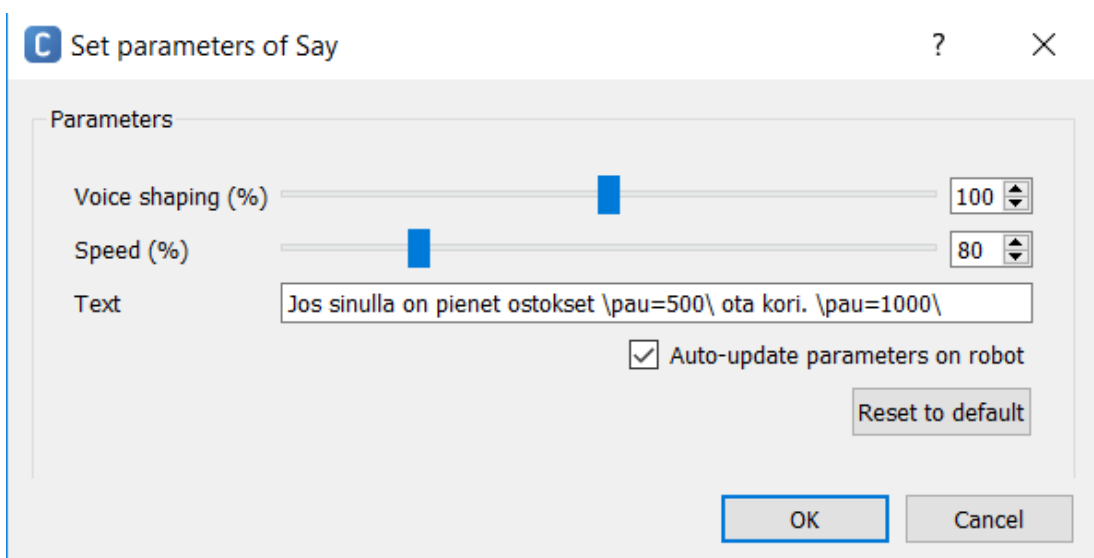
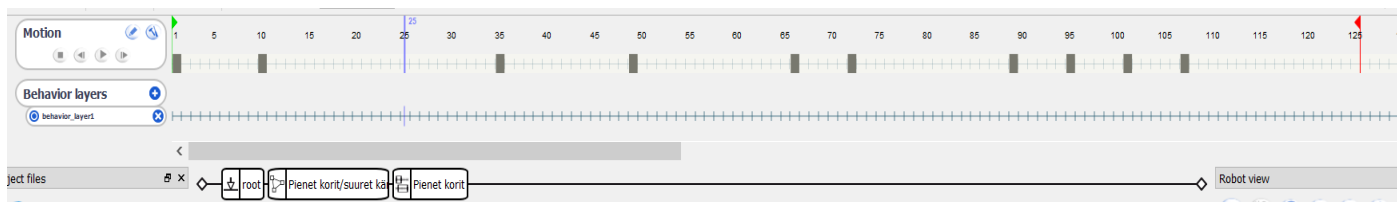
Speaking movement mode

Animation

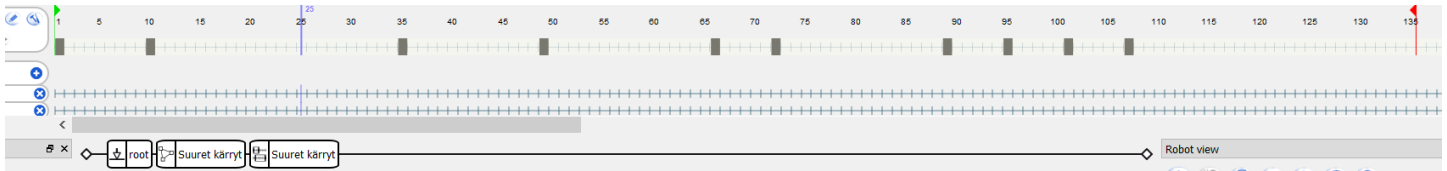
Auto-update parameters on robot

root | Kulje kauppaan liikeni

LAUSE 4



LAUSE 4 JATKUU



Set parameters of Say

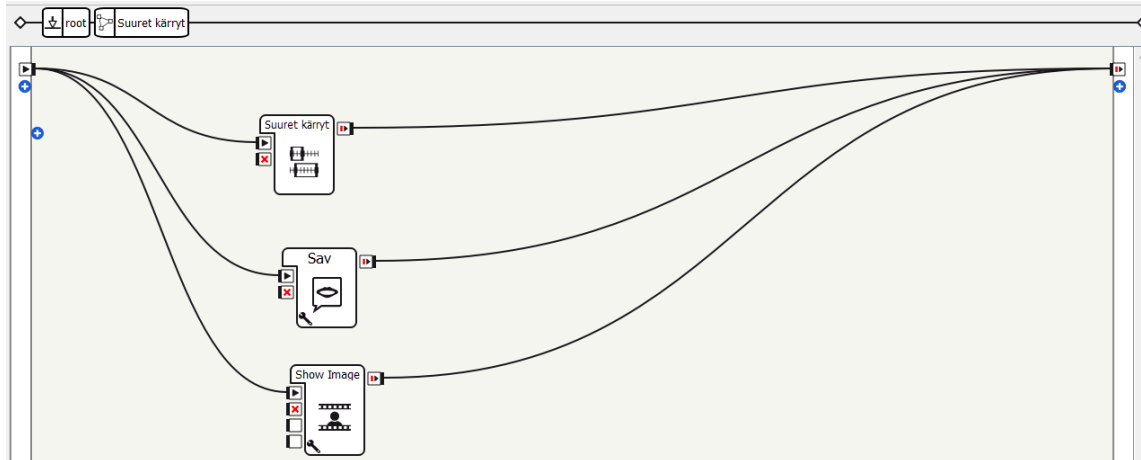
Parameters

Voice shaping (%) 100

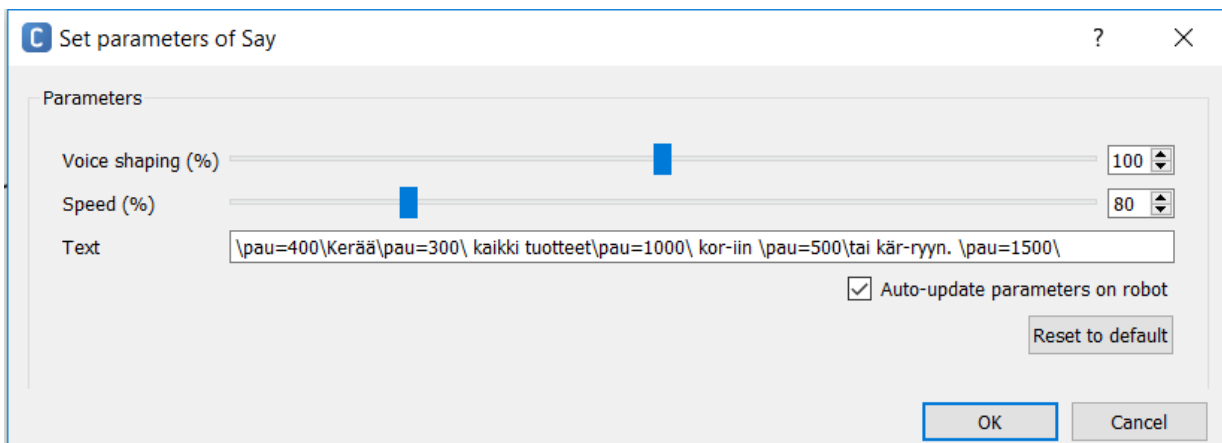
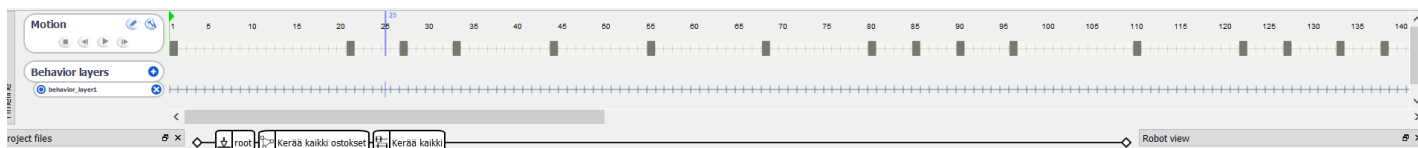
Speed (%) 80

Text Jos sinulla on suuret ostokset \pau=500\ ot-a\pau=100\ kär-ryt. \pau=1500\

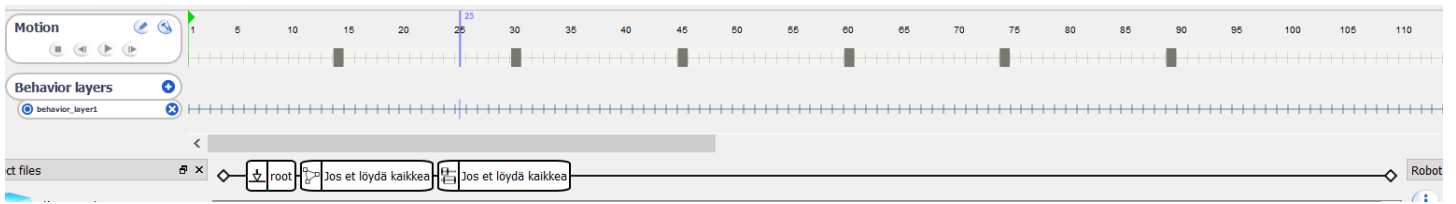
Auto-update parameters on robot



LAUSE 5



LAUSE 6



Set parameters of Say

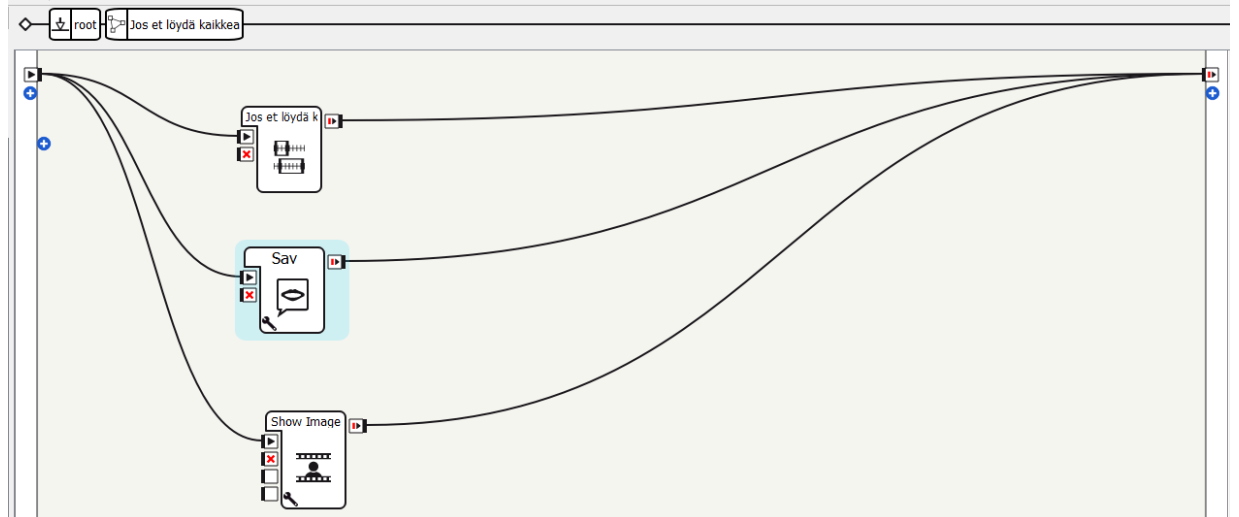
Parameters

Voice shaping (%) 100

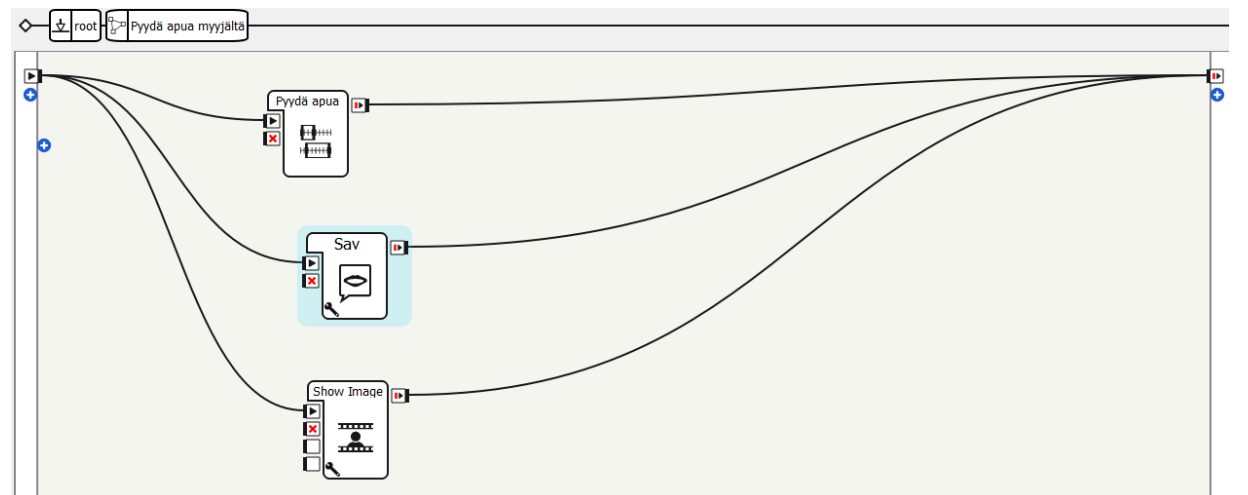
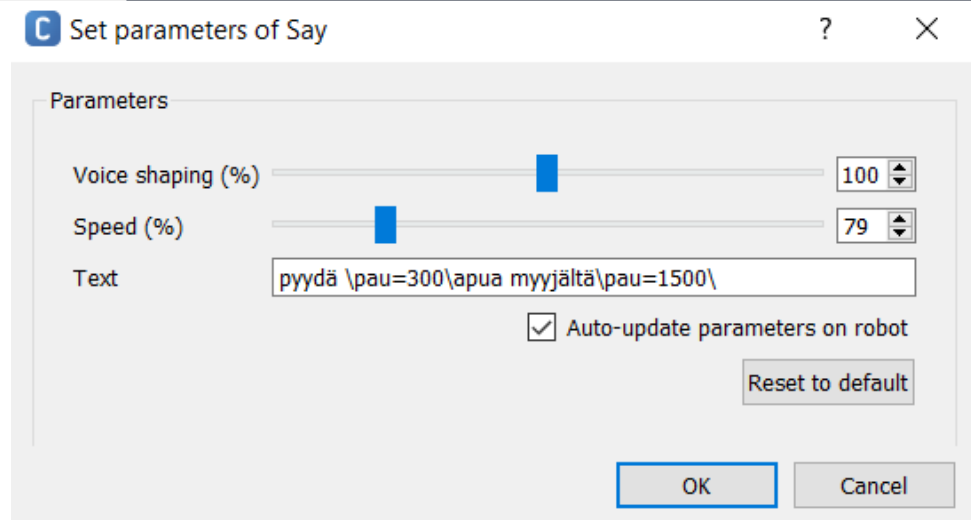
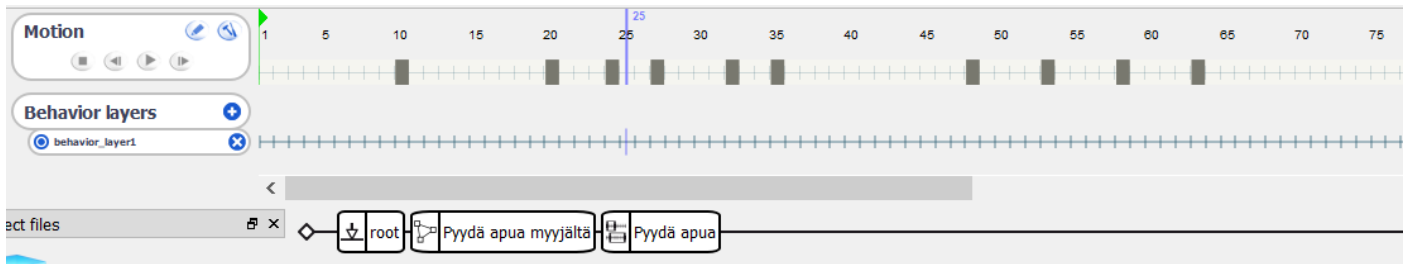
Speed (%) 79

Text Jos et löy-dä kaikkea\pau=500\tai jokin tuote on loppu\pau=500\

Auto-update parameters on robot



LAUSE 6 JATKUU



SPIIKKI 3

C Set parameters of Animated Say ? X

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

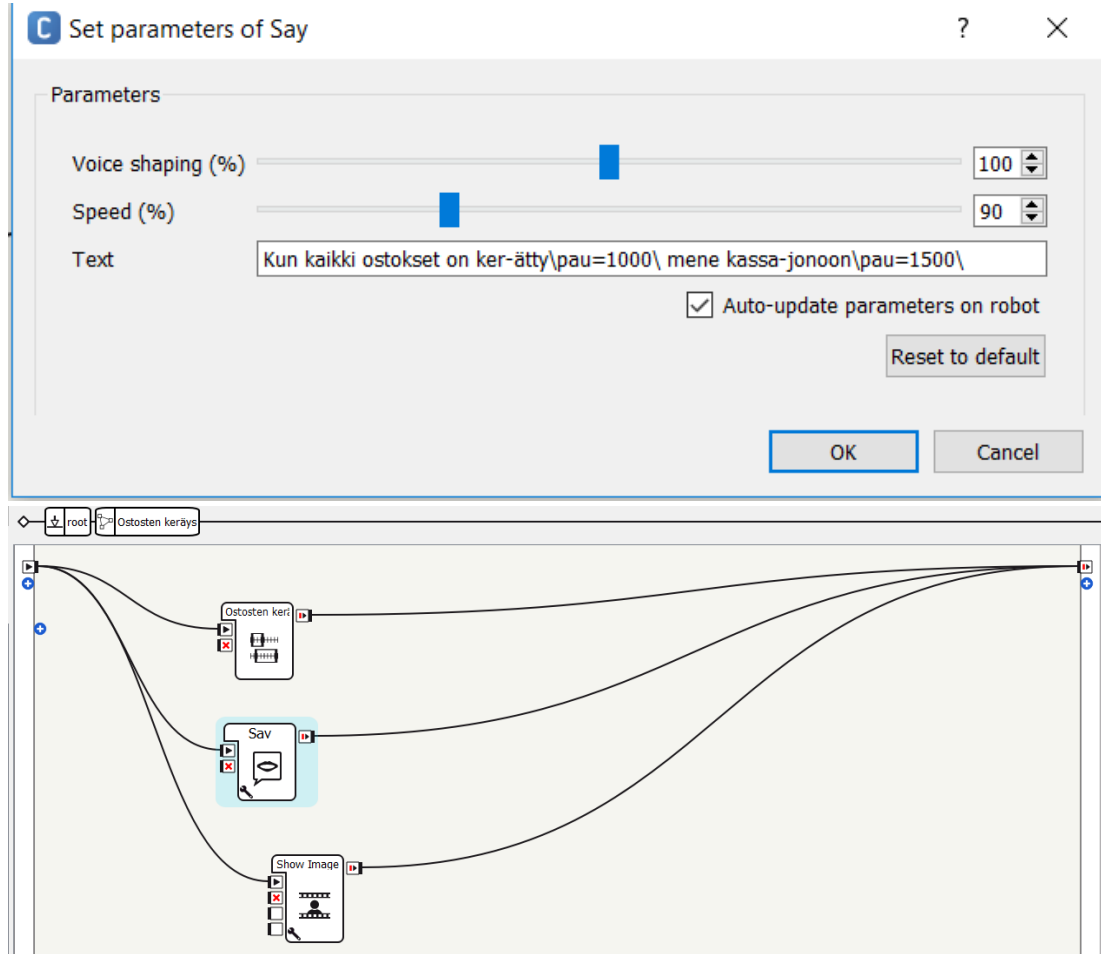
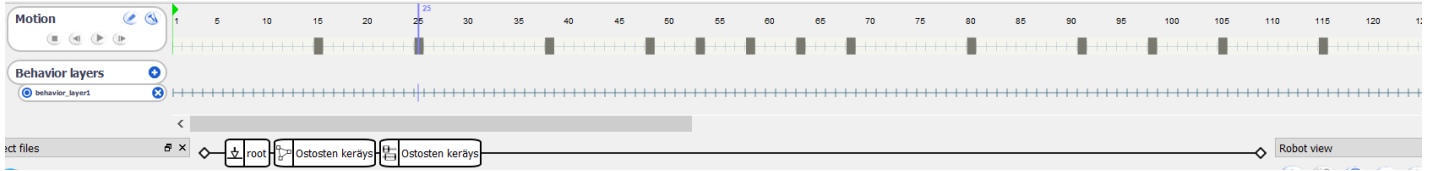
Speaking movement mode

Animation

Auto-update parameters on robot

The image shows a software interface for configuring a robot's speech. The top part is a dialog box titled "Set parameters of Animated Say". It contains several controls: two sliders for "Voice shaping (%)" and "Speed (%)" both set to 100; a text field for "Text" containing "Nyt kass-alle. \pau=1000\"; a dropdown menu for "Speaking movement mode" set to "disabled"; another text field for "Animation" set to "Stand/Gestures/Explain_2"; a checked checkbox for "Auto-update parameters on robot"; and a "Reset to default" button. At the bottom of the dialog are "OK" and "Cancel" buttons. Below the dialog is a scene view window showing a character in a scene. Two objects, "Animated Say" and "Show Image", are connected to the character by curved lines, indicating they are part of the character's current state or animation.

LAUSE 7



LAUSE 8

The image displays a software interface for a robot's behavior programming. At the top, a timeline labeled 'Motion' shows a sequence of events from 1 to 105 seconds. Below it, a 'Behavior layers' section shows a layer named 'behavior_layer1'. A navigation bar at the bottom of the timeline shows a sequence of nodes: 'root', 'root', 'Oma vuoro', and 'Oma vuoro'. A dialog box titled 'Set parameters of Say' is open, showing the following parameters:

- Voice shaping (%): 100
- Speed (%): 100
- Text: \pau=300\Kun on si-nun vuo-ro-si\pau=500\n-nosta tuot-tteet kassalle.\pau=1500\
- Auto-update parameters on robot
- Buttons: OK, Cancel, Reset to default

Below the dialog box, a detailed view of the 'Oma vuoro' node is shown. It contains three sub-nodes: 'Oma vuoro', 'Say', and 'Show Image', each with a play button icon. The 'Say' node is highlighted with a blue selection box.

LAUSE 9

C Set parameters of Maksa ostokset ? X

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

Speaking movement mode

Animation

Auto-update parameters on robot

LAUSE 10

C Set parameters of Pakkaa ostokset kaupakassiin. ? X

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

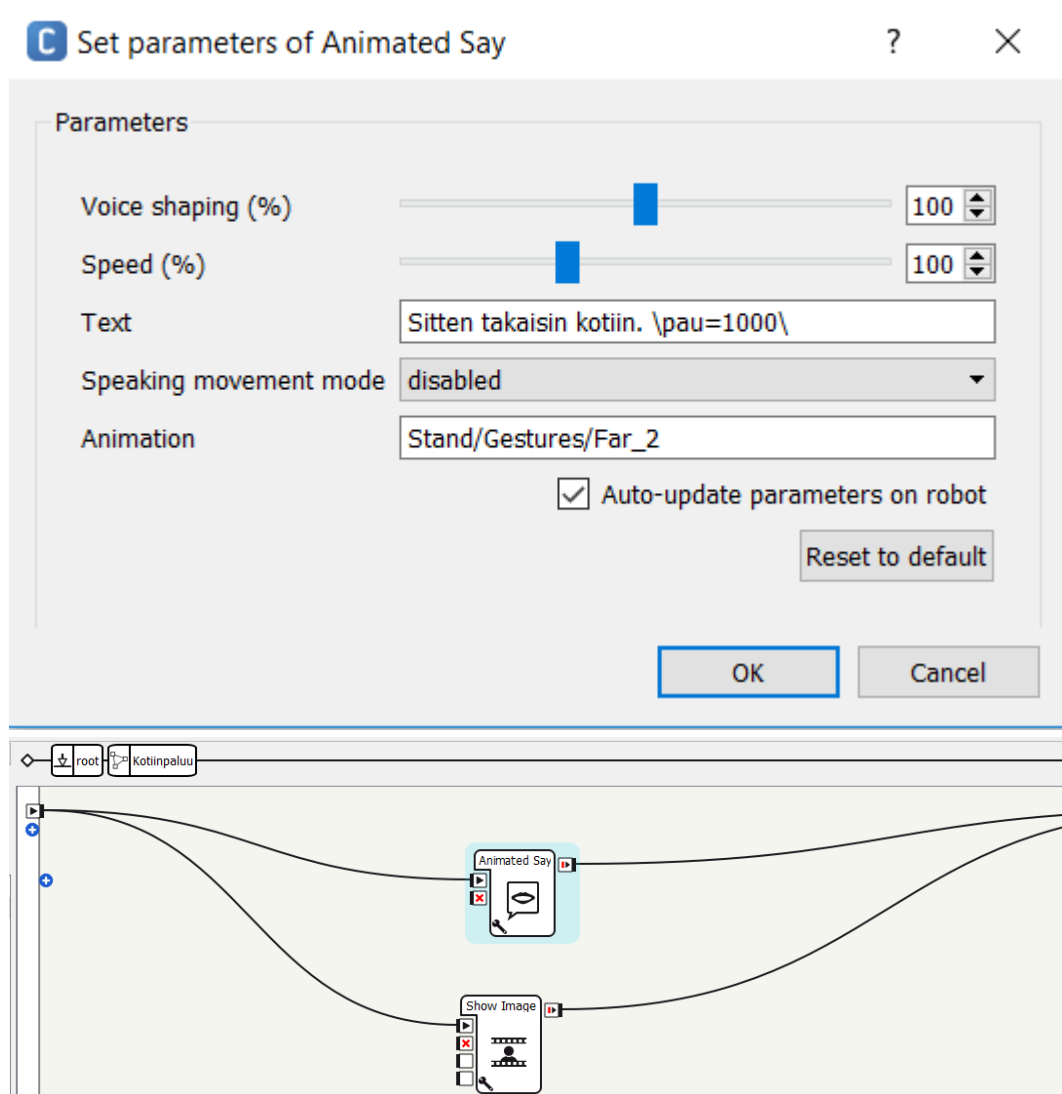
Speaking movement mode

Animation

Auto-update parameters on robot

The scene view shows a robot character in a scene. The scene is titled 'root' and contains two nodes: 'Pakkaa ostokset' and 'Show Image'. The 'Pakkaa ostokset' node is highlighted with a blue selection box. The 'Show Image' node is also highlighted with a blue selection box. The scene view shows a robot character in a scene with two nodes: 'Pakkaa ostokset' and 'Show Image'.

SPIIKKI 4



LAUSE 11

The image shows a software interface with a timeline at the top and a dialog box titled "Set parameters of Say" in the foreground. The timeline is labeled "Not running" and has a scale from 1 to 105. Below the timeline, there are "Motion" and "Behavior layers" sections. The "Behavior layers" section shows a layer named "behavior_layer1". The "Project files" section shows a hierarchy: root -> Mene kotiin liikennessä -> Mene kotiin.

The "Set parameters of Say" dialog box has the following parameters:

- Voice shaping (%): A slider set to 100.
- Speed (%): A slider set to 100.
- Text: A text field containing the string: `\pau=700\Mene\pau=300\ kotiin\pau=800\ liikenne sääntöjä noudat-ttaen\pau=1500\`
- Auto-update parameters on robot: A checked checkbox.
- Reset to default: A button.
- OK and Cancel: Buttons at the bottom.

Below the dialog box, the "Project files" section shows a diagram with three nodes: "Mene kotiin", "Sav", and "Show Image". The "Sav" node is highlighted with a blue border. Lines connect the "Mene kotiin" node to the "Mene kotiin" node in the "Project files" section, and the "Sav" and "Show Image" nodes to the "Mene kotiin" node in the "Project files" section.

LAUSE 12

C Set parameters of Pura ostoskassit heti

Parameters

Voice shaping (%) 100

Speed (%) 100

Text

Speaking movement mode

Animation

Auto-update parameters on robot

root Pura ostoskassit

Pura ostoskas

Show Image

LAUSE 12 JATKUU

The image displays a software interface with two main components. The top component is a dialog box titled "Set parameters of Laita etenkin jääkaappiin". It contains the following settings:

- Parameters
- Voice shaping (%): Slider set to 100
- Speed (%): Slider set to 100
- Text: "Laita etenkin jääkaappiin ja pakastimeen menevät ruoat heti paikoilleen. \pau=1500\"
- Speaking movement mode: "disabled"
- Animation: "Stand/Gestures/Explain_11"
- Auto-update parameters on robot
- Reset to default button
- OK and Cancel buttons

The bottom component is a scene diagram showing a path with two nodes: "Laita etenkin" and "Show Image". The path starts at a left node, goes down to "Laita etenkin", then up to "Show Image", and finally back to the right node. The "Laita etenkin" node is highlighted with a blue selection box.

SPIIKKI 5

