

Opinnäytetyö (AMK)  
TIETOJENKÄSITTELY  
2020

Tiina Virtanen

# SULAUTETUN JÄRJESTELMÄN KÄYTTÖ LASTEN OHJELMOINNIN OPETUKSESSA

– Case: Suomen Tiedekoulu Oy, Robbo Lab



Tiina Virtanen

# SULAUTETUN JÄRJESTELMÄN KÄYTTÖ LASTEN OHJELMOINNIN OPETUKSESSA

– Case: Suomen Tiedekoulu Oy, Robbo Lab

Tässä opinnäytetyössä selvitetään Robbo Lab- laitteen soveltuvuus ja sen toiminnallisuuksien mahdollinen hyödyntäminen ohjelmoinnin opettamisessa toimeksiantajan tarpeisiin. Toimeksiantajana toimii harrastuspainotteista lasten ohjelmoinnin opetusta tarjoava yritys. Kohderyhmänä lopputyössä on koodikoulun jatkoryhmän lapset iältään 9–15 vuotta.

Opinnäytetyö on toiminnallinen opinnäytetyö, jonka teoriaosuutta käsitellään olemassa olevan kirjallisuuden sekä verkkoaineiston pohjalta. Koska opinnäytetyö on tietojenkäsittelyn opinnäytetyö, pedagogisissa osissa hyödynnetään laitteen valmistajan omaa pedagogista tutkimusta sekä toimeksiantajan pedagogiikkaa.

Toiminnallinen osuus käsittää laitteeseen sekä sen ominaisuuksiin perehtymisen, laitteen visuaaliseen ohjelmointiympäristöön tutustumisen sekä sen toiminnan ja eroavaisuuksien vertailemiseen verkkopohjaiseen ohjelmointiympäristöön. Tämän jälkeen valmistettiin pedagogisten suositusten mukaan ennalta valituille teemoille ohjelmointitehtäviä, jotka testattiin kohderyhmästä valitulla pilottiryhmällä kevätlukukaudella 2018. Pilottiryhmiltä sekä ryhmien opettajilta kerättiin kirjallisen kyselyn muodossa arvioita tehtävistä sekä laitteen käytöstä. Kyselyyn vastattiin anonyymisti ja sitä analysoitiin kvalitatiivisesti. Ryhmien opettajilta saatiin myös suullista palautetta kevään aikana.

Kyselystä saatujen vastausten, suullisten palautteiden sekä opinnäytetyöntekijän selvitystyön ja omien johtopäätösten perusteella selvisi, että laite soveltuu ohjelmoinnin opetukseen, mutta ei nykytilassaan toimeksiantajan toiminta-ajatukseen. Laitteen testivaiheessa havaittiin pieniä ongelmia sekä ohjelmiston että laitteen toiminnassa. Laite ei myöskään tuo huomattavaa lisäarvoa kohderyhmän opetukseen. Laitteen käyttö on tämän opinnäytetyön valmistuttua lopetettu toimeksiantajayrityksessä. Opinnäytetyö toimii kuitenkin hyvänä selvitysprosessin dokumenttina ja voi auttaa toimeksiantajaa jatkossa nopeuttamaan sekä jäsentämään vastaavanlaisen laitteen selvitystyön prosessia sekä oppimateriaalin valmistamista sulautetun järjestelmän laitteen käyttöön.

## ASIASANAT:

ohjelmointi, ohjelmoinnin opetus, sulautettu järjestelmä

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

BUSINESS INFORMATION TECHNOLOGY

2020 | 44 pages, 25 appendices

Tiina Virtanen

# USE OF EMBEDDED SYSTEM IN TEACHING PROGRAMMING TO CHILDREN

– Case: Suomen Tiedekoulu Oy, Robbo Lab

This thesis investigates the suitability of the Robbo Lab device and the possible utilization of its functionalities in teaching programming for the needs of the commissioning company. The commissioning company provides programming education services as a free-time hobby to children. The target group for the thesis consisted of the children in the second-year group of the code school, aged 9-15 years.

The thesis is a functional thesis. The theoretical part is discussed based on the relevant literature and online material. As the thesis is in the field of data processing, the pedagogical parts utilize the device manufacturer's own pedagogical research, as well as the pedagogy of the commissioning company.

The functional part included getting acquainted with the device and its features, getting acquainted with the visual programming environment of the device, and comparing its operation and differences with the web-based programming environment. After this, programming tasks were prepared for pre-selected themes according to pedagogical recommendations. The tasks were tested in the spring semester of 2018 with a pilot group selected from the target group.

Assessments of the tasks and the usability of the device were collected from the pilot groups and their teachers in the form of a written questionnaire. The questionnaire was answered anonymously and analysed qualitatively. Oral feedback was also received from the groups' teachers during the spring.

Based on the answers to the questionnaire, the oral feedback, research, and conclusions by the author of this thesis, it became clear that the device is suitable for teaching programming, but not in its current state for the commissioning company's business idea. During the test phase of the device, minor problems were found in the operation of both the software and the device. The device also does not bring significant added value to the teaching of the target group.

The use of the device was stopped before the completion of this thesis in the commissioning company. However, the thesis serves as a good document of the investigation process and can help the client to speed up and structure the investigation process of a similar device in the future, as be useful in the preparation of study material for use in the embedded system device.

KEYWORDS:

programming, teaching programming, embedded systems

# SISÄLTÖ

<b>1 JOHDANTO</b>	<b>6</b>
<b>2 LASTEN OHJELMOINNIN OPETUS</b>	<b>8</b>
2.1 Ohjelmointi peruskouluissa	8
2.2 Ohjelmoinnin työvälineet	9
2.3 Pedagogiikka ja oppimateriaalit	11
<b>3 SULAUTETUT JÄRJESTELMÄT OPETUKSESSA</b>	<b>14</b>
3.1 Arduino Uno	14
3.2 Lego Mindstorms EV3	15
<b>4 CASE: SUOMEN TIEDEKOULU OY, ROBBO LAB</b>	<b>16</b>
4.1 Sulautetun järjestelmän laite – Robbo Lab	17
4.1.1 Robbo Lab – rakenne	17
4.1.2 Robbo Lab käyttöönotto	19
4.2 Ohjelmointityökalu Robbo Scratch	21
4.2.1 Scratch-sovellus Robbo Scratchin taustalla	22
4.2.2 Robbo Scratch johdanto	23
4.2.3 Robbo Scratch – käyttöönotto	23
4.2.4 Scratch ja Robbo Scratch – vertailu ja havainnot	24
4.3 Ohjelmointitehtävien suunnittelu ja valmistusprosessi	26
4.3.1 Tuntikortti 1 kuvaus	28
4.3.2 Tuntikortti 2 kuvaus	29
4.3.3 Tuntikortti 3 kuvaus	31
4.4 Robbo Lab – kevään 2018 testauksen tulokset	32
4.5 Robbo Lab soveltuvuuden johtopäätökset	36
<b>5 POHDINTA</b>	<b>40</b>
<b>LÄHTEET</b>	<b>43</b>

## LIITTEET

Liite 1. Tuntikortti 1

- Liite 2. Tuntikortti 2
- Liite 3. Tuntikortti 3
- Liite 4. ROBBO LAB – kysely 2018, kysymykset

## KUVAT

Kuva 1. Robbo Lab osat	18
Kuva 2. Robbo Lab pakkaus	20
Kuva 3. Robbo Lab laiteohjauspaneeli	21
Kuva 4. Scratch -online-käyttöliittymä	25
Kuva 5. Robbo Scratch -sovelluksen käyttöliittymä	25
Kuva 6. Kysymys ohjeista	33
Kuva 7. Kysymys hyödyllisyydestä	34
Kuva 8. Kysymys kohderyhmästä	35

## TAULUKOT

Taulukko 1. Robbo Lab osat. (Robbo 2016)	19
Taulukko 2. Sovellusten osat	24

# 1 JOHDANTO

Opinnäytetyöni aiheeksi valikoitui ”SULAUTETUN JÄRJESTELMÄN KÄYTTÖ LASTEN OHJELMOINNIN OPETUKSESSA - Case: Suomen Tiedekoulu Oy, Robbo Lab” silloisen työnantajani (myöhemmin tässä opinnäytetyössä toimeksiantaja tai Tiedekoulu) tarpeesta selvittää, soveltuuko lasten ohjelmoinnin opetukseen suunniteltu Robbo Lab -laite Tiedekoulun toiminta-ajatukseen. Olin työskennellyt lasten ohjelmoinnin opetuksen parissa kevätkaudesta 2017 asti, joten kiinnostus aiheeseen oli henkilökohtainen ja ajan-kohtainen.

Toimeksiantona ja samalla tutkimuskysymyksenä oli selvittää Robbon kehittämän Robbo Lab -laitteen soveltuvuus toimeksiantajan toiminta-ajatukseen ja teemapohjaiseen harrastuspainotteiseen lasten ohjelmoinnin opettamiseen, jossa oppimisen tavoite on kehittää lapsen omaa oivallusta ja ongelmanratkaisukykyä. Opinnäytetyön tavoite oli myös suunnitella ja mallintaa Robbo Lab -laitteelle tarinallisia tunnissa suoritettavia ohjelmointitehtäviä Robbo Scratch 2.1 -ohjelmointialustalle (myöhemmin tässä opinnäytetyössä Robbo Scratch). Kohderyhmänä lopputyössä on koodikoulun jatkoryhmän lapset iältään 9–15 vuotta, joilla on jo perustason taidot visuaalisesta Scratch 2 (myöhemmin tässä opinnäytetyössä Scratch) -ohjelmoinnista koodikoulun alkeisryhmästä.

Tutkimuskysymystä ei voida lähestyä täysin objektiivisesti, vaan toimeksiantajan toiminnassa mukana olevana, aktiivisena ja osallistuvana toimijana, yhtenä opettajana.

Opinnäytetyön teoriaosuutta käsitellään olemassa olevan kirjallisuuden sekä verkkoaineiston pohjalta. Opinnäytetyöni aihe oli tietotekninen, mutta aiheen toinen osio, oppimateriaalin tuottaminen, menee oman opintojeni pääaiheen ohi. Tästä syystä teoriaosuus sivuaa pedagogiikkaa, mutta pääpaino on tietoteknisissä osissa. Oppimateriaalin valmistelemissa osissa hyödynnetään laitteen valmistajan omaa pedagogista tutkimusta sekä toimeksiantajan pedagogiikkaa.

Toiminnallisessa osuudessa aihetta lähestytään empiirisesti ja subjektiivisesti aiemmilla opetuskokemuksilla, käyttäjäkokemuksilla sekä ennakko-odotuksilla, joita on jo kertynyt toimeksiantajalla työskentelyn aikana. Toiminnallinen osuus tehtiin neljässä vaiheessa, jotka olivat 1) Robbo Lab -laitteen sekä sen ominaisuuksien ja käytettävyyden havainnointi, 2) Robbo Scratch ohjelmointiympäristöön tutustuminen sekä sen toiminnan ja eroavaisuuksien havainnointi verrattuna verkko-pohjaiseen Scratch-ympäristöön, 3)

Robbon laatiman pedagogisen tutkimuksen sekä toimeksiantajan toiminta-ajatuksen pohjalta luotujen tarinallisten ohjelmointitehtävien valmistus kohderyhmälle sekä 4) ohjelmointitehtävien ja laitteen käytön testaus pilottiryhmässä, testauksesta kerättyjen palautteiden ja testauksen jälkeen tehdyn kyselyn aineiston analysointi.

Ohjelmointitehtäviä ja Robbo Lab -laitetta testattiin kevätlukukaudella 2018 kohderyhmästä valituilla pilottiryhmillä. Lukukauden loppuun pilottiryhmiltä sekä opettajilta kerättiin anonymisti Google Forms -kyselylomakkeen muodossa arvioita ohjelmointitehtävistä sekä Robbo Lab -laitteen käytöstä. Opettajilta saatiin myös kevään aikana suullista palautetta. Testauksen tulokset kuvataan yhdessä kyselyn aineiston kvalitatiivisen analysoinnin yhteydessä.

Laitteen ja ohjelmistoon perehtymisen, kyselylomakkeen sekä henkilökohtaisen opetus- ja lasten oppimiskokemusten perusteella tehdään lopuksi johtopäätöksiä, suosituksia ja kehitysehdotuksia sekä arvioidaan työn tuloksia. Lopuksi pohditaan Robbo Lab -laitteen soveltuvuutta toimeksiantajan toiminta-ajatukseseen.

## 2 LASTEN OHJELMOINNIN OPETUS

”Ohjelmointia voidaan ajatella nykyajan käsityöläistaitona.”  
(Helsingin yliopisto 2020)

Helsingin yliopiston järjestämällä Ohjelmoinnin MOOC 2020 -verkkokurssilla ohjelmoinnin merkitys 2020-luvulla, on kiteytetty hyvin tuohon yhteen lauseeseen. Ohjelmoinnin opetusta voidaan nykyään pitää lähes yhtä tärkeänä kuin käsityötaitoja. Tässä luvussa käsitellään ohjelmoinnin opetuksen lähestymistapaa peruskouluissa sekä taustaa sulautettujen järjestelmien mukaanottoon ohjelmoinnin opetukseen.

### 2.1 Ohjelmointi peruskouluissa

Peruskoulujen opetusohjelmaan (myöhemmin tässä opinnäytetyössä OPS) on kirjattu vuodesta 2016 alkaen osaamisen tavoitteisiin tieto- ja viestintäteknologinen osaaminen. Ohjelmoinnista ei siis ole tehty varsinaisesti omaa oppiainetta. Sen lisäksi, että kaikkiin oppiaineisiin kehoitetaan ottamaan mukaan tietotekniikkaa, on osana matematiikan opintosuunnitelmaa kirjattu ohjelmoinnin opetus. (Opetushallitus 2017)

OPSin mukaan matematiikan 6. vuosiluokan päätteeksi on arvioinnin kriteereihin määritetty, että oppilaan pitäisi osata ohjelmoida graafisessa ohjelmointiympäristössä toimiva ohjelma. (Opetushallitus 2014)

Ohjelmointi on kuitenkin paljon muutakin kuin matematiikkaa. Se on suureksi osaksi myös eräänlaista taidetta, loogista päättelykykyä ja vaatii rutkasti kärsivällisyyttä, kuten Tiina Partanen kertoo koodauksen abc -blogissaan tekemässään videossa. Samassa videossa esitellään hyvällä esimerkillä myös, miten pelien ohjelmoinnissa tarvitaan matemaattisia laskutoimituksia. Matematiikassa tietyt asiat ja termit, joita tarvitaan myös ohjelmoinnissa, kuten laskutoimitukset, muuttujat sekä vakiot, funktiot tai esimerkiksi koordinaatisto, tulevat koulujen opetussuunnitelmassa tietyssä järjestyksessä eri luokka-asteille. Lapsen onkin helpompi oppia ohjelmointia, kun termit ja käsitteet ovat jo ennestään matematiikasta tuttuja. (Partanen 2014)

Ohjelmoinnissa on myös hyötyä kielitaidosta. Ohjeiden ja oppaiden saatavuus on runsasta, mutta yleensä materiaali on tarjolla englannin kielellä. Viimeistään siinä vaiheessa, kun halutaan opiskella jotakin tekstuaalista ohjelmointikieltä, on ohjelmointikieli



ja ohjeet englanninkielisiä. Tämän vuoksi suositellaankin, että tekstuaalinen ohjelmointikieli aloitetaan vasta yläasteella luokilla 7–9. (Liukas & Mykkänen 2014)

Liukkaan ja Mykkäsen kirjoittaman Koodi2016 -hankkeen saman nimisessä oppaassa, kehoitetaan lähestymään ohjelmointia leikkien ja verkkoharjoitusten kautta. Samassa oppaassa opastetaan, että ohjelmoinnin oppimisen sijaan painotettaisiin ohjelmoinnillisen ajattelun perusteiden oppimista ("computational thinking"), itse tekemistä ja harjoittelua. (Liukas & Mykkänen 2014)

Koodiaapisen mukaan myöskään OPSissa ohjelmointi ei ole päätavoite oppimisessa, vaan algoritmisen ajattelun oppiminen. Ohjelmointia käytetään vain välineenä uusien ajattelutapojen saavuttamiseksi. Kun ohjelmointi sisällytetään oppiaineisiin, sen opetuksen päävastuu on aluksi luokanopettajalla, myöhemmin yläasteella myös matematiikan opettajalla, ei siis pelkästään ATK-opettajalla. Koodiaapisen toukokuussa 2017 opettajille tekemän tutkimuksen mukaan ohjelmoinnin opetuksessa ongelmia on tuonut muun muassa opettajan osaamisen riittämättömyys. (IT- kouluttajat ry 2014–2020)

On täysin koulukohtaista, miten ohjelmoinnin opetusta toteutetaan, tai miten tieto- ja viestintäteknikkaa otetaan osaksi muuta opetusta. Tietoteknisen opetuksen on tarkoitus kulkea eri oppiaineissa ikätason mukaisesti. (Liukas & Mykkänen 2014)

Ylen uutisessa vuodelta 2016 kerrotaan, miten opetusministeriö on joutunut turvautumaan it-alan yritysten apuun ja hyödyntänyt yritysten apua valmistautuessaan ohjelmoinnin opetuksen tulon peruskouluihin. Uutisessa kerrotaan syyksi opettajien ohjelmointitaitojen vaihtelevat valmiudet. Avunpyynnöstä seurasi, että useat it-alan yritykset ja yhdistykset ovat järjestäneet ns. digikouluja ympäri Suomen. (Konttinen 2016)

Tämä muutos koulumaailmassa on sysännyt alkuun harrastuspainotteisen ohjelmoinnin tarpeen koulumaailman tueksi niin oppilaille kuin opettajillekin, ohjelmointitaitoisten it-alan ammattilaisten johdolla. Opettajille on tarjolla monissa yliopistoissa ja ammattikorkeakouluissa maksutonta Opetushallituksen rahoittamaa opetustoimen henkilöstökoulutusta tieto- ja viestintäteknologian taitojen hallintaan sekä ohjelmoinnin opetukseen.

## 2.2 Ohjelmoinnin työvälineet

Opetussuunnitelmaan on kirjattuna peruskoulun 3.–6. -luokkalaisille käsityön tavoitteisiin harjoitella ohjelmointia robotiikkaan ja automaatioon liittyen. Yläasteen vuosiluokille 7–9

käsityön tavoitteisiin on kirjattuna sulautettujen järjestelmien käyttö soveltaen ohjelmointia suunnittelemiseen ja tuotteiden valmistamiseen. (Opetushallitus 2014) Tämä voi tarkoittaa esimerkiksi robotin rakentamista.

Konstruktivistisen oppimiskäsityksen edelläkävijä, emeritusprofessori Seymour Papert, oli maailmankuulu matemaatikko, oppimisteoreetikko ja kasvatusteknologian visionääri. Hän oli myös yksi Massachusetts Institute of Technology (myöhemmin tässä opinnäytetyössä MIT) Media Labin perustajajäsenistä. Papert ensimmäisten joukossa, tunnusti tietokoneiden mahdollisuudet koulutuksessa ja oppimisessa. (MIT Media Lab 2016).

Emilia Hjelm esittää pro gradu -tutkielmassaan Jean Piagetin (1896–1980) teoriaa oppimisen kehittymisestä kehitysvaiheittain lapsen ikäkauteen perustuen ja miten ympäristö ja sen ärsykkeet ohjaavat ajattelua. Hjelm esittelee myös, miten Seymour Papert (1928–2016) tutki Piagetin teorioita ja yhdisti oppimisteorian tietokoneisiin 80-luvulla, kirjoittamassaan Mindstorms-kirjassaan. Pro gradu -tutkielmassa kerrotaan myös, miten Papert piti oppimattomuuden syynä rakennusmateriaalin eli virikkeiden puutetta ja miten tietokonetta käyttämällä voidaan oppia myös muuta. Hjelm esittää Papertin näkemystä ohjelmoinnin opettelemista, joka tuo oppimiseen henkilökohtaista samaistumista, joka motivoi lasta oppimaan. Ohjelmoidessa voidaan samaistua esimerkiksi ohjelmassa olevaan esineeseen ("object-to-think-with"). (Hjelm 2016)

Työskennellessään MIT:ssä Papert kehitti lapsille suunnatun LOGO-ohjelmointikielen vuonna 1968. Papertin ajatukset toimivat myös inspiraationa MIT:n yrityssponsori Legon LEGO Mindstorms -robotiikkasarjalle. Sarja myös nimettiin hänen Mindstorms-kirjansa mukaan. (MIT Media Lab 2016)

Sulautettujen järjestelmien käyttöönottoa lapsille ohjelmoinnin työvälineenä puolustaa yksi Käsityökoulu Robotin perustajajäsenistä, Tomi Dufva, artikkelissaan Digitaalisuuden ykköset ja nollat. Dufvan mukaan, kun lisätään robotti ja elektroniikka fyysisellä tasolla mukaan ohjelmointiin, se tekee ohjelmoinnista konkreettista toimintaa ja se voi olla apuna ohjelmoinnin ymmärtämisessä. Robotin mukaan ottaminen ohjelmointiin tekee siitä myös interaktiivista toimintaa. (Dufva 2020)

Sulautettujen järjestelmien kanssa ohjelmoidessa, ruudulla tapahtuvan toiminnon sijaan käyttäjä pääsee konkreettisesti ja reaaliaikaisesti osallistumaan ohjelman suoritukseen käyttämällä laitetta käsillään. Tästä seuraa välitön ulkopuolinen palaute. Siinä missä tietokone oli uutta oppimisessa 80-luvulla, voisivat sulautettujen järjestelmien laitteet olla

nyt 2020-luvulla mukana tuomassa henkilökohtaista samaistumisen tunnetta ja lisäämässä oppimista ja oivaltamista.

### 2.3 Pedagogiikka ja oppimateriaalit

”Pedagogiikka tarkoittaa tapaa, jolla opetus järjestetään sekä sen näkemyksellisiä ja kasvatuksellisia periaatteita” (WIKIMEDIA FOUNDATION 2020c).

Opetushallitus nostaa esiin uudessa opetussuunnitelmassaan 2014 toiminnallista pedagogiikkaa. Oppilas nähdään vastuullisena ja aktiivisena toimijana, joka on myös vastuussa ja tietoinen omasta oppimisestaan ja toimii itseohjautuvasti. Opetussuunnitelmassa tuodaan esille myös kielen, kehollisuuden ja aistien käytön tärkeys ajattelun ja oppimisen kannalta. Oppimisen edistämiseksi nostetaan esille myös kokemukselliset ja toiminnalliset työtavat. Opetussuunnitelman mukaan nämä lisäävät ja vahvistavat oppimisen motivaatiota. (Opetushallitus 2014)

Toiminnallisia työtapoja tukee oppimateriaali, joka on suunniteltu oppilasta ajatellen. Opettajalehden artikkelissa Hyvä oppimateriaali syntyy oppijan saappaisiin astumalla ja opettajilta se luonnistuu – kerrotaan kolmen opettajaporukan oppikirjahankkeista. Artikkelisiin on nostettu esimerkkejä siitä, minkälainen oppimateriaali on hyvää. Artikkelissa kuvataan esimerkkejä pelillisestä lähestymistavasta, jossa oppilas saa olla aktiivinen toimija, oppimateriaalin selkokielisyydestä ja visuaalisista elementeistä, jotka voivat auttaa muistamisessa ja ymmärtämisessä. Artikkelissa on myös mainittu, että kohderyhmä tulisi huomioida tuomalla oppimateriaaliin hauskaa ja värikästä sisältöä ja käsiteltävien aiheiden tulisi sisältää kosketuspintaa kohderyhmän omaan elämään. (Ängeslevä 2019)

Development of the ROBBO pedagogical material – design Based Research Approach -raportissa kerrotaan Robbon sekä Innokas-verkoston (Opettajankoulutuslaitos, Helsingin yliopisto) yhteistyössä toteuttamasta tutkimuksesta, jonka pohjalta myöhemmin rakentui Innokas-verkoston valmistama pedagoginen opas Robbo-robotialustalle. Tutkimukseen osallistui 74 oppilasta iältään 9–16-vuotiaita sekä yhdeksän opettajaa Innokas-verkostosta. Tutkimusta tehtiin kahdessa osassa syksyllä 2015 sekä keväällä 2016. Raportissa todetaan, että ensimmäisen testausvaiheen tuloksissa kävi ilmi, miten Robbolle tehtävä pedagoginen opas tulisi suunnata vastaamaan eri tasoisten käyttäjien tarpeita. Tutkimuksen perusteella tehtiin myös huomioita, että selkeä jäsennelty materiaali hyödyttää aloittelijoita, ohjaa oppilaita toimimaan itsenäisesti. Edistyneimmille käyttäjille sen sijaan pitäisi tarjota uusia näkökulmia. Harjoitusten tulisi olla herätteleviä ja motivoivia.

Raportissa mainitaan, että tutkimusvaiheessa Lab -laitteen laitteiston sekä harjoituskorttien testaus jäi vaillinaiseksi ajan puutteen vuoksi. Tutkimuksen tuloksena syntynyt pedagoginen opas sisältää harjoituskortteja, jotka ovat valmistettu niin, että niiden pedagogiset käyttömahdollisuudet olisivat mahdollisimman monipuoliset. Opettaja pystyy käyttämään niitä apuna opetuksessa tai oppilas voi harjoitella niiden avulla itsenäisesti. (Sormunen, Korhonen ym. 2016)

Innokas-verkoston valmistamassa Robbon pedagogisessa oppaassa on ensin kerrottu teoriaa ja opetukselliset asiat opettajalle, jonka jälkeen on kuvattu harjoitustehtävässä käytettävät lohkot. Opettajan tarkoitus on olla työskentelyä eteenpäin vievä, ohjaavilla kysymyksillä oppilaiden oppimista tukeva. Harjoituskortissa on esitetty varsinainen ongelma sekä lopuksi ratkaisumalli. Harjoitukset on ryhmitelty aiheittain ja ne etenevät tehtävästä toiseen vaikeutuen kohti pelattavaa yksinkertaista peliä. Harjoitukset on suunniteltu vuorovaikutteisiksi pareittain tehtäviksi, mutta niitä voi tehdä yksin omaan tahtiin tai ryhmänä yhdessä. Aluksi harjoituksissa ohjelmointi tehdään mallin mukaan, myöhemmin jätetään tilaa oppilaan oivalluksille. (Innokas-verkosto, ROBBO 2016)

Omien havaintojeni mukaan, Innokas-verkoston valmistamaan Robbon pedagogiseen oppaaseen tehtyjen harjoituskorttien rakenne on selkeä ja visuaalinen. Harjoitukset ovat yksityiskohtaisia, teemapohjaisia ja toiminnallisia.

Tiedekoulun pedagogiikka perustuu toiminnallisuuteen, itse tekemiseen ja oivaltamiseen sekä teemaoppimiseen. Jokaisella oppitunnilla lähestytään käsiteltävää teemaa ensin teorian avulla ja sen jälkeen oppilaat saavat tutkia ja konkreettisesti tehdä harjoituksia itsenäisesti, opettajan ohjatessa ja auttaessa tarpeen mukaan. Oppilasta kannustetaan luovuuteen ja keksimään itse ratkaisuja ongelmiin. (Suomen Tiedekoulu Oy 2019)

”Tutkimalla, yrittämällä, erehtymällä ja ratkomalla ongelmia lapsi oppii oikeasti soveltamaan käytännössä sitä mitä on oppinut, kehittämään ja keksimään uutta.”  
(Suomen Tiedekoulu Oy 2019)

Tiedekoulun oppimispolkujen käytössä oleva oppimateriaali sisältää yhdenmukaisia kohtia Innokas-verkoston luoman materiaalin kanssa. Oppimateriaalit on tehty yksityiskohtaisiksi ja oppilaan ikätasolle sopiviksi. Nuoremmille lapsille visuaaliset elementit ovat pääasiassa ja edistyneemmille oppilaille harjoitustehtävät soveltuvat myös itsenäiseen tekemiseen. Oppimateriaali on teemoitettu, jossa perusteorian läpikäynnin jälkeen oppilaat soveltavat teoriaa pääasiassa itsenäisesti suorittamalla harjoituksia ja tehtäviä.

Myös opettajan rooli nähdään samanlaisena, tukevana, ohjaavana ja tarpeen mukaan auttavassa roolissa toimivana. Teemojen vaativuus kasvaa aina vuosittain.

## 3 SULAUTETUT JÄRJESTELMÄT OPETUKSESSA

Sulautetulla järjestelmällä tarkoitetaan tietokoneella ohjattua laitetta tai alustaa, jolle on tyypillistä, että käyttäjä ei ole täysin tietoinen sen sisältämästä tietokoneesta. Sulautettu järjestelmä toteutetaan käyttämällä mikrokontrolleria. Sulautetun järjestelmän laitteisto sekä ohjelmisto ovat yleensä toteutettu jotain tiettyä tarkoitusta varten. Sulautettujen järjestelmien lähdekoodi voi olla avointa tai suljettua. (Wikimedia Foundation 2020b)

Vaikka tässä opinnäytetyössä ei varsinaisesti ole tarkoitus verrata tutkimuskysymyksen kohteena olevaa Robbo Lab -laitetta toisiin sulautettujen järjestelmien laitteisiin, tähän lukuun on otettu kaksi toimeksiantajalla jo käytössä olevaa laitetta, joita on käytetty Robbo Lab -laitteen käytettävyyden havainnoimisen pohjatietoina. Arduino on mukana, koska Robbo Lab on rakennettu sen perustalle. Lego Mindstorms- robotti taas on tällä hetkellä toimeksiantajalla opetuksessa eniten käytetty sulautetun järjestelmän laite.

### 3.1 Arduino Uno

Yksi tunnetuimmista yleisesti opetuksessa käytetyistä sulautetun järjestelmän alustoista on Arduino, joka perustuu 8-bittiseen Atmel -mikrokontrolleriin. Sitä voidaan ohjelmoida USB- sarjaportin kautta Arduino IDE -ohjelmointisovelluksella C/C++ -kielellä. Arduinoon voidaan liittää elektronisia ja mekaanisia komponentteja ja näin rakentaa erilaisia laitteita tai robotteja. Arduino saa virtansa joko paristolla, tai USB-liittimen kautta. Arduinoja on saatavilla useilla erilaisilla kokoonpanoilla, tunnetuin näistä on Arduino Uno. (Blum 2013)

Arduinoon on saatavilla runsaasti edullisia lisäkomponentteja. Opetusmateriaalia löytyy runsaasti eri sosiaalisten yhteisöjen kautta. Yhteisöistä on saatavilla myös apua ja tukea. Ominaisuuksiensa vuoksi laitteen käyttö edellyttää elektroniikan ja mekaniikan peruskomponenttien tuntemista, rakentelua sekä englannin kielen taitoa. Toimeksiantaja on laitteen käytön yhteydessä havainnut ongelmia mekaanisten ja elektronisten osien irtoamisen sekä ohjelmointikielen vaikeuden vuoksi. Nämä ongelmat saattavat aiheuttaa tunneilla turhautumista, joka taas voi johtaa motivaation loppumiseen. Laitteen rakentaminen edellyttää useamman oppitunnin mittaista rakentelua, mikä edellyttää keskeneräisten rakennelmien säilyttämistä, laitteen varausta yhdelle oppilaalle kerrallaan sekä säilytystilaa. Rakentelun vuoksi tunneilla ei voida suoraan ryhtyä ohjelmointiin.

### 3.2 Lego Mindstorms EV3

Lego Mindstorms EV3 on Lego Groupin kehittämä ja valmistama robottikonaisuus. Se koostuu nimensä mukaisesti Lego palikoista sekä ohjelmoitavasta keskuspalikasta, johon voidaan liittää settiin kuuluvia antureita, sensoreita sekä moottoreita. Mikrokontrolleri sijaitsee keskuspalikassa. Virtalähteenä Lego Mindstormsissa toimii akku tai paristot. Lego Mindstormsia voidaan ohjelmoida käyttämällä USB-liitännää, tai Bluetooth- ja IR-yhteyksiä. Lego Mindstormsin ohjelmointisovelluksena toimii sen oma visuaalinen ohjelmointisovellus. Mikäli korvaa keskuspalikassa olevan alkuperäisen laiteohjelmiston toisen osapuolen valmistamalla korvaavalla laiteohjelmistolla, voidaan ohjelmoinnissa myös käyttää Javaa, C-, tai Forth-kieltä. (WIKIMEDIA FOUNDATION 2020a)

Toimeksiantajan aiempien opetustilanteiden aikana tehtyjen havainnointien perusteella Lego Mindstormsin etuna opetuksessa on sen leikillisuus sekä Lego palikoiden tuoma motivaatio. Sovelluksen logiikka on hieman vaativa ja sen käyttö vaatii myös jonkin verran englannin kielen taitoa. Laitteen rakentaminen edellyttää myös useamman oppitunnin mittaista rakentelua, mikä edellyttää keskeneräisten rakennelmien säilyttämistä ja säilytystilaa. Rakentelun vuoksi tunneilla ei voida suoraan ryhtyä ohjelmointiin.

## 4 CASE: SUOMEN TIEDEKOULU OY, ROBBO LAB

Opinnäytetyön toimeksiantajana toimiva Suomen Tiedekoulu Oy tarjoaa harrastuspainotteista tiede- ja teknologiaopetusta lapsille. Tiedekoulun toimintahistoria alkaa vuonna 2012 järjestetyistä tiedekerhoista. Vuoden 2014 loppupuolella toiminta on aloitettu virallisesti. (Suomen Tiedekoulu Oy 2019).

Tiedekoulussa järjestettävässä ohjelmoinnin opetuksessa on tarkoituksena toimia kouluissa tapahtuvan oppimisen täydentäjänä ja tukijana, mielekkäänä harrastuksena. Vuonna 2018 opinnäytetyön aloitushetkellä ryhmiä järjestettiin 30 toimipisteessä ympäri Suomen ja opetus on suunnattu pääasiassa 4–15-vuotiaille lapsille. Opetus seuraa yrityksen itse kehittämää pitkäjänteistä ja tavoitteellista opetussuunnitelmaa. Harrastuspainotteisissa ryhmissä opetussuunnitelma koostuu oppimispoluista kevät- ja syyslukukausilla. Lukukausi koostuu viidestätoista, kerran viikossa pidettävästä, tunnin mittaisesta opetustunnista. Ohjelmointi on yksi kolmesta Tiedekoulun oppimispolusta. Ohjelmoinnin opetuksen jokaisella oppitunnilla on teema, jossa perusteorian läpikäynnin jälkeen oppilaat soveltavat teoriaa pääasiassa itsenäisesti suorittamalla harjoituksia ja tehtäviä, opettajan ohjatessa ja auttaessa tarpeen mukaan. Teemojen vaativuus kasvaa aina vuosittain ja ohjelmointia käsitellään aina vähän eri näkökulmista tai ohjelmointikieltä vaihtaen. (Suomen Tiedekoulu Oy 2019).

Tunneilla oppilaalla on mukana oma tietokone. Lainakoneita on saatavilla, jos omaa tietokonetta ei ole. Opetuksessa käytetään pääsääntöisesti maksuttomia web -pohjaisia oppimisympäristöjä, jotka eivät vaadi asentamista omalle koneelle ja ovat oppilaiden käytettävissä myös harrastuksen ulkopuolella. Tunneilla käytetään myös useita erilaisia sulautettujen järjestelmien laitteita oppitunnin teemasta ja oppimispolusta riippuen.

Kohderyhmänä tässä opinnäytetyössä ovat koodikoulun jatkoryhmän oppilaat.

Ohjelmoinnin ensimmäisellä tasolla opetellaan ohjelmoinnin alkeita visuaalisella ohjelmointisovelluksella Scratchilla. Kahden lukukauden jälkeen oppilas voi siirtyä seuraavalle tasolle. Koodikoulun jatkoryhmissä syvennytään ohjelmointiin, tietotekniikkaan sekä pelien tekemiseen. Ryhmien opetuksessa käytetään muun muassa visuaalista ohjelmointisovellus Scratchia ja tekstuaalista ohjelmointikieli Pythonia. Tavoitteena on, että jatkoryhmän jälkeen oppilaalla olisi syventävät ohjelmointitaidot visuaalista ohjelmointisovellusta käyttäen sekä ainakin perustaidot tekstuaalisesta ohjelmoinnista. Oppilas



olisi myös laajentanut omaa ohjelmoinnin osaamistaan, loogista päättelykykyä sekä ongelmanratkaisutaitojaan. Jatkoryhmien oppilailla on jo perustaidot ohjelmoinnista visuaalisella ohjelmointikielellä Tiedekoulusta kahden lukukauden verran. (Suomen Tiedekoulu Oy 2019)

Tässä luvussa kuvataan opinnäytetyön toiminnallisen osuuden vaiheet, joita ovat 1) Robbo Lab -laitteen sekä sen ominaisuuksien ja käytettävyyden havainnointi, 2) Robbo Scratch ohjelmointiympäristöön tutustuminen sekä sen toiminnan ja eroavaisuuksien havainnointi verrattuna verkkopohjaiseen Scratch -ympäristöön, 3) Robbon laatiman pedagogisen tutkimuksen sekä toimeksiantajan toiminta-ajatuksen pohjalta tarinallisten ohjelmointitehtävien luominen, 4) ohjelmointitehtävien ja laitteen käytön testaus pilottiryhmässä, testauksesta kerättyjen palautteiden ja testauksen jälkeen tehdyn kyselyn aineiston analysointi. Lopuksi tehdään yhteenveto ja johtopäätökset.

#### 4.1 Sulautetun järjestelmän laite – Robbo Lab

Robbo Labin valmistaja Robbo koostuu kansainvälisestä tiimistä Suomesta ja Venäjältä. Alun perin Venäjältä lähtenyt Robbo aloitti Suomen valloituksensa 2014 osallistuttuaan Finlanding -kilpailuun, jonka tarkoituksena on auttaa venäläisiä hi-tech yrityksiä pääsemään Suomen ja Euroopan markkinoille. (Robbo 2016)

Robbo Lab kuuluu Robbo Robot -sarjaan, joka on valmistajansa mukaan erittäin muokattavissa oleva, yksinkertainen ja interaktiivinen avoimen lähdekoodin robottirakennesarja, johon saa myös magneettikiinnitteisiä antureita. Robbo Robot -sarjan robotteja voidaan käyttää ohjelmoinnin opetuksessa ja niitä voidaan ohjelmoida käyttämällä esimerkiksi visuaalista ohjelmointikieltä Scratch by MIT. (Robbo 2016)

Robbo Lab -laite, joka oli opinnäytetyössä mukana ja testikäytössä, oli Robbo Lab v3.0. Jokaisella laitteella on myös oma yksilöllinen nimensä, testilaitteen nimi oli ROB-L-3-3-119.

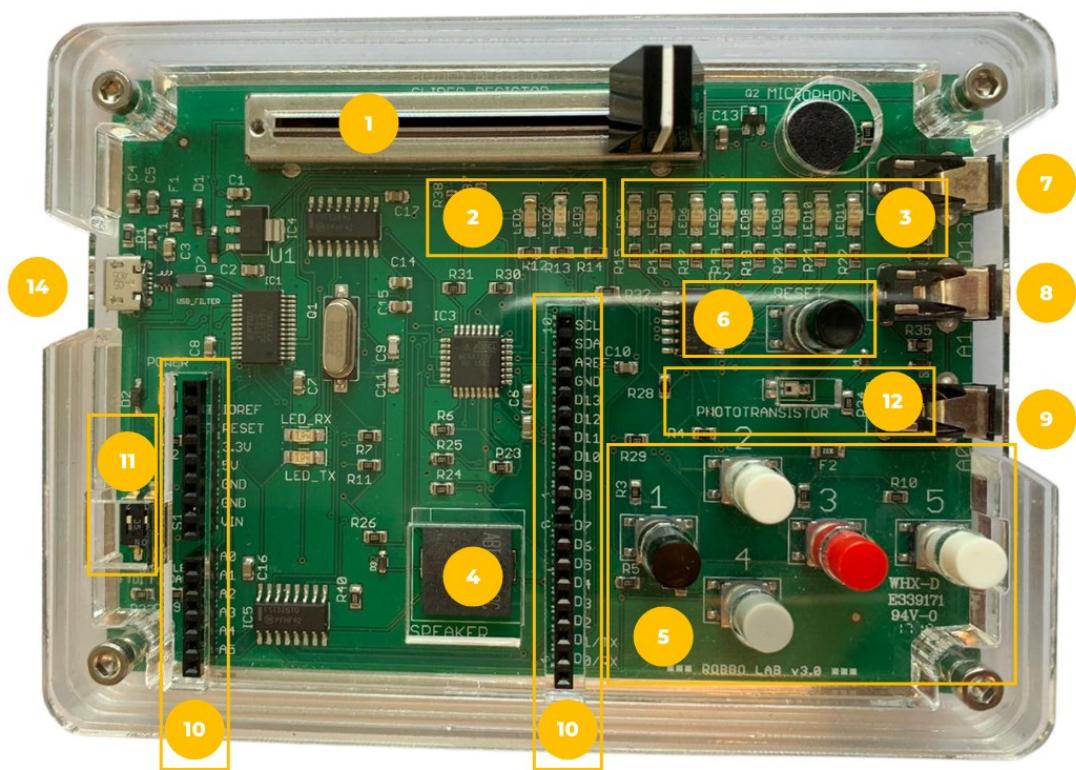
##### 4.1.1 Robbo Lab – rakenne

Robbo Lab on kauttaaltaan akryylilasiin koteloitu opetusrobotti, joka pohjautuu avoimen lähdekoodin Arduino-mikrokontrolleriin. Kotelointi suojaa laitetta ja kaikki komponentit noudattavat eurooppalaista CE-standardia. Läpinäkyvän koteloinnin ansiosta robotin

tekniikka on näkyvässä. Laite ei ole kuitenkaan vesitiivis kotelon reikien vuoksi, eikä se kestä pudottelemista. (Robbo 2016)

Koteloinnin ja integroitujen osiensa ansiosta ei erillistä sähkötekniikan tuntemusta tai osaamista vaadita käytön aloittamiseen. Laitteen käyttäjä voi keskittyä suoraan ohjelmointiin saaden kokemuksen sulautettujen järjestelmien ohjelmoinnista ja robotiikasta. Robbo Lab on kooltaan 13,6 cm\*9,1 cm\*1,8 cm.

Robbo Lab -laitteen aivoina toimii ATmega 328 MPU -mikroprosessoriyksikkö. (Robbo 2016)



Kuva 1. Robbo Lab osat

Ohjattavia osia Robbo Labissa on 13, jotka selitetty sivun 19 taulukossa 1. ja kuvattu kuvassa 1.

Taulukko 1. Robbo Lab osat. (Robbo 2016)

1.	Liukusäädin, potentiometri
2.	Kolme värilediä; punainen, keltainen sekä vihreä
3.	Ohjattu kahdeksan ledin matriisi
4.	Kaiutin
5.	5 kpl painikkeita
6.	Reset -painike
7.	D13 -liitin, digitaalisen datan lukemista ja kirjoittamista varten (0/1)
8.	A1-liitin, analogisen datan lukemista varten (0–100). Liitintä voi käyttää myös D13 liittimenä.
9.	A0 -liitin, joka toimii kuten kohdan 8. A1 liitin.
10.	Arduino Uno -yhteensopivat liitinlinjat.
11.	Dip -kytkimet, joiden asentoa muuttamalla saadaan robotin toimintoja muutettua. Vaihtoehdolla 1-On = Osa komponenteista (1,3,4,12,13) ovat irti kytkettyjä, jolloin Robbo Labia voidaan käyttää kuten Arduino Unoa. Painikkeet ja Liittimet (A0, A1, D13) ovat toiminnassa. Kaikki linjan 10 liittimet ovat toiminnassa. 1-OFF = Kaikki Robbo Labin omat komponentit ovat käytössä. Osa linjan 10 liittimistä ovat liitettyinä Labin omiin komponentteihin. 2-ON = sarjaviestintä siirtyy liitinlinjalle 10 (D0/RX, D1, TX). USB-liitin (FTDI piiri) on pois käytöstä. Ohjelmoitaessa kytkimen pitäisi olla tässä asennossa.
12.	Valosensori (fototransistori).
13.	Äänisensori (mikrofooni).
14.	USB-liitin

Vaikka laite ominaisuuksiltaan sitä tukeekin, testauksessa Robbo Lab -laitteen kohdan 11 oletusasetuksia ei muutettu, eikä ulkoisia komponentteja käytetty testissä.

#### 4.1.2 Robbo Lab käyttöönotto

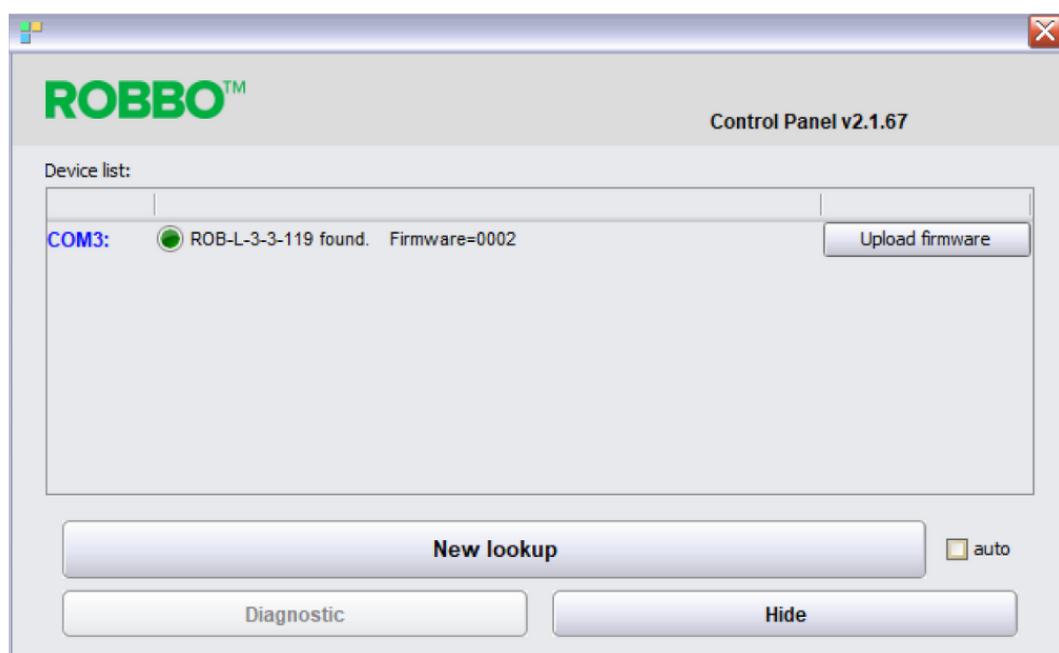
Robbo Lab toimitetaan koottuna pakkauksessaan. Pakkaus sisältää Robbo Lab -laitteen, Lab-pikaoppaan sekä USB-johdon, jonka avulla laite kiinnitetään tietokoneeseen ohjelmointia varten (Kuva 2.). Robbo Lab saa myös virtansa USB-johdon avulla.



Kuva 2. Robbo Lab pakkaus

Liitettäessä tietokoneeseen tarvittavat USB FT232R -ajurit asentuvat automaattisesti. Robbo Lab -laitteeseen syttyy power -valo. Laite pitää myös äänimerkin yhteyden muodostumisen merkiksi. Yhteyden muodostumisen merkiksi avautuu laiteohjauspaneeli (Kuva 3.). Tämän jälkeen laite on käyttövalmis ohjelmointia varten. Ohjauspaneelin kautta saa päivitettyä myös laiteohjelmiston.

Ohjauspaneeli näyttää myös, mikä Robbo -laite on kytkettynä koneeseen sekä mihin porttiin se on kytketty.



Kuva 3. Robbo Lab laiteohjauspaneeli

Varsinainen ohjelmointi Robbo Lab laitteelle voi tapahtua valmistajansa kehittämällä Robbo Scratch -sovelluksella, joka perustuu avoimen lähdekoodin visuaaliseen Scratch -ohjelmointikieleen. Ohjelmointia voi tehdä myös Arduino IDEllä, tai Snap4Arduinolla. (Robbo 2016)

Laitteen toimivuuden selvityksessä havaittiin, että laitteen saattaminen käyttövalmiiksi ohjelmointia varten oli helppoa ja nopeaa, eikä vaatinut erillistä rakentamista. Laitteen käyttöönotto tapahtui ohjeen mukaan odotetusti. Laitteen kaikki komponentit todettiin testauksen aikana ehjiksi ja toimiviksi.

Testauksen aikana ohjelmoitiin Robbo Labia käyttäen Robbo Scratch -sovellusta Windows-ympäristössä. Toiminnallisuutta ja käytettävyyttä havainnoitiin ohjelmointitehtävien luomisen aikana sekä myös pilottiryhmän kanssa tehdyn testauksen aikana.

#### 4.2 Ohjelmointityökalu Robbo Scratch

Koska Robbo Scratch on sovellus, joka on johdannainen Scratch -ohjelmointikielestä, on luonnollista avata ensin Scratch -sovelluksen taustoja sekä käyttää tätä Robbo Scratch -sovelluksen toiminnallisuuden havainnoinnin apuna.

#### 4.2.1 Scratch-sovellus Robbo Scratchin taustalla

Scratch on Scratch Foundationin kehittämä projekti yhteistyössä MIT Media Labin Lifelong Kindergarten Groupin kanssa. Scratch ei ole pelkästään visuaalinen ohjelmointikieli vaan myös online-yhteisö, jossa käyttäjät voivat ohjelmoida ja kirjautuneina käyttäjinä jakaa interaktiivista mediaa -projektejaan- kuten tarinoita, pelejä ja animaatioitaan Scratch -kehitysympäristöön. Scratch on lähinnä suunnattu 8–16-vuotiaille lapsille. Se on saatavilla ilmaiseksi ja käytettävissä verkossa osoitteessa <https://scratch.mit.edu>. Scratch toimii useimmissa nykyisissä selaimissa tietokoneissa ja tableteissa. Puhelimella voi tarkastella projekteja, mutta ei luoda tai muokata. Scratchia käytetään yli 150 maassa ja se on saatavilla yli 40 kielellä, myös suomen kielellä. (Lifelong Kindergarten Group MIT Media Lab 2020)

Scratchin ensimmäinen versio on julkaistu vuonna 2007 ja sen kehitystyö on jatkunut siitä lähtien. (Scratchers 2008)

Whatis:in mukaan Scratch on objektikeskeinen ohjelmistokehityspaketti, SDK, eli sisältää valmiiksi kääntäjän debuggerin ja ohjelmakehityksen. Scratchilla on nollakoodiliittymä, mikä tarkoittaa, että lähdekoodin kirjoittamisen sijaan käyttäjät kokoavat visuaalisia yhdeksään eri värikoodattuun toiminnallisuusluokkaan ryhmiteltyjä ohjelmointilohkoja yhteen muodostaen ohjelmakoodin. Jokaisella ohjelmointilohkolla on oma komentonsa ja yksilöllinen toimintonsa. Ohjelman valmistuttua käyttäjä käynnistää ohjelman suorituksen napsauttamalla vihreää lippua. Ohjelma sammuu napsauttamalla punaista merkkiä. (TechTarget 2017)

Käyttöliittymä on intuitiivinen ja helppokäyttöinen ohjelmointilohkojen värikoodien avulla, minkä vuoksi se sopii lasten ohjelmoinnin alkeiden opetukseen loistavasti. Scratch -yhteisöstä 14.10.2020 katsotun tilastotiedon mukaan Scratchilla on yhteensä 60 078 492 rekisteröitynyttä käyttäjää. Suomesta niitä on 155 332. Yhteisöstä löytyy kaikkiaan 61 396 603 jaettua projektia. (Lifelong Kindergarten Group MIT Media Lab 2020)

Scratchin graafiset hahmot (Sprite) ja taustat luovat pohjaa olio-ohjelmoinnille. Olioihin voidaan liittää käskyjä ja muuttujia raahaamalla hiirellä halutut ohjelmointipalikat ohjelmointialueelle ja yhdistämällä niitä toisiinsa kuten Lego-palikoita. Ohjelmointipalikat sisältävät yleisimpiä ehto- ja ohjauksrakenteita, perustason matemaattisia operaatioita sekä muuttujien käyttöä. Olioita voi visuaalisesti liikuttaa ja animoida ja niiden

koordinaatistosijainteja, suuntaa sekä törmäyksiä voi myös ohjelmoida. Scratchissa on myös omat lohkot -lisäpalikat vastaamassa funktioita tai metodikutsuja. (Sirén 2017)

Scratchin käyttäjien määrästä voidaan päätellä ohjelman saavuttamaa suosiota ja tunnettavuutta sekä sen toimivuutta. Tämä taas puolestaan lisää ohjelman luotettavuutta käyttäjä sitä myös opetuskäytössä.

#### 4.2.2 Robbo Scratch johdanto

Robbo Scratch on Robbon kehittämä visuaalinen ohjelmointikäyttöliittymä, sovellus. Robbo Scratch on avoimeen lähdekoodiin perustuvan Scratch -ohjelmointikielen muokattu versio. Siinä on kaikki vakio-ohjelmointiympäristön toiminnot. (Robbo 2016)

Muokatun version eli Modin tekemisen yleinen tarkoitus on yleensä uusien ominaisuuksien tai toiminnallisuuksien lisääminen alkuperäiseen Scratch -ohjelmaan. Lisäominaisuuksien ja/tai -lohkojen vuoksi alkuperäinen Scratch -ohjelma ja online -versio eivät tue Mod -versiossa tehtyjä projekteja. (Scratchers 2008)

Lisäominaisuuksiksi on lisätty Robbo Lab -laitteen toiminnallisten osien edellyttämät ohjelmointilohkot.

Robbo Scratch on paikallisesti asennettava Flash -pohjainen sovellus, ei online -versio.

#### 4.2.3 Robbo Scratch – käyttöönotto

Koska Robbo Scratch on paikallisesti asennettava sovellus, testaus aloitettiin lataamalla ja asentamalla sovellus tietokoneelle. Asennus aloitettiin lataamalla tarvittava asennustiedosto osoitteesta <https://www.robbo.world/apps/>.

Tiedosto latautui ensin koneelle, jonka jälkeen ladattu tiedosto käynnistettiin asennusta varten. Asennuksen aikana havainnoitiin monta vaihetta, jossa käyttäjän piti reagoida asennukseen valitsemalla asennusohjelmasta toimintoja. Valinnat oli tehty siten, että käyttäjän ei tarvinnut muuttaa asetuksia, joten asennus meni läpi hyväksymällä oletukset. Asennuksen jälkeen työpöydälle syntyi sovelluksen kuvake, jota klikkaamalla sovellus aukeaa. Ohjelma oli näin käyttövalmis. Asennusohjelma oli englanninkielinen.

Robbo Scratch -sovellusta voi käyttää ilman Robbo Lab -laitettakin, mutta ohjauslohkot eivät suorita toimintoja ilman laitteen kytkemistä tietokoneelle.

#### 4.2.4 Scratch ja Robbo Scratch – vertailu ja havainnot

Ensinäkemältä Robbo Scratchin ulkonäkö ei poikkea huomattavasti Scratch -ympäristöstä kuten kuvista 4. sekä kuvasta 5. voidaan havaita.

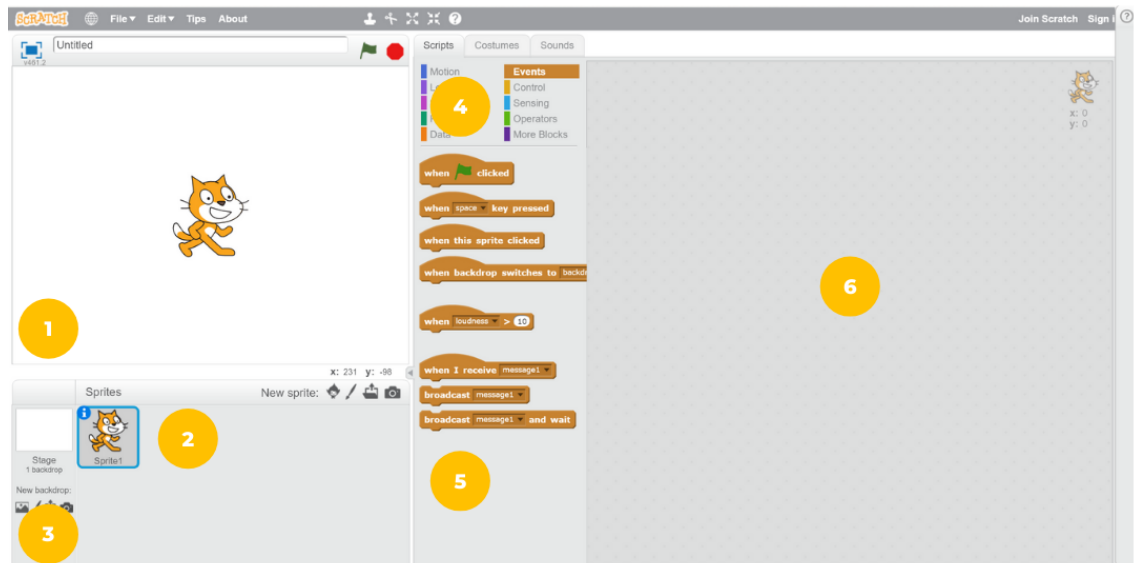
Yhtenevää molemmissa sovelluksissa on ohjelmointiprojektin käynnistys ja pysäytyspainikkeet sekä värimaailma, osioiden sijainti sekä leikkaa-, liimaa-, suurena-, pienennä- sekä leimaa-kuvakkeet. Molemmissa sovelluksissa saa myös valittua kielen suomeksi muiden kielivalikoiden joukosta.

Molemmista editoreista löytyvät seuraavat taulukossa 2. kuvatut osat.

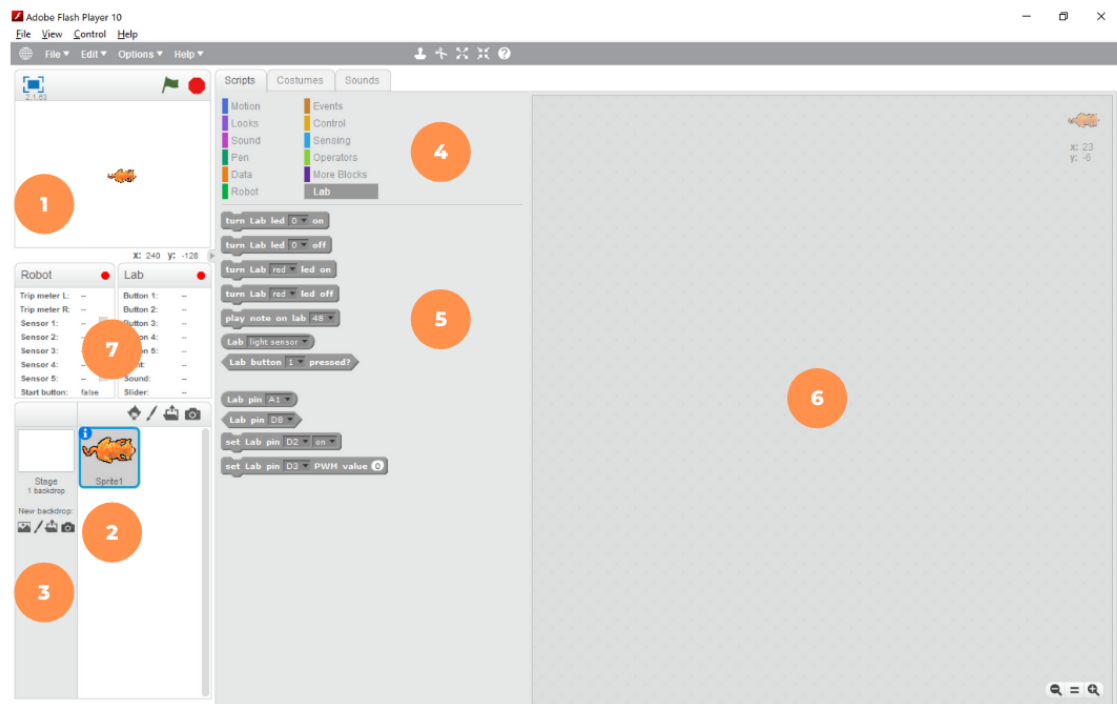
Taulukko 2. Sovellusten osat

1.	Esiintymislava
2.	Hahmojen valinnat ja ominaisuudet
3.	Taustan valinnat ja ominaisuudet
4.	Värikoodatut toiminnallisuusluokat
5.	Aktiivisen toiminnallisuusluokan ohjelmointilohkot
6.	Ohjelmointialue





Kuva 4. Scratch -online-käyttöliittymä



Kuva 5. Robbo Scratch -sovelluksen käyttöliittymä

Eroavaisuutta löytyy tarkemman tutkimuksen jälkeen niin ulkonäössä kuin toiminnallisuudessaakin. Robbo Scratchiin on lisätty kuvan 5- kohdassa 4. näkyvät toiminnallisuusluokat Robot ja Lab (aktiivisena), sekä niihin liittyvät ohjelmointilohkot, jotka näkyvät

kuvan 5 kohdassa 5. Kuvan 5. kohdassa 7 voidaan seurata myös reaaliajassa Robot- ja Lab -sensorien ja -painikkeiden arvoja.

Myös aloitushahmon ulkonäkö poikkeaa lisenssistä johtuvista syistä. Robbo Scratch -sovelluksessa on erilainen hahmo ja taustakirjasto, kuin Scratchin online -versiossa.

Opinnäytetyön toimeksiantaja käyttää koodauksen alkeisryhmällä opetuslustoana Scratch -ympäristöä suomen kielellä. Scratchissa on mahdollista luoda tunnukset ja kirjautuneena käyttäjänä tallentaa omia projekteja ja harjoituksia. Toimeksiantajalla on käytössä koodiryhmille ryhmäkohtaiset yhteistunnukset. Näin koodiryhmän oppilaat näkevät toistensa tekemät ohjelmointiharjoitukset sekä projektit, vaikka niitä ei olisikaan jaettu Scratch -yhteisölle.

Koska Robbo Scratch on sovellus, eikä verkossa toimiva kuten Scratch, ohjelman käyttötapa eroaa toimeksiantajan toimintatavoista käyttäessä Scratchia opetuksessa. Robbo Scratch -sovellukseen ei kirjauduta sisälle kuten Scratchiin. Ohjelma on saatavilla vain sille asennetulla koneella. Mikäli tunnin aikana oppilaalle tulee tietokoneen kanssa ongelmia, hän käyttää toimeksiantajan tietokonetta tunnilla tai hänellä on satunnaisesti opitunneilla oma mutta eri tietokone mukana, omien projektien käyttö edellyttää niiden siirtoa erilliselle siirrettävälle tallennusmedialle. Ongelmia voi tulla myös jo siinä, että toimeksiantajasta riippumattomista syistä ohjelman asennus koneelle ei onnistu. Projektien tallennus suoritetaan paikallisesti omalle tietokoneelle eikä tallennus toimi tutulla tavalla automaattisesti palvelimelle. Oppilaat eivät pääse näkemään toistensa tekemiä ohjelmointiharjoituksia tai projekteja, eikä projekteja pääse jakamaan yhteisölle. Opettaja ei pystyy seuraamaan etänä oppilaidensa harjoituksia eikä lisäämään harjoitustehtäviä yhteisen tunnuksen omat tavarat osioon.

Ohjelman käyttö edellyttää myös tietoteknisten perustaitojen hallintaa enemmän kuin online -versio vaatii.

#### 4.3 Ohjelmointitehtävien suunnittelu ja valmistusprosessi

Ohjelmointitehtävien suunnitteluun lähdettiin asettamalla opitunneille teemat sekä tavoitteet tunnin aikana opeteltaviksi asioiksi. Tunnin rakenteen mukaisesti tarkoitus oli tuottaa ohjelmointitehtävistä teemalliset tuntikortit, jonka kanssa oppilas voi itse edetä omalla tahdillaan, saavuttaen oppitunnin aikana omalle osaamistasolleen sopivan lopputuloksen.

Teemaoppimisen näkökulmasta oppimateriaalia voidaan rakentaa esimerkiksi joko niin, että ajatellaan teemaa abstraktisesti esim. liikkeen tai valojen toteutusta, tai subjektiivisesti kuten Robbo Lab -laitteen rakenteiden eri osia ajatellen, kuten toimintopainikkeita, ledejä tai vaikka mikrofonia ja luodaan niille erilaisia käyttötapoja.

Pedagogisesti materiaalin valmistuksessa huomioitiin aiemmin käsitellyn Innokas-verkoston oppaan huomioita ja mukailtiin sen rakennetta, kuitenkin toteuttaen toimeksiantajan toiminta-ajatusta.

Ohjelmointitehtävien teemakorttien visuaalisuus toteutettiin samaan ulkoasuun kuin muutkin toimeksiantajan tuntiteemat, joita olin toteuttanut toimeksiantajalle vuoden 2017 aikana. Tuntikorttien tutuus ja selkeys sekä samankaltaisuus muiden oppimateriaalien kanssa auttavat oman kokemukseni mukaan ohjaamaan oppilaita myös tutulla tavalla etenemään tehtävissä tunnilla, itsenäisesti ja intuitiivisesti.

Ohjeen graafiset ohjeet:

- Ylin otsikko: Tw Cen MT koko 26
- Pääotsikko eli tunnin aihe: Tw Cen MT koko 42
- Muiden sivujen otsikko + tunnin aihe: Tw Cen MT koko 20
- Leipäteksti: Calibri koko 14.
- Tekstin väri otsikoissa ja tehostepalloissa: RGB: 255, 192, 0
- Harmaat tehostepallot: RGB:192,192,192
- Kuvien ympärillä katkoviiva: RGB: 255, 192, 0
- Sivut taitettu "(c) Suomen Tiedekoulu Oy vvvv" - pohjaan

Tuntikorttien teossa pyrittiin käyttämään tekstien lisäksi paljon käsiteltävää asiaa havainnollistavia kuvia. Kuvat olivat osittain kameralla otettuja omasta testitilanteesta, osittain kuvankaappauksia, osittain itse kuvankäsittelyohjelmalla tehtyjä.

Tuntikorttien taittamiseen käytettiin Microsoft Publisher 2016 -ohjelmaa, jonka taitto-ominaisuudet soveltuvat tämänkaltaisten ohjeiden tekemiseen. Tuntikortit tallennettiin pdf -muotoon. Oppilaille tuntikortit jaettiin käyttöön käyttämällä Google Drivea.

Tässä luvussa kerrotaan kolmen ensimmäisen tuntikortin rakenteesta.

#### 4.3.1 Tuntikortti 1 kuvaus

Ensimmäisen oppitunnin teemana sekä tavoitteena oli Robbo Scratch -sovelluksen asennus, ohjelmointiympäristöön tutustuminen, Robbo Lab -laitteen perustoimintoihin tutustuminen sekä ensimmäisen Robbo Lab sovelluksen ohjelmoiminen (Liite 1.).

Teeman rakenteessa käytettiin vaiheittaista numeroitua opastusta. Tuntikorttiin pyrittiin saamaan joka vaiheesta mahdollisimman yleispätevä ja tilannetta havainnoiva kuva.

Teeman teossa huomioitiin, että tunneilla on aina läsnä opettaja opastamassa ja näyttämässä omalla esimerkillään teeman suoritusta ja kertomassa vaiheiden suoritukseen liittyvistä tarkennuksista. Mahdollisista virheistä tai asennuksen aikana tapahtuvista ongelmista tunnin opettaja pyrkii suoriutumaan itse. Mahdollisia virheitä voi ensimmäisessä teemassa tulla, jos oppilaalla ei ole oikeuksia asentaa itse koneelleen mitään, tai oppilas käyttää järjestelmää, johon Robbo Scratch ei pysty asentumaan. Vastaan voi myös tulla tilanteita, jossa oppilaan koneesta puuttuu USB-portti.

Ohjeen teossa huomioitiin, että usein tämän ikäryhmän lapsilla ei ole kokemusta tietokoneisiin liitettävien lisälaitteiden termeistä, käytöstä, eikä niiden toiminnoista. Termejä ei lähdetty tarpeettomasti avaamaan, jotta keskittyminen pysyy varsinaisessa tekemisessä. Tuntikorttiin pyrittiin saamaan kuvattua ohjelman ja laitteen asennus tietokoneelle niin, että vanhemmat oppilaat pystyvät tämän tekemään itsenäisesti ja nuoremmille oppilaille opettajan sanallinen opastus tuntikortin lisäksi auttaa vaiheiden suorituksessa.

Koska tuntikortti on taitettu pdf -muotoon, kohdassa yksi oleva Robbo Scratch -latauslinkki toimii suoraan avaamalla selaimen oikean sivun, jolloin oppilaille ei tarvitse ohjeessa erikseen opastaa selaimen aukaisemista, tai osoitteen kirjoittamista selaimen osoiteriville. Kohdasta kaksi kohtaan neljä havainnoivat kuvat auttavat oppilasta löytämään oikeat osat sekä toiminnot.

Tuntikortin kohdassa viisi näytetään Robbo Scratchin ja Scratchin erottavat Labin omat ohjauslohkot sekä sensorien ja painikkeiden arvonäkymä. Näiden näkymien sisällön tunnin opettaja voi tarkentaa ryhmälleen sanallisesti riippuen ryhmän lähtötilanteesta. Koska Robbo Scratch on asennettuna paikallisesti koneelle, on opettajan vielä opastettava oppilaille projektien tallennuskansion luominen sekä tiedostojen tallennus ja avaus. Oletuksena on, että ryhmä on käyttänyt jo Scratchia kaksi lukukautta, jolloin ohjelman

perusulkonäköön ja toimintojen tutustumiseen ei tarvitse käyttää aikaa, oppilas huomaa sovelluksen tuttuuden jo avaamishetkellä.

Robbo Lab -laboratorion osiin tutustuminen on tehty liittämällä ohjeeseen kuva Robbo Labin osista selostuksineen. Tarkoitus on, että opettaja sanallisesti kertoo tärkeimpien osien nimet ja havainnollistaa niiden sijainnit Robbo Lab -laitteella. Tarkempi tutustuminen eri osien toimintoihin tehdään myöhemmin eri teemojen merkeissä.

Ensimmäiseen teemaan on otettu mukaan myös ensimmäisen sovelluksen teko Robbo Scratchilla tutustumalla samalla Robbo Labin painikkeisiin. Toimeksiantajan aiempien kausien Scratch harjoitusten kautta oppilaille on jo tuttua hahmon liikuttaminen ohjausnäppäimin näppäimistöllä sekä hiirellä. Nyt tarkoitus on liikuttaa hahmoa Robbo Labin painikkeilla.

Kerrattavat ohjelmointikäskyt ovat perusohjelman rakenne, tapahtumat eli eventit, toistosilmukka, ehtolause, boolean-ehto, hahmon liikuttaminen. Uusi opittava ohjelmointikäsky on Labin painikkeen toiminta, jossa on myös boolean-ehto. Tuntikortissa on kerrottu muistutukseksi millä lohkoilla käskyt tehdään. Myös tallennuksen tekeminen on uutta, joten sen harjoittelu on tärkeää.

Painikkeiden arvojen muutoksen huomioiminen tehtävää suoritettaessa on myös tärkeää. Näin oppilas pääsee havainnoimaan konkreettisesti sulautetun järjestelmän ja ohjelman reaaliaikaista keskustelua sarjaportin kautta.

Tuntikorttiin on tehty malliratkaisu niitä oppilaita varten, jotka eivät omaa ratkaisua keksi itsenäisesti.

Tuntikorttiin on lisätty myös haastetehtävä edistyneimmille oppilaille. Haastetehtävälle ei ole annettu malliratkaisua. Oppilaan oletetaan löytävän tehtävään ratkaisu soveltamalla harjoitustehtävän malliratkaisua sekä jo aiemmin oppimaansa.

#### 4.3.2 Tuntikortti 2 kuvaus

Toisen oppitunnin teemana sekä tavoitteena oli tutustua Robbo Lab -laitteen ääniominaisuuksiin ja ledeihin sekä ohjelmoida näitä käyttämällä oma Robbo Lab -sovellus (Liite 2.).

Tuntikortin teossa käytettiin samaa visuaalista tyyliä kuin Tuntikortissa 1. sekä numeroitua opastusta. Tuntikortissa käytettiin myös joka vaiheesta mahdollisimman yleispätevää kuvitusta havainnoinnin apuna.

Aluksi käydään läpi teeman tavoitteet sekä johdatetaan teemaan kertomalla lyhyesti käsiteltävät komponentit. Kerrattavat ohjelmointikäskyt ovat tapahtumat eli eventit, odota - lohkon käyttö sekä funktiot, jotka toteutetaan lisää lohkoja -käyttämällä. Uudet opittavat ohjelmointikäskyt ovat Labin ledien sekä nuottien ohjelmointilohkojen käyttö. Tuntikortissa on kerrottu muistutukseksi millä lohkoilla käskyt tehdään.

Teeman alustuksessa aloitetaan kertaamalla, miten laite kytketään koneeseen ja mikä ohjelmisto avataan. Alustuksessa tutustutaan myös ledien, kaiuttimen sekä äänisensorien sijainteihin laitteella. Osiin tutustuminen on tehty liittämällä ohjeeseen kuva Robbo Labin osista ja selostukseen tuotu vain teemassa käsiteltävät komponentit selkeyden vuoksi. Teeman osiot 1–3 on suunniteltu käytäväksi yhdessä läpi ennen varsinaista tehtävän suorittamista. Tarkoitus on, että opettaja sanallisesti kertaa tärkeimpien osien nimet ja havainnollistaa teeman komponenttien sijainnit Robbo Lab -laitteella.

Varsinainen tunnin teema alkaa tehtävän 2. kerronnalla, jonka jälkeen kerrotaan teoriaa sekä ääniominaisuuksiin että ledien toimintaan. Näin oppilas voi halutessaan tehdä tehtävänannon mukaisen ratkaisun itse. Ohjeessa on kuvattuna myös mitä komentoryhmän lohkoa on tarkoitus käyttää. Opettajan on tarkoitus opastaa ja näyttää omalla esimerkillään teeman suoritusta ja kertoa vaiheiden suoritukseen liittyvistä tarkennuksista. Tarkoitus on myös innostaa tutkimaan ja kokeilemaan itse, miten arvot muuttuvat ja miten se vaikuttaa laitteen toimintaan.

Tuntikorttiin on tehty tehtävään 2. ratkaisumalli niitä oppilaita varten, jotka eivät omaa ratkaisua keksi itsenäisesti. Ratkaisumallissa on esitetty ratkaisut äänet ja ledien lisäämiselle erikseen.

Tuntikorttiin on lisätty myös kaksi haastetehtävä edistyneimmille oppilaille. Haastetehtäville ei ole annettu malliratkaisua, vaan oppilaan oletetaan löytävän tehtävään ratkaisu soveltamalla harjoitustehtävän malliratkaisua sekä jo aiemmin oppimaansa.

### 4.3.3 Tuntikortti 3 kuvaus

Kolmannen oppitunnin teemana sekä tavoitteena oli tutustua Robbo Lab in liikusäätimen toimintaan sekä ohjelmoidaan oma Robbo Lab sovellus, jossa käytetään liikusäädintä. (Liite 3.)

Ohjeen teossa käytettiin samaa visuaalista tyyliä kuin Tuntikortissa 1 ja 2 sekä numeroitua opastusta. Tuntikortissa käytettiin myös joka vaiheesta mahdollisimman yleispätevää kuvitusta havainnoinnin apuna.

Aluksi käydään läpi teeman tavoitteet sekä johdatetaan teemaan kertomalla lyhyesti liikusäätimen toimintaperiaate. alustuksessa kerrotaan myös ohjelmointiosuudessa vastaan tulevat kerrattavat ja uudet asiat.

Teeman alustuksessa aloitetaan kertaamalla, miten laite kytketään koneeseen ja mikä ohjelmisto avataan. Alustuksessa tutustutaan myös liikusäätimen sijaintiin laitteella. Liikusäätimeen tutustuminen on tehty liittämällä ohjeeseen kuva Robbo Labin osista ja selostukseen tuotu vain teemassa käsiteltävät komponentit selkeyden vuoksi. Teeman osiot 1 - 3 on suunniteltu käytäväksi yhdessä läpi ennen varsinaista tehtävän suorittamista. Tarkoitus on, että opettaja sanallisesti kertoo osien nimet ja havainnollistaa liikusäätimen sijainnin Robbo Lab -laitteella.

Varsinainen tunnin teema alkaa tehtävän 3 kerronnalla sekä viitataan ensimmäisellä kerralla tehtyyn tehtävään, jota oppilas voi halutessaan hyödyntää ja näin jatkaa suoraan uuteen opittavaan asiaan. Tämä opettaa myös rakentamaan uutta aiemman opitun asian päälle. Tuntikortti tukee näin myös eri tasolle ehtineitä oppilaita.

Teeman teoria liikusäätimen käytöstä on suunniteltu käytäväksi yhdessä läpi ennen varsinaista tehtävän suorittamista.

Ohjeessa on kuvattuna mitä komentoryhmän lohkoa on tarkoitus käyttää. Opettajan on tarkoitus opastaa ja näyttää omalla esimerkillään teeman suoritusta ja kertoa vaiheiden suoritukseen liittyvistä tarkennuksista. Tarkoitus on myös innostaa tutkimaan ja kokeilemaan itse, miten liikusäätimen arvot muuttuvat ja miten se vaikuttaa laitteen toimintaan. Näin oppilas pääsee havainnoimaan konkreettisesti sulautetun järjestelmän ja ohjelman reaaliaikaista keskustelua sarjaportin kautta.

Kerrattavat ohjelmointikäskyt ovat tapahtumat eli eventit, toistosilmukka, ehtolause, boolean-ehto, hahmon liikuttaminen, laboratorion painikkeen toiminnan ohjelmointilohkot, muuttujat, pomppaa sekä vertailulausekkeet. Uutta opittavaa on laboratorion liikusäätimen toiminnan ohjelmointilohko. Teemassa harjoitellaan myös aiemmin tehdyn tehtävän tallennuksen avaamista.

Tuntikorttiin on tehty tehtävään 3. ratkaisumalli niitä oppilaita varten, jotka eivät omaa ratkaisua keksi itsenäisesti.

Tuntikorttiin on lisätty myös haastetehtävä edistyneimmille oppilaille. Haastetehtävälle ei ole annettu malliratkaisua, vaan oppilaan oletetaan löytävän tehtävään ratkaisu soveltamalla harjoitustehtävän malliratkaisua sekä jo aiemmin oppimaansa.

#### 4.4 Robbo Lab – kevään 2018 testauksen tulokset

Keväällä 2018 aloitettiin Robbo Lab -laitteelle sekä Robbo Scratch -ohjelmointialustalle suunniteltujen ja mallinnettujen tarinallisten ohjelmointitehtävien testaukset toimeksiantajan kolmessa toimipisteessä kolmen opettajan voimin. Ohjasin itse Piikkiössä 7 oppilaan ryhmää, Turussa Nummessa testauksessa olivat mukana 4 + 12 oppilaan ryhmät sekä Olarissa 5 + 6 oppilaan ryhmät. Ryhmät olivat taustoiltaan eri oppimispoluista. Opettajat saivat tuntikortit 1.–3. testaukseen. Opettajilla oli myös käytössä Innokas-verkoston Robbolle valmistama pedagoginen materiaali.

Testauksen tarkoituksena oli selvittää, miten tehdyt tuntikortit toimivat eri lähtötasoisilla ryhmillä sekä minkälaisen vastaanoton Robbo Lab -laite saa. Testaukseen käytettiin kolmesta viiteen (3–5) opetuskertaa riippuen ryhmästä. Myös laitteiden määrä toimipisteittäin vaihteli, kaikilla oppilailla ei välttämättä ollut joka kerta laitetta saatavilla.

Tuntien jälkeen opettajat raportoivat omia sekä oppilaidensa kokemuksia opinnäytetyön tekijälle suullisesti. Palautteiden perusteella mietittiin seuraavan tunnin materiaalin muu-  
tostarpeita. Kevään koodikoulun viimeisillä viikoilla toukokuussa 2018 annettiin oppilaille ja opettajille mahdollisuus vastata Google Formsilla toteutettuun kyselyyn Robbo Labin käytöstä sekä tuntikorttien laadusta ja sisällöstä (Liite 4.). Kyselyyn vastattiin anonymisti ja sitä analysoitiin kvalitatiivisesti. Kyselyyn saatiin yhdeksän vastausta. Vastausten perusteella parannettiin ohjeiden sisältöä sekä saatiin materiaalia opinnäytetyötä varten.



Kyselyyn vastasi yksi opettaja ja kahdeksan oppilasta. Vastanneet oppilaat olivat iältään 9–13 vuotta ja ohjelmointikokemus koodikoulusta vaihteli 1 ja 3 vuoden välillä. Seitsemän oppilasta käytti laitetta yksin, yksi parin kanssa.

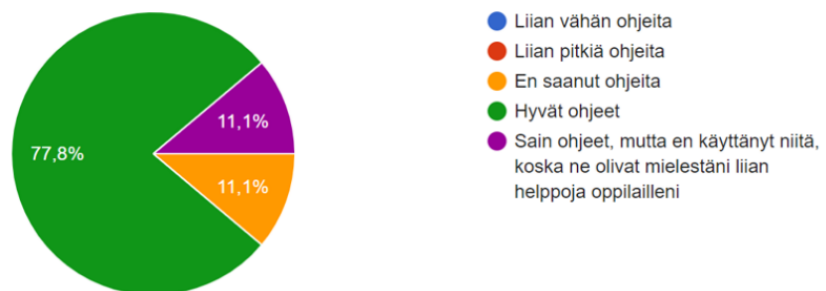
## Ohjeiden laatu

” Oliko Tiedekoulun (ei Innokas-materiaali) Robbo Lab PDF -ohjeistus mielestäsi...” (Kuva 6.)

Seitsemän oppilasta valitsi vastausvaihtoehdoksi ”Hyvät ohjeet”. Kyselyyn vastanneen opettajan mielestä sen sijaan ohjeet olivat liian helppoja: ”Sain ohjeet, mutta en käyttänyt niitä, koska ne olivat mielestäni liian helppoja oppilailleni”. Kukaan vastaajista ei valinnut vastausvaihtoehtoja ”Liian vähän ohjeita” tai ”Liian pitkiä ohjeita”.

### Oliko Tiedekoulun (ei Innokas-materiaali) Robbo Lab PDF- ohjeistus mielestäsi...

9 vastausta



Kuva 6. Kysymys ohjeista

Ohjeista pyydettiin myös kommentteja ja mielipiteitä avoimella kysymyksellä. Tämän kysymyksen vastausten mukaan ohjeet olivat yksityiskohtaiset, selkeät ja hyvät. Yksi vastaaja olisi kaivannut ohjeisiin lisää esimerkkejä siitä mitä kaikkea Robbo Lab -laitteella voi tehdä.

Näiden vastausten perusteella ohjeisiin lisättiin haastetehtäviä sekä nostettiin hieman vaikeustasoa. Yksityiskohtia ei kuitenkaan poistettu.

## Laitteen hyödyllisyys

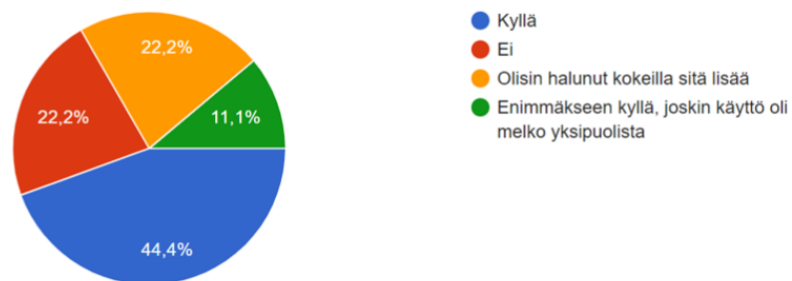
Robbo Labin hyödyllisyyttä ja motivointikykyä selvitettiin kyselyssä kysymällä suoraan laitteen hyödyllisyydestä kurssilla, sekä myös kysymällä haluaisiko vastaaja jatkaa työskentelyä laitteella.

”Oliko Robbo Lab mielestäsi hyödyllinen kurssilla?” (Kuva 7.)

Vastaajista neljä vastasi Kyllä ja yksi vastaajista koki, että Robbo Lab oli hyödyllinen, mutta käyttö oli melko yksipuolista. Kaksi vastaajaa olisi halunnut kokeilla Robbo Labin käyttöä vielä lisää.

Oliko Robbo Lab mielestäsi hyödyllinen kurssilla?

9 vastausta



Kuva 7. Kysymys hyödyllisyydestä

Hyödyllisyyttä selvitettiin myös kysymällä

”Haluatko jatkaa Robbo Labin kanssa työskentelyä?”

Kaksi vastaajaa vastasi kysymykseen myönteisesti ja seitsemän vastaajaa vastasi ”ehkä”. Kukaan ei valinnut ei-vaihtoehtoa.

Näiden kysymysten vastausten perusteella voidaan päätellä, että Robbo Lab -laitteen kanssa jatkoryhmäläiset olivat motivoituneita ohjelmoimaan ainakin testauksessa tarjottun viiden oppitunnin verran.

## Laitteen käyttö

” Oliko Robbo Labin käyttö mielestäsi...”

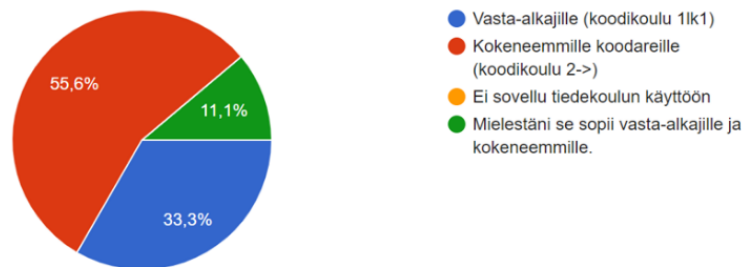
Kahdeksan vastaajaa yhdeksästä koki, että Robbo Labin käyttö oli helppoa ja vain yhden vastaajan mielestä vaikeaa.

”Kenelle Robbo Labin mielestäsi sopii?”(Kuva 8.)

Viiden vastaajan mielestä Robbo Labin käyttö on sopivaa toimeksiantajan koodikoulujen jatkoryhmille ja kolmen vastaajan mielestä laite voisi sopia jo alkeisryhmäläisille. Yksi vastaaja oli sitä mieltä, että laite sopii molemmille ryhmille.

#### Kenelle Robbo Lab mielestäsi sopii?

9 vastausta



Kuva 8. Kysymys kohderyhmästä

”Perustelu edelliseen kysymykseen, kenelle Robbo Lab mielestäsi sopii?”

Tähän kysymykseen sai vastata omin sanoin. Vastausten perusteella voidaan päätellä, että Robbo Lab laitteen käyttö ei ole liian yksinkertainen vaan sopivasti haastava, mutta vaatii käyttäjiltään jo osaamista ohjelmoinnista, jotta laitteen käyttö on luontevampaa.

Kaksi vuotta koodikoulussa ohjelmoinut 12-vuotias oppilas testasi Robbo Lab -laitetta antaen neljän oppitunnin testauksen perusteella palautteen, että laite sopii ohjelmointiin vasta-alkajille, ” koska se oli simppelempi toimiva ja sitä oli yksinkertaista käyttää”. Kun taas myös kaksi vuotta koodikoulussa ohjelmoinut 9-vuotias oppilas mielsi laitteen olevan sopiva kokeneemmille koodareille.

Hyvien ohjeiden ja selkeän ohjelmointiympäristön puolesta Robbo Scratch -ohjelma miellettiin sopivaksi vasta-alkajille, mutta sopivaksi kokeneemmillekin koodareille. Sen sijaan sopivien haastavien tehtävien keksiminen Robbo Scratch/Robbo Lab -yhdistelmästä koettiin hankalaksi. Yksi vastaaja oli myös sitä mieltä että ”Robbo Lab ei tuo paljoo lisäarvoa Scratchiin, joten sen uutuudenviehätys sopii vasta-alkajille.”

Robbo Lab -laitteesta saatiin kyselyssä myös palautetta laitteen virheellisistä toiminnoista. Vastauksissa oli mainittu ongelmista liukusäätimen kanssa. Liukusäädin ei ollut testitilanteessa nollaantunut, ellei säädintä painanut kunnolla sivuun. Monet ominaisuudet eivät myöskään toimineet kunnolla samaan aikaan. Myös yhteyden muodostamisen kanssa oli ongelmia. Robbo Lab miellettiin tunneilla peliohjaimeksi.

Robbo Scratch -sovellus sai myös palautetta kyselyssä.

"Kommentteja ja mielipiteitä Robbo Scratchin käytöstä".

Vastausten perusteella Robbo Scratch oli käytettävyydeltään "Ihan ok. Helppo", "Hidas", "Pikkuhiljaa tylsäksi käyvä". Yksi oppilaista oli myös testauksessa huomannut, että hiiren oikea näppäin ei toiminut sovelluksessa kuten Scratchissa oli totuttu.

### **Kyselyn vastausten yhteenveto**

Kyselyä analysoitiin kvalitatiivisesti. Vastauksiin suhtauduttiin myös varauksella ja kriittisesti, vastaajien ollessa lapsia. Tunneilla saadut havainnot sekä suulliset palautteet olivat yhdenmukaisia kyselyn vastausten kanssa.

Yhteenvetona kyselyn palautteista voidaan todeta, että vastaanotto Robbo Lab -laitteelle ja Robbo Scratch -sovellukselle oli suhteellisen neutraali. Oppilaiden mielenkiinto riitti ohjelmoimaan laitteella koko tarjotun materiaalin harjoitukset viiden oppitunnin verran. Lisäksi osa teki omia harjoituksia tämän lisäksi. Laitteen käyttö tunneilla oli varmempaa yli 11-vuotiailla kyselyyn vastaajilla, kun taas nuoremmat vastaajat kokivat laitteen soveltuvan kokeneemmille koodaajille riippumatta siitä, montako vuotta olivat ohjelmoineet. Laite miellettiin peliohjaimeksi, ja vastaajien olikin vaikea keksiä, mitä muuta laitteella voisi tehdä peliohjaimen käytön lisäksi. Vastaajista vain kaksi oli motivoitunut jatkamaan laitteen ohjelmointia, kun taas muiden oppilaiden mielipiteet olivat neutraalit.

Kyselyn vastauksissa oli mainittu myös sekä laitteesta että sovelluksesta löydettyjä pieniä ongelmia.

#### **4.5 Robbo Lab soveltuvuuden johtopäätökset**

Tämän opinnäytetyön toiminnallisessa osuudessa selvisi, että Robbo Lab on kauttaaltaan akryyllilasiin koteloitu, sulautetun järjestelmän laite. Laite sisältää integroituja

käyttäjälle ohjattaviksi tarkoitettuja osia (Kuva 1, Taulukko 1). Integroinnin ansiosta ei käyttäjältä vaadita erillistä sähkötekniikan, elektroniikan tai mekaniikan tuntemusta. Näin laitteen käyttäjä voi keskittyä pelkästään ohjelmointiin. Laitteen saattaminen käyttövalmiiksi ohjelmointia varten oli helppoa ja nopeaa. Laitetta ohjelmoitiin Robbo Scratch -ohjelmalla, joka on Scratch-ohjelmointikielen muokattu versio. Lisäominaisuuksiksi on lisätty Robbo Lab -laitteen toiminnallisten osien edellyttämät ohjelmointilohkot. Robbo Scratch on paikallisesti asennettava Flash-pohjainen sovellus, ei online-versio. Sovellus ei ulkonäöllisesti eronnut Scratchista, joten sovelluksen käyttö ei vaatinut erillistä opettelua. Ohjelman käyttötapa sen sijaan erosi toimeksiantajan toimintatavoista, mm. sovelluksen asennuksen, projektien tallentamisen sekä projektien yhteiskäyttöisyyden ja jakamisen osalta. Robbo Lab -laitteessa, kuten Robbo Scratchissa, havaitut tekniset ongelmat ja bugit vähensivät laitteeseen kohdistuvaa mielenkiintoa ja motivaatiota.

Laitteen fyysisten ominaisuuksien puolesta ongelmia olivat esimerkiksi reset-painikkeen sijainti, joka oli keskellä laitetta. Intuitiivisesti sekä minun että oppilaitteni mielestä tässä kohtaa olisi kuulunut olla jokin muu toiminnallinen painike. Reset-painiketta tuli vahingossa painettua useasti tunnin aikana. Tämä taas aiheutti sen, että laite käynnistyi uudelleen ja yhteyden muodostuminen Robbo Scratchiin oli hidasta ja uudelleenkäynnistymisen odottaminen oli turhauttavaa. Laite saattoi myös muista syistä kadottaa yhteyden kesken ohjelmoinnin. Myös liukusäätimen toiminnassa oli välillä ongelmia.

Robbo Scratch ohjelman käytön aikana havaittiin myös ongelmia mm. hiiren ja pikanäppäinten toimivuuden kanssa. Ohjelman pienet poikkeamat opituista ja totutuista toimintatavoista aiheuttivat hämmennystä. Tunneilla ei myöskään voitu hyödyntää yhtä helposti opettajan valmiiksi tekemiä mallipohjia kuten ennen. Tunneilla tavaksi muodostunut kaverin ohjelmointisovelluksen testaaminen ei tullut kyseeseen projektien ollessa paikallisia. Aiemmin tuntien päätteeksi kaverin sovelluksen kokeileminen oli saattanut herättää intoa itse kokeilla jotain uutta seuraavalla kerralla. Nyt tämän kaltainen vuorovaikutus ja vertaisoppiminen jäi pois tai väheni. Scratchin online-version yhteisö on myös ollut vertaisoppimisen kannalta tärkeää. Paikallisessa sovelluksessa yhteisön hyödyntäminen jäi pois. Oppilaat ovat yleensä tottuneet hakemaan ideoita myös toisten tekemistä sovelluksista. Nyt se kaikki tieto – mitä muuta Robbo Lab laitteella voi tehdä – oli opettajan kerroksen tai oppilaan oman kokemuksen varassa. Innokas-yhteisön tekemästä pedagogisesta materiaalista olisi voinut saada myös ideoita, mutta oppilaille tutumpi tapa on kokeilla ja pelata ja oppia sitä kautta.

Ohjelmointitehtävien suunnittelussa tehtiin valmiin teemoitetun opetussuunnitelman pohjalta tuntikortit, joista saatiin tuntien jälkeen suullista palautetta sekä opettajien että oppilaiden kokemuksista. Lisäksi palautetta laitteen ja sovelluksen käytöstä kerättiin kirjallisella anonyymillä kyselyllä, jota analysoitiin kvalitatiivisesti. Opetustilanteet ja oppilaat ovat aina erilaisia, eikä tunnin aikana sattuneita tilanteita voi ottaa yleispätevinä. Kyselyyn saatiin yhdeksän vastausta. Palautteiden perusteella vastaanotto oli neutraali. Tuntien aikana ei huomattu poikkeuksellista innostusta, eikä myöskään luovuttamista laitteen suhteen. Vastausten perusteella parannettiin ohjeiden sisältöä sekä saatiin materiaalia opinnäytetyötä varten ja tehtiin lopulliset johtopäätökset opinnäytetyön toimeksiannon kysymykseen, soveltuuko Robbo Lab -laite toimeksiantajan toiminta-ajatukseen ja teemapohjaiseen harrastuspainotteiseen lasten ohjelmoinnin opettamiseen, jossa oppimisen tavoite on kehittää lapsen omaa oivallusta ja ongelmanratkaisukykyä.

Toiminnallisen osuuden tulosten perusteella voidaan todeta, että Robbo Lab -laite soveltuu lasten ohjelmoinnin opettamiseen mutta ei ehkä vastaa juuri Tiedekoulun tarpeisiin ja toiminta-ajatukseen. Pienistä vioista ja havaituista bugeista huolimatta, varsinaisesti laitteen käytössä ei ollut mitään ongelmaa. Laitteen elinkaari kuitenkin miellettiin testauksen aikana kovin lyhyeksi. Lienee perusteltua myös miettiä, tuoko laite kustannuksiltaan vastaavaa lisäarvoa harrastuspainotteiseen opetukseen. Sulautettujen laitteiden käyttö sen sijaan yleisesti, tuo opetukseen kokonaan uuden ulottuvuuden. Tähän Robbo Lab soveltuu turvallisenä ja koteloituna laitteena. Laitetta voi käyttää ilman erillistä ymmärrystä mekaniikasta ja elektroniikasta. Erona esim. Arduinon tai Lego Mindstorms, Robbo Lab ei vaadi rakentelua ennen ohjelmoinnin aloittamista. Samasta syystä laite kuitenkin miellettiin enemmänkin ohjaimeksi kuin robotiksi. Laitteeseen investoinnin kannalta olisi hyvä miettiä, onko hankintahinta kustannustehokas ja verrattuna muihin markkinoilla oleviin laitteisiin, kilpailukykyinen. Investoinnin kannalta oleellista on myös, onko valmistaja sitoutunut laitteen ja ohjelmiston kehitykseen.

Varsinaisen toiminnallisen osuuden sekä lopullisen kirjallisen opinnäytetyön viimeistelyn välillä aikaa kului lähes kaksi vuotta. Kirjallista osuutta sekä teoriaa tarkistelllessani havaitsin, että Robbon omilla sivuilla olevat materiaalit eivät olleet enää ladattavissa. En saanut myöskään yhteydenottoopyyntöihini vastausta. Havaitsin myös, että tässä välissä on ehtinyt myös opinnäytetyön kohteena oleva ohjelmisto muuttua.

Robbo Scratch 2.1 versio vuonna 2018 oli opetuskäyttöön toimiva, mutta vuoden 2020 versiota 3.0 ei ole kokonaan suomennettu, mikä voi vaikeuttaa eteenkin nuorempien oppilaiden tuntityöskentelyä ja sovelluksen käyttöä. Käytön vaikeus vähentää motivaatiota.

Myös ohjeiden päivitys sovelluspäivitysten vuoksi on aikaa vievää ja lisää kustannuksia, vaikka päivitykset olisivatkin toivottuja. Pitkäjänteisesti sitoutuvan laitetoimittajan kanssa yhteistyö on kannattavaa taloudellisesti. Mikäli valmistaja ei panosta kehitykseen, laitteen käyttöikä voi jäädä lyhyeksi. Näin myös laitteiden hankintakustannukset voivat kasvaa kohtuuttoman suuriksi.

## 5 POHDINTA

Opinnäytetyöni aiheena oli selvittää, soveltuuko lasten ohjelmoinnin opetukseen suunniteltu Robbo Lab -laite toimeksiantajan, joka oli myös työnantajani, toiminta-ajatukseen ja teemapohjaiseen harrastuspainotteiseen lasten ohjelmoinnin opettamiseen. Opinnäytetyön tavoite oli myös suunnitella ja mallintaa tarinallisia tunnissa suoritettavia ohjelmointitehtäviä toimeksiantajan koodikoulun jatkoryhmän tarpeisiin. Opinnäytetyössäni onnistuin selvittämään laitteen soveltuvuuden ja vastaamaan tutkimuskysymykseen. Opinnäytetyöni tuloksina syntyi myös toimeksiannon mukaiset oppimateriaalit, joiden viimeistellyt versiot ovat liiteosioissa.

Teorian hankinta osoittautui haasteelliseksi, sillä vaikka lasten ohjelmoinnin opetus onkin nykyään ihan arkipäivää, on aiheesta kirjoitettu kovin vähän teknisestä näkökulmasta, erityisesti harrastusnäkökulmasta. Opinnäytetyöni aihe oli tietotekninen mutta aiheen toinen osio, oppimateriaalin tuottaminen, meni hieman oman opintojeni pääaiheen sivusta. Pedagogiikkaa käsittelevän osuuden vuoksi myös kirjoitusprosessini oli pitkään jumissa ja välillä rönsyili laajastikin, hakiessani sopivaa teoreettista näkökulmaa tämän aiheen käsittelemiseen. Pedagogisissa osissa hyödynsin lopulta Innokas-verkoston Robbolle rakentamaa pedagogista materiaalia, soveltaen sitä toimeksiantajan pedagogiikkaan ja toiminta-ajatukseen ja rakentaen ohjelmointitehtävät tutun visuaalisen kehyksen ympärille. Tähän malliin päädyttyäni totesin ongelmani johtuneen siitä, etten ollut nähnyt itseäni aiemmin pedagogisena toimijana, vaikka sellaisena olin toiminutkin. Aihepiiri ja teoria oppimisesta ja pedagogiikasta tuntui kovin vieraalta, vaikka olin sitä jatkuvasti työssäni ja myös opinnoissani soveltanut.

Tutkimuskysymystä ei voitu lähestyä täysin objektiivisesti, koska toimeksiantaja oli samalla työnantajani, eli olin itse mukana tutkimuksessa aktiivisena ja osallistuvana toimijana, yhtenä opettajana. Olin jo ennen opinnäytetyöprosessia ehtinyt tutustumaan toimeksiantajalla käytössä oleviin muihin sulautetun järjestelmän laitteisiin ja käyttänyt niitä myös opetuksessa. Olin myös aiemmin valmistanut opetusmateriaalia sekä opettanut ohjelmointia toimeksiantajan pedagogista toiminta-ajatusta noudattaen. Kirjoitusprosessin aikana varsinaisen selvittelytyön loppuksi heräsi itselleni kysymyksiä siitä, voiko tuloksia tästä syystä pitää luotettavina? Oma kokemus opettajana sekä testiryhmän tuttuus saattoi vaikuttaa materiaalin rakentamiseen. Olisinko tehnyt erilaisia, haastavampia tehtäviä, jos en olisi omannut ennakkokäsityksiä ryhmän motivaatiosta ja osaamisesta?



Olisiko objektiivisena tutkijana pysyminen auttanut päätyämään toisenlaiseen, kehittävämpään lopputulokseen? Tutkijana, opettajana ja kollegana toimimisella saatoinkin itse olla vuorovaikuttamassa mielipiteisiin, asettamassa ennakkokäsityksiä ja johdattelemassa motivaatiota, vaikkakin tahattomasti ja tätä tiedostamatta.

Kirjoitusprosessin aikana huomasin myös, että laitteen soveltuvuuden dokumentointi oli jäänyt vajavaiseksi. Koin myös, että vertailu johonkin toiseen jo soveltuvaan laitteeseen, olisi helpottanut objektiivista soveltuvuuden tutkimista.

Opinnäytetyön tärkeimpinä tuloksina pidän itse prosessin aikana syntynyttä lähestymistapaa siitä, miten voidaan selvittää laitteen soveltuvuutta opetukseen. Laitteen tarkastelu niin teknisten ominaisuuksien kuin sen ohjelmoinnillisten puolienkin vuoksi on tärkeää. Myös laitteen testaus pilottiryhmällä voi olla kannattavaa ennen suuria laiteinvestointeja. Laitteen soveltuvuudessa tulisi huomioida myös laitteen elinkaaren tarkastelu. Jatkossa pitäisi ehkä myös pohtia objektiivisuuden merkitystä testausvaiheessa luotettavan aineiston aikaansaamiseksi.

Kehitysideana ehdottaisin, että laitteiden soveltuvuusprosessi kuvattaisiin jatkoa ajatellen tarkemmin sekä asetettaisiin valintakriteerit, jotka olisivat pisteytettyjä juuri toimeksiantajan toiminta-ajatukseen soveltuvina. Näin jatkossa valinta eri laitteiden välillä voisi helpottua ja pysyä objektiivisena tekijästä riippumatta. Oppimateriaalin saatavuus ja käytettävyys myös helpottuisi, jos se olisi suoraan käytettävissä verkosta.

Opinnäytetyöprosessini oli lopulta pitkä ja raskas, kestäen keväästä 2018 loppuvuoteen 2020 asti elämäntilanteiden muutosten ja ympäristössä tapahtuneiden haasteiden ja vastoinkäymisten vuoksi. Prosessina opinnäytetyön tekeminen sai minut pohtimaan omaa oppimisen prosessiani. Jälkikäteen on helppoa todeta, että jos alusta saakka olisin käyttänyt työkaluja, kuten Trelloa ja RefWorksia järjestelmällisesti, olisin saanut nopeammin tuloksia aikaan. Oman työn päiväkirjamainen dokumentointi olisi auttanut oppituntien ja testaamisen aikana tehtyjen havaintojen jäämistä myös kirjallisesti talteen. Myös vertaisoppiminen – eli opinnäytetyön tekeminen yhdessä tai edes samaan aikaan toisen opiskelijan kanssa, olisi voinut auttaa jäsentämään kirjallisia osuuksia nopeammin. Prosessin ollessa jo pitkällä sain onneksi riittävästi tukea ja ohjausta opinnäytetyöni loppuun saattamiseksi. Tästä haluaisinkin kiittää kaikkia mukanani kulkeneita.

Lukiessani pedagogista teoriaa Seymour Papertista, jonka olin tallentanut alkuvuodesta 2018, uudelleen vuonna 2020, koin ahaa-elämyksen ja uudelleen innostuksen

pedagogiikkaa ja lasten ohjelmoinnin opetusta kohtaan. Papertin tarinasta saamani motivaatio saattoi minut myös saamaan tämän opinnäytetyön valmiiksi.

Muistin myös, miten innostavaa on olla mukana jakamassa eteenpäin sitä onnistumisen kokemusta, jonka itse sain kokeillessani 1980-luvulla ala-asteen atk-kerhossa LOGO-ohjelmointia. Loppusanoiksi sopiikin tähän teoriaosuuden aikana löytämäni Papertin lause, jossa kuvataan, miten asioiden ajattelemisen saattaa saada aikaan ihan uusia ajatuksia.

"You can't think about thinking without thinking about thinking about something."

– Seymour Papert

## LÄHTEET

Blum, J. 2013. Exploring Arduino: Tools and Techniques for Engineering Wizardry. New York, UNITED STATES: John Wiley & Sons, Incorporated.

DUFVA, T. 2020. Digitaalisuuden ykköset ja nollat. Saatavilla: <https://www.oph.fi/fi/koulutus-ja-tutkinnot/digitaalisuuden-ykkoset-ja-nollat> [3.11. 2020].

Helsingin yliopisto 2020. Ohjelmoinnin MOOC 2020: ohjelmoinnin perusteet, Osa 1 .Saatavissa: <https://ohjelmointi-20.mooc.fi/osa-1/7-ohjelmointi-yhteiskunnassa/>. [2.11.2020]

Hjelm, E. 2016. Tytöt, pojat ja tietojenkäsittelyn opettaminen. Helsingin Yliopisto.

IT- kouluttajat ry, 2014. Koodiaapinen. Opettajan opas koodaukseen koulussa. Saatavissa: <https://koodiaapinen.fi/> [2.11.2020]

Konttinen, M. 2016. Koodaus puskee peruskoulun opetussuunnitelmaan – ministeriö turvautui yritysten apuun. Yle uutiset 13.5.2016. Saatavissa: [http://yle.fi/uutiset/koodaus\\_puskee\\_peruskoulun\\_opetussuunnitelmaan\\_\\_ministerio\\_turvautui\\_yritysten\\_apuun/7817538](http://yle.fi/uutiset/koodaus_puskee_peruskoulun_opetussuunnitelmaan__ministerio_turvautui_yritysten_apuun/7817538) [2.11.2020].

LIFELONG KINDERGARTEN GROUP MIT MEDIA LAB, 2020. Scratch. Saatavissa: <https://scratch.mit.edu/> [27.10. 2020].

Liukas, L & Mykkänen, J. 2014. Koodi2016, Ensiapua ohjelmoinnin opettamiseen peruskoulussa. [18.6.2018]

MAMAMEDIA, 2000. Professor Seymour Papert. Saatavissa: <http://papert.org/> [2.11.2020].

Matematiikka ja ohjelmoinnin jalo taito. 2014. Video. Partanen, Tiina. Saatavissa: <https://youtu.be/XwSqUAW6Gx4>. [18.6.2018].

MIT Media Lab, 2016. Professor Emeritus Seymour Papert, pioneer of constructionist learning, dies at 88. Saatavissa: <https://news.mit.edu/2016/seymour-papert-pioneer-of-constructionist-learning-dies-0801> [15, 2020].

Opetushallitus, 2017. Uudet opetussuunnitelmat pähkinänkuoressa. 2017. Saatavissa: [http://www.oph.fi/koulutus\\_ja\\_tutkinnot/perusopetus/opetussuunnitelma\\_ja\\_tuntijako/uudet\\_opetussuunnitelmat\\_pahkinankuoressa](http://www.oph.fi/koulutus_ja_tutkinnot/perusopetus/opetussuunnitelma_ja_tuntijako/uudet_opetussuunnitelmat_pahkinankuoressa). [18.6.2018]

Opetushallitus, 2014. Perusopetuksen opetussuunnitelman perusteet 2014. Saatavissa: [https://www.oph.fi/sites/default/files/documents/perusopetuksen\\_opetussuunnitelman\\_perusteet\\_2014.pdf](https://www.oph.fi/sites/default/files/documents/perusopetuksen_opetussuunnitelman_perusteet_2014.pdf). [3.11.2020].

Robbo, 2016. Robbo - Code Your Imagination. Saatavissa: <https://www.robbo.world/> [20.8.2018].

Robbo -kysely 2018. Tiina Virtanen.

SCRATCHERS, 2008. Scratch Wiki. Saatavilla: <https://en.scratch-wiki.info/> [2.11., 2020].

SIRÉN, J. 2017. Leluohjelmointikielten ykkönen Aloitetaan Scratchista. Skrolli Tietokonekulttuurin erikoislehti, sivu. 16. Saatavissa; <https://skrolli.fi/2017.3.suihku.pdf> [27.10.2020].

SORMUNEN, K., KORHONEN, T. LAVONEN, J. 2016. Development of the ROBBO pedagogical material– Design Based Research Approach. Innokas Network, University of Helsinki.

SUOMEN TIEDEKOULU OY, 2019. Tiedekoulu. Saatavissa: <http://www.tiedekoulu.fi/> [1.11.2020].

TECHTARGET, 2017. What is Scratch? - Definition from WhatIs.com. Saatavissa: <https://whatis.techtarget.com/definition/Scratch> [27.10. 2020].

WIKIMEDIA FOUNDATION, 2020a. Lego Mindstorms. Saatavissa: [https://fi.wikipedia.org/w/index.php?title=Lego\\_Mindstorms&oldid=19276830](https://fi.wikipedia.org/w/index.php?title=Lego_Mindstorms&oldid=19276830) [3.11.2020].

WIKIMEDIA FOUNDATION, 2020b. Sulautettu järjestelmä. Saatavissa: [https://fi.wikipedia.org/w/index.php?title=Sulautettu\\_j%C3%A4rjestelm%C3%A4&oldid=19203448](https://fi.wikipedia.org/w/index.php?title=Sulautettu_j%C3%A4rjestelm%C3%A4&oldid=19203448) [2.11. 2020].

WIKIMEDIA FOUNDATION, 2020c. Pedagogiikka Saatavissa: <https://fi.wikipedia.org/w/index.php?title=Pedagogiikka&oldid=19135742> [16.11.2020].

ÄNGESLEVÄ, M. 2019. Hyvä oppimateriaali syntyy oppijan saappaisiin astumalla, ja opettajilta se luonnistuu. Opettaja.fi. OAJ. Saatavissa: <https://www.opettaja.fi/tyossa/hyva-oppimateriaali-syntyy-oppijan-saappaisiin-astumalla-ja-opettajilta-se-luonnistuu/> [2.11.2020]

## LIITE 1. Tuntikortti 1

### Robbo Lab: 1. Oppitunti

# ROBBO LAB

**Tavoite:** Asentaa Robbo Scratch. Tutustua ohjelmointiympäristöön. Tutustua Robbo Lab Robotin perustoimintoihin. Ohjelmoida ensimmäinen oma Robbo Lab sovellus.

**Johdanto Teemaan:** Aluksi tutustutaan mikä on Robbo ja asennetaan ohjelmointiympäristö Robbo Scratch omalle koneelle. Tehdään oma ensimmäinen ohjelma.

**Ohjelmointiosuus:** Kerrataan ohjelmointikäskyt; perusohjelman rakenne, tapahtumat eli eventit, toistosilmukka, ehtolause, boolean-ehto, hahmon liikuttaminen. Opitaan uusi ohjelmointikäsky; laboratorion painikkeen toiminnan ohjelmointilohkot. Harjoitellaan myös tehtävän tallennusta.

#### Teeman 1. sisältö:

1.	Robbo Scratch Asennus s. 2
2.	Robbo Lab pakkauksen osat s. 2
3.	Robbo Lab kytkentä ja osat s. 2
4.	Yhteyden muodostus, ohjauspaneeli s.3
5.	Ohjelmointiympäristö Robbo Scratch s.3
6.	Robbo Lab osat s. 4
7.	Harjoitustehtävä ja malliratkaisu s. 5–6
8.	Tallennuksen tekeminen s. 7
9.	Haastetehtävä s.8

## Robbo Lab: 1. Oppitunti ROBBO LAB

1.

**Asennus:** Robbo Scratch on laajennus Scratchille, josta löytyy Robbo Lab –laitteen ohjaamiseen tarvittavat lohkot. Lataa sopiva asennuspaketti omalle koneellesi täältä: <https://www.robbo.world/apps>.

Asenna koneelle latautuva tiedosto, valitse suomen kieli, ja hyväksy asennus oletusasetuksilla, valitsemalla OK.

Asennuksen valmistuttua löydät sovelluksen käynnistyskuvakkeen työpöydältäsi.



2.

**Osat:** Robbo Lab sekä USB johto kytkentää varten.



3.

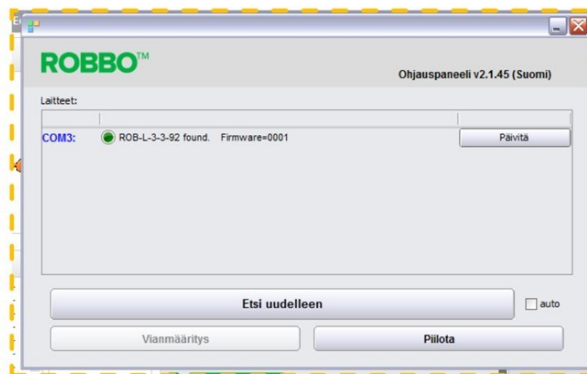
**Kytkeä:** Kiinnitä USB—kaapelin pienempi pää Robbo Lab laitteeseen ja toinen tietokoneeseen.



## Robbo Lab: 1. Oppitunti ROBBO LAB

4.

**Yhteyden muodostus:** Avaa Robbo Scratch -ohjelmisto työpöydän kuvakkeesta. Kun tietokone löytää Robbo Labin, kuuluu laitteesta äänimerkki. Odota, kunnes Ohjauspaneelin ympyrä muuttuu vihreäksi merkinä yhteyden muodostumisesta. Jos ympyrä ei muutu vihreäksi paina *Päivitä laiteohjelmisto* -painikkeesta, niin laiteohjelmisto päivittyy. *Piilota* -painikkeesta, saat ohjauspaneelin katoamaan. Ohjauspaneelin on oltava käynnissä, jotta laitteen yhteys koneeseesi säilyy.

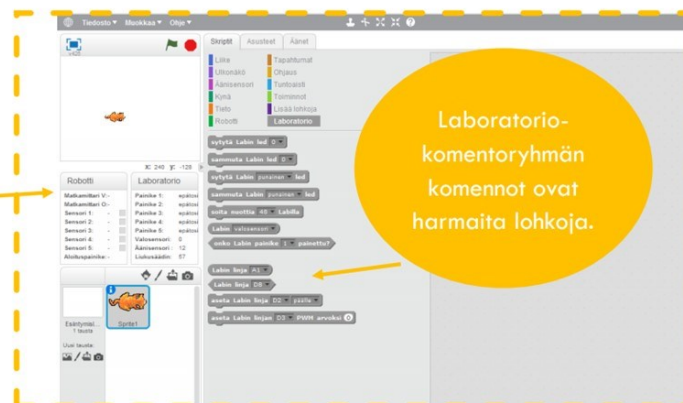


5.

**Robbo Scratch:** Ohjelmointiympäristö näyttää Scratchilta ja toimii samalla tavalla. Robbo Scratchiin on lisätty Laboratorio—komentoryhmä sekä sensorien ja painikkeiden arvonäkymä.

Koska ohjelma on asennettuna koneellesi, myös harjoitukset tallennetaan omalle koneelle.

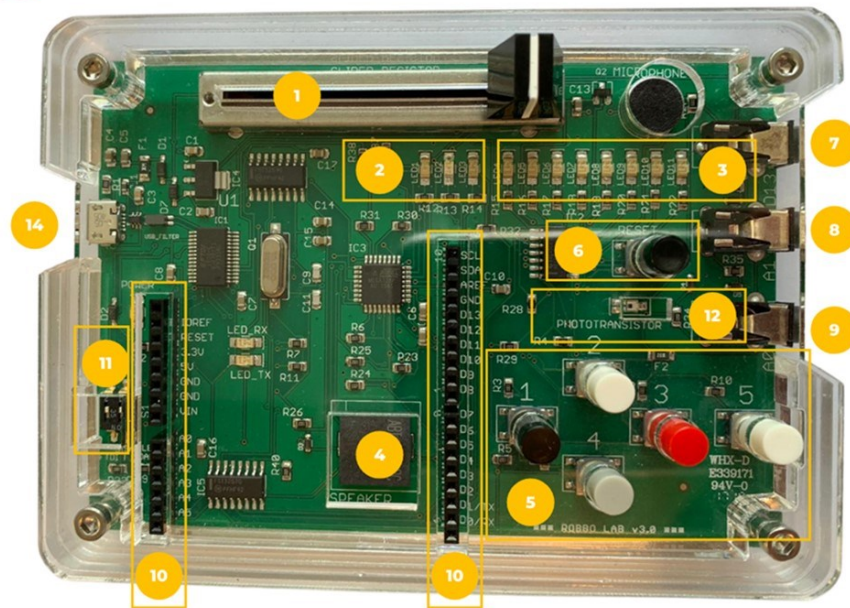
Sensorien ja painikkeiden arvat näkyvät täällä



## Robbo Lab: 1. Oppitunti ROBBO LAB

6.

## Robbo Lab osat:



1.	Liukusäädin, potentiometri
2.	Kolme värilediä, punainen, keltainen, sekä vihreä
3.	Ohjattu kahdeksan ledin matriisi
4.	Kaiutin
5.	5 kpl painikkeita
6.	Reset -painike
7.	D13 -liitin, digitaalisen datan lukemista ja kirjoittamista varten (0/1)
8.	A1 -liitin, analogisen datan lukemista varten (0–100). Liitintä voi käyttää myös D13 -liittimenä.
9.	A0 -liitin, joka toimii kuten kohdan 8. A1 -liitin.
10.	Arduino Uno -yhteensopiva liitinlinja.
11.	Dip -kytkimet, joiden asentoa muuttamalla saadaan robotin toimintoja muutettua. Vaihtoehdolla 1-On = Osa komponenteista (1,3,4,12,13) ovat irti kytkettyjä, jolloin Robbo Labia voidaan käyttää kuten Arduino Unoa. Painikkeet ja Liittimet (A0, A1, D13) ovat toiminnassa. Kaikki linjan 10 liittimet ovat toiminnassa. 1-OFF = Kaikki Robbo Labin omat komponentit ovat käytössä. Osa linjan 10 liittimistä ovat liitettyinä Labin omiin komponentteihin. 2-ON = sarjaviestintä siirtyy liitinlinjalle 10 (D0/RX, D1, TX). USB-liitin (FTDI piiri) on pois käytöstä. Ohjelmoitaessa kytkimen pitäisi olla tässä asennossa.
12.	Valosensori (fototransistori).
13.	Äänisensori (mikrofoni).
14.	USB-liitin



## 7.

## Robbo Lab: 1. Oppitunti ROBBO LAB

**Tehtävä 1.**

Tarkoitus on tutustua Robbo Labin painikkeisiin.

Tee peli, jossa pelaaja liikuttaa hahmoa Robbo Labin painikkeilla 1,2,3 ja 4.

Valitse haluamasi hahmo ja tausta.

**Muista tallentaa työsi!**

Tarvitset näitä ohjelmalohkoja :



- Tapahtuman eli Eventin käynnistäjä



- Toistosilmukka



- Ehtolause "jos, sitten"



- Robbo Lab painikkeen painamista seuraava lohko (boolean - tosi / epätosi)



- Robbo Lab painikkeiden arvot (boolean - tosi / epätosi)

- Lisää ehtolauseen jälkeen jokin haluamasi liike



7.

## Robbo Lab: 1. Oppitunti ROBBO LAB

## Tehtävä 1. - Malliratkaisu



Scratch script for the Robbo Lab robot:

```

kun klikataan
    ikuisesti
        jos onko Labin painike 1 painettu? , sitten
            osoita suuntaan -90
            liiku 10 askelta
        jos onko Labin painike 2 painettu? , sitten
            osoita suuntaan 0
            liiku 10 askelta
        jos onko Labin painike 3 painettu? , sitten
            osoita suuntaan 90
            liiku 10 askelta
        jos onko Labin painike 4 painettu? , sitten
            osoita suuntaan 180
            liiku 10 askelta
    
```

Tarkkaile painikkeiden arvojen muutosta Robbo Scratch—ohjelmassa. Kun painiketta painetaan, painikkeen arvoksi tulee tosi. Kun painiketta ei paineta, arvo on epätosi

Robbo Scratch interface showing the robot's status and sensor/button values:

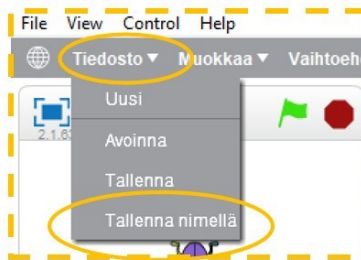
Robotti	Laboratorio
Matkamittari V:--	Painike 1: epätosi
Matkamittari O:--	Painike 2: tosi
Sensori 1: --	Painike 3: epätosi
Sensori 2: --	Painike 4: epätosi
Sensori 3: --	Painike 5: epätosi
Sensori 4: --	Valosensori: 0
Sensori 5: --	Äänisensori: 9
Aloituspainike: epätosi	Liukusäädin: 0

8.

## Robbo Lab: 1. Oppitunti ROBBO LAB

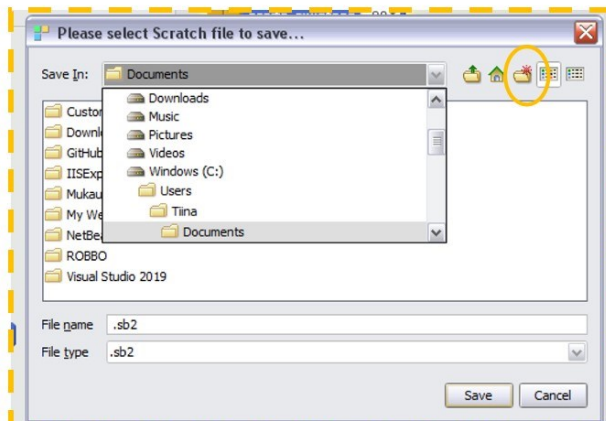
## TEHTÄVÄN TALLENNUS:

**Valitse** Tiedosto—Tallenna nimellä



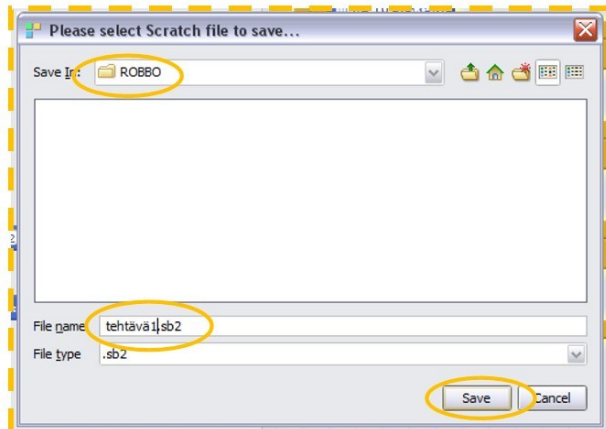
**Valitse** Tallennuskansio (Save In:)

Klikkaa hiirellä kansionnimekenttään, valitse haluamasi tallennuskansio, tai tee uusi kansio valitsemalla Uusi— kuvake. Nimeä uusi kansio esim. ROBBO. Ja valitse kansio.



**Nimeä** tehtäväsi (File name) ja paina Tallenna (Save).

Robbo Scratch—projektitiedoston päätte on sb2



## Robbo Lab: 1. Oppitunti ROBBO LAB

9.

### Haastetehtävä:

Tee peli, jossa pelaaja liikuttaa hahmoa painikkeilla.

Lisää vastustaja, joka liikkuu satunnaisesti pelialueella.

Lisää pelaajalle toiminto painikkeelle 5. , jolla saat vihollisen kukistettua.

Lisää Game Over—taustan vaihto.

Muista tallentaa työsi!



## LIITE 2. Tuntikortti 2

### Robbo Lab: 2. Oppitunti

# ROBBO LAB äännet ja ledit

**Tavoite:** Tutustua Robbo Scratch ohjelmointiympäristöön. Tutustua Robbo Lab Robotin ääniin ja ledeihin. Ohjelmoidaan oma Robbo Lab sovellus.

**Johdanto Teemaan:** Robbo Labissa on kolme värilediä, punainen, keltainen, sekä vihreä. Lisäksi Labissa on kahdeksan ledin matriisi. Labissa on myös äänisensori (mikrofooni) sekä kaiutin.

**Ohjelmointiosuus:** Kerrataan ohjelmointikäskyt; Eventit, odota – lohkon käyttö, Lisää lohkoja (omat lohkot) ja funktio. Opitaan uudet ohjelmointikäskyt; laboratorion ledien ohjelmointilohkojen käyttö, laboratorion nuottien ohjelmointilohkojen käyttö.

#### Teeman 2. sisältö:

- |    |  |
|----|--|
| 1. | Robbo Lab kytkentä s. 2                          |
| 2. | Ohjelmointiympäristö Robbo Scratch s.2           |
| 3. | Paikannetaan kaiutin, äänisensori sekä ledit s.2 |
| 4. | Äänen ohjelmointilohkot s. 3                     |
| 5. | Ledien ohjelmointilohkot s. 3                    |
| 6. | Harjoitustehtävä ja malliratkaisu s. 4–6         |
| 7. | Haastetehtävät s.7                               |

## Robbo Lab: 2. Oppitunti

# ROBBO LAB äänet ja ledit

**Tavoite:** Tutustua Robbo Scratch ohjelmointiympäristöön. Tutustua Robbo Labin ääniin ja ledeihin. Ohjelmoidaan oma Robbo Lab sovellus.

**Johdanto Teemaan:** Robbo Labissa on kolme värilediä, punainen, keltainen, sekä vihreä. Lisäksi Labissa on kahdeksan ledin matriisi. Labissa on myös äänisensori (mikrofoni) sekä kaiutin.

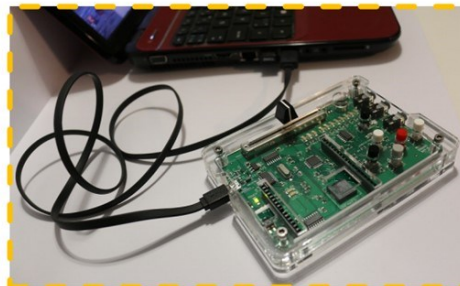
**Ohjelmointiosuus:** Kerrataan ohjelmointikäskyt; tapahtumat eli eventit, odota – lohkon käyttö, Lisää lohkoja sekä funktio. Opitaan uudet ohjelmointikäskyt; laboratorion ledien ohjelmointilohkojen käyttö, laboratorion nuottien ohjelmointilohkojen käyttö.

### Teeman 2. sisältö:

1. Robbo Lab kytkentä s. 2
2. Ohjelmointiympäristö Robbo Scratch s.2
3. Paikannetaan kaiutin, äänisensori sekä ledit s.2
4. Äänen ohjelmointilohkot s. 3
5. Ledien ohjelmointilohkot s. 3
6. Harjoitustehtävä ja malliratkaisu s. 4–6
7. Haastetehtävät s.7

## Robbo Lab: 2. Oppitunti ROBBO LAB äänet ja ledit

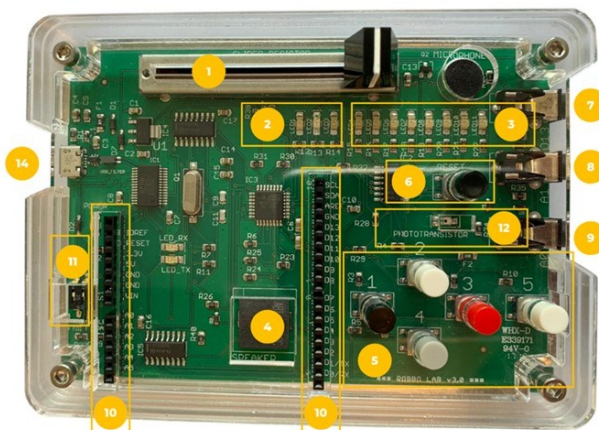
1. **Kytkeä:** Kiinnitä USB—kaapelin pienempi pää Robbo Lab laitteeseesi ja toinen tietokoneeseesi.



2. **Ohjelma:** Ohjelmoinnissa käytetään omalle koneellesi asennettua Robbo Scratch—sovellusta.



3. **Paikannetaan Robbo Labin kaiutin, äänisensori sekä ledit**



2.	Kolme värilediä, punainen, keltainen, sekä vihreä
3.	Ohjattu kahdeksan ledin matriisi
4.	Kaiutin
13.	Äänisensori (mikrofoni).

## Robbo Lab: 2. Oppitunti ROBBO LAB äänet ja ledit

4.

### Äänen ohjelmointilohko:



Valitsemalla nuolesta saat näkymään eri nuotit kirjaimina ja numerisina, ja niiden aseman pianon koskettimilla.



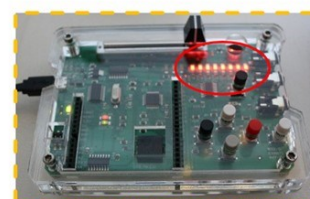
5.

### Ledien ohjelmointilohkot:

**VäriLEDit:** vihreä, keltainen ja punainen väriLEDi syttyvät ja sammuvat ohjelmointilohkoilla:



**Matriisiledit:** 8 ledin matriisiledit syttyvät ja sammuvat ohjelmointilohkoilla:





## Robbo Lab: 2. Oppitunti ROBBO LAB äänet ja ledit

6.

### Tehtävä 2.

Tämän tehtävän aiheena on tutustua äänien tuottamiseen Robbo Labilla sekä ledien toimintaan.

Tee musiikki- sovellus, jossa hahmoa painettaessa kuuluu jokin tuntemasi lyhyt laulu käyttäen Labin ääniominaisuuksia (ohje 4) . Voit myös keksiä sävelmän itse.

Lisää ledit vilkkumaan musiikin tahtiin (ohje 5).

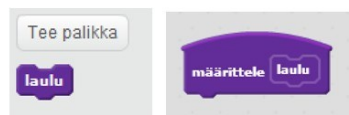
Upota tekemäsi laulu funktion sisälle, käyttämällä Lisää lohkoja— toiminnallisuutta.

Lisää halutessasi Robbo Scratchin taustalle myös animaatiota. Muista tallentaa työsi!

Tarvitset näitä ohjelmalohkoja :



- Tapahtuman eli Eventin käynnistäjä



- Lisää lohkoja, tässä funktion nimi on "laulu"



- odota —lohko



- Led ja ääni—lohkot

6.

## Robbo Lab: 2. Oppitunti ROBBO LAB äänet ja ledit

## Tehtävä 2.

## Ratkaisumalli: ÄÄNET

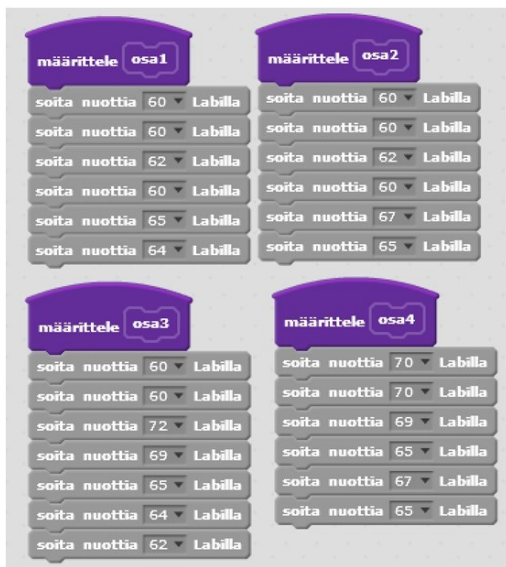
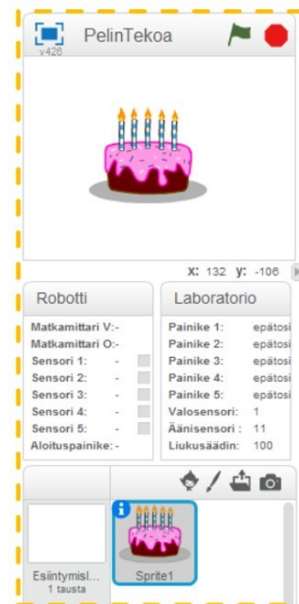
Tehdään onnittelulaulu, jossa Hahmoksi valittua kakkua painamalla, alkaa soimaan ”Paljon onnea vaan”.



Mallissa on käytetty **Lisää lohkoja**— tehden onnittelulauluun 4 eri osaa, eli funktiota.

Funktioiden nimet ovat tässä: osa1, osa2, osa3, sekä osa4.

Funktioiden määrittelyosien sisälle on laitettu onnittelulaulun osien nuotit.



Varsinainen sovellus käynnistetään Vihreän lipun painamisen jälkeen tapahtumalla (eli eventillä) ”Kun tätä hahmoa klikataan”.

Eri osien funktioiden kutsujen välillä on käytetty odota—lohkoja.

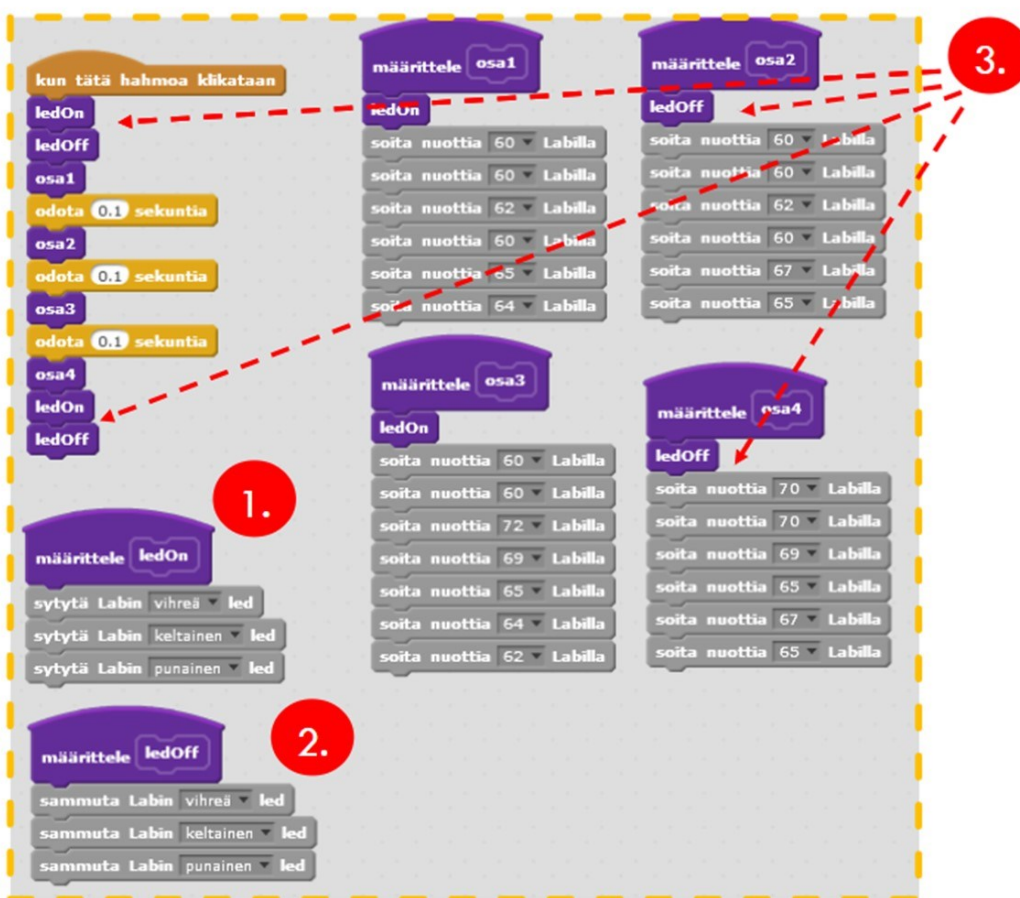


## 6. Robbo Lab: 2. Oppitunti ROBBO LAB äänet ja ledit

### Tehtävä 2.

#### Ratkaisumalli: LEDIT

Lisätään funktiot ledOn (1.) sekä Led Off (2.). Määritellään funktioiden sisälle toiminnoksi väriledien sytytys ja sammutuslohkot. Kutsutaan funktioita sekä halutuissa nuotti—osoiten koodissa (3.) että varsinaisessa pääeventissä.



## Robbo Lab: 2. Oppitunti ROBBO LAB äänet ja ledit

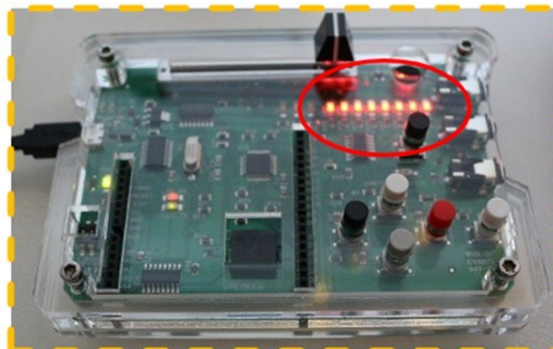
9.

### Haastetehtävä:

Tee Jukebox- sovellus jossa jokaista Labin painiketta painettaessa alkaa soimaan jokin kappale.

Laita matriisiledit liikkumaan musiikin tahdissa.

Muista tallentaa työsi!



### Haastetehtävä:

Tee Piano- sovellus.

Tee Robbo Scratchiin painikkeet joita painamalla soi nuotti kuten pianossa.

Laita matriisiledit syttymään ja sammumaan nuottien tahdissa.

Muista tallentaa työsi!

## LIITE 3. Tuntikortti 3

# Robbo Lab: 3. Oppitunti

## ROBBO LAB liikusäädin

**Tavoite:** Tutustua Robbo Labin Liikusäätimen toimintaan. Ohjelmoidaan oma Robbo Lab sovellus hyödyntäen liikusäädintä.

**Johdanto Teemaan:** Robbo Labin liikusäädin eli liukupotentiometri on vasemmassa reunassa ollessaan arvossa 0. Arvo kasvaa liukuvasti liikuttaessa säädintä oikealle. Oikeassa reunassa ollessaan, säädin on arvossa 100.

**Ohjelmointiosuus:** Kerrataan ohjelmointikäskyt; tapahtumat eli eventit, toistosilmukka, ehtolause, boolean-ehto, hahmon liikuttaminen, laboratorion painikkeen toiminnan ohjelmointilohkot, muuttujat, pomppaa, vertailulausekkeet. Opitaan uusi ohjelmointikäsky; laboratorion liikusäätimen toiminnan ohjelmointilohko. Harjoitellaan myös aiemmin tallennetun tehtävän avaamista.

### Teeman 3. sisältö:

1. Robbo Lab kytkentä s. 2
2. Ohjelmointiympäristö Robbo Scratch s. 2
3. Paikannetaan liikusäädin s. 2
4. Harjoitustehtävä ja malliratkaisu s. 3-5
5. Tallennetun tehtävän avaaminen s. 6
6. Haastetehtävä s. 7

## Robbo Lab: 3. Oppitunti ROBBO LAB liikusäädin

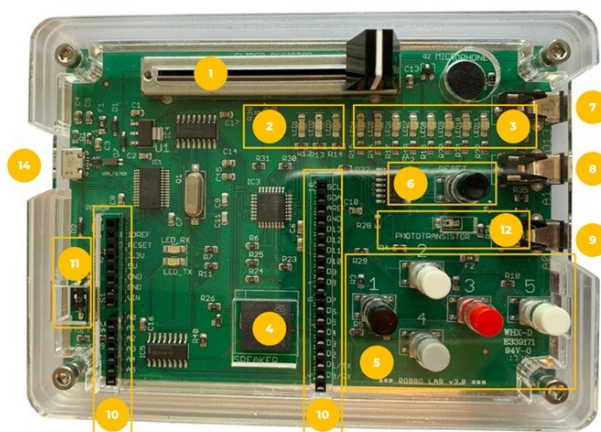
1. **Kytkeä:** Kiinnitä USB -kaapelin pienempi pää Robbo Lab laitteeseesi ja toinen tietokoneeseesi.



2. **Ohjelma:** Ohjelmoinnissa käytetään omalle koneellesi asennettua Robbo Scratch -sovellusta.



3. **Paikannetaan Robbo Labin liikusäädin**



1.	liikusäädin, potentiometri
----	----------------------------

## Robbo Lab: 3. Oppitunti ROBBO LAB liikusäädin

4.

### Tehtävä 3.

Tämän tehtävän aiheena on tutustua Robbo Labin liikusäätimen toimintaan.

Tee peli, jossa hahmoa liikutetaan painamalla Labin painikkeita. Voit halutessasi jatkaa pelin tekoa suoraan käyttämällä tehtävän 1. ratkaisua pohjana.

Lisää hahmolle muuttuja nopeus—jonka arvo muuttuu liikuttamalla liikusäädintä. Muuta hahmon liikenopeutta muuttujan arvon mukaan.

Älä päästä hahmoa liikkumaan esiintymislavan reunojen ulkopuolelle.

Muista tallentaa työsi!

Tarvitset näitä ohjelmalohkoja :



- Tapahtuman eli eventin käynnistäjä



- Toistosilmukka



- Ehtolause "jos, sitten"



- Robbo Lab painikkeen painamista seuraava lohko (boolean - tosi / epätosi)



- Robbo Lab painikkeiden arvot (boolean - tosi / epätosi)

Jatkuu



## Robbo Lab: 3. Oppitunti ROBBO LAB liikusäädin

4.

Jatkuu: Tarvitset näitä ohjelmalohkoja :



- Lisää ehtolauseen  
jälkeen jokin haluamasi liike

**nopeus** - Muuttuja nopeus.

**pomppaa reunasta** - Pomppaa reunasta

**ei** - Vertailulausekkeet

**Labin liikusäädin** - Liikusäätimen lohko

**Laboratorio**

Painike 1:	epätosi
Painike 2:	epätosi
Painike 3:	epätosi
Painike 4:	epätosi
Painike 5:	epätosi
Valosensori:	1
Äänisensori:	9
Liikusäädin:	66

- Robbo Lab liikusäätimen arvo (1-100)



4.

## Robbo Lab: 3. Oppitunti ROBBO LAB liikusäädin

## Tehtävä 3.

## Ratkaisumalli: liikusäädin

Tehdään hahmo, joka liikuu Labin painikkeilla. Lisätään muuttuja nopeus, joka saa arvonsa kun liikusäädintä liikutetaan. Vertailu tehdään käyttämällä "Ei" "On yhtä suuri kuin" nopeus, jolloin saadaan verrattua muuttuuko nopeuden arvo. Muuttujan nopeus arvo tehdään muuttamalla liikusäätimen arvo / 10, jottei hahmo liikkuisi liian nopeasti ruudulla.

Samaa pitää käyttää myös vertailussa.

Huomaa pelin alussa nollata nopeus muuttuja!

Pomppaa reunassa - tarvitaan vain kerran ikuisesti silmukan si-

Ohjelmassa voi olla kaksi rinnakkaista tapahtumaa. Muista Ikuisesti—silmukka. Vertailu "Ei" "On yhtä suuri kuin" on sama kuin "eri suuri kuin". Sellainen lohko puuttuu.

## 5.

## Robbo Lab: 3. Oppitunti ROBBO LAB liikusäädin

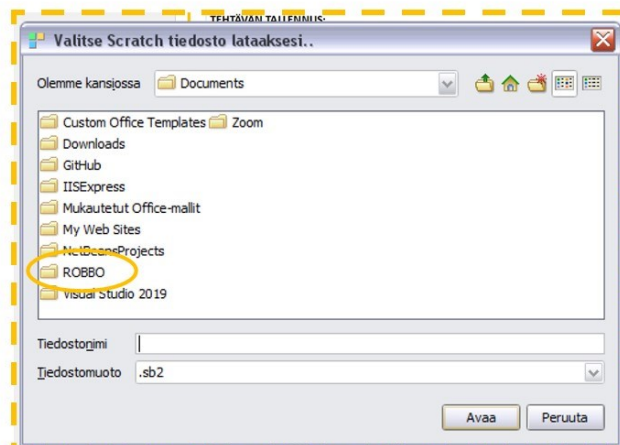
## Tallennetun tehtävän avaaminen :

**Valitse** Tiedosto—Avoinna



**Valitse** hakemisto, jonne olet luonut aiemmin tehdyn tehtäväkansion (Olemme kansiossa:)

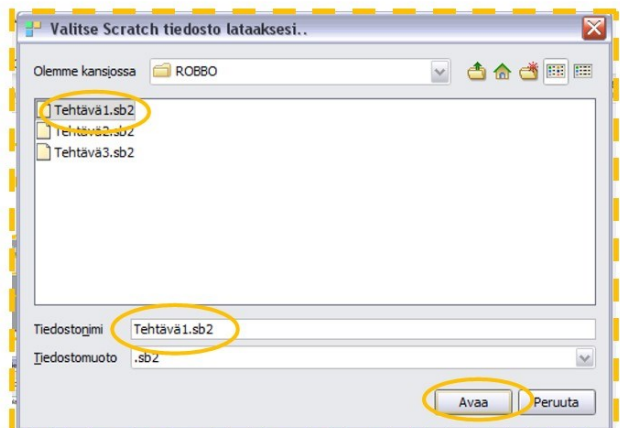
Tai **klikkaa** hiirellä näkyvässä näkyvää kansiota jossa tehtäväsi on ja valitse **Avaa**. Saat näkyviin vain .sb2—nimiset tiedostot.



**Klikkaa** haluamasi tehtävä ja valitse **Avaa**.

Robbo Scratch—  
projektitiedoston päätte on sb2

HUOM! Jos teet vanhan harjoituksen päälle uutta, muista tallentaa ja valita **Tallenna Nimellä** ennen muutosten tekemistä!



### Robbo Lab: 3. Oppitunti ROBBO LAB liikusäädin

6.

#### Haastetehtävä:

Lisää tehtävään 3 hahmolle asustenvaihdot.

Lisää tehtävään Led—matriisi, jonka valot toimivat liikusäätimen lisäefektinä.

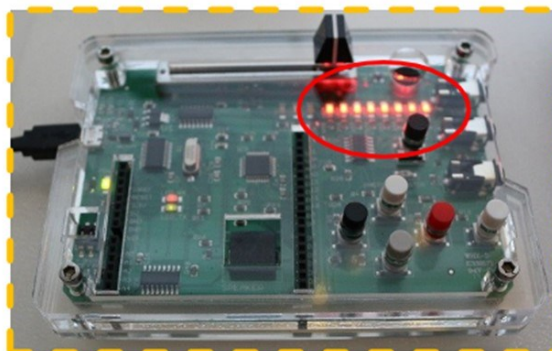
Lisää tehtävään vihollinen, joka liikkuu satunnaisella nopeudella kohti pelaajan hahmoa.

Lisää tehtävään aika (esim. 40 sek.) jonka viimeisellä 5 sekunnilla kuuluu äänimerkki pelin loppumisen merkiksi.

Jos pelaaja ehtii välttelemään vihollista, on hän voittanut. Jos vihollinen ehtii koskettaa pelaajaa ennen ajan loppumista, pelaaja häviää.

Lisää Game Over ja Voitit - taustat.

Muista tallentaa työsi!



# LIITE 4. ROBBO LAB – kysely 2018, kysymykset

LIITE 4. Robbo Lab kysely. 2018 – kysymykset

Sivu 1 / 2

## ROBBO LAB kysely

Lasten Tiedekoulun keväällä 2018 toteutettu Robbo Lab -laitteen sekä ohjeiden palautekysely. Vastaukset käsitellään nimettöminä ja luottamuksellisesti. Vastauksia käytetään materiaalin parantamiseen, sekä opettajan opinnäytetyön aineistona.

Oletko opettaja vai oppilas? \*

- Opettaja  
 Oppilas

Jos olet oppilas, kerro ikäsi sekä montako vuotta olet ohjelmoinut koodikoulussa.

[Lyhyt vastausteksti](#)

Jos olet oppilas, käytitkö Robbo Labia...

- Yksin  
 Parin kanssa

Monellako tunnilla Robbo Lab oli käytössäsi? \*

[Lyhyt vastausteksti](#)

Oliko Tiedekoulun (ei Innokas-materiaali) Robbo Lab PDF-ohjeistus mielestäsi... \*

- Liian vähän ohjeita  
 Liian pitkiä ohjeita  
 En saanut ohjeita  
 Hyvät ohjeet  
 Muu...

Valitse vaihtoehdot mitä kokeilit Robbo Labin kanssa (lisää muu- jos teit jotain omaa) \*

- Robbo Scratch asennus  
 Painikkeilla liikkuminen  
 Liukusäädin  
 Äänisensori  
 Valosensorit  
 Muu...

## LIITE 4. Robbo Lab kysely. 2018 – kysymykset

Sivu 2 / 2

Oliko Robbo Lab mielestäsi hyödyllinen kurssilla? \*

- Kyllä
- Ei
- Olisin halunut kokeilla sitä lisää
- Muu...

Oliko Robbo Labin käyttö mielestäsi... \*

- Helppoa
- Vaikeaa

Haluatko jatkaa Robbo Labin kanssa työskentelyä? \*

- Kyllä
- En
- Ehkä
- Muu...

Jos teit omia projekteja Robbolla, minkälaisia teit?

Pitkä vastausteksti

Kenelle Robbo Lab mielestäsi sopii? \*

- Vasta-alkajille (koodikoulu 1k1)
- Kokeneemmille koodareille (koodikoulu 2->)
- Ei sovellu tiedekoulun käyttöön
- Muu...

Perustelu edelliseen kysymykseen, kenelle Robbo Lab mielestäsi sopii? \*

Pitkä vastausteksti

Kommentteja tai mielipiteitä Robbo Scratchin käytöstä

Pitkä vastausteksti

Kommentteja tai mielipiteitä Robbo Labin käytöstä

Pitkä vastausteksti

Kommentteja tai mielipiteitä PDF ohjeista

Pitkä vastausteksti