

VAASAN YLIOPISTO
TEKNIIKAN JA INNOVAATIOJOHTAMISEN YKSIKKÖ
TIETOTEKNIikka

Janne Puntila

SUUNNITELMA GRAAFISESTA OHJELMOINTIYMPÄRISTÖSTÄ
PERUSOPETUKSEEN KÄYTETTÄVYYDEN EHDOKSILLA:
KOODIKOIRA

Tietotekniikan
pro gradu –tutkielma

Teknisen viestinnän maasterikoulutusohjelma

VAASA 2018

SISÄLLYSLUETTELO	sivu
1 JOHDANTO	5
1.1 Tutkimuksen tavoite ja rajaus	7
1.2 Tutkimusstrategia ja -menetelmät	8
1.3 Tutkielman rakenne	10
2 KÄYTETTÄVYYS, KÄYTTÄJÄKOKEMUS JA GRAAFISET OHJELMOINTIYMPÄRISTÖT	11
2.1 Käytettävyys	11
2.1.1 Käytettävyys ohjelmistojen näkökulmasta	13
2.1.2 Käytettävyys lasten näkökulmasta	15
2.2 Käyttäjäkokemus	17
2.2.1 Käyttäjäkokemus lasten näkökulmasta	18
2.2.2 Pelillistäminen	19
2.3 Graafiset ohjelmistoympäristöt	20
3 MÄÄRÄLLINEN TUTKIMUS JA SUORITETTU KYSELYTUTKIMUS	22
3.1 Määrällinen tutkimus	22
3.2 Suoritettu kyselytutkimus	23
3.3 Kyselytutkimuksen tulosten läpikäynti	25
4 OHJELMAN SUUNNITTELU JA PROTOTYYPPI	33
4.1 Ohjelman suunnittelu	33
4.1.1 Ohjelman nimi	33
4.1.2 Ohjelman värit	34
4.1.3 Minimalistinen suunnittelu & standardit	35
4.1.4 Muistamisen sijaan tunnistus & käytön tehokkuus	36
4.1.5 Tosielämän vastaavuus	36
4.1.6 Johdonmukaisuus	37
4.1.7 Käyttäjän vapaus	38

4.1.8	Järjestelmän näkyvä tila	38
4.1.9	Virheiden korjaaminen ja estäminen	39
4.2	Prototyyppi	39
5	KÄYTETTÄVYYSTESTAUS	40
5.1	Käytettävyystestaus teoriassa	40
5.2	Käytettävyystestauksen suunnittelu	42
5.3	Käytettävyystestauksen läpivienti	43
5.4	Käytettävyystestauksen tulosten analysointi ja esittämien	45
5.5	Käytettävyystestaus lasten näkökulmasta	47
5.6	Ääneenajattelutekniikka ja paritestausta	49
6	KÄYTETTÄVYYSTESTIN SUUNNITTELU, TOTEUTUS SEKÄ TULOKSET	51
6.1	Käytettävyystestin suunnittelu	51
6.1.1	Testitehtävät	52
6.1.2	Testihenkilöt	53
6.1.3	Testiympäristö	53
6.2	Käytettävyystestin läpikäynti	54
6.2.1	Alkutietokysely	54
6.2.2	Tehtäväkohtainen läpikäynti	55
6.2.3	Loppuhaastattelu	61
6.3	Käytettävyystestin tulosten analysointi	64
7	JOHTOPÄÄTÖKSET	69
	LÄHDELUETTELO	72
	LIITTEET	83
LIITE 1.	Kyselylomake	83
LIITE 2.	Kyselylomakkeen tulokset	84
LIITE 3.	Testihenkilöiden huoltajille lähetetty tiedote ja lupalomake	90
LIITE 4.	Käytettävyystestauksen testihenkilöt	92

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen yksikkö**

Tekijä:	Janne Puntila	
Tutkielman nimi:	Suunnitelma graafisesta ohjelmointiympäristöstä perusopetukseen käytettävyyden ehdoilla: Koodikoira	
Ohjaajan nimi:	Juho-Pekka Mäkipää	
Tutkinto:	Kauppätieteiden maisteri	
Ohjelma:	Teknisen viestinnän maisterikoulutusohjelma	
Pääaine:	Tietotekniikka	
Opintojen aloitusvuosi:	2012	
Tutkielman valmistumisvuosi:	2018	Sivumäärä: 93

TIIVISTELMÄ:

Suomalainen koulujärjestelmä on yksi maailman parhaista. Tapahtuva teknologinen kehitys kuitenkin edellyttää myös koulujärjestelmäämme muuttumaan jatkuvasti ajan-tasaisemmaksi. Ohjelmointi on merkittävä tulevaisuuden ala, johon valmistautuminen tulee aloittaa jo aikaisessa vaiheessa; jokaiselle lapselle tulisikin tarjota perusopetuksessa tasapuoliset mahdollisuudet innostua ohjelmoinnista. Tähän on pyritty tuomalla ohjelmoinnin opetus osaksi opetussuunnitelmaa, mutta tällä hetkellä opetuksen apuna käytettävissä olevat graafiset ohjelmointiympäristöt eivät ole käytettävyydeltään toivotulla tasolla. Käytettävyyso ongelmia aiheuttavat esimerkiksi ulkomaalaisesta alkuperästä johtuvat käänös vaikeudet. Tämän tutkielman tavoitteena on luoda kattava ja toteutettavissa oleva suunnitelma ensiluokkaisen käytettävyyden omaavasta graafisesta ohjelmointiympäristöstä ohjelmoinnin perusopetukseen.

Tutkielman teoreettinen viitekehys koostuu ennen kaikkea käytettävyyden ja käyttäjäkokemuksen käsitteistä. Näitä käsitteitä tarkastellaan erityisesti lasten näkökulmasta. Lisäksi viitekehyksessä käydään läpi graafiset ohjelmointiympäristöt ja pelillistäminen sekä määrällinen tutkimus ja käytettävyydestaus. Tutkielmassa suoritetaan suunnittelu-tutkimus, jossa tutkimusmenetelminä käytetään observointia, kyselytutkimusta sekä käytettävyydestausta. Käytettävyydestauksessa käytetään apuna äänenajattelutekniikkaa.

Tutkimuksessa suoritettujen suunnittelututkimuksen avulla luotiin suunnitelma lapsille suunnatusta graafisesta ohjelmointiympäristöstä kiinnittäen huomiota erityisesti käytettävyyteen ja käyttäjäkokemukseen. Koodikoiraksi nimetyn ohjelman hi-fi prototyypille suoritettua käytettävyydestaustaessa löytyi verrattain useita käytettävyyso ongelmia, mikä on normaalia testattaessa kehitysvaiheessa olevalla järjestelmällä. Ongelmista kuitenkin vain yksi oli niin katastrofaalinen, ettei järjestelmää voisi julkaista ennen sen korjaamista. Lisäksi tutkielmassa koottiin yhteen laajalti kirjallisuuden ohjeita sekä hyvän käytettävyyden että lapsille suunnatun järjestelmän suunnitteluun; kootut ohjeet myös todistettiin toimiviksi.

AVAINSANAT: käyttäjäkokemus, käytettävyys, ohjelmistosuunnittelu, graafinen ohjelmointiympäristö, lapset

UNIVERSITY OF VAASA**Technology and Innovations****Author:**

Janne Puntila

Topic of the Master's Thesis:

A plan of a visual programming language for primary education on usability's terms: Koodikoira

Instructor:

Juho-Pekka Mäkipää

Degree:

Master of Science in Economics and Business Administration

Major:

Computer Science

Degree Programme:

Technical Communication

Year of Entering the University:

2012

Year of Completing the Master's Thesis: 2018**Pages:** 93

ABSTRACT:

The Finnish schooling system is one of the best in the world. However, the technological development also requires our school system to constantly change to be more up-to-date. Programming is a significant line in the future, for which the preparation should be started in an early stage; every child should have an equal opportunity to get excited about programming during their primary education. This has been worked towards by introducing programming to the curriculum, but currently the usability of the visual programming languages used as teaching aides are not on a desired level. Usability problems are caused, for example, by translation difficulties due to their foreign origin. The purpose of this thesis is to create a comprehensive and feasible plan for a visual programming language for primary education with a first rate usability.

The theoretical framework of the thesis consists of, above all, the concepts of usability and user experience. These concepts are considered especially from a child's point of view. In addition, the framework covers visual programming languages, gamification, quantitative research and usability testing. A design research is carried out, where the research methods used are observing, a questionnaire and usability testing. In the usability testing the think-aloud protocol is utilized.

The design research carried out in the study created a plan for a visual programming language for children, paying particular attention to usability and user experience. In the usability testing performed on the hi-fi prototype of the Koodikoira program there were relatively many usability problems, which is normal when testing on a system that is still in a development phase. However, only one of the problems was so catastrophic that the system could not be released before it was fixed. In addition, the thesis aggregated literacy guidelines for designing a system for children with a good usability, and proved the guidelines to be functional.

KEYWORDS: user experience, usability, software design, visual programming language, children

1 JOHDANTO

Suomalainen koulujärjestelmä on maailmankuulu, ja se kuuluu edelleen Suomen suurimpiin valttikortteihin. Nykytilanteeseen ei kuitenkaan voida tuudittautua, vaan katseet täytyy suunnata tulevaisuuteen. Tapahtuva teknologinen kehitys muuttaa maailmaa koko ajan yhä tietoteknisemmäksi ympäristöksi, joten myös koulujärjestelmämme on muututtava jatkuvasti ajantasaisemmaksi. Merkittävänä tulevaisuuden alana ohjelmoinnin perusajatuksen ymmärtäminen tulee jatkossa kuulumaan yleisivistykseen. Jopa kattava ohjelmointitaito voidaan katsoa vastedes kuuluvaksi yleisiin taitoihin (Koodikerho 2017). Lasten on kuitenkin joka tapauksessa opittava asioiden ymmärtämisen lisäksi luomaan itse uutta, myös tietoteknisesti (Toikkanen 2015).

Ohjelmointi tulikin Suomen peruskouluihin vuonna 2016 osana matematiikan opetusta (Liukas & Mykkänen 2017). Tämä on tärkeä askel oikeaan suuntaan, jossa jokaiselle oppilaalle annetaan tasapuolinen mahdollisuus innostua ohjelmoinnista, ymmärtää ohjelmoinnillista ajattelua ja nähdä ihmisten tekemiä ratkaisuja jokapäiväisen teknologian takana. Vaikka jokaisesta lapsesta ei luonnollisestikaan tule ohjelmoijaa, biologia tai kielitieteilijää, on kaikille tarjottava yhtäläiset edellytykset alan syvempään opiskeluun jatkossa (Toikkanen 2015).

Opetushallituksen (2014: 235–239) mukaan 3.–6.-luokkalaisten ohjelmoinnin opetuksessa käytetään avuksi graafisia ohjelmointiympäristöjä. Tähän tarkoitukseen onkin jo kehitetty useita vaihtoehtoja, kuten esimerkiksi Scratch, Blockly, Alice ja Swift Playgrounds. Yksikään näistä ei kuitenkaan ole kehitetty Suomessa. (Karch 2017.) Nämä ohjelmointiympäristöt näyttävät myös olevan kehitetty pääasiallisesti koodauksen ehdoilla, eivätkä esimerkiksi käytettävyyden edellä.

Käytettävyys eli järjestelmän soveltuvuus käyttötarkoitukseen, sen tehokkuus ja miellyttävyys (ISO 9241-11: 1998), sekä *käyttäjäkokemus* eli järjestelmän käytöstä johtuva kokonaisvaltainen elämys ovat tärkeitä käsitteitä aina kun puhutaan opetuksessa käytettävistä apuvälineistä. Näiden käsitteiden tärkeys korostuu entisestään lasten

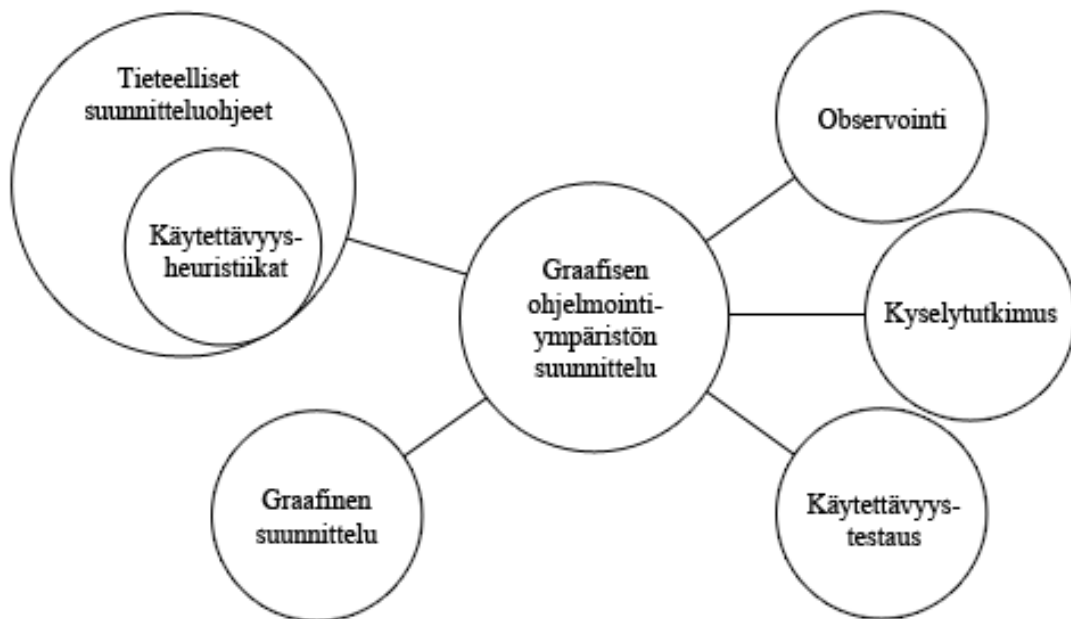
ollessa välineiden pääasiallisia käyttäjiä, sillä orastava innostus oppiaineeseen saattaa lakata helpostikin vain käytettävyysohjelmien takia. Myös opetuksen kannalta hyvästä käytettävyydestä on hyötyä, sillä oppiminen paranee kun käytössä on interaktiivinen oppimisympäristö (Khalifa & Lam 2002: 354–355). Kokkonen ja Ahtinen (2000) muistuttavat, ettei hyvä käytettävyys itsessään takaa hyviä oppimistuloksia, mutta antaa oppimiselle kuitenkin paremmat edellytykset. Hyvä käytettävyys on siis tavoiteltavaa niin lasten innostuksen säilyttämisen kuin oppimisenkin kannalta; onhan sillä ainoastaan positiivisia puolia.

Vaikka hyvän käytettävyyden peruseriaatteet ovat samat sekä aikuisille että lapsille tarkoitetuilla järjestelmillä, lasten parissa työskennellessä tulee kiinnittää huomiota vielä useampaan näkökulmaan. Suurimpia eroja lasten ja aikuisten välillä ovat tietoteknisen tietämyksen taso, motorisen kehityksen vaihe ja erityisesti huomiojännitteen pituus. Mikäli lapset eivät löydä etsimäänsä asiaa vaistomaisesti, he saattavat lopettaa ohjelman käytön kokonaan, eivätkä siis siten käytä sitä opetellakseen ohjelmointia. (Idler 2014.) Ottaen siis huomioon myös kiinnostuksen ylläpitämisen tärkeyden, on äärimmäisen tärkeää luoda graafinen ohjelmointiympäristö ensisijaisesti käytettävyyden ehdoilla.

Koska graafisten ohjelmointiympäristöjen käyttö perusopetuksessa on kirjattu Suomessa opetussuunnitelmaan, olisi tärkeää ottaa ohjelman valinnassa huomioon nimenomaan suomalaisuus ja suomenkielisyys. Tämänhetkiset ohjelmointiympäristöt eivät ole suomalaisten kehittämiä, eikä niitä ole kehitetty suomalaisia lapsia silmällä pitäen. Jotkut graafiset ohjelmointiympäristöt on käännetty suomen kielelle, mutta osittain epätäydellisesti ja epäsystemaattisesti, mikä aiheuttaa sekavuutta niin käytettävyyden kuin oppimisenkin kannalta. Parhaassa tilanteessa käytettävän ohjelman alkuperäinen kieli olisi tietenkin suomi. Ohjelmointiympäristön tehtävien perustuminen suomalaisiin arvoihin, periaatteisiin ja kulttuuritaustaan myös vahvistaisi suomalaista kansallisidentiteettiä.

1.1 Tutkimuksen tavoite ja raja

Tutkimuksen tavoitteena on luoda kattava ja toteutettavissa oleva suunnitelma ensiluokkaisen käytettävyyden omaavasta graafisesta ohjelmointiympäristöstä perusopetukseen. Ohjelma tulee olemaan yksinkertainen, mutta opetustarkoituksessa perusteellinen. Lopputuloksena esiteltävä suunnitelma perustuu kuvion 1 mukaisesti useisiin eri tekijöihin. Niitä ovat hyvän käytettävyyden heuristiikat, muut tieteelliset suunnitteluohjeet, aiemmin suoritettu observointi (Puntila 2016), tehtävän kyselytutkimuksen tulokset ja prototyypin käytettävyydestä saatava aineisto, kuin myös graafisen suunnittelun suositukset.



Kuvio 1. Tutkimuksen liitännät ja myötävaikuttajat

Varsinaisen tavoitteen alatavoitteena on luoda prototyyppi graafisesta ohjelmointiympäristöstä. Prototyypin avulla suoritettava käytettävyydestä kohderyhmän kanssa mahdollistaa todenmukaisten tulosten saamisen ohjelman käytettävyyden senhetkisestä tasosta sekä tietoa mahdollisista käytettävyysongelmista. Käytännön testauksella var-

mistetaan hyvän käytettävyyden lisäksi ohjelman riittävä yksinkertaisuus ja mukaansa-temppaavuus. Käytettävyydestäuksen tulosten avulla parannetaan päätavoitteena olevaa suunnitelmaa.

Ohjelmointiympäristön prototyyppi rakennetaan React-kirjaston avulla JavaScript-kielellä. Prototyypin varsinainen tekninen toteutus rajataan kuitenkin tutkimuksen ulkopuolelle, sillä tutkielma keskittyy käytettävyyteen, käyttöliittymään ja käyttäjäkokemukseen. Näihin asioihin ohjelman ulkokuorella on taustalla olevia teknisiä ratkaisuja suurempi vaikutus. Vaikka graafinen ohjelmointiympäristö kehitetään opetustarkoitukseen, myös opetuksen toimivuus rajataan tutkimuksen ulkopuolelle. Ohjelma on tarkoitettu käytettäväksi opettajan apuna lähiopetustilanteessa, joten opetuksen toimivuus on kiinni myös lukuisista muista tekijöistä, joiden tutkiminen vaatisi pidemmän aikaikkunan ja jatkuvan tutkimuksen.

Graafinen ohjelmointiympäristö suunnitellaan toimivaksi kaikilla käyttöjärjestelmillä ja internetselaimilla, mutta tässä tutkielmassa niitä tarkastellaan Windows 10:n ja Google Chromen näkökulmista, sillä nämä ovat selvästi käytetyimpiä erityisesti Suomessa (StatCounter 2017).

1.2 Tutkimusstrategia ja -menetelmät

Tutkielman tutkimusstrategiana on suunnittelututkimus. Se erottuu tavallisesta järjestelmän suunnittelusta esimerkiksi suunnittelun arvioinnin menetelmien, saatujen tulosten ja täsmällisyyden perusteella (Hevner, March & Park 2004: 98–99). Pönkän (2008: 5) mukaan suunnittelututkimuksen tavoitteena on kehittää eteenpäin sekä teoriaa että käytäntöä. Käytännön tavoitteena on siis luoda hyödyllinen järjestelmä (Hevner ym. 2004: 98), mutta Järvinen (2017: 3) puoltaa tavoitteena olevan myös teoreettisen totuuden selvittäminen. Tässä tutkielmassa käytetään Hevnerin ym. (2004: 86) mainitsemista suunnittelututkimuksen menetelmistä jo aiemmin toteutettua observointia (Puntila 2016) sekä tässä tutkielmassa suoritettavaa käytettävyydestäusta.

Suunnittelu pohjautuu ensiksi mahdollisuuksien mukaan suoritettuun observointiin sekä tunnettuihin ja laajalti käytössä oleviin heuristiikkalistoihin, joista valitaan tämän tutkimuksen osalta relevantimmat. Apuna käytetään myös graafisen suunnittelun suosituksia sekä tehtävän kyselytutkimuksen kvantitatiivisia tuloksia. Näitä ovat esimerkiksi esiin nousseet trendit sekä suosituimmat ilmaukset eri asioille.

Graafisen ohjelmointiympäristön suunnittelutyön lisäksi tutkimuksessa suoritetaan kehitettävän prototyypin käytettävyystudkimus, josta saatavien kvalitatiivisten tulosten avulla parannetaan ohjelman käytettävyyttä. Laadullisten tulosten avulla on tarkoitus perehtyä tarkasteltavaan ilmiöön perusteellisemmin ja ymmärtää sitä syvällisemmin. Löydetyt käytettävyysongelmat järjestellään niiden vakavuuden ja yleisyyden perusteella, ja niihin pyritään löytämään ratkaisut.

Prototyypin käytettävyytestaus suoritetaan kohderyhmän edustajilla; koska ohjelmointia on Opetushallituksen (2014: 235–239) mukaan tarkoitus opettaa graafisten ohjelmointiympäristöjen avulla 3.–6.-luokkalaisille, testihenkilöinä toimivat neljäs-luokkalaiset oppilaat. Testihenkilöt ovat keskenään samaa ikäluokkaa sekä lapsen kehitystason vaikutuksen minimoimiseksi että testijärjestelyistä johtuen. Käytettävyytestaus suoritetaan paritestauksena, ja käytössä on ääneenajattelutekniikka. Ääneenajattelutekniikassa nimensä mukaisesti testihenkilö puhuu ääneen kaikki ajatuksensa. Tämä auttaa testin järjestäjiä pääsemään syvemmälle sisälle testihenkilön ajatuksenkulkuun, johon ei vain ulkoisesti tarkkailemalla pääsisi. Paritestauksen tarkoituksena on saada ääneenajattelutekniikan käyttö tuntumaan luonnollisemmalta; on joku tuttu ystävä, jolle puhua. Lisäksi paritestaus auttaa ujoja lapsia tuomaan ajatuksensa paremmin esille.

Tehtävässä suunnittelututkimuksessa uuden ohjelmiston suunnittelutyö yhdistää observoinnin tulokset ja aiemmat kirjallisuuden ohjeet käytännön käytettävyystudkimukseen. Tutkimus on siis pääosin laadullinen, mutta käytössä on myös määrällisen tutkimuksen keinoja.

1.3 Tutkielman rakenne

Seuraavassa luvussa käydään läpi tutkielman teoreettisen viitekehyksen tärkeimmät käsitteet: käytettävyys, käyttäjäkokemus ja graafiset ohjelmointiympäristöt. Nämä määritellään olennaisimpien lähteiden avulla ja niitä tarkastellaan yksityiskohtaisemmin sekä ohjelmistojen että lasten näkökulmista. Käyttäjäkokemuksen yhteydessä pohditaan myös pelillistämistä. Kolmannessa luvussa määritellään määrällinen tutkimus, sekä esitetään tässä tutkielmassa suoritetun kyselytutkimuksen kysymykset ja tulokset.

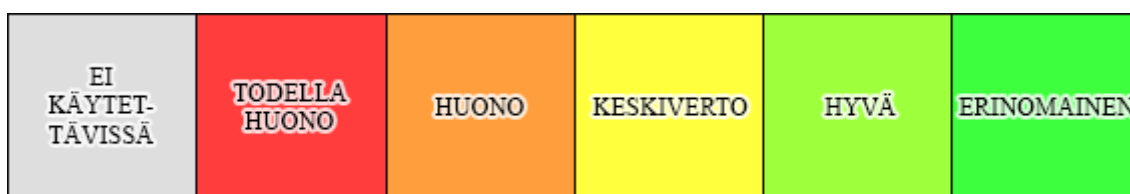
Luvussa neljä esitellään uuden graafisen ohjelmointiympäristön suunnitelma. Suunnitelmaa selventää rakennetusta prototyypistä otetut kuvakaappaukset. Viidennessä luvussa määritellään käytettävyysestaus, käydään läpi tärkeimmät ohjeet järjestelyihin ja kiinnitetään huomiota testikäyttäjänä oleviin lapsiin. Luvussa kuusi suunnitellaan oma käytettävyysestaus, käydään läpi toteutetun testauksen kulku sekä esitetään siitä saadut tulokset. Luvussa seitsemän kootaan yhteen tämän suunnittelututkimuksen tärkeimmät huomiot ja löydökset, sekä pohditaan tutkielman rajoituksia ja jatkotutkimusmahdollisuuksia.

2 KÄYTETTÄVYYS, KÄYTTÄJÄKOKEMUS JA GRAAFISET OHJELMOINTIYMPÄRISTÖT

Käytettävyys ja käyttäjäkokemus ovat käsitteitä, joiden tärkeys korostuu koko ajan entistä enemmän. Jo tällä hetkellä hyvä käytettävyys ja käyttäjäkokemus ovat merkittäviä myyntivaltteja järjestelmilleen. Graafiset ohjelmointiympäristöt puolestaan ovat keino helpottaa, nopeuttaa ja tämän tutkielman aihepiirissä erityisesti opettaa ohjelmointia. Tässä luvussa määritellään käsitteet käytettävyys, käyttäjäkokemus ja graafinen ohjelmointiympäristö. Käytettävyyteen ja käyttäjäkokemukseen pureudutaan syvemmin myös ohjelmistojen ja lasten kannalta katsottuna.

2.1 Käytettävyys

Käytettävyydelle on monia määritelmiä, mutta usein ne eivät ole täysin selkeitä ja yksiselitteisiä. Tällainen sekavuus yhdistettynä tietämättömyyteen voi johtaa useisiin väärinkäsityksiin. Esimerkiksi puhekielessä käytettävyyden voi sekoittaa helposti käytettävissä olevuuteen, joka on kuitenkin aivan eri asia. Jos jokin asia ei ole käytettävissä, sillä ei edes voi olla käytettävyyttä. Käytettävyyttä voisi todeta arvioitavan kuvion 2 mukaisesti asteikolla hyvästä huonoon. Lisäksi asteikolla on kohta ”ei käytettävissä”, jolloin käytettävyyttä ei ole voitu arvioida ollenkaan. Tämä ei kuitenkaan tarkoita että arvosana olisi huonoin mahdollinen, vaan arvioinnin mahdollistuttua arvosana voi olla jopa erinomainen.

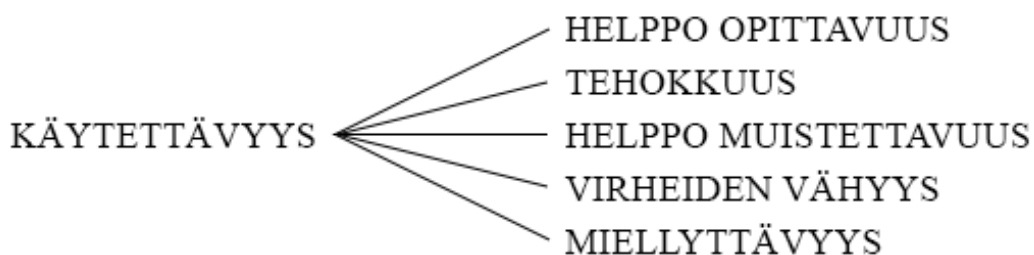


Kuvio 2. Yksinkertainen asteikko käytettävyyden tasosta.

Toinen yleinen virhe on luulo siitä, millä kaikella käytettävyyttä voi ja ei voi olla. Vaikka käytettävyydestä tulee mieleen helposti tietokoneohjelmat, miltei kaikella voi katsoa olevan käytettävyyttä aina oven kahvasta teollisiin tuotteisiin ja jopa palveluihin. Käytettävyyden tärkeyden vuoksi ja väärinkäsitysten välttämiseksi käyttöön on vakiintunut kaksi suosittua määrittelytapaa, jotka ovat ISO 9241-11 -standardi (1998) sekä Nielsenin (1993: 25) kehittämä malli.

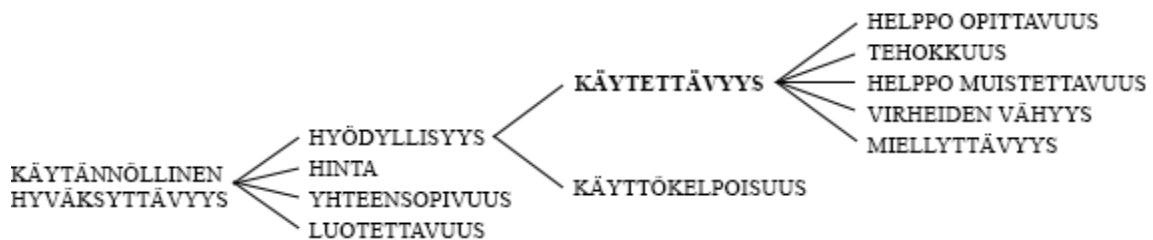
ISO 9241-11 -standardin (1998) mukaan käytettävyys määritellään järjestelmän toimivuutena tuloksellisesti, tehokkaasti ja miellyttävästi tietyssä tilanteessa. Käytettävyys onkin siis aina tapauskohtaista, sillä järjestelmän käyttäjät, tavoitteet ja ympäristö voivat vaihdella runsaastikin tapauksesta riippuen. Toisin sanoen tämän määritelmän mukaan käytettävyys tarkoittaa järjestelmän sopivuutta käyttötarkoitukseensa.

Nielsenin (1993: 25) kehittämä malli (ks. kuvio 3) ottaa huomioon ISO 9241-11 -standardissa mainittujen lisäksi inhimillisempiä ominaisuuksia, kuten opittavuuden ja muistettavuuden. Vaikka ne voidaan katsoa kuuluvan osaksi ISO-standardin määrittelemää tehokkuutta, ne on hyvä mainita myös erikseen; pelkällä tehokkuudella voi helposti ymmärtää tarkoitettavan järjestelmän toiminnan nopeutta sen sijaan, että esimerkiksi järjestelmän käyttämisen opettelu kuuluisi siihen. Käytön opittavuus ja myöhemmin sen pariin palatessa muistettavuus, sekä virheiden vähyys vaikuttavat kuitenkin käyttöprosessin kestoon merkittävästi.



Kuvio 3. Nielsenin määritelmä käytettävyydestä (Nielsen 1993: 25).

Kuten kuvio 4 voidaan todeta, Nielsenin (1993: 25) malli sijoittaa käytettävyyden osaksi suurempaa kokonaisuutta. Käytettävyyden yläkäsitteenä on hyödyllisyys ja rinnakkaiskäsitteenä käyttökelpoisuus. Tämä tarkoittaa sitä, että järjestelmä voi olla hyödyllinen vaikka sen käytettävyys olisikin todella huono, kunhan se on käyttökelpoinen. Tällaisia tapauksia voivat olla esimerkiksi tilanteet, joissa järjestelmää tarvitsee käyttää vain vähän aikaa kerrallaan tai verrattain harvoin.



Kuvio 4. Nielsenin määritelmä käytännöllisestä hyväksyttävyydestä (Nielsen 1993: 25).

Hyödyllisyys on edelleen osa käytännöllistä hyväksyttävyyttä. Hyödyllisyyden rinnakkaiskäsitteitä ovat esimerkiksi hinta, yhteensopivuus ja luotettavuus. (Nielsen 1993: 25) Järjestelmä voi siis olla jopa käytännöllisesti hyväksyttävä vaikka sen hyödyllisyys olisi kiistanalainen. Jos hinta on sopiva ja toimintavarmuus riittävä, järjestelmä voi olla käytännöllisesti hyväksyttävä vaikka esteettisyytensä vuoksi. Käytännöllinen hyväksyttävyys on puolestaan osa järjestelmän kokonaisvaltaista hyväksyttävyyttä, jossa toisena osana sosiaalinen hyväksyttävyys (Nielsen 1993: 25).

2.1.1 Käytettävyys ohjelmistojen näkökulmasta

Ohjelmistojen kannalta katsottuna käytettävyys on suurimmaksi osaksi sen käyttöliittymän käytettävyyttä. Tästä syystä on kehitetty useita erilaisia ohjeita nimenomaan käyttöliittymän suunnitteluun hyvän käytettävyyden saavuttamiseksi. Tunnetuin lista ohjeita on Nielsenin (1995) kymmenen käytettävyysheuristiikkaa käyttöliittymälle. Listan mukaan käyttöliittymän tulee antaa käyttäjälle vapaus ja

kontrolli kaikkeen, näyttää systeemin tila jatkuvasti, olla joustava ja tehokas, sekä olla yhtenäinen standardien kanssa. Lisäksi käyttöliittymän tulisi olla minimalistinen, suosia tunnistamista muistamisen sijaan, tarjota apua dokumentaationa, pyrkiä estämään järjestelmän virheellinen käyttö ja virheen tapahtuessa auttaa käyttäjää eteenpäin (Nielsen 1995). Juristo, Moreno ja Sánchez-Segura (2007: 745) puolestaan esittävät listan käytettävyyttä parantavista ominaisuuksista, jotka ohjelmistossa olisi hyvä olla; mm. keskeytä- ja kumoa-toiminnot, näppäinoikotiet, käyttäjäprofiilit ja informaation helppo uudelleenkäyttö.

Nämä heuristiikat ja ominaisuudet ovat kuitenkin vain tarkoitettu toimimaan ohjenuorana, sillä aivan jokainen ohje ei ole relevantti kaikille järjestelmille, ja useat järjestelmät tarvitsevat myös muita näkökulmia käytettävyyteen. Tarkoituksena onkin lukea useampia heuristiikkalistoja, valita niistä olennaiset ohjeet, ja tarvittaessa kehittää jopa omia, joiden avulla sitten toteuttaa ohjelmistosuunnittelua. Carvajal, Moreno, Sánchez-Segura ja Seffah (2013: 1595) muistuttavat, että vaikka erinäisiä käytettävyyden ohjelista pyrittäisiin toteuttamaan ohjelmistokehityksessä, ei niiden oppien välittäminen ohjelmiston koodiin ole suoraviivaista. Paras tapa varmistaa ohjelmiston hyvä käytettävyys on kiinnittää siihen huomiota jo suunnitteluprosessin alusta alkaen (Juristo ym. 2007: 755).

Koska käytettävyys tulee ottaa huomioon laajalti suunnitteluprosessin aikana, Gould, Boies ja Lewis (1991) tuovat esille neljä hyödyllistä toimintatapaa, jotka kuuluvat korvaamattomana osana ohjelmistokehitykseen, jonka tarkoituksena on tuottaa hyvän käytettävyyden omaava ohjelmisto. Ensinnä huomio tulee kiinnittää ohjelman loppukäyttäjiin jo alkuvaiheessa pyrkimällä ymmärtämään heidän lähtökohtiaan ja ohjelmalle asettamia tavoitteitaan. Toisena koko ohjelmistoa tulisi kehittää eteenpäin yhtäaikaaisesti yhtenäisyyden säilyttämisen vuoksi. Kolmantena toimintatapana Gould ym. (1991) mainitsevat empiirisen käytettävyydestauksen, joka tulisi aloittaa mahdollisimman aikaisin, ja sen tulisi jatkua koko ohjelmistokehityksen ajan. Viimeisenä asiana iteratiivinen eli toistuva suunnittelu, jossa aikaisempaa versiota parannetaan testauksen, palautteen ja arvioinnin seurauksena.

Ohjelmiston käytettävyys ei kuitenkaan ole kokonaan pelkästään käyttöliittymästä kiinni, vaan myös itse ohjelman toimintalogiikka tulee olla johdonmukainen (Carvajal ym. 2013). Vaikka käyttöliittymä olisi ensiluokkainen, ei se voi korjata perustavanlaatuisia epäloogisuuksia ohjelman toimintaprosesseissa. Esimerkiksi verkkokaupan tilausprosessin tulee toimia oikeassa ja ymmärrettävässä järjestyksessä, sillä käyttöliittymä ei voi järjestystä vaihtaa. Järjestelmän tulee myös tukea puhtaasti käytettävyyden parantamiseksi käyttöliittymään lisättyjä toimintoja kuten näppäinoikoteitä.

Käytettävyyden määritelmä ei muutu oli kyseessä sitten ohjelmisto, fyysinen tuote tai mikä tahansa muu. Joka tapauksessa järjestelmältä vaaditaan hyvää opittavuutta, tehokkuutta, hyvää muistettavuutta, virheiden vähyyttä ja miellyttävyyttä (Nielsen 1993: 25), sekä lisäksi tuloksellisuutta (ISO 9241-11). Tässä alaluvussa mainitut heuristiikkalistat ja toimintatavat ovat vain keinoja näiden ominaisuuksien saavuttamiseen; työkaluja käytötarkoitukseensa sopivan järjestelmän kehittämiseen.

2.1.2 Käytettävyys lasten näkökulmasta

Käytettävyyden peruseriaatteet ovat samat sekä lapsille että aikuisille tarkoitetuilla järjestelmillä (Nam 2010). Kuitenkin koska käytettävyys on aina tapauskohtaista, lasten ollessa järjestelmän käyttäjinä tulee kiinnittää huomiota useisiin asioihin, jotka ovat aikuisille helppoja ja itsestäänselviä. Taulukossa 1 on esitetty lasten erityispiirteitä aikuisiin verrattuna internetsivuja käytettäessä. Käytettävyyden kannalta suurin ero aikuisten ja lasten välillä on kärsivällisyys (Logan 2009; Quibly 2013). Lasten kärsivällisyys voi loppua nopeastikin esimerkiksi latausaikojen pituuden tai navigaation sekavuuden takia, mikä johtaa heidän poistumiseen sivulta välittömästi (Quibly 2013).

Toinen erityispiirre ja yksi syy kärsimättömyyteen on lasten tarve tulla viihdytetyiksi. Lasten ensisijainen tavoite ohjelmistoa käyttäessä on pitää hauskaa (Quibly 2013; Idler 2014), kun taas aikuisilla tavoitteena on useimmiten tiedon hankkiminen (Idler 2014). Jatkuva odottelu ja sekavuuden tunne ei ole hauskaa.

Idler (2014) jatkaa kertomalla vielä muita erityispiirteitä, joista ensimmäinen on

fyysinen kehitys. Lasten fyysinen kehitys on vielä kesken, eikä heidän tietotekninen osaaminen ole vielä korkealla tasolla. Tämä tulee erityisesti ilmi hiiren käytössä, sillä motoriikan vasta kehittyessä hiirellä on vaikea osua pieniin kohteisiin, puhumattakaan raahaa ja pudota (eng. drag and drop) -toiminnoista. Seuraavana piirteenä on sivuston käyttötapa. Lapset ovat uteliaita ja haluavat tutkivat koko sisällön, kun taas aikuiset tietävät mitä etsivät ja toimivat totuttujen kaavojen mukaan löytääkseen etsimänsä. Tekstin lukeminen ja hiiren rullan käyttäminen ovat myös hyvin erilaisia lasten ja aikuisten välillä. Lapset lukevat mahdollisimman vähän tai ei ollenkaan tekstiä, eikä sivuston rullaaminen ole heidän mielestä itsestään selvä etenemistapa. Viimeisenä piirteenä on mainonta, jota lapset eivät erota varsinaisesta sisällöstä ollenkaan. (Idler 2014.)

Taulukko 1. Lasten erityispiirteitä aikuisiin verrattuna internetsivujen käytössä (Quibly 2013; Idler 2014).

	LAPSET	AIKUISET
Kärsivällisyys	todella lyhyt	rajattu
Tavoite	viihtyminen	tiedonhankinta
Fyysinen kehitys	kesken	tapahtunut
Sivuston käyttötapa	tutkii kaiken	toimii kaavan mukaan
Lukeminen ja rullaaminen	ongelmallista	ei ongelmaa
Mainonta	ei erota sisällöstä	jättää huomiotta

Yksittäisiä keinoja panostaa lasten käyttämien ohjelmien käytettävyyteen on käyttää mahdollisimman vähän ja yksinkertaista kieltä, varmistaa lyhyet latausajat, kehittää selkeä navigaatio ja erottaa klikattavat asiat muusta sisällöstä ymmärrettävästi. (Logan 2009.) Tärkeintä on kuitenkin toteuttaa käytettävyydestä lasten itsensä kanssa ohjelmistokehityksen aikana.

2.2 Käyttäjäkokemus

Käyttäjäkokemus on käytettävyyttä laajempi kokonaisuus, eikä senkään tarkka määrittelemine ole helppoa. Termin kehittäjä Norman toteaa käyttäjäkokemuksen nimittäin kattavan kaiken kanssakäymisen ja vuorovaikutuksen, joka tapahtuu loppukäyttäjän ja yrityksen, sen palveluiden ja tuotteiden kanssa. Tällainen kokonaisvaltainen elämys vaatii tietenkin käyttäjän tarpeiden täyttämisen vaivatta ja järjestelmän käytöstä aiheutuvan tyytyväisyyttä, mutta siihen liittyy myös paljon muuta. Ensiluokkaisen käyttäjäkokemuksen saavuttaminen edellyttää monen osaston, kuten markkinoinnin, ohjelmistosuunnittelun, graafisen suunnittelun ja lopullisen toteutuksen toimimista saumattomasti yhteen. (Norman & Nielsen 2018.)

Gabriel-Petit (2018) tarkentaa käyttäjäkokemuksen määritelmää kertomalla sen kuvaavan miten käyttäjät kokevat, oppivat ja käyttävät tuotetta tai palvelua, sen muotoja, sisältöä ja jopa sen käyttäytymistä. Tärkeimpiä tekijöitä kokemuksen kannalta ovat hänen mukaansa käytettävyys, opittavuus, hyödyllisyys sekä visuaalinen viehättävyys. Käyttäjäkokemus käsitteenä laajentuu kuitenkin kattamaan esimerkiksi myös sen, miten käyttäjä kokee kyseisen brändin kokonaisuutena ja mitä tunteita se heissä herättää. (Gabriel-Petit 2018.)

On tärkeä erottaa termit käyttäjäkokemus ja käytettävyys toisistaan. Samoin täytyy erottaa käyttäjäkokemus ja käyttöliittymä. Vaikka käytettävyydellä ja käyttöliittymällä on jopa suurimmat roolit kokonaisvaltaisessa käyttäjäkokemuksessa, kokemuksesta puhuttaessa on kyseessä kattavampi konsepti. Vaikka käyttöliittymä olisi täydellinen ja järjestelmän käytettävyys ensiluokkainen, saattaa käyttäjäkokemus olla huono esimerkiksi puuttuvan sisällön vuoksi. Puuttuva sisältö voi olla sellaista, jonka ei edes ole tarkoitus olla järjestelmässä, mutta koska käyttäjä odottaa sen olevan, koettu käyttäjäkokemus laskee. (Norman & Nielsen 2018.)

Loppujen lopuksi käyttäjäkokemus koostuu nimenomaan siitä, kuinka käyttäjä tulkitsee ja kokee tapahtuneen vuorovaikutuksen; mitä se saa hänet tuntemaan. Hyvä käyttäjäkokemus aiheuttaa tärkeyden, mielihyvän, pätevyyden ja tehokkuuden tunteita, kun

taas vastavuoroisesti huono käyttäjäkokemus aiheuttaa turhautumista, närkästymistä ja tyytymättömyyttä. Juuri tästä syystä käyttäjäkokemus on tärkeä käsite. (Schmidt & Etches 2012: 11.)

2.2.1 Käyttäjäkokemus lasten näkökulmasta

Kuten käytettävyyteen, myös käyttäjäkokemukseen vaikuttavilla tekijöillä on eroja riippuen käyttäjän iästä. Suuri osa lasten käyttäjäkokemusta tulee käytettävyyden kautta, sillä käytettävyyttä suunniteltaessa huomiota kiinnitetään itse asiassa hyvin pitkälle myös käyttäjäkokemukseen. Esimerkiksi käyttäjäkokemuksen kannalta suurin ero aikuisten ja lasten välillä on tavoite järjestelmien käytössä, joka vaikuttaa myös hyvään käytettävyyteen tähtääviin menettelytapoihin. Lasten ensisijaisena tavoitteena on siis pitää hauskaa ja viihtyä käyttämänsä järjestelmän parissa (Quibly 2013; Idler 2013; Idler 2014).

Sim, MacFarlane ja Read (2006) tutkivat käytettävyyden, hauskuuden ja oppimisen suhdetta. He toteavat käytettävyydeltään hyviä ohjelmia pidettävän lasten parissa myös hauskempina. Hauskuuden ja käytettävyyden suoraa yhteyttä oppimiseen ei puolestaan todettu, mutta hauskempien ohjelmien parissa vietetään yleensä enemmän aikaa ja siten oppimiselle on paremmat mahdollisuudet. Hietanen ja Ovaska (2002: 11) korostavatkin, että lapsille suunnatun käyttöliittymän tulee olla innostava ja mukaansatempaava, sillä juuri motivaatiolla on suuri rooli niin ohjelmiston parissa viihtymisen kuin oppimisenkin kannalta. Kokonaisvaltaisen käyttäjäkokemuksen näkökulmasta hauskuus ja käytettävyys ovat siis äärimmäisen tärkeitä käsitteitä.

Viihtymisen ja käytettävyyden lisäksi hyvää käyttäjäkokemusta suunniteltaessa tulee muistaa visuaalinen viehättävyys. Toisin kuin aikuiset, lapset tarvitsevat yleensä jonkin ulkoisen motivaation käyttääkseen ohjelmaa. Pelkästään houkuttelevan ja hauskan näköinen ulkonäkö voi jo itsessään olla riittävä motivaatio. (Idler 2013.) Lapset pitävät erityisesti kirkkaista ja iloisista väreistä, ja värit ovatkin yksi merkittävimmistä tekijöistä suunniteltaessa lapsille. On totuttu toteamaan, että käyttöliittymässä olisi hyvä käyttää korkeintaan kolmea eri väriä, mutta tätä sääntöä voi lasten parissa rikkoa, sillä

suurempi määrä erilaisia värejä on jopa hyvä asia. Eri värejä voi olla jopa tuplasti aikuisten käyttöliittymiin verrattuna. Värien tulisi olla keskenään selvästi erilaisia sen sijaan, että käyttäisi saman värin eri sävyjä. (Cousins 2014.)

Ohjelman sisällön on puolestaan oltava lapsen ikäryhmälle sopivaa sekä luonnostaan kiinnostavaa. Liian yksinkertainen sisältö koetaan helposti tylsäksi, mutta liian monimutkainen sisältö puolestaan aiheuttaa lapsille keskittymisongelmia. Koska lapset kehittyvät nopeasti, on jopa mahdotonta suunnata samanlaista ohjelmaa eri ikäryhmille. Ohjelman tulisikin palvella tarkasti nimenomaan kohderyhmänsä käyttäjiä, tai tarjota useita tasoja niin, että jokainen ikäryhmä löytäisi kiinnostavaa ja haastavaa, mutta selkeätä sisältöä. (Idler 2013.)

2.2.2 Pelillistäminen

Pelillistäminen on yksi erittäin suosittu tapa parantaa sen käytöstä aiheutuvaa käyttäjäkokemusta ja lisätä motivaatiota järjestelmän käyttämiseen. Sillä tarkoitetaan Deterdingin, Dixonin, Khaledin ja Nacken (2011: 13) mukaan pelillisten elementtien käyttämistä muissa kuin pelien kontekstissa. Pelillistetyt järjestelmät eivät siis ole itsessään pelejä, vaan käyttävät hyväkseen samoja elementtejä, jotka saavat pelit tuntumaan mukaansatempaavilta ja viehättäviltä (Deterding ym. 2011: 13). Tondello (2016: 15) toteaa, ettei järjestelmän hyödyllisyys itsessään ole aina riittävä motivaattorin käyttämiseen, vaan se vaatii myös muita tekijöitä, joihin pelillistäminen on yksi ratkaisu.

Käytännössä pelillistetyssä järjestelmässä saatetaan suorittaa erilaisia tehtäviä, kerätä pisteitä tai pyrkiä voittamaan vastustajia. Tondellon (2016: 16) mukaan pelillistäminen vastaa näin kolmeen psykologiseen tarpeeseen; taitavuuden, autonomian ja yhteenkuuluvuuden tunteisiin. Näiden tekijöiden täyttäminen johtaa usein pelistä nauttimiseen (Rigby & Ryan 2011). Vaikean tehtävän suorittaminen saa käyttäjän tuntemaan itsensä taitavaksi, mahdollisuus valita eri vaihtoehdoista aiheuttaa autonomian tunteen, ja pelaaminen muiden kanssa joko yhdessä tai vastaan saa heidät tuntemaan yhteenkuuluvuutta (Tondello 2016: 16).

Pelillistämisen vaikutus kokonaisvaltaiseen kokemukseen oli Fitz-Walterin, Johnsonin, Wyethin, Tjondronegoron ja Scott-Parkerin (2017: 594) suorittaman empiirisen tutkimuksen mukaan positiivinen. He myös totesivat, että motivaatio ja pelistä nauttiminen kasvoivat. Winter (2015) puolestaan toteaa, että pelillistäminen tekee järjestelmän käytöstä jopa hauskaa. Hän korostaa, että pelit ovat kuin tarina, johon voi uppoutua, sekä taival, joka tarjoaa pelaajalle ainutlaatuisen kokemuksen.

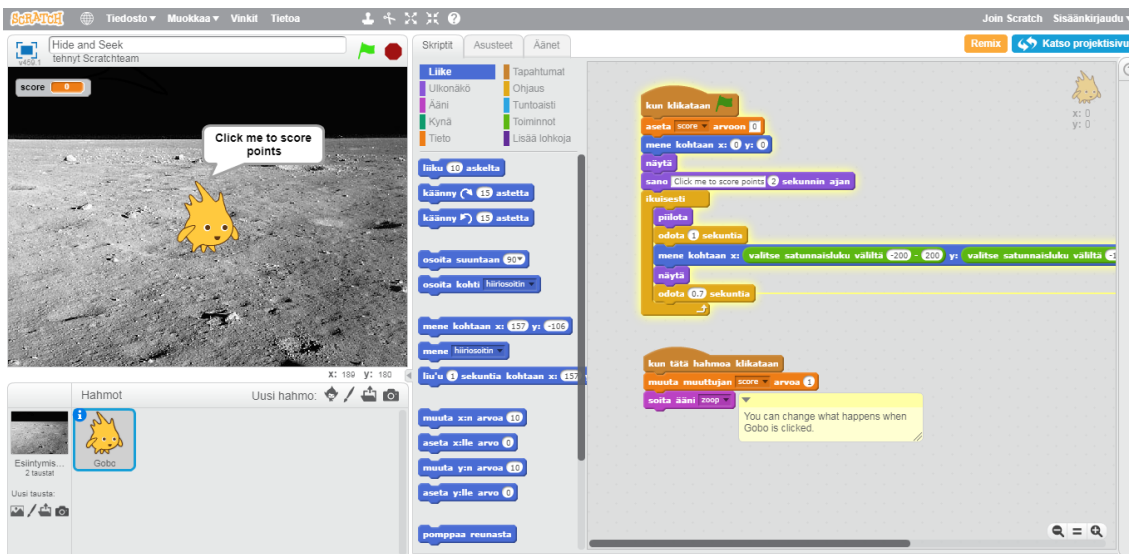
Pelillistämisen suurimmat vaikutukset ovat siis motivaatioon ja viihtymiseen liittyviä. Nämä olivat myös käyttäjäkokemuksen tärkeimpiä ominaisuuksia erityisesti lasten kannalta tarkasteltuna. Pelillistäminen ei ole kuitenkaan varma tapa parantaa käyttäjäkokemusta, sillä esimerkiksi sekavat tai liian vaikeat tehtävät saavat käyttäjät helposti hämilleen (Winter 2015).

2.3 Graafiset ohjelmistoympäristöt

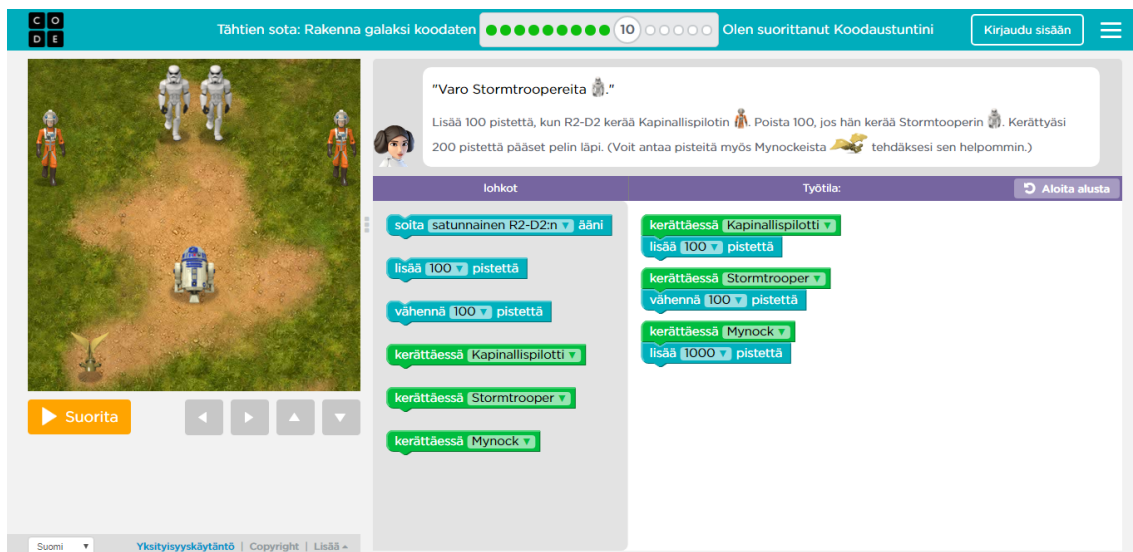
Graafisilla ohjelmointiympäristöillä tarkoitetaan ohjelmia, joissa koodia ei tarvitse itse kirjoittaa. Sen sijaan koodia rakennetaan pelkästään järjestelemällä ja yhdistelemällä erilaisia graafisia elementtejä systemaattisesti. Tällaisia elementtejä ovat esimerkiksi kuvat, kuviot, lohkot, ikonit, symbolit ja teksti. Elementit toimivat graafisissa ohjelmointiympäristöissä sekä syötteinä, toimintoina että tulosteina. Graafisia ohjelmointiympäristöjä kutsutaan myös visuaalisiksi ohjelmointiympäristöiksi. (Opetushallitus 2017; Technopedia 2018.)

Graafisten ohjelmointiympäristöjen on tarkoitus tehdä ohjelmointia helpommin lähestyttäväksi varsinkin aloittelijoille. Aloittelijat voivat usein turhautua syntaksivirheiden takia, mutta tällaisten ohjelmointiympäristöjen avulla heidän ei tarvitse osata kirjoittaa mitään ohjelmointikieltä; ohjelman ja koodin voi rakentaa kasaamalla eri näköisiä palikoita. Palikat kasataan yhteen ainoalla tavalla jolla ne toisiinsa sopivat. (Dehouck 2015.) Syntaksivirheiden mahdollisuus on siis estetty jo alkuvaiheessa. Tämä mukailee yhtä Nielsenin (1995) kymmenestä käyttöliittymän käytettävyyshuristiikasta, jonka mukaan järjestelmän tulisi pyrkiä torjumaan virheiden mahdollisuutta.

Graafisia ohjelmointiympäristöjä pidetään parhaana pedagogisena tapana aloittaa ohjelmoinnin opettelua (Kekäläinen 2014). Etuna niissä on virheiden torjunnan lisäksi Opetushallituksen (2017) mukaan nopea alkuun pääseminen. Yksinkertainen ohjelma on valmis jo oppitunnin aikana, joten alkukankeus ei pääse jarruttamaan innostusta. Kuvissa 1 ja 2 ovat esimerkit opetustarkoitukseen kehitetyistä ohjelmointiympäristöistä.



Kuva 1. Graafinen ohjelmointiympäristö Scratch sekä eräs esimerkkiohjelma.



Kuva 2. Graafinen ohjelmointiympäristö Code.org sekä eräs ohjelmointikurssi.

3 MÄÄRÄLLINEN TUTKIMUS JA SUORITETTU KYSELY- TUTKIMUS

Tässä luvussa määritellään määrällinen tutkimus, sekä käsitellään kyselytutkimuksen hyviä ja huonoja puolia. Lisäksi esitellään tehdyn kyselytutkimuksen (liite 1) kysymykset ja käydään läpi saadut tulokset.

3.1 Määrällinen tutkimus

Määrällisellä eli kvantitatiivisella tutkimuksella tarkoitetaan tutkimusmenetelmiä, joissa kerätään, analysoidaan ja esitetään numeerista tietoa. Nimensä mukaisesti määrällinen tutkimus keskittyy määrällisiin paljoutta mittaaviin tekijöihin, kuten numeroihin, prosenttiosuuksiin ja tilastoihin. (Donmoyer 2008: 713; Sheragy 2018.) Määrällisessä tutkimuksessa eräs äärimmäisen tärkeä käsite on sen validiteetti, joka kuvaa tutkimusmenetelmän kykyä antaa tuloksia juuri tutkittavaan asiaan. (Donmoyer 2008: 713–714). Jos validiteetti on huono, eli tutkimusmenetelmä ei anna vastauksia esitettyyn kysymykseen, menetelmää voidaan pitää tutkimuksen kannalta tarpeettomana.

Sheragy (2018) esittelee kolme määrällisen tutkimuksen hyvää puolta, joista ensimmäisenä hän mainitsee sen nopeuden. Tietoa saa sekä kerättyä että analysoitua todella nopeasti. Toisena on tulosten yleistettävyyden myös tutkimuksessa mukana olleiden yksilöiden ulkopuolelle. Siitä huolimatta on tärkeätä saada vastauksia edustavalta joukolta (Rohan 2012: 12). Viimeisenä Sheragy mainitsee tutkimuksissa vallitsevan anonymiteetin, joka voi auttaa käsittelemään arkojakin aiheita. Mander (2017) on hyvin pitkälle samaa mieltä Sheragyn kanssa, mutta tuo esiin vielä lisäksi objektiivisuuden ja taloudellisuuden. Määrällinen tutkimus on numeerista, joten objektiivisuus on sille jopa luontaista. Tutkimus on myös taloudellisempaa tarkasteltaessa hintaa yhtä osallistujaa kohden.

Myös huonoja puolia määrällisessä tutkimuksessa on, kuten vastausmahdollisuuksien

rajallisuus. Samoin kysymyksenasettelussa tulee olla tarkkana, ettei pakota vastaajia tiettyihin vastauksiin, jotka eivät välttämättä ole todenmukaisia. Mikäli tutkimuksen suorittaa epäluonnollisessa ympäristössä, saadut tulokset eivät välttämättä vastaa todellisuutta. Viimeisenä, mutta usein ratkaisevana huonona puolena määrällisessä tutkimuksessa ovat kustannukset, sillä tutkimuksen laajuuden kasvaessa sen kokonaishinta voi nousta korkealle. (Mander 2017; Sheragy 2018.)

Yksi tavallisimmista määrällisistä tutkimuksista on kyselytutkimus, jonka tekemiseen Froats (2013) antaa kuusi neuvoa. Ensimmäisenä tulee tietää, mitä tietoa kyselystä tarvitsee, sillä muuten kysymyksiä ei voi muodostaa. Toisena kysymykset tulee pitää yksinkertaisina ja yhtenäisinä. Kolmantena vastaamisen tulee olla helppoa, eikä kahta useampaa vastaustapaa (esim. valintaruutu, ympyröinti tai värittäminen) suositella käytettäväksi yhdessä kyselyssä.

Neljäntenä neuvona Froats (2013) jatkaa kyselyn tulevan suunniteltavan niin, että siihen vastaamiseen menee alle viisi minuuttia. Rohanin (2012: 12) mukaan puolestaan kyselytutkimus voi periaatteessa olla kuinka pitkä tai monimutkainen tahansa, kunhan se on rakennettu selkeästi ja jaoteltu ymmärrettäviin osiin. Pituus voi kuitenkin muodostua ongelmaksi, jos se alkaa lannistamaan vastaajia. Viidentenä ohjeena Froats (2013) kehottaa tarjoamaan ihmisille jonkun syyn osallistumaan kyselyyn. Tällainen syy voi olla esimerkiksi lahjakortti. Viimeisenä neuvona on, että olisi hyvä jos henkilötietoja ei olisi välttämätöntä antaa.

3.2 Suoritettu kyselytutkimus

Suoritetun kyselytutkimuksen lomake on nähtävissä kokonaisuudessaan liitteessä 1. Tutkimus suoritettiin neljäsluokkalaisilla, sillä he ovat loppukäyttäjiä hyvin edustava joukko. Kyselylomake pidettiin yksisivuisena, että sen pituus ei aiheuttaisi lapsissa lannistumisen tunnetta. Kysymyksiä oli seitsemän, joista kahdessa viimeisessä oli kolme alakohtaa. Opettajan ohjeistuksen avustuksella lomakkeen täyttämiseen suunniteltiin kuluvaksi noin viisi minuuttia.

Ensimmäinen kysymys koski lapsen lempiväriä. Yleensä hyvä aloittaa helpolla kysymyksellä, johon jokainen osaa vastata helposti. Tämä antaa vastaajalle itsevarmuutta ja siten koko kyselyyn vastaaminen tuntuu helpommalta. Toisena kysymyksenä kysyttiin lapsen harrastuksia tai harrastuksia, joita hän haluaisi harrastaa. Kolmannessa kysymyksessä tiedusteltiin lapsen lempihahmoa ja neljännessä idolia. Viides kysymys puolestaan tiedusteli lapsen mielestä parasta asiaa maailmassa. Näiden viiden ensimmäisen kysymyksen tarkoituksena on havaita trendejä, jotka vallitsevat loppukäyttäjien keskuudessa juuri nyt.

Kysymykset kuusi ja seitsemän jaettiin kolmeen kohtaan, joissa kysyttiin lapsen mielestä sopivimpia ilmaisuja. Kuudennessa kysymyksessä (kuva 3) lapsia pyydettiin ympäröimään kuvaavimpia ikoneja liikkumiselle, uuden asian lisäämiselle ja kääntymiselle. Vaihtoehtoina jokaiseen annettiin kolme eri ikonia, jota pyrkivät kuvaamaan toimintoja mahdollisimman hyvin. Viimeisessä, seitsemännessä kysymyksessä puolestaan tiedusteltiin kuvaavimpia sanallisia ilmaisuja. Vaihtoehtoina kuvaamaan yhden ruudun mittaista etenemistä olivat *askel*, *ruutu*, *sentti* ja *hyppy*. Jaoteltua tietoa sisältäville paikoille vaihtoehtoja olivat *välilehti*, *sivu*, *lehti* ja *taso*. Viimeisenä kysyttiin kuvaavinta sanaa tietokoneohjelman käytön aloittamiselle, jolle vaihtoehtoina olivat *ajaminen*, *aloittaminen*, *kokeileminen* sekä *käynnistäminen*.

6. Ympyröi seuraavista kuvioista se, joka kuvaa mielestäsi parhaiten...

a) liikkumista



b) uuden asian lisäämistä



c) kääntymistä



Kuva 3. Suoritetun kyselytutkimuksen kysymys numero kuusi.

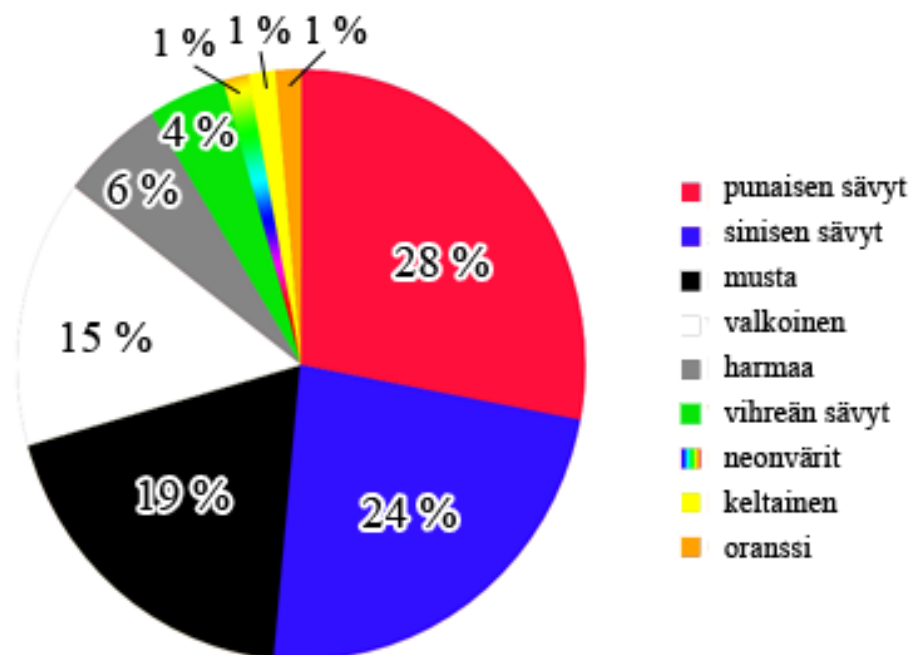
Kyselytutkimuksessa paljastuneita lasten lempivärejä pyritään käyttämään hyväksi ohjelman käyttöliittymässä mahdollisimman paljon. Löydettyjä trendejä hyödynnetään ohjelman suunnittelussa valitsemalla teemoiksi suosittuja hahmoja ja harrastuksia mahdollisuuksien mukaan. Lapsista on tärkeä saada puuhailla heille mielestään siistien ja ei liian lapsellisten asioiden parissa (Druin, Bederson, Boltman, Miura, Knotts-Callahan & Platt 1999). Selvittämällä todelliset kiinnostuksen kohteet voidaan päästä mahdollisimman lähellä tätä haluttua siisteyttä.

3.3 Kyselytutkimuksen tulosten läpikäynti

Kyselytutkimukseen osallistui yhteensä 39 oppilasta. Tutkimus suoritettiin yhdessä koulussa kahdella eri rinnakkaisluokalla, joiden oppilaat myös osallistuvat myöhemmin tehtävään käytettävyydestiin. Kaikki vastanneet olivat siis täydellisesti loppukäyttäjiä edustavia neljäsluokkalaisia oppilaita. Aivan jokainen vastaaja ei vastannut kaikkiin kysymyksiin ja suurimpaan osaan kysymyksistä sai vastata halutessaan useamman vaihtoehdon. Tästä syystä eri kysymysten kokonaisvastausten määrä vaihtelee kolmenkymmenenkahdeksan ja kuudenkymmenenkahdeksan välillä.

Tässä alaluvussa esitetyissä kuvaajissa osa vastauksista on jaoteltu teemoittain yhden käsitteen alle; esimerkiksi tummansininen ja vaaleansininen ovat merkittyinä sinisiksi. Osassa taas tuodaan esille vain suosituimmat vastaukset. Kyselytutkimuksen tulokset kokonaisuudessaan, kuten eri värisävyjen osuudet ovat esitettynä liitteessä 2. Kaikki jakaumat on pyöristetty vain prosentin tarkkuudelle, joten niiden summaaminen yhteen ei aina johda tasan sataan prosenttiin.

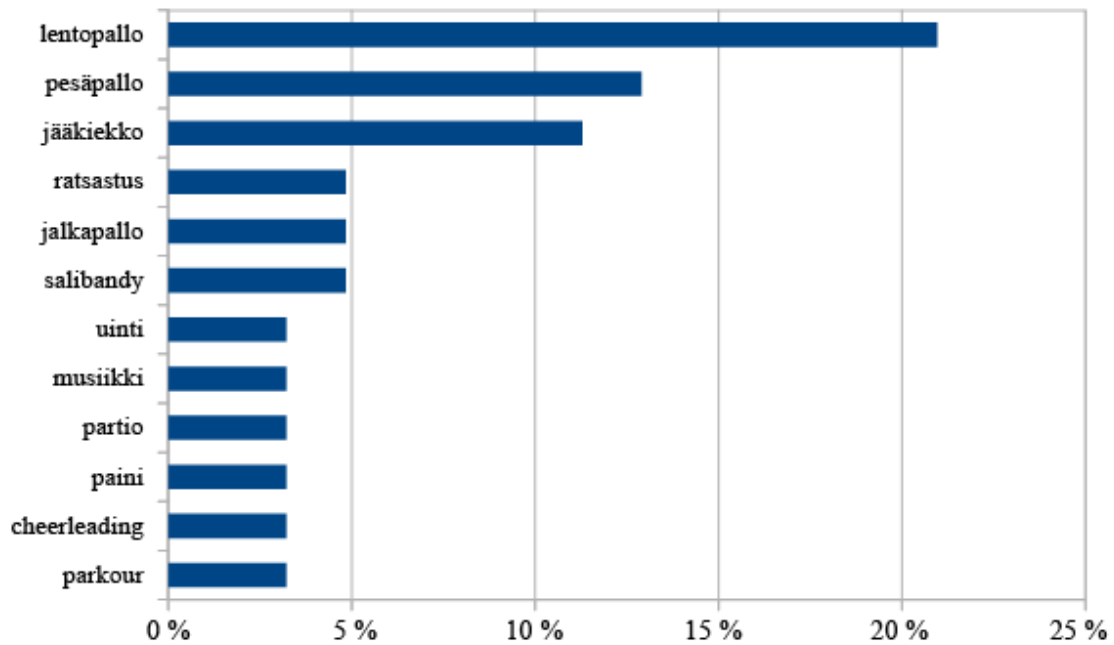
Ensimmäisessä kysymyksessä kysyttiin lasten lempiväriä. Kuten kuviosta 5 voidaan nähdä, suosituimmat teemavärit lasten keskuudessa olivat ehkä odotetustikin punaisen ja sinisen eri sävyjä. Seuraavaksi suosituimpia teemavärejä olivat musta, valkoinen ja harmaa. Vähiten kyselyyn vastattuja lempivärejä olivat vihreän eri sävyt sekä keltainen, oranssi ja neonvärit yleensä.



Kuvio 5. Lasten lempivärit kyselytutkimuksen mukaan. Eri sävyt yhdistetty.

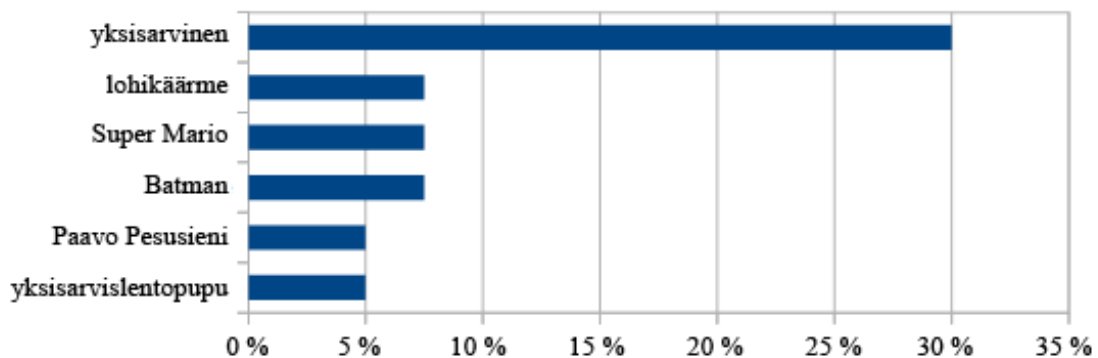
Suosituimmat yksittäiset sävyt (liitteessä 2) olivat kuitenkin hieman yllättäen musta ja valkoinen. Näiden jälkeen suosituimmat yksittäiset sävyt olivat järjestyksessä vaaleanpunainen, turkoosi, sininen ja pinkki. Näiden tulosten perusteella voidaan todeta, että lasten maut ovat muuttuneet jo hieman aikuisemmiksi, eikä suuri määrä erilaisia kirkkaita värejä enää välttämättä vetoa tähän kohderyhmään.

Toisessa kysymyksessä lapset saivat kertoa nykyisistä harrastuksistaan tai nimetä harrastuksen, jota he haluaisivat harrastaa (ks. kuvio 6). Lähes kaikki suosituimmat nimetyt harrastukset olivat urheilullisia joukkuelajeja. Kaksi suosituinta harrastusta, lentopallo ja pesäpallo löytyivät toistuvasti molemmat samoilta lapsilta, kun taas jääkiekko oli useimmiten lapsen ainut harrastus. Muita verrattain suosittuja harrastuksia olivat ratsastus, jalkapallo ja salibandy. Myös uintia, musiikkia, partiota, painia, cheerleadingia ja parkouria harrastavat lapset ilmoittivat usein harrastavansa useampaa eri aktiviteettia.



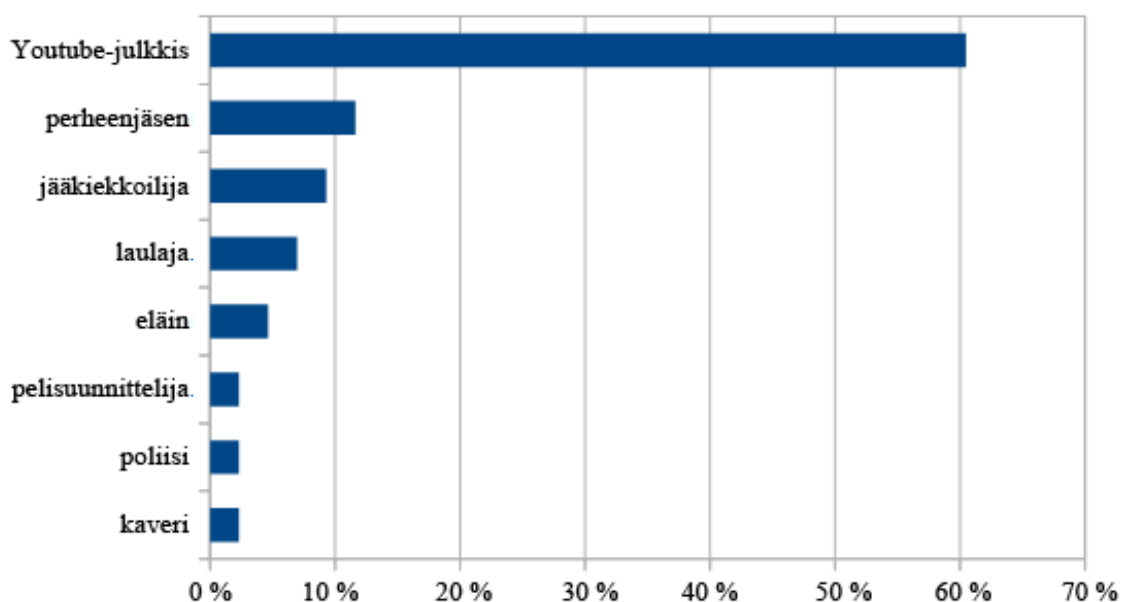
Kuvio 6. Lasten suosituimmat harrastukset kyselytutkimuksen mukaan.

Kysymyksessä numero kolme kysyttiin lasten lempihahmoja, joista selväksi voittajaksi osoittautui yksisarvinen. Kuvio 7 voidaan todeta, että toiseksi suosituimmat hahmot lohikäärme, Super Mario ja Batman hävisivät yksisarviselle huomattavasti. Myös Paavo Pesusieni ja yksisarvislentopupu saivat enemmän kannatusta kuin loput kyselyssä esille tulleet yhden kerran äänestetyt hahmot. Ainoa selvästi esille noussut hahmotrendi oli siis yksisarviset.



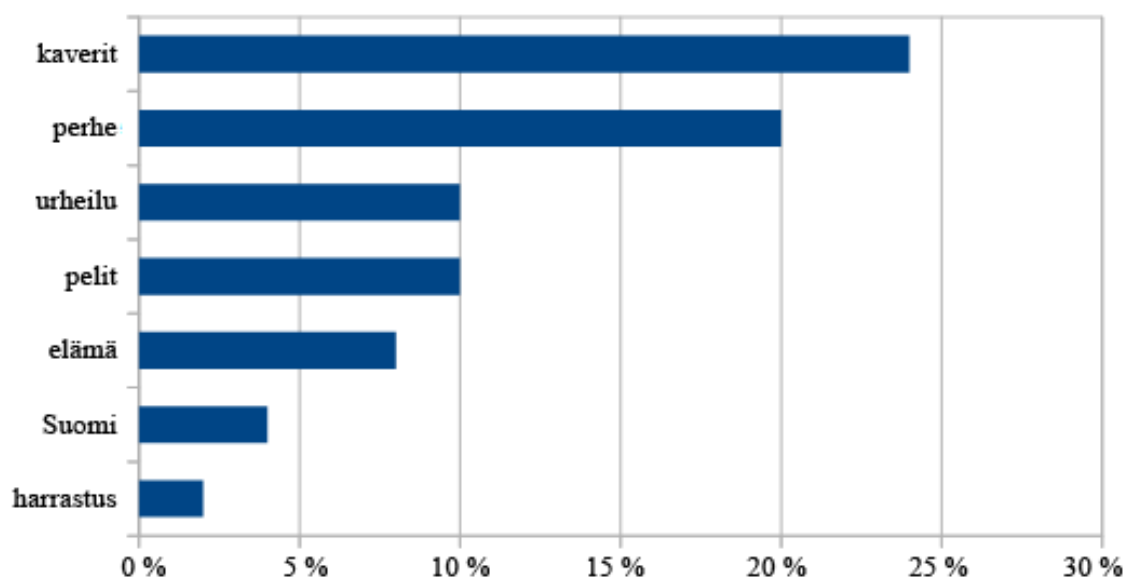
Kuvio 7. Lasten suosituimmat lempihahmot kyselytutkimuksen mukaan.

Neljännessä kysymyksessä puolestaan tiedusteltiin lasten idoleita (ks. kuvio 8). Selvästi eniten idoleina mainittiin erilaisia Youtube-julkkaisia, kuten Roni Back, Lakko ja Dita Bibiana. Youtube-julkkaisvastauksia oli jopa enemmän kuin muita vastauksia yhteensä. Tätä selittää se, että ”tubettajat” ovat keskenään hyvin erilaisia; jotkut tekevät videoita peleistä, haasteista tai musiikista, toiset askartelusta, terveellisyydestä tai elämästä yleensä. Jokaiselle katsojalle löytyy siis varmasti itseään kiinnostava vaihtoehto. Muita suosittuja idoleita olivat perheenjäsenet, jääkiekkoilijat ja laulajat. Myös eläimet, pelinsuunnittelijat, poliisi ja kaverit saivat hieman ääniä.



Kuvio 8. Lasten idolit kyselytutkimuksen mukaan.

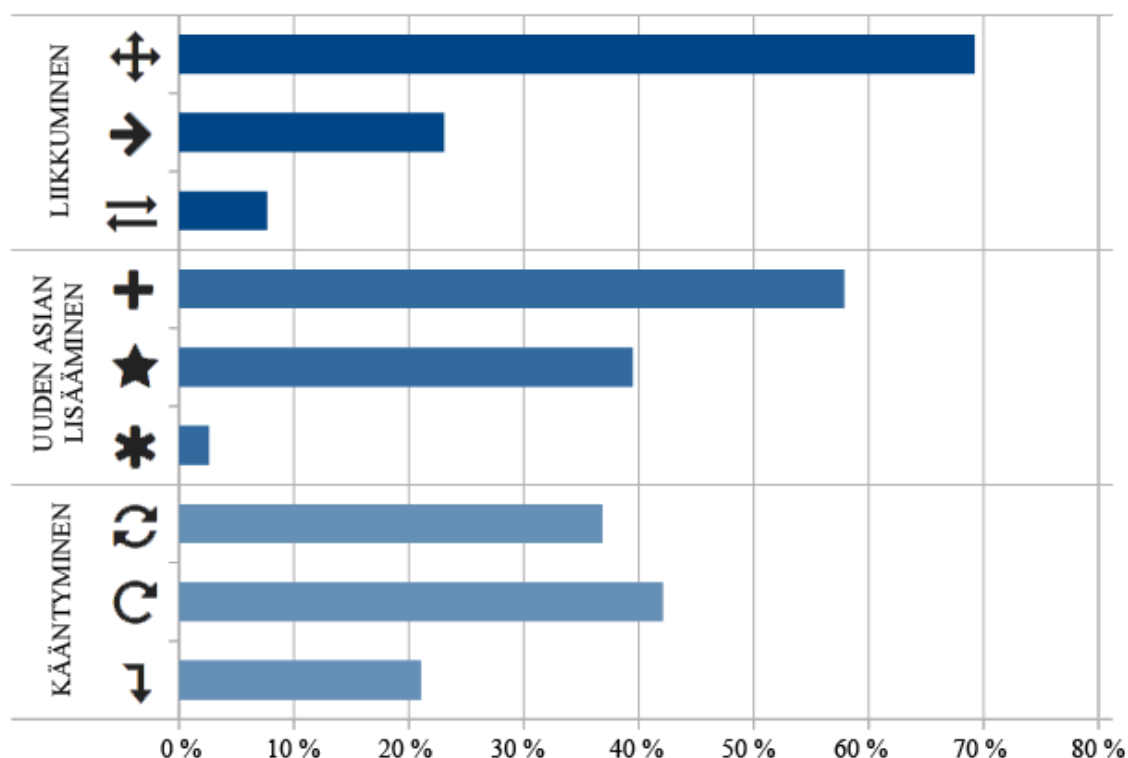
Viimeinen avoin kysymys koski parasta asiaa maailmassa. Kuvion 9 mukaisesti suosituimmat vastaukset tähän olivat kaverit ja perhe. Seuraavaksi suosituimpia olivat eri urheilulajit ja videopelit. Useampia ääniä saivat myös elämä yleensä, Suomi sekä muut harrastukset. Lisäksi kysymykseen vastattiin paljon vain yksittäisiä ääniä saaneita asioita, kuten esimerkiksi eläimet, yksisarviset, ruoka, teknologia ja vintage.



Kuvio 9. Lasten suosituimmat vastaukset parhaaseen asiaan maailmassa.

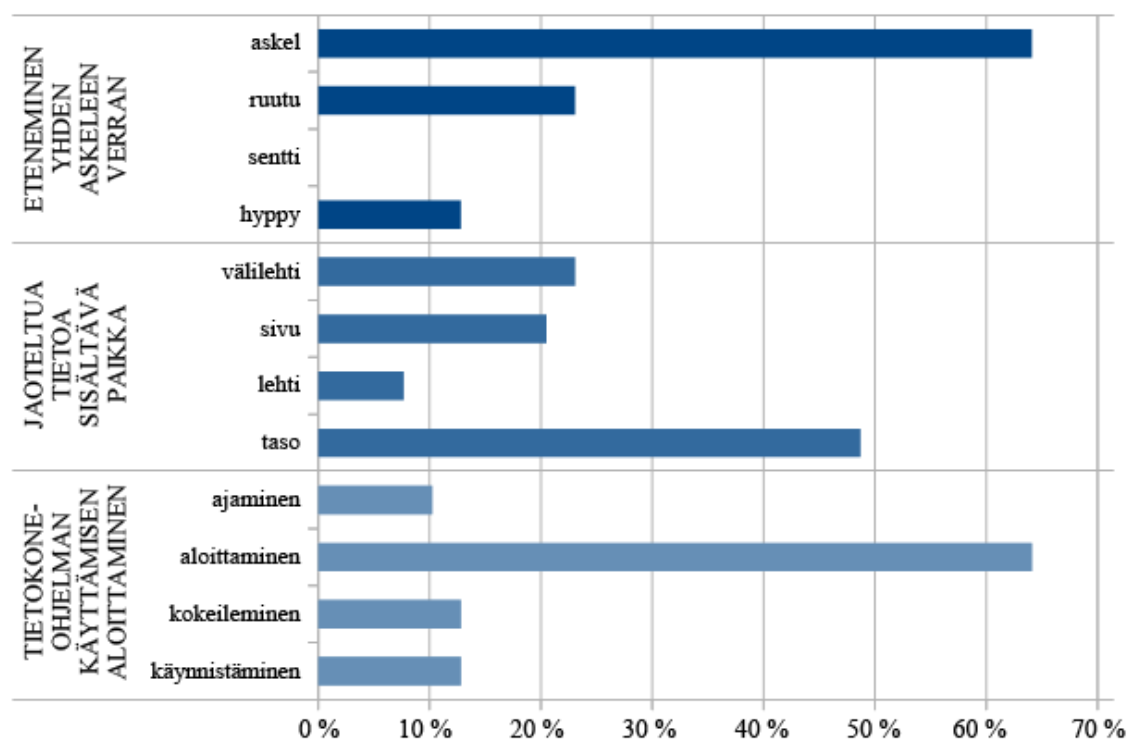
Lapset, jotka vastasivat harrastuksikseen lentopallon ja pesäpallon, nimesivät lempivärikkeseen toistuvasti vaaleita värejä ja punaisen sävyjä. Idolikseen he mainitsivat usein Youtube-julkikkusen. Jääkiekon harrastajat puolestaan saattoivat antaa muihin kysymyksiin videopelisiin liittyviä vastauksia ja nimesivät lempivärikkeseen usein sinisen sävyjä. Vaikka tällainen korrelaatio eri vastausten välillä oli osittain vähäistä, oli se silti selvästi huomattavissa.

Kuudennessa kysymyksessä lapsia pyydettiin ympyröimään ikoneja, jotka heidän mielestään kuvaavat parhaiten kolmea eri toimintoa (ks. kuvio 10). Toiminnot olivat *liikkuminen*, *uuden asian lisääminen* ja *kääntyminen*. Selkeimmän vastauksen näistä kolmesta toiminnoista sai *liikkuminen*, johon lähes 70 % vastanneista valitsi neljään suuntaan osoittavan nuolen. Toiseksi eniten äänestettiin yksisuuntaista ja vähiten kaksisuuntaista nuolta. *Uuden asian lisäämiselle* kuvaavimmaksi ikoniksi äänestettiin plusmerkki, mutta toiseksi tullut tähti ei jäänyt kauas taakse. Vähiten ääniä kaikista sai asteriski. Viimeisistä ikoneista *kääntymistä* kuvaavin oli lasten mielestä oikealle osoittava pyöreä nuoli. Se voitti molempiin suuntiin osoittavan pyöreän nuolen kuitenkin vain noin viidellä prosenttiyksiköllä. Alas osoittava kulmanuoli sai näistä kolmesta vaihtoehdosta vähiten ääniä.



Kuvio 10. Kyselytutkimuksen mukaan lasten mielestä kuvaavimmat ikonit.




Seitsemännessä tehtävässä pyydettiin vastaavasti ympäröimään kuvaavin sana kolmelle asialle; *etenemiselle yhden ruudun verran*, *jaoteltua tietoa sisältävälle paikalle* ja *tietokoneohjelman käyttämisen aloittamiselle*. Tulokset ovat näkyvissä kuviossa 11. *Etenemistä yhden ruudun verran* kuvasi lasten mielestä eniten sana askel. Seuraavaksi eniten ääniä saivat sanat ruutu ja hyppy, kun taas sanaa sentti ei äänestetty kertaakaan. Yllättävin tulos oli, että *jaoteltua tietoa sisältävälle paikalle* kuvaavimmaksi sanaksi äänestettiin taso. Muut vaihtoehdot (välilehti, sivu ja lehti) saivat prosentuaalisesti huomattavasti vähemmän ääniä. Koko kyselytutkimuksen viimeisenä tuloksena saatiin suosituin sana *tietokoneohjelman käyttämisen aloittamiselle*: aloittaminen. Tämä kysymys oli asetettu verrattain huonosti, sillä se ei vastannut täysin haettua asiaa ja se jo itsessään johdatteli antamaan tietyn vastauksen. Todennäköisesti ohjelmistokehittäjien keskuudessa olisi vastattu aloittamisen sijasta muita vaihtoehtoja (ajaminen, kokeileminen ja käynnistäminen) huomattavasti enemmän.



Kuvio 11. Kyselytutkimuksen mukaan lasten mielestä kuvaavimmat sanat.

Saadut kyselytutkimuksen tulokset on esitetty yhteenvetona taulukossa 2, jossa mainitaan jokaisen kysymyksen kohdalla joko tärkeimmät esiin nousseet trendit tai vaihtoehto, joka oli suosituin. Tuloksia hyödynnetään itse ohjelmaa suunniteltaessa niin paljon kuin on mahdollista. Esimerkiksi lasten mielestä kuvaavimmat ikonit ja sanat tulevat käyttöön. Avoimissa kysymyksissä kaikista selvimmin esiin nousseita trendejä olivat yksisarviset, Youtube-julkkikset, lentopallo ja pesäpallo. Muita trendejä olivat jääkiekko, ystävät, perhe ja hahmot, joilla on supervoimia. Yksittäisistä väreistä pidettiin selvästi eniten mustasta ja valkoisesta, sekä punaisen ja sinisen hillityistä sävyistä, kuten vaaleanpunaisesta ja turkoosista.

Taulukko 2. Kyselytutkimuksen tulosten yhteenveto.

	esiin noussut trendi/suosituin vaihtoehto
lempiväri	musta, valkoinen, punaisen ja sinisen hillityt sävyt
harrastus	lentopallo, pesäpallo, jääkiekko
lempihahmo	yksisarvinen, lohikäärme, superhahmo
idoli	Youtube-julkkikset, perheenjäsenet, jääkiekkoilijat
paras asia maailmassa	kaverit, perhe
ikoni liikkumiselle	
ikoni uuden asian lisäämiselle	
ikoni kääntymiselle	
ilmaus etenemiselle	askel
ilmaus paikalle, jossa jaoteltua tietoa	taso
ilmaus ohjelman aloittamiselle	aloittaminen

4 OHJELMAN SUUNNITTELU JA PROTOTYYPPI

Tässä luvussa esitellään suunnitelma uudesta graafisesta ohjelmointiympäristöstä perusopetukseen. Suunnittelu toteutetaan kyselytutkimuksen tulosten sekä käytettävyysskirjallisuudesta saatujen ohjeiden perusteella. Myös graafisen suunnittelun ohjeisiin kiinnitetään huomiota. Suunnitelmaa selventää prototyypistä otetut kuvakaappaukset. Lopussa käydään vielä läpi lyhyesti prototyypin suunnittelua koskevia aiheita.

4.1 Ohjelman suunnittelu

Ohjelmaa suunniteltaessa tulee aina muistaa ottaa huomioon kohderyhmä, varsinkin jos suunnittelijat eivät itse kuulu siihen. On helppo ajatella, että koska aikuiset ovat olleet itse lapsia, he muistavat ja tietävät kuinka lapset ajattelevat. Kuitenkin lasten ajatusmallit ja tietotekniset taidot eroavat aikuisten vastaavista merkittävästi. Näitä eroja kasvattaa entisestään uusi diginatiivien sukupolvi, joka on tottunut käyttämään vastaavaa teknologiaa jo pienestä pitäen (Gallavin 2015). Lapset nykyään eivät siis ole samanlaisia kuin suunnittelijat muistavat itse joskus olleensa. Tässä alaluvussa esitellään ohjelman suunnittelussa esille nousseita osa-alueita käyttäen apuna myös osaa Nielsenin (1995) kymmenestä käytettävyyssheuristiikasta käyttöliittymälle.

4.1.1 Ohjelman nimi

Graafisen ohjelmointiympäristön nimeksi valittiin Koodikoira. Se on selkeä, korostaa ohjelman suomalaista alkuperää, sekä antaa jo itsessään viittauksen siitä, mitä ohjelmalla voi tehdä. Lisäksi se tarjoaa mahdollisuuden luoda erilaisia käyttäjiin vetoavia koirahahmoja, joiden kautta tehtävät ja ohjeet kommunikoidaan käyttäjille. Prototyyppiä varten luotiin kyselytutkimuksen vastausten trendien pohjalta kaksi koirahahmoa; Raketti ja Yasmin. Kun ohjelman käyttö aloitetaan, se avautuu aloitusnäkyymään (kuva 4). Aloitusnäkyymässä käyttäjät valitsevat haluamansa näistä kahdesta hahmosta, joka sitten toimii heidän ohjeistajanaan.



Kuva 4. Koodikoiran aloitusnäky, jossa käyttäjät valitsevat toisen hahmoista.

4.1.2 Ohjelman värit

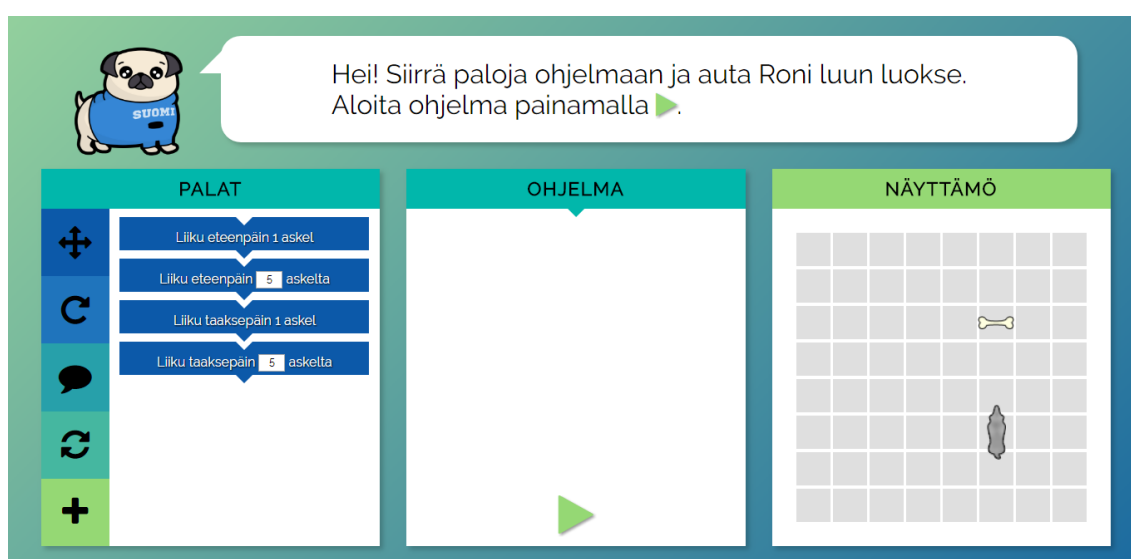
Vuoden 2018 graafisen suunnittelun trendien (Cann 2017; Cass 2017; Triner 2018; McCready 2018) mukaisesti ohjelmassa käytetään kirkkaita ja eloisia värejä, liukuvärejä ja puolilitteää ulkoasua. Kuten kyselytutkimuksessa todettiin, lasten lempivärit olivat jo hieman aikuistuneet, eikä suuri määrä erisävyisiä kirkkaita värejä enää välttämättä vetoa tähän kohderyhmään. Käytettävät värit eivät kuitenkaan ole lapsellisen kirkkaita, vaan nimenomaan eloisia. On tärkeätä, ettei käyttöliittymä ole kohderyhmälle liian lapsellisen oloinen, sillä lapset eivät tavallisesti pidä sellaisesta.

Koska kyselytutkimuksesta huomattiin mustan ja valkoisen jälkeen kaksi väritrendiä, päätettiin ohjelmaa varten luoda kaksi eri värimaailmaa. Värimaailma vaihtuu hahmon valinnan mukaan. Raketin värimaailmassa on sinistä, turkoosia ja vihreää (kuva 5), kun taas Yasminin vastaava on vaaleanpunainen, pinkki ja violetti (kuva 6). Vaikka suosituimmat yksittäiset lempivärit olivat musta ja valkoinen, ei se tarkoita sitä, että mustavalkoinen käyttöliittymä olisi parhaiten lapsiin vetoava. Tästä syystä käyttöön valittiin seuraavaksi suosituimmat värisävyt.

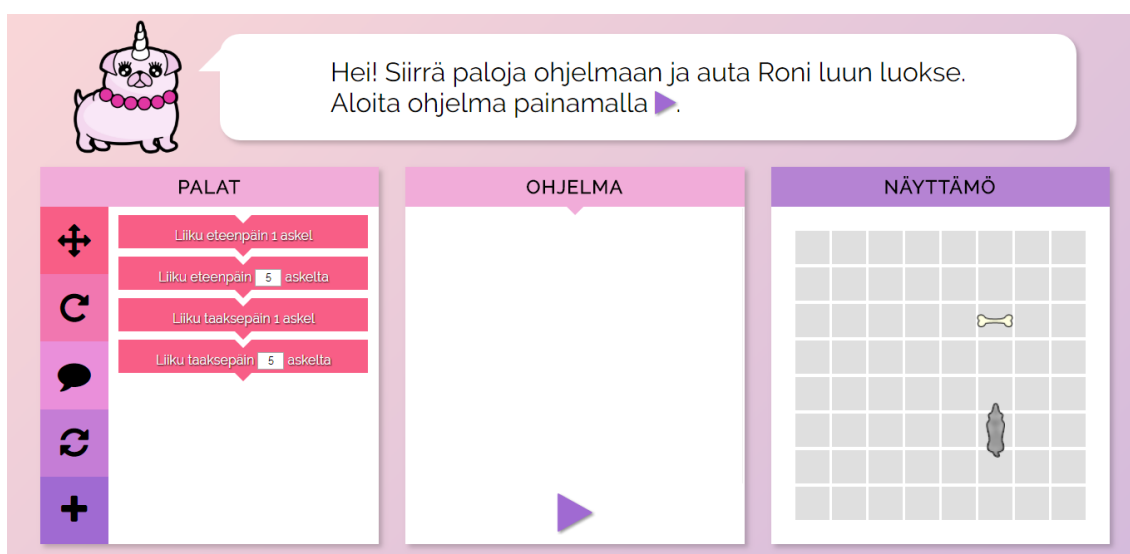
4.1.3 Minimalistinen suunnittelu & standardit

Ohjelmasta on jätetty pois kaikki ylimääräinen, ja esillä on vain tarpeellisia elementtejä. Asioiden tarpeellisuus tulee kuitenkin pohtia nimenomaan loppukäyttäjän kannalta; vaikka ohjelman suunnittelijoille ja kehittäjille tarpeellisia elementtejä olisi suuria määriä, ei loppukäyttäjät aina ajattele samalla tavalla (Duggirala 2016). Nielsenin (1995) mukaan jokainen ylimääräinen elementti vähentää oikeasti tarpeellisen tiedon suhteellista havaittavuutta.

Myös asioiden esittämiseen käytetään minimalistista ilmaisutapaa. Eri asioita kuvaavimmat ikonit valittiin sekä kyselytutkimuksen tulosten perusteella että yleisiä standardeja mukaillen. Käytössä on siis joko ikoneja, joita on jo totuttu käyttämään (kuten *toista*-ikoni), selkeästi toimintaansa kuvaavia (*puhekupla*- ja *roskakori*-ikonit) tai kyselytutkimuksen mukaan lasten mielestä parhaiten kuvaavimpia (*liiku*-ikoni). Suurin osa ikoneista on näkyvillä kuvassa 5, mutta esimerkiksi *roskakori*-ikoni on näkyvillä vasta kuvassa 7. Mikäli ikoni, elementin muoto tai muu ilmaisu ei ole itsessään riittävän ymmärrettävä, lisätietoa ja ohjeita elementin käytöstä saa näkyviin pitämällä hiirtä sen yläpuolella.



Kuva 5. Ensimmäisen tehtävän alkutilanne. Käytössä Raketin väriteema.



Kuva 6. Ensimmäisen tehtävän alkutilanne. Käytössä Yasminin väriteema.

4.1.4 Muistamisen sijaan tunnistus & käytön tehokkuus

Ikoneiden käyttö myös suosii asioiden tunnistamista niiden muistamisen sijaan. Yksinkertaisista kuvaavista ikoneista sekä otsikoista on helppo tunnistaa eri elementtien toiminnot ja tehtävät, vaikka niitä ei ulkoa muistaisikaan. Tämä kuormittaa huomattavasti vähemmän käyttäjien työmuistia. Tunnistamista auttavat myös ohjetekstit, jotka saa näkyville pitämällä hiirtä elementin päällä, sekä koirahahmon antamat ohjeet. Kaikki ohjeet saa tarvittaessa luettua uudestaan milloin tahansa. Kokeneemmat Koodikoiran käyttäjät eivät tarvitse kaikkia näitä ohjeita, ja siten voivat tehokkaasti käyttää ohjelmaa ilman, että pysäyttäsivät hiirtään jatkuvasti elementtien päälle.

4.1.5 Tosielämän vastaavuus

Käytettävät sanat ja ilmaisut on pyritty muodostamaan vastaamaan loogisesti tosielämän vertailukohtia. Sanat kuten *palat* ja *näyttämö* eivät sinänsä ole tyypillisesti käytössä ohjelmoinnissa, mutta kuvaavat erinomaisesti tarkoitusta sanan takana. Kuten palapeleissä, myös Koodikoirassa rakennetaan kokonaisuus yksittäisistä paloista, jotka sopivat toisiinsa vain tietyllä tavalla. Näyttämöllä puolestaan voidaan nähdä toiminnan

tulos. Turhat asiat taas heitetään roskakoriin (kuva 7). Käytettävät ilmaukset ovat nimenomaan lasten tosielämästä, sillä Lyonlaisin (2017) mukaan eri käyttäjät kommunikoivat eri tavalla; Koodikoirassa pyritään käyttämään juuri sen kohderyhmän kieltä.



Kuva 7. Palojen poistaminen tapahtuu vetämällä ne *roskakori*-ikonin päälle.

4.1.6 Johdonmukaisuus

Kaikki ohjelman elementit, joista voin painaa, suurenevat käyttäjän viedessä hiiren niiden päälle. Tämä selkeyttää huomattavasti sitä, mistä käyttäjä voi painaa ja mistä ei. Toimintamalli jatkuu johdonmukaisesti läpi koko ohjelman ja pätee jokaiseen painettavissa olevaan elementtiin. Koodikoirassa myös käytetään yhdestä asiasta aina samaa sanallista tai visuaalista ilmaisua. Esimerkiksi näyttämöön viitataan aina sanalla *näyttämö*, eikä esimerkiksi *piha*, *toistoikkuna* tai muu vastaava

Johdonmukaisuus näkyy myös graafisten elementtien asettelussa, etäisyyksissä toisiinsa sekä väreissä. Esimerkiksi *toista*-ikoni sekä *uuden asian lisääminen* -välilehti ovat saman värisiä näyttämön otsikon taustan kanssa ilmaistakseen yhteyden näiden

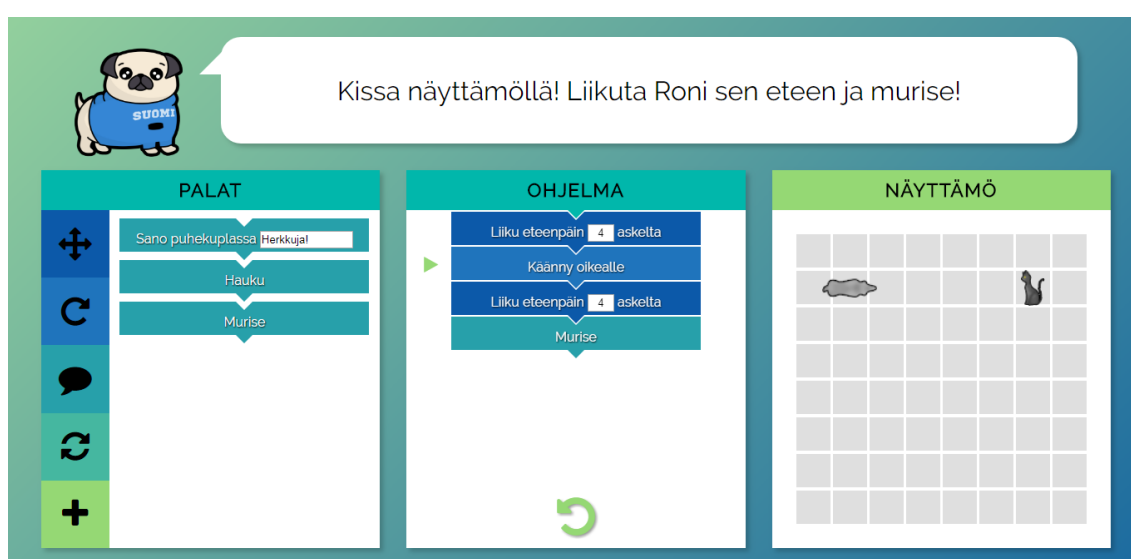
elementtien välillä; uusi asia, kuten esimerkiksi luu, lisätään näyttämölle ja *toista-*ikonista painettaessa näyttämöllä alkaa tapahtumaan asioita.

4.1.7 Käyttäjän vapaus

Koodikoira tarjoaa käyttäjilleen vapauden suorittaa tehtävät haluamallaan tavalla. Ei siis ole vain yhtä tiettyä palojen järjestystä, jolla tehtävät saisi suoritettua. Kuten ohjelmoinnissa yleensäkin, haluttuun tavoitteeseen voi päästä usealla eri tavalla. Käyttäjät voivat myös halutessaan kuunnella tehtävänannon uudelleen milloin tahansa uudestaan painamalla valitsemaansa koirahahmoa.

4.1.8 Järjestelmän näkyvä tila

Kun käyttäjän rakentama ohjelma ajetaan, järjestelmä osoittaa sitä aina palaa, jonka toistaminen on kullakin hetkellä menossa. Näin käyttäjä tietää missä kohdassa ohjelmaa on mahdollisesti virhe, milloin ohjelman toistaminen loppuu tai miksi järjestelmä teki kuten se teki. Kuvassa 8 ohjelman toistaminen etenee kuvakaappaushetkellä oikealle kääntävän palan kohdassa.



Kuva 8. Käynnissä oleva ohjelma.

4.1.9 Virheiden korjaaminen ja estäminen

Koska Koodikoira on kehitetty opetustarkoitukseen, on oletettavaa että virheitä tapahtuu verrattain usein. Tästä syystä virheiden korjaaminen on osa järjestelmän normaalia käyttöä. Vaikka paloista rakennettu ohjelma ei tekisikään asioita yhtään tehtävänannon mukaisesti, tai täyttäisi tehtävänannon jopa lähes oikein, ohjelma kelataan aina alkuun ennen seuraavaa toistokertaa. Kelauspainike ilmestyy *toista*-painikkeen tilalle kuvan 8 mukaisesti, joten järjestelmän virheellinen käyttö on estetty jo ennen kuin sitä tapahtuu. Tämä kaikki myös kehittää ohjelmoinnillista ajattelutapaa, jossa ohjelmaa kokeillaan hyvin pitkälle uudestaan ja uudestaan aina alusta.

4.2 Prototyyppi

Koodikoirasta päätettiin luoda käytettävyydestiä varten hi-fi prototyyppi, sillä Adiseshiahin (2016) mukaan sellaisten esittäminen ei-suunnittelijoille on tyypillisesti järkevää. Testikäyttäjien ollessa lapsia heidän ei-suunnittelijuutensa korostuu entisestään. Hi-fi prototyypeilla eli korkean todenmukaisuuden prototyypeilla tarkoitetaan lähes lopullisen ulkonäön ja toiminnallisuuden omaavia järjestelmiä. Hi-fin vastakohtana ovat lo-fi (matalan todenmukaisuuden) prototyypit, joissa ei ole minkäänlaista toiminnallisuutta, vaan ovat esimerkiksi vain kynällä paperille piirrettyjä (Adiseshiah 2016). Pernice (2016) kertoo korkean todenmukaisuuden prototyyppien hyväksi puoliksi myös niiden nopeamman ja realistisemmän vuorovaikutuksen sekä mahdollisuuden testata graafisia elementtejä. Hän korostaa vielä todenmukaisen ulkoasun auttavan testikäyttäjiä toimimaan kuten oikeassa käyttötilanteessa.

Koodikoiran prototyypillä on jo lopulliset valikkojen ja ohjelmointipalojen toiminnallisuudet sekä graafiset elementit. Tietenkin täytyy muistaa, että prototyypin käytettävyydestissä nousee todennäköisesti esille parannettavia asioita. Koodikoira ominaisuuksista testattavasta prototyypistä puuttuvat ainoastaan tehtävän suorittamisen jälkeinen automaattinen siirtyminen seuraavaan tehtävään sekä jotkut ohjelmointipalat (kuten toisto- ja jos-rakenteet).

5 KÄYTETTÄVYYSTESTAUS

Käytettävyyden ensimmäinen sääntö on ”tunne käyttäjäsi” (Norman 2005). Tämä ei kuitenkaan ole aina niin yksinkertaista, sillä käyttäjät voivat olla hyvin erilaisia keskenään ja suunnittelijoihin verrattuna, ja loppukäyttäjien maailma voi olla suunnittelijoille täysin tuntematon. Kuitenkin käytettävyys on niin suuri tekijä johtamassa käyttäjäytyvyyteen, että siihen tulisi panostaa suuressa määrin. Norman kuitenkin jatkaa, ettei käyttäjän tunteminen riitä, vaan huomiota tulisi kiinnittää jopa enemmän käyttäjien suorittamien toimintojen tuntemiseen ja niiden suunnitteluun. Nielsen (2001) toteaaakin, ettei käyttäjiä tulisi itse asiassa kuunnella, vaan tietoa tulisi kerätä havainnoimalla heidän toimintaansa. Tähän tarkoitukseen on kehitetty käytettävyystestaus.

Tässä luvussa käydään läpi käytettävyystestausta ja kiinnitetään huomiota sen erityispiirteisiin testihenkilöiden ollessa lapsia. Lisäksi käydään läpi ohjeita testauksen suunnitteluun, läpivientiin ja analysointiin. Tämän jälkeen käytettävyystestauksen työkaluista määritellään ääneenajattelutekniikka ja paritestaus.

5.1 Käytettävyystestaus teoriassa

Käytettävyystestauksella tarkoitetaan tutkimusta, jossa tarkastellaan käyttäjiä suorittamassa todellisia tai todellisen kaltaisia tehtäviä testattavalla tuotteella (Barnum 2011: 13). Käytettävyystestauksesta puhuttaessa se voidaan helposti sekoittaa käytettävyystutkimukseen yleensä. Käytettävyystutkimuksen alle kuuluu kuitenkin pelkän käytettävyystestauksen lisäksi paljon muitakin työkaluja, kuten esimerkiksi heuristinen arviointi sekä erilaiset läpikäynnit ja lajittelut. (Rubin & Chisnell 2008: 16–21.)

Käytettävyystestaus on siis yksi menetelmä käytettävyyden arviointiin. Sen tarkoituksena on löytää järjestelmästä ongelmakohtia, joita todelliset käyttäjät oikeasti kohtaavat käyttäessään testattavaa järjestelmää. Havaitut ongelmat tulevat sitten

ratkaista järjestelmää jatkokehitettäessä; lopullisena tarkoituksena on siis testattavan järjestelmän käytettävyyden parantaminen. Hyvänä puolena käytettävyydestestauksessa on sen vastaaminen Normanin (2005) periaatteeseen käyttäjien ja heidän suorittamiensa toimintojen tuntemisesta sekä (Nielsenin (2001) ohjeeseen käyttäjän havainnoinnista. Lisäksi saadaan tietoa järjestelmän osista, joiden käytettävyys on jo testaushetkellä hyvä. Toisaalta käytettävyydestestaus on verrattain vaativaa, sillä sen suunnittelu, toteutus ja analysointi vie huomattavan paljon aikaa ja rahaa. (Koskinen 2005: 187–188.)

Käytettävyydestestaukset voidaan jakaa kahteen joukkoon niiden toteutusajankohdasta ja tarkoituksesta riippuen. Formattiivista testausta suoritetaan tuotekehityksen aikana ja sen tarkoituksena on löytää tuotteesta vikoja ja edelleen korjata ne. Tyypillisesti formattiivinen käytettävyydestestaus toteutetaan pienillä ja usein toistettavilla testeillä. Summatiivisen käytettävyydestestauksen tarkoituksena on puolestaan vahvistaa valmiin tuotteen soveltuvuus tehtäväänsä, tai määrittellä vertailukohta tulevaisuutta varten. Summatiivinen testaus toteutetaan aina valmiille tuotteelle, ja se vaatii yleensä suurempia testausmääriä tilastollisen pätevyyden varmistamiseksi. (Barnum 2011: 14.)

Dumas ja Redishin (1999) mukaan mikäli käytettävyydestestausta ei suorita todellisilla käyttäjillä, ei tiedä miten todelliset käyttäjät oikeasti toimivat. Siitä syystä he, sekä monet muut (Gould & Lewis 1985: 300; Cao 2015; Mortensen 2018; Usability.gov 2018) suosittelevat käytettäväksi testikäyttäjiä, jotka ovat joko todellisia nykyisiä tai tulevia käyttäjiä järjestelmälle. Mikäli testikäyttäjien taitotaso eroaa todellisten loppukäyttäjien taidoista, järjestelmälle voi tehdä tarpeettomia muutoksia (Dumas & Redish 1999).

Tavallisesti käytettävyydestestauksessa on kerrallaan läsnä yksi testikäyttäjä ja yksi tai useampi valvoja. Testikäyttäjälle annetaan mahdollisimman hyvin järjestelmän todellisia käyttötilanteita vastaavia tehtäviä, joiden suorittamista valvojat tarkkailevat. Aineiston saamiseksi testitilanteesta voidaan tehdä muistiinpanoja ja se voidaan videokuvata. Testin jälkeen aineistoa analysoidaan. Useimmiten käytettävyydestestauksia suoritetaan peräkkäin useampia eri testihenkilöillä. (Dumas & Redish 1999: 23–24; Koskinen 2005: 188.)

5.2 Käytettävyydestestauksen suunnittelu

Ennen käytettävyydestestauksen suorittamista se täytyy suunnitella tarkkaan. Ensimmäinen pohdittava asia on, onko käytettävyydestestauksen järjestämiselle todellista syytä ja onko se edes oikea työkalu kyseisessä tilanteessa. Mikäli käytettävyydestestaus päätetään toteuttaa, seuraavaksi tulee miettiä testaukselle selkeät tavoitteet. Tavoitteita määriteltäessä tulee ottaa huomioon käytettävissä olevat resurssit. (Koskinen 2005: 189.) Resursseihin nähden liian laajan tutkimuksen tekeminen laskee yleisesti ottaen laadullisten tulosten tasoa (Loranger 2016).

Kun käytettävyydestestauksella on selkeät tavoitteet, tulee miettiä minkälaisia tehtäviä testihenkilöiden tulisi suorittaa, että saataisiin vastauksia testauksen tavoitteisiin. Tavoitteeseen vastaamisen lisäksi hyvä testitehtävä vastaa todellista tilannetta. Mikäli testihenkilöt käyttävät järjestelmää kuten tavallista, myös ilmenevät ongelmat ovat todellisia. (Nielsen 1994; Dumas & Redish 1999: 23–24; Koskinen 2005: 189–191.)

Koskisen (2005: 191) mukaan testitehtävien tulisi olla yksikäsitteisiä ja kieleltään ymmärrettäviä ja luontevia. Hän toteaa, että tehtävänanto voidaan antaa jopa pienen tarinan muodossa sen ymmärtämisen helpottamiseksi. Tehtävät eivät saa kuormittaa testihenkilöiden muistia, eivätkä itsessään antaa vastausta tai vinkkiä tehtävän suorittamiseen. (Koskinen 2005: 191.) Nielsenin (1993: 187) mukaan ensimmäinen testitehtävä tulee suunnitella niin helpoksi, että se on kaikkien osattavissa. Tämä purkaa jännitystä ja auttaa rentoutumaan.

Ennen lopullista käytettävyydestestausta olisi suotavaa suorittaa pienimuotoinen pilottitesti, jonka tarkoituksena on kokeilla testin toimivuutta. Pilottitesti auttaa hiomaan tehtävien sanallista muotoa, sekä antaa tietoa yksittäisten tehtävien ja koko testin suorittamisen kestosta. Olisi tärkeää, että mahdollisimman moni ongelma huomattaisiin nimenomaan pilottitestin aikana, että ne olisivat korjattuina varsinaista käytettävyydestestausta suoritettaessa. (Loranger 2016.)

Kuten testitehtävien kohdalla, myös testihenkilöiksi tulee valita mahdollisimman

todellisia käyttäjiä. Mikäli testihenkilöt eivät vastaa todellisia käyttäjiä, ei saada tietää miten todellisten käyttäjät oikeasti toimisivat järjestelmää käyttäessään. Tällaisissa tapauksessa käytettävyydestä voidaan pitää jopa täysin turhana. Liian kokeneet testihenkilöt eivät paljasta kaikkia käytettävyyso ongelmia, kun taas liian osaamattomat testihenkilöt saattavat haluta muutoksia, joita todellisuudessa ei edes tarvittaisi. (Dumas & Redish 1999: 23.)

Vaikka kaikkien käytettävyyso ongelmien löytämiseen tarvittaisiin arviolta viisitoista testihenkilöä, todellisuudessa käytettäväksi suositellaan vain noin viittä. Viidellä testihenkilöllä saa selville jopa viisi kuudesta, eli noin 85 % kaikista käytettävyyso ongelmista. Verrattaessa testihenkilöiden määrää testaukseen sijoitettuihin resursseihin, kustannustehokkaimmin tuloksia antaa juuri viisi testihenkilöä. (Nielsen 2000; Loranger 2016.) Nielsen (2000) jatkaa, että mikäli resursseja on käytettävissä enemmän, tulee otannan kasvattamisen sijasta panostaa mieluummin käytettävyystestauksen toistamiseen. Loranger (2016) puolestaan toteaa, että jos käytössä on katseentunnistusteknologiaa, tulee testihenkilöiden määrää kasvattaa, sillä katseentunnistusta voidaan pitää määrällisenä tutkimuksena, joka vaatii suuremman otannan.

Testiympäristön tärkeimmät ominaisuudet ovat sen turvallisuus sekä testihenkilön viihtyvyys (Dumas & Redish 1999: 276). Testiympäristön tulisi olla myös mahdollisimman todellisen käyttöympäristön kaltainen (Koskinen 2005: 191). Lorangerin (2016) mukaan testin valvojan olisi hyvä olla samassa tilassa testihenkilön kanssa, sillä siten on mahdollista tunnistaa helpommin kehonkielen hienovaraisia ilmauksia. Andrzejczak ja Liu (2010: 1258) kuitenkin selvittivät, ettei testin valvojan sijainnilla lähi- ja etätestauksen välillä ole vaikutusta testihenkilöiden stressitasoon.

5.3 Käytettävyystestauksen läpivienti

Käytettävyystestaukseen järjestäessä paikalle tulee saapua hyvissä ajoin ennen ensimmäisen testihenkilön saapumista. Näin ehtii tarkistaa huolellisesti kaikkien käytettävien laitteiden ja ohjelmistojen toiminnan, sekä varmistaa että kaikki tarvittavat lomakkeet ja

muistiinpanovälineet ovat tallessa. (Koskinen 2005: 192.) Koskinen (2005: 192) jatkaa, että Barnumin (2002) mukaan testihenkilöt eivät saisi päästää näkemään testin valmistelua, sillä se saattaa aiheuttaa heissä yleistä hermostuneisuutta. Käytettävyydesti tulisikin valmistella näkymättömissä ja käyttäjä päästää sisään vasta kaiken ollessa aivan valmista testin aloittamista varten. (Koskinen 2005: 192.)

Kun testihenkilöt saapuvat sisään, on tärkeää saada heidät tuntemaan itsensä tervetulleiksi ja rennoksi. Tämän voi saada aikaan esimerkiksi vitsillä tai muulla pinnallisella jutustelulla. Aluksi täytetään esitietokysely, jonka jälkeen testihenkilöille tulee selittää ja esitellä testijärjestelyt, kuten testipaikka ja laitteisto, tutkijat sekä testattava asia. Seuraavaksi kerrotaan, että testihenkilöiltä odotetaan ääneenajattelutekniikan käyttöä, selitetään mitä se tarkoittaa, ja annetaan siitä esimerkki. (Barnum 2011: 201–202; Morrison 2018.) Morrisonin (2018) mukaan on hyvä vielä muistuttaa, että testauksen kohteena on vain ja ainoastaan järjestelmä, eivät testihenkilöt. Hän jatkaa kehottamalla testihenkilöitä olemaan mahdollisimman rehellisiä sen sijaan, että olisivat kohteliaita järjestelmän tekijöitä kohtaan. Koskisen (2005: 193) mukaan testihenkilöitä tulee viimeiseksi vielä muistuttaa, että testin saa halutessaan lopettaa milloin tahansa.

Käytettävyydestin alettua testin järjestäjän toiminta ja käyttäytyminen on edelleen tärkeässä roolissa. Tähän Rubin ja Chisnell (2008: 201–211) antavat useita ohjeita. Tärkeintä on, että testin järjestäjän tulee suhtautua täysin objektiivisesti kaikkeen, mutta toisaalta pitää silti tilanne rentona. Testin järjestäjän tulisikin etukäteen päättää, että testin tuloksilla ei ole hänelle mitään väliä, vaan hän vain raportoi kaiken havaitsemansa. Jokaisen vuorovaikutuksen järjestäjän ja testihenkilön välillä pitäisi tapahtua vain tarvittaessa ja vaikuttamatta tuloksiin. Järjestäjän tulee myös olla tietoinen oman äänensävyensä ja kehonkielensä välittämistä viesteistä. Kuitenkin jos järjestäjä tekee jonkin virheen, tulee hänen yksinkertaisesti vain jatkaa eteenpäin. Ennen jatkamista seuraavaan tehtävään tulee kuitenkin vielä varmistaa, että testihenkilö on varmasti lopettanut edellisen tehtävän suorittamisen. (Rubin & Chisnell 2008: 201–211.)

He jatkavat vielä, että toimiessa useiden testihenkilöiden kanssa saman päivän aikana on

helppo unohtaa, että jokainen heistä on yksilö, joka tekee testiä ensimmäisen kerran. Aiemmin suoritettut testit eivät kuitenkaan saa vaikuttaa mitenkään toimintaan uuden testihenkilön kanssa. Testihenkilöä ei saa myöskään avustaa, jos hänellä ilmenee ongelmia, sillä ongelmatilanteiden itsenäinen ratkaisu on yksi tärkeä osa käytettävyyttä. Mikäli heitä autetaan jatkuvasti, käytettävyyso ongelmia ei päästä oikeastaan edes löytämään. Auttamisen tulee olla vasta viimeinen keino, jota käytetään vain mikäli käytettävyydestä ei pääse muuten jatkumaan. Testihenkilöiden auttamisella on aina vaikutusta tuloksiin. (Rubin & Chisnell 2008: 201–211.)

Kun käytettävyydestä on loppunut, testihenkilöiltä kerätään yleensä subjektiivista tietoa loppuhaastattelun muodossa. Haastattelu voi olla kirjallinen, suullinen tai näiden yhdistelmä, mutta sen olisi hyvä olla varsinaista testiä vapaamuotoisempi. Siinä voidaan käydä läpi esimerkiksi testihenkilöiden kohtaamia ongelmia, yleisiä mietteitä sekä mielipiteitä järjestelmän käytön hauskuudesta. Epämuodollinen keskustelu, kiittäminen ja testihenkilöiden tarjoaman avun merkityksen mainitseminen saavat heidät tuntemaan itsensä merkityksellisiksi. Vaikka seuraavat testihenkilöt olisivat jo tulossa, ei edellisiä tulisi siirtää alta pois lyhytsanaisesti ja tyyliä, ikään kuin heti kun heistä on saatu kaikki hyöty irti. (Koskinen 2005: 196.)

5.4 Käytettävyydestä tulosten analysointi ja esittämien

Käytettävyydestä testit ovat täysin turhia, ellei niistä saatua aineistoa ja tuloksia analysoida. Analysointi olisi hyvä suorittaa mahdollisimman pian testien jälkeen kun tapahtumat ovat vielä tuoreessa muistissa, mutta Dalrymple (2016) mukaan muistiin ei siltikään tule täysin luottaa. Sen sijaan analyysi tulisi perustaa mahdollisimman paljon äänitettyyn ja kuvattuun materiaaliin sekä muistinpanoihin, sillä nämä kuvaavat tapahtumien kulkua todenmukaisemmin. Dalrymple (2016) jatkaa vielä, että huomiota tulee myös kiinnittää mahdollisiin omiin ennakoasenteisiin, jotka ovat voineet vaikuttaa käytettävyydestä tai suoritettavassa analysointiprosessissa.

Yksi paljon käytetty tapa analysoida käytettävyydestä havaittuja käytettävyyss-

ongelmia on Nielsenin (1993: 103) kehittämä vakavuusluokittelu. Sen mukaan kaikki löydetyt käytettävyysoingelmat jaotellaan viiteen vakavuusluokkaan (0–4), joista numerolla neljä merkittävät ongelmat ovat kaikista suurimpia. Nämä ongelmat ovat katastrofaalisia, eikä sellaisia ongelmia tulisi koskaan jäädä julkaistavaan järjestelmään. Numerolla kolme merkittävät ongelmat ovat suuria, ja ne tulee korjata korkealla prioriteetilla. Numerolla kaksi merkittäviä ongelmia tulee korjata matalammalla prioriteetilla, sillä ne ovat jo huomattavasti pienempiä. Vakavuusluokan yksi käytettävyysoingelmat ovat lähinnä vain kosmeettisia, ja ne tulisikin korjata vain mikäli projektissa ilmenee ylimääräistä aikaa. Nollalla merkittävät ongelmat eivät puolestaan ole käytettävyysoingelmia ollenkaan. Havaittujen käytettävyysoingelmien jako näihin luokkiin suoritetaan niiden yleisyyden, toistuvuuden ja vaikutuksen perusteella. (Nielsen 1993: 103.)

Toinen käytettävyysoingestien analysointitapa on Rosenbergin (2018) esittelemä prosessi, jonka voidaan katsoa kattavan myös Nielsenin (1993: 103) vakavuusluokituksen. Rosenbergin prosessi jatkuu ongelmien korjausjärjestyksen määrittämisen jälkeen vielä kahdella askeleella, joista ensimmäinen on ratkaisujen kehittäminen. Jokaiselle käytettävyysoingelmalle tulisi kehittää useita eri ratkaisuvaihtoehtoja. Viimeisessä askeleessa kehitetyt ratkaisuvaihtoehdot priorisoidaan niiden tehokkuuden ja monimutkaisuuden perusteella. (Rosenberg 2018.)

Käytettävyysoingestauksen viimeisenä tuloksena tulee kirjoittaa käytettävyysoingestiraportti, joka esittelee käytettävyysoingestauksen lähtökohdat, testausprosessin lyhyesti sekä löydetyt käytettävyysoingelmat. Tämän raportin tarkoituksena on välittää tieto järjestelmän kehittäjälle tai muulle toimeksiantajalle, mutta oikein toteutettuna se myös parantaa käytettävyyden arvioinnin laatua. Raportissa esiteltävät käytettävyysoingelmat on välttämätöntä priorisoida, sillä ainoastaan raportointi suuresta määrästä erilaisia ongelmia ei johda käytettävyyden parantamiseen. Priorisoitujen ongelmien lisäksi erityisen tärkeää onkin esitellä yhdessä asiantuntijoiden kanssa pohdittuja ratkaisuja havaittuihin käytettävyysoingelmiin. Käytettävyysoingestiraportin itsensä tulisi olla helposti luettava, perusteleva ja rakentava. (Koskinen 2005: 199–200; Perälä 2005: 309.)

5.5 Käytettävyytestaus lasten näkökulmasta

Yleisesti ottaen ohjeet käytettävyytestaukseen lasten kanssa ovat samat kuin aikuis-tenkin kanssa. Lasten kanssa testattaessa tulee kuitenkin muokata muuttamia asioita testin suunnittelussa, testiympäristössä ja vuorovaikutuksessa testihenkilöiden kanssa. Että käytettävyytestauksesta olisi mahdollisimman paljon hyötyä, testihenkilöiden tulee tuntea olonsa mukavaksi, turvalliseksi ja tyytyväiseksi. Tämä tarkoittaa lapsille ja aikuisille hieman eri asioita. (Hanna, Risdén & Alexander 1997: 14; Dunn 2006.)

Haasteellisinta on ymmärtää kaikkia lapsen ominaisuuksia. Lapset kehittyvät koko ajan jokaisella kehityksen osa-alueella ja jokainen omassa tahdissaan. Iän perusteella ei siis voida olettaa mitä testihenkilöt osaavat, sillä lapset muuttuvat jopa päivittäin. Lisäksi lapset kehittyvät vielä jokainen omaan suuntaansa. Lapsen persoonallisuudella uskotaan olevan vaikutusta niin optimaalisissa testijärjestelyissä kuin myös käytettävyysongel-mien löytämisessä. Käytettävyytestaajien olisi siis suotavaa ymmärtää myös lasten kehityspsykologiaa. (Höysniemi 2005: 259–261.)

Hanna ym. (1997: 11–13) kertovat, miten käytettävyytestausta tulee muokata lapsille sopivaksi. Testiympäristön tulisi olla sopiva sekoitus aikuisten testiympäristöä ja leikkiin kutsuvaa tilaa. Leikki ei kuitenkaan saa olla ylikorostettua niin, että se vie huomiota pois varsinaisesta testistä. Testitilanteen aluksi valvojan tulee tutustua lapseen juttelemalla helpoista asioista, kuten kouluaineista tai harrastuksista. Tämä rauhoittaa ja helpottaa muuten mahdollisesti jännittävää tilannetta. Lapset ovat myös tottuneet kysymään kysymyksiä aikuisia enemmän, ja testin aikana he saattavat pyytää apua useasti. Koska tarkoituksena on testata kuinka lapset itse järjestelmää käyttäisivät, lapsille tulee usein vastata johdattelevasti kysymyksillä ”*mitä mieltä itse olet?*” tai ” *yritä keksiä tai arvata*”.

Dunn (2006) pohtii samoja asioita ja on yhtä mieltä Hanna ym. (1997: 11–13) kanssa, mutta korostaa esimerkiksi sanattoman viestinnän havainnoinnin tärkeyttä. Lapset eivät välttämättä saa sanottua mitä he oikeasti ajattelevat, sillä he voivat olla ujoja tai haluavat miellyttää testin valvojaa. Tästä syystä heidän huokauksensa, hymyt,

haukotukset ja muu kehonkieli ovat tärkeitä vihjeitä siitä, mitä he todella tuntevat. Lisäksi on tärkeätä, että lapset ymmärtävät tehtävänannon oikein. Tämän voi varmistaa pyytämällä lasta kertomaan omin sanoin hänen tavoitteensa kyseisessä tehtävässä. Lasta voi myös pyytää vähän ajan päästä toistaa tehtävänsä, jos hän näyttää unohtaneen sen.

Hanna ym. (1997: 14) kertovat lopuksi, että kaikkien ohjeiden tarkoitus on kuitenkin vain ottaa huomioon lasten kehitystaso. Tärkeimpiä muistettavia ovat selkeä ja yksinkertaisia sanoja käyttävä tehtävänanto sekä testauksen rytmitys lasten huomiojäteiden mukaan. Vaikka käytettävyydestä lapsilla vaatii lisäjärjestelyjä ja ylimääräistä huomiota, se on ainoa tapa saavuttaa oikeasti lapsiystävällisiä ja lasten hyväksymiä järjestelmiä. (Hanna ym. 1997: 14.)

Höysniemi (2005: 274) muistuttaa vielä, että järjestelmän menestymisen kannalta lasten maailmassa hauskuudella on suuri rooli. Tästä syystä käytettävyytutkimuksessa olisi suotavaa käyttää aikaa myös hauskuuden arviointiin. Hauskuuden arviointiin on kehitetty useita työkaluja, kuten Readin, MacFarlanen ja Casey'n (2002) kehittämä hymiömittari (kuva 9), jossa lapsi saa valita kuvaavimmat kasvot ensin odotetun ja sitten toteutuneen hauskuuden perusteella (Höysniemi 2005: 274).



Kuva 9. Hymiömittari (Read ym. 2002).

5.6 Ääneenajattelutekniikka ja paritestausta

Ääneenajattelutekniikaksi kutsutaan yksinkertaista tiedonkeruutekniikkaa, jossa testihenkilöä pyydetään puhumaan ääneen kaikki ajatuksensa tämän suorittaessa hänelle annettuja tehtäviä. Tekniikka on ainutlaatuinen kognitiivisen tiedon lähde, sillä se antaa reaaliaikaista dataa ihmisen sisäisistä tuntemuksista ja ajatuksista, jotka eivät näy ulospäin. Ääneenajattelu on siis tapa kerätä tietoa ja keino ymmärtää syvemmin ihmisen ajatusprosesseja. (Lewis & Rieman 1994: 83; Jaspers, Steen, van den Bos & Geenen 2004: 783–784.) Jaspers ym. (2004: 783) toteavat, että ääneenajattelutekniikka koostuu tutkimustyökaluna monen muun tekniikan tavoin tiedon keräämisestä ja sen analysoinnista. Vaikka tieto saadaankin kerättyä välittömästi, täytyy se silti merkitsevien tulosten saamiseksi analysoida jälkikäteen. Ääneenajattelua käytetään apuna tyypillisesti käytettävyydestäuksen yhteydessä (Ilves 2005: 209).

Nielsen (1993: 195) nimeää ääneenajattelutekniikan ehkä tärkeimmäksi yksittäiseksi työkaluksi käytettävyydestutkimuksessa. Tekniikalla on monia hyviä puolia; sillä esimerkiksi saa pelkkien ongelmakohtien löytämisen lisäksi selville myös syitä ongelmien taustalla. Lisäksi tekniikka soveltuu oikeastaan minkä tahansa järjestelmän arviointiin, ja sillä saa verrattain helposti suuren määrän laadullista aineistoa suhteellisen pieneltä määrältä testihenkilöitä. (Ilves 2005: 209.) Vaikka ei olisikaan ammattilainen ääneenajattelutekniikan käytössä, sitä on vaikea käyttää väärin. Tekniikka on myös äärimmäisen halpa. Muita hyviä puolia ovat sen helppo opittavuus, vakuuttavuus ja joustavuus. (Nielsen 2012.) Nielsenin (2012) mukaan joustavuus tarkoittaa ääneenajattelutekniikan sopivuutta mihin tahansa järjestelmän kehityksen vaiheeseen, mutta Ilveksen (2005: 209) mukaan parhaat tulokset saadaan, kun järjestelmä on riittävän hyvä käyttöä varten, mutta ei liian vaikeakäyttöinen tai monimutkainen.

Vaikka ääneenajattelutekniikka on paljon suositeltu, on sillä myös huonoja puolia. Van Someren, Barnard ja Sandberg (1994) toteavat esimerkiksi, ettei tekniikka sovi ihan aina käytettäväksi, sillä sen tarkoitus on kerätä kognitiivista tietoa, joka puolestaan ei aina vastaa haluttuihin kysymyksiin. Ääneenajattelutekniikka ei siis ole ihmelääke, joka ratkaisee kaikki ongelmat. Muita huonoja puolia sillä ovat testin valvojan mahdollinen

vaikutus testihenkilön lausuntoihin ja heidän kommenttiensa pohtiminen etukäteen. Testihenkilöt saattavat pohtia kommenttejaan vaikuttaakseen järkevämmiltä, vaikka tavoitteena on mahdollisimman käsittelemättömien ajatusten ilmaiseminen. Testitilanne on myös varsin epäluonnollinen, sillä harva selostaa ajatuksiaan ja tekemisiään jatkuvasti ääneen. (Nielsen 2012.)

Ilves (2005: 216) kertoo paritestauksen olevan Nielsenin (1993) mukaan muunnelma ääneenajattelutekniikasta. Siinä käytettävyydestiin osallistuu kaksi testihenkilöä samanaikaisesti, jotka keskustelevat järjestelmän käytöstä keskenään. Tämä tekee ääneenajattelusta paljon luonnollisempaa, sillä puhuminen onkin nyt yhteistä ongelmanratkaisua. Erityisen hyvin paritestaus sopii käytettäväksi lasten kanssa, sillä heidän saattaa olla vaikea ymmärtää kaikkia ääneenajattelutekniikan ohjeita. (Nielsen 1993; Ilves 2005: 206.) Paritestauksessa testihenkilöiden tulee olla keskenään melko samankaltaisia, ettei yhden dominoiva käytös vääristä toisen mielipiteitä tai jätä niitä kokonaan huomiotta (Carr 2016).

Carr (2016) listaa lisää paritestauksen hyviä puolia, mutta tuo esille myös huonoja. Toisen testihenkilön mukana oleminen tuo rentoutta jännittävään testitilanteeseen, mikä korostuu varsinkin lasten toimiessa testihenkilöinä. Toisen henkilön läsnäolo myös antaa testin valvojalle paremmat mahdollisuudet pysyä neutraalina ja ainoastaan seurata keskustelua taka-alalta. Lisäksi käytettävyyso ongelmia on mahdollista löytää enemmän lyhyemmässä ajassa. Toisaalta koska testihenkilöitä ja siten tarkkailtavia asioita on kaksinkertainen määrä, on myös todennäköisempää olla huomaamatta jotain. Paritestauksessa luotava testitilanne voi myös olla epäluonnollinen, jos vaikka testihenkilöt eivät tunne toisiaan, ja sen takia eivät käyttäydy normaalilla tavalla. Lisäksi kahden henkilön yhteistyöstä voi saada erilaisia tuloksia, kuin kummankaan työskentelystä yksin olisi saanut.

6 KÄYTETTÄVYYSTESTIN SUUNNITTELU, TOTEUTUS SEKÄ TULOKSET

Tässä luvussa esitetään ensin käytettävyystestien suunnitteluvaihe, sen jälkeen käydään läpi testin toteutus ja viimeisenä tulosten analysointi. Suunnitteluvaiheessa keskitytään erityisesti testitehtäviin, testihenkilöihin sekä testiympäristöön. Testin varsinainen toteutus puolestaan raportoidaan tehtäväkohtaisena läpikäyntinä.

6.1 Käytettävyystestien suunnittelu

Käytettävyystestit suoritetaan paritesteinä ja käytössä on ääneenajattelutekniikka. Paritestauksen myötä myös ujommat oppilaat saavat helpommin ilmaista ajatuksiaan, ja ääneenajattelu tuntuu luonnollisemmalta. Yhtä paria varten varataan aikaa noin 45 minuuttia, sillä sen tulisi riittää kaikkien tehtävien rauhalliseen suorittamiseen helposti. Tehtäviä testatessa aikuiset testikäyttäjät, jotka eivät osaa ohjelmoida, tekivät testin noin 20 minuutissa. Varattuun aikaan sisältyy myös alku- ja loppuhaastattelu sekä testien välillä suoritettavat toimenpiteet. Testitilanne videokuvataan sen myöhemmän analysoimisen helpottamiseksi.

Ennen testin alkua testihenkilöille esitellään testin järjestäjä, käytettävät laitteet sekä testattava ohjelma pääpiirteittäin. Tämän jälkeen esitellään testin kulku Hannan ym. (1997: 12) esittämää esimerkkiä mukailten. Seuraavaksi annetaan esimerkki ääneenajattelutekniikasta. Ennen testin aloittamista suoritetaan vielä alkuhaastattelu, jossa kysytään testihenkilöiden perustietoja. Varsinaisen testin alkaessa Koodikoira on avattuna sen aloitusnäkympään (kuva 4), sillä ohjelman löytäminen ei suoranaisesti kuulu sen käytettävyyteen.

Testin jälkeen suoritetaan loppuhaastattelu, jossa testihenkilöt saavat sanoa oman mielipiteensä Koodikoirasta. He voivat tuoda esille esimerkiksi erityisen helppoja tai vaikeita kohtia tai yleisiä parannusehdotuksia. Testihenkilöitä myös pyydetään arvioimaan

Koodikoiran käytön hauskuutta käyttäen luvussa 5.5 esiteltyä hymiömittaria. Loppuhaastattelu on tyyliltään luvun 5.3 ohjeiden mukaisesti vapaamuotoisempi, vaikka testin järjestäjä esittääkin joitain suunniteltuja kysymyksiä.

6.1.1 Testitehtävät

Testitehtävien tulee olla luvussa 5.2 läpikäytyjen ohjeiden mukaisesti mahdollisimman todenmukaisia ja järjestelmän oikeaa käyttötilannetta vastaavia. Koska tässä tutkimuksessa on mahdollisuus rakentaa tehtävät järjestelmän sisään, Koodikoira antaa itse käytettävyydestin tehtävät testikäyttäjille. Tämä on myös parhaiten todellista käyttötilannetta vastaava tapa. Osaan tehtävistä on kehitetty pienimuotoinen tarina parantamaan niiden mukaansatempaavuutta. Vaikka Koskisen (2005: 191) mukaan testi-tehtävänannon ei saisi itsessään antaa vihjeitä tehtävän suorittamiseen, Koodikoira tekee niin. Tämä johtuu siitä, että tehtävänannot ovat nyt osa testattavaa ohjelmaa, jossa erilaiset vinkit on suunniteltu parantamaan sen käytettävyyttä.

Alla on listattu tehtävät, jotka käytettävyydestissä suoritetaan. Ensimmäinen tehtävä on suunniteltu edelleen luvun 5.2 ohjeiden mukaisesti äärimmäisen helpoksi, että jokainen osaa varmasti sen tehdä.

1. Valitse hahmo
2. Hei! Siirrä paloja ohjelmaan ja auta Roni luun luokse. Aloita ohjelma painamalla ►.
3. Nyt luita on kaksi! Auta Ronia keräämään molemmat.
4. Ronin turkki on märkä! Kuivaa se kääntymällä puolia kierrosta kierroksia viiteen kertaan.
5. Kissa näyttämöllä! Liikuta Roni sen eteen ja murise!
6. Kissa ei ota lähteäkseen. Hauku sille ja sano puhekuplassa ”Häivy!”
7. Peruuta nyt kissan päälle!
8. Kissa on poissa! Juokse ympyrää luun ympärillä ja sano puhekuplassa ”Voitto!”
9. Lisää nyt 5 luuta satunnaisiin paikkoihin näyttämöllä ja kerää ne kaikki!
10. Nyt voit tehdä ihan mitä haluat! Kun olet valmis, paina ».

6.1.2 Testihenkilöt

Testihenkilöiden tulee niin ikään luvun 5.2 ohjeiden mukaisesti olla mahdollisimman samankaltaisia loppukäyttäjien kanssa. Koska Koodikoira on suunniteltu käytettäväksi 3.–6.-luokkalaisten ohjelmoinnin opetuksessa, testeihin valittiin neljäsluokkalaisia oppilaita. Testihenkilöt ovat siis täydellisesti loppukäyttäjien kaltaisia, sillä todellisuudessa he myös ovat loppukäyttäjiä. Testiin valittiin vain yhden ikäluokan lapsia kehitystason vaikutuksen minimoimiseksi, sillä kehitystason ei tulisi vaikuttaa testin tuloksiin. Todellisuudessa tätä ei kuitenkaan voida välttää, sillä jokainen lapsi on yksilö ja vähintään hieman eri kehitysvaiheessa. Myös testijärjestelyistä johtuen vain tietyn luokan oppilaiden käyttäminen osoittautui järkeväksi.

Testihenkilöiden huoltajille lähetettiin tiedote ja lupalomake (liite 3). Tiedotteessa kerrotaan tietoja suoritettavasta käytettävyytestauksesta ja lupalomakkeella varmistetaan huoltajien suostumus siihen, että heidän lapsensa osallistuvat testiin.

Testin suorittaa ainoastaan kuusi paria. Luvussa 5.2 todetun Nielsenin (2000) ohjeen mukaan jo viisi testihenkilöä riittää löytämään suurimman osan käytettävyysongelmista. Käytössä on kuitenkin kuusi paria sen varalta, että joku pari päättää lopettaa testin kesken. Tällöin saadaan edelleen tietoa viideltä testin loppuun asti suorittaneelta parilta. Yhtä paria ei toisaalta voida pitää kahtena testihenkilönä, vaan ennemminkin vain yhtenä testisuorituksena.

Testihenkilöistä saadaan lisää tietoa alkuhaastattelun aikana, joka suoritetaan ennen varsinaisen testin aloittamista. Alkuhaastattelussa testihenkilöiltä kysytään heidän sukupuolen ja iän lisäksi tottumusta graafisten ohjelmointiympäristöjen, pelien ja tietokoneen käyttäjinä, sekä viimeisintä matematiikan numeroa.

6.1.3 Testiympäristö

Testiympäristönä käytetään mahdollisimman turvallista, viihtyisää ja todellisen käyttöympäristön kaltaista tilaa, kuten luvussa 5.2 todettiin olevan optimaalista. Testi

suoritetaankin oppilaiden omassa koulussa yhden koulupäivän aikana. Käytössä on rauhallinen huone (kuva 10), jossa testin aikana ei ole yhden testihenkilöparin ja testin järjestäjän lisäksi muita ihmisiä. Testilaitteistona käytetään testin järjestäjän kannettavaa tietokonetta, sillä se helpottaa testin järjestämistä ja etukäteisvalmisteluja.



Kuva 10. Huone, jossa käytettävyystestit suoritettiin.

6.2 Käytettävyystestin läpikäynti

Tässä alaluvussa esitellään ensin alkutietokyselyn tulokset, jonka jälkeen siirrytään tehtäväkohtaiseen läpikäyntiin. Tehtäväkohtaisessa läpikäynnissä kerrotaan kaikkien testiparien testisuorituksista yksi tehtävä kerrallaan muistiinpanojen ja litteroidun videokuvanauhan perusteella. Lopuksi käsitellään vielä loppuhaastattelussa esille nousseita seikkoja, kuten Koodikoiran käytön hauskuus.

6.2.1 Alkutietokysely

Käytettävyystestiin osallistui siis kuusi paria, eli kaksitoista oppilasta, joista puolet

olivat tyttöjä ja puolet poikia. Kaikki testihenkilöt olivat joko kymmenen tai yksitoista vuotta vanhoja. Jokainen testihenkilö oli myöskin käyttänyt jo aiemmin jonkinlaista graafista ohjelmointiympäristöä, joista esimerkkeinä mainittiin Scratch ja Code.org. Testihenkilöiltä kysyttiin myös kuinka paljon he tavallisesti pelaavat pelejä, johon seitsemän testihenkilöä vastasi *päivittäin* tai *lähes päivittäin*, ja viisi *vain silloin tällöin* tai *ei ollenkaan*. Kaikki testiin osallistuneet oppilaat kertoivat saaneensa matematiikan kokeista viime aikoina joko kiitettäviä tai hyviä arvosanoja. Alkutietokyselyn tulokset kokonaisuudessaan on esitetty taulukossa liitteessä 4.

6.2.2 Tehtäväkohtainen läpikäynti

Ensimmäisenä tehtävänä testihenkilöillä oli valita koirahahmoista joko Raketti tai Yasmin yksinkertaisesti klikkaamalla sitä hiirellä. Tämä tehtävä oli niin helppo, että testihenkilöt tuskin tajusivat sitä edes varsinaiseksi testitehtäväksi. Mikäli tässä tehtävässä viivyttiin paria sekuntia enemmän, se johtui siitä, etteivät testihenkilöt osanneet päättää kumman hahmon he valitsisivat.

Toisessa tehtävässä testihenkilöiden tuli käyttää Koodikoiraa ensimmäisen kerran ohjelmointimielessä. Näyttämöllä oleva koirahahmo, jota nimitetään Roniksi, piti liikuttaa luun luokse. Tämä vaati hahmon liikuttamista kolme askelta eteenpäin. Puolet testipareista käyttivät kolme kertaa peräkkäin ”liiku eteenpäin 1 askel” -palaa, kun taas puolet ottivat käyttöön ”liiku eteenpäin 5 askelta” -palan, johon he vaihtoivat askelten määräksi tarvittavat kolme askelta. Kaikille testihenkilöille ei ollut täysin selvää milloin Roni on luun luona, ja osa liikuttikin hahmoa eteenpäin aluksi vain kaksi kertaa. Kokeillessaan ohjelman toimintaa he kuitenkin huomasivat ettei hahmo syönyt luuta, jolloin myös he lisäsivät ohjelmaan kolmannen askeleen.

Ohjelman aloituspainike ei ollut hankala löytää yhdellekään parille, vaan sitä käytettiin automaattisesti. Myös ohjelman aloittamisen konsepti ymmärrettiin todella hyvin verrattuna esimerkiksi Puntilan (2016: 24–30) suorittamaan käytettävyydestiin. Tämä voi johtua siitä, että nyt testihenkilöt olivat käyttäneet huomattavasti enemmän graafisia ohjelmointiympäristöjä niin koulussa kuin vapaa-aikanaankin ennen testin suorittamista.

Tehtävässä kolme testikäyttäjien tuli liikuttaa Ronia keräämään kaksi luuta. Tämä vaati liikkumisalojen käytön lisäksi myös kääntymisalojen käyttöä. Vaikka tehtävän suorittaminen onnistui pääosin hyvin, myös ongelmia havaittiin. Testihenkilöt laskivat edelleen joitakin askelmatkoja väärin, mutta uutena ongelmana esiin nousi vasemmalle ja oikealle kääntymisen valitseminen. Tavallisen oikean ja vasemman erottamisen harpuuden lisäksi joidenkin testihenkilöiden oli hankala ajatella kääntymistä nimenomaan näyttämöllä olevan koirahahmon kannalta katsottuna. Kuitenkin tehtävänannossa pääasiassa haettu toiminto, eli alojen käyttö myös toiselta välilehdeltä onnistui hyvin viideltä testiparilta kuudesta. Yksi pari ei etsinyt kääntymistä ollenkaan, vaan pohti liikkumista sivullepäin. Kun pari lopulta löysi *kääntymisen*-välilehden, he aloittivat ohjelman ja yrittivät kääntää hahmoa painamalla kääntymisalaa juuri oikealla hetkellä. Edelleen heidän lisäessä kääntymisaloja ohjelmaan he sekoittivat ne liikkumiseen vasemmalle, mikä johti hahmon kääntymään paikallaan useita kertoja peräkkäin. Kuitenkin myös tämä testipari onnistui suorittamaan tehtävän loppuun ilman testin valvojan puuttumista asiaan.

Kolmannessa tehtävässä ilmeni ohjelmaan jo asetettujen alojen poistamista ensimmäisen kerran. Alojen poistaminen ei kuitenkaan aiheuttanut yhdellekään parille minkäänlaisia vaikeuksia, vaan niiden vetäminen ilmestyvään roskakoriin oli heidän mielestään täysin loogista ja luontevaa. Lisäksi edellisessä tehtävässä ”liiku eteenpäin 1 askel” -aloja käyttäneet parit käyttivät myös nyt useamman askeleen etenemisiä kerralla.

Neljännessä tehtävässä, jossa Roni piti laittaa kääntymään puolia kierroksia viiteen kertaan, ei ilmennyt minkäänlaisia ongelmia. Päinvastoin, kaikki lapset pitivät tätä tehtävää huomattavasti helpompana verrattuna edelliseen tehtävään. Yksi pari jopa asetti koiranhahmon haukkumaan tehtävänannon mukaisen suorituksen lisäksi. Suurin yllätys kuitenkin oli pari, joka etsi omien sanojensa mukaan ”tee kokoajan -lohkoa”, jolla olisi asettanut koiran toistamaan samaa liikettä. Koodikoiran testattuun prototyyppiin ei toistaloja ehditty tehdä, mutta kommentti paljasti kyseisten testihenkilöiden kokemuksen opetuksellisten graafisten ohjelmointiympäristöjen käytöstä, sekä jotain tällaisen opetuksen toimivuudesta. Kun etsittyjä aloja ei löytynytäkään,

myös tämä pari asetti koiran toistamaan haluttua asiaa tuomalla samoja paloja ohjemaan useamman kerran.

Viidennessä tehtävässä näyttämölle oli ilmestynyt kissa, jonka eteen koirahahmo piti liikuttaa. Lisäksi ohjelman lopussa tuli murista kissalle. Useat testihenkilöt osoittivat välittömästi välilehteä, jossa on puhekuplan kuva. He totesivat, että sieltä saa lisättyä murinan ohjelmaan. Käytetty ikoni oli siis riittävän kuvaava, vaikka sitä ei valittukaan kyselytutkimuksen avulla. Käytettävyysongelmana tässä tehtävässä ilmeni ensimmäistä kertaa eteenpäin ja taaksepäin liikkumispalojen sekoittaminen keskenään. Testihenkilöt eivät lukeneet liikkumispalassa olevaa tekstiä riittävän tarkasti, vaan lisäsivät ensimmäisenä silmään osuvan liikkumispalan ohjelmaan. Tämä käytettävyysongelma ilmeni myöhemmissä tehtävissä lähes jokaisella testiparilla. Ongelma tuli kuitenkin aina esille ohjelmaa testatessa, jolloin kyseinen ongelma korjattiin vaihtamalla pala oikeaksi.

Tehtävä kuusi oli yksinkertainen, sillä siinä tarvittiin paloja ainoastaan edellisessä tehtävässä löydettyltä *puhuminen*-välilehdeltä. Minkäänlaisia käytettävyysongelmia ei tehtävää suoritettaessa ilmennyt. Esille nousi ainoastaan tasoeroja näppäimistöllä kirjoittamisen nopeudessa. Kuvassa 11 on esitetty ja ratkaistu tehtävä numero kuusi.



Kuva 11. Kuudes tehtävä ja tehtävänannon mukainen ohjelma ajettuna loppuun.

Seitsemännessä tehtävässä koirahahmo piti laittaa peruuttamaan kissahahmon päälle. Tämän toteuttamiseksi Ronia piti ensin kääntää puoli kierrosta ja sitten liikuttaa yksi askel taaksepäin. Monet testihenkilöt pohtivat kuinka koirahahmon saa peruuttamaan, ja erilaisia pohdintoja käytiin liikkumisesta eteenpäin, kääntymisestä kerran vasemmalle ja puolen kierroksen kääntymisestä kahteen kertaan. Kaikissa testipareissa kuitenkin vähintään toinen testihenkilö ymmärsi tehtävän oikean toteutustavan heti tehtävänannon jälkeen, vaikka ei saanut selkeästi sanottua sitä heti parilleen. Tehtävä seitsemän oli kuitenkin yksi nopeimmista suorittaa.

Tehtävässä numero kahdeksan testihenkilöillä oli edelleen erimielisyyksiä tarvittavien askelten määrästä ja kääntymissuunnista. Tehtävänannon mukaan Roni piti liikuttaa luun ympäri, ja sen tuli sanoa puhekuplassa ”Voitto!”. Kaikki testiparit saivat tehtävän suoritettua, vaikka useat kommentoivatkin tehtävää jo vaikeammaksi. Tässä tehtävässä erityisesti keskeneräisen ohjelman kokeileminen nousi suureen rooliin. Yksi testihenkilö kertoi katsovansa ohjelmaa ajettaessa ilmestyvän osoittimen avulla, missä kohtaa heidän ohjelmassaan on virhe. Myös toisessa testiparissa tiedettiin keskustella, että osoitin kertoo kohdan, jossa ohjelma on kullakin ajanhetkellä menossa. Monet taas kokeilivat, mihin asti he rakentamallaan ohjelmalla ovat koirahahmoa jo liikkuneet, että osaavat vielä lisätä oikean määrän paloja ohjelman loppuun.

Testiparit tekivät muokkauksensa ohjelmaan käytännössä aina kelaamatta näyttämön ohjelmaa ensin takaisin alkuun. Tämä johti kaikilla testihenkilöillä ensimmäistä kertaa uudelleen testattaessa lyhyeen pohdintataukoon miksi ohjelma ei alkanutkaan, vaikka he painoivat samasta kohtaa kuin viimeksi. Käytetyt ikonit auttoivat heitä ymmärtämään alkuun kelaamisen ja ohjelman aloittamisen eron, jonka he sitten muistivat aina opittuaan sen. Kuitenkin muokkaukset ohjelmaan tehtiin edelleen aina ennen ohjelman kelaamista alkuun.

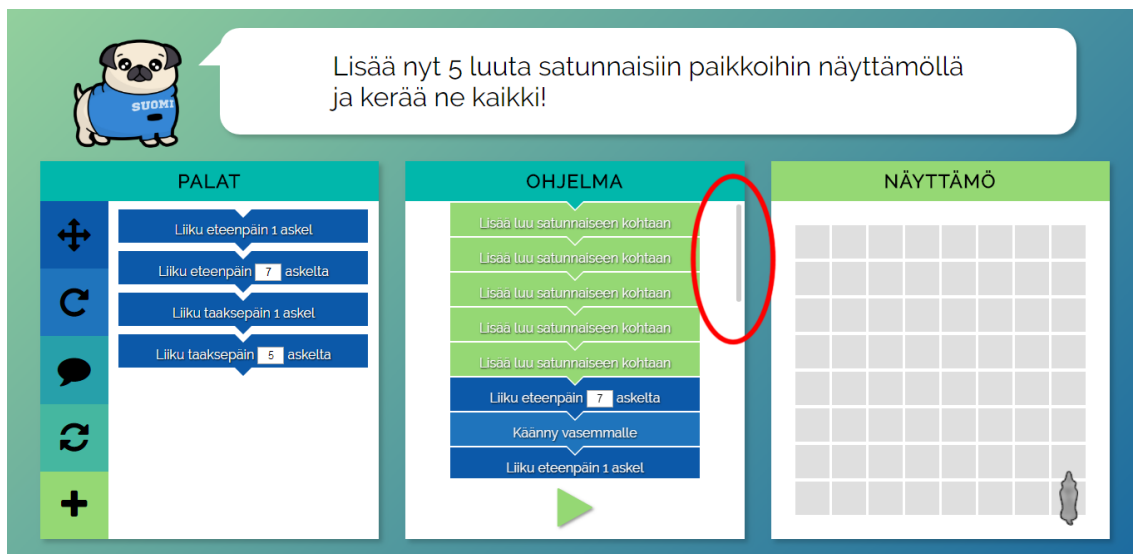
Koodikoira ehkäisee erityisesti lapsilla esiintyviä fyysisen kehityksen tasosta johtuvia ongelmia raahaa ja pudota -toimenpiteen suorituksessa asettamalla uuden lisättävän palan aina ohjelman alimmaiseksi riippumatta siitä, mihin kohtaa ohjelmaa sen pu-dottaa. Tämä toiminnallisuus näytti palvelevan järjestelmän käyttöä hyvin, kunnes

testihenkilöt yrittivät lisätä tehtävässä kahdeksan uusia paloja ohjelman väliin. Mikäli ohjelma oli riittävän lyhyt, he huomasivat palan ilmestyvän alimmaiseksi. Kuitenkin jos ohjelman kaikki palat eivät näkyneet kerralla, vaan saatiin näkyviin vasta vierityspalkkia käyttämällä, testihenkilöt saattoivat lisätä useita kertoja saman palan tajuamatta sen aina asettuvan ohjelman alimmaiseksi.

Yhdeksännessä tehtävässä, joka oli viimeinen varsinainen ohjeistettu tehtävä, oli jo paljon monimutkaisempi ohjelmoinnillinen ajatus, ja se näkyi tehtävään käytetyssä ajassa. Lähes kaikki testiparit käyttivät tähän yksittäiseen tehtävään aikaa yhtä paljon kuin kaikkiin aiempiin tehtäviin yhteensä. Tehtävänä oli lisätä näyttämölle viisi luuta satunnaisiin kohtiin ja kerätä ne kaikki. Aivan tehtävän aluksi jokainen testipari alkoi ratkaista tehtävää väärällä tavalla. Viisi kuudesta testiparista asetti oikein tarvittavat viisi ”lisää luu satunnaiseen kohtaan” -palaa ohjelmaan, mutta sitten he aloittivat ohjelman saadakseen luut näyttämölle, ja ohjelmoivat Ronin keräämään luut niiltä paikoilta. He aloittivat ohjelman sitten uudestaan vain huomatakseen, että luut ilmestyvät toisella kerralla eri kohtiin näyttämöllä. Yksi testipari puolestaan luuli lisäävänsä luut ”lisää luu satunnaiseen kohtaan” -palalla Ronin senhetkiseen sijaintiin.

Erilaiset väärät ratkaisuyritykset kestivät vaihdellen parista minuutista viiteentoista minuuttiin. Yrityksiin kuului esimerkiksi luun lisäyspalojen poistamista ohjelman ajamisen jälkeen sekä yrityksiä aloittaa ohjelma sen keskeltä. Mikäli tähän oli kulunut aikaa yli viisitoista minuuttia, testin valvoja auttoi testihenkilöitä kysymyksellä ”mikä olisi keino, että saisitte kaikki luut kerättyä riippumatta siitä, mihin ne ilmestyvät?” Kolme testiparia keksi oikean ratkaisun jo ennen valvojan antamaa vihjettä, kun taas kaksi paria vihjeen jälkeen. Yksi pari halusi puolestaan siirtyä seuraavaan tehtävään heti epäonnistuttuaan tässä tehtävässä. Erityisen mielenkiintoista oli se, että useista pareista toinen testihenkilö tuntui tietävän tehtävän yhdeksän oikean ratkaisutavan lähes heti, mutta heiltä puuttui rohkeutta esittää ajatuksensa tarpeeksi selkeästi parilleen. Vaikka viisi paria keksikin tehtävälle oikean ratkaisutavan, heillä oli edelleen ongelmia sen toteutuksessa esimerkiksi kääntymisten, suuntien ja matkojen ymmärtämisessä. Lopulta vain kaksi testiparia sai tehtyä tehtävän onnistuneesti valmiiksi asti ennen kuin testin valvoja päätti aikarajan lähestyessä päästää testihenkilöt seuraavaan tehtävään.

Tehtävässä yhdeksän ohjelmaan lisättiin paloja enemmän kuin näytölle kerrallaan mahtui; tällaisessa tapauksessa Koodikoiran ohjelma-osioon ilmestyy pystysuuntainen vierityspalkki kuvan 12 mukaisesti. Vaikka Idlerin (2014) mukaan lapsille suunnatuissa järjestelmissä tulee mahdollisuuksien mukaan välttää pystysuuntaisten vierityspalkkien käyttöä, tässä käytettävyydestissä kaikki testiparit osasivat käyttää sitä välittömästi. Puolet pareista suosivat vierityspalkin käyttöä hiiren rullalla, kun taas puolet vetämällä palkista hiiren vasemmalla näppäimellä. Vierityspalkki toi kuitenkin esille myös selkeän virheen Koodikoirassa; palojen uudelleenjärjesteleminen vierityspalkin ollessa olemassa oli lähes mahdotonta, sillä Koodikoiran tulkinta hiiren sijainnista ei ottanut huomioon palkin sijaintia. Palat siis siirtyivät aina siten, kun ne olisi pudotettu paikalleen palkin ollessa täysin ylhäällä.



Kuva 12. Ohjelman pituuden kasvaessa ilmestyvä vierityspalkki ympyröity punaisella.

Koska yhdeksännessä tehtävässä toistettiin seitsemän askeleen etenemistä useita kertoja, kaksi testiparia päätti vaihtaa edettävien askelten määrää jo palat-osiossa. Näin heidän ei tarvinnut jokaisella kerralla vaihtaa askelten määrää erikseen. Tämä ominaisuus tarjoaa kokeneemmille käyttäjille mahdollisuuden käyttää järjestelmää tehokkaammin. Useimmat testiparit kuitenkin syöttivät tekstikenttään tietoa vasta siirrettyään sen ohjelmaan.

Viimeisessä tehtävässä, eli tehtävässä kymmenen testihenkilöille annettiin vapaat kädet; he saivat tehdä vapaasti haluamansa kaltaisen ohjelman. Vaikka ohjelmat olisivat voineet olla minkälaisia tahansa, eri parien tekemät ohjelmat olivat yllättävän samankaltaisia. Useat parit päättivät kokeilla paloja, joita ei oltu tarvittu aiemmissa tehtävissä, kuten esimerkiksi kissan lisäämistä näyttämölle. Erityisen suosittua olivat myös haukkumisen ja murinan toistaminen todella moneen kertaan, sekä satojen askeleiden etenemisen yrittäminen. Kun testiparit olivat mielestään suorittaneet viimeisenkin tehtävän, he painoivat etenemispainiketta, joka toi näkyviin ilmoituksen testin päättymisestä ja onnittelut sen suorittamisesta. Aivan kaikille ei ollut heti selvää, mitä etenemispainikkeella tarkoitettiin, mutta käytetty uniikki ikoni ei antanut testihenkilöille erehtymisen mahdollisuutta.

Mikäli testihenkilöt eivät olleet viimeiseen tehtävään mennessä yrittäneet kokeilla tehtävänannon kuuntelemista uudelleen, testin valvoja kehotti heitä tekemään niin. Useimmiten testihenkilöiden ensimmäinen reaktio oli painaa puhekuplasta, mutta koska siitä ei tapahtunut mitään, he hyvin nopeasti veivät hiiren alussa valitsemansa koirahahmon päälle. Kun koirahahmo tällöin suureni, testihenkilöt uskalsivat painaa siitä. Jotkut testiparit eivät ymmärtäneet mitä valvoja tällä kehotuksellaan haki, vaan yrittivät aluksi etsiä jotain palaa, jolla saisi tehtävänannon toistumaan. Lopulta kuitenkin viisi testiparia kuudesta onnistui kuuntelemaan tehtävänannon uudestaan, sillä yksi pari tulkitsi valvojan kommentin kehotuksena toimimaan tehtävänannon mukaisesti ja painoi välittömästi testin päättymiseen johtavaa painiketta.

6.2.3 Loppuhaastattelu

Kun näytölle oli ilmestynyt viesti testin päättymisestä ja onnittelu sen suorittamisesta, käytiin testihenkilöiden kanssa vapaamuotoisempi keskustelu Koodikoiran toiminnasta. Keskustelussa testihenkilöt saivat esittää omia ajatuksiaan helpoista ja vaikeista kohdista, sekä mielipiteensä käyttäisivätkö he Koodikoiraa uudestaan. He saivat myös esittää mielestään sopivia parannusehdotuksia. Haastattelun loppuksi käytössä oli vielä kuvassa 9 esitetty hymiömittari, josta jokainen testihenkilö sai ympyröidä parhaiten käytön hauskuutta kuvaavan hymynaaman.

Erityisen helppoina kohtina testissä pidettiin ensimmäisiä tehtäviä, mutta monet kuitenkin lisäsivät, että niiden avulla oppi Koodikoiran toiminnan mukavasti ennen siirtymistä vaikeampiin tehtäviin. Vaikeina kohtina puolestaan pidettiin loppupuolen tehtäviä, joita jotkut pitivät sopivina oppimishaasteina, kun taas jotkut yksinkertaisesti liian vaikeina. Lopulta kaikki kuusi testiparia kuitenkin totesivat, että käyttäisivät Koodikoiraa uudestaan, mikäli se olisi mahdollista. Verrattuna Code.orgiin Koodikoiraa pidettiin sekä selkeämpänä, samankaltaisena että sekavampana. Kaikki testiparit kuitenkin kertoivat käyttävänsä kaikista kokeilemistaan graafisista ohjelmointiympäristöistä mieluiten juuri Koodikoiraa.

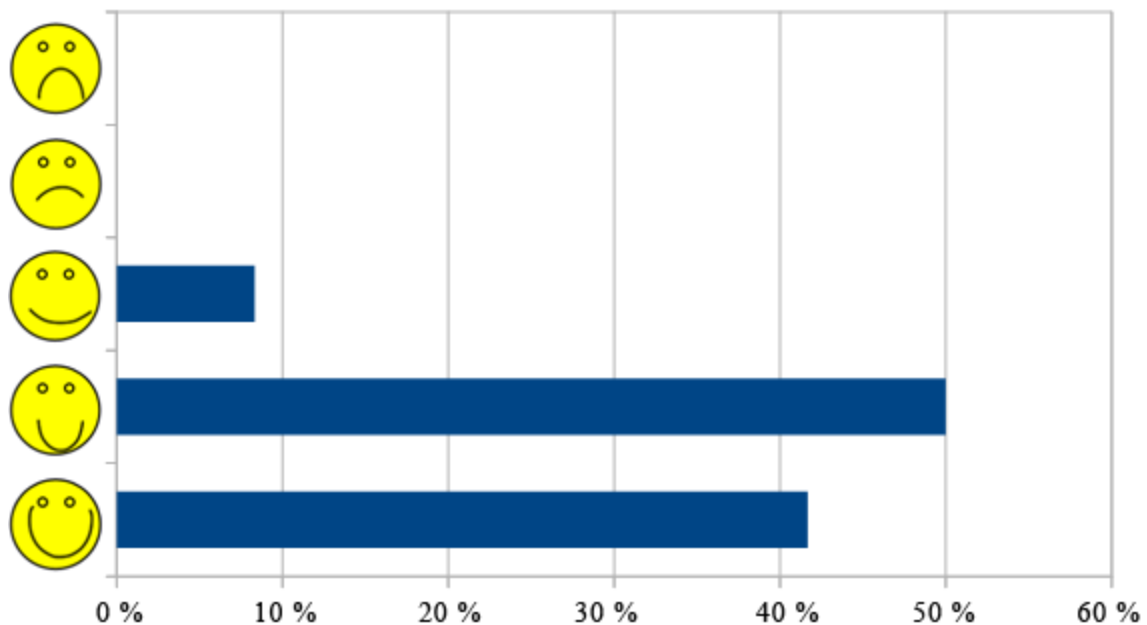
Kiitosta Koodikoira sai loppuhaastattelussa esimerkiksi äänistä, joita keuhuttiin pääosin mahtaviksi ja hauskoiksi. Yksi testihenkilö olisi kuitenkin halunnut hieman aidommalta kuulostavat haukkumis- ja murina-äännet. Myös Koodikoiran selkeitä ohjeita keuhuttiin, mikä kuulostaa käytettävyyden kannalta tarkasteltuna erinomaiselta kommentilta.

Parannusehdotuksina testihenkilöillä oli useita, joista selkeimpänä ja useimmiten nousi esille toive erilaisten palojen lisäämisestä. Prototyypissä oli käytettävissä ainoastaan kaksitoista erilaista palaa, eikä niihin kuulunut esimerkiksi toistorakennetta tai jorakennetta. Testihenkilöt haluavat siis selkeästi rakentaa monipuolisempia ohjelmia, kuten Koodikoiran lopullisessa versiossa on tarkoitus olla mahdollista. Toinen parannusehdotus oli valittavina olevien hahmojen lisääminen. Yksi testihenkilö myös halusi kaikkien kussakin tehtävässä tarvittavien palojen olevan näkyvillä samaan aikaan, eikä jaoteltuna eri välilehtien alle. Tämä saattaa olla aivan ensimmäisillä käyttökerroilla hyvä idea, mutta lopulta käyttäjän vapauden suosimiseksi käytössä ovat kaikki palat, joita on niiden määrän takia välttämätöntä jaotella. Tämä tapa myös vastaa paremmin todellista ohjelmointia, jossa ei kerrota valmiiksi tarvittavia käskyjä, vaan käytössä on aina kaikki ohjelmointikielen tarjoamat mahdollisuudet.

Käytettävyyden kannalta annettiin vielä kaksi mielenkiintoista parannusehdotusta: useamman palan poistaminen kerralla sekä palojen kopiointi. Joissain tehtävissä testihenkilöt olivat asettaneet ohjelmaan jopa kymmeniä paloja, joiden poistaminen niiden osoittauduttua vääräksi oli hidasta vetämällä yksitellen roskakoriin. Eräs

testihenkilö toivoi poistavansa kaikki palat yksittäisen poistettavan palan alapuolelta. Tämä kuitenkin karsisi mahdollisuuden poistaa ylimääräisiä paloja keskeltä ohjelmaa. Yksi toimiva ratkaisu olisi pitää nykyinen toiminnallisuus ennallaan, mutta lisätä myös mahdollisuus poistaa kaikki palat yhdellä painalluksella. Palojen kopiointia puolestaan toivottiin tehtäviin, joissa piti toistaa täysin samanlaisia paloja tai rakenteita useita kertoja. Tällainen kopiointitoiminto palvelisi siis täysin samaa tarkoitusta kuin toistorakenne, joten toistamisen mahdollistuttua palojen kopiointia ei tarvittaisi.

Aivan keskustelun lopuksi toteutetun hymiömittarin mukaan Koodikoiran käytön hauskuus oli testihenkilöiden mielestä todella hyvällä tasolla (ks. kuva 13). Jokainen testihenkilö sai ympyröidä mielestään järjestelmän käytön hauskuutta parhaiten kuvaavan hymynaaman, ja puolet testihenkilöistä päätyikin ympyröimään toiseksi parhaan hymynaaman. Jopa 42 % testihenkilöistä ympyröi kaikista iloisimman hymynaaman, kun taas vain 8 % vastasi keskiverron. Kahta surullisinta hymynaamaa ei ympyröity lainkaan. Hymiömittarin tuloksia tukee myös yhden testihenkilön kommentti, joka kuului käytävästä parin poistuessa testistä: ”Toi oli oikeesti tosi hauskaa!”



Kuva 13. Koodikoiran käytön hauskuus hymiömittarin avulla mitattuna.

6.3 Käytettävyydestin tulosten analysointi

Tässä alaluvussa tarkastellaan Koodikoiran prototyypin käytettävyydestestauksessa esille nousseita käytettävyysoongelmia. Löydetyt käytettävyysongelmat jaotellaan Nielsenin (1993: 103) vakavuusluokittelun (ks. luku 5.4) mukaisesti viiteen eri kategoriaan vakavuuden sekä esiintymistiheyden perusteella. Havaitut käytettävyysongelmat ovat esitetty luokiteltuna taulukossa 3. Tämän jälkeen pohditaan, miten kyseiset ongelmat voitaisiin parhaiten ratkaista.

Taulukko 3. Käytettävyydestestissä havaitut käytettävyysongelmat luokiteltuna.

Käytettävyysongelma	Vakavuusluokka	Esiintymistiheys
Palojen paikkojen vaihtaminen käytännössä mahdotonta vierityspalkin ollessa olemassa	4/4	3/6
Eteenpäin- ja taaksepäin-liikkumisalojen sekoittaminen keskenään	3/4	4/6
Palojen poistaminen yksitellen hidasta	3/4	3/6
Ronin sijainnin ja liikuttavien matkojen ymmärtäminen hankalaa	3/4	3/6
Palan lisääminen keskelle ohjelmaa mahdotonta	3/4	1/6
Muutosten tekeminen kelaamatta ohjelmaa alkuun johtaa helposti harhaan	2/4	6/6
Toistorakenteen / palojen puuttuminen	2/4	4/6
Oikean ja vasemman erottaminen	2/4	3/6
Liikkumisen ajattelemisen hahmon näkökulmasta hankalaa	2/4	2/6
Samojen palojen lisääminen useita kertoja peräkkäin hidasta / kopioinnin puuttuminen	2/4	2/6
Välilehtien käyttö hankalaa	2/4	1/6
Tehtävänannon toistamista ei löydetä	1/4	5/6
Ohjelman alusta alkamisen ymmärtäminen hankalaa (ainoastaan yhdessä tehtävässä)	0/4	6/6

Vakavimpia mahdollisia, luokan 4 käytettävyysoongelmia löytyi yksi kappale ja se on luokituksen mukaisesti korjattava ennen järjestelmän julkaisemista. Ongelmana on ohjelmassa olevien palojen paikkojen vaihtaminen pystysuuntaisen vierityspalkin ollessa olemassa. Mikäli paloja on niin paljon että vierityspalkki on ilmestynyt, ei paloja käytännössä voi liikuttaa, sillä järjestelmä tulkitsee hiiren sijainnin palan pudotushetkellä ikään kuin vierityspalkkia ei olisi olemassa, tai se olisi täysin ylhäällä. Tämä johtaa palan putoamiseen väärään kohtaan ohjelmaa. Ongelma on kuitenkin verrattain helppo korjata, sillä järjestelmä asetetaan yksinkertaisesti ottamaan huomioon myös vierityspalkin sijainti, jolloin pala putoaa haluttuun kohtaan. Muita ennen järjestelmän julkaisemista korjattavaksi luokiteltuja käytettävyysoongelmia ei testauksessa löytynyt.

Vakavuusluokan 3, eli korkealla prioriteetilla korjattavia käytettävyysoongelmia löytyi yhteensä neljä kappaletta. Kun useampi ongelma on asetettu samaan vakavuusluokkaan, niiden korjaamistarkeys voidaan määritellä esimerkiksi esiintymistiheyden perusteella. Useimmiten esiintynyt kolmannen vakavuusluokan ongelma on eteenpäin- ja taaksepäin-liikkumispalojen sekoittaminen keskenään. Tämä johtuu siitä, etteivät lapset lukeneet paloissa olevia tekstejä niiden ensimmäisten käyttökertojen jälkeen lähes ollenkaan. Palat ”Liiku eteenpäin 1 askel” ja ”Liiku taaksepäin 1 askel” ovat nopeasti katsottuina hyvin samankaltaisia myös kokeneemmille käyttäjille. Ongelman voi ratkaista vaihtamalla taaksepäin liikkumisen ilmaisun peruuttamiseksi. ”Peruuta 1 askel” on jo huomattavasti vaikeampi sekoittaa eteenpäin liikkumiseen.

Kolmanteen vakavuusluokkaan luokiteltiin myös palojen yksitellen poistamisen hitaus sekä Ronin sijainnin ja liikuttavien matkojen ymmärtämisen hankaluus. Joissakin tehtävissä testihenkilöt ehtivät rakentaa pitkän ohjelman, joka osoittautui testauksen jälkeen toimimattomaksi. Tällöin he joutuivat poistamaan ohjelmasta jokaisen palan yksi kerrallaan, joka oli aikaa vievä ja samaa liikettä toistava toimenpide. Toimiva ratkaisu olisi lisätä uusi painike, josta voi poistaa kaikki ohjelmaan siirretyt palat kerralla. Tällöin ohjelman rakentaminen uudelleen alusta asti olisi helpompaa, mutta nykyinen toiminnallisuus yksittäisten palojen poistamiseksi säilyisi ennallaan. Näyttämöllä olevan koirahahmon ja liikuttavien matkojen hahmottamista puolestaan voitaisiin helpottaa pienentämällä hahmoa jonkin verran, sillä tällä hetkellä se on hieman yhtä näyttämön

ruutua suurempi. Hahmon ei kuitenkaan tarvitse mahtua ruutuun aivan täysin, vaan pieni rajojen rikkominen vähentää visuaalista mekaanisuutta. Toinen selventävä keino on muokata ensimmäistä tehtävänantoa kehottamaan käyttäjiä *keräämään* luu sen sijaan, että siinä pyydetäisiin liikuttamaan hahmoa luun *luokse*.

Viimeinen käytettävyysoongelma vakavuusluokassa 3 on palojen lisäämisen mahdottomuus keskelle ohjelmaa. Vaikka alun perin kyseinen toiminto oli suunniteltu ehkäisemään raahaa ja vedä -toiminnon suorittamisen epätarkkuudesta johtuvia ongelmia asettamalla uuden palan aina ohjelman alimmaiseksi, se muotoutui myös käytettävyysongelmaksi itsessään. Testihenkilöistä ei ollut aina selkeää, miksi pala menee alimmaiseksi vaikka he yrittivät pudottaa sitä olemassa olevien palojen väliin. Kuitenkin toiminto palveli hyvin myös sen alkuperäisessä tarkoituksessa. Yksi ratkaisu olisi lisätä asetus siitä, pudotetaanko uusi pala aina alimmaksi vai hiiren sijaintiin. Tällöin kokeneemmat käyttäjät voisivat halutessaan vaihtaa asetusta ja lisätä paloja myös keskelle ohjelmaa.

Vakavuusluokkaan 2 kuuluvia käytettävyysogelmia havaittiin eniten, yhteensä kuusi kappaletta. Nämä ongelmat tulee korjata, mutta matalalla prioriteetilla. Yksi näistä ongelmista esiintyi kaikilla testikäyttäjillä: muutosten tekeminen ohjelmaan kelaamatta sitä ensin alkuun johtaa helposti harhaan. Varsinkin vaikeammassa tehtävissä testihenkilöt saattoivat tehdä muutoksia ohjelmaan olettaen sen sitten jatkavan pisteestä, johon se edellisellä kokeilulla jäi. Ongelma havaittiin vasta seuraavalla kokeilulla, johon mennessä oli voitu kuluttaa paljon aikaa ja työtä. Tämä ongelma on hieman hankalampi korjata, sillä ohjelman toistaminen alusta kuuluu olennaisesti ohjelmoinnilliseen ajattelutapaan, eikä se tuottanut vaikeuksia esimerkiksi helpommissa testitehtävissä. Lisäksi joissain tehtävissä testihenkilöt käyttivät toimintoa hyväkseen tarkastellakseen, missä kohdassa ohjelmaa on virhe tai mihin asti he rakentamallaan ohjelmalla hahmoa tällä hetkellä liikuttavat. Selkein ratkaisu olisi pakottaa käyttäjiä kelaamaan ohjelmaa alkuun aina ennen kuin ohjelman muokkaaminen on edes mahdollista. Tällöin ohjelmaa voitaisiin kokeilla nykyisen lailla keskeneräisenä, mutta myös ohjelman alusta alkamisen ajastus selkeytyisi.

Toistorakenteen ja muiden lisäpalojen puuttuminen puolestaan ei varsinaisesti ole suoraan käytettävyysongelma. Siitä muodostuu ongelma ensin käyttäjäkokemukseen, mikäli testihenkilöt olettavat järjestelmässä olevan sieltä puuttuvaa sisältöä ja pettyvät kun eivät sitä löydä. Käytettävyysongelmaksi se puolestaan muodostuu testihenkilöiden ollessa niin varmoja esimerkiksi toistorakenteen olemassa olosta, että he käyttävät huomattavasti aikaa sen etsimiseen. Tästä syystä palojen puuttuminen on asetettu vakavuusluokkaan 2. Koodikoiran jatkokehityksessä on kuitenkin tarkoitus lisätä järjestelmään toisto- ja jos-rakenteet sekä muita paloja, mikä ratkaisee tämän käytettävyysongelman. Toistorakenteen lisääminen ratkaisee myös kopiointitoiminnon puuttumisen. Osa testihenkilöistä toivoi mahdollisuutta kopioida ohjelmaan jo asettamia palasekvenssejä, sillä samaa rakennetta piti toistaa useita kertoja peräkkäin. Toistorakenteen lisäämisen jälkeen palojen kopioimista ei enää tarvita.

Muita vakavuusluokan 2 käytettävyysongelmia ovat oikean ja vasemman erottaminen ja liikkumisen ymmärtäminen nimenomaan Ronin näkökulmasta. Koodikoiran kohde-ryhmänä olevilla 3.–6.-luokkalaisilla voi olla vielä vaikeuksia erottaa oikeata ja vasenta, ja tämä johti väärin kääntymispalojen asettamiseen ohjelmaan. Ongelmaa vaikeutti entisestään se, että suunnat piti tulkita näyttämöllä olevan koirahahmon näkökulmasta. Oikean ja vasemman sekoittamista voisi ehkäistä asettamalla kääntymispaloihin kääntymissuuntaa osoittavat nuoli-ikonit. Suuntien tulkitseminen Ronin näkökulmasta puolestaan tukee olio-ohjelmoinnin ajattelutapaa, jossa oliot ovat itsenäisempiä toimijoita, eivätkä vain ”suoraan ylhäältä ohjattuja”. Tämä on hyvä oppia jo aikaisessa vaiheessa. Lisäksi nuolten lisääminen kääntymispaloihin auttaa todennäköisesti myös suuntien ymmärtämiseen koirahahmon näkökulmasta katsottuna.

Viimeinen vakavuusluokan 2 käytettävyysongelma on välilehtien käytön hankaluus, joka esiintyi vain yhdellä testiparilla. Välilehdet ovat kuitenkin tärkeä osa Koodikoiran käyttöä. Hankaluutta tuotti lähinnä ymmärtää, minkä välilehden palat ovat kulloinkin näkyvissä; prototyypissä nykyisen välilehden tunnistaa ainoastaan näkyvissä olevien palojen väristä. Välilehtien käyttöä helpottaisi selkeämpi indikaattori siitä, mikä välilehti on valittuna.

Vakavuusluokan 1 käytettävyysongelmilla tarkoitetaan kosmeettisia ongelmia, eli ongelmia, jotka korjataan vain jos projektissa ilmenee ylimääräistä aikaa. Suoritetussa käytettävyytestissä havaittiin yksi tällainen ongelma: tehtävänannon toistamista ei löydetä ilman kehotusta etsiä sitä. Ongelma on kuitenkin todella pieni, sillä tehtävänannot ovat yksinkertaisia ja koko ajan luettavissa. Testihenkilöt eivät missään vaiheessa edes kokeneet tarvitsevansa kuulla tehtävänantoa uudelleen. Ongelman voisi kuitenkin ratkaista lisäämällä valitun koirahahmon, joko Raketin tai Yasminin viereen kaiutinta kuvaavan ikonin.

Vakavuusluokituksessa nolllalla merkitään ongelmia, jotka eivät todellisuudessa ole käytettävyyso ongelmia. Testauksessa nousi esille yksi tällainen ongelma, joka on ohjelman alusta alkamisen ymmärtäminen. Ongelmalle annettiin kuitenkin vakavuusluokitus nolla, koska ajatus on niin olennainen osa ohjelmointia, ettei sitä voi pitää varsinaisena käytettävyyso ongelmana. Lisäksi se esiintyi varsinaisesti vain yhdessä tehtävässä, vaikka kaikki testiparit kohtasivatkin sen. Tämä ”ongelma” on hyvin lähellä aiemmin mainittua käytettävyyso ongelmaa, joka johtuu muokkausten tekemisestä ohjelmaan kelaamatta sitä ensin alkuun. Ratkaisu näihin molempiin on kuitenkin pohjimmiltaan sama; ennen kuin ohjelmaa voi muokata, se täytyy aina kelata ensin alkuun. Näin myös ohjelmoinnillinen ajatus on selkeä: ohjelma aloitetaan aina alusta.

Koodikoirasta löytyi verrattain paljon käytettävyyso ongelmia, mutta se on normaalia testattaessa vielä kehitysvaiheessa olevilla järjestelmillä. Tärkeää onkin ymmärtää syyt käytettävyyso ongelmien takana ja korjata ne ennen kuin ohjelmistokehityksessä edetään pidemmälle. Käytettävyyso ongelmien korjaaminen on järkevintä juuri järjestelmän kehitysvaiheessa, kun resursseja ei ole ehditty käyttämään ylimääräisiin tarkoituksiin. Aikaisessa vaiheessa kerätty tieto järjestelmän käytettävyydestä on noin sata kertaa hyödyllisempää kuin loppuvaiheessa kerätty tieto (Nielsen 2003).

7 JOHTOPÄÄTÖKSET

Tutkielmassa suoritettiin suunnittelututkimus, jonka ensisijaisena tarkoituksena oli luoda toteutettavissa oleva suunnitelma ensisijaisen käytettävyyden omaavasta graafisesta ohjelmointiympäristöstä perusopetukseen. Suunnitelma toteutettiin olemassa olevan kirjallisuuden pohjalta nojaten erityisesti käytettävyyshauristiikkoihin, mutta huomioita kiinnitettiin myös graafiseen suunnitteluun. Tutkimusmenetelminä käytettiin observointia, kyselytutkimusta sekä käytettävyydestausta. Tärkeimpänä prioriteettina läpi tutkielman pidettiin sekä käytettävyyttä että eri asioiden tarkastelemista erityisesti lasten näkökulmasta.

Tieteellisten suunnitteluohjeiden, kuten käytettävyyshauristiikkojen lisäksi graafisten suunnittelun ohjeiden sekä observoinnin ja kyselytutkimuksen avulla luodusta suunnitelmasta rakennettiin hi-fi prototyyppi. Prototyypin avulla suoritettiin käytettävyydestausta, jonka tulosten perusteella suunnitelmaa parannettiin entisestään. Käytettävyyso ongelmia havaittiin verrattain paljon, mikä on normaalia testattaessa kehitysvaiheessa olevalla järjestelmällä. Vakavin käytettävyyso ngelma oli selkeä virhe järjestelmässä, sillä ohjelmointipalojen paikkojen vaihtamien ei toiminut oikein pystysuuntaisen vierityspalkin ollessa olemassa. Vähemmän vakavia ongelmia olivat esimerkiksi eteenpäin- ja taaksepäin-liikkumispalojen sekoittaminen keskenään, palojen yksitellen poistamisen hitaus, koirahahmon ja liikuttavien matkojen ymmärryksen hankaluus, sekä keskelle ohjelmaa palojen lisäämisen mahdottomuus. Edelleen havaittiin pienempiä käytettävyyso ngelmia, joihin löydettiin selkeät ratkaisut. Vaikka käytettävyyso ngelmia löydettiinkin yhteensä useita, suurin osa niistä on helposti korjattavissa. Lisäksi vain yksi ongelma oli niin katastrofaalinen, ettei järjestelmää voisi julkaista ennen sen korjaamista. Kaikkien ehdotettujen korjausten jälkeen järjestelmän käytettävyyso nousee ensiluokkaiselle tasolle.

Havaitut käytettävyyso ngelmat olivat suurimmaksi osaksi järjestelmäkohtaisia, joten saatuja tuloksia ei voida yleistää koskemaan myös muita graafisia ohjelmointiympäristöjä. Kuitenkin tutkielmassa yhteen kerätyt suunnitteluohjeet ovat yleispäteviä.

Yleisenä tuloksena voisi myös todeta testikäyttäjien tietoteknisten taitojen sekä graafisten ohjelmointiympäristöjen käyttötaitojen kasvaneen huomattavasti verrattuna Puntilan (2016) suorittamaan tutkimukseen. Lisäksi tutkielmassa luotu prototyyppi onnistui myös käyttäjäkokemuksen kannalta katsottuna, sillä testihenkilöt kuvasivat sen käytön hauskuutta hymiömittarin avulla käyttämällä pääosin kahta iloisimmista hymynaamoista. Erityisen positiivisesti käyttäjäkokemukseen vaikuttivat sopiva sekoitus suloisia hahmoja ja hullunkurisia ääniä.

Tämä tutkielma osoitti siinä käytettyjen tieteellisten ohjeiden toimivuuden käytännössä sekä käytettävyyden että lasten näkökulmasta tarkasteltuna. Vaikka käytettävyyden konsepti onkin kehitetty lähinnä aikuisia silmällä pitäen, näyttävät samat ohjeet pitävän hyvin paikkansa myös lapsille suunnatuille järjestelmille. Kuitenkaan kehitettäessä järjestelmää lapsille ei pidä jättää huomiotta lasten erityisominaisuuksia, kuten huomiojäljen pituutta, fyysisen kehityksen tasoa, järjestelmän käytön tavoitetta tai navigoinnin hankaluutta. Lisäksi paritestausta osoittautui menetelmänä hyvin tehokkaaksi keinoksi saada lapset toimimaan rennommin käytettävyydestään aikana sekä käyttämään ääneenajattelutekniikka luonnollisemmin.

Jatkossa tässä tutkielmassa yhteen kerättyjä ohjeita lapsille suunnatun järjestelmän suunnittelemisesta käytettävyyden ehdoilla voidaan hyödyntää minkä tahansa järjestelmän kehityksessä, ei pelkästään graafisten ohjelmointiympäristöjen. Lisäksi suoritettujen käytettävyydestausten tulokset antavat suuntaviivoja mahdollisista kohdattavista ongelmista, joita vastaavia suunnitelmia tehtäessä tulee ottaa huomioon. Lisäksi suuri käytännön tulos on Koodikoira itse, jota voidaan käyttää havaittujen käytettävyysongelmien korjaamisen jälkeen kouluissa osana ohjelmoinnin opetusta, mikäli opettajat kokevat järjestelmän riittävän hyödyllisenä.

Tutkimusta rajoittaa kuitenkin jonkin verran se, että käytettävyydetutkimus suoritettiin vain yhdessä koulussa yhden vuosiluokan oppilailla. Tällöin ei voida havaita eri luokkien tai eri koulujen asiaan perehtyneisyydestä johtuvia eroja. Myöskään lasten kehityserojen vaikutusta järjestelmän käytettävyyteen ei voida tulkita järjestämättä samaa käytettävyydestä useille eri vuosiluokille. Lisäksi esitetty suunnitelma on vain

yhden henkilön tulkinta kirjallisuuden ohjeista käytettävyyden ja lasten huomioon ottamisesta järjestelmän suunnittelussa.

Jatkossa aihetta voi tutkia kehittämällä vastaavaa suunnitelmaa entistä pidemmälle. Tällä hetkellä esitetty suunnitelma ei ota huomioon esimerkiksi erilaisia näyttökokoja tai kosketusnäyttöjä; nykyaikaiselta järjestelmältä kuitenkin vaaditaan toimivuutta kaikenlaisilla laitteilla. Erityisesti mobiilikokoisten ja kosketusnäyttöisten laitteiden huomioon ottaminen tärkeää, sillä suuri osa järjestelmien käytöstä tapahtuu nykyään juuri sellaisilla laitteilla. Sekä näytön koko että kosketusominaisuus tuovat useita haasteita, mutta myös mahdollisuuksia järjestelmien kehitykseen. Lisäksi tutkielmassa luodun Koodikoiran käytettävyyttä voisi tutkia laajemmin eri kouluissa sekä eri ikäisten testikäyttäjien avulla. Tällöin saataisiin kattavampi kuva järjestelmän käytettävyyden tasosta. Myös Koodikoiran opettavuuden toimivuutta tulisi tutkia, sillä opetustarkoitukseen suunniteltu järjestelmä on täysin hyödytön, mikäli se ei auta opetuksessa.

LÄHDELUETTELO

- Adiseshiah, E. (2016). *Effective prototyping and the fidelity spectrum* [Verkkodokumentti]. [Viitattu 5.3.2018]. Saatavissa: <https://www.justinmind.com/blog/effective-prototyping-and-the-fidelity-spectrum>.
- Andrzejczak, C. & D. Liu (2010). The effect of testing location on usability testing performance, participant stress levels, and subjective testing experience. *The Journal of Systems and Software* 83: 7, 1258–1266. ISSN 0164-1212.
- Barnum, C. (2002). *Usability testing and research*. New York: Longman. 428 s. ISBN 0-205-31519-4.
- Barnum, C. (2011). *Usability Testing Essentials*. Burlington: Morgan Kaufmann. 328 s. ISBN 978-0-12-375092-1.
- Cao, J. (2015). *Choosing Your Usability Tests and Participants* [Verkkodokumentti]. [Viitattu 1.2.2018]. Saatavissa: <https://www.sitepoint.com/choosing-usability-tests-participants>.
- Cann, M. (2017). *10 inspirational graphic design trends for 2018* [Verkkodokumentti]. [Viitattu 28.2.2018]. Saatavissa: <https://99designs.com/blog/trends/graphic-design-trends-2018>.
- Carr, C. (2016). *The Pros and Cons of Conducting Usability Testing with Multiple Participants at the Same Time* [Verkkodokumentti]. [Viitattu 31.1.2018]. Saatavissa: <http://www.bunnyfoot.com/blog/2016/11/the-pros-and-cons-of-conducting-usability-testing-with-multiple-participants-at-the-same-time>.

- Cass, J. (2017). 2018 *Graphic Design Trends + Inspirational Showcase* [Verkkodokumentti]. [Viitattu 28.2.2018]. Saatavissa: <http://justcreative.com/2017/11/28/graphic-design-trends-2018>.
- Carvajal, L., A. Moreno, M. Sánchez-Segura & A. Seffah (2013). Usability through Software Design. *IEEE Transactions on Software Engineering* 39: 11, 1582–1596.
- Cousins, C. (2014). *Website Design for Kids: Tips and Advice* [Verkkodokumentti]. [Viitattu 22.1.2018]. Saatavissa: <https://designshack.net/articles/graphics/website-design-for-kids-tips-and-advice>.
- Dalrymple, B. (2016). *First-Time Usability: Analyzing Results* [Verkkodokumentti]. [Viitattu 14.2.2018]. Saatavissa: <https://medium.com/user-research/first-time-usability-analyzing-results-7eb748ede9eb>.
- Dehouck, R. (2015). *The maturity of visual programming* [Verkkodokumentti]. [Viitattu 23.1.2018]. Saatavissa: <http://craft-ai-f366c44b33.drafts.github.io/blog/the-maturity-of-visual-programming>.
- Deterding, S., D. Dixon, R. Khaled & L. Nacke (2011). From Game Design Elements to Gamefulness: Defining “Gamification”. *MindTrek '11*, 9–15. ISBN 978-1-4503-0816-8.
- Donmoyer, R. (2008). Quantitative Research. Teoksessa: *The Sage Encyclopedia of Qualitative Research Methods*, 713–718. Given, L. Sage Publications, Inc. ISBN 978-1-4129-4163-1.
- Druin, A., B. Bederson, A. Boltman, A. Miura, D. Knotts-Callahan & A. Platt (1999). Children as Our Technology Design Partners. Teoksessa: *The design of children's interactive technologies*, 51–72. Druin, A. San Francisco: Morgan Kaufman Publishers.

- Duggirala, S. (2016). *10 Usability Heuristics with Examples* [Verkkodokumentti]. [Viitattu 1.3.2018]. Saatavissa: <https://blog.prototypr.io/10-usability-heuristics-with-examples-4a81ada920c>.
- Dumas, J. & J. Redish (1999). *A Practical Guide to Usability Testing*. Revised edition. Yhdysvallat: Intellect Books. ISBN 1-84150-020-8.
- Dunn, T. (2006). *Usability testing with children* [Verkkodokumentti]. [Viitattu 30.1.2018]. Saatavissa: <https://www.webcredible.com/blog/usability-testing-children>.
- Fitz-Walter, Z., D. Johnson, P. Wyeth, D. Tjondronegoro & B. Scott-Parker (2017). Driven to drive? Investigating the effect of gamification on learner driver behavior, perceived motivation and user experience. *Computers in Human Behavior* 71, 589–595. ISSN 0747-5632.
- Froats, L. (2013). *How Do I Create a Good Questionnaire?* [Verkkodokumentti]. [Viitattu 24.1.2018]. Saatavissa: <https://www.simplycast.com/blog/how-do-i-create-a-good-questionnaire>.
- Gabriel-Petit, P. (2018). *Glossary: User Experience* [Verkkodokumentti]. [Viitattu 22.1.2018]. Saatavissa: <https://www.uxmatters.com/glossary>.
- Gould, J., S. Boies & C. Lewis (1991). Making usable, useful, productivity-enhancing computer applications. *Communications of the ACM* 34: 1, 74–85. ISSN 0001-0782.
- Gould, J. & C. Lewis (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM* 28: 3, 300–311. ISSN 0001-0782.

- Hanna, L., K. Ridsen & K. Alexander (1997). Guidelines for Usability Testing with Children. *Interactions* 4: 5, 9–14. ISSN 1072-5520.
- Hevner, A., S. March & J. Park (2004). Design Science in Information Systems Research. *MIS Quarterly* 28: 1, 75–106. ISSN 0276-7783.
- Hietala, P. & S. Ovaska (2002). Lapset ja käyttöliittymät – johdatus aihepiiriin. Teoksessa: *Lasten käyttöliittymät*, 1–19. Hietala, P. & S. Ovaska. Tampereen yliopiston laitosten julkaisut. ISBN 951-44-5446-4.
- Höysniemi, J. (2005). Käytettävyytestaus lasten kanssa. Teoksessa: *Käytettävyystudkimuksen menetelmät*, 259–282. Ovaska, S., A. Aula & P. Marjaranta. Tampereen yliopiston laitosten julkaisut. ISBN 951-44-6230-0.
- Idler, S. (2013). *5 Key Criteria Of A Good User Experience For Children* [Verkkodokumentti]. [Viitattu 22.1.2018]. Saatavissa: <http://uxkids.com/blog/5-key-criteria-of-a-good-user-experience-for-children>.
- Idler, S. (2014). *Comparing Usability for Kids and Adults (part 2)* [Verkkodokumentti]. [Viitattu 20.11.2017]. Saatavissa: <http://uxkids.com/blog/comparing-usability-for-kids-and-adults-part-2>.
- Ilves, M. (2005). Ääneenajattelu. Teoksessa: *Käytettävyystudkimuksen menetelmät*, 209–222. Ovaska, S., A. Aula & P. Marjaranta. Tampereen yliopiston laitosten julkaisut. ISBN 951-44-6230-0.
- ISO (1998). *ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 11: Guidance on usability*. Geneva, Sveitsi.

- Jaspers, M., T. Steen, C. van den Bos & M. Geenen (2004). The think aloud method: a guide to user interface design. *International Journal of Medical Informatics* 73: 11–12, 781–795.
- Juristo, N., A. Moreno & M. Sánchez-Segura (2007). Guidelines for Eliciting Usability Functionalities. *IEEE Transactions on Software Engineering* 33: 11, 744–758.
- Järvinen, P. (2017). Two different goals in design science research, one from science another from practice. *Informaatitieteiden yksikön raportteja* 52. ISBN 978-952-03-0401-0.
- Karch, M. (2017). *7 Free Programming Languages to Teach Kids How to Code* [Verkkodokumentti]. [Viitattu 20.11.2017]. Saatavissa: <https://www.lifewire.com/kids-programming-languages-4125938>.
- Kekäläinen, O. (2014). *Code.org oppimateriaalina Suomessa* [Verkkodokumentti]. [Viitattu 23.1.2017]. Saatavissa: <https://koodikerho.fi/blog/2014/10/21/code-org-oppimateriaalina-suomessa>.
- Khalifa, M. & R. Lam (2002). Web-based learning: effects on learning process and outcome. *IEEE Transactions on Education* 45: 4, 350–356.
- Kokkonen, A. & A. Lehtinen (2000). *Hyvää käytettävyyttä, hyvää oppimista?* [Verkkodokumentti]. [Viitattu 20.11.2017]. Saatavissa: http://www.sis.uta.fi/ipopp/ipopp2000/AhtinenKokkonen/kaytettavyys_oppiminen.html.
- Koodikerho (2017). *Ohjelmointiopetusta Suomessa* [Verkkodokumentti]. [Viitattu 20.11.2017]. Saatavissa: <https://koodikerho.fi/ohjelmointiopetusta-suomessa>.

- Koskinen, J. (2005). Käytettävyysestaus. Teoksessa: *Käytettävyytutkimuksen menetelmät*, 187–208. Ovaska, S., A. Aula & P. Marjaranta. Tampereen Yliopistopaino Oy: Juvenes Print. ISBN 951-44-6230-0.
- Lewis, C. & J. Rieman (1994). *Task-centered User Interface Design* [Verkkodokumentti]. [Viitattu 31.1.2018]. Saatavissa: <http://hcibib.org/tcuid/tcuid.pdf>.
- Liukas, L. & J. Mykkänen (2017). *Koodi2016 yhdeksässä kohdassa* [Verkkodokumentti]. [Viitattu 20.11.2017]. Saatavissa: <http://koodi2016.fi>.
- Logan, B. (2009). *Usability for Children* [Verkkodokumentti]. [Viitattu 18.1.2018]. Saatavissa: <https://www.spotless.co.uk/blog/usability-for-children>.
- Loranger, H. (2016). *Checklist for Planning Usability Studies* [Verkkodokumentti]. [Viitattu 1.2.2018]. Saatavissa: <https://www.nngroup.com/articles/usability-test-checklist>.
- Lyonnais, S. (2017). *Do the 10 Usability Heuristics Still Hold Up Over Two Decades Later?* [Verkkodokumentti]. [Viitattu 5.3.2018]. Saatavissa: <https://theblog.adobe.com/do-the-10-usability-heuristics-still-hold-up-over-a-decade-later>.
- Mander, J. (2017). *How to Use Qualitative and Quantitative Research to Your Advantage* [Verkkodokumentti]. [Viitattu 24.1.2017]. Saatavissa: <https://blog.globalwebindex.net/marketing/qualitative-vs-quantitative>.
- McCready, R. (2018). *8 New Graphic Design Trends That Will Take Over 2018* [Verkkodokumentti]. [Viitattu 28.2.2018]. Saatavissa: <https://venngage.com/blog/graphic-design-trends>.

- Morrison, C. (2018). *10 Simple Steps to Start User Testing by Tomorrow (and Why You Should)* [Verkkodokumentti]. [Viitattu 8.2.2018]. Saatavissa: <https://usabilityhour.com/start-user-testing>.
- Mortensen, D. (2018). *The Basics of Recruiting Users for Usability Testing* [Verkkodokumentti]. [Viitattu 1.2.2018]. Saatavissa: <https://www.interaction-design.org/literature/article/the-basics-of-recruiting-users-for-usability-testing>.
- Nam, Heather (2010). *Designing User Experiences for Children* [Verkkodokumentti]. [Viitattu 22.1.2018]. Saatavissa: <https://www.uxmatters.com/mt/archives/2010/05/designing-user-experiences-for-children.php>.
- Nielsen, J. (1993). *Usability Engineering*. Yhdysvallat: Academic Press. 362 s. ISBN 978-0125184069.
- Nielsen, J. (1994). Usability laboratories. *Behaviour & Information Technology* 13: 1–2, 3–8. ISSN 0144-929X.
- Nielsen, J. (1995). *10 Usability Heuristics for User Interface Design* [Verkkodokumentti]. [Viitattu 16.1.2018]. Saatavissa: <https://nngroup.com/articles/ten-usability-heuristics>.
- Nielsen, J. (2000). *Why You Only Need to Test with 5 Users* [Verkkodokumentti]. [Viitattu 1.2.2018]. Saatavissa: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users>.
- Nielsen, J. (2001). *First Rule of Usability? Don't Listen to Users* [Verkkodokumentti]. [Viitattu 29.1.2018]. Saatavissa: <https://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users>.

- Nielsen, J. (2003). *Paper Prototyping: Getting User Data Before You Code* [Verkkodokumentti]. [Viitattu 4.4.2018]. Saatavissa: <https://www.nngroup.com/articles/paper-prototyping>.
- Nielsen, J. (2012). *Thinking Aloud: The #1 Usability Tool* [Verkkodokumentti]. [Viitattu 31.1.2018]. Saatavissa: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool>.
- Norman, D. (2005). Human-Centered Design Considered Harmful. *Interactions* 12: 4, 14–19. ISSN 1072-5520.
- Norman, D. & J. Nielsen (2018). *The Definition of User Experience (UX)* [Verkkodokumentti]. [Viitattu 22.1.2018]. Saatavissa: <https://www.nngroup.com/articles/definition-user-experience>.
- Opetushallitus (2014). *Perusopetuksen opetussuunnitelman perusteet 2014*. 4. painos. Helsinki: Opetushallitus. 473 s. Saatavissa: http://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf.
- Opetushallitus (2017). *Ohjelmointi* [Verkkodokumentti]. [Viitattu 23.1.2018]. Saatavissa: http://www.edu.fi/perusopetus/matematiikka/ops2016_tukimateriaalit/ohjelmointi.
- Pernice, K. (2016). *UX Prototypes: Low Fidelity vs. High Fidelity* [Verkkodokumentti]. [Viitattu: 5.3.2018]. Saatavissa: <https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity>.
- Perälä, R. (2005). Arvioijan vaikutus. Teoksessa: *Käytettävyyystutkimuksen menetelmät*, 299–312. Ovaska, S., A. Aula & P. Marjaranta. Tampereen Yliopistopaino Oy: Juvenes Print. ISBN 951-44-6230-0.

- Puntila, J. (2016). *Käytettävyystutkimus visuaalisesta ohjelmointiympäristöstä perusopetuksessa: case Scratch*. Opinnäytetyö. Vaasan yliopisto. 42 s.
- Pönkä, H. (2008). *Design-tutkimus* [Verkkodokumentti]. [Viitattu 21.2.2018]. Saatavissa: <https://www.slideshare.net/hponka/designitutkimus>.
- Quibly, J. (2013). *Designing A Child-Friendly Website: A True Challenge* [Verkkodokumentti]. [Viitattu 18.1.2018]. Saatavissa: <https://usabilitygeek.com/designing-child-friendly-website-true-challenge>.
- Read, J., S. MacFarlane & C. Casey (2002). Endurability, Engagement and Expectations: Measuring Children's Fun. *Interaction Design and Children* 2, 1–23.
- Rigby, S. & R. Ryan (2011). *Glued to Games: How Video Games Draw Us In and Hold Us Spellbound*. Yhdysvallat: Greenwood Publishing Group. 186 s. ISBN 978-0-313-36224-8.
- Rohan, E. (2012). Spss for Real Beginners. Teoksessa: *Approaches to Quantitative Research*. 12–25. Chen, H. Irlanti: Oak Tree Press. ISBN 978-7-8119-059-3.
- Rosemberg, C. (2018). *Turning Usability Testing Data into Action without Going Insane* [Verkkodokumentti]. [Viitattu 14.2.2018]. Saatavissa: <https://www.toptal.com/designers/usability/turning-usability-testing-data-into-action>.
- Rubin, J. & D. Chisnell (2008). *Handbook of Usability Testing : How to Plan, Design, and Conduct Effective Tests*. 2. painos. Indianapolis: Wiley Publishing, Inc. 386 s. ISBN 978-0-470-18548-3.
- Schmidt, A. & A. Etches (2012). *User Experience (UX) Design for Libraries*. Yhdysvallat: ALA Editions. 105 s. ISBN 978-1-55570-781-1.

- Sheragy, K. (2018). *Pros & Cons of Qualitative & Quantitative Research* [Verkkodokumentti]. [Viitattu 24.1.2018]. Saatavissa: <http://classroom.synonym.com/pros-cons-qualitative-quantitative-research-8179604.html>.
- Sim, G., S. MacFarlane & J. Read (2006). All work and no play: Measuring fun, usability, and learning in software for children. *Computers & Education* 46: 3, 234–348.
- Someren, van, M., Y. Barnard & J. Sandberg (1994). *The think aloud method: a practical approach to modelling cognitive processes*. Lontoo: Academic Press. 209 s. ISBN 0-12-714270-3.
- StatCounter (2017). StatCounter GlobalStats [Verkkodokumentti]. [Viitattu 4.12.2017]. Saatavissa: <http://gs.statcounter.com>.
- Technopedia (2018). *Visual Programming Language (VPL)* [Verkkodokumentti]. [Viitattu 23.1.2018]. Saatavissa: <https://www.techopedia.com/definition/22855/visual-programming-language-vpl>.
- Toikkanen, T. (2015). *10 teesiä koodauksesta koulussa* [Verkkodokumentti]. [Viitattu 20.11.2017]. Saatavissa: <http://koodiaapinen.fi/2015/06/10-teesia-koodauksesta-koulussa>.
- Tondello, G. (2016). An introduction to gamification in human-computer interaction. *XRDS: Crossroads, The ACM Magazine for Students* 23: 1, 15–17. ISSN 1528-4972.
- Triner, F. (2018). *2018 Design Trends* [Verkkodokumentti]. [Viitattu 28.2.2018]. Saatavissa: <https://www.behance.net/gallery/60273889/2018-Design-Trends>.

Usability.gov (2018). *Recruiting Usability Test Participants* [Verkkodokumentti]. [Viitattu 1.2.2018]. Saatavissa: <https://www.usability.gov/how-to-and-tools/methods/recruiting-usability-test-participants.html>.

Winter, N. (2015). *How Gamification Can Improve UX* [Verkkodokumentti]. [Viitattu 15.2.2018]. Saatavissa: <http://blog.usabilla.com/how-gamification-can-improve-ux>.

LIITTEET

LIITE 1. Kyselylomake

KYSELYTUTKIMUS
JANNE PUNTILA

päivämäärä: _____ 2018

luokka: _____

1. Mikä on lempivärisi?

2. Mitä harrastat tai haluaisit harrastaa? Voit mainita useammankin.

3. Kuka on lempihahmosi? (esim. supersankari, sarjakuvahahmo, elokuvahahmo, jne.)

4. Kuka on idolisi? (esim. perheenjäsen, youtube-julkkis, näyttelijä, laulaja, urheilija, jne.)

5. Mikä on paras asia maailmassa?

6. Ympyröi seuraavista kuvioista se, joka kuvaa mielestäsi parhaiten...

a) liikkumista



b) uuden asian lisäämistä



c) kääntymistä



7. Ympyröi seuraavista sanoista se, joka kuvaa mielestäsi parhaiten...

a) etenemistä yhden ruudun verran

askel

ruutu

sentti

hyppy

b) jaoteltua tietoa sisältäviä paikkoja

välilehti

sivu

lehti

taso

c) tietokoneohjelman käyttämisen aloittamista

ajaminen

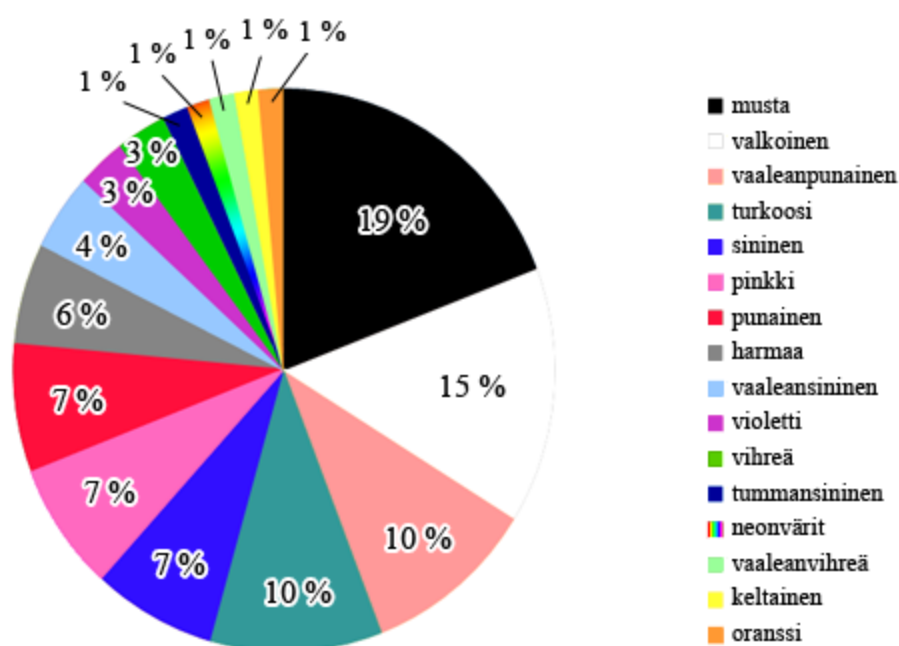
aloittaminen

kokeileminen

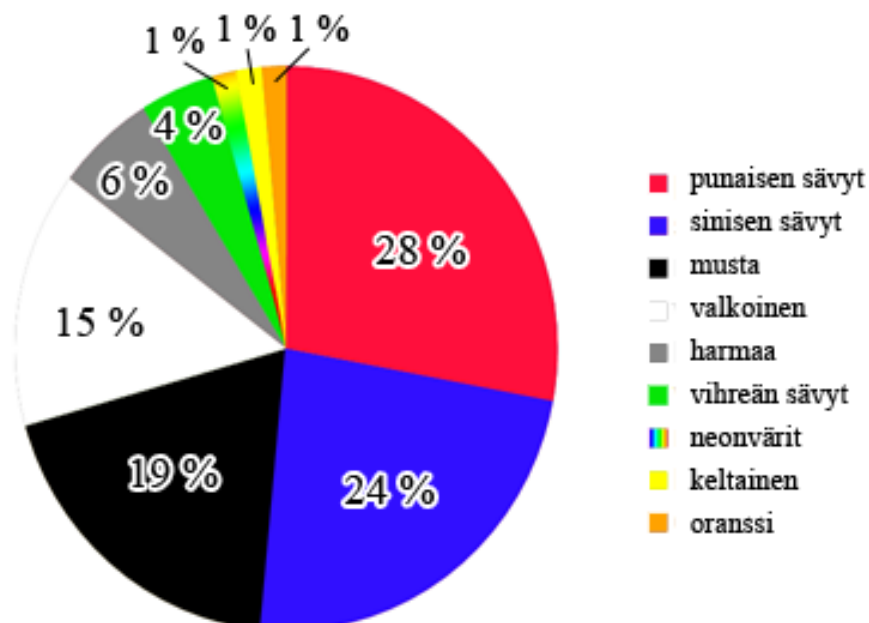
käynnistäminen

LIITE 2. Kyselylomakkeen tulokset

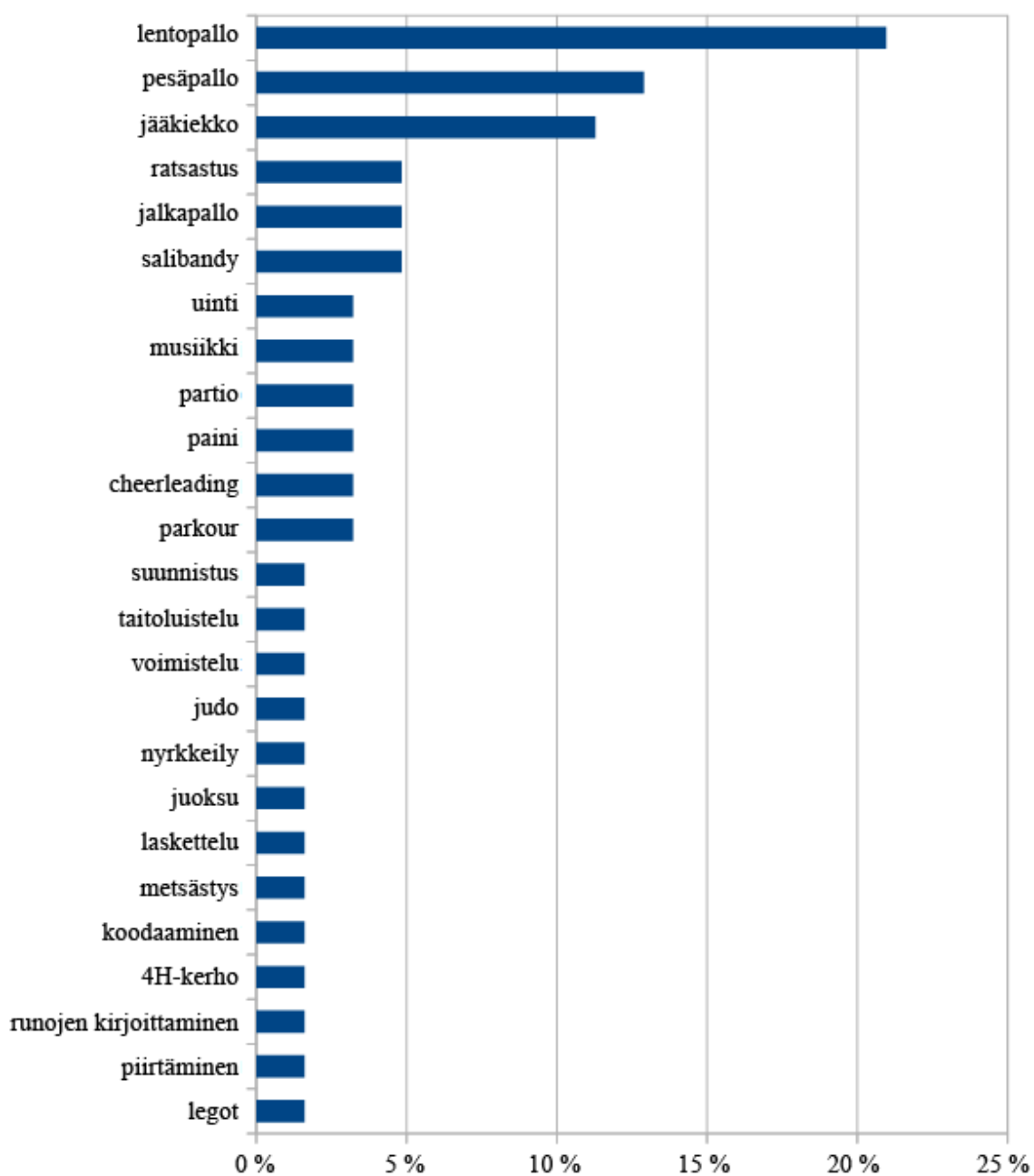
Lasten lempivärit kyselytutkimuksen mukaan.



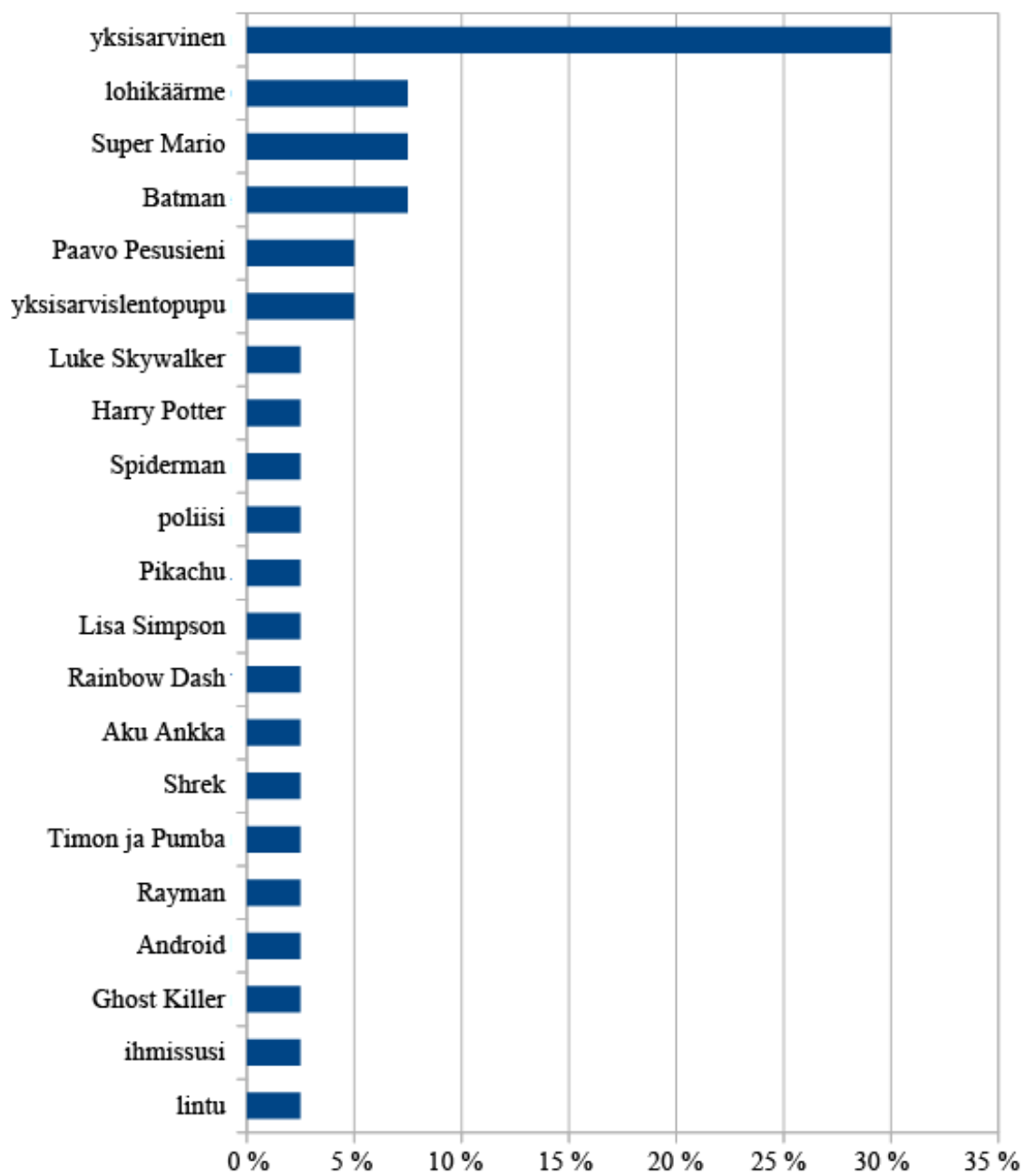
Lasten lempivärit kyselytutkimuksen mukaan. Eri sävyt yhdistetty.



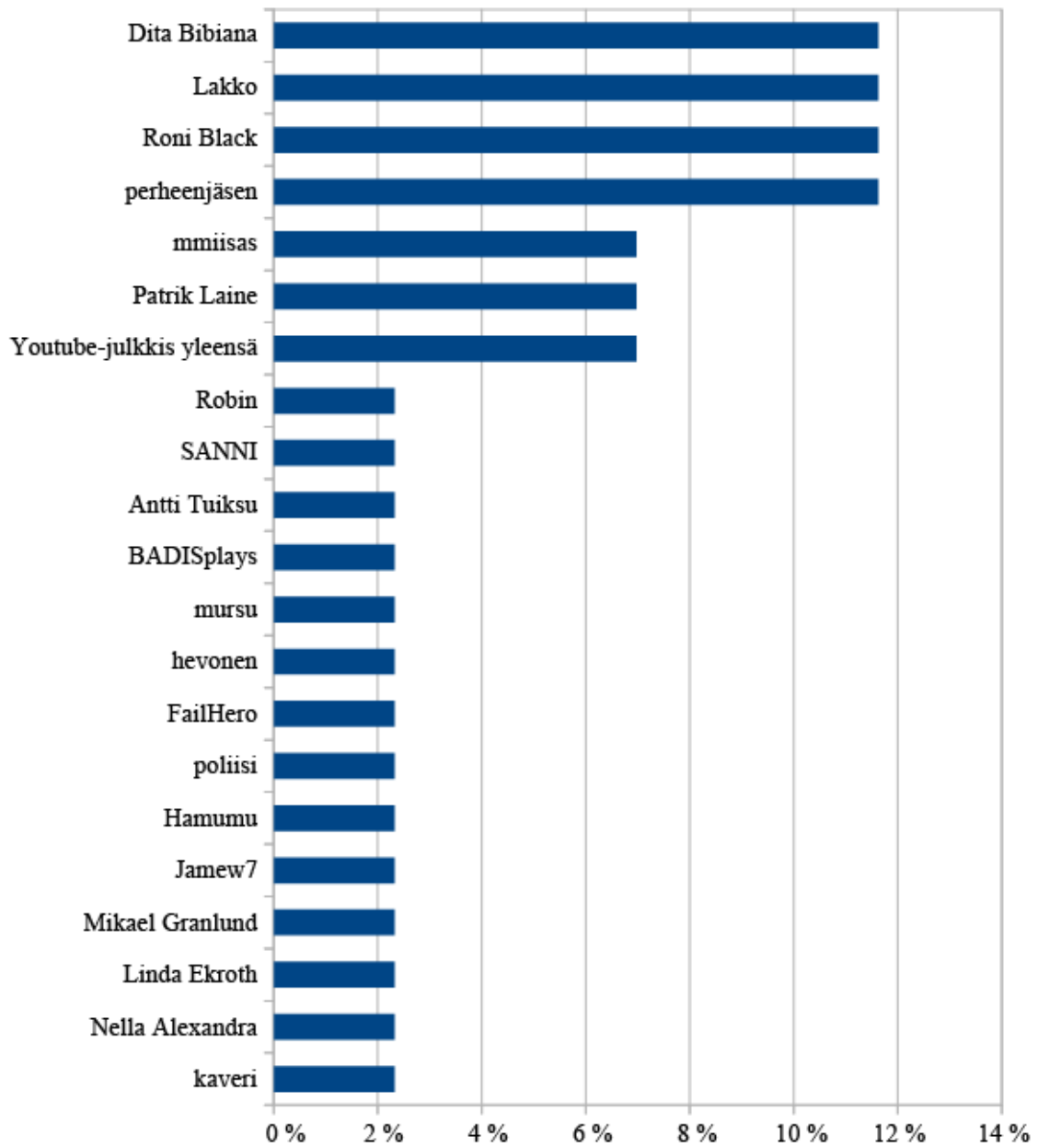
Lasten harrastukset kyselytutkimuksen mukaan.



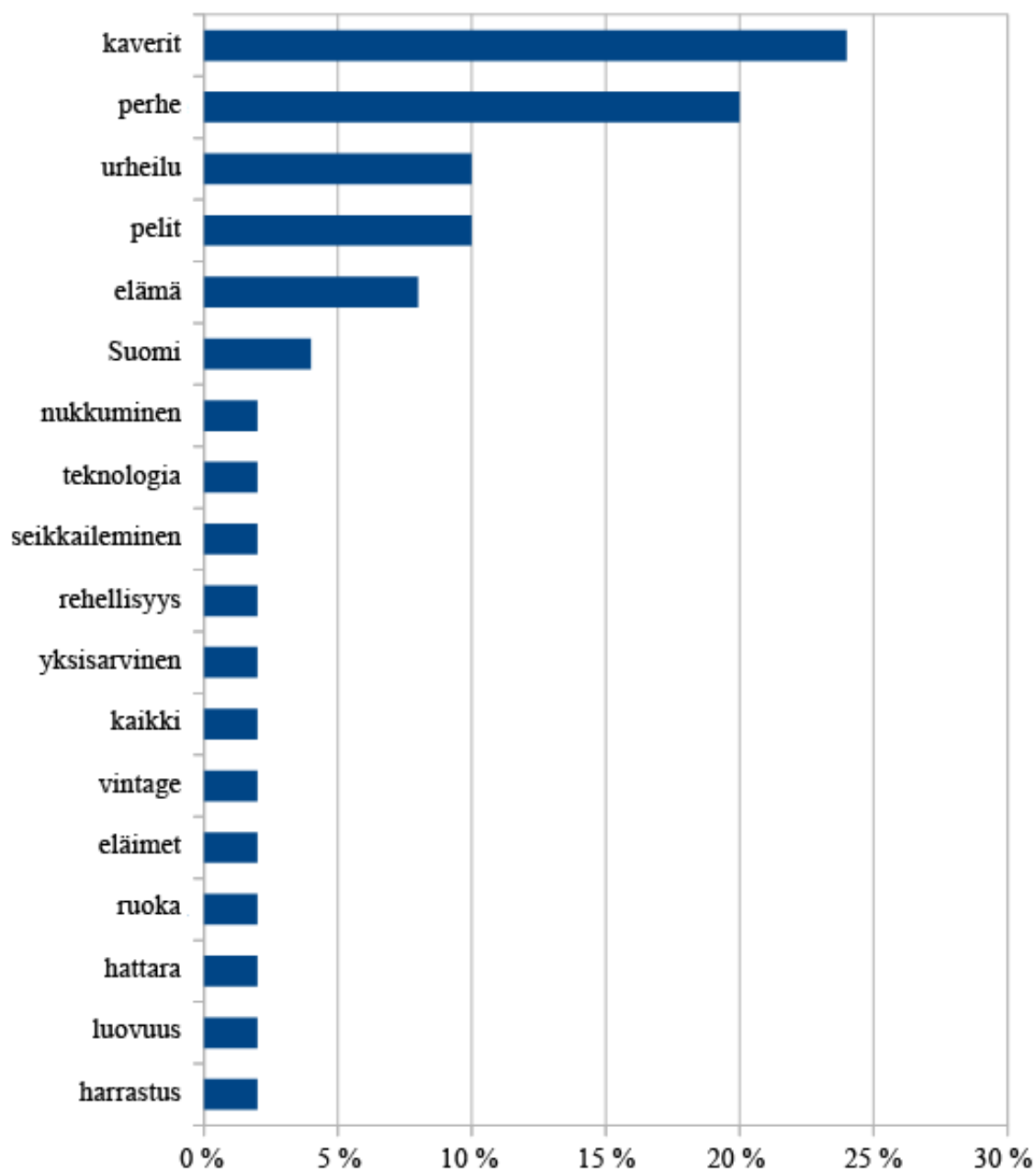
Lasten lempihahmot kyselytutkimuksen mukaan.



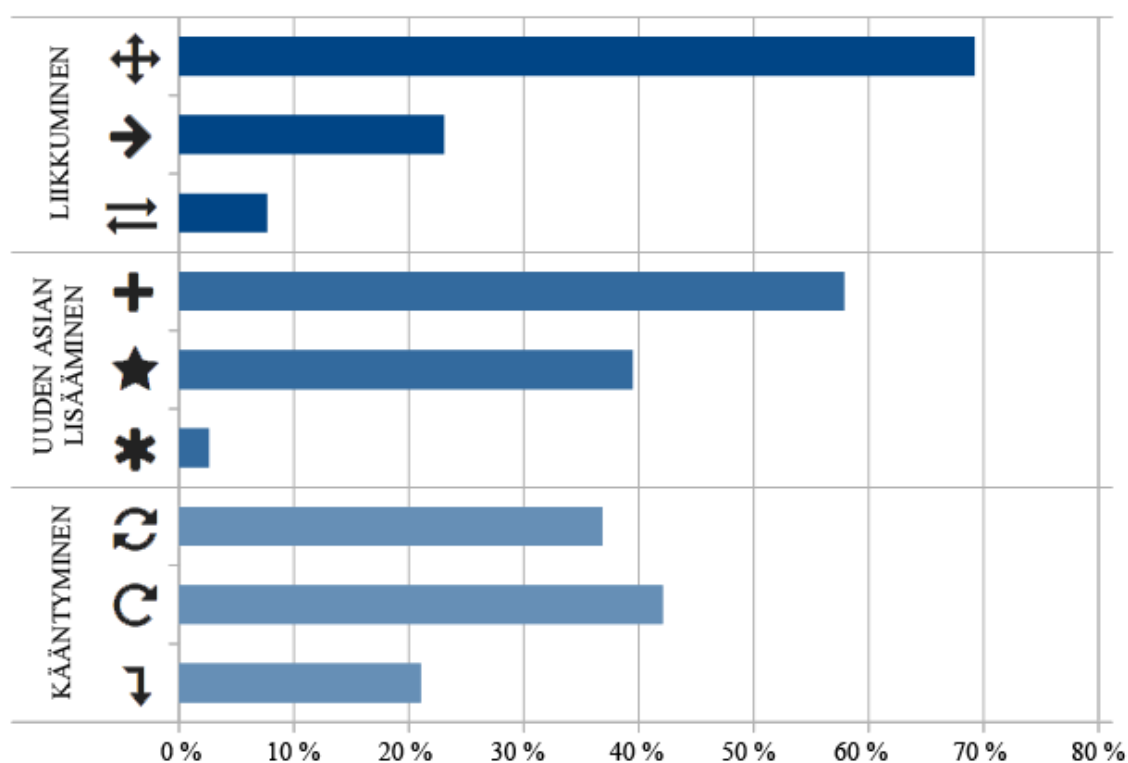
Lasten idolit kyselytutkimuksen mukaan.



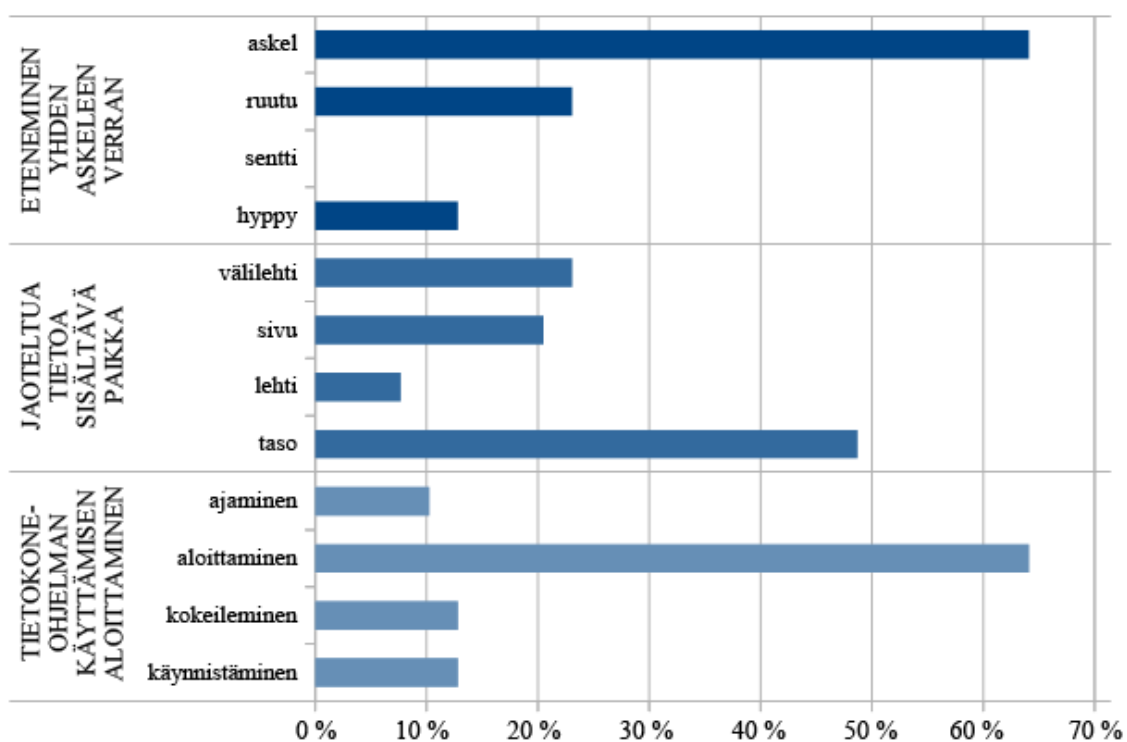
Lasten mielestä parhaat asiat maailmassa kyselytutkimuksen mukaan.



Kyselytutkimuksen mukaan lasten mielestä kuvaavimmat ikonit kolmelle toiminnolle.



Kyselytutkimuksen mukaan lasten mielestä kuvaavimmat sanat neljälle asialle.



LIITE 3. Testihenkilöiden huoltajille lähetetty tiedote ja lupalomake

Sivu 1/2

Hyvät huoltajat!

Olen Janne Puntila – 24-vuotias tietotekniikan opiskelija Vaasan yliopistosta. Pääsin ylioppilaaksi Jurvan lukiosta keväällä 2012. Olen loppusuoralla suorittamassa kauppatieteiden maisterin tutkintoa, ja pro gradu -tutkielmani aiheena on graafiset ohjelmointiympäristöt sekä ohjelmistosuunnittelu käytettävyyden ehdoilla. Tarkemmin sanottuna tarkoitukseni on suunnitella uusi graafinen ohjelmointiympäristö ohjelmoinnin perusopetukseen.

Tärkeä osa tutkielmaa on käytännön käytettävyydestä, johon tarvitsisin 4. luokkalaisia testihenkilöitä. Sana ”testihenkilö” on varsin harhaanjohtava, sillä tutkimuksessa mitataan vain ja ainoastaan ohjelmointiympäristöä, **ei oppilaita**.

Noin puoli tuntia kestävässä testitilanteessa oppilaille annetaan yksinkertaisia tehtäviä suoritettavaksi tekemälläni ohjelmointiympäristön prototyypillä pareittain. Oppilaiden tulisi testin aikana puhua ääneen mahdollisimman paljon mitä he ajattelevat;

- ”*missäs se nappi olikaan*”
- ”*nyt lisätään tähän toinen pala*”
- ”*jos laitetaan näin, niin toi hahmo liikkuu ylös*” jne...

Olen itse mukana testissä mukana antamassa ohjeita, kannustamassa ja tekemässä muistiinpanoja toiminnasta. Testin voi halutessaan lopettaa missä tahansa vaiheessa. Mikäli hyväksytte, testitilanne myös äänitetään ja videokuvataan. Tämä helpottaisi huomattavasti tulosten analysoinnissa. Tutkimus olisi tarkoitus suorittaa noin viikolla 10.

Tärkeä muistaa:

- Tulokset esitetään täysin anonyymisti. Testihenkilöitä ei voida saada selville.
- Mitään kuvattua/äänitettyä materiaalia ei missään tapauksessa näytetä muille kuin itselleni. Materiaali tuhoetaan analysoinnin jälkeen.
- Oppilaita itseään ei testata.

Olisin todella kiitollinen avustanne tässä asiassa!

Ystävällisin terveisin

Janne Puntila

Oppilas _____

- saa osallistua sivulla 1 mainittuun käytettävyytestaukseen.
- Testitilanteen saa äänittää.
- Testitilanteen saa videokuvata.

Huoltajan allekirjoitus _____

LIITE 4. Käytettävyydestauksen testihenkilöt

TESTIPARI 1		
	Testihenkilö 1	Testihenkilö 2
Ikä:	11	10
Sukupuoli:	tyttö	tyttö
Oletko käyttänyt graafisia ohjelmointiympäristöjä?	kyllä, Code.org	kyllä, Code.org
Kuinka paljon pelaat pelejä?	ei ollenkaan	vain silloin tällöin
Viimeaikainen matematiikan numero:	9	9
TESTIPARI 2		
	Testihenkilö 1	Testihenkilö 2
Ikä:	10	10
Sukupuoli:	poika	poika
Oletko käyttänyt graafisia ohjelmointiympäristöjä?	kyllä, Code.org	kyllä, Code.org
Kuinka paljon pelaat pelejä?	vain silloin tällöin	lähes päivittäin
Viimeaikainen matematiikan numero:	9	10
TESTIPARI 3		
	Testihenkilö 1	Testihenkilö 2
Ikä:	10	11
Sukupuoli:	tyttö	tyttö
Oletko käyttänyt graafisia ohjelmointiympäristöjä?	kyllä, Scratch ja Code.org	kyllä, Scratch ja Code.org
Kuinka paljon pelaat pelejä?	lähes päivittäin	lähes päivittäin
Viimeaikainen matematiikan numero:	9	9

TESTIPARI 4		
	Testihenkilö 1	Testihenkilö 2
Ikä:	10	11
Sukupuoli:	poika	poika
Oletko käyttänyt graafisia ohjelmointiympäristöjä?	kyllä, Code.org	kyllä, Code.org
Kuinka paljon pelaat pelejä?	lähes päivittäin	päivittäin
Viimeaikainen matematiikan numero:	10	8
TESTIPARI 5		
	Testihenkilö 1	Testihenkilö 2
Ikä:	10	10
Sukupuoli:	poika	poika
Oletko käyttänyt graafisia ohjelmointiympäristöjä?	kyllä, Scratch	kyllä, Scratch
Kuinka paljon pelaat pelejä?	vain silloin tällöin	päivittäin
Viimeaikainen matematiikan numero:	9	9
TESTIPARI 6		
	Testihenkilö 1	Testihenkilö 2
Ikä:	11	11
Sukupuoli:	tyttö	tyttö
Oletko käyttänyt graafisia ohjelmointiympäristöjä?	kyllä, Code.org	kyllä, Code.org
Kuinka paljon pelaat pelejä?	vain silloin tällöin	vain silloin tällöin
Viimeaikainen matematiikan numero:	10	9