

Karoliina Savolainen

**Katsaus ohjelmoinnin opettamisen tavoitteisiin
alakoulussa**

Tietotekniikan kandidaatintutkielma

29. huhtikuuta 2017

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Karoliina Savolainen

Yhteystiedot: karoliina.j.savolainen@student.jyu.fi

Ohjaaja: Sanna Juutinen

Työn nimi: Katsaus ohjelmoinnin opettamisen tavoitteisiin alakoulussa

Title in English: Review of the objectives of teaching programming in Finnish primary schools

Työ: Kandidaatintutkielma

Sivumäärä: 21+0

Tiivistelmä: Ohjelmointi tuli uutena osana vuoden 2014 opetussuunnitelmaan ja perusopetuksessa se otettiin käyttöön syksyllä 2016. Aiemmin on todettu, että tieto- ja viestintäteknologiaa opettavilla opettajilla on puute hyvistä oppimateriaaleista. Tämän kandidaatintutkielman tarkoituksena oli selvittää opetussuunnitelmaan peilausten, millä tavoin ja mitä opetusmateriaaleja käyttäen alakouluissa opetetaan ohjelmointia ja algoritmista ajattelua. Kandidaatintutkielman tulosten perusteella voidaan todeta, että oppimateriaaleja ei ole laajasti tarjolla ohjelmoinnin opettamiseen alakouluissa ja opetuksen sisältö tulee olemaan opettaja- ja koulukohtaista tavoitteiden kannalta.

Avainsanat: alakoulu, ohjelmointi, opetus

Abstract: Programming was made a part of new Finnish curriculum in 2014. Previous research has pointed out a lack of proper ICT teaching materials. This bachelor thesis was made to explore what kind of programming study materials are used and how algorithmic thinking and programming are taught in Finnish primary schools. The results show the number of good quality study materials to be low and content of programming lessons to be school-specific with little nationwide constraints.

Keywords: primary school, programming, teaching

Kuviot

- Kuvio 1. Code.orgin tehtävän suoritettuaan voi katsoa valmista ohjelmakoodia. .10
- Kuvio 2. Scratchin työskentelytila. Ohjeet ohjelman tekemiseen löytyvät oikeasta reunasta, koodialue on sen vieressä ja komentoalue löytyy keskeltä. Vasemmalla on näyttämö.....11

Sisältö

1	JOHDANTO	1
2	OHJELMOINNIN OPETTAMINEN: MIKSI?.....	3
3	OPETUSSUUNNITELMAT JA OHJELMOINTI.....	4
3.1	Valtakunnallinen OPS	4
3.1.1	Laaja-alainen osaaminen.....	4
3.1.2	Ohjelmointi oppiaineissa	5
3.2	Jyväskylän alakoulujen OPS	5
3.2.1	Vuosiluokat 1-2.....	6
3.2.2	Vuosiluokat 3-6.....	6
3.3	Jyväskylän yliopiston opettajankoulutuslaitoksen luokanopettaja- koulutuksen OPS	6
4	KÄYTÖSSÄ OLEVAT OPPIMATERIAALIT	8
4.1	Code.org	8
4.2	Scratch ja Scratch Jr	10
4.3	Tavoitteiden toteutuminen oppimateriaaleissa	12
4.4	Yhteenvedo oppimateriaaleista	12
5	YHTEENVETO	14
	LÄHTEET.....	16

1 Johdanto

Syksyllä 2016 otettiin käyttöön esi- ja perusopetuksessa uusi opetussuunnitelma, jossa huomiota herättävä lisäys on ohjelmoinnin sisällyttäminen peruskoulun opetuksessa ja se sisällytetään suurimmaksi osaksi matematiikan oppisisältöön. Tietotekniikan opetuksessa on ollut vain muutama varsinainen oppikirja viimeisen vuosikymmenen aikana, ja tieto- ja viestintäteknologiaa opettavat kärsivätkin hyvien oppimateriaalien puutteesta (Ekonoja 2014, s. 16). Koska tietotekniikka ei ole oma erillinen opetettava oppiaine alakoulussa, on tietotekniikan opettaminen integroitu muihin oppiaineisiin ja täten myös luokanopettajakoulutuksessakin tietotekniikkaa käsitellään muiden oppiaineiden kursseilla.

Tutkimuskysymyksiä on tässä tutkielmassa kaksi:

- Millä tavoin ja mitä oppimateriaaleja käyttäen perusopetukseen sisällytetään ohjelmoinnin alkeita ja algoritmisen ajattelun kehittämistä?
- Millä tavoin Jyväskylän yliopiston luokanopettajaopiskelijat tutustuvat ohjelmointiin osana matematiikkaa?

Kandidaatintutkielma on toteutettu kirjallisuuskatsauksena. Aiheen ajankohtaisuuden takia ei aihetta käsittelevää tutkimusta löydy juurikaan. Jyväskylän yliopistossa on tehty muun muassa vuonna 2015 kandidaatintutkielma Ohjelmointisatukirjat (Potila 2015), jossa tutkitaan julkaistuja ohjelmointisatukirjoja ja opetussuunnitelman tavoitteiden toteutumista näissä teoksissa.

Luvussa 2 pohjustan uuden opetussuunnitelman uutta osaa, eli ohjelmoinnin sisällyttämistä opetukseen ja pyrin avaamaan sitä, mitkä asiat ovat olleet taustalla ohjelmoinnin tuomiseksi uuteen opetussuunnitelmaan ja osaksi opetusta.

Luvussa 3 tutkin kolmen eri tason opetussuunnitelmia: miten ohjelmointi ja algoritmisen ajattelu ja sen kehittäminen nousevat esille valtakunnallisella ja paikallisella tasolla opetussuunnitelmissa, ja millä tavoin luokanopettajakoulutuksessa opetaan tieto- ja viestintäteknologista osaamista ja sen opettamista.

Luvussa 4 tutkin niitä valmiita ja tarjolla olevia oppimateriaaleja, joita löytyy eri sivustoilta internetistä. Tutkin luvussa myös sitä, mitä oppimateriaaleja on millekin vuosiluokalle tarjolla. Peilaan näitä löytämiäni oppimateriaaleja opetussuunnitelmaan ja pohdin, millä tavoin nämä oppimateriaalit sopivat opetussuunnitelman mukaiseen opetukseen.

Luvussa 5 summaan tämän kandidaatintutkielman tulokset ja teen johtopäätökset tuloksien perusteella. Pohdin myös sitä, mitä aiheita tutkielmani tarjoaa jatkotutkimukselle.

2 Ohjelmoinnin opettaminen: miksi?

Koodiaapisen mukaan ohjelmointi on tietokoneiden ohjaamista antamalla käskyjä niille (Koodiaapinen 2017). Ohjelmointi tapahtuu tietokoneella yleisesti käyttämällä jotain ohjelmointikieltä, jota tietokone suorittaa sen jälkeen, kun ohjelmakoodi on käännetty konekielille. Ohjelmointia voi opettaa myös ilman minkään ohjelmointikielen esittelemistä.

Vuonna 2014 opetusministeri Krista Kiuru linjasi, että uuteen opetussuunnitelmaan sisällytetään ohjelmoinnin opettamista. Hän perusteli sitä nuorten tieto- ja viestintäteknologisten taitojen parantamisella ja suomalaisten vahvan ICT-alan osaamisen turvaamisella (Verkkouutiset 2014). Euroopan maista muun muassa Viro ja Englanti ovat aloittaneet vuonna 2014 ohjelmoinnin opettamisen perusopetuksessa. Koska opetussuunnitelmaa päivitetään noin kymmenen vuoden välein, olisi seuraavaan opetussuunnitelmaan tulevat muutokset olleet hyvin paljon myöhässä verrattuna muihin Euroopan maihin (Liukas & Mykkänen 2014), (Yle 2013).

Ohjelmoinnin opettamista on perusteltu myös nykypäivän vaatimuksilla: yhteiskunnassamme on entistä enemmän tietotekniikkaa ja teknologiaa, joten lasten tulisi ymmärtää sitä. Samalla tavalla sotien jälkeen lasten perustaitoja ovat olleet viljely ja metsänhoito, sen ajan yleishyödyllisinä taitoina (Koodiaapinen 2017).

Ohjelmoinnin opettaminen lisää lapselle työkaluja ratkaista erilaisia ongelmanratkaisutehtäviä. Ohjelmointia opetellessaan lapsi opettelee myös loogista ajattelua ja erilaisia käytännön taitoja, joita tarvitsee arjessa. Ohjelmointitaidon oppeja voi siirtää sitten kiinnostuksen mukaan arkea helpottamaan tarvittaessa. Koodiaapinen lisää, että koodaaminen on luovaan ilmaisuun käytettävä työkalu ja koodaustaidon myötä pystyy hyödyntämään tietokoneita opiskelun tukena. Itse koodaaminen ei ole kuitenkaan päätavoite uudessa OPS:ssa, vaan algoritminen ajattelu (Koodiaapinen 2017).

3 Opetussuunnitelmat ja ohjelmointi

Tässä luvussa käsittelen miten ohjelmoinnin opetus näyttäytyy kolmessa eri opetussuunnitelmassa. Tarkastelussa ovat valtakunnallinen opetussuunnitelma (2016), Jyväskylän alueen koulujen opetussuunnitelma (2016) sekä Jyväskylän yliopiston opettajankoulutuslaitoksen luokanopettajakoulutuksen opetussuunnitelma (vuosille 2014–17).

3.1 Valtakunnallinen OPS

Uusi perusopetuksen opetussuunnitelman (POPS) perusteet julkaistiin vuonna 2014 ja se otettiin käyttöön alakouluissa 1.8.2016. Edellinen opetussuunnitelma hyväksyttiin vuonna 2004 ja otettiin käyttöön 2006. Valtakunnallinen opetussuunnitelma luo perustan paikalliselle opetukselle, ja tätä paikallista opetusta ohjaa paikallinen opetussuunnitelma.

Koska alakoulun opetuksessa ei ole tietotekniikan opetusta omana aineenaan, sisältyy se ja tieto- ja viestintäteknologisten (tvt) taitojen opetus moniin muihin aineisiin, esimerkiksi matematiikkaan, ympäristöoppiin ja käsityöhön.

3.1.1 Laaja-alainen osaaminen

Uudessa OPS:ssa mainitaan yhtenä laaja-alaisen osaamisen osa-alueena on tieto- ja viestintäteknologinen osaaminen (POPS 2014, s. 23). Perusopetuksessa tvt-osaamisen kehittäminen on jaettu neljään alueeseen, joista ensimmäinen on tvt:n käyttö- ja toimintaperiaatteiden ymmärtämiseen ohjaaminen. Vuosiluokilla 1-2 tämä tvt-osaaminen (L5) on ohjelmoinnin kannalta vain oppilaiden kokemusten jakamista ikäkaudelle sopivasta ohjelmoinnista (POPS 2014, s. 101). Opetussuunnitelmassa todetaan vuosiluokille 3-6 oppilaiden saavan kokemuksia ohjelmoinnin kokeilemisen yhteydessä siitä, millä tavoin ihmisen tekemät ratkaisut vaikuttavat teknologian toimintaan (POPS 2014, s. 157).

3.1.2 Ohjelmointi oppiaineissa

Opetussuunnitelmassa ohjelmointi mainitaan suoraan matematiikan osiossa. Vuosiluokilla 1-2 opetuksen tavoitteissa mainitaan (T12) vaiheittaisten toimintaohjeiden laatiminen ja ohjeen mukaan toimiminen. Sisältöalueissa (S1) mainitaan, että ohjelmoinnin alkeisiin tutustutaan algoritmista ajattelua kehittämällä (POPS 2014, s. 129).

Matematiikan vuosiluokkien 3-6 käsitteellisissä ja tiedonalakohtaisissa tavoitteissa todetaan, että opetuksen tavoitteena on "innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmina graafisessa ohjelmointiympäristössä"(POPS 2014, s. 235). Saadakse arvosanan kahdeksan kuudennen luokan päätteeksi oppilaan tulee osata ohjelmoida toimiva ohjelma, joka on tuotettu graafisessa ohjelmointiympäristössä (POPS 2014, s. 239).

Käsityön opetuksen sisältöalueissa mainitaan hieman yllättäen ohjelmointiin vahvasti kytköksissä olevat robotiikka ja automaatio (POPS 2014, s. 271).

3.2 Jyväskylän alakoulujen OPS

Tässä kappaleessa lähteenä on OPS2016 peda.net-sivusto, josta löytyy eri kuntien tai kaupunkien seutujen opetussuunnitelmat. Tutkin tässä kappaleessa Jyväskylän kaupungin opetussuunnitelmaa,¹ jota Jyväskylän kaupungin koulut noudattavat opetuksessaan.

Jyväskylän OPS:ssa laaja-alaisen osaamisen painotuksina on muun muassa valtakunnallisen OPS:n L5 Tieto- ja viestintäteknologisen osaamisen kehittäminen. Tämän laaja-alaisen osaamisen sisällä Jyväskylässä painotetaan opetussuunnitelmaratkaisuissa kasvun kansion käyttöönottoa, jolloin oppilas harjoittelee oman oppimisen dokumentointia. Myös mediasisältöjen turvallinen käyttö on toinen painotettavista opetussuunnitelmaratkaisuista.

¹<https://peda.net/opetussuunnitelma/ksops/jyvaskyla>

3.2.1 Vuosiluokat 1-2

Vuosiluokilla 1-2 matematiikan tavoitteiden sisältöalueisiin on mainittu (S1) ohjelmoinnin alkeisiin tutustuminen, jota tehdään laatimalla ja testaamalla vaiheittaisia toimintaohjeita. Suomen kielen ja kirjallisuuden tavoitteiden sisältöalueessa S3 mainitaan näppäintaitojen kehittäminen. Käsityön tavoitteiden sisältöalueessa on oma kohtansa dokumentoinnille (S5), jota hyödynnetään tieto- ja viestintätekniikan käytön tutustumiseen. Dokumentoinnin opettelua voisi ajatella alkuopetteluna valmiin koodin dokumentointiin ja kommentointiin. Liikunnassa on mainittu sisältöalueessa S2 sosiaalinen toimintakyky, jolloin opetukseen sisällytetään liikunnallisia yksinkertaisia sääntöleikkejä. Uskoisin, että nämä tukevat esimerkiksi ohjelmoinnissa olevien ehtolauseiden pohjustamista.

3.2.2 Vuosiluokat 3-6

Vuosiluokilla 3-6 suomen kielen ja kirjallisuuden tavoitteiden sisältöalueessa S3 mainitaan näppäintaitojen sujuvoittaminen. Matematiikassa opetuksen tavoitteissa mainitaan selvästi kohdassa T14 graafiset ohjelmointiympäristöt ja niiden avulla toimintaohjeiden laatimisen tietokoneohjelmoina. Myös matematiikan tavoitteiden sisältöalueessa S1 mainitaan ohjelmien suunnittelu ja toteutus graafisessa ohjelmointiympäristössä.

3.3 Jyväskylän yliopiston opettajankoulutuslaitoksen luokanopettajakoulutuksen OPS

Jyväskylän yliopiston luokanopettajakoulutuksessa oppiaineiden opetuksen keskeiset sisällöt käydään läpi POM-opinnoissa (Perusopetuksessa opettavien aineiden ja aihekokonaisuuksien monialaiset opinnot) (Jyväskylän Yliopisto 2014). Täten tv-taitojen opetus on integroitu POM-opintoihin. Erillistä opetusta tv-taitojen opettamiseen ei ole luokanopettajakoulutuksessa, joten valmistuvien luokanopettajien valmiudet opettaa ohjelmointia voivat olla haasteelliset riippuen opettajan omasta harrastuneisuudesta. Kuitenkin Jyväskylän yliopistossa tietotekniikka on vapaasti

opiskeltava sivuaine, jossa on mahdollista suorittaa kursseja liittyen muun muassa tietotekniikan opettajan työvälineisiin tai tietotekniikan rooliin opetuksessa. Luokanopettajaopiskelija on omillaan ohjelmoinnin opiskelussa, sillä ohjelmoinnin kurssit eivät ole välttämättömyys tai kannustetuin vaihtoehto sivuaineen kannalta. Ohjelmoinnin kurssit eivät ole myöskään välttämättömiä valmistumiseen.

Jyväskylän yliopiston luokanopettajakoulutuksen POM-opintojen matematiikan osi-
oissa puhutaan opetuksen tavoitteissa vain siitä, että opiskelija pystyy toteuttamaan matematiikan opetusta kansallisen opetussuunnitelman mukaisesti. Opiskelijan tulee myös osata hyödyntää "erilaisia ajattelutapoja – keskeisten matematiikan käsitteiden tarkastelussa" (Jyväskylän Yliopisto 2014). Ohjelmoinnin opettamista (tai muutakaan matematiikan osasisältöä) ei siis mainita erikseen, sillä se sisältyy kansalliseen opetussuunnitelmaan.

Hähkiöniemi, Tuohilampi ja Kaartinen, jotka vastaavat JY:n OKL:n luokanopettajakoulutuksen POM-opintojen matematiikan kursseista, kirjoittavat, että pääpaino matematiikan ydinosan ja soveltavalla kurssilla on pedagogisen näkemyksen omaksumisella (Hähkiöniemi, Kaartinen & Tuohilampi 2017). Matematiikan kursseilla käsitellään matematiikan pedagogiikkaa, jolloin kurssin työtunnit eivät anna periksi perehtyä mihinkään yksittäiseen matematiikan aiheeseen. Kursseilla on heidän mukaansa ohjelmoinnin sisältöjen esittely tehty muutamissa demonstraatioissa. Ydinosan kurssilla on pohdittu yhdessä sitä, miten alakoulussa voisi käsitellä ohjelmointia. Demonstraatioissa on voitu myös lyhyesti käsitellä tarkkoja ohjeita (algoritmita ajattelua) ja blokkiohjelmointikieltä. Kurssilla opiskelijat ovat kuulemma pitäneet myös opetustuokioita muille opiskelijoille. Soveltavan osan kurssilla taas on voitu perehtyä teknologiaan syvemmin yhteistyössä IT-tiedekunnan kanssa. Teknologia on ollut myös tällä kurssilla yhden demonstraation teemana.

4 Käytössä olevat oppimateriaalit

Kuten johdannossa todettiin, tietotekniikan opettamiseen on ollut viime vuosikymmenenä hyvin vähän valmiita oppimateriaaleja tarjolla (Ekonoja 2014, s. 16), (Ilomäki 2012, s. 7), (Yle 2013). Kuusisto (1989, tiivistelmä) määrittelee oppimateriaalin oppi- ja työkirjoiksi, jotka on tuotettu ensisijaisesti valtakunnallisesti. Määritelmän mukaan oheismateriaalit ovat myös oppimateriaaleja. Koska tv:n opettaminen on enemmän painottunut tietokoneiden ja muiden sähköisten välineiden pariin, ovat sähköiset oppimateriaalit myös lisääntyneet. Sähköinen oppimateriaali on Ekonon mukaan oppimateriaalia, joka on käytettävissä tietokoneella tai muulla tietoteknisellä laitteella (Ekonoja 2014, s. 58-59).

Alakouluun suunnattuun ohjelmoinnin opettamiseen löytyy internetistä valmiita, erilaisia sähköisiä oppimateriaaleja tarjoavia sivustoja, joita voidaan käyttää opetuksessa. Seuraavaksi esitellyt esimerkit ovat vain muutamia tähän kandidaatintutkimusta varten nostettua sivustoja. Oppimateriaaleja ja oppimisympäristöjä on olemassa muitakin, kuten Alice¹ ja Codecombat².

4.1 Code.org

Code.org³ on vuonna 2013 perustettu voittoa tavoittelematon sivusto, joka pyrkii laajentamaan tietotekniikan harrastamista ja tarjoaa mahdollisuuden viedä tietotekniikan opetusta enemmän kouluihin. Code.orgin ovat perustaneet iranilaiset Hadi ja Ali Partovi ja sivuston yhteistyökumppaneita ovat mm. Amazon, Apple, Facebook, Google ja Microsoft. Myös suomalainen Rovio on sivuston yhteistyökumppani, mikä näkyikin Angry Birds -hahmojen näkymisenä tehtävissä. Code.org järjestää vuosittain Code of Hour -tapahtuman (suom. Koodaustunti) (Code.org 2017).

Sivustolta löytyy kahdentyypisiä opetusmateriaaleja: 20 tunnin kurseja ja koo-

¹<http://www.alice.org/index.php>

²<https://codecombat.com/>

³<https://code.org/>

daustunteja. 20 tunnin kurssit ovat jaoteltu ikäryhmittäin neljään eri kurssiin, joita on luonteva suorittaa suoritettuaan edellisen kurssin materiaalit. Lisäksi sivustolta löytyy nopeutettu versio kursseista 2-4. 20 tunnin kurssit ovat hyvin perusteellisia: seuraavaan tehtävään pääseen jatkamaan, kun edellinen tehtävä on ratkaistu onnistuneesti. Code.orgin materiaalit suuntautuvat 4-15-vuotiaille, joten sivustolta löytyy materiaalia myös yläkoulun oppilaille.

Koodaustunnit on tarkoitettu kaikille ikäryhmille ja ne ovat jaettu erilaisiin teemoihin. Valittavissa on muun muassa Frozen, Minecraft ja Tähtien Sota -aiheiset koodaustunnit ja ne sisältävät myös 20 erilaista oppimistehtävää ohjelmoinnin parissa. Koodaustunneissa valittu tema on tiukemmin mukana oppimistehtävissä verrattuna normaaleihin 20 tunnin kursseihin.

20 tunnin kurssien lisäksi sivusto tarjoaa myös opetusmateriaaleja, joita voi hyödyntää, mikäli luokkahuoneesta ei löydy tietokonetta ("unplugged activity"). Nämä tunnit ovat jaettu muun muassa algoritmeihin, binäärilukuihin, ehtolauseisiin ja debuggaamiseen. Opettajan hyötynä ovat valmiit videot, joissa on esitetty opetusketken rakenne ja tehtävät. Tällöin opettaja selviää vähemmällä suunnittelutyöllä, kun valmiin materiaalin voi siirtää suoraan omaan opetukseen. Videot ovat englannin kielisiä. Opettaja voi käyttää videota opetuksen suunnittelussa apuna, mutta videota on mahdollista käyttää myös johdatuksena esimerkiksi seuraavan tunnin asiaan kotiläksyn muodossa, jolloin oppilaille tulee kokemusta myös englannin kielen kuuntelemisesta.

Sivuston etuna on suuri määrä erilaisia tehtäviä. Ohjelmoinnin oppiminen tapahtuu oppimistehtävissä lähinnä raahaamalla toimintapalikoita työtilaan peräkkäin sopivassa järjestyksessä, jotta ennalta annettu tehtävä toteutuisi oikein. Oppilaat voivat siirtyä oman tason mukaan aina eteenpäin, jolloin eriyttäminen toimii niin ylöskuin alaspäinkin helposti. Tehtävän suoritettuaan oppilas voi myös katsoa miltä ohjelmakoodi ratkaistussa tehtävässä näyttää.

Sivusto on alunperin englanninkielinen, mutta suuri osa siitä on käännetty myös suomeksi. Osalla sivustosta on käännöstyö vielä kesken, jolloin kielitaidottomampi



Kuvio 1. Code.orgin tehtävän suoritettuaan voi katsoa valmista ohjelmakoodia.

oppilas voi hämääntyä etsiessään sopivaa oppimateriaalia.

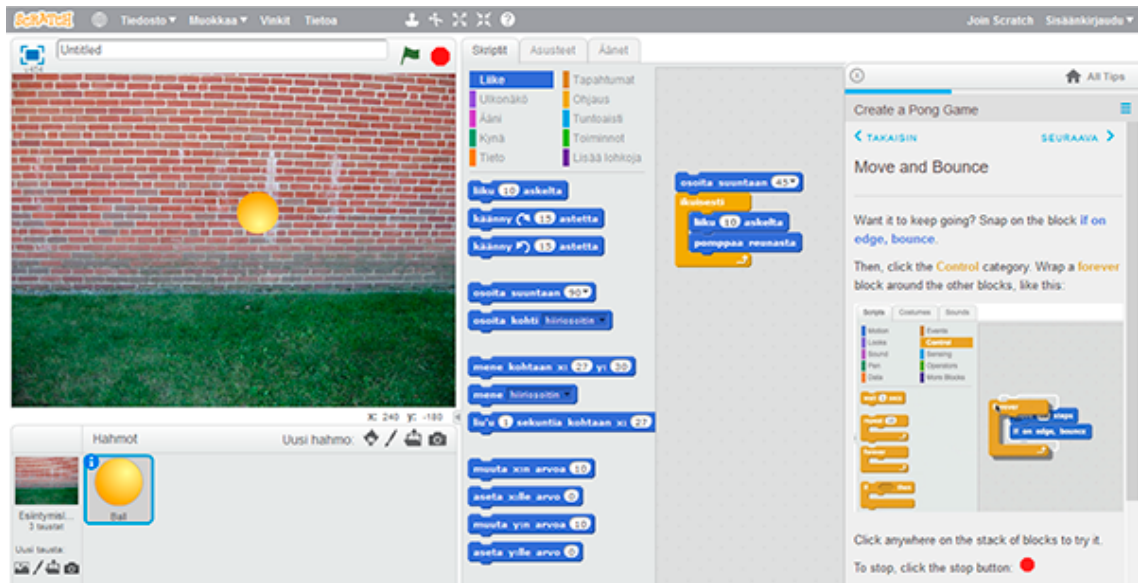
4.2 Scratch ja Scratch Jr

Scratch ja Scratch Jr ovat MIT (Massachusetts Institute of Technology) Media Lab -laitoksen suunnittelemat ja kehittämät ohjelmointiympäristöt. Ne ovat suunniteltu 5-16 vuotiaille lapsille.

Scratch Jr⁴ on tabletilaitteissa toimiva sovellus, joka sisältää muutamia eri harjoituksia ohjelmoinnin opetteluun. Scratch Jr on saanut vaikutteita Scratch -ohjelmointikielestä, mutta käyttöliittymä ja ohjelmointikieli ovat suunniteltu uudelleen. Koodiaapinen

⁴<https://www.scratchjr.org/>

suosittelee⁵ jo alaluokille, jopa esiopetukseen, Scratch Jr:ia.



Kuvio 2. Scratchin työskentelytila. Ohjeet ohjelman tekemiseen löytyvät oikeasta reunasta, koodialue on sen vieressä ja komentoalue löytyy keskeltä. Vasemmalla on näyttämö.

Koodi2016-sivusto taas mainitsee⁶ vanhemmille oppilaille Scratchin 3.-6. luokkien opetusmateriaaleihin sopivana. Scratch toimii verkkoselaimessa, eikä sitä pelataksaan tarvitse ladata erillistä sovellusta. Scratch:ssä ja Scratch Jr:ssa lapsi raahaa ohjelmointikäskypalikoita komentoalueelta koodialueelle ja saa näin näyttämöllä olevat objektit tekemään koodissa määriteltyjä asioita.

Scratch Jr:n käyttömahdollisuus tablet-laitteella parantaa oppimateriaalin tavoitettavuutta verrattuna pöytätietokoneeseen tai kannettavaan. Sovelluksen alustaratkaisu parantaa oppilaiden innostuneisuutta oppimateriaalia ja opiskeltavaa aihetta kohtaa pelimäisyytensä vuoksi. Opiskelu ei rajoitu tällöin pelkästään erillisiin tietokoneiluokkiin koulussa tai sisätiloihin kotona, vaan oppilaat voivat tehdä tehtäviä esimerkiksi omissa kotiluokissaan, vapaa-ajallaan ulkona tai koulumatkalla bussissa tai autossa.

⁵<http://koodiaapinen.fi/kirjasto/>

⁶<http://koodi2016.fi/opetus.html#section-4>

Kuten Code.org -sivustolla, Scratch on osittain käännetty suomeksi, mutta esimerkiksi tehtävien ohjeistukset ovat englanniksi. Tämä voi hidastaa oppilaan työskentelyä. Valmiiden tehtävien määrä on vähäinen, jolloin Scratchia käytettäessä oppimateriaalina ohjelma vaatii opettajalta enemmän suunnittelua.

4.3 Tavoitteiden toteutuminen oppimateriaaleissa

Opetussuunnitelmassa alaluokille todettu kokemusten jakaminen oppilaiden kesken (kts. 3.1.1) tukee yhteisöllistä ja tutkivaa oppimista. Lakkala (2012, s. 93) toteaa, että tutkivan oppimisen tukena voidaan käyttää esimerkiksi simulaatiota, joka mahdollistaa ilmiön toimintaperiaatteiden kokeilua. Edellä mainitut graafiset oppimisympäristöt simuloivatkin ohjelmoinnin erilaisia elementtejä.

Code.orgin tehtävien monipuolisuus ja runsaus mahdollistaa sen, että oppilas oppii perusteet ohjelmoinnista ja algoritmisesta ajattelusta hyvin tekemällä sivuston tehtäviä. Code.orgin tehtävissä oppilaalle tulee tutuiksi komentosarjat, muuttujat, ehtolauseet, silmukat ja funktiot. Scratchia käytettäessä oppimateriaalina opettajan vastuu on hieman suurempi, sillä valmiiden tehtävien lukumäärä on pienempi. Tällöin opettaja voi joutua turvautumaan lisäoppimateriaaleihin.

Opetussuunnitelma ei anna muita oppimistavoitteita kuin graafisessa ohjelmointiympäristössä tuotetun ohjelman kuudennen luokan päättävälle oppilaalle. Tällöin arvosanaan kahdeksan oikeuttavan oppilaan ohjelmoima ohjelma voi olla opettajasta riippuen sisällöltään hyvin paljon erilainen eri kouluissa ja eri luokkien sisällä. Tämän vuoksi oppilaiden tasoerot voivat olla todella suuret siirryttäessä yläkouluun ja oikeiden koodirivien ääreen.

4.4 Yhteenveto oppimateriaaleista

Code.orgin ja Scratchin materiaalit ovat hyvin visuaaliseen oppimiseen ja kokeiluun pohjautuvia oppimateriaaleja. Graafisissa ohjelmointiympäristöissä oppilas tekee itse komentosarjan koodipätkistä ja suorittaa sen. Tämän jälkeen ohjelma antaa

palautteen oppilaalle, joten oppilas oppii tällöin ohjelmoinnin eri aiheita itse kokeilemalla ja testaamalla. Kokeileminen onkin yksi eurooppalaisen CELEBRATE-hankkeen määrittelemä oppimisaihio, ja kokeilemisen tavoitteena opetuksessa on muun muassa virittää oppijan ajattelua. Graafiset ohjelmointiympäristöt voidaan lukea käsitteellisiin sovellusaihioihin, sillä oppilas ratkoo erilaisia ongelmia ja sovellukset tarjoavat oppilaille mahdollisuuden aktiiviseen tutkimiseen ja työskentelyyn (Jaakkola, Nirhamo, Nurmi ja Lehtinen 2012, s. 17.)

Lapsen ohjelmoimaa käskysarjaa suorittavat lapsille helposti lähestyttävät hahmot, kuten mehiläiset, Angry Birds:n linnut ja muut piirretyt hahmot. Tapola & Veermans (2012, s. 75) toteavat, että oppilaan motivaation herättävä opiskelumateriaali sisältää elementtejä, jotka ovat usein muun muassa humoristisia, yllätyksellisiä ja intensiivisiä. Code.orgin ja Scratchin materiaalit ovat myös hyvin värikkäitä ja pelimäisiä, ja nämä elementit myös Tapola ja Veermans mainitsevat artikkelissaan. Jaakkola (2012, s. 87) taas kehoittaa käyttämään useita erilaisia esitystapoja opetettavan ilmiön esittämisessä. Code.orgin tietokoneettomat tehtävät tukevatkin opetettavien asioiden esittämistä erilaisilla tavoilla, jolloin oppilaat voivat oppia ohjelmoinnin asiayhteyksiä uusilla tavoilla, ja opetus monipuolistuu, kun opetusmateriaalia voidaan käyttää eri tilanteissa ja tiloissa.

On todettu, että nuoret opettajat käyttävät innokkaasti sähköisiä oppimateriaaleja opetuksessa tekniseltä kannalta osaavasti, mutta pedagoginen näkökulma ei ole todennäköisesti sisäistetty opetuksen kannalta (Ilomäki 2012, s. 7). Tällöin vaarana on se, että sähköisten oppimateriaalien tuominen opetukseen ei välttämättä monipuolista lapsen tapoja oppia opetettavia asioita. Opettajan tulisikin pohtia, että tuoko sähköinen oppimateriaali opetukseen jotain lisää verrattuna perinteiseen opettamiseen. On kuitenkin todettu, että sähköiset oppimateriaalit ovat oppilaiden mielestä parempia kuin painetut oppimateriaalit ja ne palvelevat paremmin eri tavoilla opiskelevia oppilaita (Ekonoja 2014, s. 183).

5 Yhteenveto

Tutkielmassa on tähän mennessä tullut ilmi, että oppimateriaalien määrä ohjelmoinnin opettamiseen on vähäinen, mutta uuden opetussuunnitelman minimivaatimuksiin nähden riittävä: Code.org, Scratch ja Scratch Jr tarjoavat materiaalia enemmän kuin oppitunteja olisi käytettävissä ohjelmoinnin parissa. Eriyttäminen opetuksessa on opettajan kannalta helppoa ainakin Code.org:ia käytettäessä oppimateriaalina, sillä tehtävät vaikeutuvat sitä mukaan, kun oppilas tekee tehtäviä.

Ohjelmoinnin opettaminen tulee olemaan hyvin suurella todennäköisyydellä integroituna matematiikkaan. Tulee kuitenkin olemaan opettaja- ja koulukohtaistakin, miten paljon ohjelmointia opetetaan matematiikan tai käsityön tunnilla, sillä opetussuunnitelma ei rajaa tarkasti opetuksen sisältöä. Ohjelmoinnin opettamisen haasteina ovat opetussuunnitelman oppimistavoitteet, jotka antavat opettajille paljon tulkinnan varaa. Toisaalta löyhästi määritellyt oppimistavoitteet antavat opettajalle vapauksia soveltaa ohjelmoinnin opetuksessa, mutta oppilaiden taidot voivat olla kuudennen luokan päättyessä hyvin erilaiset opettajasta ja tämän kiinnostuksesta riippuen.

Yleinen puute ohjelmoinnin oppimateriaaleissa on keskeneräinen käännöstyö. Nykypäivän alakoululaiselle englannin kieli ei kuitenkaan ole välttämättä suuri haaste, sillä oppilaat altistuvat helposti englannin kielelle juuri tieto- ja viestintäteknologian viihteellisen käytön kautta verrattuna aiempiin sukupolviin. Opettajan tulee kuitenkin arvioida oppimateriaalin soveltuvuutta luokalle tapauskohtaisesti.

Opettajien valmiudet opettaa ohjelmointia ovat kyseenalaiset. Vaikka valmistuvat luokanopettajat ovat enemmän innostuneempia sähköisten oppimateriaalien käytöstä opetuksessa ja he käyttävät teknologiaa opetuksessa sujuvasti (Ilomäki 2012), on osa luokanopettajista vielä kuitenkin sitä sukupolvea, joilla ei ole ollut kontaktipintaa teknologiaan ja täten kynnys ottaa teknologiaa luontevasti opetuksen osaksi voi olla todella korkea.

Ohjelmoinnin opettaminen on ajankohtainen taito, sillä tulevaisuudessa ohjelmoin-

nin oppimisen ja tv-taitojen merkitys ei oppilailla tule vähenemään. Yliopistojen opettajankoulutuslaitoksilla olisi tämän takia hyvä kannustaa luokanopettajaopiskelijoita opiskelemaan tietotekniikkaa ja muita opettajankoulutuslaitoksen ulkopuolisia sivuaineita osana opintojaan. Tietotekniikan opinnot ovat opettajille yleishyödyllisiä: ne lisäävät yleistä ammattitaitoa ja tällöin myös opettajan valmiudet kehittää opetusta paranevat yleisesti.

Kandidaatintutkielmani aihealueen tutkimusta voisi tulevaisuudessa jatkaa esimerkiksi kartoittamalla luokanopettajien tapoja opettaa ohjelmointia perusopetuksessa. Olisi kiinnostavaa tietää, mitä mieltä luokanopettajat ovat uudesta sisällöstä perusopetuksen osana ja miten he ovat vastaanottaneet uuden oppisisällön. Laatuerot ohjelmoinnin opetuksessa ovat olleet huolena luokanopettajien keskuudessa, joten jatkotutkimus voisi myös perehtyä kuudesluokkalaisten osaamistasoihin alakoulun päättyessä. Myös oppimateriaalien käyttö opetuksessa ja niiden runsaus tai monipuolisuus voisi tulevaisuudessa tarjota mielenkiintoisen tutkimuskysymyksen, sillä vielä ainakaan laadukasta oppimateriaalia ei ole runsaasti tarjolla. Myös oppilaiden kokemuksia ohjelmoinnista ja sen opiskelusta olisi mielenkiintoista selvittää.

Lähteet

Code.org 2017. *Code.org: About Us*. Saatavilla WWW-muodossa <URL: <https://code.org/about> Viitattu 22.3.2017.

Ekonoja, A. 2014. *Oppimateriaalien kehittäminen, hyödyntäminen ja rooli tieto- ja viestintäteknikan opetuksessa*. Jyväskylän yliopisto. Jyväskylä Studies in Computing 193.

Hahkiöniemi, M., Kaartinen S., & Tuohilampi L. 2017. *Henkilökohtainen tiedonanto sähköpostitse*. 20.3.2017.

Ilomäki, L. *Erilaiset e-oppimateriaalit*. Teoksessa Ilomäki, L. (toim.) *Laatua oppimateriaaleihin: E-oppimateriaalit opetuksessa ja oppimisessa*. Opetushallitus. Oppaat ja käsikirjat 2012:5 7-11.

IT-kouluttajat ry 2017. *Koodiaapinen*. Saatavilla WWW-muodossa <URL: <http://koodiaapinen.fi/> Viitattu 7.2.2017.

Kuusisto, J. 1989. *Oppimateriaalit peruskoulun ala- ja yläasteella 1988*. Jyväskylän yliopisto. Kasvatustieteiden tutkimuslaitoksen julkaisusarja, sarja A, osa 26.

Jaakkola, T. *Esitä ilmiö usealla tavalla*. Teoksessa Ilomäki, L. (toim.) *Laatua oppimateriaaleihin: E-oppimateriaalit opetuksessa ja oppimisessa*. Opetushallitus. Oppaat ja käsikirjat 2012:5 7-11.

Jaakkola, T., Nirhamo, L., Nurmi, S., ja Lehtinen, E. *Erilaiset oppimisaihiot osana joustavaa kokonaisuutta*. Teoksessa Ilomäki, L. (toim.) *Laatua oppimateriaaleihin: E-oppimateriaalit opetuksessa ja oppimisessa*. Opetushallitus. Oppaat ja käsikirjat 2012:5 7-11.

Jyväskylän Yliopisto 2014. *Opettajankoulutuslaitoksen opetussuunnitelma 2014-2017, Perusopetuksessa opettavien aineiden ja aihekokonaisuuksien monialaiset (POM) opinnot (OKLPOM 60 op)*. Saatavilla WWW-muodossa <URL: <https://www.jyu.fi/edu/opiskelu/oppaat/edu-opetussuunnitelmat/EDU-OPS-2014-2017/OKL-OPS-2014-17/luokanopettajakoulutuksen-opetussuunnitelma-2014-17/kasvatustieteen-kandidaatin-tutkinnon-opintojaksot/perusopetuksessa-opettavien-aineiden-ja-aihekokonaisuuksien-monia> Viitattu 9.2.2017.

- Lakkala, M. *Tutkiva oppiminen*. Teoksessa Ilomäki, L. (toim.) *Laatua oppimateriaaleihin: E-oppimateriaalit opetuksessa ja oppimisessä*. Opetushallitus. Oppaat ja käsikirjat 2012:5 7-11.
- Liukas, L. & Mykkänen, J. 2014. *Koodi2016*. Saatavilla WWW-muodossa <URL: <http://koodi2016.fi/>>. Viitattu 7.2.2017.
- Opetushallitus 2014. *Perusopetuksen opetussuunnitelman perusteet 2014*. Saatavilla WWW-muodossa <URL: http://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf> Viitattu 7.2.2017.
- Potila, J. 2015. *Ohjelmointisatukirjat*. Jyväskylän yliopisto. Tietotekniikan laitos. Kandidaatintutkielma.
- Tapola A. & Veermans M. *Herätä ja tue kiinnostusta ja motivaatiota*. Teoksessa Ilomäki, L. (toim.) *Laatua oppimateriaaleihin: E-oppimateriaalit opetuksessa ja oppimisessä*. Opetushallitus. Oppaat ja käsikirjat 2012:5 74-81.
- Yle 2013. *Suomen koulujen it-opetus Jordanian tasolla – uutta mallia haetaan Virosta*. Saatavilla WWW-muodossa <URL: <http://yle.fi/uutiset/3-6885099>> Viitattu 2.3.2017
- Verkkouutiset 2014. *Ministeri Kiuru: Ohjelmointi peruskoulun opetussuunnitelman perusteisiin*. Saatavilla WWW-muodossa <URL: http://www.verkkouutiset.fi/kotimaa/Kiuru_ohjelmointi-14614> Viitattu 2.3.2017.
- Wikipedia 2017. *Ohjelmointi*. Saatavilla WWW-muodossa <URL: <https://fi.wikipedia.org/wiki/Ohjelmointi>> Viitattu 22.3.2017.