

Universal declaration of cultural diversity declaration (2001). [www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CLT/diversity/PDF/declaration\\_cultural\\_diversity.fi.pdf](http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CLT/diversity/PDF/declaration_cultural_diversity.fi.pdf) (Luettu 2.1.2019)

Vitikka, E. (2015). Curriculum reform and development in Finland. *Diaesitys* 12.11.2015 Opetushallitus. [https://www.curriculum.ut.ee/sites/default/files/ht/12.11.15\\_curriculum\\_reform\\_in\\_finland\\_vitikka.pdf](https://www.curriculum.ut.ee/sites/default/files/ht/12.11.15_curriculum_reform_in_finland_vitikka.pdf) (Luettu 28.12.2018)

Vorkabić, S. T. (2016). Promotion of internationalization of teacher education: A case study from Croatia. Teoksessa D. M. Velliaris & D. Coleman-George (toim.) *Handbook of research on study abroad programs and outbound mobility*. Hershey, PA: IGI Global, 651–671.

Weimer, L., Hoffman, D. & Silvonen, A. (2019). *Internationalisation at home in Finnish higher education institutions and research institutes*. Opetus- ja kulttuuriministeriön julkaisuja 2019:21. [http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161606/OKM\\_2019\\_21\\_Internationalisation\\_at\\_Home.pdf](http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161606/OKM_2019_21_Internationalisation_at_Home.pdf)

## 6 MOTIVOIVAA JA TAVOITTEELLISTA OPPIMISTA OHJELMOINTIYMPÄRISTÖJÄ HYÖDYNTÄEN

**Janne Fagerlund ja Susanne Roos**

---

### TIIVISTELMÄ

Tämä artikkeli johdattelee ohjelmoinnin oppimisen mahdollisuuksiin perusopetuksen opetussuunnitelmaan kuuluvien eri oppiainekokonaisuuksien ja laaja-alaisen osaamisalueiden kautta. Artikkelin tarkoituksena on toimia myös oppaana opettajalle ohjelmointia sisältävien tehtävien toteuttamiseen omassa opetuksessaan. Tarkastelemme ohjelmoinnin yhdistämistä erilaisten taitojen ja oppimistavoitteiden harjoitteluun erityisesti alakoulun ensimmäisillä luokka-asteilla. Ohjelmointia voidaan toteuttaa ja oppia alkuopetusluokilla monin eri tavoin, sillä se on opittavan sisällön lisäksi ennen kaikkea työtapaa. Artikkelissa esittelemme useita erilaisia ohjelmointitehtäviä, jotka on suunniteltu, toteutettu ja testattu vuosina 2015–2020 alakoulun 1.–2.-luokilla. Esimerkkien avulla havainnollistamme, miten on mahdollista oppia samalla sekä ohjelmoinnin ja ohjelmoinnillisen ajattelun alkeita että myös eri oppiaineisiin ja laaja-alaiseen osaamiseen kuuluvia taitoja. Lisäksi avaamme pedagogisesta näkökulmasta tarkemmin 3. luokalla toteutettua projektia, jossa oppilaat suunnittelevat ja ohjelmoivat omia pelejään Scratch Junior-ohjelmointiympäristössä. Projektista voi havaita, miten alkuopetuksen aikana tehty huolellinen ohjelmoinnillinen pohjatyö, ohjelmoinnin perusteiden oppiminen, voi tarjota mahdollisuuden myös oppilaslähtöiseen ohjelmoinnin hyödyntämiseen.

Pyrkimyksenämme tässä artikkelissa on erityisesti hälventää ohjelmoinnin ympärillä yhä leijuvaa arvoituksellisuutta ja epävarmuutta ja tuoda aihepiiriä tutummaksi ja helpommin lähestyttäväksi sekä ennen kaikkea luonnolliseksi osaksi kouluarkea. Kaikki artikkelissa esitellyt tehtäväesimerkit on suunniteltu ja testattu alkuopetusluokilla ja ne on tarkoitettu vapaasti sovellettaviksi omaan opetukseen.

## Ohjelmoinnin tulo koulumaailmaan

Ohjelmointi on peruskoulussa varsin uusi kaikille yhteinen oppisisältö ja työskentelytapa, joka jatkaa opettajien haastamista eri tavoin. Millaisilla välineillä ja materiaaleilla ohjelmointia voidaan toteuttaa lasten kanssa? Miten sitä integroidaan osaksi erilaisten taitojen opiskelua? Riittääkö oma ymmärrys ja osaaminen opetuksen suunnitteluun ja toteutukseen, oppimisen ohjaamiseen ja arviointiin?

Tieto- ja viestintäteknologinen osaaminen eri osa-alueineen on taitokokonaisuuksia, jota jokaisen 2000-luvun kansalaisen tulisi keskeisesti osata (Binkley ym. 2012). Useat maat, Suomi mukaan lukien, ovat sisällyttäneet ohjelmoinnin tai siihen liittyvän aihepiirin, kuten tietojenkäsittelyn, perusopetuksen ja toisen asteen opetussuunnitelmaan joko pakollisena tai valinnaisena kokonaisuutena vastatakseen tulevaisuuden yhteiskunnan osaamistarpeisiin erityisesti teknologiaosaamisen näkökulmasta (Heintz ym. 2016). Ohjelmoinnin opetuksen ajatuksena peruskoulussa on muun muassa ongelmanratkaisutaitojen sekä loogisen ajattelun kehittäminen, mutta nähdään sillä olevan merkitystä myös hahmottamisen ja sosiaalisten taitojen kehittymisessä (Opetushallitus 2014; eNorssi 2018). Ohjelmoinnin opetusta koululaisille on yhtäältä perusteltu lisääntyneellä ammattiohjelmoijien tarpeella mutta myös ennen kaikkea sillä, että ohjelmointia käytetään eri työaloilla yhä runsaammin. Lisäksi ohjelmoinnin kautta voidaan oppia tärkeitä digitaalisen kansalaisuuden taitoja. (Kafai ym. 2019; Grover 2018.) Ohjelmoinnin odotetaan myös kehittävän yleismaailmallisia loogisen päättelyn ja ongelmanratkaisun taitoja, joita voidaan soveltaa osana opiskelua ja muuta arjen tekemistä muissakin kuin vain ohjelmoinnillisissa tilanteissa. Ohjelmoinnin kautta opittavaan laajempaan taitokokonaisuuteen onkin viitattu usein nimityksellä ”ohjelmoinnillinen ajattelu” (ks. esim. Wing 2006), jota voidaan kehittää käytännössä jopa pienten lasten kanssa eri tavoin, muun muassa ohjelmoimalla itse omaa digitaalista laitetta. Kuten monissa muissakin opittavissa taidoissa, peruskouluikäisten lasten kanssa ohjelmointi aloitetaan alkeita opetellen.

## Mitä ohjelmointi on?

Ohjelmointi, kansankielisesti ”koodaaminen”, on ihmisen toteuttamaa käsityön tapaista käytännönläheistä toimintaa, jonka päämääränä on suunnitella ja rakentaa ohjelmia, joita tietokoneet voivat suorittaa automaattisesti. Ohjelmointi perustuu ennen kaikkea tietojenkäsittelyn toimialaan, jossa on jo vuosituhansien ajan ennen yhdenkään sähköisen laitteen keksimistä ratkaistu erilaisia käytännön ongelmia ohjelmoinnillisin keinoin, kuten helmi- tauluja ja jakokulmia hyödyntämällä. Niin kutsutun algoritmin<sup>1</sup> avulla mikä

<sup>1</sup> Esimerkiksi jakokulman algoritmi voidaan kuvata ”pseudokoodilla” eli keksityllä ihmiskielellä vaikkapa näin: (1) Jaa (2) Kerro (3) Vähennä (4) Luku alas pudota (5) Alusta taas aloita.

tahansa tietojenkäsittelyyn kykenevä toimija – oli se sitten ihminen tai digitaalinen laite – pystyisi lähtökohtaisesti ratkaisemaan käytännön ongelman, jota varten kyseinen algoritmi on suunniteltu. (Tedre & Denning 2016.) Verrattain moniosaisempien algoritmien avulla esimerkiksi nykypäivän teollisuusrobotit noudattavat täsmällisen mikroskooppisia liikeratoja tietystä järjestyksessä, kun taas tietokonepelit tuottavat laitteen ruudulle näyttäviä virtuaalimaailmoja sekä miellyttäviä pelimekaaniikkoja ja -tuntumaa.

Ohjelmointia tehdään erilaisilla ohjelmointikielillä, joilla jollekin digitaaliseen laitteeseen kerrotaan sen ymmärtämässä muodossa haluttu algoritmi eli yksityiskohtainen kuvaus siitä, mitä kyseisen laitteen tulee tehdä. Ohjelmoinnissa tuleekin tyypillisesti miettiä, millaisista osatoimenpiteistä jonkin mahdollisesti suuren ja monimutkaisen ongelman ratkaiseminen koostuu ja millaisilla tietokoneen ymmärtämällä komennoilla (esim. graafisilla eli kuvakepohjaisilla tai tekstipohjaisilla ohjelmointikielillä) ongelman ratkaisu saadaan kuvattua ja syötettyä tietokoneeseen toteutettavaksi käytännössä. Erilaisia ohjelmointikieliä ja ohjelmointikielten taustalla olevia ohjelmien rakennustapoja on runsaasti, mutta oletus on, että ohjelmoinnin kautta opittava ohjelmoinnillinen ajattelu antaa valmiuksia soveltaa keskeisiä ohjelmoinnillisen ongelmanratkaisun ydinperiaatteita ja -tekniikoita riippumatta käytössä olevasta ohjelmointiympäristöstä.

Lasten ja nuorten kanssa ohjelmoinnin alkeita harjoitellaan usein leikin ja pelinomaisissa ympäristöissä, kuten alkeisroboteilla, ja omien digitaalisten animaatioiden, tarinoiden tai pelien tekemisen kautta (esitellään tarkemmin myöhemmässä luvussa). Myös erilaiset laitteettomat leikit ovat olleet suosittuja ohjelmoinnin alkeiden oppimisen väyliä. (Innokas-verkosto 2019; Lee ym. 2011.) Ohjelmointiin oleellisesti kuuluvaa yksiselitteisten käskyjen antamista oikeassa järjestyksessä on itse asiassa toteutettu jo iät ja ajat koululuokissa päivittäin erilaisten toimintaohjeiden kautta (esim. ”pese kädet”, ”mene ruokalaan”). Ohjelmoinnin oppimisen taustalle usein sijoitettuun konstruktionistiseen oppimiskäsitykseen kuuluukin teesi, jonka mukaan vaikkapa robotin tai yhtäältä luokkakaverinkin liikuttamiseen tarkoitettujen käskysarjojen laatimisen taitoa vahvistaakin esimerkiksi omien asentojen ja liikkeiden sekä niissä tapahtuvien muutosten havainnointi (Papert 1980). Käytännössä lasten omaa ohjelmointikäsitystä onkin mielekästä herättää kouluarjessa esimerkiksi pohtimalla, toiminko itse tarkasti saamieni ohjeiden mukaan, menivätkö oikea ja vasen oikein päin ja millaisilla käskyillä voisin saada aikaan luokkakaverissa tavoittelemani kaltaista toimintaa. Leikin- ja pelinomaisista ohjelmoinnin oppimisen ympäristöistä huolimatta ohjel-

montia ei tule pitää pelkkänä ”puuhasteluna”, jossa tehdään mitä huvittaa saaden aikaan satunnaisen hupaisaa liikettä liikkuvassa laitteessa tai sekalaisia toimintoja tabletin tai tietokoneen ruudulla. Pedagogisesti ohjelmointi on ennen kaikkea suunnitelmallista ongelmanratkaisua: siinä on tavoite, usein jokin ratkaistava ja jokseenkin monimutkainen ongelma, johon tulee suunnitella ja toteuttaa sopivat ohjelmoinnilliset osaratkaisut, jotka saavat kokonaisuutena aikaan haluttua toimintaa ongelman kokonaisratkaisussa. Ohjelmoinnin oppimisen eräs keskeinen tarkoitus onkin oppia ymmärtämään, mihin kaikkeen ohjelmointia käytetään oikeassa maailmassa ja miten oikean elämän ongelmia voidaan ratkaista ohjelmoimalla (Denning & Tedre 2019). Jo alkuopetuksessa tällaisille käsityksille voidaan luoda pohjaa muun muassa ohjelmoimalla leikkillisessä ympäristössä ikäkauden ja kehitystason mukaisella tavalla hyötykäyttörobottien liikkeitä jotakin päämäärää varten (esim. kulkeminen paikasta toiseen) tai animointi tiettyyn käyttötarkoitukseen (esim. käytännön toiminnan ohjeistus).

## Ohjelmointi opetus suunnitelmassa vuosiluokilla 1–6

Valtakunnallisissa peruskoulun opetus suunnitelman perusteissa (Opetushallitus 2014) ohjelmointi sisältyy laaja-alaiseen osaamisen alueeseen L5, *Tieto- ja viestintäteknologinen osaaminen*, kaikilla perusopetuksen luokka-asteilla. Lisäksi siitä mainitaan erikseen jokaisen vuosiluokkahaitarin matematiikan tavoitteissa sekä vuosiluokkien 3–6 ja 7–9 käsityön tavoitteissa. Käytännössä ohjelmointia on kuitenkin sovellettu osana kaikkien oppiaineiden opintoja.

Vuosiluokkien 1–2 Tieto- ja viestintäteknologisen osaamisen (L5) kuvauksen mukaan ”oppilaat saavat ja jakavat keskenään kokemuksia digitaalisen median parissa työskentelystä sekä ikäkaudelle sopivasta ohjelmoinnista” (emt. 101). Matematiikan opetuksen tavoitteiden mukaan ”tutustuminen ohjelmoinnin alkeisiin alkaa laatimalla vaihteellaisia toimintaohjeita, joita myös testataan” (emt. 129).

Vuosiluokkien 3–6 Tieto- ja viestintäteknologisen osaamisen (L5) kuvauksen mukaan ”ohjelmointia kokeillessaan oppilaat saavat kokemuksia siitä, miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista” (emt. 157). Matematiikan opetuksen tavoitteisiin kuuluu ”innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmina graafisessa ohjelmointiympäristössä” (emt. 235). Käsityön opetuksen tavoitteisiin liittyviin keskeisiin sisältöalueisiin kuuluu harjoitella ”ohjelmoimalla aikaan saatuja toimintoja, joista esimerkkinä robotiikka ja automaatio” (emt. 271). Matematiikan hy-

vän osaamisen arviointikriteeri 6. vuosiluokan päätteeksi on muun muassa, että ”oppilas osaa ohjelmoida toimivan ohjelman graafisessa ohjelmointiympäristössä” (emt. 239).

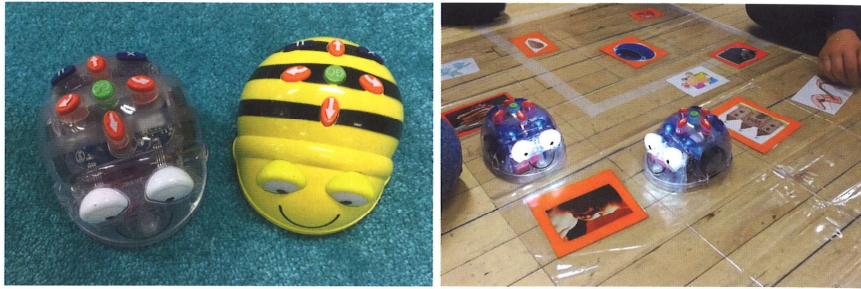
Perusopetuksen opetus suunnitelman perusteiden pohjalta laaditussa Suomen harjoittelukoulujen ohjelmointipolussa (eNorssi 2018) on lisäksi jäsennetty ohjelmoinnin roolia opetus suunnitelmassa sekä eri vuosiluokkien opetuksessa. Vuosiluokkien 1–2 päämääränä on tutustua ohjelmoinnin perusteisiin muun muassa käskyjen, syy-seuraussuhteiden ja ohjeen antamisen kautta. Ohjelmoinnin tavoitteena on erityisesti tutustua ohjelmoinnin perusasioihin, kuten ohjeiden antamiseen, ja saada aikaan positiivisia kokemuksia ohjelmointiin liittyen. Ohjelmoinnin materiaaleina ja välineinä voidaan käyttää monenlaisia pelejä, leikkejä ja kirjoja sekä Blue-bot/Bee-bot-alkeisrobotteja ja graafista eli kuvakepohjaista Scratch Junior -ohjelmointiympäristöä (kuvattu seuraavassa luvussa). Vuosiluokilla 3–4 ohjelmoinnissa tutustutaan muun muassa ohjelmointiin arjessa, luokitteluun ja vertailuun sekä graafiseen ohjelmointiin esimerkiksi Scratch Juniorin kanssa. Vuosiluokilla 5–6 ohjelmoinnissa tutustutaan lisäksi muun muassa robotiikkaan, ehtolauseisiin ja kaavoihin taulukkolaskennassa. Ohjelmoinnin tavoitteena on erityisesti tarkastella ohjelmoinnin läsnäoloa ympäristössä ja kehittää ongelmanratkaisua ja luovuutta. Ohjelmointia rohkaistaan toteutettavan monissa oppiaineissa myös matematiikan ulkopuolella.

## Alkuopetuksen ohjelmointiympäristöjä

Ohjelmoinnin kouluopetukseen tulon myötä tulleet ohjelmoinnin työvälineet (esim. ohjelmointiympäristöt) ovat melko uusia. Siitä huolimatta ohjelmointiin luokiteltavat tehtävät ovat monilta osin tuttuja: ohjelmoinnin opiskelu on ennen kaikkea loogista ajattelua, ohjeiden noudattamista ja niiden antamista, syy-seuraussuhteiden hahmottamista ja sääntöjen merkityksen ymmärtämistä. (eNorssi 2018.).

## Alkeisrobotiikkaa Blue-boteilla/Bee-boteilla

Blue-botit ovat pienehköjä robotteja, joita voidaan ohjelmoida liikkumaan niiden selkämyksessä olevien nuolinäppäinten avulla (kuva 1). Blue-boteihin voidaan myös muodostaa yhteys mobiililaitteen Bluetooth-ominaisuuden välityksellä, jolloin niitä voidaan ohjelmoida myös etäyhteydellä. Blue-boteista on olemassa myös keltaisia Bee-bot-vaihtoehtoja, joita ohjelmoidaan vain robottien selästä ohjaten ilman Bluetooth-ominaisuutta. Blue-botit ovat olleet suosittuja vaihtoehtoja alkeisrobotiikan toteuttamiseen erityisesti peruskoulun alemmilla luokka-asteilla.



KUVA 1. Blue-bot ja Bee-bot (vasemmalla). Kaksi Blue-botia läpinäkyvällä taskumattoalustalla (oikealla).

Leikkisän ajoneuvon kaltaisille Blue-boteille ohjelmoidaan liikettä aikaansaavia algoritmeja painamalla oransseja nuolinäppäimiä peräkkäin halutussa järjestyksessä. Painallukset tallentuvat kyseisessä järjestyksessä robotin muistiin suoritettavaksi ohjelmaksi, ja GO-painiketta painettaessa robotti suorittaa liiketoiminnot yksi kerrallaan siinä järjestyksessä kuin ne on robotin ohjelmamuistiin syötetty. Yhden ”liikkeen” pituus on aina 15 senttimetriä, joten robotille on helppo suunnitella erilaisia alustoja, joissa olevia tasamittaisia ruutuja pitkin robotti voidaan ohjelmoida saavuttamaan tavoitteellisesti erilaisia määränpäitä.

Blue-botien pedagogisena vahvuutena voidaan pitää ennen kaikkea sitä, että ne tuovat oppimisyöskentelyyn toiminnallisuutta ja motivaatiota. Lisäksi robotteja pidetään sympaattisina ja leikinomaisina, joten niitä voidaan mielekkäästi yhdistää esimerkiksi osaksi tarina- ja satuympäristöjä. Robottien ohjelmointi ei myöskään edellytä varsinaisen ohjelmointikielen tuntemusta, vaan liikealgoritmit voidaan laatia laitteelle suoraan painikkeilla.

### Mediaprojektien tekoa Scratch Juniorilla

Scratch Junior on ilmainen, mobiililaitteilla (esim. iPadilla) käytettävä ohjelmointiympäristö, jolla voidaan suunnitella ja toteuttaa omia mediaprojekteja, kuten tarinoita, animaatioita tai pienoisperlejä (kuva 2). Scratch Junior on erityisesti luovan ja tuottavan ohjelmoinnin ympäristö, joka mahdollistaa omaan mielikuvitukseen ja kiinnostuksenkohteisiin perustuvan itseilmaisun. Ohjelmointiympäristöstä löytyy lukuisia erilaisia valmiita hahmoja, kuten ihmisiä ja eläimiä, ja taustoja, kuten satumetsiä ja vedenalaisia maisemia, joita voidaan hyödyntää omassa projektissa. Toisaalta projekteihin voidaan myös esimerkiksi nauhoittaa omaa ääntä tai ottaa omia kuvia. Töitä voidaan myös jakaa toisten mobiililaitteille ja niitä voidaan taltioida mobiililaitteen ruutukaappauksella.



KUVA 2. Scratch Junior ja sillä käytettävä kuvakepohjainen ohjelmointikieli. Ruodon alalaidassa näkyy valitulle hahmolle eli Koira-hahmolle laadittu kolmikaskyinen ohjelma: kun hahmoa klikataan sormella (keltainen käskylohko, aloitusmerkki), se liikkuu viisi askelta oikealle (sininen käskylohko, varsinainen toiminta) ja toistaa kyseistä ohjelmaa ikuisesti (punainen käskylohko, lopetusmerkki).

Ohjelmoinnin tarkoitus Scratch Juniorissa on herättää omaan projektiin valitut hahmot ja tapahtumapaikat henkiin. Ohjelmointiympäristössä käytetään graafista eli kuvakepohjaista ohjelmointia: algoritmit esimerkiksi hahmojen liikuttamiseen, vuoropuhelujen aikaansaamiseen tai äänien ja musiikkien soittamiseen muodostetaan ohjelmointiympäristöstä valmiiksi löytyvillä eri-värisillä ohjelmointilohkoilla (esim. ”liiku”, ”sano”, ”kasvata kokoa”). Lohkot liitetään yhteen tiettyyn järjestykseen erikseen jokaiselle hahmolle, jolloin hahmot suorittavat sille annetut komennot, ja projekti herää eloon. Jokainen algoritmi käynnistyy jostakin tapahtumasta, esimerkiksi hahmon osumisesta toiseen hahmoon tai uuden tapahtumapaikan avautumisesta.

### Ohjelmoinnin yhdistäminen opittavien asioiden harjoitteluun

Seuraavissa luvuissa esittelemme Blue-boteilla ja Scratch Juniorilla toteutettuja ohjelmoinnin oppimistehtäviä alkuopetusluokissa. Tehtävissä ohjelmointi on ollut opittavan sisällön ohella ennen kaikkea työtapaa, joka on tukenut ja motivoinut muiden opetussuunnitelmaan kuuluvien taitojen harjoittelua. Käytännössä toteutettuja ohjelmointitehtäviä on ollut lukuisia, joista olemme koostaneet esimerkkejä tähän artikkeliin seuraavien kokonaisuuksien mukaisesti:

- Lukeminen
- Kirjoittaminen
- Matemaattiset taidot
- Luonnontieteet
- Taito- ja taideaineet

Tehtävissä kuvatut oppimistavoitteet pohjautuvat sekä valtakunnalliseen perusopetuksen opetussuunnitelmaan (Opetushallitus 2014) että Jyväskylän normaalikoulun vastaavaan (Jyväskylän normaalikoulu 2016) ja lisäksi Suomen harjoittelukoulujen valtakunnallisiin oppilaiden TVT- ja ohjelmointitavoitteisiin (eNorssi 2018). Tehtävien esittelyn lisäksi avaamme kolmasluokkalaisten oppilaiden omien Scratch Junior -oppimispelien ohjelmointiprojektin prosessia erityisesti pedagogisesta näkökulmasta valottaaksemme yksityiskohtaisemmin ohjelmoinnissa sovellettua työskentelymallia. Tämä monialainen projekti nitoo yhteen alkuopetusvuosien aikana opitut ohjelmointiperustat, erilaiset oppimistavoitteet sekä vuorovaikutus- ja yhteistyötaidot pienryhmien kautta koko luokan osallistamiseen.

Kaikkien näiden harjoitusten aikana tapahtuvan ohjelmointityöskentelyn pedagogisena perusideana on ollut oppilaiden välinen yhteistyö ja vertaisoppiminen: pohtiminen, oivaltaminen ja yhdessä oppiminen. Opettaja neuvoi aluksi ohjelmoinnista vain esitettyjen oppimistavoitteiden mukaisia perusasioita, kuten sen, mistä erilaiset toiminnot (esim. taustan lisääminen) löytyvät. Sen jälkeen oppilaat yhdessä miettivät, kokeilivat, oivalsivat ja ratkaisivat erilaisia ohjelmoinnillisia haasteita ja asioita omien lähtökohtiensa ja kiinnostuksensa mukaisesti. Tarvittaessa opettaja ohjasi ja opetti myös eriyttään pari- tai ryhmäkohtaisesti eteenpäin. Oppimistuokioiden ja projektien aikana pysähdyttiin välillä myös yhdessä miettimään uusia ratkaisuja sekä jakamaan oivalluksia ja ideoita koko luokalle. Tällaisen yhteistoiminnallisen ja vuorovaikutuksellisen, eri oppimisympäristöissä tapahtuvan työskentelytavan avulla kehittyvät samalla myös lasten kielelliset taidot (Opetushallitus 2014). Oppimisprosesseissa on pyritty huomioimaan oppilaslähtöisyys, motivaatio sekä alkuopetusikäisten lasten kehittyvä taito toimia ja oppia yhdessä. Samalla kun keskustellaan ja harjoitellaan vuorovaikutuksellisia ja yhteistoiminnallisia taitoja, voidaan oppia myös aihekokonaisuuksiin sekä ohjelmointiin liittyviä tavoitteita. Siksi nämä kaikki ohjelmointiympäristöissä tapahtuvat keisit toteutettiin pareittain tai pienissä ryhmissä. Eriyttämissen ja oppimisen kannalta huomioitiin se, että työskentely tapahtui taidollisesti suunnilleen samantasoisissa ryhmissä (Ferguson-Patrick 2007), jotta vertaisoppimisen hyödyt ja oppimista edistävät asiat toteutuivat. Artikkelisiin valitut ohjelmointitehtävät on kirjoitettu ohjeistettuun ja mahdollisimman selkeään muotoon, jotta lukijan olisi halutessaan mahdollisimman helppo kokeilla ja toteuttaa harjoituksia omiin tarpeisiinsa soveltaen.

## Lukeminen

Opetussuunnitelman laaja-alaisissa osaamistavoitteissa lukutaidon käsite laajennetaan monilukutaidoksi L4, joka perustuu niin sanottuun laaja-alaiseen tekstikäsitteeseen. Sanallisten, kuvallisten, auditiivisten, numeeristen ja kinesteettisten symbolijärjestelmien sekä näiden yhdistelmien avulla ilmaista tietoa eli tekstiä voidaan tulkita ja tuottaa esimerkiksi kirjoitetussa, puhutussa, painetussa, audiovisuaalisessa tai digitaalisessa muodossa. Monilukutaito käsitetään taidoksi hankkia, yhdistää, muokata, tuottaa, esittää ja arvioida tietoa eri muodoissa, tilanteissa ja ympäristöissä erilaisten välineiden avulla. Käytännössä oppilaat harjoittelevat näitä taitoja perinteisten oppimisympäristöjen lisäksi myös teknologiaa eri tavoin hyödyntävissä oppimisympäristöissä. (Opetushallitus 2014.) Ohjelmoinnin avulla on mahdollista harjoitella lukutaidon eri osa-alueita, kuten teknistä lukemista, luetun ymmärtämistä ja tekstin analysointia, monilukutaidon näkökulmasta.

### **Tehtävä 1: Eriyttäviä lukutehtäviä ohjelmoiden (Blue-bot ja satataulualusta)**

Lukemisen oppimistavoitteet

- kirjainten nimeäminen
- tavujen, sanojen ja virkkeiden sujuva lukeminen

Ohjelmoinnin oppimistavoitteet

- yksinkertaiset käskyt (liikekomennot) ja niiden antaminen ohjeina (käskysarjoina) botille, jotta se löytää satataululle asetettujen tehtävien luo ja osaa pysähtyä

Tehtävä tehdään parityöskentelynä kielellisesti samantasoisien parin kanssa tason mukaan tehtäviä eriyttäen. Opettaja jakaa oppilaat pareiksi etukäteen ja suunnittelee parille soveltuvat lukutehtävät (n. 10–20 tehtävää). Tehtävissä on erilaisia lukutaitoa harjoittavia tehtäviä aina kirjainten nimeämisestä tavu-, sana- ja virketasoiseen lukemiseen. Jos satatauluja on vain yksi, parien tehtävät voidaan merkitä erivärisillä lapuilla, jolloin saman maton ympärillä voi useampi pari työskennellä samaan aikaan. Botin ohjelmoidaan yhdessä lähtöpaikalta jokaisen tehtävän kohdalla tehtävalapun luokse. Liikekäskysarjojen suunnittelun ja botin muistiin syöttämisen apuna voidaan suunnitella reitti etukäteen kirjoittamalla tarvittavat liikekäskyt oikeassa järjestyksessä (ts. algoritmi) muistiin paperille. Tehtävät luetaan vuorotellen paria arvioiden ja auttaen, ja jos lukeminen onnistuu sujuvasti, saa lukija lapun itselleen. Jos parit saavat kerättyä kaikki laput, he voivat keksiä niistä yhdessä suullisesti tai kirjoittaen pienen tarinan.



KUVA 3. Lukutehtäviä ohjelmoiden Blue-botilla satataulualustalla.

### Tehtävä 2: Luetun tekstin animointi (Scratch Junior)

Lukemisen oppimistavoitteet

- annetun tekstin ymmärtäminen lukemalla

Ohjelmoinnin oppimistavoitteet

- vaadittavien taustojen ja hahmojen luominen
- ohjelointi graafisessa ohjelointiympäristössä toimintaan tarvittavia käskyjä ja ohjelmoinnin perusrakenteita (esim. aloitusmerkki, kuten tapahtuma, varsinainen toiminta käskysarjana, lopetusmerkki) soveltamalla tekstissä kerrotun mukaisesti

Tehtävä tehdään parityöskentelynä kielellisesti samantasoisien parin kanssa tason mukaan eriyttäen. Opettaja jakaa oppilaat pareiksi etukäteen ja valitsee pareille soveltuvan tekstin. Tähän sopii tekstiksi aapisen tai lukukirjan kertomusten lisäksi esimerkiksi pieni satu, katsomusaiheinen kertomus tai uutinen. Parit lukevat tekstin ääneen vuorotellen virke kerrallaan, minkä jälkeen he tekevät tekstistä yhteisen animaation Scratch Juniorilla. Hahmot voidaan ohjelmoida aloittamaan toimintaansa esimerkiksi silloin, kun niitä klikataan, tai ajoittamalla. Lisäksi ne voidaan ohjelmoida esimerkiksi liikkumaan, hyppimään, keskustelemaan ja päästelemään ääniä sekä muuttamaan muotoaan erilaisissa tapahtumapaikoissa. Animaatiot esitetään lopuksi yhteisesti luokalle.



Kuva 4. Luetunymmärtämisen harjoittelua ohjelmoinnin avulla

### Tehtävä 3: Reseptin tai muun ohjeistuksen animointi (Scratch Junior)

Lukemisen oppimistavoitteet

- annetun tekstin/kuvien/selostusten ymmärtäminen lukemalla (monilukutaito)

Ohjelmoinnin oppimistavoitteet

- vaadittavien taustojen ja hahmojen luominen
- animoida tekstillä ja kuvilla ohjeistus toimintaan tarvittavia käskyjä ja ohjelmoinnin perusrakenteita soveltamalla kerrotun mukaisesti kuvitetusta tekstistä

Tehtävä tehdään parityöskentelynä kielellisesti samantasoisien parin kanssa tason mukaan eriyttäen. Opettaja jakaa oppilaat pareiksi etukäteen ja valitsee pareille soveltuvan ohjetekstin, esimerkiksi reseptin tai askarteluohjeen. Parit lukevat tekstin ääneen vuorotellen ja kertovat kuvista toisilleen. Sen jälkeen he tekevät tekstistä yhteisen animaation Scratch Juniorilla. Animaatioissa on tarkoitus käyttää myös äänitysominaisuutta. Lopuksi ohjeistusta voidaan noudattaa käytännössä esimerkiksi leipoen tai askarrella.

### Kirjoittaminen

Laajan tekstikäsityksen mukaan alakoulussa tekstien tuottamisen taitojen opetuksen tavoitteena on rohkaista ja innostaa oppilaita kertomaan, kuvaamaan ja ilmaisemaan kokemuksiaan, ajatuksiaan, mielipiteitään ja tarinoitaan puhumalla ja kirjoittamalla samalla vahvistaen myönteistä minäkuvaansa erilaisten tekstien tuottajana (Opetushallitus 2014). Tärkeää on myös tarjota kannustaen ja ohjaten mahdollisuuksia harjoitella kielentämään omia ajatuksia ja tuottamaan omantansoisia kertovia, kuvaavia, ohjaavia ja kantaa ottavia tekstejä myös monimediaisissa ympäristöissä. Motivaatio kirjoittamistehtävää kohtaan on myös oleellista (Hidi & Boscolo 2008).

### Tehtävä 4: Kirjoittamis- ja tavutehtäviä ohjelmoiden (Blue-bot)

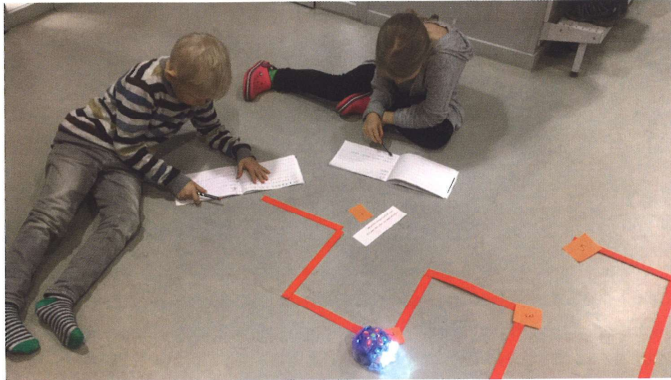
Kirjoittamisen oppimistavoitteet

- sanojen tavuttaminen
- sanojen ja tavujen kirjoittaminen
- virkkeiden löytäminen tekstistä ja niiden kirjoittaminen

Ohjelmoinnin oppimistavoitteet

- yksinkertaiset käskyt (liikekomennot) ja niiden antaminen ohjeina (käskysarjoina) botille, jotta se löytää labyrintissa seuraavan tehtävän luo ja osaa pysähtyä siihen

Tehtävä tehdään parityöskentelynä. Tehtäviä vaihdellaan harjoiteltavan asian mukaisesti kielellisesti samantasoisien parien kanssa tason mukaan eriyttäen. Opettaja jakaa oppilaat kielellisiltä taidoiltaan samantasoisien kanssa pareiksi



Kuva 5. Kirjoitusharjoituksia ohjelmoiden

etukäteen ja suunnittelee parille soveltuvat tehtävät (n. 5 tehtävää). Oppilaat saavat paperiliuskoja (leikattu A4-paperiarkeista pituussuunnassa, joka on kaksi kertaa botin kulkema yksittäinen matka eli 15 senttimetriä), joista he muodostavat yhdessä labyrintin. Tehtävälaput asetellaan reitille. Boti ohjelmoidaan yhdessä ensimmäisen tehtävän luokse, ja kun se onnistuu, tehtävä tehdään, käydään tarkistuttamassa se opettajalla tai tarkistuspuolella ja jatketaan eteenpäin.

#### **Tehtävä 5: Virkekirjoittamista ohjelmoiden (Scratch Junior)**

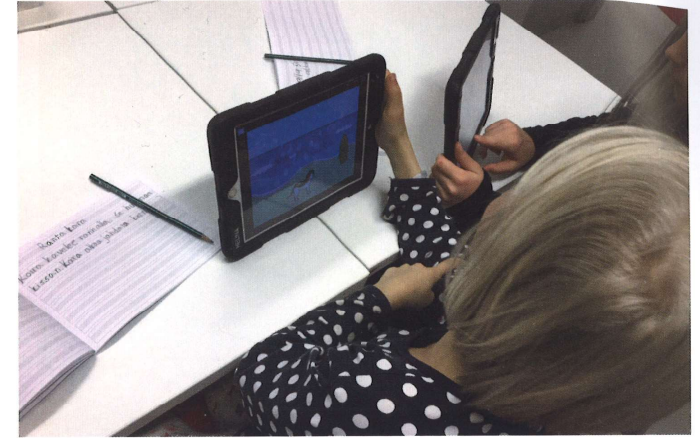
Virkekirjoittamisen oppimistavoitteet:

- virkkeen elementtien muistaminen (iso alkukirjan, lopetusmerkki ja välille toiminta, jota virkkeessä tarkoitetaan) ja kirjoittaminen

Ohjelmoinnin oppimistavoitteet:

- algoritmin eli ”ohjelmointilausekkeen” muodostaminen niin, että siinä on toimintaan tarvittavat käskyt (aloitusmerkki, kuten tapahtuma, varsinaisen toiminta käskysarjana sisältäen erilaisia ohjelmoinnin perusrakenteita, lopetusmerkki)

Harjoitus aloitetaan koko luokan kanssa yhteisesti mallintaen: jokaisella oppilaalla on kynä, kumi, kirjoitusvihko ja oma iPad, jossa Scratch Junior -ohjelma on auki. Keksitään yhdessä ensimmäinen virke niin, että siinä on jokin tekijä, jokin tekeminen ja jokin paikka (esim. ”Kissa kävelee rannalla.”). Jokainen kirjoittaa virkkeen vihkoonsa, ja virkkeet tarkistetaan opettajan taululle tekemästä mallista. Virkkeeseen valitaan myös sopiva hahmo ja tausta Scratch Juniorissa ja virke ohjelmoidaan. Virkettä ja ohjelmoitua lauseketta verrataan yhdessä: molemmissa tulisi olla aloitusmerkki (esim. kun hahmoa painetaan), toiminta (esim. hahmo liikkuu ja sanoo jotakin) ja lopetusmerkki (esim. ohjelma päättyy ilman erityistoimenpiteitä). Tehtävää voi jatkaa vielä yhdessä, mutta vähitellen on tarkoitus, että sitä jatketaan yhdessä parin kanssa eteenpäin virke kerrallaan



Kuva 6. Opetellaan kirjoittamaan virkeitä sekä kynällä että ohjelmointikielillä.

kirjoittaan ja ohjelmoiden. Taitavammat parit voivat yhdistää virkkeensä tarinaksi ja näin muodostaa yhtenäisen animaation.

#### **Tehtävä 6: Kertomuksen yhteistoiminnallinen kirjoittaminen (Blue-bot tai Scratch Junior)**

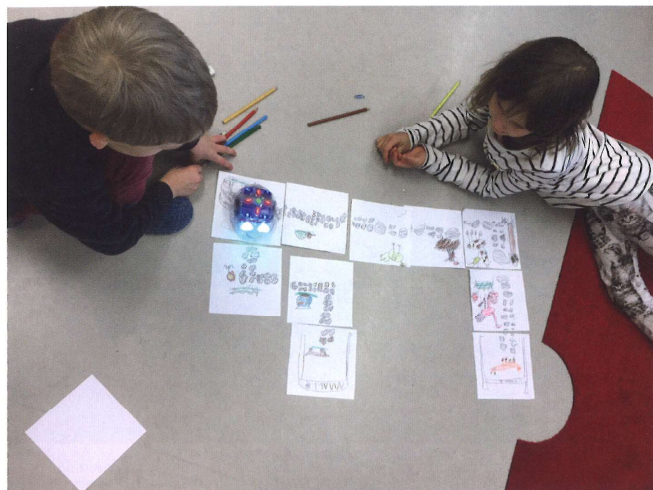
Kirjoittamisen oppimistavoitteet

- aiheeseen liittyvän johdonmukaisen juonellisen kertomuksen kirjoittaminen

Ohjelmoinnin oppimistavoitteet

- (Blue-bot) yksinkertaiset käskyt (liikekomennot) ja niiden antaminen ohjeina (käskysarjoina) botille, jotta se kulkee tarinan mukaisesti esimerkiksi paperiliuskoista muodostetussa labyrintissä tai ruutumatolla reittiä eteenpäin
- (Scratch Junior) vaadittavien taustojen ja hahmojen luominen
- (Scratch Junior) tekstin animointi toimintaan tarvittavia käskyjä ja ohjelmoinnin perusrakenteita soveltamalla kuvauksen mukaisesti

Tehtävä tehdään parityöskentelynä opettajan valitseminen kielellisesti samantasoisten pariin kanssa. Ensin parit luonnostelevat juonikaavion, jonka pohjalta suunnittelevat ja kirjoittavat yhdessä pienen kertomuksen joko tekstinkäsittelyohjelmalla tai kynällä. Kertomuksen teemana voi olla esimerkiksi luontopolku tai muu tavaroiden kerääminen. Kun pari kirjoittaa, toinen tarkistaa ja korjaa tarvittaessa. Valmis tarina ohjelmoidaan valitulla tavalla. Luontopolulla ohjelmitava hahmo voi esimerkiksi pysähtyä keräämään asian tai ”ihaillemaan perhosta”. Botilla ohjelmoidessa asiat voivat olla ihan konkreettisesti reitillä, ja niiden kohdalle ohjelmoidaan pysähtyminen. Tarinaa luetaan Botin edetessä.



Kuva 7. Ohjelmointia juonellista ja kuvitettua tarinaa mukaillen

### Tehtävä 7: Vuoropuhelun harjoittelu (Blue-bot / Scratch Junior)

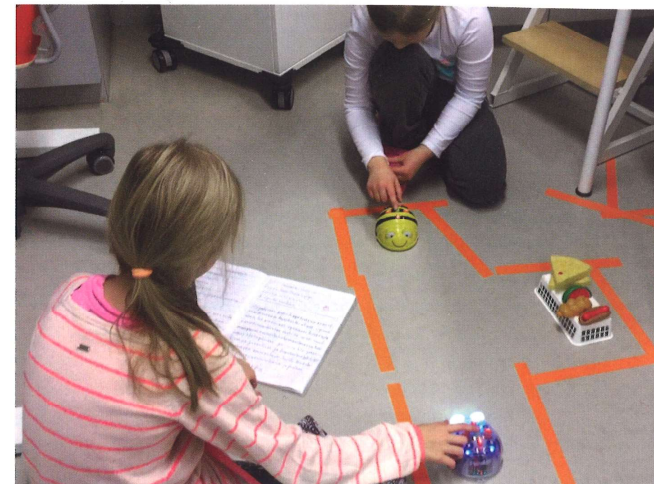
Kirjoittamisen oppimistavoitteet:

- vuoropuhelun kirjoittaminen johtolauseita ja vuorosanaviivoja käyttäen

Ohjelmoinnin oppimistavoitteet:

- (Blue-bot) yksinkertaiset käskyt (liikekomennot) ja niiden antaminen ohjeina (käskysarjoina) botille, jotta se kulkee tarinan mukaisesti esimerkiksi labyrintissä tai matolla reittiä eteenpäin
- (Scratch Junior) vaadittavien taustojen ja hahmojen luominen
- (Scratch Junior) tekstin animointi toimintaan tarvittavia käskyjä ja ohjelmoinnin perusrakenteita soveltamalla kuvauksen mukaisesti

Tehtävä toteutetaan vastaavalla tavalla kuin edellä esitetty Tehtävä 6 (Kertomuksen yhteistoiminnallinen kirjoittaminen). Kirjoitettu kertomus sisältää vuoropuhelun. Ohje voi olla esimerkiksi: ”Kirjoittakaa kertomus, jossa kaksi hahmoa lähtevät liikkeelle eri paikoista, kohtaavat matkan varrella, keskustelelevat keskenään ja jatkavat yhdessä eteenpäin tiettyyn määrän päähän.” Blue-boteilla ohjelmoidessa huomioidaan keskustelun aikainen pysähtyminen eli ohjelman lopetus tai ajallinen tauotus sekä erillisten ohjelmien (esim. eri hahmoissa tai boteissa) synkronointi. Tehtävä voidaan esittää lukemalla kertomus bottien liikkuessa tai videoida siten, että toinen oppilaista lukee samalla, kun toinen kuvaa. Scratch Junior -ohjelmalla tehdessä kertomukseen ohjelmoidaan puhelukuplat ja keskustelun voi myös äänittää.



Kuva 8. Botit tapaavat matkalla ja lähtevät yhdessä kauppaan.

### Matemaattiset taidot

Ohjelmointi on perinteisesti mielletty sisältyvän oleellisesti osaksi matematiikan oppiainetta, ja monet matematiikan oppikirjat ovat ottaneet osa-alueekseen ohjelmoinnin. Opetussuunnitelmassa ohjelmoinnille asetetaan matemaattisten taitojen kohdalla tavoitteet ja sisältöalueet erityisesti ajattelun taitoihin liittyen myös jo alkuopetusluokkien osalta (Opetushallitus 2014). Itse ohjelmointitaitojen opetteluun lisäksi myös muiden matemaattisten taitojen ja tavoitteiden opettelemisen apuna voidaan hyödyntää ohjelmointia.

### Tehtävä 8: Lasku ja vastaus (Blue-bot)

Matematiikan oppimistavoitteet:

- aihealueen laskujen harjoittelu ja vahvistaminen

Ohjelmoinnin oppimistavoitteet:

- ohjelman suunnittelu
- yksinkertaiset käskyt (liikekomennot) ja niiden antaminen ohjeina (käskysarjoina) botille, jotta se löytää satataululle asetettujen tehtävien luo ja osaa pysähtyä

Jokaisella työskentelyparilla on oma botti. Opettaja on tehnyt valmiiksi paperilapuille teemaan liittyviä harjoiteltavia laskuja, kuten yhteen-, vähennys- tai kertolaskuja, joita on asetettu nurinperin satataulualustalle. Parit ohjelmoivat bottinsa vuoron perään lapulle ja ratkaisevat laskun, kunnes alusta on tyhjä. Jos vastaus on oikein, pari saa laskun itselleen. Harjoitteesta saadaan motivoivampi tekemällä siitä kahden parin välinen kisa tai eriyttämällä lasku-



tehtävät pareille sopiviksi. Liikekäskysarjojen suunnittelun ja botin muistiin syöttämisen apuna voidaan suunnitella reitti etukäteen kirjoittamalla tarvittavat liikekäskyt oikeassa järjestyksessä (ts. algoritmi) muistiin paperille.

### Tehtävä 9: Ohjelmointirivi (Blue-bot)

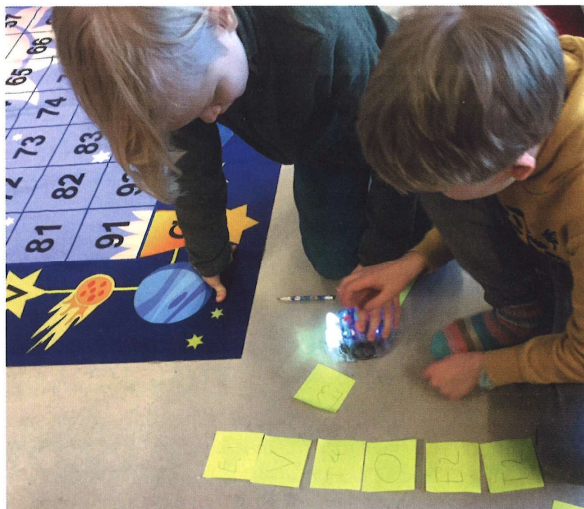
Matematiikan oppimistavoitteet:

- käskysarjan visuaalinen hahmottaminen
- yksinkertaisimman ohjelmointireitin suunnittelu
- käskysarjan toteuttaminen visuaalisesti

Ohjelmoinnin oppimistavoitteet:

- yksinkertaiset käskyt (liikekomennot) ja niiden antaminen ohjeina (käskysarjoina) botille, jotta se löytää satataululle asetettujen tehtävien luo ja osaa pysähtyä
- ohjelman kirjoittaminen
- ohjelman testaaminen ja virheidenkorjaus ("debuggaaminen")

Satataulualustalle laitetaan merkki johonkin tiettyyn ruutuun, joka on etäällä botin aloituspisteestä. Pari kirjoittaa käskysarjan paperille miettien, mikä on yksinkertaisin reitti kulkea lähtöpaikasta kohderuutuun. Sen jälkeen he ohjelmoivat bottinsa ohjeen mukaan ja kokeilevat, pääseekö botti perille vai tuleeko ohjelmointiohjetta korjata. Virheidenkorjaus eli debuggaaminen on tärkeä ohjelmoinnillinen taito, josta on hyötyä kaikessa ohjelmoinnissa tarkemman ja tehokkaamman lopputuloksen varmistamiseksi.



Kuva 9. Ohjelmointirivin harjoittelua teoriassa ja käytännössä

## Ympäristööppi

Luonnonilmiöitä tutkittaessa on oleellista niiden syy-seuraussuhteiden ymmärtäminen. Ilmiöiden visualisoiminen auttaa niiden hahmottamisessa ja ymmärtämisessä. Ohjelmoinnin mahdollistama ilmiön animointi tukee hyvin luonnonilmiöiden oppimista. Ohjelmoinnin hyödyntäminen osana ympäristööppiä tukee myös opetus suunnitelman asettamaa tutkimisen ja toimimisen taitojen tavoitetta ohjata oppilasta käyttämään tieto- ja viestintäteknologiaa tiedon hankkimisessa sekä havaintojen taltiointisessa ja esittämisessä (Opetushallitus 2014).

### Tehtävä 10: Animoil luonnonilmiö (Scratch Junior)

Ympäristööpin oppimistavoitteet:

- käsiteltävään luonnonilmiöön (esim. veden kiertokulku, perhosen muodonvaihdos) tutustuminen
- luonnonilmiöön keskeisesti liittyvien käsitteiden ymmärtäminen ja sen perusasioista kertominen

Ohjelmoinnin oppimistavoitteet:

- vaadittavien taustojen ja hahmojen luominen
- animoida tekstillä, kuvilla ja äänittämällä kyseinen luonnonilmiö tarvittavia käskyjä ja ohjelmoinnin perusrakenteita (esim. interaktiiviset tapahtumat) soveltamalla

Oppilaat tutustuvat itse, pareittain, pienissä ryhmissä tai opettajajohtoisesti käsiteltävään luonnonilmiöön. Ilmiöön tutustumisen jälkeen siitä ohjelmoidaan opetusanimaatio, jossa kyseistä luonnonilmiötä täydennetään kuvien lisäksi tekstillä ja äänellä. Animaatioon voidaan myös ohjelmoida interaktiivisia ominaisuuksia: esimerkiksi klikkaamalla animaation eri osia animaation voi ilmestyä tarkentavia lisätietoja kyseisestä luonnonilmiöstä puhekuplien, äänen tai muiden keinojen muodossa.

### Tehtävä 11: Oma Suomi-peli (Scratch Junior)

Ympäristööpin oppimistavoitteet:

- Suomen karttaan tutustuminen (mm. Suomen rajat ja rajanaapurit)
- Suomen pääkaupunkiin ja suurimpiin kaupunkeihin tutustuminen
- Suomen pinnanmuotojen tunteminen

Ohjelmoinnin oppimistavoitteet:

- peli-idean keksiminen ja ohjelmoitavan pelin suunnittelu
- animoida tekstillä, kuvilla ja äänittämällä esittelyanimaatio Suomen suurimmista kaupungeista tarvittavia käskyjä ja ohjelmoinnin perusrakenteita soveltamalla

Tämä monialainen ja pitkäkestoinen Suomi-oppimisprojekti toteutetaan ryhmätöinä ja siinä sovelletaan luonnontieteen ja ohjelmoinnin lisäksi myös kuvataiteen keinoja. Projektiä ennen Suomea on käsitelty ympäristöopissa maantieteellisesti. Ryhmät hahmottelevat pahvialustalle Suomen ääriiviat ja maalaavat Suomen pinnanmuotojen mukaisin värein. Sen jälkeen he merkitsevät maarajat punaisella ja nimeävät naapurivaltiot. Suomen karttaan merkitään oikealle kohdalle pääkaupunki Helsinki ja muut määrätyt Suomen väkiluvultaan suurimmat kaupungit. Ryhmä suunnittelee karttapohjaan haluamansa kaltaisen peli-idean. Kartta muokataan pelinomaiseksi lisäämällä siihen erilaisia keskeisiä pelielementtejä, kuten pelihahmoja ja tehtäväpaikkoja. Peliin tarvittavat hahmot ja muu rekvisiitta askarrellaan erikseen. Peli-idean yhteiseksi ohjeeksi annetaan se, että kun pelaaja saapuu johonkin kaupunkiin, tutustutaan kyseisen kaupungin esittelyanimaatioon ja ratkaistaan kaupunkiin liittyviä tehtäviä tai vastataan kysymyksiin. Ryhmät animoivat karttaan merkityistä kaupungeista Scratch Juniorilla yhdessä ideoidut esittelyt. Ohjelmoimalla voidaan myös toteuttaa kokonainen peli.



Kuva 10. Kaupunkianimaation suunnittelua omaan peliin

### Taito- ja taideaineet

Ohjelmointia voi mainiosti yhdistää myös moniin taito- ja taideaineisiin ja erityisesti monialaisiin oppimiskokonaisuuksiin täydentämään oppimistavoitteita (Opetushallitus 2014). Ohjelmoinnin avulla erilaisiin taito- ja taideaineiden projekteihin saadaan liitettyä järkevästi tvt-taitoja, ja esimerkiksi animointi täydentää projektin visualisointia.

### Tehtävä 12: Draamaharjoituksia (Scratch Junior)

Ilmaisutaidon oppimistavoitteet:

- yksin tai ryhmässä esittäminen ja itseilmaisuus
- oma esityksen tai pienen näytelmän (esim. tekstin pohjalta) suunnittelu ja toteuttaminen

Ohjelmoinnin oppimistavoitteet:

- vaadittavien taustojen ja hahmojen luominen
- animoida tekstillä, kuvilla ja äänittämällä annettu teksti, tilanne tai teksti tarvittavia käskyjä ja ohjelmoinnin perusrakenteita (esim. synkronointi) soveltamalla

Tehtävää voi soveltaa itsensä esittelyyn tai jonkin ilmiön dramatisointiin (esim. katsomukselliset kertomukset, koulukiusaaminen). Ohjelmointia voidaan tehdä joko ennen, jälkeen tai samanaikaisesti esityksen kanssa, ja se antaa kuvallisen lisän ilmaisutaitoon sekä voi olla rohkaisevana tukena esityksessä. Scratch Juniorilla animoidaan esimerkiksi oma esittely, jokin katsomuksellinen kertomus tai koulukiusaamistilanne. Animointi voi olla alustuksena itse draamalliselle kohtaukselle tai toimia loppuratkaisuna: esimerkiksi koulukiusaamistilanne voidaan ensin esittää draaman keinoin, ja ryhmässä tai itsenäisesti ohjelmoidaan kiusaamistilanteeseen ratkaisu. Erityisesti hahmojen välisten puheenvuorojen synkronointi ohjelmoimalla tulee keskeiseksi.

### Tehtävä 13: Liikunnallinen tuokio animoiden (Scratch Junior)

Liikunnan oppimistavoitteet:

- pienen liikunnallisen tuokion tai ohjeistuksen (esim. pallopeleihin, hippaan) suunnittelu ja toteuttaminen

Ohjelmoinnin oppimistavoitteet:

- vaadittavien taustojen ja hahmojen luominen
- tuottaa ohjeistusanimaatio sekä omia ja annettuja taustakuvia ja hahmoja käyttämällä ja tarvittavia käskyjä ja ohjelmoinnin perusrakenteita (esim. synkronointi) soveltamalla

Tehtävän voi tehdä yksin, pareittain tai pienissä muutaman hengen ryhmässä. Suunnitellaan yhdessä esimerkiksi pallopeleihin liittyvä liikuntatuokio ja ohjelmoidaan se käyttämällä tilanteeseen sopivia taustakuvia, hahmoja ja välineitä. Erityisesti usean hahmon (pelaajat, pelivälineet) liikkeiden synkronointi ohjelmoimalla tulee keskeiseksi. Luotu animaatio esitetään ja tuokion/ohjeen havainnollistaminen tehdään animaation lisäksi liikkuen.

#### **Tehtävä 14: Stailaa unelmien huone, luokkatila tms. (Scratch Junior)**

Kuvataiteen oppimistavoitteet:

- huoneen sisustuksen ja värimaailman ideointi ja suunnittelu
- pohjapiirroksen tutustuminen
- kolmiulotteisuuden ja mittasuhteiden hahmottaminen
- kuvasta kolmiulotteisen rakennelman askarteleminen

Ohjelmoinnin oppimistavoitteet:

- taustakuvan muokkaaminen
- valokuvan ottaminen ja sen liittäminen osaksi taustaa (esim. keskustaksi taulun kehyksiin taustakuvan seinällä)

Projekti tehdään itsenäisesti tai parin kanssa hyödyntäen Scratch Juniorin taustakuvia, joihin liitetään omia kuvia. Sovelluksesta etsitään sopiva taustakuva (esim. luokkatila), jota muokataan esimerkiksi kalusteita siirtämällä, poistamalla tai muuttamalla niiden väritystä. Samoin kuvaan lisätään omia kuvia esimerkiksi tauluiksi ja huonekaluiksi. Kun huone on stailattu mieleiseksi, suunnitellaan ja toteutetaan askarrellen pieneen laatikkoon sen mukainen huone erilaisia materiaaleja hyödyntäen. Huonekalut pyritään tekemään mahdollisuuksien mukaan kuvan mukaisessa mittakaavassa kolmiulotteisesti.

#### **Tehtävä 15: Oma robotti (Scratch Junior)**

Kuvataiteen oppimistavoitteet:

- kierrätysmateriaaleista suunnitelman mukaisen robotin suunnittelu ja rakentaminen
- erilaisten materiaalien yhdistäminen järkevällä tavalla

Ohjelmoinnin oppimistavoitteet:

- uuden hahmon luominen omasta työstä
- hahmon ohjelmointi liikkumaan ja suorittamaan annettuja tehtäviä erilaisia käskyjä ja käskysarjoja noudattamalla

Tutustutaan erilaisiin arjessa esiintyviin roboteihin. Oppilaat suunnittelevat pienessä ryhmässä piirtämällä oman tulevaisuuden robotin. Robotille voidaan antaa nimi, kuvaus ja käyttötarkoitus: mitä se tekee, miten se liikkuu, mitä hyötyä siitä on. Robotille voidaan keksiä korkealentoisiakin ominaisuuksia ja toiminnallisuksia (esim. kutistuminen ja kasvaminen, lentäminen). Robotti rakennetaan kierrätysmateriaaleista. Valmiista robotista luodaan hahmo Scratch Junior -ohjelmaan ja se animoidaan ohjelmoimalla toteuttamaan erilaisia tehtäviä.



Kuva 11. Robotin toimintojen suunnittelua ja ohjelmoimista ryhmässä

#### **Monialainen oppimiskokonaisuus: Oma peli ohjelmoimalla**

Edellä kuvattujen harjoitteiden kautta ohjelmoinnin perusteita alkuopetusvuosina oppineet oppilaat kehittivät kolmannella luokalla oppilaslähtöisesti monialaisen oppimiskokonaisuuden. Peliprojekti käynnistyi oppilaiden omista ideoista: eräs oppilas halusi luokkakaverinsa kanssa esitellä kotona tekemänsä pelin. Oppimiskokonaisuuden tarkoituksena oli suunnitella ja toteuttaa pienryhmissä omat pelillisyyden kautta motivoivat Scratch Junior-sovelluksella tehdyt oppimispelit. Projektin alkaessa jokainen oppilas tunsivat sovelluksen vähintäänkin perustasolla, joskin suurin osa oppilaista osasi ohjelmoida jo varsin kekseliäästi. Projektin aikana oppilaat oivalsivat kuitenkin monia uusia ohjelmoinnillisia keinoja (esim. ohjelmointirakenteita) sekä itsenäisesti että yhdessä.

Peliprojektin tavoitteena yksinkertaistetusti oli suunnitella ja toteuttaa jokin oppimispeli yhdessä oppimisen ja jakamisen hengessä. Jokainen oppilas kirjoitti projektista BookCreator-ohjelmalla kuvin täydennettyä päiväkirjaa, johon tallennettiin muun muassa pelin suunnitteluideat, peli-idea, kehitysvaiheet ja pohdintoja toteutuksesta ja onnistumisesta. Pelejä esiteltiin laajemmalle yleisölle Jyväskylän yliopistolla valtakunnallisessa ainedidaktiikan symposiumissa sekä omalla koululla järjestetyssä LUMA StarT -gaalassa, jossa myös yleisö pääsi kokeilemaan pelejä.

Oppimispelien suunnittelu ja toteuttaminen ohjelmoimalla oli oppilaista hyvin motivoivaa. Peleistä tuli hyvin pelattavia ja erilaisia, eri oppisisältöjä harjaannuttavia oivaltavia oppimispelisiä. Menestyksekkään lopputuloksen lisäksi koko projektin aikana heijastui vahvasti yhteistoiminnallisuus ja oppilaiden vahva osallisuus, jotka kuuluvat oleellisesti sekä uuden opetus suunnitelman sisältöihin (Opetushallitus 2014) että 2000-luvun taitoihin (Binkley ym. 2012). Samoin projektissa yhdistyivät autenttisella tavalla monet oppiaineet, kuten matematiikka, äidinkieli, liikunta ja ympäristöoppi, yhdessä tv- ja ohjelmointitaitojen harjoittelun kanssa. Päiväkirjaa kirjoittamalla oppilaat saivat projektinsa myös dokumentoitua vaihe vaiheelta<sup>2</sup>.

### 1. viikko: Suunnitteluvaihe

Omien pelien ideoiminen aloitettiin miettimällä oppilaiden kanssa yhdessä, millainen on hyvä oppimispeli ja miten pelistä voi tehdä motivoivan, kiinnostavan ja opettavaisen. Tutustuimme kokeillen ja pelaillen erilaisiin lautapeleihin, koodauspeleihin, ongelmanratkaisupeleihin ja liikunnallisiin peleihin. Jokainen oppilaista kirjoitti tekstinkäsittelyohjelmalla oman pelin suunnitelman. Lopuksi hyvän pelin ominaisuuksia koottiin padlet-seinälle omilla tableteilla.

Nopeus	Kanavama	Tavoit
Liikunta	Reitike vika avaka	Kissapomppu
Labyrintti	Liikunta peli	SAÄNNÖT
Vakautta mutta ei liian vaikea	Reilu peli	YKSINKERTAISET SAÄNNÖT
1. Tiimi	Hauskuus	Mieltava hait palko
Säännöt	Yösi kiva peli	Juoni
Juoni/tarina	Häpkyä tavoite	Online peli
	Jännä	

Kuva 12. Oppilaiden ideointia hyvän pelin ominaisuuksista

### 2. viikko: Pelitiimien valinta ja suunnittelu ryhmittäin

Jokainen oppilas teki oman pelisuunnitelman, ja ne esiteltiin toisille. Samantapaisten suunnitelmien tekijöiden perusteella muodostettiin 2–4 oppilaan kokoiset suunnittelutiimit. Peli-ideoihin kuului muun muassa erilaisia la-byrinttepelejä, tietovisoja, matemaattisia pelejä sekä eläimiin liittyviä pelejä. Tiimit yhdistivät ideansa ja alkoivat suunnitella yhteistä oppimispeliään.

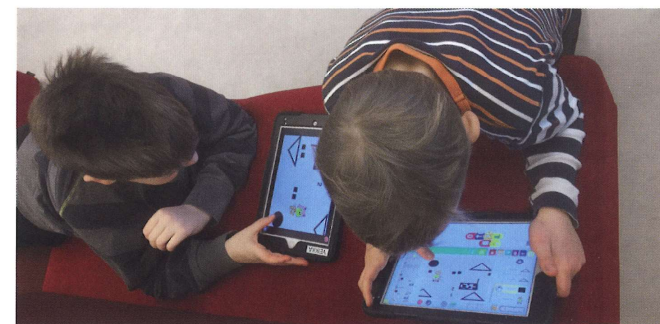
<sup>2</sup> Oma peli ohjelmoimalla -projektiä on esitelty lisäksi videolla osoitteessa [http://bit.ly/pellit2018\\_jy-norssi](http://bit.ly/pellit2018_jy-norssi)



Kuva 13. Yhteissuunnittelua ja ideoiden yhdistämistä

### 3. ja 4. viikko: Pelien työstöä ja päiväkirjan kirjoittamista

Tiimit suunnittelivat, ohjelmoivat, toteuttivat, kokeilivat ja kehittivät peliään yhdessä kahden viikon ajan. Suunnitelmat ja työstövaiheet dokumentoitiin päiväkirjoihin kuvaten ja kirjoittaen. Tuokioiden aluksi ja lopuksi koko luokka kokoontui keskustelemaan ja jakamaan ideoita yhdessä, mikä oli tärkeä osa ohjelmointitaitojen oppimisessa ja kehittämisessä. Oppilaat kertoivat, millaisia haasteita heille oli tullut vastaan peliä ohjelmoimissaan ja miten he olivat asian ratkaisseet. Ratkaisemattomia haasteita mietittiin yhdessä.



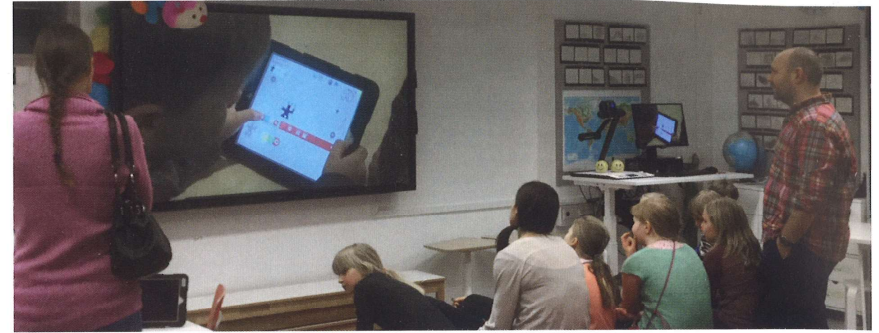
Kuva 14. Pelin suunnittelua testaten ja yhdessä kehitellen



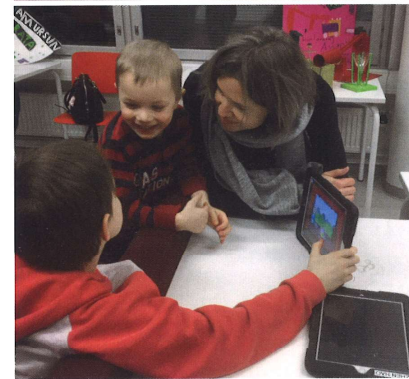
Kuva 15. Peliprojektin kaikki vaiheet dokumentoitiin päiväkirjaan

### 5. ja 6. viikko: Pelien valmistuminen ja esittely

Pelien valmistuttua jokainen ryhmä suunnitteli pelinsä taustalle oman pelimusiikin GarageBand-ohjelman avulla. Aluksi ryhmät esittelivät moninaiset ja visuaalisesti näyttävät valmiit pelit ja päiväkirjansa omalle luokalle. Valmiiden pelien joukossa oli muun muassa Suomeen liittyviä tietokilpailuja, labyrinteja, interaktiivinen koiranhoitopeli sekä matematiikkapelejä. Esittelyn ohessa valmistauduimme Jyväskylän yliopistolla järjestettyyn symposiumiin, jossa jokainen halukas tiimi sai esitellä pelinsä. Lisäksi Jyväskylän normaalkoulussa järjestettiin keväällä 2018 koulun yhteinen gaala valtakunnalliseen LUMA StarT -hankkeeseen liittyen. Oppilaat esittelivät gaalassa peliprojektiaan ja pelejään, ja yleisö pääsi pelaamaan niitä. Projekti palkittiin ensin alueellisilla StarT-festareilla, minkä jälkeen se voitti myös kansallisen palkinnon. Projekti valittiin lisäksi kymmenen parhaan LuMa-projektin joukkoon saman vuoden kansainvälisessä StarT-sarjassa. Myöhemmin luokka sai kutsun esitellä pelejään myös Innokas-verkoston Jyväskylän yliopistolla järjestämässä peliohjelmoinnin kehittämistapaamisessa.



Kuvat 16 ja 17. Pelien esittelyä kouluväelle ja huoltajille Norssi-gaalassa



Kuvat 18 ja 19. Pelien esittelyä yliopistolla Ruusupiistossa





Kuvat 20 ja 21. StarT-projektin alueellisissa ja kansallisissa palkitsemistilaisuuksissa

## Ohjelmoinnin monet mahdollisuudet

Tässä artikkelissa havainnollistimme erilaisia tapoja toteuttaa ohjelmointia 1.–3. luokilla lukemaan ja kirjoittamaan oppimisessa, matemaattisten taitojen harjoittelussa, luonnontieteen sisältöalueilla, taito- ja taideaineissa sekä isompien monialaisten projektien toteuttamisessa. Tavoitteenamme oli korostaa, että ohjelmoinnin kautta opittavien taitojen ei tarvitse rajoittua yksinomaan ohjelmoinnin oppimiseen. Esitimme useita omaan opetukseen sovellettavia valmiiksi kokeiltuja ja testattuja tehtävävaihtoehtoja. Näiden tuella voi havaita, että vain kekseliäisyys on rajana, kuinka ohjelmointia voidaan soveltaa erilaisten taitojen oppimisessa esimerkiksi Blue-botin kaltaisilla alkeisroboteilla tai Scratch Junior -ohjelmointiympäristöllä. Ohjelmointitehtäviä teettäessämme huomasimme, että monipuoliset ohjelmointia yhdistävät tehtävät ovat oppilaita hyvin motivoivia ja tukena myös muiden tavoitteiden oppimiselle. Keskeistä ohjelmoinnissa on, että opiskellaan aluksi yhdessä ohjelmoinnin perusasioita, kuten sitä, mistä erilaiset toiminnot (esim. taustan lisääminen) löytyvät. Oppilaat voivat sen jälkeen kokeilemalla oivaltaa uutta ja omaksua syventäviä asioita, kuten erilaisia ohjelmointitekenteitä, oman kiinnostuksensa ja ohjelmointityönsä tarpeiden mukaisesti.

Alkuopetuksessa ohjelmoinnin perustaidoissa on kyse varsin perustavanlaatuisista ja yksinkertaisista asioista (ks. esim. eNorssi 2018). Siitä huolimatta osoitimme, että alkuopetuksessa luodun vahvan ohjelmoinnillisen perustan tuella oppilaat pystyvät ohjelmoimaan esimerkiksi omia pelejä, mitä on tyypillisesti pidetty huomattavasti vanhemmille oppilaille sopivana työtapana. Perustason ohjelmointitaidot omaavien oppilaiden kanssa voidaan soveltaa tässä artikkelissa esimerkinomaisesti nostetun oman pelin ohjelmoinnin kaltaisia projektimaisia työtapoja. Tällöin oppimisessa korostuu erityisesti se, että siirrytään alkuopetukselle ominaisesta perustaitojen harjoittelusta autenttisempiin oppimisen konteksteihin. Ohjelmoinnilliseen ajatteluun kuuluu myös tunnistamaan aitoja yhteiskunnan ohjelmoinnillisia ulottuvuuksia ja ratkaisemaan erilaisia oikeaan elämään liittyviä käytännön ongelmia ohjelmoinnin keinoin (Denning & Tedre 2019; Grover 2018). Koulussa vastaanvaiseen ohjelmoinnillisen ajattelun maailmaan tutustumiseen soveltuu mielekkäästi esimerkiksi animointi, pelinteko tai erilaisten omien keksintöjen suunnittelu ja valmistaminen. Kouluikäisille lapsille ja nuorille soveltuvia ohjelmointiympäristöjä on monenlaisia, kuten ohjelmointi ilman laitteita erilaisten leikkien kautta, Scratch (edistyneempi versio Scratch Juniorista), Legon rakennettavat robotit (esim. WeDo, SPIKE, EV3) sekä uudenlaiset ohjelmoitavat pienoistietokoneet, kuten Micro:bit (Innokas-verkosto 2019; Lee ym. 2011). Kuten tässä artikkelissa havainnollistettiin, keskeisinä pedagogisina vahvuuksina erityisesti projektimaisissa ohjelmoinnin työskentelytavoissa voidaan pitää ennen kaikkea yhteistoiminnallisuutta (esim. suunnittelu ja ohjelmointi pareittain tai ryhmissä), oppilaslähtöisyyttä (esim. oman työn suunnittelu, itseilmaisu) ja monialaisuutta (esim. erilaisten sisältöjen tai teemojen käsittely ohjelmoitavassa tuotoksessa) oppilaita motivoivassa ympäristössä.

Lopuksi haluamme korostaa, että on tärkeä oivaltaa, että vaikka ohjelmointi kuuluu perusopetuksen opetussuunnitelmassa laaja-alaisen osaamisen osalta vain alueeseen ”L5. Tieto- ja viestintäteknologinen osaaminen” (Opetushallitus 2014), ohjelmointi on itsessään varsin laaja-alaista tekemistä. Oman ohjelmoitavan tuotoksen suunnittelun ja varsinaisen ohjelmakoodin laatimisen sekä muiden tuotokseen kuuluvien elementtien valmistamisen yhdessä muiden kanssa voidaan nähdä pitävän sisällään erilaisia osaamisalueita ja sisältöjä, joita kuvataan myös perusopetuksen opetussuunnitelman laaja-alaisessa osaamisessa. Tavoitteenamme onkin jatkossa pohtia, kuinka ohjelmoinnissa näyttäytyvät esimerkiksi erilaiset ajattelun taidot, oppimaan oppiminen, vuorovaikutus, monilukutaito ja muut laaja-alaiset osaamisalueet.

## Lähteet

- Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining Twenty-First Century Skills. Teoksessa P. Griffin, B. McGaw, & E. Care (toim.) *Assessment and Teaching of 21st Century Skills*. Dordrecht: Springer, 17–66.
- Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. Cambridge, MA: The MIT Press.
- eNorssi (2018). *Suomen harjoittelukoulujen ohjelmointipolku*. Haettu osoitteesta: <https://peda.net/jyu/normaalikoulu/ops/suunnitelmat/tjvos/6/liitteet/o2>
- Perfusun-Patrick, K. (2007). Writers develop skills through collaboration: an action research approach. *Educational Action Research* 15 (2), 159–180.
- Grover, S. (2018). *STELAR Webinar: Unpacking Computational Thinking & CT's Role in Interdisciplinary Learning*. Haettu osoitteesta: <http://stellar.edc.org/events/stelar-webinar-unpacking-computational-thinking-cts-role-interdisciplinary-learning>
- Heintz, F., Mannila, L., & Färnqvist, T. (2016). A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education. *2016 IEEE Frontiers in Education Conference (FIE)*, 1–9. IEEE.
- Hidi, S., & Boscolo, P. (2006). Motivation and writing. Teoksessa C. A. MacArthur, S. Graham, & J. Fitzgerald (toim.) *Handbook of Writing Research*. New York: The Guildford Press, 144–157.
- Innokas-verkosto (2018). *Materiaalit - Innokas*. Haettu osoitteesta: <https://www.innokas.fi/materiaalit/>
- Jyväskylän normaalikoulu (2016). *Perusopetuksen opetussuunnitelma 2016*. Haettu osoitteesta: <https://peda.net/jyu/normaalikoulu/ops>
- Kafai, Y., Proctor, C., & Lui, D. (2019). From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. Teoksessa L. Malmi, A. Korhonen, R. McCartney, & A. Petersen (toim.) *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '18)*. New York, NY: ACM, 101–109.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads* 2 (1), 32–37.
- Opetushallitus (2014). *Perusopetuksen opetussuunnitelman perusteet 2014*. Helsinki: Opetushallitus.
- Papert, S. (1980). *Mindstorms: Children, Computers, And Powerful Ideas*. New York, NY: Basic Books Inc.
- Tedre, M., & Denning, P. (2016). The long quest for computational thinking. Teoksessa J. Sheard, & C. Suero Montero (toim.) *Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16)*. New York, NY: ACM, 120–129.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM* 49 (3), 33–35