

Ohjelmoinnin opetussuunnitelman laatimisen tueksi

Anna-Maija Partanen, matematiikan didaktiikan yliopistonlehtori, Lapin yliopisto

anna-maija.partanen@ulapland.fi

Tämä teksti on kirjoitettu pohdinnan ja keskustelun alustukseksi. Sitä pitää tarkastella kriittisesti, eikä se missään tapauksessa ole tarkoitettu sellaisenaan kopioitavaksi. Olen tulkinnut Perusopetuksen opetussuunnitelman perusteita 2014, tehnyt *oman sovellukseni opetussuunnitelmasta* ja esittänyt käytännön ideoita miten tätä opetussuunnitelmaa voi toteuttaa. Esitän myös kysymyksiä, joita mielestäni on tarpeellista käsitellä ohjelmoinnin opetussuunnitelmaa laadittaessa.

Teksti on melkein valmis alkuopetuksen osalta ja osittain valmis luokkien 3. – 6. osalta. Yläkoulun osuuden kirjoittaminen on pahiten kesken.

Perusopetuksen opetussuunnitelmien perusteiden 2014 mukaisesti ohjelmointi tulee uutena sisältönä ja opiskeltavana taitona alakouluun, ja sen opiskelu tulee kaikille pakolliseksi yläkoulussa. Koulut voivat sen lisäksi halutessaan tarjota valinnaiskursseja ja kerhoja, joissa ohjelmointia ja robotiikkaa harrastetaan syventäen osaamista. Meillä suomalaisessa alakoulussa ei ole ohjelmoinnin opetuksen traditiota, ja tästä syystä ohjelmoinnin opetussuunnitelman laatiminen voi olla vaikeaa.

Olen havainnut, että monilla eri tahoilla (henkilöillä, järjestöillä, verkostoilla jne.) on spesifiä kokemusta ohjelmoinnin opettamisesta pienille lapsille ja nuorille, mutta kokonaisnäkemystä aiheesta on ollut vaikea löytää. Monesti on lähdetty liikkeelle teknologia edellä ja kokeiltu mitä juuri tällä ohjelmalla tai robottisarjalla voisi tehdä. Tämän tekstin tarkoituksena on pohtia miten ohjelmoinnin opetuksen järjestämiseen voitaisiin löytää punainen lanka alkuopetuksesta yhdeksännelle luokalle. Pohdin muun muassa seuraavia kysymyksiä. Mitä Perusopetuksen opetussuunnitelman perusteissa sanotaan ja mitä sillä tarkoitetaan? Miten sen pohjalta voidaan suunnitella opetusta alkuopetuksessa, alakoulun luokilla 3. – 6. sekä yläkoulussa? Millaisia työskentelymahdollisuuksia on olemassa, eli miten ohjelmointia opetetaan? Miten nämä eri työmuodot ja ideat sopivat eri-ikäisille lapsille? Pitäisikö ohjelmoinnin opetuksessa käsitellä tietojenkäsittelyn peruskäsitteitä, vai olisiko se lähinnä toimintaa ja projektityöskentelyä jonka aikana itsestään opitaan ajattelun taitoja? Pyrin antamaan aineksia ohjelmoinnin opetussuunnitelman tekoa sekä ja ohjelmoinnin opetuksen suunnittelua ja kehittämistä varten. Kuten jo edellä mainitsin, ehdotuksiani ei ole tarkoitettu sellaisenaan toteutettavaksi, vaan pohdittavaksi ja edelleen kehitettäväksi.

1) Ohjelmointi Perusopetuksen opetussuunnitelman perusteissa 2014

Ohjelmointi mainitaan eksplisiittisesti laaja-alaisen osaamisen yhteydessä osioissa: Tieto- ja viestintäteknologinen osaaminen (L5) sekä matematiikan tavoitteissa ja sisällöissä ja myös käsityön sisällöissä.

Laaja-alaisen osaamisen taitojen kuvauksessa Tieto- ja viestintäteknologinen osaaminen sanotaan:

- ▶ ”Oppilaat **saavat ja jakavat keskenään** kokemuksia digitaalisen median parissa työskentelystä sekä **ikäkaudelle sopivasta ohjelmoinnista.**” (1.-2. lk)
- ▶ ”Ohjelmointia kokeillessaan oppilaat saavat kokemuksia siitä, **miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista.**” (3. – 6. lk)
- ▶ ”Ohjelmointia **harjoitellaan osana eri oppiaineiden opintoja.**” (7. – 9. lk)

Matematiikan osuudessa ohjelmointi on mainittu keskeisten sisältöalueiden kohdassa S1 Ajattelun taidot.

- ▶ Luokkien 1. – 2. osiossa todetaan: ”Tutustuminen ohjelmoinnin alkeisiin alkaa laatimalla **vaiheittaisia toimintaohjeita joita myös testataan.**”
- ▶ Luokkien 3. – 6. osalta sanotaan: ”**Suunnitellaan ja toteutetaan ohjelmia graafisessa ohjelmointiympäristössä.**”
- ▶ Luokilla 7 – 9: ”**Syvennetään algoritmista ajattelua.** Ohjelmoidaan ja samalla **harjoitellaan hyviä ohjelmointikäytäntöjä.** Sovelletaan itse tehtyjä tai valmiita tietokoneohjelmia osana matematiikan opiskelua.”

Matematiikan alkuopetuksen tavoitteeksi annetaan T12: harjaannuttaa oppilasta laatimaan vaiheittaisia toimintaohjeita ja toimimaan ohjeen mukaan, luokille 3. – 6. asetetaan tavoite T14: innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmia graafisessa ohjelmointiympäristössä ja yläkoululle annetaan tavoite T20: ohjata oppilasta kehittämään algoritmista ajatteluaan sekä taitojaan soveltaa matematiikkaa ja ohjelmointia ongelmien ratkaisemiseen. Hyvän osaamisen kuvauksessa luokille 3. – 6. sanotaan, että oppilas osaa ohjelmoida toimivan ohjelman graafisessa ohjelmointiympäristössä ja luokille 7. – 9., että oppilas osaa soveltaa algoritmisen ajattelun periaatteita ja osaa ohjelmoida yksinkertaisia ohjelmia.

Käsityön tavoitteisiin liittyvissä keskeisissä sisältöalueissa vuosiluokille 3. – 6. mainitaan S3:n kuvauksessa, että ”**Harjoitellaan ohjelmoimalla aikaan saatuja toimintoja, joista esimerkkinä robotiikka ja automaatio. Kokeilujen pohjalta tuotetta tai teosta kehitetään vielä eteenpäin.**” Luokkien 7. – 9. kohdalla mainitaan S3:n kuvauksessa, että ”**Käytetään sulautettuja järjestelmiä käsityöhön eli sovelletaan ohjelmointia suunnitelmiin ja valmistettaviin tuotteisiin.**”

Kati Sormunen Helsingin yliopistosta (Opettajankoulutuslaitosten seminaari Raumalla 17.9.2015) esitti mukaillen CS Unplugged -sivustoa (<http://csunplugged.org/>) sekä Mykästä ja Liukasta (2014, KOODI2016), että *ohjelmoinnilliseen ajatteluun* kuuluu mm.

1) ongelman purkaminen osiin, 2) kaavojen tunnistaminen, 3) algoritmien luominen, 4) ratkaisun yleistäminen ja automatisointi sekä 5) tietokonekieli. Nämä kaikki liittyvät hänen mukaansa luontevasti opetussuunnitelman perusteiden laaja-alaisen tavoitteiden alueisiin L1 Ajattelu ja oppimaan oppiminen sekä L5 Tieto- ja viestintäteknologinen osaaminen. Kaavojen tunnistaminen ja algoritmien luominen liittyvät alueeseen L3 Itsestä huolehtiminen ja arjen taidot, sekä kaavojen tunnistaminen ja tietokonekieli alueeseen L4 Monilukutaito.

2) Ohjelmoinnin opetussuunnitelman hahmottelua

Olen pohtinut opetussuunnitelman perusteiden tekstejä, tutustunut erilaisiin ohjelmoinnin opetuksen ideoihin, välineisiin ja ohjelmiin sekä kokeillut niitä pienimuotoisesti alakoulun oppilaiden kanssa. Tässä kappaleessa esitän omia tulkintojani ja sovelluksiani opetussuunnitelman perusteiden mukaisesta opetuksesta. Kolmannessa kappaleessa havainnollistan esimerkkien avulla niitä aktiviteetteja ja työtapoja, joista kussakin osiossa kirjoitan.

1. Alkuopetus

Kuvaus: Laaditaan ja testataan toimintaohjeita puhutun kielen ja symbolien avulla, myöhemmin kirjoitetun kielen avulla. Lapset itse laativat ohjeita, joita testataan, sekä opettelevat noudattamaan toisten antamia ohjeita. Voidaan opetella graafisen ohjelmoinnin alkeita.

Käytännön toteutus: Leikki, pelit, harjoitukset ja erilaiset aktiviteetit, pienet projektityöt

Käsitteitä: Algoritmi, ohjelma (ja ohjelman luonne), ohjelmointi

Algoritmi on ohjeet jonkun tehtävän suorittamiseksi. Ohjelma on *koneen ymmärtämällä kielellä* kirjoitetut ohjeet tehtävän suorittamiseksi. Ohjelma kirjoitetaan ensin kokonaan. Sen jälkeen se annetaan ”koneelle” suoritettavaksi. Kone toimii mekaanisesti ohjeiden mukaisesti eikä ota vastaan lisäohjeita suorituksen aikana, pitsi jos se on ohjelmoitu tekemään niin. Jos kone ei pysty suorittamaan jotakin käskyä, se sanoo ERROR.

Kehittyvät ajattelun taidot: Ongelmanratkaisu, suunnitelmallisuus, kokonaisuuden pilkkominen osiin ja hahmottaminen osista, täsmällinen ajattelu (tarkkojen ohjeiden antaminen ja ohjeiden kirjaimellinen seuraaminen), omien virheiden etsiminen ja korjaaminen

Eriyttäminen: Monista aktiviteeteista on muotoiltavissa ”helpot versiot” ja enemmän kompleksisuutta ja luovuutta vaativat/kehittävät versiot.

Sosiaaliset taidot: Parityöskentelyn oppiminen, eri roolien ottaminen ja roolien vaihtaminen

2. Luokat 3. – 6.

Kuvaus: Tutustutaan graafiseen ohjelmointiin ja siinä käytettyihin rakenteisiin. Tukena rinnalla voidaan tehdä sopivia ”unplugged” (ilman tietokonetta tehtäviä) harjoituksia. Edetään pikkuhiljaa pidemmälle, robottien ohjelmointiin tai pelien ohjelmointiin Internetissä. Tutustutaan graafisen ohjelmoinnin ympäristöön (esim. Scratch) ja toteutetaan projekti/projekteja joko sillä tai esim. robottien kanssa tai pelejä ohjelmoiden. Projektien aiheet valitaan oppilaiden kiinnostuksen mukaisesti liittäen ne esimerkiksi eri oppiaineisiin, aihekokonaisuuksiin tai oppilaiden harrastuksiin.

Käytännön toteutus: Opiskellaan graafisen ohjelmoinnin opetusohjelmien avulla (esim. Internetissä), tehdään ”unplugged” harjoituksia, rakennetaan ja ohjelmoidaan robotteja tai

pelejä (Internetissä tai ladattavilla ohjelmilla). Tutustutaan Scratchin tapaiseen graafisen ohjelmoinnin ympäristöön. Tehdään projekti/projekteja.

Käsitteitä: Tutustutaan graafisen ohjelmoinnin periaatteisiin. Tutustutaan ohjelmoinnin perusrakenteisiin kuten silmukka, funktio, if – then – else -rakenne, tietojenkäsittelyn peruskäsitteisiin kuten esimerkiksi bitti ja pikseli sekä loogisiin operaatioihin EI, JA ja TAI.

Kehittyvät ajattelun taidot: Luova ongelmanratkaisu kontekstissa, kollektiiviseen ideointiin osallistuminen, algoritmien ajattelu graafisen ohjelmoinnin ympäristössä, täsmällinen ja looginen ajattelu, ongelman pilkkominen osiin sekä kokonaisuuden hahmottaminen osista, säännönmukaisuuksien (jaksojen) havaitseminen, ratkaisun yleistäminen

Sosiaaliset taidot: Tiimissä tehtävässä projektityössä tarvittavien taitojen kehittäminen

Eriyttäminen: Graafisessa ohjelmoinnissa eri vaikeustason opetusohjelmia, robotiikkaa ja peliohjelmointia kiinnostuneille, projektitöissä eriyttäminen tulee itsestään, kun kukin valitsee itsellensä sopivan aiheen ja tekee sen omista lähtökohdistaan. Innokkaimpia ja taitavimpia oppilaita voidaan ohjata tekstuaalisen ohjelmointikielen pariin esim. robottien ohjelmoinnissa.

3. Yläkoulu

Kuvaus: Opetellaan ohjelmoimaan jollakin tekstuaalisella ohjelmointikielellä. Hyödynnetään sekä symbolisella kielellä että graafisessa ympäristössä tehtyjä ohjelmia projekteissa ja eri aineiden opiskelussa. Kiinnitetään huomiota hyviin ohjelmointikäytänteisiin. Voidaanko rakentaa kytkös graafisen ja symbolisen ohjelmoinnin välille?

Käytännön toteutus: Tutustutaan johonkin tekstuaaliseen ohjelmointikieleen ja tehdään sillä harjoituksia. Tehdään projekteja ohjelmoiden tekstuaalisella kielellä tai graafisessa ympäristössä. Robottien ja innovatiivisten prototyyppien rakentaminen, pelien ohjelmointi, nettisivujen suunnittelu ja taiteet tarjoavat aiheita projekteihin jotka voivat liittyä aihekokonaisuuksiin tai oppilaiden harrastuksiin ja kiinnostuksen kohteisiin. Matematiikassa??

Käsitteitä: Hyvät ohjelmointikäytännöt; opitaan hyödyntämään ohjelmointiympäristön automaattisia toimintoja ja valmiita rakenteita. Tehdään selkeitä, mahdollisimman yksinkertaisia ohjelmia ja hyödynnetään tehokkaasti ohjelmoinnin perusrakenteita. Sitkeästi kehitetään ohjelmista toimivia ja parannetaan niiden ominaisuuksia.??

Kehittyvät ajattelun taidot: Algoritmien ajattelu syvenee (Mitä tarkoittaa?): ongelman purkaminen osiin ja kokonaisuuden hahmottaminen osistaan, säännönmukaisuuksien tunnistaminen ja huomioiminen ohjelman kirjoittamisessa, algoritmien ja ohjelmien luominen, ratkaisun yleistäminen ja automatisointi sekä tietokonekielen luonteen ymmärtäminen??

Sosiaaliset taidot: Tiimissä tehtävässä projektityössä tarvittavien taitojen kehittäminen

Eriyttäminen: Syventävät valinnaiset kurssit ohjelmoinnista, edistyneet ohjelmoijat ekspertin roolissa, kun opiskellaan koko luokan kanssa, projekteissa kukin toimii omista lähtökohdistaan käsin

Heränneitä ajatuksia ja kysymyksiä

Projektityöskentely sopii ohjelmoinnin opiskeluun. Kun oppilaat saavat itse keksiä aiheen, työskentely on motivoivaa, innostavaa ja luovaa. Siitä saa myös voimaa projektin loppuun viemiseen. Projektien aiheissa kannattaa muistaa myös taiteelliset ja liikunnalliset projektit sekä teknologisten innovaatioiden suunnittelu.

Robottiikan parissa lapset näkevät itse konkreettisesti, miten käskyjen antaminen robotille merkitsee jotain fyysisessä todellisuudessamme. Peliin ohjelmoinnissa voi saada saman kokemuksen, kun joku muu pelaa suunnittelemaasi peliä. Näillä on siis lisäarvo verrattuna pelkästään tietokoneella tapahtuvaan ohjelmointiin.

Ohjelmoinnin tehtävissä ja projektitöissä on luontevia tilanteita harjoitella parityötä, ryhmätyötä ja koko luokan kanssa työskentelyä. Näissä tarvittavia taitoja voidaan harjoitella tietoisesti. Katso esimerkiksi video opetusohjelman alussa:

<https://studio.code.org/s/course1/stage/3/puzzle/1>

Miten huomioidaan tyttöjen motivointi ohjelmoinnin pariin sekä sukupuolten välinen tasa-arvo? (tanssi robotin kanssa, Frozen-opetusohjelma, itse valitut projektit, yhteistoiminnallisuus, asennemuokkaus ja roolimallit) Katso esimerkiksi video

<https://studio.code.org/s/frozen/stage/1/puzzle/1> .

Alkuopetuksessa ohjelmoinnin opetus voi tapahtua suureksi osaksi ilman tietokonetta, luokilla 3. – 6. graafisen ohjelmoinnin lisäksi voidaan tehdä tukevia ”unplugged” harjoituksia. Miten tehtäisiin liukuva siirtyminen graafisesta ohjelmoinnista symboliseen ohjelmointikieleen?

Vaikka ohjelmointi on mainittu erityisesti matematiikan sisällöissä, sitä ei kannata sitoa matematiikkaan liian tiukasti. Miten ohjelmointia tuodaan muiden aineiden opiskeluun, erityisesti yläkoulussa? Kenen asia on sitä pohtia?

3) Esimerkkejä työtavoista ja käytännön toteutuksia

Alkuopetus

Harjoitellaan täsmällisten ohjeiden antamista ensin puhutun kielen ja symbolien avulla

Esim. 1 Anna kaverille ohjeet, miten hän piirtää täsmälleen samanlaisen kuvion kuin sinä olet piirtänyt näkemättä sinun piirrostaasi.

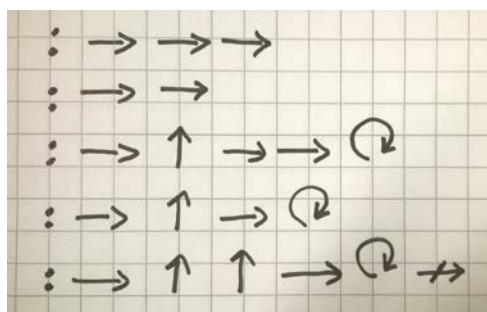
Esim. 2. Anna ihmisrobotille tarkat ohjeet, miten se palikka palikalta rakentaa samanlaisen kappaleen kuin sinulla näkemättä alkuperäistä. Sinä et saa nähdä robotin työtä. Robotti ei kysy eikä vastaa, vaan tottelee. Jos se ei pysty suorittamaan jotain käskyä, se sanoo ERROR.

Esim. 3. Anna ohjeet toiselle lapselle, miten hänen pitää liikkua tai tehdä jokin kokonaisuus.

Esim. 4. Keksikää itse leikki/koreografia johonkin lauluun ja opettakaa se muille.

Esim. 5. Useasta kuvasta koostuu ohje siemenen istuttamiseksi. Opettaja antaa sekoitetut kuvat lapsille, joiden tehtävä on laittaa ne mielekkääseen järjestykseen (ja vaikkapa liimata paperille) ja sitten suorittaa istuttaminen.

Esim. 6. Taitellaan paperi kuvien mukaisesti kuten origameja tehdessä. Laaditaan itse kuvamuotoiset ohjeet lennokin taittelemiseksi.



Esim. 7. Opettaja tai oppilas leikkii robottia. Sille annetaan nuolikieliset ohjeet kuinka liikutella kertakäyttömukeja yksi kerrallaan. Ohjeiden tuloksena pitäisi syntyä kertakäyttömukeista kuvan mukainen rakennelma.

Esim. 8. Ruutulattialle tehdään "vaarallinen polku". "Sokean" lapsen tulee kulkea polun läpi. Hänelle annetaan liikkumisohjeet yhdessä sovituilla merkeillä selkään piirtäen.

Esim. 9. RoboGem lautapelissä omia robotteja liikutellaan tekemällä pieniä ohjelmia nostetuista nuolikorteista. Peli tarjoaa hauskoja ja jännittäviä pelihetkiä ja samalla opetellaan ohjelmoinnin alkeita. Sanoma Pro, <https://sanomapro.fi/>

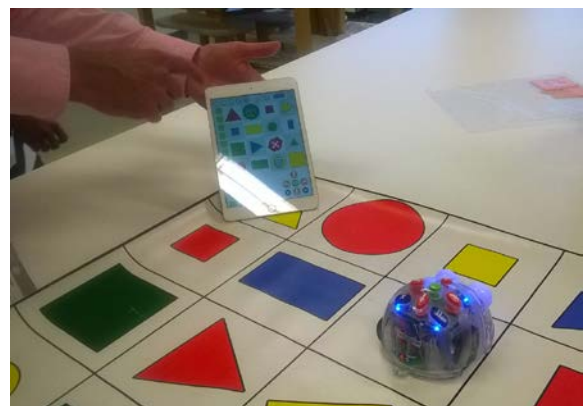
Laaditaan kirjallisia toimintaohjeita

Esim. 10. Kirjoita ohjeet vierailijalle, kuinka hän osaa kulkea koulu ulko-ovelta teidän luokkaan, tai vanhemmille, kuinka he osaavat kulkea luokkanne oven luota sinun pulpetillesi.

Esim. 11. Leikkikää aarteensintää. Kukin ryhmä piilottaa aarteen toiselle ryhmälle ja kirjoittaa sanalliset ohjeet aarteen löytämiseksi.

Esim. 12. Kirjoittakaa suosikki-smoothien tai -kuppikakun valmistusohje ja vaihtakaa ohje kaverin kanssa.

Robottien ohjelmointia



Bee-Bot on pienille lapsille tarkoitettu yksinkertainen robotti. Bee-Bottia voi ensin opetella ohjelmoimaan ilman alustaa. Sen selässä on merkit: nuoli eteenpäin, nuoli taaksepäin, kääntyminen 90° oikealle, kääntyminen 90° vasemmalle ja tauko. Liikkumisohjeet annetaan painelemalla näppäimiä järjestyksessä ja painamalla lopuksi GO-näppäintä. Vaikeustasoa voidaan lisätä ohjelmoimalla kaksi tai useampi robotti liikkumaan yhdessä suunnitelman mukaisesti, vaikkapa symmetrisesti niin, että radat muodostavat toistensa peilikuvat. Blue-Bot-robottia voidaan ohjelmoida lisäksi tablettiin asennetulla sovelluksella graafisen ohjelmoinnin tapaan.

Bee-Botteja voi käyttää erilaisilla alustoilla

Alustan tyhjiin taskuihin voi sijoittaa esimerkiksi geometrisia kuvioita ja niiden nimiä, vieraan kielen sanoja ja niiden suomenkielisiä vastineita, laskulausekkeita jne. Taskuihin sijoitettujen kuvien avulla voidaan rakentaa kaupunki tai vaikkapa "Bee-Bot maailma".

Lapset voivat piirtää radan Bee-Botille ja keksiä rataa liittyvän tarinan. Radalla voi liikkua useampikin robotti tarinan mukaan. Edellisissä esimerkeissä työskentelystä Bee-Bottien kanssa on jo projektityöskentelyn piirteitä.

Graafista ohjelmointia

Pienille lapsille tarkoitettuja graafisen ohjelmoinnin alkeisiin opastavia opetusohjelmia löytyy esimerkiksi osoitteesta: <https://studio.code.org/> valitsemalla Kurssi 1.

Scratch Jr on pienille lapsille soveltuva graafisen ohjelmoinnin ympäristö. Siihen voi tutustua osoitteessa <http://www.scratchjr.org/index.html>

Muuta

Parityöskentelyn periaatteita esittelee lyhyt video, joka löytyy osoitteesta: <https://studio.code.org/s/course1/stage/3/puzzle/1>

Luokkien 3. – 6. ilman tietokonetta tehtävistä harjoitteista löytyy myös alkuopetukseen sopivia tehtäviä, esimerkiksi säännönmukaisuuksien hahmottaminen jonossa sekä loogisten operaatioiden EI, JA ja TAI harjoitteet.

Luokat 3. – 6.

Graafiseen ohjelmointiin tutustuminen on helppo aloittaa valmiiden opetusohjelmien avulla. Katso esimerkiksi Koodaustunti tai Kurssit 2, 3 ja 4 osoitteessa <https://studio.code.org/> Hieman vaativampi ja matematiikan osaamista vaativa opetusohjelma on Frozen samassa osoitteessa. Internetistä löytyy paljon muita samaan tarkoitukseen sopivia ohjelmia. Myös pelien ohjelmointiin on mahdollista päästä heti kiinni sivustojen tarjoamien opetusohjelmien avulla.

Scratch-ohjelmointiympäristöä voidaan käyttää Internetissä tai se voidaan ladata omalle koneelle <https://scratch.mit.edu/>

Robottien ohjelmoinnista

Yleisiä robotteja löytyy Legon sarjasta Lego Mindstorms (RCX, 1998, NXT, 2006, NXT 2.0, 2009 ja EV3, 2013). Helpoin tapa ohjelmoida niitä perustuu graafiseen ohjelmointiin. Kullekin sarjalle on kehitetty myös ”software development kit”, jolla ohjelmointiominaisuuksia voidaan kehittää. Näitä ohjelmia pyörittää laitteiston ohjelmointiyksikön sisäinen kiinteä ohjelmisto. Myös tämä kiinteä ohjelmisto on muokattavissa esim. Java-, C- tai Forth-kieltä ymmärtäväksi versioksi. Esimerkiksi NXT-versiota voidaan ohjelmoida NXC-kielellä ja NXT- ja EV3-versioita RobotC-kielellä.

VEX Robotics IQ on alakoulun yläluokille ja yläkouluun sopiva robottisarja. Myös niitä voidaan ohjelmoida ensin raahaamalla ja yhdistelemällä valmiita palkkeja (EasyC-kieli) sekä myöhemmin esimerkiksi RobotC-kielellä. <http://www.vexrobotics.com/>

Pro-Bot on Bee-Botin kehittyneempi versio, sen isoveljeksikin mainostettu. Se on auton muotoinen ja sitä voidaan ohjelmoida sekä näppäilemällä tai rakentamalla ohjelma tietokoneen ruudulla. Näppäimistö koostuu nuolista ja numeroista. Samalla, kun ohjeet auton liikkumiseksi kirjoitetaan näppäimistöltä, ohjelma tulee näkyviin auton katolla olevaan näyttöruutuun. Ohjelmaa voidaan editoida näyttöruudussa poistamatta sitä kokonaan. Pro-Bottiin voidaan kiinnittää kynä, joka piirtää auton radan sen liikkuessa. Sen etu- ja takapuskurissa on kosketussensorit ja siihen tulee myös ääni- ja valosensorit. Näin Pro-Bot

muodostaa välivaiheen Bee-Botista vaativampaan robottien ohjelmointiin.

<https://www.bee-bot.us/probot.html>

Pelien ohjelmointia

Sivustolla studio.code.org voi myös aloittaa tekemään pelejä opetusohjelmien Pelilabra ja Infinity pelilabra avulla.

Scratch:illa voi tehdä pelejä ja siihen on saatavissa myös sensorilevyjä.

<https://scratch.mit.edu/>

Kodu Game Lab on ympäristö, jossa jopa alakouluikäiset voivat harjoitella peliohjelmointia <http://www.kodugamelab.com/> Sen voi ladata omalle koneelle Internetistä.

MinecraftEdu on kolmiulotteinen peli/virtuaalimaailma, jossa oppilaat voivat seikkailla kollektiivisesti. Pelillisyyttä hyödyntäen sillä myös opetetaan erilaisia asioita.

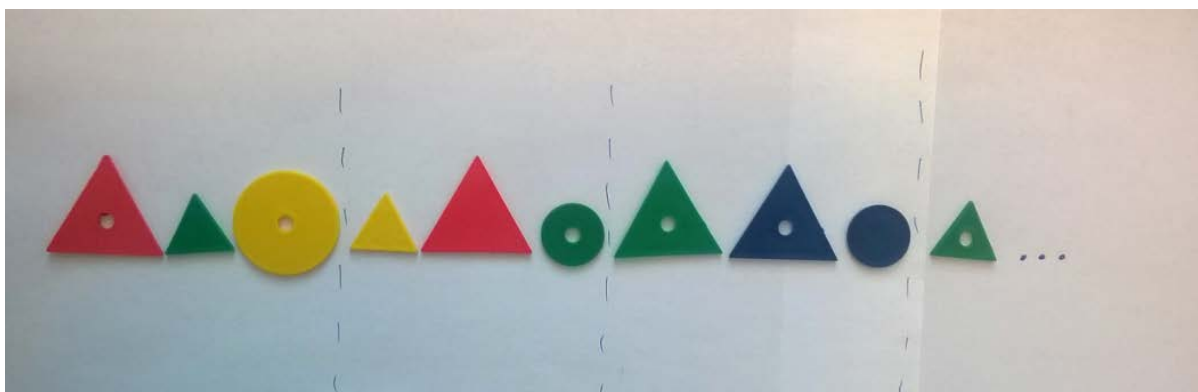
ComputercraftEdu (<http://computercraftededu.com/>) tuo ohjelmoitavan kilpikonnarobotin tavallisiin Minecraft-ympäristöihin. Kilpikonnat voivat esimerkiksi lentää, kaivaa ja rakentaa. Kilpikonnia ohjelmoidaan ensin näppäimistön avulla vetämällä ja pudottamalla kuvia ruudukkoon. Ohjelmoinnissa voidaan edetä lopulta varsinaisen symbolisen koodin käyttöön.

Graafisen ohjelman rakenteiden ymmärtämistä tukevia “unplugged” harjoituksia

Esim. 1. Ehtolauseen mukaan toimimista ja ehtolauseiden rakentamista voidaan harjoitella pelikorttien kanssa. Kaksi joukkuetta nostaa vuorotellen kortteja sekoitetusta pakasta. He saavat pisteitä annetun säännön mukaisesti, esimerkiksi ”Jos kortti on punainen, niin joukkue saa pisteitä sen tuplan verran, muutoin vastapuoli saa pisteitä kortin luvun verran.” Oppilaat voivat itse keksiä sääntöjä, jotka noudattavat rakennetta **jos – niin – muutoin** ja pelata niiden mukaan. Säännöissä voi myös olla päällekkäisiä jos – niin – muutoin rakenteita. Tästä esimerkki voisi olla vaikkapa ”Jos kortti on punainen, niin saatte pisteitä sen tuplan verran, muutoin jos se on musta ja kuvakortti, vastapuoli saa kortin luvun verran pisteitä, muutoin vastapuoli saa sen tuplan verran pisteitä.”

Esim. 2. Tanssien tai laululeikkien koreografia voidaan kertoa ja näyttää toisille tai se voidaan ilmaista kuvien avulla. Näihin on helppo istuttaa ohjelmoinnin perusrakenteita, kuten esimerkiksi silmukka tai funktio. Kun kerrotaan, että tee tämä sarja vaikkapa neljä kertaa, kyseessä on silmukka. Kun johonkin kokonaisuuteen viitataan lyhyesti yhdellä nimellä, kyseessä on funktio.

Esim. 3. Jaksojen hahmottaminen jonossa on taito, jota voi harjoitella jo pienten lasten kanssa. Se on yksi ns. esimatemaattinen taito. Voidaan muodostaa vaikka jono lapsista niin, että siinä kaksi seisoo ja yksi on kyykyssä, kaksi seisoo ja yksi on kyykyssä, kaksi seisoo jne. Siinä tapauksessa jakso on kaksi seisovaa ja yksi kyykyssä oleva lapsi. Loogisilla paloilla voidaan muodostaa jaksoja hyvin monella eri tavalla. Lapset voivat muodostaa jakson loogisten palojen muodostamaan jonoon ja kavereiden tehtävänä on hahmottaa jakso tutkimalla jonoa.



Kun jonosta etsitään säännönmukaisuuksia tai jaksoja, se vastaa säännönmukaisuuksien etsimistä maailmasta tai omasta koodista. Jotta koodi olisi eleganttia ja tehokasta, säännönmukaisuuksien (toistojen) yhteydessä kannattaa käyttää silmukkaa (tai funktiota). Säännönmukaisuuksien etsiminen jonosta on mainittu myös luokkien 3 – 6. matematiikan osuudessa sisältöalueessa S3.

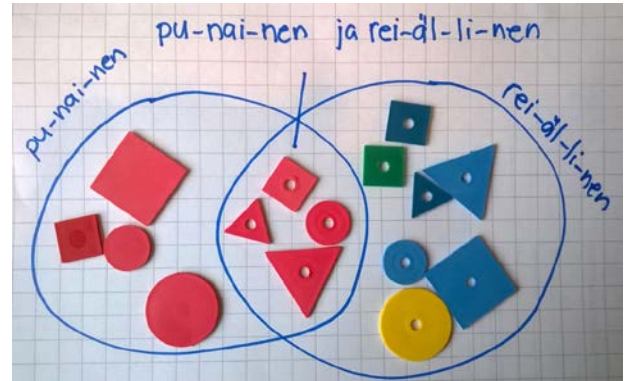
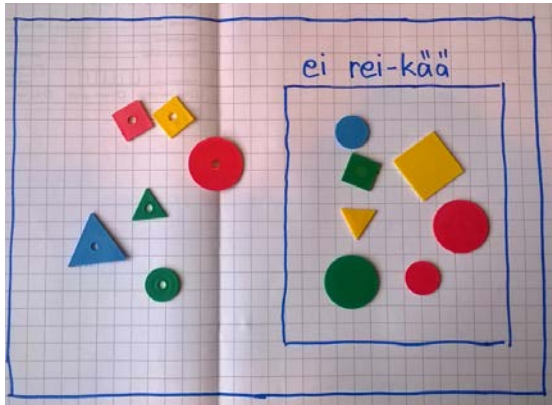
Esim. 4. RoboGem lautapelissä on F-kortti. Sille voidaan määritellä sisältö, joka koostuu useammasta käskystä. Kun useamman käskyn yhdistelmä suorittaa jonkun kokonaisuuden, sitä sanotaan funktioksi.

Esim. 5. Ohjeiden mukaan piirtämisessä voidaan määritellä jotkin käskyjen yhdistelmät funktioksi. Esimerkiksi funktio "portaat" voisi koostua käskyistä: Piirrä viiva ylös, viiva oikealle, viiva ylös, viiva oikealle, viiva ylös, viiva oikealle, viiva ylös ja viiva oikealle.

Esim. 6. Kun luokka on opiskellut koordinaatiston, voidaan harjoitella piirtämistä ohjeiden mukaan siinä. Esim. käsky "Aseta kynä pisteeseen (5, 6), piirrä kolmen ruudun mittainen vaakasuora viiva oikealle, piirrä kolmen ruudun mittainen viiva alas, piirrä kolmen ruudun mittainen vaakasuora viiva vasemmalle ja piirrä kolmen ruudun mittainen viiva ylös." voisi olla funktio "neliö(5,6,3)". Lapset voisivat kehittää itse koordinaatistossa piirtämistä varten tehokkaan kielen, joka sisältää funktioita ja silmukoita.

Esim. 7. Tietokone käsittelee informaatiota bittien muodossa. Bitillä voi olla arvo nolla tai yksi. Ne vastaavat käytännössä sitä, että virta kulkee tai ei kulje, jännite on korkealla tai matalalla, tietty paikka on magnetoitunut tai ei ole jne. Symbolisesti bittejä voidaan esittää luvuilla 1 ja 0. Kun kahdeksan bittiä yhdistetään, saadaan järjestelmä, jonka avulla voidaan esittää 256 merkkiä. Näiden joukossa ovat myös kirjaimet. (Esimerkiksi isoa kirjainta A vastaa 0100 0001 ja kirjainta M vastaa 0100 1101.) Oppilaille voidaan antaa "tietokoneen kielellä" kirjoitettu viesti, joka heidän pitää ratkaista. Oppilaat voivat kirjoittaa oman nimensä bittijärjestelmän avulla. Kirjaimet ja niitä vastaavat kymmenjärjestelmän luvut löytyvät osoitteesta <http://www.mit.jyu.fi/opetus/Ciao/ciao135.htm>. Luvut pitää vielä muuttaa kaksilukujärjestelmään ja esittää kahdeksan bitin avulla. Esimerkiksi kirjainta A (iso a-kirjain) vastaa kymmenjärjestelmän luku 65, joka voidaan esittää muodossa $64 + 1 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 100\ 0001$. Kahdeksan bittisessä muodossa tähän laitetaan vielä nolla eteen 0100 0001. Oppilaille annetaan vastaavuudet kirjainten ja bittiesitysten välillä valmiina.

Esim. 8. Voidaan kehittää symbolikieli, jonka avulla voidaan antaa ohjeet ruudun pikku ruutujen (pikseleiden) värittämiseksi mustaksi tai jättämiseksi valkoiseksi. Kun kuvaa katsotaan kaukaa, se näyttää esimerkiksi a-kirjaimelta. Näin tietokoneen ja television ruudun mustavalkoiset kuvat muodostuvat pikseleistä.



Esim. 9. Tärkeitä loogisia operaatioita ovat EI, JA ja TAI. Näitä voidaan harjoitella loogisten palojen avulla.

Yläkoulu

Tekstuaalisten "oikeiden" ohjelmointikielten opettamiseen on kehitetty oppilasystävällisiä lähestymistapoja. Järjestö Teaching Kids Programming on kehittänyt Java-kielen opetusta Eclipse-ympäristössä (Lwellyn Falco ja Maaret Pyhäjärvi, ohjelmoinnin opetuksen työpaja Oulun yliopistossa 25.8.2015). Heidän menetelmässään hyödynnetään kolmivaiheista lähestymistapaa: ensin tehtävä kuvataan suomen kielellä, sen jälkeen se käännetään englanniksi ja siitä ohjelmointikielelle. On tärkeää oppia hyödyntämään ohjelmien automaattisia valikoita, joista käskyjen rakenteet saa poimittua.

<http://teachingkidsprogramming.org/>

Racket-kielille on myös kehitetty opetusta tukeva ympäristö DrRacket. Racket kieltä voi opiskella Koodiaapisen kurssilla (www.koodiaapinen.fi) tai Innokas-hankkeen tapahtumakiertueella Kiinnostaako koodaus? www.innokas.fi

Turun yliopiston tietojenkäsittelylaitoksen tutkijat ovat Mikko-Jussi Laakson johdolla kehittäneet sähköisen oppimisympäristön, Ville-järjestelmän. LUMA-Suomi kehittämisohjelman hankkeena he ovat yhdessä yläkoulun opettajien kanssa kehittämässä tähän ympäristöön ohjelmoinnin opetuksen mallia, jonka avulla ohjelmointia voidaan opettaa monipuolisesti missä tahansa yläkoulussa. Kurssi (tai malli) sisältää tehtäviä, joiden avulla koodausta voidaan harjoitella Java-, Python-, C++ ja C#-ohjelmointikielillä. Ohjelmointikurssi Ville-järjestelmässä tulee yleiseen käyttöön mahdollisesti keväällä 2016. Sen käyttöönotto edellyttää opettajalta mooc-verkkokurssina toteutetun Ville-koulutuskurssi suorittamista. Tällä kurssilla opettajat perehdytetään ympäristön käyttöön.

Arduino on helppokäyttöinen alusta/ympäristö innovatiivisten prototyyppien rakentamiselle. Arduino-levyt lukevat inputteja – sensoriin tuleva valo, napin painallus, tai Twitter viesti – ja muuttavat ne outputeiksi – moottorin käynnistäminen tai LED-valon sytyttäminen, esimerkiksi. Arduino-ohjelmointikiellä ja Arduino-ohjelmistolla voidaan ohjata

levyjen toimintaa. Opettajat ja opiskelijat voivat rakentaa halpoja välineitä kemian ja fysiikan ilmiöiden tutkimiseen tai ohjelmoinnin ja robotiikan opiskeluun. Suunnittelijat ja arkkitehdit rakentavat Arduinolla interaktiivisia prototyyppkejä. Muusikot ja taiteilijat tekevät sillä installaatioita ja kokeilevaa musiikkia. Arduino-yhteisö jakaa jo olemassa olevia projektejaan, joita voi soveltaa omiin tarkoituksiinsa. <https://www.arduino.cc/>

Legon Mindstorms -sarjan ja VEX Robotics IQ –sarjan robottien ohjelmointimahdollisuuksia on mahdollista laajentaa. Kullekin Legon sarjalle on kehitetty ”software development kit”. Lisäohjelmia pyörittää laitteiston ohjelmointiyksikön sisäinen kiinteä ohjelmisto. Myös tämä kiinteä ohjelmisto on muokattavissa esim. Java-, C- tai Forth-kieltä ymmärtäväksi versioksi. Esimerkiksi NXT-versiota voidaan ohjelmoida NXC-kielellä ja NXT- ja EV3-versioita RobotC-kielellä. VEX Robotics IQ –sarjan robotteja voidaan ohjelmoida myöhemmin esimerkiksi RobotC-kielellä. <http://www.vexrobotics.com/>

Jos alakoulussa on tutustuttu graafisen ohjelmoinnin ympäristöön kuten Scratch, niin sitä voidaan käyttää yläkoulussa erilaisissa yhteyksissä, esimerkiksi kielten opiskelussa. (Eri hahmot voidaan ohjelmoida keskustelemaan keskenään puhekuilien tai äänitallenteiden avulla.) Myös ”unplugged” harjoituksia voidaan hyödyntää eri aineiden opiskelussa. Oppilaat voidaan pyytää laatimaan tuotepakkauksissa tyypillisesti esiintyviä ohjeita jonkun tehtävän suorittamiseksi vieraalla kielellä, esim. kastikkeen valmistaminen, siemen istuttaminen, teen hauduttaminen, kirjahyllyn kokoaminen jne. Ohjeet testataan. Tai vaikka prepositioiden opiskelun jälkeen voidaan harjoitella laatimalla piirrosohjeita muille.

Pelisuunnittelua eri-ikäisille on kehitetty LUMA-keskus Saimaan Pelisuunnittelu (Peliohjelmointi) henkkeessa. Hankkeen yläkoulujen ryhmissä tutustutaan peliohjelmoinnin alkeisiin Unity-työkalun sekä ohjelmoinnin Codecademy-verkkokurssin avulla. <http://luma.fi/artikkelit/3944/pelisuunnittelu-tuottaa-luovia-ideoita-seka-luovia-keksijoita>

Yläkoulussa ohjelmoinnin opetus voisi mielekkäästi tapahtua myös eri oppiaineita yhdistävissä projekteissa. Esimerkiksi robotteja ja automaatioteknologiaa hyödyntävissä projekteissa voisivat fysiikan, matematiikan ja käsityön opettajat ohjata oppilaita. Peliohjelmoinnin projektit olisivat luontevia yhteyksiä historian, kuvataiteiden ja musiikin opettajille tehdä yhteistyötä tietotekniikan opettajien kanssa. Nettisivuja voitaisiin ohjelmoida tietojenkäsittelyn opettajien sekä äidinkielen ja vieraiden kielten opettajien johdolla. Musiikin opettajat ja liikunnan opettajat voisivat olla mukana erilaisten installaatioiden ja tanssi/performanssiesitysten kehittämisessä.

4) Lopuksi

Esittämäni ajatukset eivät ole kehittyneet eristyksissä. Kiitos kaikille henkilöille, joiden kanssa olen voinut keskustella ja muodostaa näkemyksiäni! Heidän joukossaan on tietojenkäsittelyn ammattilaisia, Innokas-hankkeen opettajia Lapista ja Helsingistä, Lapin yliopiston harjoittelukoulun opettajia, Lapin ammattikorkeakoulun opettajia, eri yliopistojen opettajankouluttajia, Lapin yliopiston luokanopettajaopiskelijoita ja ihan tavallisia opettajia joiden kanssa olemme tutustuneet ohjelmoinnin opetukseen vetäessäni lyhyitä työpajoja Lapissa.