Miten opettaa ohjelmointia vuosiluokilla 7-9?

Perusopetuksen vuosiluokilla 7-9 ohjelmointia tulisi opettaa osana kaikkien oppiaineiden opetusta, vaikka selkeät maininnat oppiainekohtaisista tavoitteista löytyykin vain matematiikan ja käsityön osalta. Perusopetuksen yläluokilla syvennetään edelleen ohjelmoinnillista ajattelua ja siirrytään visuaalisista ohjelmointiympäristöistä varsinaiseen koodaamiseen.

Opetussuunnitelman perusteissa ei määritellä lainkaan käytettäviä ohjelmointikieliä tai sisältöjä yläkoulun osalta. Niissä määritellään vain osaamistavoitteeksi se, että jokainen yläkoulun päättävä osaa ohjelmoida toimivan tietokoneohjelman; mille ei tosin määritellä sen tarkempia toiminnallisuusvaatimuksia.

Ohjelmoinnin opetuksen voi yläkoulussakin aloittaa visuaalisella Scratch-ympäristöllä, jossa haastetta riittää myös yläkoululaisille. Scratch on hyvä lähtökohta myös silloin, kun ohjelmoinnin opiskelua aloitetaan vasta yläkoulussa (katso myös Ohjelmointia aloittelijoille). Scratch luo hyvän pohja muille luonnollisille ohjelmointikielille kuten Pascal, C tai Java. Tietotekniikan laitoksella kehitetty Jypeli-ohjelmointikirjasto on toteutettu C#-ohjelmointikielellä, joten jatkumo Scratchistä onnistuu hyvin. Myös suositut nuorten peliohjelmointikurssit 12-19 -vuotiaille hyödyntävät Jypeli-kirjastoa.

Maailmalla sekä meilläkin suosittu Racket-ohjelmointiympäristö on tekstuaalinen ohjelmointiympäristö ja sen kielen syntaksi on aivan erilainen luonnollisen kielen mukaisiin ohjelmointikieliin. Näin ollen jatkumo eri kouluasteille on haastavampi luoda Racketin avulla.

Miten aloittaa ohjelmointi yläkoulussa?

Perusopetuksen uusien opetussuunnitelman perusteiden astuessa voimaan moni yläkoululainen joutuu opettelemaan ohjelmointia ilman alakoulun tuomia ohjelmoinnillisen ajattelun valmiuksia. Näin ollen, heitä ei kannata ohjata suoraan koodaamisen pariin, vaan kannattaa aloittaa samoista alkeista kuin alakoululaisetkin aloittavat.

Joidenkin mielestä visuaalinen ohjelmointi yläkoulussa voi tuntua lapselliselta, mutta esimerkiksi Koodaustunti-tehtävät voi oikeasti tehdä tunnissa läpi, joten ne toimivat hyvin ensikosketuksena ohjelmoinnilliseen ajatteluun. Myös Robogem-lautapeli sopii yläkoululaisille. Näiden jälkeen voidaan tutustua visuaaliseen ohjelmointiin Scratch-ympäristössä esimerkiksi animaatioprojektin kautta. Myös Kodu Game Lab on tähän hyvä vaihteoehto.

Yläkoululaiset selviytyvät näistä em. vaiheista varmasti alakoululaisia nopeammin, jolloin päästään jo muutamassa viikossa varsinaisen koodaamisen pariin.

Ohjelmointi vuosiluokkien 7-9 opetussuunnitelmassa

Laaja-alainen osaaminen vuosiluokilla 7-9
Tieto- ja viestintäteknologinen osaaminen L5

"Ohjelmointia harjoitellaan osana eri oppiaineiden opintoja." (POPS 2014, 284)

Matematiikan tavoitteet vuosiluokilla 7–9
S1 Ajattelun taidot ja menetelmät

"Syvennetään algoritmista ajattelua. Ohjelmoidaan ja samalla harjoitellaan hyviä ohjelmointikäytäntöjä. Sovelletaan itse tehtyjä tai valmiita tietokoneohjelmia osana matematiikan opiskelua. (POPS 2014, 375)

Käsityön tavoitteet vuosiluokilla 7-9
S3 Kokeilu

"Käytetään sulautettuja järjestelmiä käsityöhön eli sovelletaanohjelmointia suunnitelmiin ja valmistettaviin tuotteisiin." (POPS 2014, 431)

Scratch sopii myös yläkouluun

Vuosiluokkien 3-6 osuudessa esiteltiin tarkemmin visuaalinen ohjelmointityökalu Scratch, joka soveltuu toki myös yläkoulun ohjelmoinnin opetukseen. Ohjelmointitehtävien haasteellisuus on lähinnä opettajan mielikuvituksesta kiinni, joten aihepiirejä ja ideoita kannattaakin etsiä muiden toteutuksista. Hyvänä esimerkkinä mainittakoon MIT:n luovat ohjelmointitehtävät (https://scratch.mit.edu/hoc/).

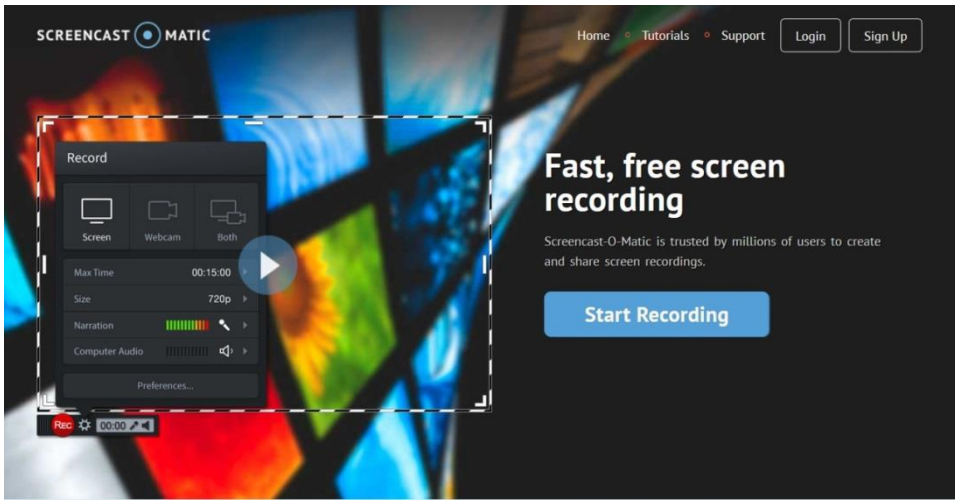
Seuraava esimerkki on tehty edellä mainitun MIT:n tanssiohjeistuksen mukaan.

Videolla tanssiva balleriina on toteutettu Scratch-koodilla seuraavasti:

Scratch-ohjelmointia voi sujuvasti laajentaa esim. kuvataiteen, musiikin, kielten ja äidinkielen opetuksen. Scratch-ympäristöön voi tehdä omia taustoja piirtämällä, kuvankäsittelyssä tai valokuvaamalla. Myös hahmoja voi piirtää itse. Taustamusiikit voi tuottaa musiikin tunnilla. Eri kielivaihtoehtoja voi hyödyntää kieltenopetuksessa sekä maahanmuuttajien äidinkielen opetuksessa. Lopputuloksesta voi kuvaruutukaappauksella tehdä videon, johon voi äidinkielen tunneilla suunnitella taustatarinan. Sisältöjä voi toki helposti tuoda myös muista oppiaineista ja ohjelmoinnin rinnalle voikin tuoda myös esimerkiksi animaatioiden tuottamisen.

Lisäosia

Oheinen kuvaruutukaappausvideo Scratchistä on toteutettu Screencast-o-Matic'illä, jonka ilmaisella versiolla voi tehdä maksimissaan 15 minuutin videoita.



Jypeli

Jyväskylän yliopiston tietotekniikan laitoksella on kehitetty ohjelmointityökalu, jonka avulla kännykkä- ja tietokonepelien kehittäminen on entistä helpompaa. Jypeli-ohjelmointikirjastoa on kehitetty vuodesta 2009 alkaen ja sen avulla on tuotettu jo yli 700 peliä. Ohjelmointikirjaston avulla aloituskynnys madaltuu ja ensimmäisten pelien tekeminen helpottuu. Jypeli-ohjelmointityökalun avulla tehdään 2D-pelejä, esim. matopelejä, tankkipelejä, autopelejä, lentopelejä, biljardi-pelejä, minigolf-pelejä, urheilupelejä, tasohyppelypelejä tai seikkailupelejä.

Jypeli-ohjelmointikirjasto tarvitsee toimiakseen Microsoftin Visual Studion. Tarkemmat ohjeistukset käyttöönottoon ja ohjelmointiin löytyy nuorten peliohjelmointi -kurssien kotisivuilta os. https://trac.cc.jyu.fi/projects/npo.

https://youtu.be/ruF8WgQMALU
Upotuskoodi:
<iframe width="640" height="390" src="https://www.youtube.com/embed/ruF8WgQMALU" frameborder="0" allowfullscreen></iframe>

Tekstuaalista ohjelmointia Racketillä

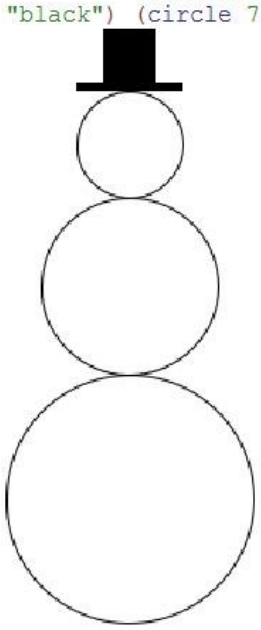
Meillä ja maailmalla on saanut suurta suosiota tekstuaalisen ohjelmoinnin ympäristö Racket. Ko. ympäristössä ohjelmoinnin opiskelu onnistuu varmasti ihan hyvin, mutta tekstuaalisuutensa puolesta ympäristö edellyttää paljon ulkoaoppimista. Racket-ympäristössä on paljon valmiita koodikirjastoja, joiden funktioiden ja komentojen muistaminen on perusedellytys koodaamiselle. Toki ne voi myös tarkistaa aina ohjeistuksesta, mutta tällöin koodaaminen on melko hidasta.

Seuraavassa on muutama esimerkkinäkymä DrRacket-sovelluksesta. Ensimmäisenä yksinkertaisia laskutoimituksia sekä merkkijonoja ...

```
Welcome to DrRacket, version 6.2.1 [3m].
Language: Beginning Student; memory limit: 128 MB.
> 175
175
> "Hello World!"
"Hello World!"
> (+ 175 35)
210
> "+ on tässä funktio"
"+ on tässä funktio"
> "funtioiden ja muuttujien välissä pitää aina olla välilyönti;
ylimääräiset välilyönnit ei haittaa"
"funtioiden ja muuttujien välissä pitää aina olla välilyönti;
ylimääräiset välilyönnit ei haittaa"
> (* 175 35)
6125
> (- 175 35)
140
> (+ 175 35 45 20)
275
> "muuttujia voi olla enemmänkin"
"muuttujia voi olla enemmänkin"
> (* (- 175 35) (+ 45 20))
9100
> (* 3 (- 4 5))
-3
>
```

... toisena esimerkkinä piirtofunktioiden testausta ja ...

```
> (above (square 30 "solid" "black") (rectangle 60 5 "solid"
"black")
         (circle 30 "outline" "black") (circle 50 "outline"
"black") (circle 70 "outline" "black"))
```



```
>
```

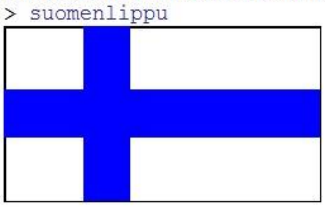
... kolmantena esimerkkinä omien funktioiden määrittely erilaisten piirtofuntioiden yhdistämiseksi.

```
(require 2htdp/image)

(define kaksi 2)
(define tausta (rectangle 200 110 "outline" "black"))
(define vaaka (rectangle 200 30 "solid" "blue"))
(define lippu1 (overlay vaaka tausta))
(define pysty (rectangle 30 110 "solid" "blue"))


 (define suomenlippu (overlay/xy pysty -50  0 lippu1))
```

```
Welcome to DrRacket, version 6.2.1 [3m].
Language: Beginning Student; memory limit: 128 MB.
> suomenlippu
```



```
>
```

Racket-ohjelmoinnin tueksi löytyy laajat ohjeet suomenkielisestä Racket-koodiaapisesta. http://racket.koodiaapinen.fi/). Racket-koodausympäristönä voi käyttää joko selainpohjaista editoria (http://www.wescheme.org/openEditor) tai koneelle ladattavaa DrRacket-sovellusta (http://download.racket-lang.org/).

Lähde:
https://peda.net/jyu/it/koulutusteknologia/op/ov7