

Oppiainerajat ylittävän peliprojektin suunnittelu omaan kouluun - Vinkkipankki

LUMA SUOMI kehittämishanke: ”Oppiaineet ja opettajat integroivaa tietotekniikan ja ohjelmoinnin opetusta” 2017-2019

Materiaalin tekijä:

Hanna Hankaniemi
LUMA-kouluttaja 2019,
LUMA-keskus Pohjanmaa

Alkuperäiset opetuskokonaisuuden kehittäjät 2017:

Timo Ahola ja Olli Raatikainen, Koodiopisto (www.koodiopisto.fi)
Jaakko Yli-Ojanperä, Fysiikan lehtori, Vaasan yliopisto
Anna-Sofia Kantola, Pro Gradu, Tampereen yliopisto



Vinkkipankki - Ohjelmointi 1.-2. luokka



Kuva 1.

Ohjelmointi 1.-2. luokilla:

- Opetellaan antamaan yksikäsitteisiä komentoja ihmiseltä toiselle. Ohjelmointi on käskyjen antamista tietokoneelle, joten tämä pohjustaa sen ymmärtämistä.
- Opitaan sitä, että täsmälliset ohjeet tuottavat täsmällistä toimintaa, ja epämääräiset ohjeet tuottavat epämääräistä toimintaa.
- Toimintaohjeita voidaan harjoitella antamaan leikkien ja pelien avulla, tietokoneella tai ilman sitä. (Kuva 1.)

Tieto- ja viestintäteknologinen osaaminen

- Oppilaat saavat ja jakavat keskenään kokemuksia digitaalisen median parissa työskentelystä sekä ikäkaudelle sopivasta ohjelmoinnista (OPS 2016, 101).

Matematiikan tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 1–2

- Tutustuminen ohjelmoinnin alkeisiin alkaa laatimalla vaiheittaisia toimintaohjeita, joita myös testataan (OPS 2016, 129).

OPS

Vinkkejä ja linkkejä:

- Ohjelmoinnillista ajattelua kehittävät leikit:
<https://drive.google.com/file/d/1NXLvpzjq1neHtmg7owfmGUtpojUgjWpg/view>
- Code.org-sivusto: <https://studio.code.org/courses>
- Scratch Junior- sovellus (Ipad/Android): <https://www.youtube.com/watch?v=aj6YWOF-2cM>
- Robottiikka:
 - Esim. Bee-Bot-robotti – ohjelmoidaan nuolikomennoilla. Esittelyvideo: https://www.youtube.com/watch?time_continue=4&v=Y6hhSNXXUOA
 - Sphero SPRK+ - pallorobotti – ohjelmoidaan piirtämällä. Esimerkki: https://www.youtube.com/watch?time_continue=77&v=ldjdVF97uVo



Kuva 2. Bee-Bot



Kuva 3. Code.org

Vinkkipankki - Ohjelmointi 3.-6. luokka

Ohjelmointi 3.-6. luokilla:

- Aletaan perehtyä tekemiseen, joka on lähempänä ohjelmointia. Työkaluna ei vielä ole varsinainen ohjelmointikieli vaan jokin visuaalinen ohjelmointiympäristö, jossa työskennellään käytännössä hiiren avulla, ei välttämättä kirjoittamalla.
- Ohjelmointia voidaan harjoitella raahaamalla tai siirtelemällä asioita, luomalla asioista ja visuaalisista komennoista sarjoja sekä tutustumalla esimerkiksi komennoilla muodostettaviin toistorakenteisiin.

Tieto- ja viestintäteknologinen osaaminen (L5)

- Ohjelmointia kokeillessaan oppilaat saavat kokemuksia siitä, miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista (OPS 2016, 157).

Matematiikan tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 3–6

- Suunnitellaan ja toteutetaan ohjelmia graafisessa ohjelmointiympäristössä (OPS 2016, 235).

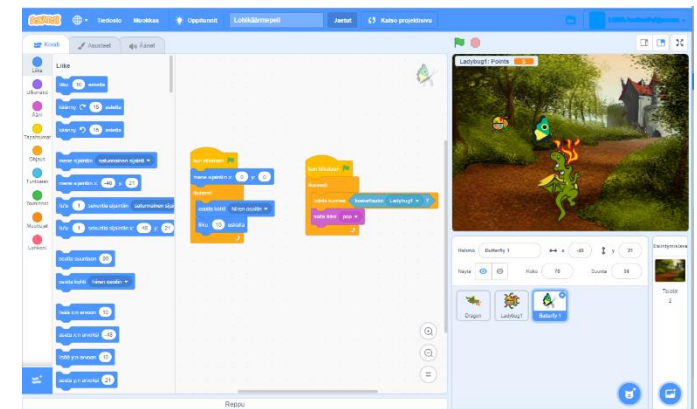
Käsityön tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 3–6

- Harjoitellaan ohjelmoimalla aikaan saatuja toimintoja, joista esimerkkinä robotiikka ja automaatio (OPS 2016, 271).

OPS

Vinkkejä ja linkkejä:

- Code.org-sivusto: <https://studio.code.org/courses>
- Scratch-ohjelmointiympäristö: <https://scratch.mit.edu/> Videotutoriaalit: https://www.youtube.com/playlist?list=PLEPhdRNnAKPvso9B5wAuMZ9N_egbrkyVs
- Robotiikka ja automaatio:
 - Roboteilla ja muilla ohjelmoitavilla välineillä on tavallisesti omat ohjelmointikielensä, mutta ne ovat periaatteeltaan samanlaisia.
 - Lego Mindstorms: <https://www.lego.com/fi-fi/mindstorms>
 - ROBBO: <https://peda.net/jyu/it/koulutusteknologia/op/keos-2017/robbo>
 - Sphero SPRK+: <https://www.sphero.com/education/>
 - Pro-Bot: <https://www.youtube.com/watch?v=bGSf9vA7Dlo>
 - 3D-tulostus: http://www.edu.fi/tvt_opetuksessa/3d_tulostaminen



Kuva 4. Scratch 3.0

Vinkkipankki - Ohjelmointi yläkoulu

Ohjelmointi 7.-9. luokilla:

- Aloitetaan perehtyminen johonkin oikeaan ohjelmointikieleen.
- Tarkoituksena on, että ymmärretään perusasioita kielestä ja osataan tulkita ohjelmakoodia – ”tuossa kohdin ohjelma ottaa jonkin luvun sisään, tuossa se tekee sille jotain ja niin edelleen.”
- Hyviä ohjelmointikäytäntöjä ja algoritmisen ajattelun periaatteita voi aluksi harjoitella graafisessa ohjelmointiympäristössä ilman oikeiden ohjelmointikielten syntaksi opettelua.

Tieto- ja viestintäteknologinen osaaminen (L5)

- Ohjelmointia harjoitellaan osana eri oppiaineiden opintoja (OPS 2016, 284).

Matematiikan tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 7–9

- Ohjelmoidaan ja samalla harjoitellaan hyviä ohjelmointikäytäntöjä (OPS 2016, 375).

- **Arviointi:** Oppilas osaa soveltaa algoritmisen ajattelun periaatteita ja osaa ohjelmoida yksinkertaisia ohjelmia (OPS 2016, 379).

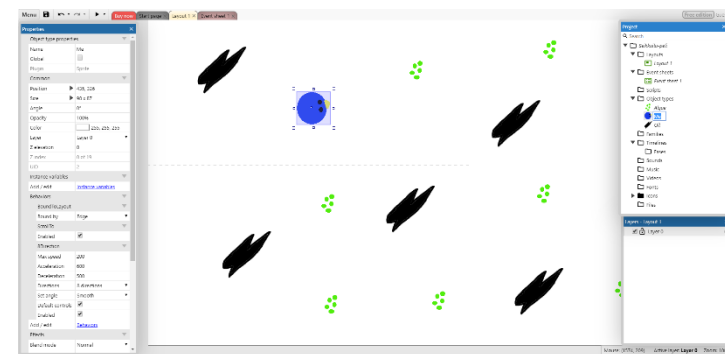
Käsityön tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 7–9

- Käytetään sulautettuja järjestelmiä käsityöhön eli sovelletaan ohjelmointia suunnitelmiin ja valmistettaviin tuotteisiin (OPS 2016, 431).

OPS

Vinkejä ja linkkejä:

- Scratch-ohjelmointiympäristö: <https://scratch.mit.edu/> Videotutoriaalit: https://www.youtube.com/playlist?list=PLEPhdRNnAKPvso9B5wAuMZ9N_egbrkyVs
- Construct 3 –pelinteko-ohjelma: <https://editor.construct.net/>
- Robottiikka ja automaatio:
 - Roboteilla ja muilla ohjelmoitavilla välineillä on tavallisesti omat ohjelmointikielensä, mutta ne ovat periaatteeltaan samanlaisia.
 - Lego Mindstorms: <https://www.lego.com/fi-fi/mindstorms>
 - ROBBO: <https://peda.net/jyu/it/koulutusteknologia/op/keos-2017/robbo>
 - Sphero SPRK+: <https://www.sphero.com/education/>
 - 3D-tulostus: http://www.edu.fi/tvt_opetuksessa/3d_tulostaminen



Kuva 5. Construct 3

Vinkkipankki - Ohjelmointi yläkoulu

Oikeiden ohjelmointikielten harjoittelu on huomattavasti helpompaa ja innostavampaa, kun ohjelmoinnin periaatteet ovat tutut ja algoritmisen ajattelun taidot ovat kehittyneet. Yläkoululaiselle sopivia ohjelmointiprojekteja ovat yksinkertaiset tekstipohjaiset pelit ja ohjelmat. Opetukseen sopivia ohjelmointikieliä on satoja, mutta periaatteiltaan ne ovat samanlaisia

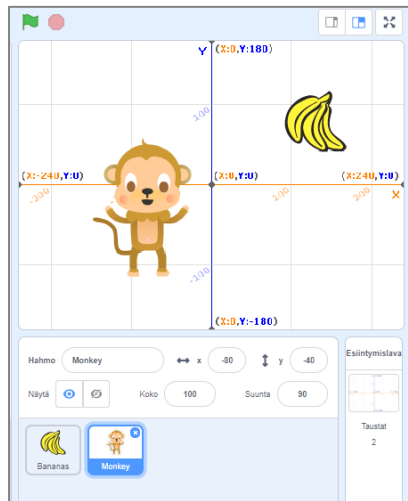
Yksi suosituista kielistä ohjelmoinnin aloittamiseen on Python: <https://linkki.github.io/python2017/>

```
1 #Ohjelma, joka laskee käyttäjän keskinopeuden.
2 print('Nyt lasketaan mikä oli sinun keskinopeutesi.')
3 matka = input('Kuinka pitkä matka oli kilometreissä?')
4 matkan_pituus = float(matka)
5 aika = input('Kuinka monta tuntia siihen meni?')
6 kaytetty_aika = float(aika)
7 print('Keskinopeutesi oli ' + str(matkan_pituus/kaytetty_aika) + ' km/h')
8 input('Paina ENTER poistuaaksesi')
9
```

Kuva 6. Esimerkki yksinkertaisesta Python-ohjelmasta.

Vinkkipankki – Pelit opetuksessa ja eri oppiaineissa

Pelit mahdollistavat monipuolisen ja luovan tavan yhdistää eri oppiaineet saman projektin ympärille. Peliprojektin kesto voi vaihdella yhdestä päivästä koko lukuvuoden mittaiseen projektiin. Peli voi sisällöstään riippuen yhdistää mitä tahansa eri oppiaineita. Tässä on esitelty ideoita mitä pelin tekemisen taitoja ja osa-alueita voidaan liittää eri oppiaineiden tunneille.



Kuva 7. Scratch - koordinaatisto

Behaviors	
Platform	
Max speed	330
Acceleration	1500
Deceleration	1500
Jump strength	650
Gravity	1500
Max fall speed	1000
Double-jump	<input type="checkbox"/>
Jump sustain	0
Default controls	<input checked="" type="checkbox"/>
Enabled	<input checked="" type="checkbox"/>

Kuva 8. Construct 3

Matematiikka ja fysiikka

Pelihakmojen sijainnin ja liikeratojen määrittäminen pelikentällä vaatii matemaattista osaamista, esim.

- Koordinaatisto ja asteet (Kuva xx)
- Hahmot sijoitetaan pelikentälle x- ja y-koordinaatein ja niiden liikeratoja määritellään muun muassa asteilla

Pelihakmojen liikettä määrittävät tekijät vaativat fysiikan ilmiöiden ymmärrystä

- Painovoima, kitka, nopeus jne.
- Pelihakmolle voidaan määritellä erilaisia fysiikka-ominaisuuksia kuten tiheys, elastisuus jne. (Kuva xx)

Äidinkieli, reaaliaineet ja muut kielet

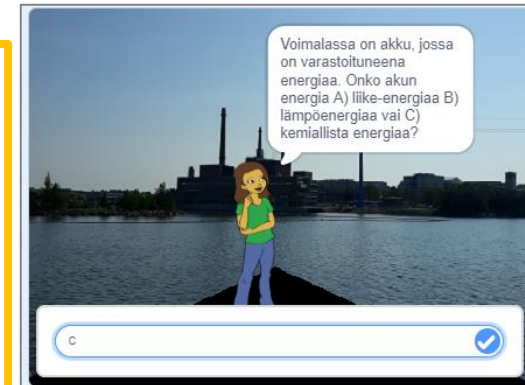
Pelin taustatarina ja käsikirjoitus voidaan kirjoittaa äidinkielen tunnilla

- Taustatarina on hyvä inspiraation lähde pelin suunnittelussa
- Käsikirjoitus on tärkeä osa pelin suunnittelua

Tarinalliset pelit, tietovisat, taitopelit yms. voivat liittyä mihin tahansa reaaliaineeseen

- Napoleon ja 1800-luku, ihmisen biologia, kemialliset reaktiot... jne.

Pelin dialogit, ohjeistus, kysymykset yms. voidaan toteuttaa eri kielillä.



Kuva 9. Scratch - Energiapeli



Kuva 10. Scratch - Tunnistuspeli

Vinkkipankki – Pelit opetuksessa ja eri oppiaineissa

Kuvaamataito

Pelin taustan ja hahmot voi

- Piirtää käsin ja/tai valokuvata

Sumopaint on ilmainen selaimessa toimiva kuvankäsittelyohjelma (Kuva 10.)

<https://www.sumopaint.com/home/>

- Piirtää tietokoneella

Paint.Net, Scratch tms.

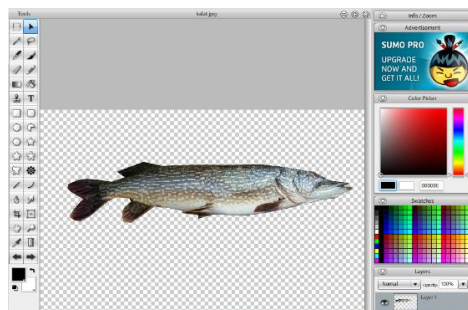
Piskel on selaimessa toimiva sovellus pikseli- hahmojen piirtämiseen (Kuva 11.)

<http://www.piskelapp.com/>

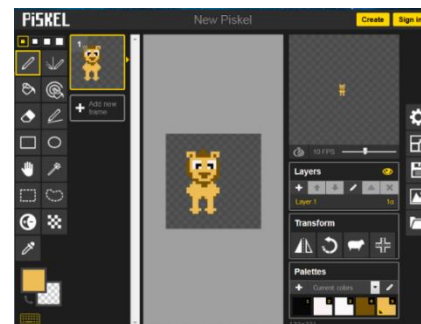
- Käyttää valmiita hahmoja, esim.

Scartchin omasta kirjastosta tai

OpenGameArt.org- sivustolta <https://opengameart.org/>



Kuva 11. Sumopaint



Kuva 12. Piskelapp

Musiikki

Pelin äänitehosteet ja taustamusiikin voi

- Äänittää ja tuottaa itse

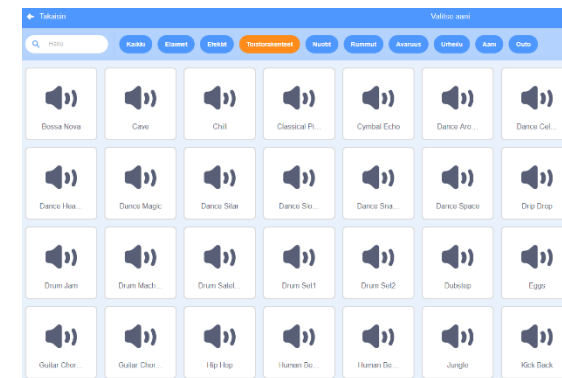
Audacity on ilmainen äänenkäsittelyohjelmisto (Windows/Mac/Linux)

<http://www.audacityteam.org/>

- Käyttää valmiita ääniefektejä esim. Scratchin omasta kirjastosta tai

Sounddogs.com-sivustolta

<https://www.sounddogs.com/>



Kuva 13. Scratchin ääni-kirjasto

Vinkkipankki – Peliprojektin vaiheet

Peliprojektin toteutus voidaan hahmottaa seuraavien vaiheiden avulla:



Ohjelmointiympäristön hallinta

- Ohjelmointiympäristön käyttöä tulee harjoitella ennen peliprojektin aloittamista. Tähän käytetään aikaa oppilaiden taitotason ja opettajan arvion mukaan. Harjoittelussa voi esimerkiksi käydä läpi ohjelmointialustan alkeet ja koodata erityylisiä, eri pelilogiikan omaavia esimerkkipelejä.
- Jos ohjelmointiohjelman käyttö ei ole ennestään tuttua, aikaa voi käyttää esim. n. 3 x 90 min (Scratch ja Construct 3)

Pelin alustaminen

Peliprojektin alustamista määrittää projektin pedagogiset tavoitteet ja sekä sen sisällön että ohjelmoinnin osalta. Alustamisessa tulisi miettiä

- Millainen tehtävänanto peliprojektilla on?
- Mitkä ovat pelin tekemisen reunaehdot? Mitä asioita siltä vaaditaan?

Vinkkipankki – Peliprojektin vaiheet



Pelin suunnittelu

” Hyvin suunniteltu on puoliksi tehty!” Ennen pelin koodaamista kannattaa hioa pelin ”juoni” kuntoon.

- *Inspiraatio:* Ennen suunnittelua oppilaiden mielikuvitus tulee saada liikkeelle. Tämä voi tapahtua esim. pelidemojen tai vaihtoehtoisesti peli-ideoiden avulla.
- *Taustatarina:* Taustatarina on omiaan vauhdittamaan ja inspiroimaan pelin suunnittelua.
- *Miellekartta:* Pulppuavia ideoita on hyvä koota paperille. Kaikkien ideoiden keskeltä kootaan sitten ne parhaat (ja realistiset) ideat, joihin projektissa keskitytään.
- *Hahmottelua piirtämällä:* Pelin taustoja, hahmoja ja etenemistä voi hahmotella myös piirtämällä esim. sarjakuvan avulla.
- *Käsikirjoitus:* Pelin käsikirjoittaminen toimii hyvänä lisäharjoituksena ja edistää pelin kehittämistä.

Vinkkipankki – Peliprojektin vaiheet



Pelin toteutus

- *Sisällön hallinta:* Peli voi sisällöltään liittyä useampaan eri aiheeseen ja näin ollen viedä paljonkin aikaa. Pelissä käsiteltäviin aiheisiin on mahdollista pureutua sekä pelin suunnittelu- että sen toteuttamisvaiheessa.
- *Pelimekaniikan luominen ja ohjelmoiminen:* pelit voivat olla mekaniikaltaan todella yksinkertaisia tai toisaalta erittäin monimutkaisia. Pelin logiikan ohjausta tulee miettiä oppilaiden taitotason ja käytettävän ajan suhteen.
- *Pelin ulkoasu:* Pelin ulkoasun muokkaamiseen voi käyttää valmiita hahmoja tai vaihtoehtoisesti ne voi luoda itse esimerkiksi tietotekniikan tai kuvaamataidon tunnilla.
- *Pelin äänimaailma:* Pelin äänimaailmaa voi luoda joko valmiilla ääniefekteillä tai niiden luomisesta voi tehdä oman oppisisältönsä esimerkiksi tietotekniikan tai musiikin tunnille.

Pelin esittely

- Pelin esittely on tärkeä ja motivoiva osa peliprojektia. Siitä voi tehdä oman ohjelmanumeronsa esimerkiksi luokan tai koko koulun yhteiseen tapahtumaan.