



Kodarit - Koodikerho

Koodikerho -opetusmateriaali

Ohjeet aloittelijoiden seitsemän ensimmäisen
kerhokerran ohjaamiseen

(kerrat 1-7)

V0.1

04.10.2017



Tervetuloa opettamaan koodausta!

Tämän seitsemän opetuskerran (kestoltaan yhteensä 7-10 tuntia) koodausopetusmateriaalin aloittelijoille on tuottanut sinulle [Kodarit Oy](#) Koodikerho.fi verkoston toimeksiantamana. Koodikerhoa koordinoi Suomen avoimien tietojärjestelmien keskus - COSS ry.

[Koodikerho](#) on kerran viikossa järjestettävä koululaisten iltapäiväkerho, jossa lapset oppivat ohjelmoinnin perusteet. Koodikerho on tarkoitettu yli kolmasluokkalaisille, eli 9+ -vuotiaille lapsille. Kerhot järjestetään paikallisten koulujen tiloissa ja tuella. Kerhoa vetää ohjelmoinnista kiinnostunut opettaja tai muu aikuinen, esimerkiksi IT-alalla työskentelevä vanhempi. Koodikerho.fi kerää tukivaroja lapsille ilmaisten koodikerhojen järjestämiseen sekä oppimateriaalien tuottamiseen.

Kodarit on lasten, nuorten ja aikuisten matalankynnyksen koodausopetukseen keskittynyt yritys. Kodareiden opetus perustuu 15 tunnin tasokursseihin. Kodarit ovat kehittäneet ja testanneet matalan kynnyksen koodausopetuspedagogiikkaansa vuodesta 2015. Tämä materiaali sisältää Kodareiden taso 1 kurssin seitsemän ensimmäistä tuntia. Lisätietoa Kodareiden opetuksesta [Kodarit.fi](#).

Tämä koodausopetusmateriaali on julkaistu avoimesti Koodikerhon verkkosivustolla CC BY-SA 4.0 -lisenssillä. Lisenssi sallii sinun kopioida, välittää, levittää ja esittää materiaalia sekä sen pohjalta tehtyjä muokattuja versioita vain, jos alkuperäiset tekijä mainitaan. Lisenssi antaa muiden julkistaa teoksesta muokattuja teoksia vain samalla lisenssillä, jolla alkuperäinen teos on julkaistu. Materiaaleja käytettäessä on mainittava molemmat Kodarit ja COSS ry/Koodikerho.

Lisenssi mainitaan lauseilla "Tekijänoikeuden haltijat Koodikerho/Suomen avoimien tietojärjestelmien keskus – COSS ry ja Kodari Oy 2017. Lisensoitu Creative Commons Nimeä-JaaSamoin 4.0 Kansainvälinen -lisenssillä." tai lyhyemmin "Lisensoitu CC BY-SA 4.0 -lisenssillä."

Creative Commons -lisenssi



Tämä teos, jonka tekijä on Kodarit ja Koodikerho, on lisensoitu [Creative Commons Nimeä-JaaSamoin 4.0 Kansainvälinen](#) -lisenssillä.



Sisällys

Tuntirakenne ja yleisiä ohjeita opettajalle	4
Oppitunti 1: Koodinimi ja karkkirobotti	5
Oppitunti 2: Code.org: Ohjelmoinnin perusteet	8
Oppitunti 3: Code.org: Frozen ja Taiteilija	11
Oppitunti 4: Johdatus pelien tekemiseen Code.org: Flappy code, Bounce ja Star Wars	13
Oppitunti 5: Scratch: Kirjautuminen ja tarina	15
Oppitunti 6: Scratch: Selviytymispeli	20
Oppitunti 7: Oma peli Scratchilla	23



Tuntirakenne ja yleisiä ohjeita opettajalle

Kurssi koostuu seitsemästä aivan aloittelijalle tarkoitettusta koodaustunnista ja sen rakenne noudattaa alla olevaa ohjelmaa. Ryhmät ovat kuitenkin erilaisia, joten kannattaa soveltaa sen mukaan. Kurssin harjoitukset tehdään lapsille ja nuorille suunnitelluissa hauskoissa visuaalisissa code.org sekä [Scratch](https://scratch.mit.edu) -oppimisympäristöissä.

Kurssin ohjelma:

Tunti 1: Koodinimi ja karkkirobotti

Tunti 2: Code.org: Ohjelmoinnin perusteet

Tunti 3: Code.org: Parametrit ja funktiot

Tunti 4: Code.org: Eventit/johdatus pelien tekemiseen

Tunti 5: Scratch: Kirjautuminen ja tarina

Tunti 6: Scratch: Selviytymispeli

Tunti 7: Scratch: Oma peli

Yleisiä ohjeita:

- Pyydä oppilaita laittamaan läppärit kiinni tai ainakin irrottamaan katseen näytöstä silloin kun opettaja näyttää jotain taululta
- Apua saa nostamalla käden ylös
- Apua kannattaa kysyä ensin vieruskaverilta
- Kaikkien ei ole pakko tehdä kaikkea loppuun asti. Joskus opettaja voi jatkaa seuraavaan aiheeseen, vaikka viimeisillä on tekeminen vielä kesken. Opettaja voi myös kehottaa jatkamaan loppuun kotona, tai koittaa ehtiä neuvomaan myöhemmin/seuraavalla tunnilla
- Kun valmiita pelejä ja tuotoksia esitetään muulle ryhmälle, opettajan kannattaa korostaa ettei ole pakko esittää, jos ei halua



Oppitunti 1: Koodinimi ja karkkirobotti

Tavoite

- Tutustua muihin kerholaisiin ja opetella nimet
- Tutustua käsitteisiin **koodiavain, enkoodaus ja dekoodaus**
- Oppia koodaamisen periaate: tietokone on tyhmä ja noudattaa orjallisesti koodia, jonka pitää olla täsmälleen oikein
- Tietokoneettomassa harjoituksessa tutustutaan **liikkumiskomentoihin ja toistolausekkeeseen**

Valmistelut

- Ensimmäisellä kokoontumiskerralla ei tarvita tietokoneita
- Käytä oheista materiaalia
- Kaikilla osallistujilla tulee olla paperilappu, johon voi kirjoittaa nimen, ja kynä
- Kirjoita A4 paperille suurella: askel eteen, askel taakse, käänny vasemmalle, käänny oikealle, ojenna karkkipussi
- Hanki pussillinen makeisia, mieluiten yksittäispakattuja, sulamattomia ja tahmaamattomia.
- Teippaa lattialle maalarinteipillä ruudukko 6x6 ruutua. Ruutujen pitää olla vähintään 0,5 m x 0,5 m. Lattian pitää olla vapaa kalusteista ja ruudukon tulee rajautua kehän reunalla oleviin oppilaiden tuoleihin siten, että jokainen oppilas voi istua (tai seistä) omassa reunaruudussaan. Laattalattian valmiita ruutuja voi myös käyttää, niin ei tarvitse teipata.

Tutustumisleikki: Koodinimi

Anna jokaiselle paperilappu ja heijasta seinälle ohjeet koodinimen luomiseen koodiavaimen avulla (Koodinimi-ohje).

Kerro kurssilaisille, että koodiavainta käyttämällä he voivat salata eli enkoodata oma nimensä. *Enkoodausta käytetään esimerkiksi viestien salaamiseen. Näin ollen, henkilö tai muu vastaanottaja, jolla on sama koodiavain käytössään, pystyy ymmärtämään eli purkamaan enkoodatun viestin sisällön. Viestin purkaminen tehdään koodiavaimen avulla ja sitä kutsutaan dekoodaamiseksi.* Osallistujien ollessa valmiita, he ovat kirjoittaneet



enkoodatun (=salatun) nimensä lapulle. **(Vinkki: Tässä kannattaa käyttää oppilaan toista nimeä.)**

Kysy jokaiselta kurssilaiselta yksitellen heidän koodinimensä. Muiden osallistujien on tarkoitus arvata koodiavainta apuna käyttäen, mikä on henkilön oikea nimi. *Tämä on käytännön esimerkki dekoodaamisesta.*

Harjoitus: Karkkirobotti

Keskustelua:

- Kysy kuinka monella on tietokone kotona ja minkälaisia ne ovat.
- Kysy missä muualla on tietokoneita ja mitä ne tekevät. Avaa osallistujien silmät sille, kuinka paljon tietokoneita on kaikkialla: puhelimet, pelikonsolit, digiboksit, televisiot, hissit, autot, pankit, potilasrekisterit, verotustietokannat jne.
- Kysy lapsilta tietävätkö he kuka päättää miten tietokoneet toimivat - kerro että ne noudattavat koodia, jotka koodaaja on tehnyt.
- Kysy tietävätkö lapset, mitä koodi on?
- Kerro koodauksen periaatteista: tietokoneet ovat tyhmiä, ja tekevät vain mitä koodaaja on käskenyt, juuri siinä järjestyksessä kuin hän on ohjeet kirjoittanut. Käytännössä koodi on vain kasa peräkkäisiä komentoja, joita tietokone lukee eli suorittaa rivi riviltä.
- Koodaajan pitää kirjoittaa tarkkaan koodi, koska jos siinä on virhe niin tietokone ei tajua sitä.

Kierros 1

Heijasta seinälle karkkirobotin ohje.

Kerro, että nyt kaikki pääsevät kokeilemaan koodaamista. Näytä lista komentoista (eteen, taakse, vasen, oikea, ojenna). Jokaisen tehtävä on kirjoittaa pieni ohjelma käyttäen listattuja komentoja: **askel eteen**, **askel taakse**, **käännä vasemmalle**, **käännä oikealle** ja **ojenna karkkipussi**.

Näytä esimerkkikoodi: askel eteen, askel eteen jne ja lopussa ojenna karkkipussi. Suorita koodi lukemalla se ääneen ja kävelemällä. Lopuksi ojenna karkkipussi jollekin, joka saa ottaa siitä karkin, jos "robotti" (sinä) ja karkinottaja ovat samassa ruudussa, jossa komento "ojenna" suoritetaan.

Kerro, että jokainen saa nyt itse kirjoittaa paperille koodin, jonka karkkirobotti suorittaa. Jos koodi on oikein, saa karkin. Jos karkkirobotti pysähtyy väärään kohtaan, menee liian



pitkälle tai yrittää suorittaa komennon, jota ei ymmärrä, palaa robotti alkuun ja ottaa seuraavan paperin (ohjelman) ja suorittaa sen. Jokaisen pitää kirjoittaa lappuunsa myös nimensä.

Kun ensimmäinen on valmis, ota paperi ja suorita se ja samalla lue ohjelma ääneen robottiäänellä. Tämän jälkeen tee kierros ja ota systemaattisesti jokaiselta yksi lappu koodia, jotta jokaisella on yhtäläinen mahdollisuus saada karkki. Tee pari kierrosta, jotta karkkirobotti suorittaa jokaisen koodin. Väärin tehty ohjelma annetaan tekijälleen takaisin, jotta sitä saa korjata.

Kierros 2

Heijasta seinälle karkkirobotin toisen kierroksen ohje (jossa myös toistokomento) ja anna esimerkki toistolausekkeen käytöstä.

Nyt osallistujat ovat karkkirobotteja. Yksi lapsista asettuu aloitusruutuun robotiksi, toiset kirjoittavat koodin, jolla he ohjaavat robotin luokseen käyttäen tällä kertaa **toistolauseketta**. Käytännössä he siis voivat käyttää jo kertaalleen kirjoittamaansa koodia ja yksinkertaistaa sitä toistolausekkeella. *Toistolauseketta käyttäen samaa komentoa voidaan toistaa halutun verran, eikä komentoa tarvitse kirjoittaa montaa kertaa peräkkäin.* Voit myös haastaa osallistujia tekemään jekkua robotille laittamalla sen toistamaan esim. eteen-taakse komentoa useita kertoja, sen sijaan, että ohjaisivat robotin suoraan luokseen. Voit myös vaihtaa robotin aloituspaikkaa, jolloin pitää kirjoittaa uusi koodi.

Kun on valmista, kysy osallistujilta miltä tuntui kirjoittaa koodi? Oliko se vaikeaa? Miten pitää ajatella, kun kirjoittaa koodia, suoritus pitää ehkä itse kuvitella päässä ennen kuin sen kirjoittaa? Voiko jonkun toisen koodia lukea ja kuvitella mitä se tekee?

Opitut asiat:

Kerro lopuksi, mitä olette oppineet tänään:

- **Koodaamisen periaatteen:** tietokone lukee kirjoitettua koodia rivi riviltä ja tekee kaiken kirjoitettujen ohjeiden mukaan.
- **Kodiavaimella** pystyy salaamaan ja purkamaan viestin.
- **Enkoodaus** tarkoittaa viestin salaamista ja **dekoodaus** viestin purkamista.
- **Toistolausekkeen** avulla samaa komentoa voidaan toistaa halutun verran, eikä komentoa tarvitse kirjoittaa montaa kertaa peräkkäin.



Oppitunti 2: Code.org: Ohjelmoinnin perusteet

Tavoite

- Suorittaa ohjelmoinnin perusteet [Code.org](https://code.org) -harjoituksia, jossa annetaan tietokoneen ruudulla olevalle hahmolle liikkumiskomentoja
- Ensikosketus graafiseen ohjelmointiin tietokoneella
- Oppia käytännössä ohjelmoinnin perusrakenteita: **kirjastofunktiot**, **toistolauseke** (for loop), **kunnesilmukka** (while loop), **ehtolauseet** (if, else -rakenne)

Valmistelut

- Harjoittele kotona valmiiksi seuraavat perusharjoitukset:

[Code.org: Moana/Vaiana](#)

[Code.org: Angry Birds](#)

[Code.org: Minecraft](#)

- Lisäharjoitukset:

[Mehiläinen: silmukoita](#)

[Mehiläinen: funktiot](#)

- Ennen tunnin alkua; varmista että tietokoneista on äänet pois päältä.
- Varmista, että saat heijastettua tykillä koneesi näytön luokan seinälle/kankaalle.
- Kirjoita harjoitusten URL esimerkiksi taululle, heijasta tykillä tai jaa muistilaput jokaiselle. Mikäli oppilaat haluavat jatkaa harjoituksia kotona, on muistilappu paras vaihtoehto.

Jos haluat seurata oppilaidesi edistymistä Code.orgissa voit tehdä oman opettajaprofiilin. Tällöin voit myös valita harjoituksen, johon oppilaat automaattisesti Code.orgiin kirjautumiskorteilla päätyvät.

- Rekisteröidy opettajana code.org-verkkosivustolle
- Sivulta <http://code.org/teacher-dashboard> luo oma ryhmä ja luo oppilaille käyttäjätunnukset syöttämällä heidän tietonsa sivun Hallinnoi oppilaita -kautta. Valitse haluamasi harjoitus. Paina lopuksi *Tulosta kortit oppilaidesi kirjautumistiedoista* ja tulosta oppilaiden kirjautumiskortit. Tulosteesta voi leikata kirjautumiskortit, jotta ne on helppo jakaa oppilaille.



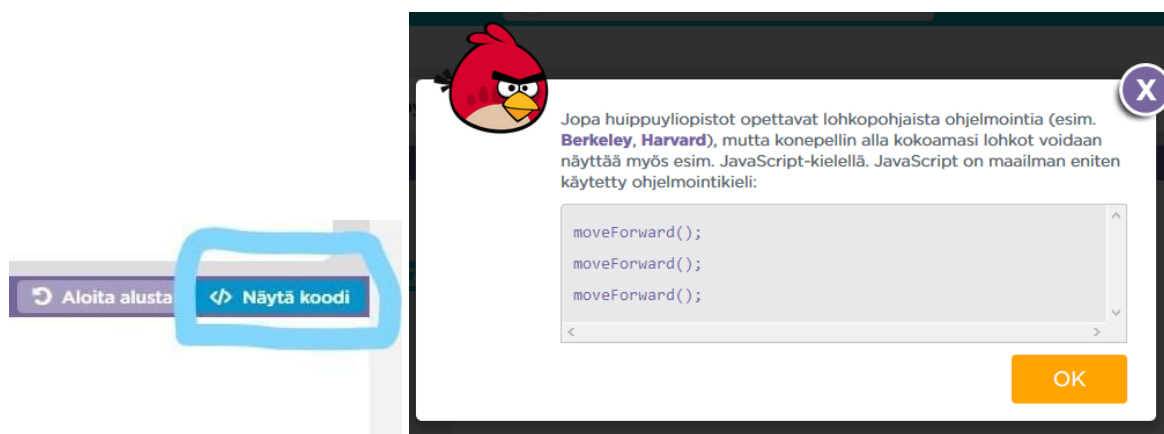
Harjoitus:

Näytä miten tietokone avataan, miten kirjaudutaan sisään, miten avataan nettiselain ja kirjoitetaan urliksi code.org, josta avataan koodaustuntisivu ja sieltä valitsemasi perusharjoitus. Suosittelen aloittamaan Moanalla/Vaianalla, koska se on harjoitus, jota moni ei ole tehnyt. Angry Birds ja Minecraftin ovat monet tehneet koulussa. Toki asiaa voi kysyä oppilailta ja edetä sen mukaan.

Näytä ensimmäisen tehtävän suorittaminen. Kerro yhtäläisyyksistä Karkkirobottiin: tietokone on tyhmä, koodin pitää olla täsmälleen oikein ja koodin suorittamista saa kokeilla niin monta kertaa kuin haluaa. Kerro myös, ettei maksimilohkomäärää saa ylittää, koska silloin koodi on spaghetti-koodia, eli opetettavia rakenteita ei käytetä oikein/tehokkaasti. Eli kaikkien suorituspallojen tulee olla tummanvihreitä. Jos suorituspallo jää vaaleanvihreäksi se kertoo, että oppilas on oikonut. Kerro, että valmiita olemassa olevia funktiota (esim. liikkumiskomennot) sanotaan kirjastofunktioiksi, ne ovat kaikkien käytettävissä ja näin ollen esimerkiksi liikkumiskoodia ei tarvitse itse erikseen kirjoittaa.

Vaihtoehtoinen harjoitus: Mikäli ryhmässä on harjoituksen jo aiemmin tehneitä tai osa ryhmästä ehtii saada sen valmiiksi, voivat oppilaat jatkaa tehtävistä [Mehiläinen: silmukoita](#) ja harjaantuneille [Mehiläinen: funktiot](#)

Muistuta, että tehtävän suoritettuaan voi käydä katsomassa "Näytä koodi" (Moanassa/Vaianassa tätä mahdollisuutta ei ole), joka näyttää ohjelmoidut lohkot JavaScript-kielellä. JavaScript on maailman eniten käytetty ohjelmointikieli.





Opittu asia:

Kerro lopuksi, että Code.org-harjoituksia voi jatkaa halutessaan kotona ja että tänään opittiin käyttämään seuraavia ohjelmoinnin perusrakenteita:

- **Kirjastofunktiot** ovat kaikkien käytettävissä olevia valmiita toimintoja, joita voi käyttää omassa ohjelmakoodissa (esim. liikkumiskomento = moveForward())

- **Toistosilmukan** (toistolauseke) eli for loopin avulla samaa toimintoa voi toistaa niin monta kertaa kuin haluaa.



- **Kunnessilmukan** (kunneslauseke) eli while loopin avulla samaa toiminto toistetaan niin kauan, kunnes silmukan pysähtymiseksi asetettu ehto täytetään, esimerkkinä tästä Angry Birds -harjoitus, jossa toistetaan liiku eteenpäin -komentoa, kunnes saavutetaan possu.



- **Ehtolausekkeessa** eli if, else -rakenteessa tietokone valitsee toiminnon tiettyjen ehtojen mukaan. Esimerkiksi Angry Birds -tehtävää mukaillen: jos edessä on polku, liiku eteenpäin, muuten käänny vasempaan.





Oppitunti 3: Code.org: Frozen ja Taiteilija

Tavoite

- Suorittaa Code.org:n Frozen ja Taiteilija, jossa annetaan tietokoneen ruudulla olevalle hahmolle liikkumiskomentoja
- Ymmärtää, mitä ovat kirjastofunktioille annettavat **parametrit**
- Ymmärtää kuljettu matka kuvapisteidien avulla
- Ymmärtää kääntyminen ja kulmat

Valmistelut

- Harjoittele kotona valmiiksi seuraavat harjoitukset:
[Code.org: Frozen \(code with Anna and Elsa\)](#)
[Code.org: Taiteilija](#)
- Kysy Koodikerholta mahdollisuutta saada Code.org-robotti lainaksi lisäharjoitusta varten.

Harjoitus: Frozen ja Taiteilija

- Kerro, että tässä harjoituksessa opetellaan liikkumista, kääntymistä ja parametrien antamista.
- Näytä mikä on kuvapiste: hyvin pieni neliö tietokoneen ruudulla.
- Näytä ohje kulmista (kun kääntymislohkosta klikataan kulmaparametria) ja kerro, että niitä käyttämällä saa hahmon kääntymään tarkasti haluttuun suuntaan.
- Pyydä lapsia kirjautumaan code.org – sivustolle ja valitsemaan koodaustuntien alta Frozen ja kun sen on saanut valmiiksi jatkaa haasteellisemmalla taiteilijalla.
- Korosta, että tehtävän suoritettuaan voi käydä katsomassa ”Näytä koodi”, joka näyttää ohjelmoidut lohkot JavaScript-kielellä. JavaScript on maailman eniten käytetty ohjelmointikieli.

Opitut asiat:

Kerro lopuksi, mitä ollaan opittu ja mitä ne tarkoittavat:

- Liikkuminen kuvapisteidien avulla, kääntyminen sekä kulmat.
- **Parametrit** ovat komennoille (funktiolle) annettavia alkutietoja, joiden perusteella funktio toimii tietyllä tavalla, esimerkiksi Frozenin *liiku eteenpäin* -komennolle annettava kuvapistemäärä sekä suunta ovat kyseisen funktion parametreja.





- **Funktiot** (taiteilijan lopussa on muutama funktio eli aliohjelmaharjoitus edistyneemmille)

Lisäharjoitus, jos sinulla on mahdollisuus saada Koodikerholta lainaksi Code.org -robotti

Kysy millaisia robotteja osallistujat tietävät esim. elokuvissa: Star Wars, Terminator jne. Kerro, että oikeat robotit yleensä suorittavat tehtäviä, jotka ovat liian yksinkertaisia, likaisia tai vaarallisia ihmiselle tai joihin ihmisen hienomotoriikka ei ole riittävän tarkkaa. Sovelluksia ovat mm. lattioiden siivous, nurmikonleikkuu, myrkkujen siivous, vedenalainen ja avaruudessa tapahtuva tutkimus, kirurgia, kaivostoiminta, etsintä ja pelastus, räjähteiden paikannus. Kotitalouksiin on kehitetty muun muassa imurointi- ja ruohonleikkuurobotteja.

Kerro, että robotiksi kutsuttu laite voi suorittaa monimutkaisia tehtäviä joko suoraan ihmisen käskyttämänä, osittain ihmisen käskyttämänä, ihmisen valvonnan alla tai täysin autonomisesti (tietokoneen käskyttämänä).

Kerro että Frozenin kuvapiste on robottia ohjatessa 1 sentti.

Jaa ryhmä pareihin. Parin toinen kone on internetissä Code.org-sivustolla, mistä voi katsoa mallia Frozenin/Taiteilijan koodista. Toinen kone on yhteydessä robotin WLANiin, jolloin sitä voi ohjata kirjoittamalla selaimen osoitteeksi <http://192.168.0.1/>. Robottia voi ohjata useampi yhtä aikaa, mutta kuitenkin vuorotelle, joten pyydä että parit pyytävät viittaamalla koodillensa suoritusluvan.

Koodikerhon robottiohjeita:

<https://github.com/ottok/code-org-robot/raw/master/instructions/Finnish.odt>



Oppitunti 4: Johdatus pelien tekemiseen Code.org: Flappy code, Bounce ja Star Wars

Tavoite

- Oppia mitä tarkoittaa, että tietokoneohjelma on vuorovaikutteinen
- Mitä ohjelmoinnissa tarkoittaa **event** eli tapahtuma

Valmistelut

- Harjoittele kotona valmiiksi seuraavat harjoitukset:

[Code.org: Flappy](#)

[Code.org: Bounce](#)

[Code.org: Star Wars](#)

Harjoitus

- Kerro eventeistä eli tapahtumista:
- Peleissä tapahtumia on pääasiassa kahdenlaisia.

Ensimmäinen on **käyttäjän aiheuttama tapahtuma**. Eli joka kerta, kun käyttäjä/pelaaja napsauttaa nappia tai kirjoittaa tietokoneella, siitä luodaan "tapahtuma", ja siihen tapahtumaan liitetty/kirjoitettu koodi päättää, **mitä tehdään, kun tuo tapahtuma aktivoituu**. Esimerkiksi "kun hiirtä napsautetaan, soita ääni". Kerro, että myös Code.orgissa jo paljon käyttämämme suorita-napin painaminen on käyttäjän aiheuttama tapahtuma, josta seuraa ohjelmalohkojen suorittaminen eli tietokone käy läpi lohkojen alle piilotetun koodin ja toimii sen mukaan.

Toinen tapahtumatyyppi on **pelin logiikan aiheuttama tapahtuma**, joka aktivoituu silloin, kun pelissä tapahtuu jotain. Esimerkiksi pelihahmon osuessa maahan tai toiseen pelihahmoon, siihen liitetty tapahtuma eli eventti laukaisee tietyn toiminnan, jonka seurauksena eventin alle kirjoitettu koodi suoritetaan. Esimerkkinä tästä "kun pelihahmo osuu seinään, peli loppuu".



Opitut asiat:

Kerro lopuksi, mitä ollaan opittu ja mitä ne tarkoittavat:

- **Tietokoneohjelma on vuorovaikutteinen;** käyttäjän hiiren liikuttaminen tai napin painallus vaikuttaa esimerkiksi tietokonepelin kulkuun.
- **Eventtejä eli tapahtumia** on kahdenlaisia; käyttäjän aiheuttamia tai pelin logiikan johdosta tapahtuvia.



Oppitunti 5: Scratch: Kirjautuminen ja tarina

Tavoite

- Kirjautua Scratchiin
- Käydä läpi Scratchin ominaisuudet
- Opetella Scratchin käyttöä ohjatun yhteisharjoituksen avulla
- Tutustua käsitteisiin: luokat, oliot, olioiden metodit ja viestien välitys oliolta toiselle (= ohjelmassa tapahtuva tapahtuma eli event)

Valmistelut:

- Tee itsellesi gmail-osoite, jota oppilaat voivat käyttää, kun tekevät Scratch-tunnukset. Monella kun ei ole omaa sähköpostiosoitetta. Scratchissa koko luokka voi käyttää yhtä ja samaa sähköpostiosoitetta ja tästä on sekin hyöty, että jos joku unohtaa tunnuksensa tai salasansa tai molemmat, pystyt katsomaan luodut tunnukset luomastasi sähköpostista ja tarvittaessa resetoimaan salasanan, jotta oppilas pääsee sillä kerralla jatkamaan.
- Ota kurssille mukaan paperilappuja, joihin oppilaat merkkäavat Scratch-tunnukset muistiin.
- Harjoittele käyttämään scratchia, linkin takana valmiita tutorialeja peleihin. Esimerkiksi *Race to the finish* on idealtaan yksinkertainen ja helppo ymmärtää. Tutoriaalit ohjeistaa, miten peli tehdään ja neuvoo sen vaihe vaiheelta. [Scratch: tutoriaaleja erilaisiin peleihin](#) . Hyviä neuvoja Scratchin käyttöön löytyy myös Aku Ankan sivulta: [Koodaa Ankat liikkeelle](#)
- Harjoittele tarina, joka koodataan yhdessä tunnilla. Linkkien takana esimerkkejä Kodareiden Suvin tekemistä tarinoista:

Ötökkäapu: <https://scratch.mit.edu/projects/121952384/#editor>

Perhoset lähtevät kaupunkiin:

<https://scratch.mit.edu/projects/145452909/#editor>

Talvi tulee: <https://scratch.mit.edu/projects/140058464/#editor>

- Vinkkejä: tulosta harjoittelemasi tarina paperille, josta voit luntata, mikäli et muista tarinan koodausta ulkoa. Hahmoja liikutellaan usein miten

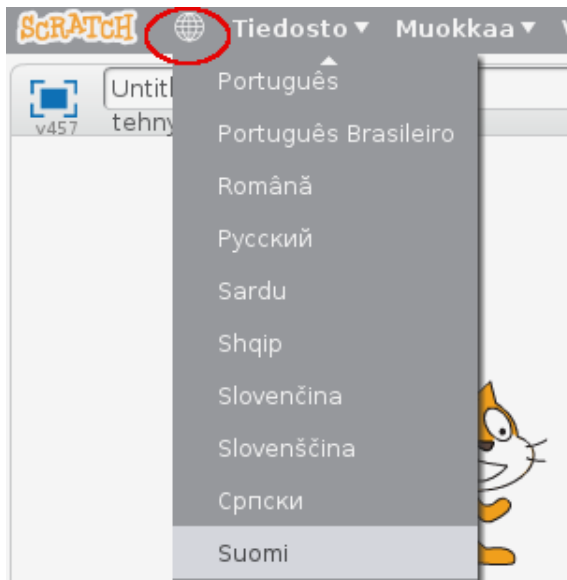


xy-koordinaateilla, mieti miten selität ne oppilaille, Scratchistä saat päälle koordinaatiston, joka helpottaa sen ymmärtämistä. Oppilaat saattavat tarvita paljon apua, mieti etukäteen, mitkä ovat tarinan minimitoiminnot, jotka koodaatte: esim. ötökkäapu-tarinassa ravun saksien ei tarvitse liikkua ravun kävellessä ja saksien liikuttamiseen voi palata, mikäli oppitunnilla jää aika.

Harjoitus: Scratchiin kirjautuminen ja tutustuminen

Kirjautuminen

- Scratch-pääsivun yläreunassa on Liity Scratchiin. Käykää yhdessä sen vaiheet läpi. Jaa paperilaput ja pyydä kirjoittaman tunnukset muistiin. Kerää tunnin lopussa laput talteen seuraavaa kertaa varten.
- Scratch-tunnus vaatii mailiosoitteen. Jos osallistujilla ei ole omaa osoitetta, neuvo heitä käyttämään luomaasi sähköpostiosoitetta. Scratch lähettää mailin jokaisesta uudesta kirjautumisesta, käy Gmailissa hyväksymässä ne.
- Varmista, että **kaikilla oppilailla on käytössä suomenkielinen Scratch**. Kielen saa vaihdettua Scratchin käyttöliittymästä maapallon kuvasta



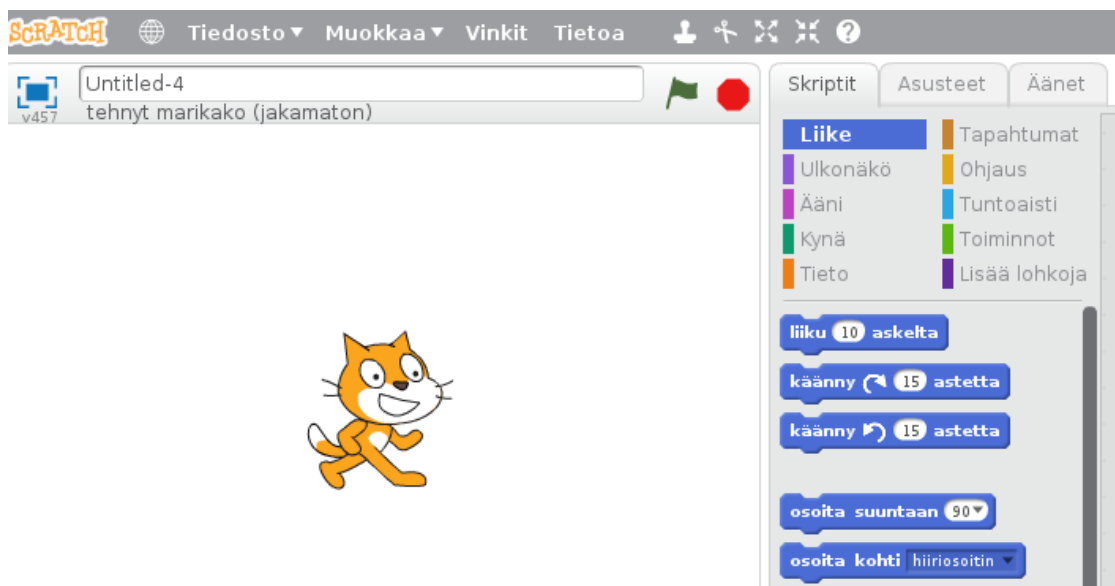


Kerro Scratchista:

- Aloittaa tyhjältä pöydältä, toisin kuin Code.org, jossa tehdään ohjattuja tehtäviä.
- Monipuolinen, sisältää paljon ominaisuuksia joita voi soveltaa mm. omiin peleihin ja animaatioihin
- Tuotokset voi jakaa internetissä
- Grafiikoiltaan yksinkertainen, mutta niin monet suosikkipelit ovat
- Scratchissä voi kuitenkin hakea hahmojen (png-tyyppisiä rasteritaustaisia hahmoja) sekä taustojen grafiikoita myös internetistä, jolloin peleistä saa persoonallisempia.

Käy läpi työpöydän toiminnot, näytä tykillä seinälle:

- Luo esimerkkihahmo, tausta, esitele liike yms. laatikoiden toiminnot. Kerro luokista ja olioista. Käytä esimerkkinä, että hahmot ovat hahmoluokan olioita ja tausta taustaluokan olio ja että näistä luokista luoduilla olioilla on erilaisia kykyjä eli metodeja. Esimerkiksi, että hahmoluokan olio pystyy liikkumaan, mutta taustaluokasta luotu taustaolio ei.



Yhteisharjoitus Tarina:

- Kun kaikki ovat kirjautuneet omilla tunnuksillaan, tehkää yhdessä harjoitustarina, jossa opetellaan Scratchin perusominaisuuksia.



- Näytä koko ajan omalta koneelta tykin kautta. Kaikki koodaavat samoin toimivan tarinan, mutta osallistujat voivat itse päättää omat hahmot ja taustat.
- Hahmottamisen helpottamiseksi voit näyttää aluksi valmiin tarinan, jota alatte tekemään. Siten oppilaat näkevät, mitä tarinalla yleensäkin tarkoitetaan.

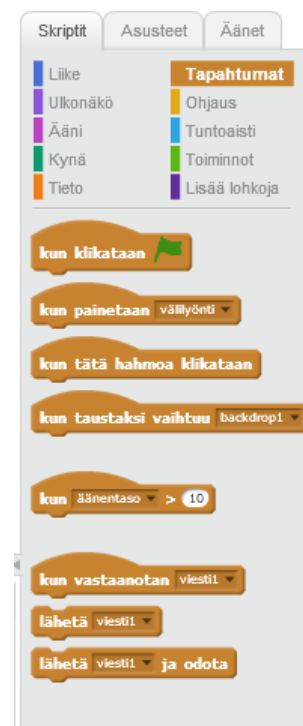
Harjoituksen vaiheet:

- Luodaan tausta ja kaksi hahmoa (esim. rapu ja sammakko)
- Toinen hahmoista aloittaa tarinan lähtee liikkumaan kohti ruudun oikeaa reunaa.
- Reunaan osuttaessa ihmettelee miksei pääse eteenpäin ja lähettää paikalle jääneelle hahmo-oliolle viestin.
- Kun toinen hahmo-olio vastaanottaa viesti-eventillä viestin se jatkaa tarinaa sanomalla jotain ja liukuu toisen hahmon luokse, jonka jälkeen voi tapahtua jotain hassua :)

Opitut asiat:

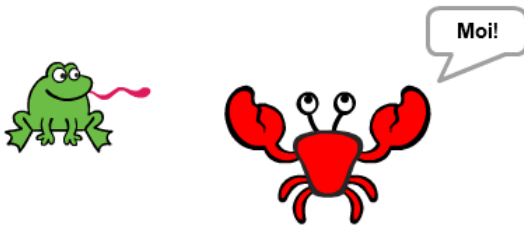
Kerro lopuksi, mitä ollaan opittu ja mitä ne tarkoittavat. Lasten saattaa olla vaikeata ymmärtää luokkia, olioita ja metodeita näin ohjelmoinnin alkuvaiheessa, joten riittää, että kertoo mihin ne liittyvät.

- **Luokat, oliot ja metodit:** hahmot ovat hahmoluokan olioita ja taustat ovat taustaluokan olioita. Taustaluokasta luoduilla olioilla on erilaisia ominaisuuksia kuin hahmoluokalla; esimerkiksi taustaluokan oliot eivät voi liikkua ja hahmoluokan oliot voivat. Olioiden eri toiminnallisuuksia/kykyjä sanotaan olioiden metodeiksi.





- Viestejä välitetään oliolta toiselle **eventtien** eli tapahtumien avulla, ks. alla oleva sammakko-rapu-esimerkki, jossa sammakko on viestin lähettäjä ja rapu vastaanottaja.





Oppitunti 6: Scratch: Selviytymispeli

Tavoite

- Tehdä Scratch-peli
- Tutustua käsitteisiin: muuttujat, satunnaisluvut, ikuisestsilmukka ja eventit eli tapahtumat.

Valmistelut

- Harjoittele valmiiksi esimerkin mukainen selviytymispeli Scratcilla:
 Avaruuspele: <https://scratch.mit.edu/projects/130926641/#editor>
 Sukelluspeli: <https://scratch.mit.edu/projects/104296345/#editor>
- Vinkki: tulosta selviytymispelin koodit paperille mukaan, niistä voi helposti luntata tunnin aikana pelin kulkua, eikä tarvitse muistaa koko koodia ulkoa.
- Mieti etukäteen, miten selität hahmon liikuttamisen x- ja y-koordinaateilla, se on yllättävän haastavaa ala-asteikäisille. Scratchissä saa xy-koordinaatiston näkyville, sitä kannattaa käyttää opettamisessa hyödyksi.
- Mieti etukäteen myös, mitkä asiat haluat ainakin opettaa/koodata oppitunnin aikana, sillä koko peliä ei välttämättä kerkeä tekemään valmiiksi. Aikaa kuluu yllättävän paljon oppilaiden auttamiseen. Voit esim. aloittaa päähahmon toiminnoista tai sitten koodata vihollishahmot ensin.
- Jaa edelliskerralla tehdyt Scratch-tunnukset osallistujille. Jos joku on kirjoittanut lapullensa väärin, katso luomastasi sähköpostista oikea tunnus ja reseto salasanana tarvittaessa.

Harjoitus: Selviytymispeli

Harjoituksen vaiheet:

- Luodaan tausta
- Luodaan yksi hahmo keskelle (kohtaan $x=0, y=0$). Hahmo liikkuu nuolinäppäimillä (ylös, alas, oikea, vasen)
- Luodaan 3 muuta hahmoa, jotka liikkuvat satunnaisesti (satunnaisluku x ja y)
- Pelin tavoitteena on, että keskushahmon pitää väistellä muita hahmoja. Pelin häviää, jos keskushahmo koskettaa muita hahmoja. Pelin voittaa, jos onnistuu välttelemään hahmoja esim. 20 sekuntia
- Koodatkaa "hävisit pelin", jossa osallistuja saa itse valita kuinka esim. keskushahmo sanoo sen
- Tähän saakka pelin voi koodata pyörimään "ikuisesti"

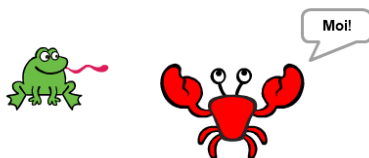


- Luo muuttuja kello
- Koodatkaa pelin voittaminen: Vaihda "ikuisesti" tapahtumaan "toista 20 kertaa"
- Muistuta säännöllisesti tallentamisesta, joka tapahtuu menusta tiedoston alta tai käyttöliittymän oikeasta yläkulmasta
Tallenna nyt -kohdasta

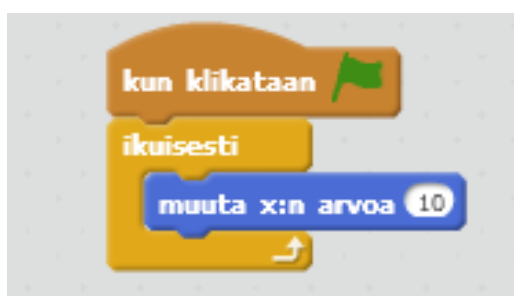


Opitut asiat:

- **Eventit** ovat tapahtumia ja niiden välityksellä voidaan esimerkiksi välittää tietoa toiselta hahmolta toiselle. Toinen hahmoista on lähettäjä osapuoli ja toinen vastaanottaja. Alla olevassa esimerkissä sammakko on lähettäjä ja kun sitä klikataan se lähettää eventin nimeltä sanoMoi. Rapu on vastaanottaja ja saadessaan viestin se sanoo Moi!



- **Ikuisestisilmukka** toistaa silmukan sisällä olevia toimintoja sanansa mukaan ikuisesti. Scratchissä ikuisestisilmukan avulla hahmot saadaan esimerkiksi liikkumaan taukoamatta laittamalla silmukkaan, jokin liikkeeseen liittyvä metodi.

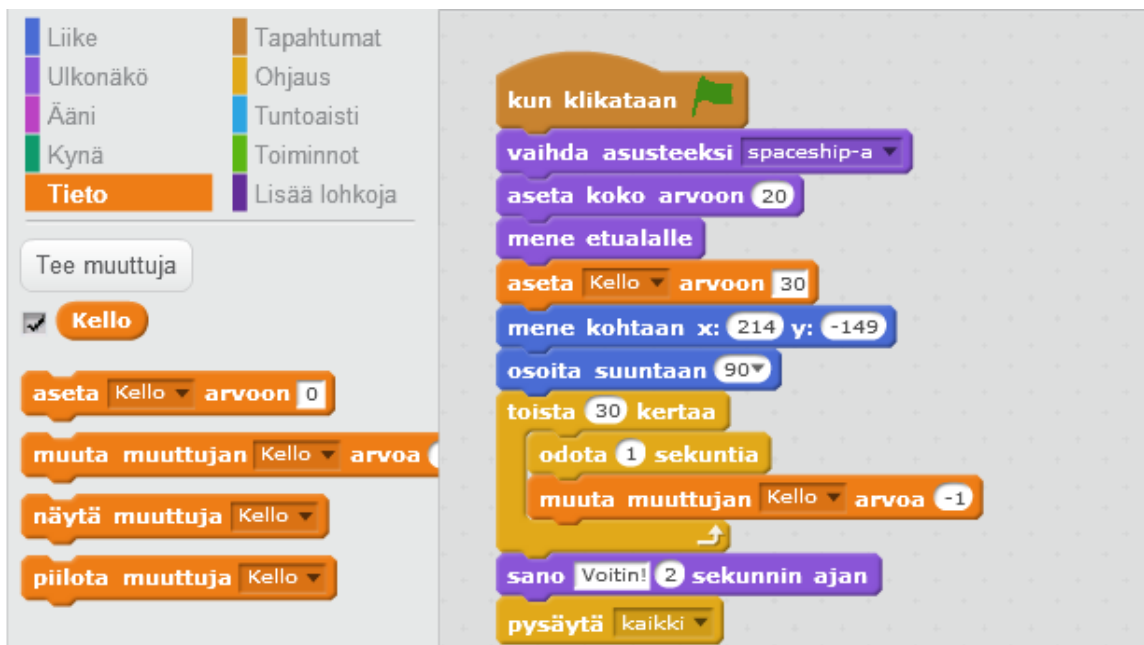




- **Satunnaislukua** voidaan käyttää esimerkiksi muuttamaan x:n ja y:n arvoja satunnaisesti tietystä arvosta toiseen arvoon. Näin tekemällä saadaan esimerkiksi hahmot liikkumaan edes takaisin ruudulla. Huom. x voi saada arvoja väliltä -240 – 240 ja y väliltä -180 – 180



- **Muuttujia** voidaan luoda taustaoliolle sekä hahmoille, niihin tallennetaan tietoa, jota voidaan muokata koodissa, miten halutaan. Esimerkkinä tästä avaruuspelissä luotu kello, joka asetetaan arvoon 30 ja toistosilmukan avulla aikaa vähennetään sekunnin välein.





Oppitunti 7: Oma peli Scratchilla

Tavoite

- Osallistujat luovat oman pelin/tarinan Scratchilla tai tuunaa/muokkaa olemassa olevia
- Kerrata edellisillä tunneilla opittuja käsitteitä: luokat, oliot, eventit, silmukat, muuttujat.
- Harjoitella oman tuotoksen esittelyä muille

Valmistelut

- Harjoittele valmiin Scratch-projektin muokkaamista siltä varalta, että joukossa on lapsia, jotka eivät keksi millaisen tarinan/pelin tekisivät. Aloitteilyryhmän kanssa voi oman pelin sijaan remiksata valmiita Scratch-projekteja: https://scratch.mit.edu/starter_projects/ . Näytä muutama valmis esimerkki tykillä yhteisesti. Projektia pääsee remiksaamaan Katso sisälle / Remix-nappulalla oikeassa yläkulmassa. **Muistuta säännöllisesti tallentamisesta.**
- Kodareiden Suvin tekemä hiiriralli on myös hauska harjoitus lapsille, jotka eivät keksi omaa projektia <https://scratch.mit.edu/projects/179558760/>
- Jaa edelliskerralla luodut Scratch-tunnukset osallistujille.



Harjoitus: Oma peli/tarina Scratchilla, esittely

- Anna oppilaiden suunnitella ja tehdä scratchilla oma peli. Näytä tykillä ennen tehtävään ryhtymistä, muutamia valmiita projekteja https://scratch.mit.edu/starter_projects/ Kiertele neuvomassa, miten omia ideoita saa muutettua koodiksi.
- Tunnin lopuksi ne jotka haluavat voivat esitellä oman tuotoksensa. Pyydä heijastamaan tykillä seinälle, kertomaan, mitä ovat tehneet ja näyttämään myös koodin.
- Rohkaise jatkamaan kotona Scratchilla ohjelmointia. Scratchissa on paljon jaettuja pelejä, joiden koodia katsomalla voi oppia lisää.

Opitut asiat:

- Ohjelmoinnin peruskäsitteet toistuvat ja tulevat yhä tutuimmiksi; luokat, oliot, eventit, silmukat, muuttujat ja satunnaisluvut
- Oman idean muuttaminen toimivaksi peliksi/tarinaksi sekä siihen liittyvät haasteet: miten saan muutettua päässäni olevan idean toimivaksi koodiksi?