

Ensiaskleet ohjelmointiin

S U O M E N
**KOODI-
KOULU**



Opettajan opas



Jussi Koivisto,
Juuso Nieminen

Ensiaskleet ohjelmointiin

Opettajan opas - Versio 2.0

Kustantaja: Suomen Koodikoulu Oy
www.codeschool.fi

Copyright © 2022 Suomen Koodikoulu Oy



Tämä teos on lisensoitu Creative Commons
Nimeä-EiKaupallinen-JaaSamoin 4.0 Kansainvälinen -lisenssillä.
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

ISBN: 978-952-7403-29-7

Opettajan oppaan rakenteesta

Opettajan opas sisältää kaikki *Oppilaan kirjan* sivut (vasen), sivulta 5 alkaen. Opettajalle suunnattu ohjeet, tiedot ja työkalut on sivun oikealla puolella. Opettajan opas sisältää esimerkkiratkaisut haasteisiin, esimerkki projekteja, työkaluja arviointiin ja sisällön laajentamiseen.

Symbolien selitykset

Oppilaan kirjassa toistuu neljä toimintaa kuvaavaa symbolia: **Ohjeet**, **Testaa ja tutki**, **Haaste!** ja **Keskustele**.

Ohjeet

→ *Oppilas seuraa kirjan ohjeita tai opettaja ohjaa osion.*

Testaa ja tutki!

→ *Oppilaat tutkivat ja kokeilevat eri työkaluja ilman tarkkoja ohjeita.*

H **Haaste!**

→ *Oppilaat ratkaisevat rajattuja tai avoimia ongelmanratkaisutehtäviä.*

Keskustele

→ *Oppilaat keskustelevat aiheista parin, ryhmän tai koko luokan kesken.*

Arviointi

→ *Tämä symboli esiintyy ainoastaan opettajan oppaassa. Sen yhteydessä esitellään työkaluja ja vinkkejä arviointiin.*

Ensiaskleet ohjelmointiin - Oppimiskokonaisuuden esittely

Tässä oppimiskokonaisuudessa oppilaat tutustuvat ohjelmoinnin alkeisiin ja projektilähtöiseen ohjelmoinnin oppimiseen tarkoituksenmukaisella tavalla.

Oppimiskokonaisuus on tarkoin suunniteltu polku, joka mukailee pelillistä oppimista: Oppilaalle esitellään uusia työkaluja, niiden käyttöä harjoitellaan, sovelletaan ja yhdistetään aikaisemmin opittujen työkalujen kanssa. Harjoituksia ja projekteja suositellaan tekemään parityönä. Oppimiskokonaisuuden oppimistavoitteet on sovellettu *Uudet lukutaidot* -hankkeessa tarkennetuista, POPS:iin pohjautuvista ohjelmointitaitojen tavoitteista*.

Ohjelmointiharjoitukset tehdään Scratch-ohjelmointiympäristössä. Scratch on ilmainen verkkoselaimessa toimiva ohjelmoinnin opettamiseen kehitetty ohjelma, jota oppilaat voivat halutessaan käyttää myös kotona.

Ensiaskleet ohjelmointiin rakentuu kahdesta osiosta. Ensimmäisessä osiossa tutustutaan Scratch-ohjelmointiympäristöön, opitaan perusteet toimivien ohjelmien tekemiseen ja tehdään interaktiivinen tarina, joka voidaan yhdistää haluttuun oppiaineeseen. Osiossa 2 jatketaan ohjelmoinnin perusteiden oppimista ja tehdään oma sokkelopeli, joka myös voidaan yhdistää haluttuun oppiaineeseen.

* *Uudet lukutaidot, Ohjelmointiosaaminen*: <https://uudetlukutaidot.fi/ohjelmointiosaaminen/>

Oppitunnit

Oppimiskokonaisuus koostuu 24 oppitunnista. Alla on ehdotus tuntien käytöstä.

Lista oppitunneista		
Oppitunti #	Sisältö	Oppilaan kirjan sivut
1	Aloitus	8-10
2	Perusteet	11-16
3	Hahmojen lisääminen ja muokkaus, piirtäminen	17-20
4	Animointi ja taustat	21-25
5	Toistorakenteet, ongelmanratkaisu	26-29
6	Miniprojekti piirtäen	30
7	Projektin esittely: Suunnittelu ja aloitus	31-32
8	Projektityö	31-37
9	Projektityö	31-37
10	Projektin viimeistely	31-37
11	Projektien esittely ja testaaminen, itsearviointi	38
12	Osion 2 aloitus, ehtolauseet	39-44
13	Ongelmanratkaisu	45-49
14	Projektin esittely, esimerkkiprojektien testaus, suunnittelu	50-51
15	Pelipohjan tekeminen	52-58
16	Pelipohjan viimeistely	59-61
17	Projektityö	62-71
18	Projektityö	62-71
19	Projektityö	62-71
20	Projektityö, projektin viimeistely	62-71
21	Projektin viimeistely	62-71
22	Pelien testaaminen, vertaisarviointi	Lomake (tuloste)
23	Loppukoe	Koe (tuloste)
24	Arviointi, päätös	Koe (tuloste) + 72

Tämä ajankäyttösuunnitelma pohjautuu pääasiassa 4lk oppilaiden kanssa tehtyihin testauksiin. Mikäli oppilaat ovat vanhempia, voi eteneminen olla nopeampaa.

Sisällysluettelo

Osa 1 - Tutustuminen Scratchiin	7
Aloitetaan - Osa 1	8
Luetaan - Mitä ohjelmointi on?	9
Ohjelmoidaan - Ensimmäinen Scratch-ohjelmasi!	11
Scratchin käynnistäminen	11
Näin ohjelmoidaan	11
Tapahtumien käyttö	15
Ohjelmoidaan - Hahmojen lisääminen ja muokkaus	17
Lisää hahmo kirjastosta	17
Hahmojen muuntelu ja piirtäminen	19
Asusteiden ja animaatioiden tekeminen	21
Lisää esiintymislavalle tausta	24
Luetaan - Tietokoneet ovat hyviä toistamaan	26
Ohjelmoidaan - Ongelmanratkaisua toistolla	28
Määrätyn toiston haasteet	28
Jatkuvan toiston haasteet	29
Projekti - Interaktiivinen tarina	31
Suunnitelma	32
💡 Vinkkejä tarinan tekemiseen	33
Rinnakkaiset skriptit	33
Äänien käyttö	33
Ääniefektit ja musiikkiluupit	34
Ohjelman ohjaus taustaa vaihtamalla	35
Hahmojen piilotus ja näyttäminen	36
Odotakunnes -lohko (haastava)	36
Jatkuvien toistojen keskeyttäminen	37
Hahmon siirtäminen vasemmalle, oikealle, ylös, alas (haastava)	37
Lopuksi!	38

Tietoa opettajalle

Tämä oppimiskokonaisuus on suunniteltu *POPS 2014* ja *Uudet lukutaidot* -hankkeen osaamistavoitteiden pohjalta. Oppimiskokonaisuudessa korostuvat erityisesti laaja-alaisen osaamisen alueista L1, L2, L4 ja L5.

Oppimiskokonaisuus on kestoaltaan noin 24 tuntia. Se toimii erityisen hyvin monialaisena oppimiskokonaisuutena tai valinnaisena kurssina. Kohderyhmä on ohjelmointia aloittelevat vuosiluokkien 3-7 oppilaat.

Osa 2 - Ehtolauseet ja pelinteko	39
Aloitetaan - Osa 2	40
Luetaan - Ehtolauseet ja totuusarvo	41
Ehdot Scratchissa	41
Testaa ehtolauseita	44
Ehtolause peli	45
Ohjelmoidaan - Ehtolauseiden käyttäminen	46
Mr. Nosey -haasteita	46
Projektin aloitus - Oma sokkelopeli!	50
Tee pelille alustava tarina	51
Piirrä oma pelihahmo	52
Tee hahmosta riittävän pieni	53
Upeita värejä - Liukuvärit	54
Piirrä maali	55
Piirrä ensimmäiset kaksi sokkeloa	56
Ohjelmoi aloitustilanne	59
Projektityö - Viimeistele sokkelopeli	62
Minimivaatimukset projektille	62
💡 Vinkkejä pelin tekemiseen	62
Pelin tarina	62
Kentän vaihtumisen ongelmat	63
Liikkuvat viholliset	65
Ohjaa hahmoja tapahtumilla	66
Animaatiot	67
Graafiset tehosteet	68
Viestien käyttö (haastava)	69
Syvyysvaikutelmaa varjoilla (haastava)	69
Lopuksi!	72

Osa 1 - Tutustuminen Scratchiin

Tämän päivän maailma toimii tietokoneiden varassa ja tulevaisuudessa tietokoneet ovat varmasti vieläkin tärkeämpiä.

Tietokoneet eivät toimi sattumanvaraisesti. Ne seuraavat **koodia**. Sinä voit komentaa tietokoneet tekemään mitä tahansa **ohjelmoimalla niitä**. Ohjelmointi onnistuu erilaisilla **työkaluilla** joiden käyttö sinun on ensin opittava. Tämä kirja auttaa sinua pääsemään alkuun juuri tässä!

Aloitetaan tutustumalla Scratchiin antamalla yksinkertaisia komentoja.

Osa 1, sivut 7-38

Yleiskatsaus

Osassa 1 oppilaat tutustuvat ohjelmointiin ja työskentelyyn Scratchissa ja ohjelmoinnin peruskäsitteisiin. **Luetaan**-osioissa esitellään uudet käsitteet ja työkalut. **Ohjelmoidaan**-osioissa työkalujen käyttöä harjoitellaan käytännössä seuraamalla pääasiassa tutkien, testaten ja ratkaisten haasteita. **Projekti**-osuus esittelee osan 1 pääprojektin. Osan 1 suorittaminen kokonaan on suositeltua ennen osaan 2 siirtymistä.

Koko oppimiskokonaisuudessa suositetaan parityöskentelyä, mutta toteutuksen voi tehdä myös yksin tai 3 hengen ryhmissä.

Ajankäyttö

Osa 1 koostuu noin 10-14 oppitunnista (45 min/oppitunti).

Tarvikkeet

- Tietokone, näppäimistö, hiiri (väh. 1/oppilaspari)

Tärkeää osassa 1

- Rohkaise yhteistyöhön ja vertaisoppimiseen
- Anna oppilaiden tutkia, kokeilla ja epäonnistua
- Anna tilaa ja aikaa luovuudelle sekä itseilmaisulle

Aloitetaan - Osa 1

1. Missä olet nähnyt tietokoneita tänään? Oletko itse käyttänyt tietokoneita tänään? Kuinka monta kertaa?

2. Kokeile **leikkiä tietokonetta!** Pelaa tätä peliä parin kanssa. Yksi teistä on **tietokone** ja toinen on **ohjelmoija**. Tietokoneen pitäisi piirtää **kuva** ohjelmoijan antamien **ohjeiden** perusteella.

Säännöt:

- Tietokone **sulkee silmänsä**.
- Ohjelmoija antaa **ohjeita** yksinkertaisen **kuvan** piirtämiseksi.
- **Tietokone ei voi puhua tai pyytää apua.**
- Vaihtakaa osia ja kokeilkaa uudelleen!



3. a) Oliko vaikeampaa olla tietokone vai ohjelmoija?

- b) Oliko ohjelmoija *älykäs*? Entä tietokone?

4. a) Lopuksi - **ota puhelimesi selfie**, joka kuvaa **tunnettasi nyt kun alat harjoittelemaan ohjelmointia!**

- b) **Vertailkaa** selfie-kuvianne toisten oppilaiden kanssa.

Aloitetaan - Osa 1

Ajankäyttö

10 min

Työtapa

Työskennellään **pareittain**.

Keskustelua voidaan käydä myös parien välillä, **4 hengen ryhmissä**.

Selfie tai hymiö

Tehtävässä neljä oppilaita ohjataan ottamaan heidän tunnetilaa kuvaava kuva. Opettaja voi ohjata oppilaita myös piirtämään tunnetilaa kuvaavan hymiön.

Luetaan - Mitä ohjelmointi on?

Ohjelmointi on...



Ohjeiden antamista **tietokoneelle**. Ohjelmointi ja koodaus tarkoittavat samaa asiaa.



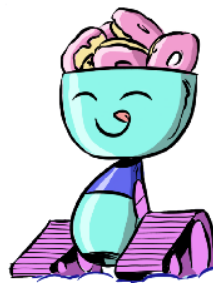
Leipurin tarina

Leipuri päätti leipoa **kakun** uudella **reseptillä**. Reseptin oli kirjoittanut Vitsi-Ville.

Taikinaa tehdessään leipuri huomasi reseptissä oudon ohjeen: **“Lisää kaksi kuppia suolaa.”**

Leipuri rapsutti päätään hämmentyneenä, mutta päätti seurata reseptiä sanatarkasti.

Kahden tunnin kuluttua kakku oli valmis. Leipuri maistoi sitä. **“YÄK”** kiljaisi leipuri. Kakku oli aivan karmea.



Keskustele

Pohtikaa seuraavia kysymyksiä **ryhmissä**. **Kirjoittakaa** vastauksenne.

1. Miksi kakku maistui pahalta?

2. Kenen syytä oli kakun epäonnistuminen?

Luetaan - Mitä ohjelmointi on?

Ajankäyttö

25 min

Työtapa

Työskennellään **2-3** hengen yksiköissä, lukien ja keskustellen

Avainsanat

Ohjelmointi, koodaus, ohjelma, koodi, ohje, hahmo, komento, lohko, skripti

Leipurin tarina

Tarina esittelee ohjelmoinnin vertauskuvallisesti. Ohjelmoidessa ohjelmoija tekee virheitä, ei tietokone; Kuten tietokone, myös tarinan leipuri seuraa vain ohjeita.

Esimerkkivastaukset tehtäviin

1. *Koska kakkuun laitettiin kaksi kuppia suolaa. Reseptissä on virhe.*
2. *Vitsi-Villen, koska hän on kirjoittanut reseptin.*

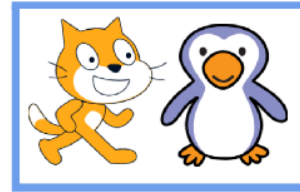
Ohjelmointi Scratchissa

Scratchissa voit **tehdä omia ohjelmia**, kuten pelejä ja sovelluksia. Seuraavat **tärkeät sanat** on hyvä selventää itselleen ennen Scratchin käytön aloitusta:

Hahmo on...



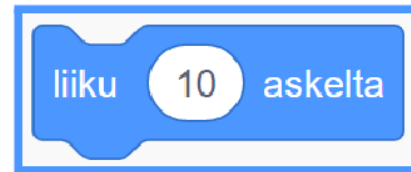
Digitaalinen hahmo, joka **seuraa komentoja** sanatarkasti. Scratchissa ohjelmoidaan hahmoja.



Komento on...



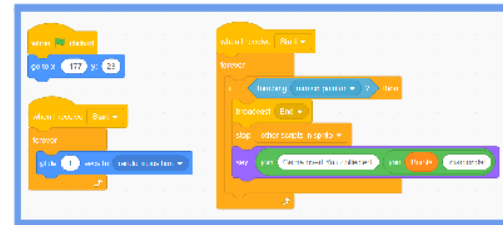
Yksi ohje hahmolle. Esimerkiksi **liiku (10) askelta**. Scratchissa komentoja kutsutaan **lohkoiksi**. **Skriptissä** on useita lohkoja kytkettynä toisiinsa.



Koodi on...



Joukko komentoja. Kun koodi muodostaa toimivan kokonaisuuden, sitä kutsutaan **ohjelmaksi**. Pelit ja sovellukset ovat ohjelmia.



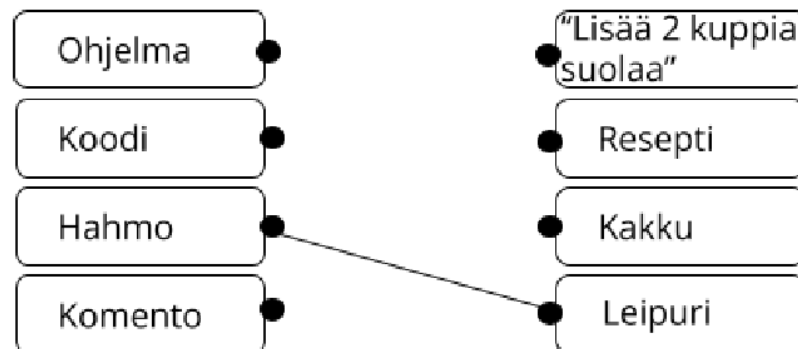
Keskustele

Keskustelkaa ryhmissä ja täyttäkää alla olevat tehtävät.

1. **Luekaa** "Leipurin tarina" **uudelleen**.

2. **Yhdistäkää** pisteet.
Esimerkki:

Leipuri on hahmo, koska hän seuraa reseptiä sanatarkasti, kuten hahmot Scratchissa.



Hahmot Scratchissa

Scratchissa perusajatuksena on ohjelmoida kuvaobjekteja tekemään haluttuja asioita kaksiulotteisella alustalla. Näitä kuvaobjekteja kutsutaan **hahmoiksi ja taustoiksi**. Tietokone toimii "kapellimestarina" kuvaobjektien käsittelyssä.

Ohjelmointiympäristö Scratch

Scratch on ilmainen verkkoselaimessa toimiva ohjelmointiympäristö, jota oppilaat voivat käyttää myös kotona. Scratchissa voidaan luontevasti tehdä esimerkiksi **animaatioita, pelejä ja yksinkertaisia hyötysovelluksia** ja se sopii erinomaisesti sekä alakoulun että yläkoulun ohjelmointiprojekteihin.

Scratch on tällä hetkellä **maailman suosituin graafinen ohjelmointiympäristö** ja sillä on noin 57 miljoonaa käyttäjää ympäri maailmaa. Se on käännetty yli 70 eri kielelle, mukaan lukien suomeksi. Jos oppilaiden Scratch ei ole valmiiksi suomen kielellä, kielen saa vaihdettua vasemman yläkulman maapallokuvakkeesta.



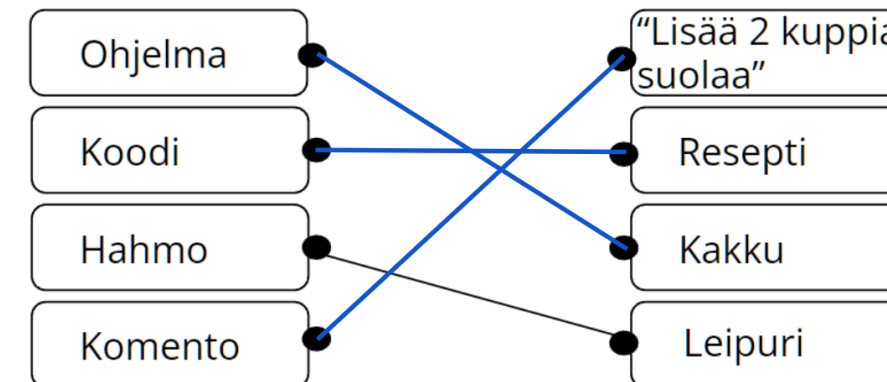
Scratchin perusidea on projektien jakaminen ja niiden "remiksaaminen" omiksi projekteiksi. Älä siis epäröi käyttää valmiita projekteja omiin tarkoituksiisi; se ei ole pelkästään sallittua vaan suositeltavaa! Esimerkiksi hakusanoilla "sustainability" tai "math quiz" Scratchista löytyy useita projekteja, joita voit pienin muutoksin käyttää opetuksessasi.



Scratchissa voit helposti remiksata valmiista projekteista omia projektejasi!

Esimerkkivastaukset tehtäviin

-
-



Ohjelmoidaan - Ensimmäinen Scratch-ohjelmasi!

Scratchin käynnistäminen

Ohjeet

↑ Tämä **Ohjeet-kyltti** tarkoittaa, että kirja haluaa sinun seuraavan alla olevia ohjeita.

1. **Avaa verkkoselain** tietokoneellasi
2. **Mene** osoitteeseen scratch.org
3. **Klikkaa Luo**

Luo

Huomaa: Jos opettajasi haluaa teidän käyttävän Scratch-tunnuksia, **odota opettajan ohjeita** ennen kuin jatkat.

Näin ohjelmoidaan

Nämä ohjeet sekä seuraavan sivun **Scratchin yleiskuva** auttavat sinut alkuun. Pääset pian kokeilemaan omia juttujasi!

Ohjeet

1. Vasemmassa laidassa näet kaikki lohkot. Eri toiminnoilla on eri väri. Avaa **Liike**-valikko.
2. **Siirrä liiku (10) askelta**-lohko **ohjelmointialueelle** klikkaamalla ja raahaamalla.



3. Koeta nyt **klikata lohkoa**. Kissa-hahmon tulisi liikkua joka kerta kun klikkaat.

Ohjelmoidaan - Ensimmäinen Scratch-ohjelmasi!

Ajankäyttö

45 min

Työtapa

Työskennellään **1-3** hengen yksiköissä

Avainsanat

Ohjelma, koodi, hahmo, komento, lohko, skripti, peräkkäisyys, liike, ulkonäkö, tapahtuma

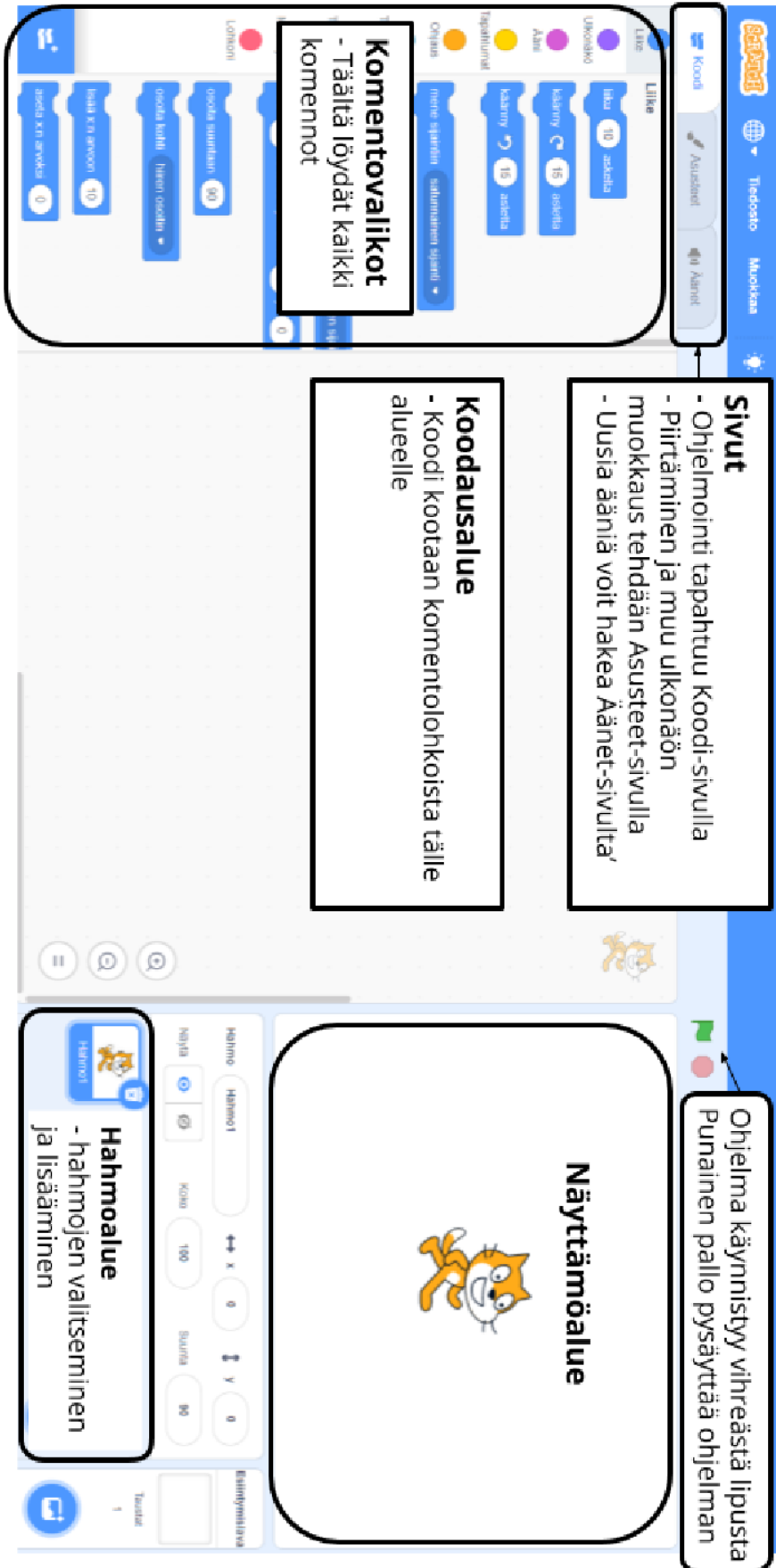
Scratchin avaaminen

Scratchin virallinen sivusto on scratch.mit.edu, mutta scratch.org ohjaa myös Scratchin etusivulle. Useimmille mobiililaitteille on saatavilla Scratchin sovellusversio (app).

Komentojen/lohkojen kokeilu

Oppilaille esitellään yksi tapa suorittaa lohkoja: klikkaamalla niitä. Tämä ominaisuus on kuitenkin vain lohkojen kokeilemista varten.

Scratchin yleisnäkymä



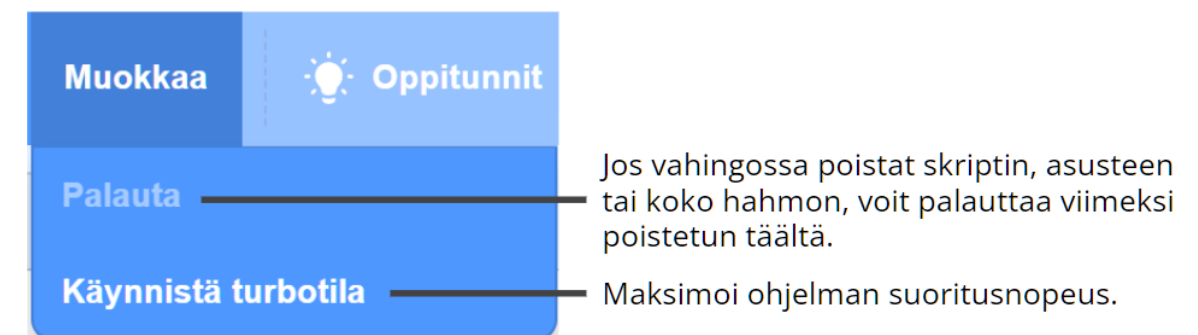
Ohjelmointiympäristön osioiden nimeäminen

Opettajan kannattaa käyttää ohjelmointiympäristön osista ja valikoista puhuessa johdonmukaisia nimiä. Esimerkit eri osioiden nimistä on esitelty oppilaan kirjan sivulla.

Kirjoissa käytetään sanaa **lohko** ja **komento** kuvaamaan yksittäisiä toimintaohjeita. Sanat ovat siis samaa tarkoittavia.

Tiedosto ja Muokkaa yläpalkissa

Yläpalkissa on tärkeitä ohjelmointiympäristön käyttöön liittyviä valikoita. Näitä ei esitellä oppilaan kirjassa. Ne on syytä esitellä oppilaille sopivassa vaiheessa opettajan ohjaamana.



Ohjeet

Nyt voimme yrittää **komentolohkojen kytkemistä** toisiinsa. Käytä nyt alussa paria lohkoa.

Liike- ja **Ulkonäkö-**valikoista.

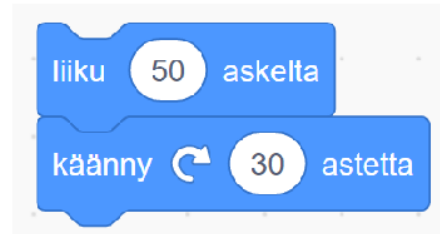
Näin pääset alkuun:

1. Aloita kytkemällä **kaksi lohkoa** **Liike-**valikosta.

2. Klikkaa niitä **monta kertaa**.

Olet tehnyt ensimmäisen **skriptisi!**

3. Kokeile lohkoissa olevien **numeroiden vaihtamista** ja kokeile, mikä muuttuu.



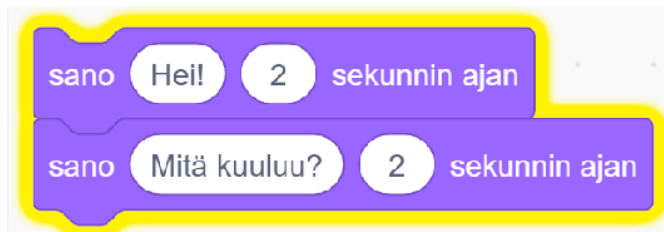
4. Siirry nyt **Ulkonäkö-**valikkoon.

Kytke **kaksi sano (Hei!) (2) sekunnin ajan**-lohkoa.

Vaihda tekstit sellaisiksi kuin haluat.

5. **Klikkaa** skriptiä **kerran**. Voit **vaihtaa numeroa** säätääksesi kauanko kummankin lohkon suorittaminen kestää.

Huomaa: Skriptissä on keltainen kehys kun sitä suoritetaan. Tämän näkee helposti **sano (Hei!) (2) sekunnin ajan**-lohkoilla, koska niitä suoritetaan aina tietty sekuntimäärä.



6. **Rakenna oma skriptisi** ainakin **5 lohkoilla** **Liike-** ja **Ulkonäkö-**valikoista.

Testaa skriptiäsi **klikkaamalla sitä**.

Tässä esimerkkiskripti



Arvojen muuttaminen

Tässä oppilaat kokeilevat lohkoissa esiintyvien arvojen muuttamista.

Askeleet perustuvat näyttämön kokoon (*lisätietoa tämän sivun alareunassa*). **Käännös** lasketaan asteina, välillä 0-360 (tai -180 - 180). Sekä *liiku* että *käänny* ovat **välittömiä komentoja**, eli ne suoritetaan viiveettä ja seuraavaan lohkoon siirrytään myös viiveettä.

Myöhemmin käytetään myös **toistorakenteita**, jotka mahdollistavat Scratchissa hahmon sulavan liikkumisen.

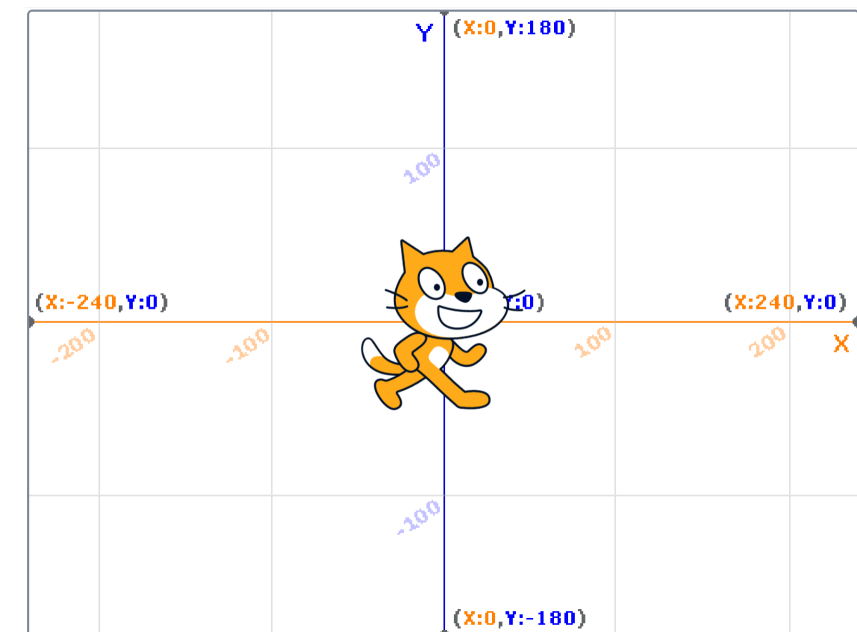
Viiveelliset komennot

Kun **sano () (2) sekunnin ajan** -lohko suoritetaan, seuraavaan lohkoon siirrytään vasta 2 sekunnin jälkeen. Komennoissa on siis **viive**. Myöhemmin esitellään myös **odota**-lohko, jolla saadaan aikaa pelkkää viivettä.

X,Y -koordinaatisto

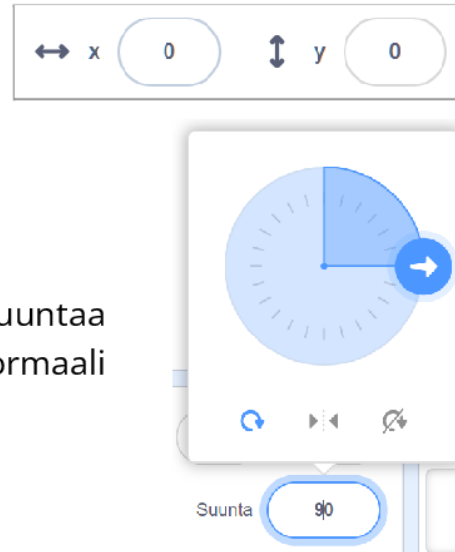
Scratchin **näyttämö** on kaksiulotteinen tila. Näyttämö on 480 pikseliä leveä (x-akseli) ja 360 pikseliä korkea (y-akseli). Esimerkiksi **liiku** komennon askeleet kuvaavat siirryttävien pikselien määrää.

Tässä oppimiskokonaisuudessa ei perehdytä koordinaatistoon, mutta sen voi halutessa esitellä oppilaille.



Vinkkejä:

- Voit siirtää hahmoa raahaamalla sitä esiintymislavalla.
- Jos hahmo liikkuu lavan ulkopuolelle, saat sen takaisin keskelle **vaihtamalla sen sijainnin** nolaksi (x: 0 ja y: 0) **esiintymislavan alapuolella** olevista säädöistä.
- Voit kääntää hahmoa itse säätämällä sen suuntaa **esiintymislavan alapuolella**. 90 on normaali aloitussuunta.



Keskustele

Keskustelkaa ryhmissä vastaukset seuraaviin tehtäviin.

1. Missä järjestyksessä skriptin lohkot suoritetaan?

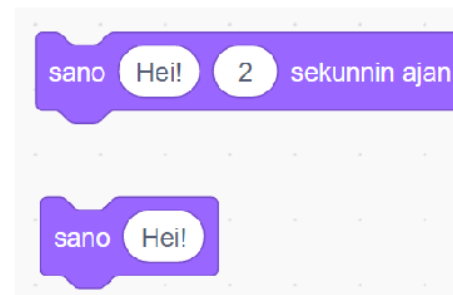
Ympyröi oikea vaihtoehto.

Yksi kerrallaan
ylhäältä alas

Yksi kerrallaan
alhaalta ylös

Kaikki
samanaikaisesti

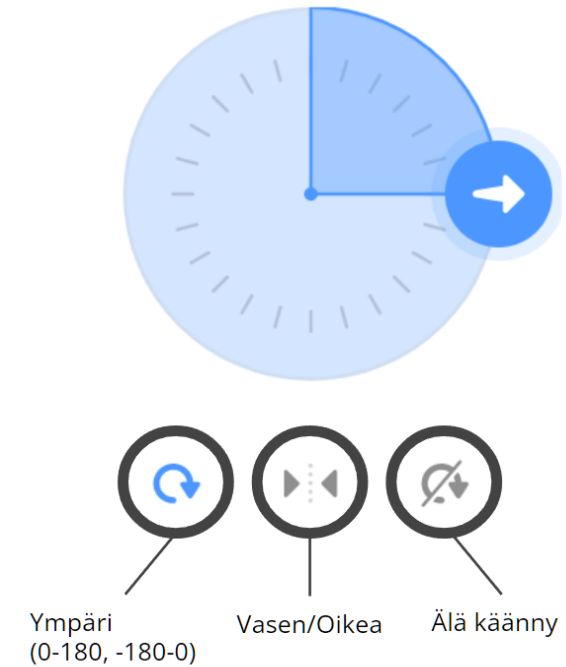
2. Mitä eroa näillä kahdella loholla on?



Eri tavat kääntää hahmoa

Oppilaat saattavat epähuomiossa muuttaa hahmon tapaa kääntyä. Normaali tapa on "Ympäri" (vasemmanpuolimmaisoin).

Joskus on parempi käyttää "Vasen/oikea" -tyyliä, jolla hahmo käännetään peilikuvana ympäri, kun käännöksen arvo on alle 0 (tai välillä 181-359).



Esimerkkiratkaisut tehtäviin

1. Yksi kerrallaan ylhäältä alas

Huomio: Oppilaat saattavat luulla, että lohkot suorituvat välittömästi, kaikki samanaikaisesti, koska lohkot suoritetaan niin nopeaa. Viiveelliset komennot kuitenkin osoittavat, että kaikki ei suoriteta samanaikaisesti vaan järjestyksessä ylhäältä alas.

2. Ensimmäinen laittaa hahmon sanomaan "Hei!" ja pysäyttää skriptin suorittamisen kahden sekunnin ajaksi. Toinen tekee saman ilman, että skriptin suorittaminen pysähtyy.

Huomio: Oppilaat saavat joka tilanteessa kokeilla tehtävissä kysyttäviä asioita vastatessaan tehtäviin. Jos oppilas ei osaa vastata tehtävään, opettaja voi ohjata häntä ottamaan selvää itse.

Tapahtumien käyttö

Ohjelmat eivät oikeasti toimi siten, että skriptejä klikataan. Scratchissa komentojen suorittaminen alkaa **Tapahtumista**.



Tapahtumat

Ohjeet

1. Avaa **Tapahtumat**-valikko. **Raahaa** **kun klikataan** -lohko koodialueelle.

2. **Rakenna uusi skripti tai käytä jotain vanhaa** tämän tapahtuman kanssa. Esimerkki tässä →



3. **Klikkaa vihreää lippua** esiintymislavan yläpuolella. Skriptisi suoritetaan.

Scratchissa **vihreää lippua** käytetään **ohjelman käynnistämiseen**.

Joten **kun klikataan** on paikka, jonne laitamme kaikki lohkot, jotka haluamme suorittaa **ohjelman alussa**.

4. Voit myös **kokeilla muita tapahtumia** skripteissäsi. Esimerkiksi **kun tätä hahmoa klikataan**, joka suorittaa skriptin aina, kun **klikkaat hahmoa**.

Huomaa: Muista **tallentaa ohjelmasi silloin tällöin!** Pyydä opettajalta apua jos et tiedä, miten ohjelma tallennetaan.

Tapahtumien käyttäminen

Tapahtumat ovat olennainen osa visuaalista ohjelmointia. Niiden avulla ohjelmasta tulee oikeasti ohjelma. Scratchissa lohkoja ei suoriteta ilman tapahtumia.

Tallentaminen

Jos oppilaat kirjautuvat Scratch-tunnuksilla, heidän projektit tallentuvat automaattisesti. Oikeassa yläkulmassa on myös "Tallenna nyt" painike tilanteisiin, josta viimeisimmät muutokset tallentuvat, mikäli niitä ei ole automaattisesti tallennettu.

Jos Scratch-tunnukset eivät ole käytössä, ohjelman voi helposti tallentaa suoraan tietokoneelle valitsemalla **Tiedosto->Tallenna tietokoneellesi**.

Version hallinta

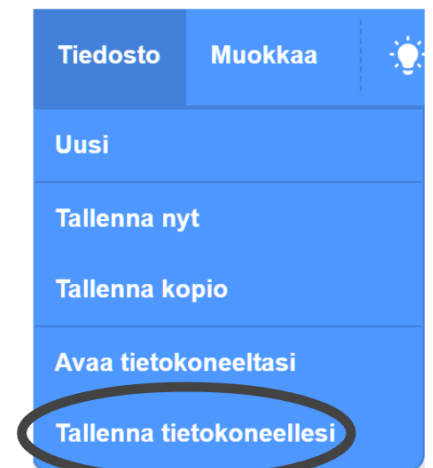
Isoja projekteja tehdessä on hyvä tallentaa eri versioita samasta ohjelmasta niiden kehittämisen matkan varrelta. Tämä auttaa esimerkiksi tilanteissa, joissa oppilas on vahingossa poistanut merkittäviä osia ohjelmaa.

Eri versiot voidaan nimetä esimerkiksi näin:

- "Minun peli 1", "Minun peli 1"... tai
- "Minun peli 0.1", "Minun peli 0.2"...

Kun käytetään Scratch-tunnuksia, eri versioita voidaan tallentaa valitsemalla **Tiedosto->Tallenna kopio**. Tietokoneelle tallentaminen toimii tietenkin myös tunnuksilla.

Tallenna nyt



Testaa ja tutki!

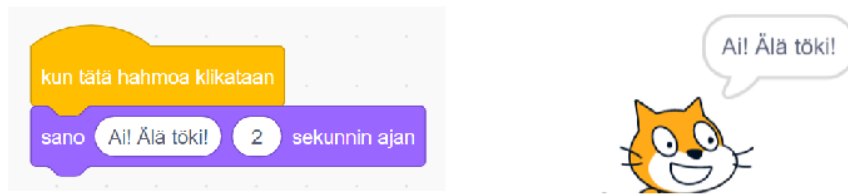
↑ Tämä **Testaa ja tutki!**-merkki tarkoittaa, että saat rohkeasti tutkia, miten asiat toimivat.

On aika aloittaa jonkin ihan oman tekeminen! Tässä vaiheessa **tutkimme miten eri komennot ja tapahtumat toimivat yhdessä**. Voit tehdä jotain hauskaa, jännittävää tai ihan vain kokeilla eri työkaluja.

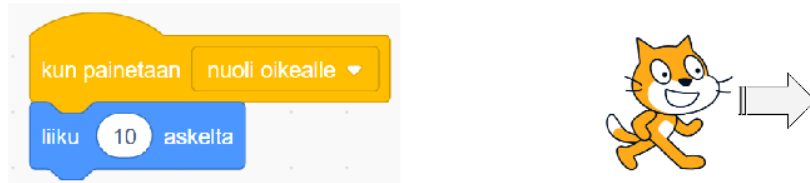
Kun olet valmis, **näytä ohjelmasi luokkatovereillesi** ja katso, **mitä muut ovat saaneet aikaan**.

Tässä on joitakin **esimerkkejä**, joista voit saada ideoita:

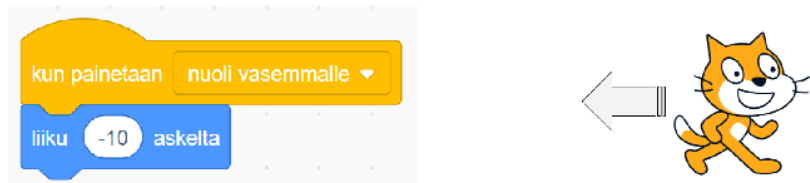
Hahmo **sanoo** jotain kun sitä **klikataan**.



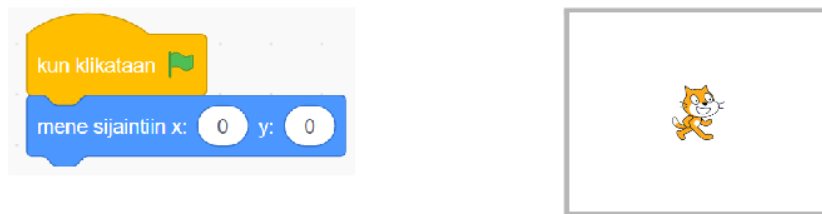
Hahmo **liikkuu oikealle** painamalla **nuolta oikealle**.



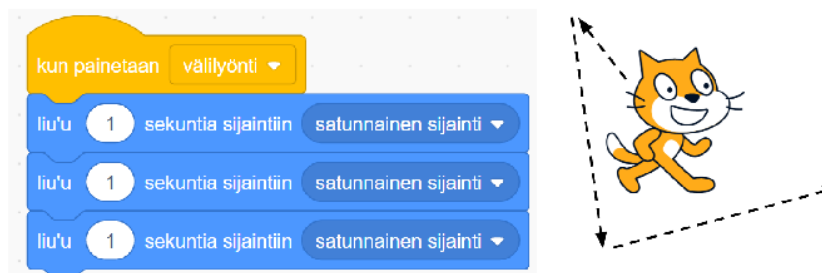
Hahmo **liikkuu vasemmalle** painamalla **vasenta nuolta**.



Hahmo **siirtyy lavan keskelle** ohjelman käynnistyessä.



Hahmo **liukuu satunnaiseen paikkaan** 3 kertaa kun **välilyöntiä painetaan**.

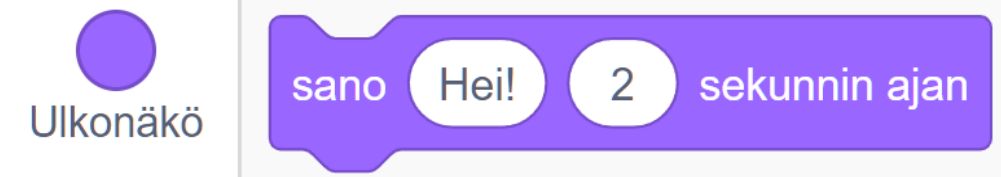


Testaa ja tutki

Ohjelmoinnin oppiminen perustuu hyvin paljon tutkimiseen ja ongelmien kohtaamiseen. Ohjelmointi on työkalu, joka avaa uusia ulottuvuuksia luovuudelle. Siksi oppilaiden tulee päästä kokeilemaan siipiään mahdollisimman usein.

Scratch lohkovalikon värit

Lohkon väri kertoo, mistä valikosta se on haettu.. Esimerkiksi *Ulkonäkö*-valikon kaikki lohkot ovat violetteja.





Lisätietoa eri lohkoista: https://en.scratch-wiki.info/wiki/Block_Categories

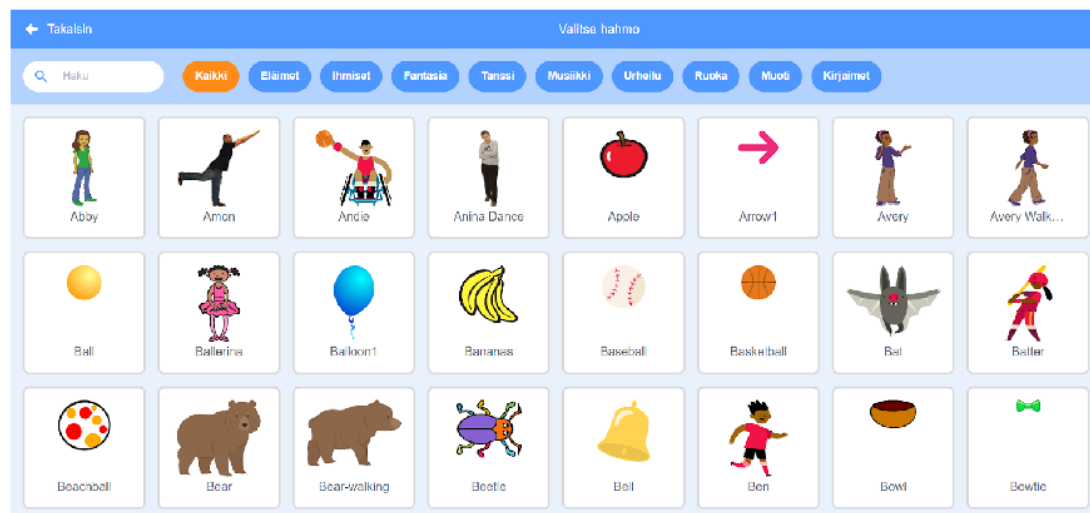
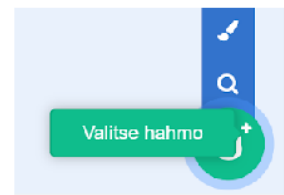
Ohjelmoidaan - Hahmojen lisääminen ja muokkaus

Lisää hahmo kirjastosta

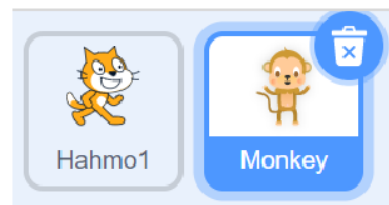
Scratchissa on paljon erilaisia hahmoja sisältävä hahmokirjasto. Aloitetaan sieltä!

Ohjeet

1. Voit lisätä hahmon kun **viet hiiren  -kuvakkeen päälle**. Nyt voimme valita, miten haluamme lisätä hahmon.
2. Hahmokirjasto aukeaa **klikkaamalla  -kuvaketta**.
3. Seuraavaksi **valitse hahmo** jonka haluat tuoda ohjelmaasi.



Näyttämöllä pitäisi nyt olla kaksi hahmoa.



Huomaa: Hahmoilla on oma koodinsa. Kissahahmolla on aiemmin tekemäsi koodi, mutta **uudella hahmolla ei ole lainkaan koodia**.

Ohjelmoidaan - Hahmojen lisääminen ja muokkaus

Ajankäyttö

90min

Työtapa

Työskennellään **1-3** hengen yksiköissä

Avainsanat

Hahmo, odota, piirtäminen, asuste, animaatio, nimeäminen

Kohti monialaista oppimista

Hahmojen muokkaaminen ja piirtäminen ovat pääasiassa kuvankäsittelyä, mutta animoinnin yhteydessä palataan taas ohjelmointiin. Kun opitaan käyttämään ohjelmointiympäristön perusominaisuuksia, avataan mahdollisuudet **monialaiselle oppimiselle**.

Hae hahmo kirjastosta

Scratchissa on hahmokirjasto, jossa on kokoelma valmiiksi tehtyjä hahmoja. Tässä vaiheessa oppilaat valitsevat valmiin hahmon, jota he muokkaavat.

Tämän jälkeen oppilailla tulisi olla kaksi hahmoa projektissaan.

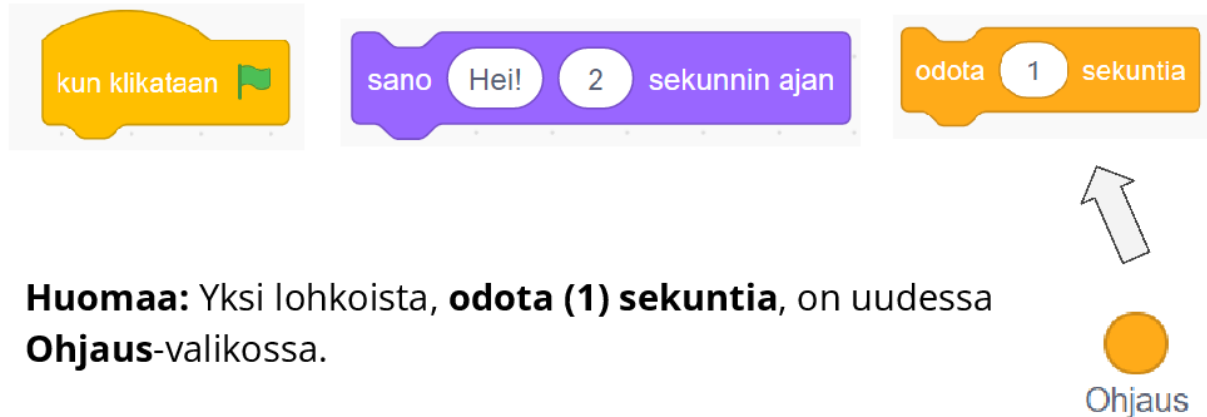


H Haaste!

↑ Tämä H tarkoittaa haastetta, jonka yrität ratkaista itse.

Haaste: Tee ohjelma, jossa **kaksi hahmoa käy lyhyen keskustelun** kun **klikataan**.

Tarvitset näitä lohkoja:

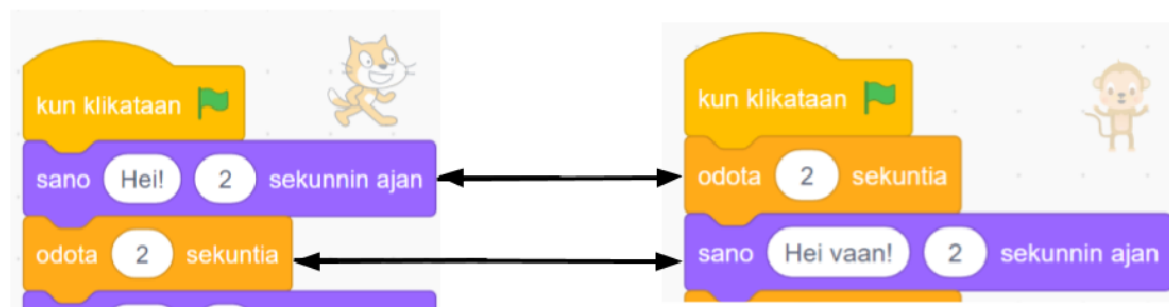


Huomaa: Yksi lohkoista, **odota (1) sekuntia**, on uudessa **Ohjaus**-valikossa.

Vinkkejä haasteeseen:

⇒ Keskustelussa vain yhden hahmon tulisi puhua kerrallaan. Sinun ohjelmassasi yhden **hahmon on odotettava kun toinen puhuu**.

Alla on esimerkki, jossa kissa puhuu 2 sekuntia ja apina odottaa. Tämän jälkeen roolit vaihtuvat.



Sivun 18 haastetehtävä

Opettaja voi demonstroida haastetehtävän tavoitetta seuraavalla ohjelmalla:

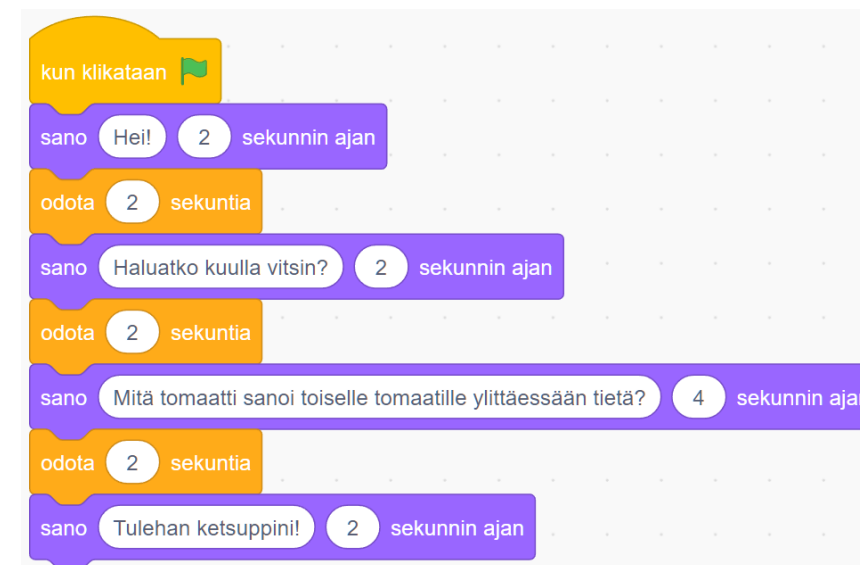
- codeschool.fi/keskustelu

Älä näytä koodia! Näytä vain lopputulos näyttämöllä.

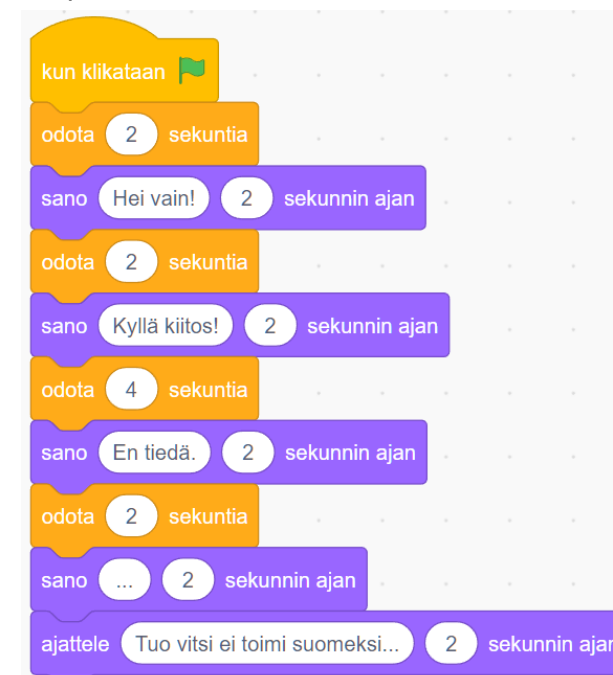
Huomio: *Odota-lohkoa käytetään tässä ensimmäistä kertaa.*

Esimerkkiratkaisut

Skripti Hahmolle 1



Skripti Hahmolle 2

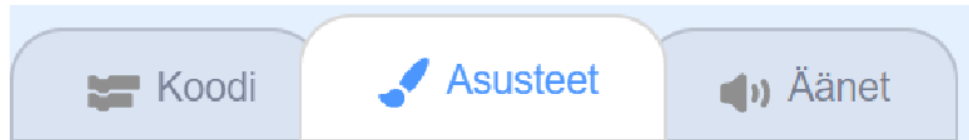


Hahmojen muuntelu ja piirtäminen

Scratchissa on hyvät piirtotyökalut valmiiden hahmojen muunteluun ja omien laatimiseen. Voit vapaasti alkaa muokata jotakin valmiista hahmoista.

Testaa ja tutki!

Joka hahmolla on kolme välilehteä: **Koodi**, **Asusteet** ja **Äänet**. Aloita **valitsemalla muokattava hahmo** ja siirry **Asusteet**-välilehdelle.



Huomaa: Aina kun piirrät jonkin osan Scratchissa, sitä voi myöhemmin **liikuttaa ja muokata**.

Voit nyt aloittaa hahmon muuntelun! Alla ovat eri piirtotyökalujen ohjeet ja esimerkkikuvat:

Valitse: Valitse, liikuta, poista ja säädä osan kokoa.

Muotoile uudelleen: Muuntelee osia.

Sivellin: Vapaata piirtämistä.

Pyyhekumi: Poistaa osan osasta.

Hahmojen piirtäminen ja muokkaaminen

Oppilaat saavat jälleen kokeilla työkalujen käyttämistä vapaasti.

Bittikartta vs vektori

Piirtoalueen alapuolella on painike **“Muunna bittikartaksi”**. Sitä klikatessa hahmo muuttuu yksinkertaiseksi piirroksiksi, jossa eri objekteihin ei enää pääse käsiksi. Tämä vaikeuttaa esimerkiksi hahmon animointia huomattavasti. Lisäksi hahmon tarkkuus laskee ja siitä tulee pikselinen. **Vältele tätä painiketta** ellei sille tule erityistä tarvetta.



Jos oppilas klikkaa sitä, muutoksen voi peruuttaa **peruutus**-painikkeella tai näppäinyhdistelmällä **ctrl+z**



Vektoripiirros



Bittikarttapiirros



Täytä: Värjää koko osan uudella värillä.



Teksti: Kirjoittaa tekstiä.



Viiva: Piirtää viivan



Ympyrä: Piirtää ympyrän tai ellipsin.



Suorakulmio: Tekee neliön tai suorakulmion.



Piirtotyökalujen yhdistely:

Voit vaikkapa tehdä hienon hatun hahmollesi yhdistämällä **ympyrän** ja **suorakulmion** joita sitten muovaillet **muotoile uudelleen** -työkalulla. Hatun väri on helppo vaihtaa **täytä**-työkalulla.



Osien poistaminen:

Osien poistaminen on helpointa klikkaamalla niitä **valitse**-työkalulla ja painamalla **Poista**-näppäintä tai -kuvaketta.

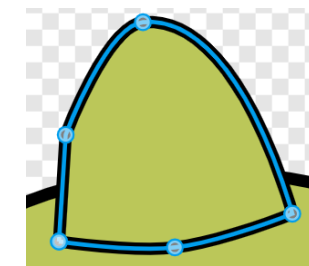
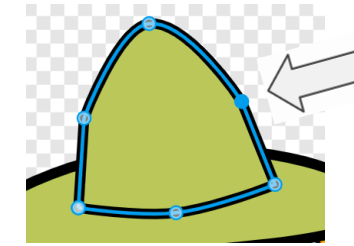


Voit aina peruuttaa poiston painamalla **peru**.

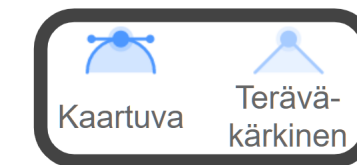
Muotoilutyökalu

Muotoile uudelleen-työkalulla saadaan esille objektien mumotoilupisteet. Niiden liikuttamisen lisäksi voit tehdä seuraavaa:

- **Poistaa** pisteen kaksoisklikkauksella



- Valita, miten piste muuttaa muotoa (kaartuva vai terävä)



Asusteiden ja animaatioiden tekeminen

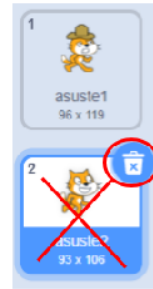
Kun olet saanut hahmosi ulkoasun kuntoon, on aika tehdä hahmolle uusi asuste!

Ohjeet

Tehdään kaksi asustetta, jotka ovat vain hieman erilaisia, jotta voimme tehdä **yksinkertaisen animaation**.

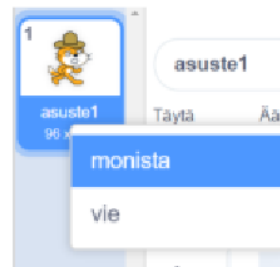
1. **Piirtoalueen vasemmalla laidalla** näet hahmon eri asusteet. Tässä vaiheessa niitä pitäisi olla kaksi.

Voit **poistaa tarpeettomat asusteet** painamalla roskakori-kuvaketta.



2. Tarvitsemme animaatioomme **kaksi versiota asusteestamme**: Yhdessä silmät avoinna ja toisessa silmät suljettuina.

Aloitetaan tekemällä ensimmäisestä asusteesta kopio **klikkaamalla sitä hiiren oikealla napilla** ja valitsemalla **monista**.



Nyt meillä on kaksi samanlaista asustetta. Muokataan toista, jotta siinä hahmon silmät ovat kiinni.

3. **Valitse hahmon pää**. Päässä on monia osia, joita emme voi muokata yksittäin. Osat on **ryhmitetty**.



Voit irrottaa osat painamalla **pura ryhmitys** kun pää on valittuna.

Tämän jälkeen voit muokata silmiä.



4. Voimme saada hahmon sulkemaan silmänsä monella tavalla. Voit vaikkapa **litistää** silmiä tai poistaa ne ja korvata ne **viivoilla**. Tee kuten haluat.

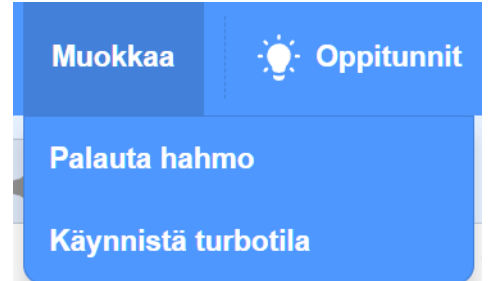


Tärkeää poistamisesta ja perumisesta

Oppilaille esitellään tässä osiossa objektien poistaminen. Se tapahtuu klikkaamalla yksittäistä objektia ja valitsemalla **Poista** (roskakori). Jos objektia ei valita vaan klikataan pelkkää poistoa, kaikki objektit poistetaan. Tämän voi perua klikkaamalla **peru**.



Myös kokonaisia hahmoja ja asusteita voidaan palauttaa valitsemalla yläpalkista **Muokkaus->Palauta**. Tämä kuitenkin palauttaa vain edellisen poistetun asian.

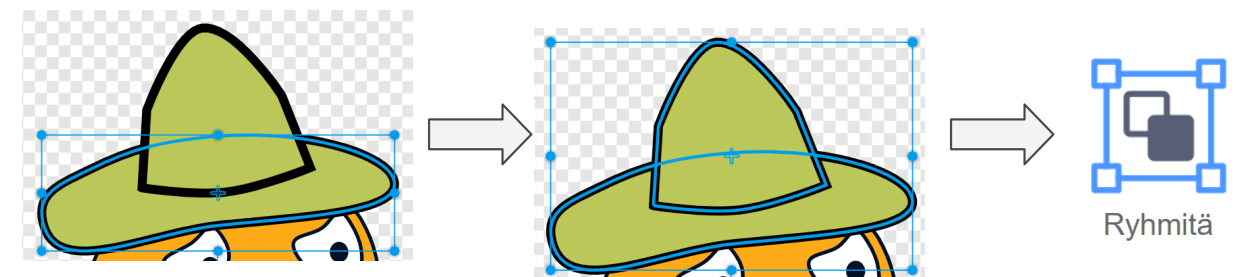


Animointi

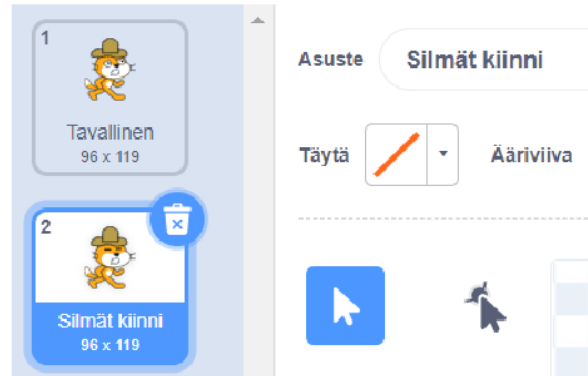
Sovellukset ja pelit voivat olla toiminnaltaan loistavia, mutta tuntua elottomilta, mikäli niissä ei ole animaatioita. Käytännössä animaatio muodostuu peräkkäin näytettävistä kuvista, jotka ovat hieman erilaisia. On hyvä aloittaa animointi mahdollisimman yksinkertaisesta animaatiosta.

Ryhmittäminen ja ryhmityksen purkaminen

Ryhmitys-työkalu tekee kokonaisuuksien liikuttelusta helpompaa. Esimerkiksi jos haluamme tehdä hatun, joka muodostuu kahdesta objektista, objektit valitaan hiirellä maalaamalla tai SHIFT-näppäintä pohjassa pitämällä ja valitaan **Ryhmitä**. Nyt objekteja ei voi siirtää yksinään vaan ne kulkevat ryhmässä, kunnes valitaan **Pura ryhmitys**.



5. Lopuksi **anna kuvaavat nimet kullekin asusteelle**. Esimerkiksi "Tavallinen" ja "Silmät kiinni".



Nyt on aika tehdä animaatio! Se on samalla hauska ohjelmointihaaste!

H Haaste!

Haaste 1: Ohjelmoi hahmo **sulkemaan silmänsä** ja **sanomaan** jotain kun **sitä klikataan**. Suljettuaan silmänsä hahmon ulkoasun tulisi palata tavalliseksi.



Huomaa: Seuraavalla sivulla on **toinen haaste** ja **vinkkejä**. →

Kuvaava nimeäminen

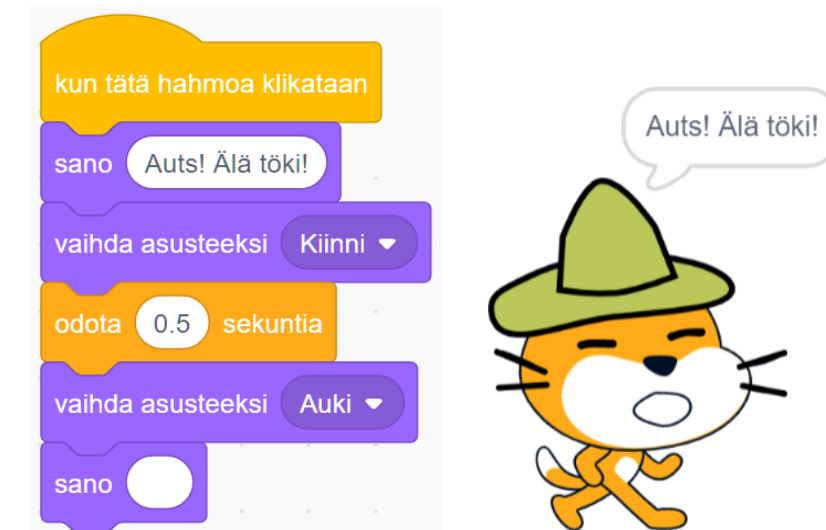
Hyvät ohjelmointikäytännöt ovat erityisen tärkeitä sekä ammattilaisille että vasta-alkajille. Yksi hyvä käytäntö on **kuvaava nimeäminen**. Esimerkiksi hahmojen eri asusteet voidaan nimetä kuvaamaan sen tarkoitusta; Jos hahmolla on silmät kiinni, asusteen nimi voi olla "räpäytys" tai "silmät kiinni". Tämä on tärkeää erityisesti seuraavista syistä:

- Kun oppilas palaa tekemänsä ohjelman pariin myöhemmin, hänen on helpompi jatkaa kun asiat on nimetty johdonmukaisesti.
- Muiden on helpompi lukea oppilaan koodia. Opettajan ja vertaisten on helpompi auttaa oppilasta.

Opettajan kannattaa vaatia oppilailta kuvaavaa nimeämistä.

Esimerkkiratkaisut - Sivujen 22-23 haastetehtävät

Haaste 1:



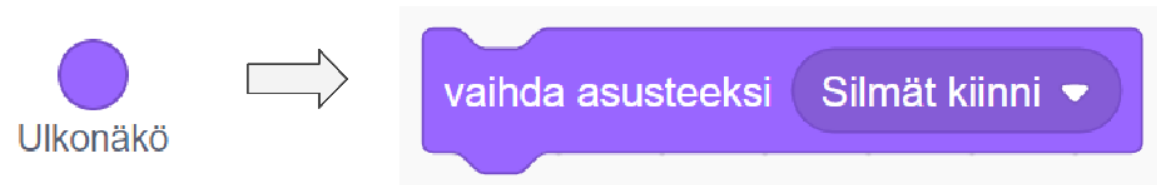
Haaste 2:



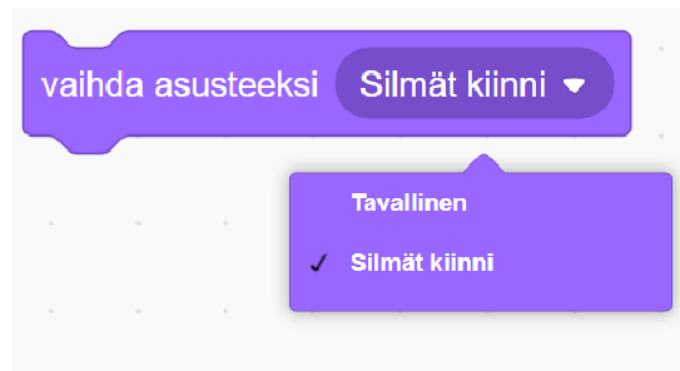
Haaste 2: Ohjelmoi **toinen** hahmo **vaihtamaan ilmettään** ja **sanomaan** jotain kun sitä klikataan.

💡 Vinkkejä haasteisiin:

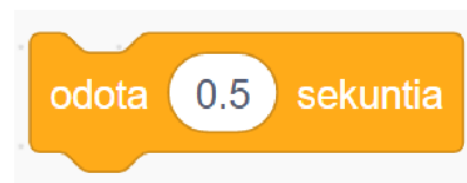
- ⇒ Vaihda asusteita **Ulkonäkö**-valikon **vaihda asusteeksi ()** -lohkolla.



- ⇒ Voit päättää, **mikä asuste** tulee näkyväksi **valitsemalla sen lohkoissa**.



- ⇒ Voit lisätä **odota**-lohkon myös alle yhdeksi sekunniksi. Esimerkiksi **0.5** tarkoittaa puolta sekuntia. Käytä pistettä desimaalipilkun sijaan!



Pisteitä pilkkujen sijaan

Ohjelmoinnissa kokonaisluvut erotetaan desimaaleista pisteellä. Tämä johtuu siitä, että tekstuaalisessa ohjelmoinnissa pilkkua käytetään kokonaisuuksien erottamiseen esimerkiksi listoissa.

Esimerkiksi Pythonilla voidaan tehdä lista, jonka yksittäiset alkiot ovat desimaalilukuja:



```
lista_luvuista = [0.5, 2.3, 5.2]
```

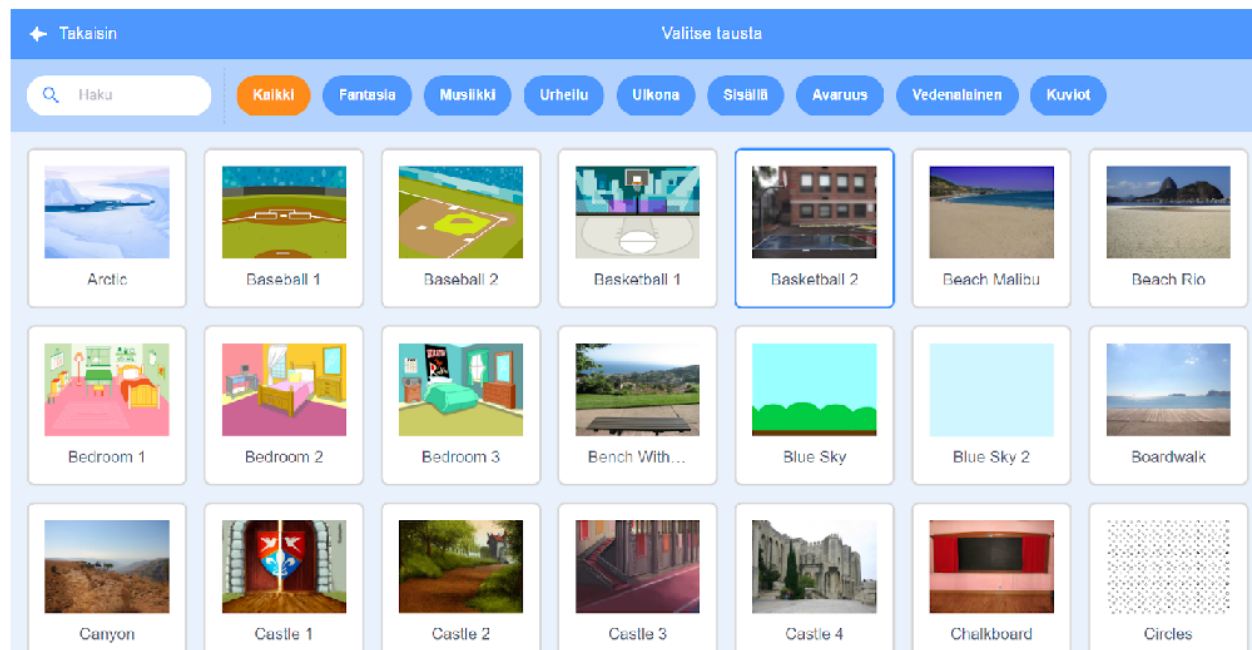
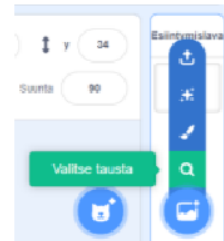
Lisää esiintymislavalle tausta

Nyt ohjelmamme tekee jotain yksinkertaista, mutta tausta on pelkkää valkoista.

Lisätään lavalle tausta!

Testaa ja tutki!

1. Voit lisätä taustan siirtämällä **hiiren**  -kuvakkeen päälle. Se on hahmo-kuvakkeen vieressä. Nyt voimme päättää, miten lisäämme taustan.
2. Taustakirjasto aukeaa **klikkaamalla**  -kuvaketta.
4. Voit nyt **valita taustan** jonka haluat tuoda ohjelmaasi.



Kun olet tehnyt valinnan, tausta vaihtuu.

Lisätään tausta

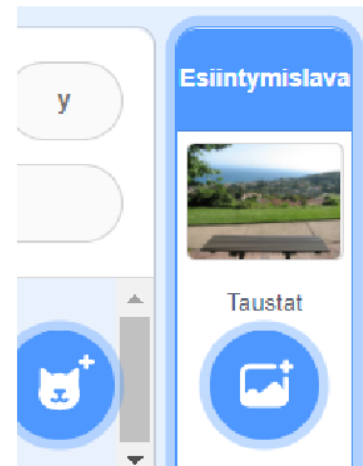
Hahmojen lisäksi Scratchissa käsitellään **näyttämöä**, jolle voidaan antaa asuste kuten hahmolle, mutta näyttämön asustetta kutsutaan **taustaksi**.

Näyttämöllä on oma koodialueensa kuten hahmoillakin, mutta se voidaan jättää halutessa tyhjäksi. Näyttämöltä puuttuu muun muassa *Liike*-valikon komennot, koska taustat eivät voi liikkua.

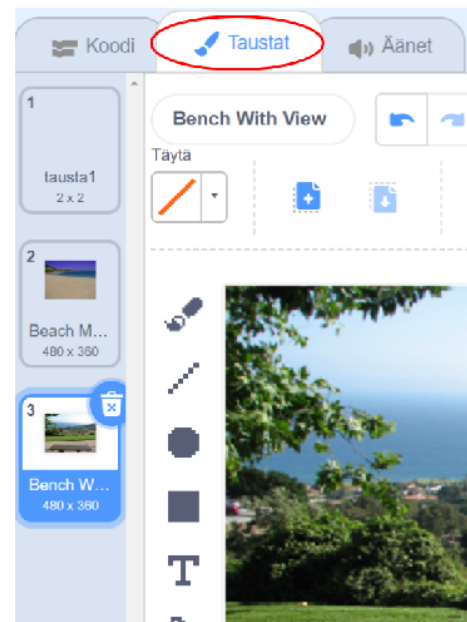
Hahmot eivät voi suoraan käskää toisiaan, mutta taustoja ne voivat käskää; Hahmoilla on *Ulkonäkö*-valikossa komento **vaihda taustaksi** ().

💡 Vinkkejä taustojen kanssa toimimiseen:

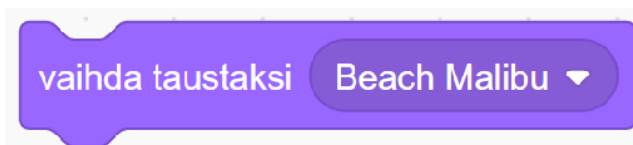
- ⇒ Jos valitset **Esiintymislavan** ruudun oikeasta alareunasta, voit **muokata sen koodia ja taustoja** aivan kuten hahmojen koodia ja asusteita.



- ⇒ Jos lisää kirjastosta useita taustoja, ne kaikki löytyvät **Taustat**-välilehdeltä. Sieltä voit valita, mitä taustaa haluat käyttää.

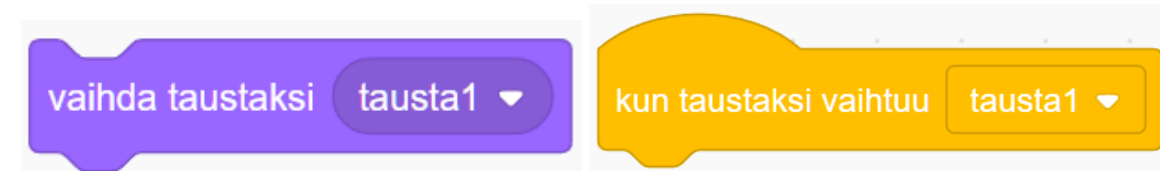


- ⇒ Hahmot eivät voi yleensä suoraan komentaa toisia hahmoja. Tähän on yksi poikkeus: **Kaikki hahmot voivat vaihtaa taustaa** loholla **vaihda taustaksi ()**.



Taustan vaihdot kommunikoinnissa

Hahmot eivät voi suoraan käskää toisiaan, mutta taustoja ne voivat käskää; Hahmoilla on *Ulkonäkö*-valikossa komento **vaihda taustaksi ()**. *Tapahtumat*-valikosta löytyy myös tähän liittyvä tapahtuma. Jos yksi hahmo vaihtaa taustan, muut hahmot voivat reagoida siihen, kun tausta vaihtuu. Näin hahmot voivat kommunikoida taustojen vaihdoilla.



Suorempi tapa kommunikoinnille on **viestit**, mutta ne esitellään tässä oppimiskokonaisuudessa vasta myöhemmin ylöspäin eriyttävänä sisältönä (sivulla 69).

Luetaan - Tietokoneet ovat hyviä toistamaan

...mene eteenpäin, mene eteenpäin, mene eteenpäin, mene eteenpäin, mene eteenpäin...



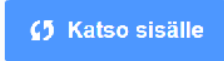
Kun haluamme tehdä saman toiminnon monta kertaa, voimme toistaa sitä.

Toistaminen on kaikessa ohjelmoinnissa yksi tärkeimmistä asioista.

Toistoja kutsutaan usein **loopeiksi tai silmukoiksi**.

Testaa ja tutki!

Työskentele parin kanssa ja tee seuraavat tehtävät.

1. a) Avatkaa selaimessa tämä osoite: <https://codeschool.fi/ea1>
b) Klikatkaa **Katso sisälle** -kuvaketta.

2. Älkää vielä painako mitään näppäimistön näppäimiä! Katsokaa koodia ja keskustelkaa parin kanssa, mitä luulette tapahtuvan jos painatte...
 - a) **A-näppäintä?**
 - b) **B-näppäintä?**
 - c) **C-näppäintä?**
3. Kokeilkaa nyt **painaa A, B tai C yksi kerrallaan**. Ennen seuraavan näppäimien painamista palauttaa ohjelma alkutilaansa painamalla **vihreää lippua**.

Ohjelmoidaan - Hahmojen lisääminen ja muokkaus

Ajankäyttö

20 min

Työtapa

Työskennellään **2-3** hengen yksiköissä, lukien ja keskustellen

Avainsanat

Toistorakenne, toisto, luuppi, määrätty, ikuinen, ehdollinen, kunnes

Silmukoiden testaus

Lisätehtävä: Kun oppilaat ovat tehneet harjoitukset 1-3, opettaja voi ohjata heitä tekemään muutoksia silmukoihin muuttamalla niiden sisällä olevia komentoja ja tutkimalla sen vaikutuksia.

Olet juuri kokeillut **kolmea erilaista toistorakennetta: määrättyä, ehdollista ja ikuinen toistoa**. Tässä on niistä lyhyet selitykset:

Määrätty toisto...



Toistaa komentolohkoja **niin monta kertaa** kuin toistossa määrätään.



Ehdollinen toisto...



Toistaa komentolohkoja kunnes **ehto muuttuu todeksi**.



Ikuinen toisto...

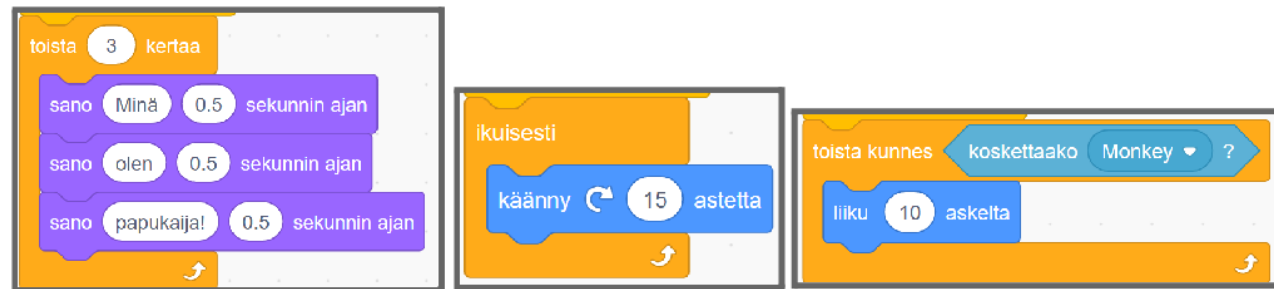


Toistaa komentolohkoja **loputtomasti**.



Vielä yksi tehtävä! Kirjoita vastaukset.

4. Kirjoita näiden erilaisten toistojen oikeat nimet:

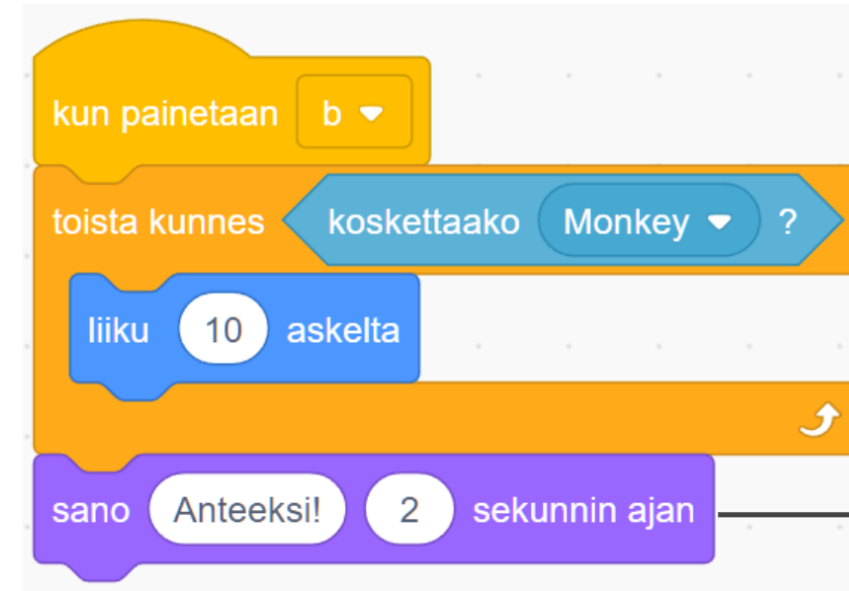


a) _____ b) _____ c) _____

Ehdollisen toiston toiminta

Papukaija-ohjelmassa B-näppäimestä papukaija-hahmo liikkuu, kunnes se osuu apinaan. Opettajan kannattaa painottaa oppilaille, kuinka skriptin suoritus on "jumissa" toistorakenteessa toistorakenteessa esiintyvän tekstin kuvaamalla tavalla. Esimerkiksi kunnes on toistettu 10 kertaa, kunnes ehto on toteutunut tai ikuisesti.

Linkki ohjelmaan: codeschool.fi/toistavapapukaija



Voidaan ajatella, että skriptin suoritus on "jumissa" tässä, kunnes ehto toteutuu.

Tämä suoritetaan vasta ehdon toteuduttua.

Esimerkkivastaus

4. a) Määrätty toisto
- b) Ikuinen toisto
- c) Ehdollinen toisto

Ohjelmoidaan - Ongelmanratkaisua toistolla

Aloitetaan toistojen käyttö testaamalla niitä **piirto-ohjelmassa!**

Ohjeet

1. Mene selaimella tähän osoitteeseen: <https://codeschool.fi/ea2> ja
2. Klikkaa **Katso sisälle** -nappia.

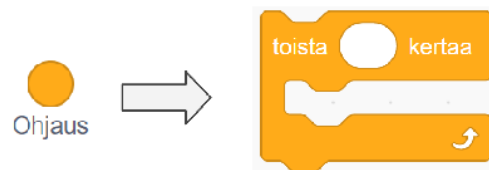
Katso sisälle

3. Tässä ohjelmassa hahmo **piirtää viivan** ja **kääntyy** aina kun **painat välilyöntiä**. Voit aloittaa alusta painamalla vihreää lippua.

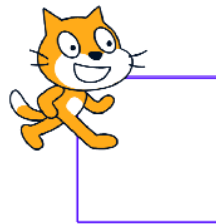
Määrätyn toiston haasteet

H Haaste!

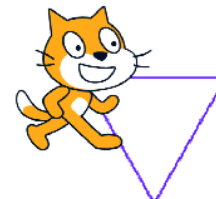
Haasteissa 1-3 käytät **Ohjaus**-valikon **toista ()** -lohkoa.



Haaste 1: Ohjelmoi kissa **piirtämään neliö yhdellä näppäimen painalluksella** (vaikkapa välilyönnillä). Käytä **toista ()** -lohkoa.



Haaste 2: Ohjelmoi kissa piirtämään **kolmio** yhdellä näppäimen painalluksella. Käytä **toista ()** -lohkoa.



Haaste 3: Ohjelmoi kissa piirtämään **ympyrä** yhdellä näppäimen painalluksella. Käytä **toista ()** -lohkoa.



Ohjelmoidaan - Ratkaistaan ongelmia toistolla

Ajankäyttö

70 min

Työtapa

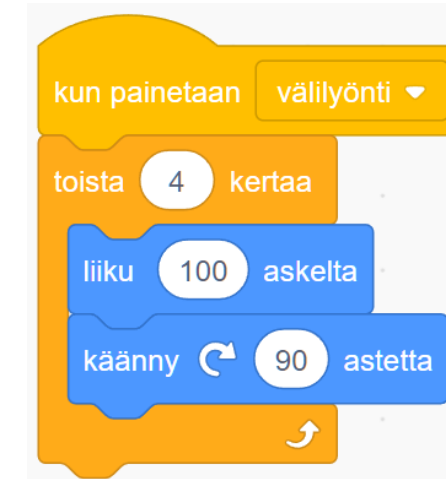
Työskennellään **1-3** hengen yksiköissä

Avainsanat

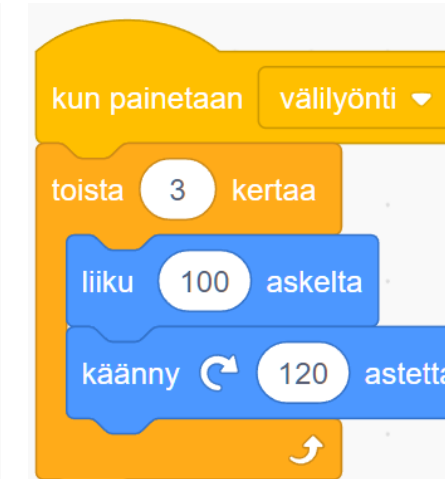
Toisto, silmukka, määrätty, ikuinen, kunnes, monikulmio, ympyrä, kynä, satunnainen

Esimerkkiratkaisut - Sivun 28 haastetehtävät

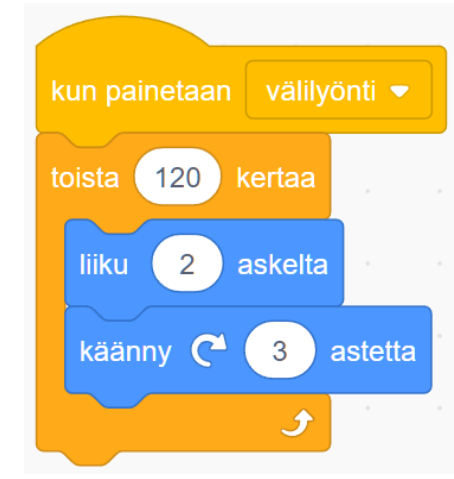
Haaste 1:



Haaste 2:



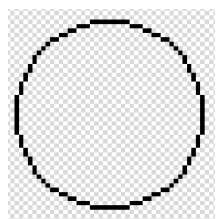
Haaste 3:



Huomio: Periaatteessa oikean ympyrän piirtäminen on mahdotonta.

Esimerkkiratkaisun kuvio on 120-kulmainen monikulmio. Mitä enemmän monikulmiossa on kulmia, sitä enemmän se muistuttaa ympyrää.

Toistojen määrän ja käänösasteiden tulo on oltava 360, jotta saadaan aikaan suljettuja geometrisiä kuvioita.



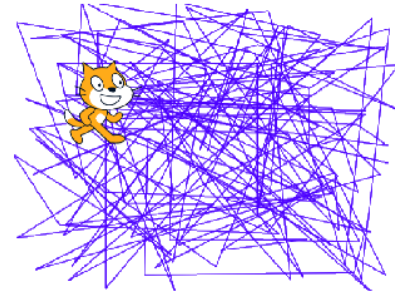
Jatkuvan toiston haasteet

H Haaste!

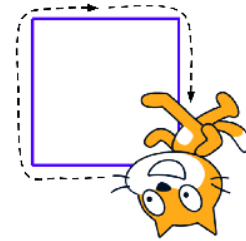
Haasteissa 4-6 tarvitset **Ohjaus**-valikon **ikuisesti** -lohkoa.



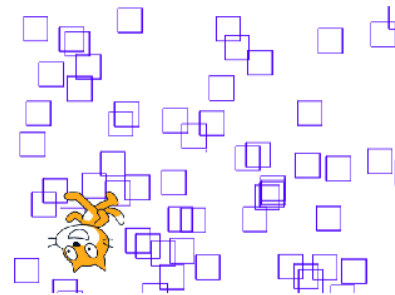
Haaste 4: Ohjelmoi kissa piirtämään **satunnaisia viivoja** loputtomasti näppäimen painalluksella.



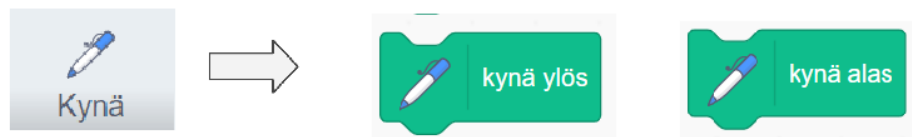
Haaste 5: Ohjelmoi kissa **piirtämään neliötä samaan paikkaan loputtomasti** näppäimen painalluksella.



Haaste 6: Ohjelmoi kissa **piirtämään neliötä satunnaisiin paikkoihin loputtomasti** näppäimen painalluksella.



Huomaa: Haasteessa 6 tarvitset **Kynä**-valikon **kynä alas** ja **kynä ylös** -lohkoja.

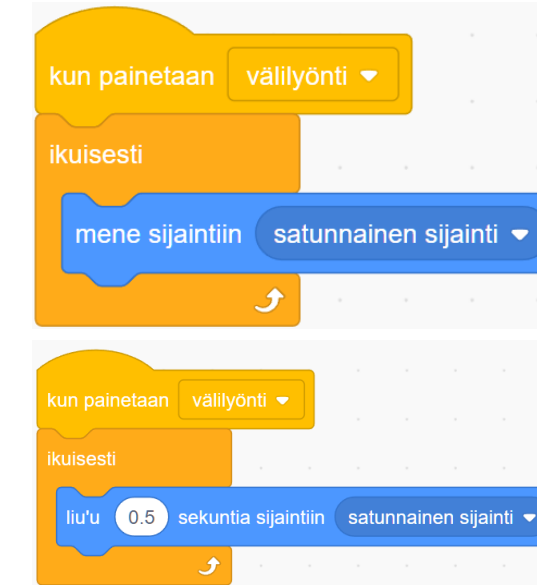


Haasteen 5 esittely oppilaille

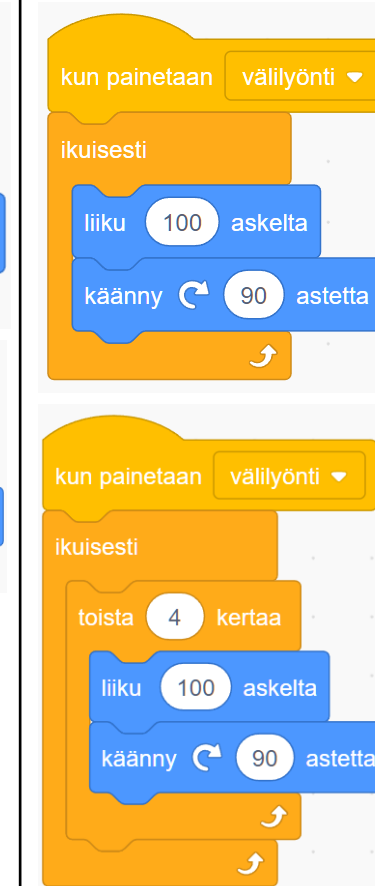
Opettajan on hyvä esitellä haasteen 5 lopputulos näyttämällä, ilman koodin näyttämistä, jotta oppilaat ymmärtävät tavoitteen: Kissa on vain tarkoitus saada piirtämään neliötä yhteen paikkaan loputtomasti. Haasteessa 6 tarkoituksena on yhdistää haasteen 4 ja 5 ideat yhdeksi koodiksi.

Esimerkkiratkaisut - Sivun 29 haastetehtävät

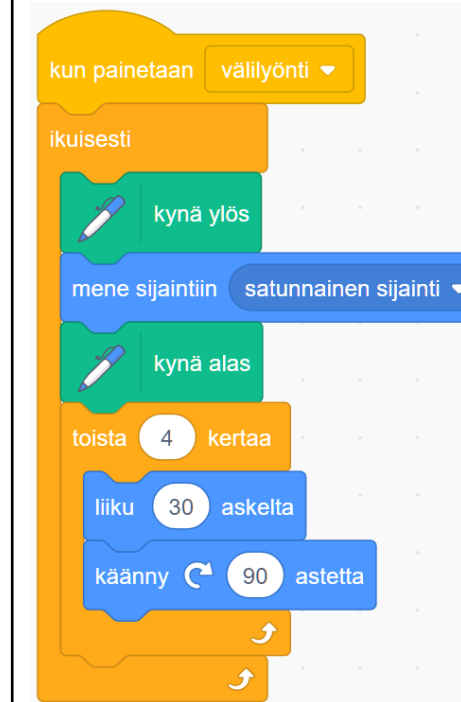
Haaste 4 (2 esimerkkiä)



Haaste 5 (2 esimerkkiä)



Haaste 6

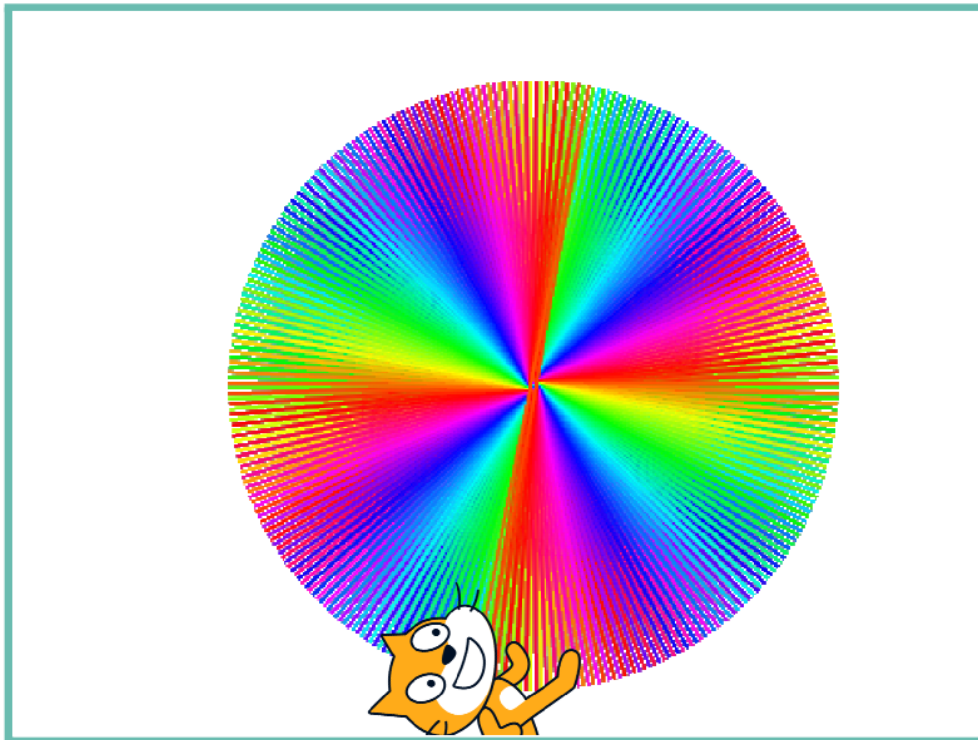


Testaa ja tutki!

Pystyt nyt tekemään omia piirustuksiasi **toistoilla** ja **Kynä**-valikon lohkoilla!

Vaatimukset:

- Käytä **toistoja**
- Käytä **useita värejä**
- Piirustus pyyhitään **vihreällä lipulla**
- Piirtäminen alkaa **näppäimen painalluksella**
- Piirustus **ei ole sattumanvarainen**: Hahmo piirtää saman kuvion joka kerralla.
- **Hauskoja piirustushetkiä!**

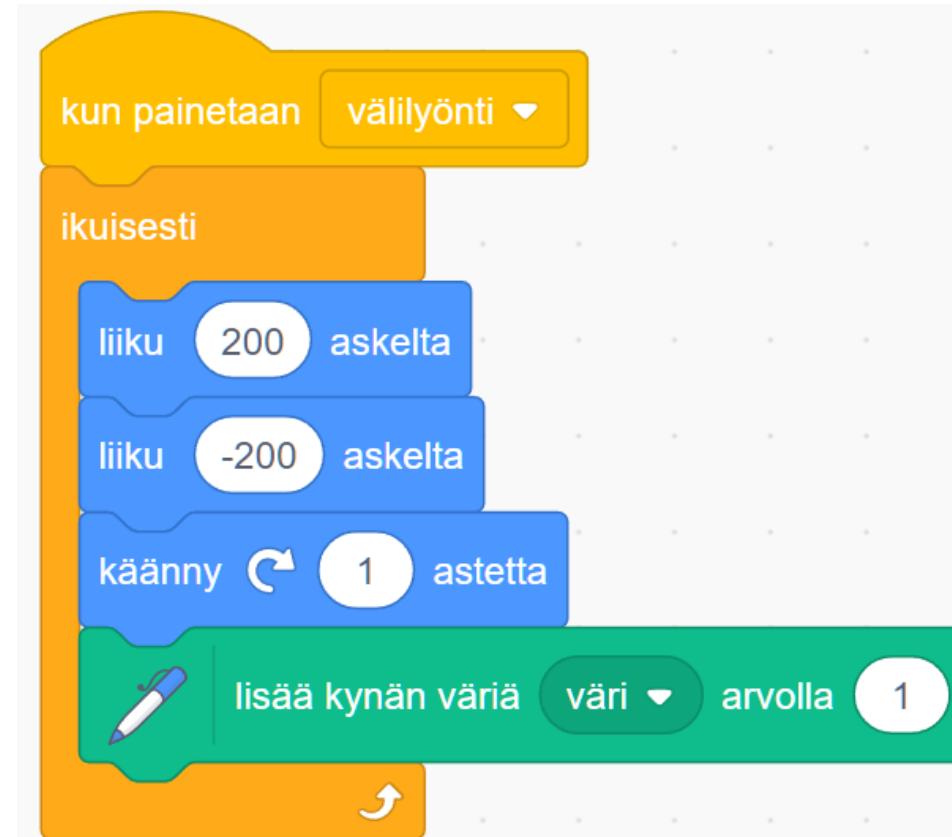


Tämä on vain esimerkki. Sinun piirustuksesi voi olla mitä vain haluat!

Kokeilua toistolla ja kynällä

Viimeisenä tehtävänä oppilaat saavat tehdä vapaamuotoisen miniprojektin. Tämä harjoitus sopii hyvin kotitehtäväksi.

Esimerkkiohjelma (sivun 30 esimerkki)



Projekti - Interaktiivinen tarina

Nyt on aika luoda jotain ihan omaa: Oma Interaktiivinen tarina!

Projektin vaatimukset:

Käytä **vähintään kahta taustaa**

Käytä **vähintään kahta hahmoa**

Ohjelmoi **vähintään kahden hahmon välinen keskustelu**

Tee **ainakin yksi animaatio**

Ohjelmalla pitää olla **alku ja loppu**

Ohjelma alkaa alusta kun **vihreää lippua painetaan**

Tässä on joitakin esimerkkiprojekteja, joita voit tutkia ja kenties saada ideoita omiin projekteihinne:

- Kaksi apinaa aavikolla: <https://codeschool.fi/ea3>
- Koira ja pallo: <https://codeschool.fi/ea4>
- Bobon loputon matka: <https://codeschool.fi/ea5>

Seuraavaksi suunnittelet oman tarina!

Suunnittelupohja on seuraavalla sivulla →

Projekti - Interaktiivinen tarina

Nyt kun oppilaat ovat harjoitelleet Scratchin käyttöä ja ohjelmoinnin perus työkaluja, he pääsevät tekemään isomman projektin. Opettaja voi päättää projektille tiettyyn oppiaineeseen tai ajankohtaiseen aiheeseen sopivan teeman tai teema voidaan päättää yhdessä oppilaiden kanssa.

Ajankäyttö

4-6 oppituntia

Työtapa

Työskennellään **1-3** hengen yksiköissä

Avainsanat

Projekti, koodi, ohjelma, skripti, animaatio, resetointi, hahmo, tausta, interaktiivinen, tarina, peräkkäisyys, toisto.

Lisää tavoitteita

Oppilaan kirjan sivulla esitetyt projektin vaatimukset ovat vähimmäisvaatimukset projektille. Projekti ei ole kuitenkaan valmis, kun pelkät vaatimukset on saavutettu; Luovuus ja oppilaan omat tavoitteet ovat vähintään yhtä tärkeässä roolissa.

Oppimistavoitteiden perusteella opettaja voi myös lisätä listaan lisää vaatimuksia. Tässä esimerkkejä:

- Käytä ohjelmassasi **vähintään yhtä määrättyä ja yhtä loputonta** toistoa
- Ohjelmoi hahmot **tekemään eri asioita eri paikoissa** (taustojen vaihtuessa)
- Tee tarinalle **2 erilaista loppua**
- Käytä ohjelmassasi **ehdollista toistoa**
- Tee **toinen tarina uuteen projektiin**

Suunnitelma

Ryhmän jäsenten nimet:

Kuvaile tarina tässä:

Tarvitsemme apua ohjelmoinnin kanssa

Kyllä

Ei

Ehkä

Tälle alueelle voitte piirtää ja kirjoittaa hahmotelmia tarinastanne:

Valmistaudu esittelemään suunnitelmasi opettajalle ja muille oppilaille!

Tehdään suunnitelma

Tällä sivulla oppilaat suunnittelevat tarinan suurpiirteisesti. Oppilaat saavat myös käyttää Scratchia suunnitellessaan tarinaa, tutkiakseen mahdollisuuksia. Suunnitelmat esitellään pienissä ryhmissä tai koko luokan kesken. Suunnitteluun, tutkimiseen ja esityksen valmisteluun on hyvä käyttää reilusti aikaa (30-40 min).

Suunnitelma on alustava ja siitä saa poiketa projektia tehdessä. Uudet ideat ja inspiraatio syttyvät usein vasta, kun on päästy vauhtiin.

Vertaisarviointi: Miten palautetta annetaan?

Koska oppilaiden tulisi kommentoida toistensa projektisuunnitelmia, on tärkeää miettiä palautteen antamista: Millainen palaute auttaa toista? Tässä kaksi harjoitusta, joilla voidaan pohtia palautteen antamista ja tehdä siitä monimuotoista.

1. Kaikkihan tykkäävät positiivisesta palautteesta, eikö? Miettikää ryhmissä, millainen positiivinen palaute ei juurikaan auta toista. Jokainen ryhmä miettii omasta elämästään esimerkin tai hypoteettisen esimerkin, jossa positiivinen palaute ei auttanut. Tilanteet voidaan näytellä tai kertoa.
2. Projektityössä palautetta voidaan antaa ja saada eri muodoissa. Käyttäkää palautteen antamiseen suoran suullisen palautteen sijaan jotain seuraavista vaihtoehdoista:
 - Video
 - Kuva
 - Ääninauha
 - Piirros
 - Runo
 - Sähköpostiviesti

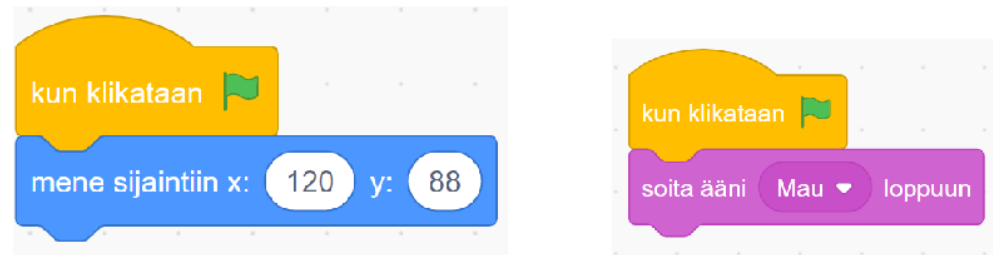
Voit nyt aloittaa tekemisen ja ohjelmoinnin!

💡 Vinkkejä tarinan tekemiseen

Seuraavilla sivuilla on **vinkkejä** jotka auttavat teitä lisäämään **kiinnostavia ominaisuuksia projektiinne**. Käyttäkää niitä jos haluatte!

Rinnakkaiset skriptit

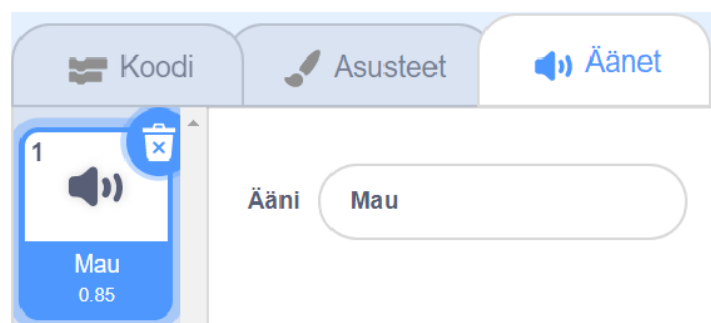
Voit tehdä **rinnakkaisia skriptejä!** Jos käytät **samaa tapahtumaa kahdesti**, molemmat suoritetaan **samaan aikaan**. Voit vaikka käynnistää taustamusiikin yhdessä skriptissä ja toisessa muut samaan aikaan käynnistettävät asiat.



Äänien käyttö

Hahmoilla voi olla omia ääniä aivan kuten omaa koodia tai asusteita.

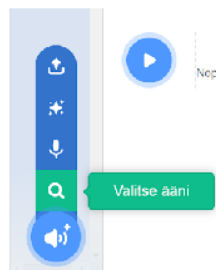
1. Uusia ääniä voi tehdä **Äänet**-välilehdellä.



2. Lisää uusi ääni viemällä **hiiri -kuvakkeen** päälle. Kuvake on vasemmassa alanurkassa.

3. Avaa äänikirjasto **🔍 -kuvakkeesta**.

- 4.



Projektityö alkaa!

Suunnitelmien esittelyn jälkeen oppilaat aloittavat projektit. Ajankäytön suunnittelun vuoksi, oppilaiden on hyvä tietää projektin takaraja (deadline). Projektiin on hyvä käyttää 3-5 tuntia. Opettaja voi myös ohjata oppilaita ajankäytöllisesti tuntikohtaisesti esimerkiksi seuraavalla tavalla:

- Tunti 1:** Hahmojen tuominen ja ohjelmointi
- Tunti 2:** Ohjelmointi ja piirtäminen
- Tunti 3:** Piirtäminen ja animointi
- Tunti 4:** Viimeistely ja esitysten valmistelu
- Tunti 5:** Esitykset

Vinkkejä interaktiivisen tarinan ohjelmointiin

Oppilaan kirjan sivuilla 33-37 on vinkkejä ohjelmointiin.

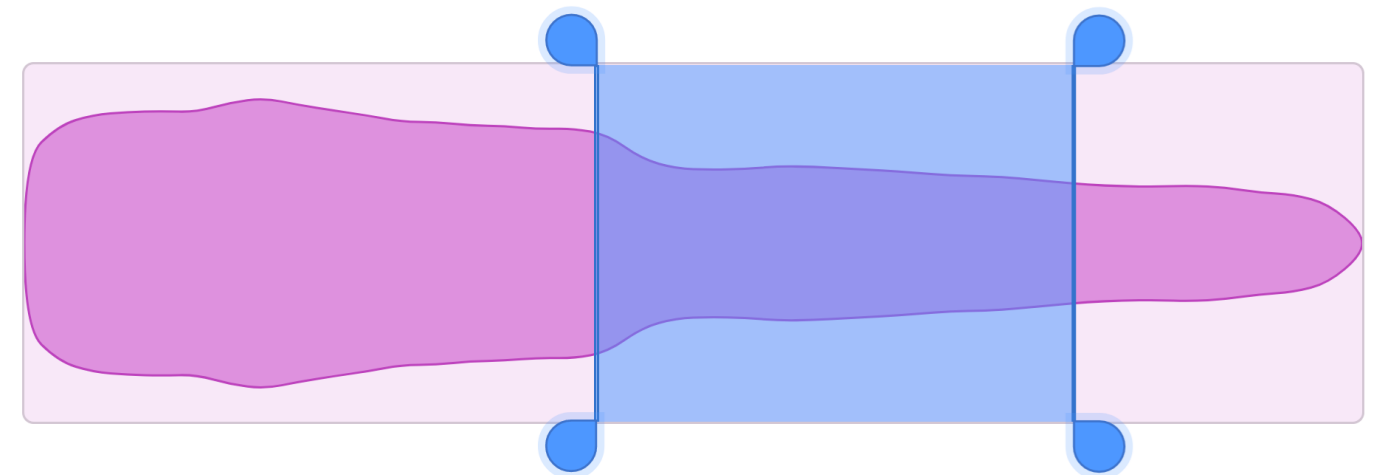
Hyvä tietää äänien käytöstä


Hahmojen äänet toimivat kuten asusteet: Jokaisella on omansa ja ne täytyy hahmoille hakea erikseen. Ääniä voidaan myös äänittää käyttämällä tietokoneen mikrofonia.

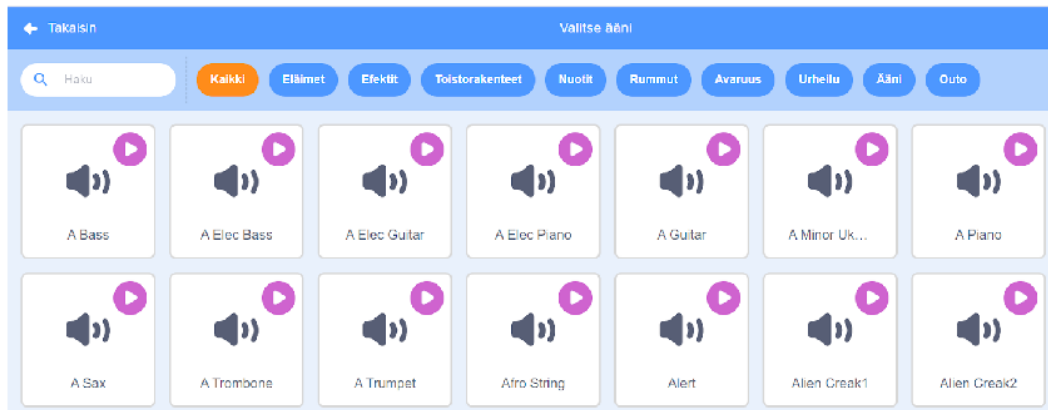
Ääni voidaan myös muokata leikkaamalla tai käyttämällä efektejä:



Efekti voidaan määrätä tiettyyn osaan ääntä maalaamalla se hiiren vasemmalla näppäimellä ja klikkaamalla haluttua efektiä.

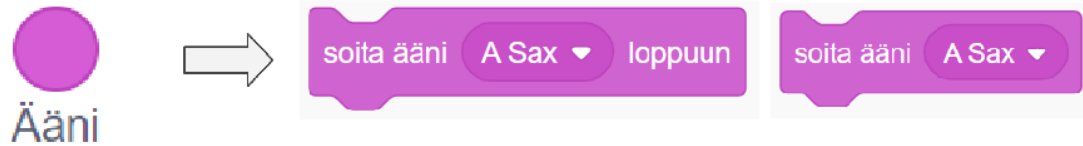


5. Voit kuunnella ääniä viemällä hiiren -**kuvakkeen** päälle.



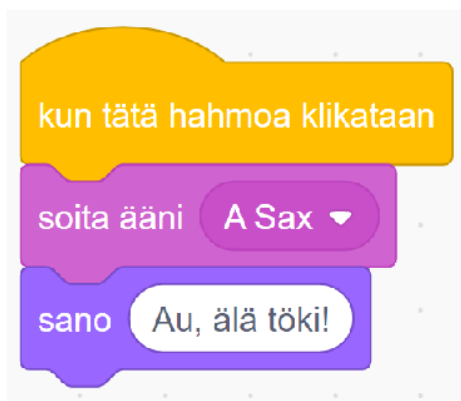
6. **Lisää haluamasi ääni** hahmollesi **klikkaamalla sitä**.

7. Kun hahmolla on ääni, sitä voi käyttää koodissa. **Ääni**-valikossa on kaksi lohkoa äänien käyttöön: **soita ääni () loppuun** sekä **soita ääni ()**.



Ääniefektit ja musiikkiluupit

1. **Ääniefektejä** tehdessäsi kannattaa käyttää **soita ääni ()**-lohkoa.

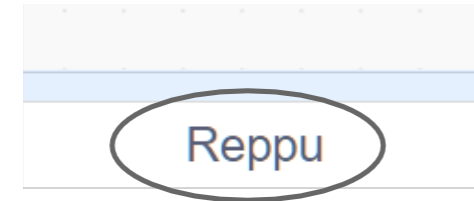


Tällöin ääni soitetaan ja **seuraava lohko suoritetaan välittömästi** ilman viivettä.

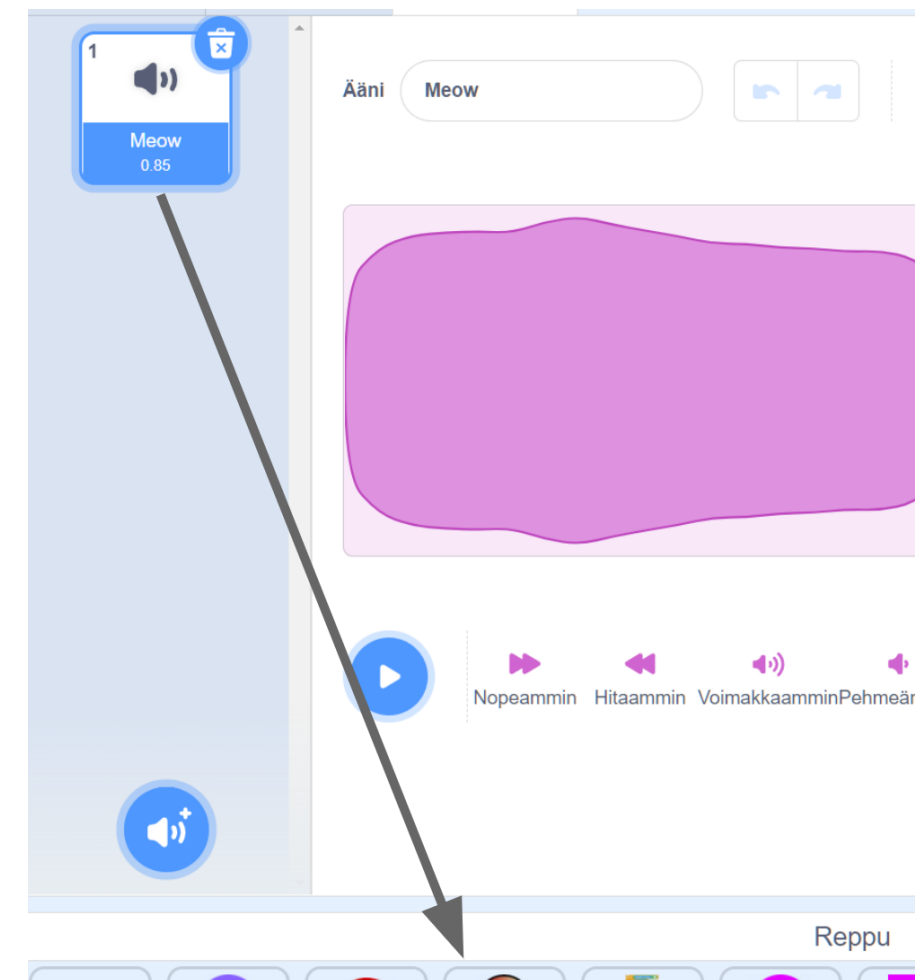
Repun käyttäminen

Kun Scratchia käytetään Scratch tunnuksilla, käyttäjä voi siirtää komentoja, ääniä, skriptejä tai kokonaisia hahmoja talteen eri projekteissa käytettäväksi. Voit hakea kenen vain projektista vaikkapa ääniä, joita haluat käyttää omissa projekteissa.

1. Reppu avataan klikkaamalla alareunasta.

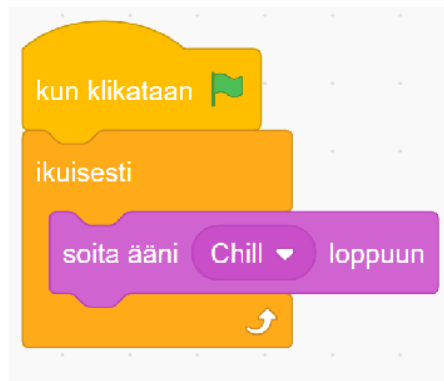


2. Nyt haluttu asia voidaan tallettaa reppuun tarttumalla siihen hiiren vasemmalla näppäimellä ja siirtämällä se reppuun.



3. Nyt talletettu asia säilyy repussa ja se voidaan tuoda mihin vain projektiin.

2. **Taustamusiikkiin** kannattaa käyttää **soita ääni () loppuun** -lohkoa.



Tällöin ääni soitetaan **loppuun asti** ja seuraava lohko suoritetaan vasta sitten. Jos laitamme lohkon **jatkuvaan toistoon**, musiikki alkaa soida uudelleen ja siitä tulee **loppumaton taustamusiikki**.

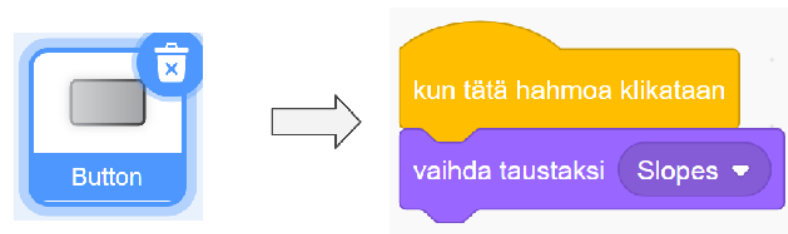
Tällä tavoin toistettavaksi soveltuvat äänet ovat äänikirjaston **Toistorakenteet**-kategoriassa.

Toistorakenteet

Ohjelman ohjaus taustaa vaihtamalla

Voit helposti vaihtaa tapahtumien kulkua taustaa vaihtamalla. Tämä esimerkki on "Kaksi apinaa aavikolla" -esimerkistä:

1. Voimme tehdä nappulan näköisen hahmon, jota klikattaessa **tausta vaihtuu**.



2. Hahmot eivät tiedä, milloin toisia hahmoja klikataan. Mutta ne saa **reagoimaan taustan vaihtumiseen!** Käytä **kun tausta vaihtuu ()** -tapahtumaa.



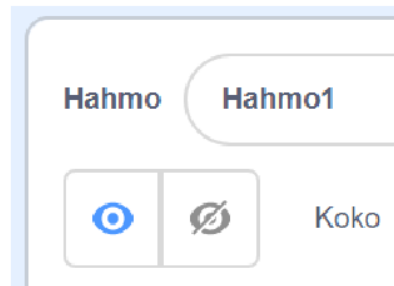
Hahmojen piilotus ja näyttäminen

Kaikkien hahmojen ei tarvitse olla näkyvissä koko ajan. Voit ohjelmoida hahmot näkymättömiksi ja taas näkyviksi.

Ulkonäkö-valikossa on lohkot **piilota** ja **näytä**. Voit niillä piilottaa ja näyttää hahmoja.



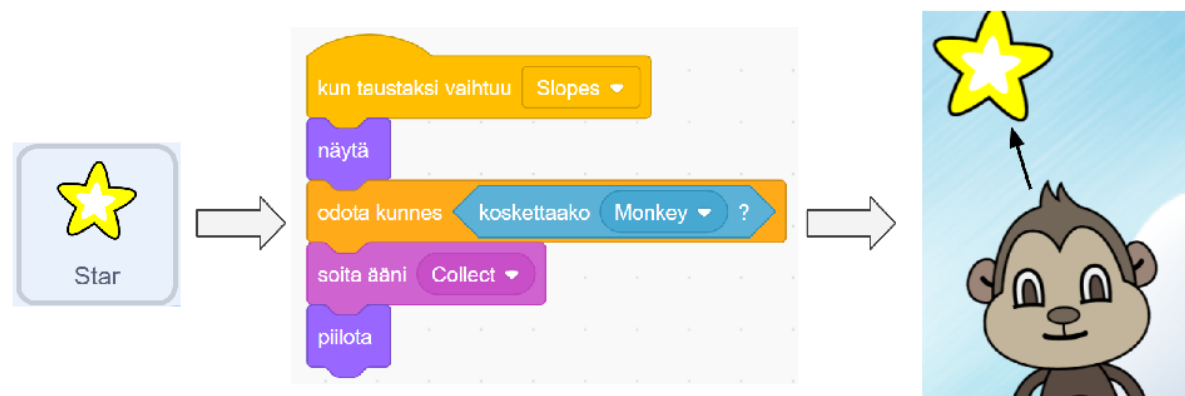
Voit myös piilottaa tai näyttää hahmon sen **asetuksissa esiintymislavan alapuolella**.



Odota kunnes -lohko (haastava)

Odota kunnes -lohkoa voidaan käyttää, kun hahmon halutaan "jäädän jumiin" kunnes jokin **ehto täyttyy**.

Alla on esimerkki, jossa **Tähti-hahmo** odottaa, kunnes **apina-hahmo koskee siihen**. Kun tämä tapahtuu, tähti soittaa äänen ja häviää.



Palaute: Projektien esittely ja testaus

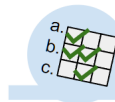
Kun projektit on tehty, ne **esitellään toisille oppilaille**.

Tämän voi toteuttaa esimerkiksi koko luokan kesken tai ryhmissä (3-5 tuotosta per ryhmä). Projekteista annetaan palautetta ainakin seuraaviin kysymyksiin vastaten:

- Mikä esitellyssä projektissa oli parasta?
- Mitä parannettavaa projektissa on?

Vaihtoehtoisia tapoja esittelyyn:

Oppilaat jättävät projektinsa auki tietokoneelle. Luokassa kierretään järjestyksessä projektilta toiselle. Jokaisen tietokoneen luona on kynä ja paperia, jolle kerätään kehuja ja kehittämissuhteita.



Arviointi

Tuotosten arviointi

Opettaja voi suorittaa tuotosten arvioinnin kahdella eri tavalla tai niiden yhdistelmällä:

- Vertaisarviointi:** Jokainen yksikkö/ryhmä antaa opettajan määrittämällä asteikolla arvosanan jokaisen ryhmän tuotokselle, myös omalleen. Dokumentoinnin helpottamiseksi opettajan kannattaa numeroida ryhmät. *Tässä arviointitavassa oppilaille täytyy tehdä selväksi, mitä he arvioivat: Ei esimerkiksi tuotosten tehneitä ihmisiä vaan sitä, kuinka hyvin projekti saavuttaa **projektin vaatimukset (sivu 31)** ja mahdolliset opettajan asettamat lisävaatimukset. Arvioinnin kohteeksi voidaan ottaa myös esimerkiksi tuotoksen ulkonäkö digitaalisen piirtämisen näkökulmasta.*
- Opettajan arviointi:** Opettaja arvioi tuotokset **projektin vaatimusten (sivu 31)** ja mahdollisten opettajan asettamien lisävaatimusten perusteella.

Arvioinnin tukena voidaan käyttää myös itsearviointia: Oppilas antaa arvosanan omalle työpanokselleen ryhmässä tehtyyn projektiin liittyen.

Lopullisessa oppimiskokonaisuuden arvioinnissa käytetään seuraavaa kouluarvosanoihin vastaavuutta: Perustaidot kuvaa arvosanaa 6-7, Hyvät taidot arvosanaa 8 ja erinomaiset taidot arvosanoja 9-10.

Jatkuvien toistojen keskeyttäminen

Jatkuvat toistot voi pysäyttää **Ohjaus**-valikon **pysäytä**-lohkoilla.



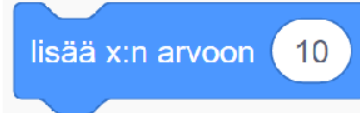
Pysäytä-lohko voi pysäyttää **koko ohjelman**, lohkon oman **skriptin** tai **kaikki saman hahmon skriptit**. Alla on esimerkki, jossa **vihreä lippu saa hahmon pyörimään**. Pyöriminen **päätyy** kun **hahmoa klikataan**.



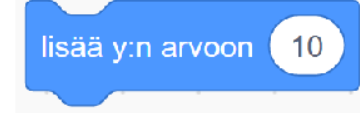
Hahmon siirtäminen vasemmalle, oikealle, ylös, alas (haastava)

Liikkuminen ylös, alas ja sivuille onnistuu muuttamalla hahmon **koordinaatteja**.

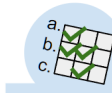
Muuttamalla hahmon **x-koordinaattia** se liikkuu **vasemmalta oikealle**.



Muuttamalla hahmon **y-koordinaattia** se liikkuu **alhaalta ylös**.



Tässä olivatkin vinkit! Voit löytää ideoita ja ohjeita myös aiemmilta sivuilta. Esimerkiksi **sivulla 11** on skriptejä, joista voi olla hyötyä projektissasi. Onnea matkaan ja pidä hauskaa!



Arviointi

Oppimiskokonaisuuden arviointi

Oppimiskokonaisuuden perimmäisenä tavoitteena on innostaa digitaalisen tuottamiseen, ohjelmointiin ja luovuuteen, joten summatiivisia arvosanoja ei välttämättä tarvitse painottaa.

Esimerkki arviointisuunnitelmasta

Opettaja voi antaa halutessaan oppimiskokonaisuudesta numeroarvosanan.

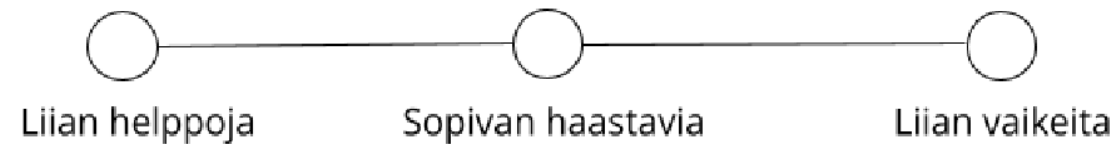
Esimerkki arvioinnista:

Prosentuaalinen painotus	Arvioitava tuotos
25 %	Interaktiivisen tarinan lopputulos esityksen perusteella (osa 1)
25 %	Sokkelopelin lopputulos esityksen perusteella (osa 2)
25 %	Loppukoe (osa 2)
25 %	Aktiivisuus ryhmätyöskentelyssä (osat 1-2)

Lopuksi!

Vastaa seuraaviin kysymyksiin tehtyäsi kaikki kirjan harjoitukset:

1. Ohjelmointitehtävät olivat ... (keskustele vastauksista parin kanssa!)



2. Mieti opiskeluasi. Mistä olet erityisen ylpeä?

3. Mieti opiskeluasi. Mitä tekisit eri tavalla ensi kerralla?

4. a) Lopuksi - Mieti miltä **omat ohjelmointitaitosi nyt tuntuvat** ja **ota selfie** jossa tämä tunne näkyy!

b) **Vertaa** kuvaa kurssin **alussa ottamaasi**.

c) **Keskustele** parin kanssa: **Ovatko kuvat erilaisia? Miksi?**

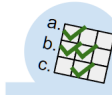
Lopuksi - Osan 1 reflektointi

Ajankäyttö

25-45 min

Työtapa

Työskennellään **2-3** hengen yksiköissä. Rohkaistaan keskustelemaan.



Arviointi

Tehtävä 1

Monipuolinen arviointi antaa opettajalle mahdollisuuden muuttaa järjestelyjä oppilaille suotuisiksi kesken oppimiskokonaisuuden, eikä vain anna kylmää numeroa suorituksesta. Opettaja voi jakaa tehtävään 1 annettujen vastausten perusteella oppilaat tasoryhmiin tai rakentaa parit/ryhmät jatkoa ajatellen niin, että jokaisessa yksikössä on ainakin yksi hyvin osaava (jolle ohjelmointitehtävät olivat "Liian helppoja"). Hyvin osaavista voidaan myös tehdä tutoreita tai apuopettajia osaa 2 varten.

Selfie tai hymiö

Kuten oppimiskokonaisuuden alussa, tässäkin oppilaita ohjataan ottamaan heidän tunnetilaa kuvaava kuva. Opettaja voi ohjata oppilaita myös piirtämään tunnetilaa kuvaavan hymiön.

Osa 2 - Ehtolauseet ja pelinteko

Luulemme, että tietokoneet ovat **älykkäitä**. Todellisuudessa **tietokoneiden älykkyys riippuu ohjelmoijasta**.

Pelillä komennoilla ja toistorakenteilla ei vielä luoda älykkyyttä. Siihen tarvitaan **ehtolauseita!** Niiden avulla voimme tehdä **ohjelmia, jotka reagoivat käyttäjän tekemisiin**.

Pelit ovat yksi parhaista tavoista tutustua **ehtoihin** ja **ehtolauseisiin**. Tässä osassa teette omia **sokkelopelejä**.

Tässä osassa opit lisää toistorakenteista sekä ehtojen ja ehtolauseiden käytöstä harjoitusten ja pelin tekemisen kautta.

Osa 2, sivut 39-72

Yleiskatsaus

Osassa 2 oppilaat syventyvät ehtoihin, ehtolauseisiin ja niiden yhdistämiseen toistorakenteiden kanssa. Projektina toteutetaan peli, jossa sovelletaan kaikkea tähän asti opittua. Myös osassa 2 suositaan parityöskentelyä, mutta toteutuksen voi tehdä myös yksin tai 3 hengen ryhmissä.

Ajankäyttö

Osa 2 koostuu noin 12-14 oppitunnista (45 min/oppitunti).

Tarvikkeet

- Tietokone, näppäimistö, hiiri (väh. 1/oppilaspari)

Tärkeää osassa 2

- Motivointi (*katso sivu 50*)
- Rohkaise yhteistyöhön ja vertaisoppimiseen
- Anna riittävästi aikaa löytää omia ratkaisuja ongelmiin, erityisesti projektityössä

Oppitunnit

Lista oppitunneista		
Oppitunti #	Sisältö	Oppilaan kirjan sivut
12	Osion 2 aloitus, ehtolauseet	39-44
13	Ongelmanratkaisu	45-49
14	Projektin esittely, esimerkkiprojektien testaus, suunnittelu	50-51
15	Pelipohjan tekeminen	52-58
16	Pelipohjan viimeistely	59-61
17	Projektityö	62-71
18	Projektityö	62-71
19	Projektityö	62-71
20	Projektityö, projektin viimeistely	62-71
21	Projektin viimeistely	62-71
22	Pelien testaaminen, vertaisarviointi	Lomake (tuloste)
23	Loppukoe	Koe (tuloste)
24	Arviointi, päätös	Koe (tuloste) + 72

Aloitetaan - Osa 2

Ennen jatkamista, aseta tavoitteita omalle oppimiselle ja toiminnalle.

1. Osaat nyt jo paljon. Olet koodari!

Piirrä itsesi ohjelmoijana! →

Voit myös antaa itsellesi humoristisen koodarinimen.

Koodarinimesi:

2. Avun pyytäminen on tärkeä osa oppimista, Erityisesti ohjelmoinnin oppimisessa.

Keskustele alla olevien kysymysten avulla parin kanssa tai ryhmässä:

a) Osaitko pyytää apua osassa 1, mikäli sitä tarvitsit?

b) Miksi? Miksi et?

3. Aseta itsellesi **yksi lupaus** oppimiseen liittyen. Esimerkiksi: "Lupaan pyytää apua kun sitä tarvitsen." tai "Lupaan auttaa ainakin neljää luokkatoveria."

→ **Nauhoita** tai **kirjoita** lupaus muistiin. Muista lupauksesi ja **pyri pitämään se.**

Aloitetaan - Osa 2

Ajankäyttö

15 min

Työtapa

Työskennellään **pareittain**.

Keskustelua voidaan käydä myös parien välillä, **4 hengen ryhmissä**.

Teemaan johdattelu

Tehtävä 1 voidaan toteuttaa myös esimerkiksi kuvataiteeseen liittyvänä pienenä projektina, jolloin siihen voidaan käyttää enemmän aikaa.

Luetaan - Ehtolauseet ja totuusarvo

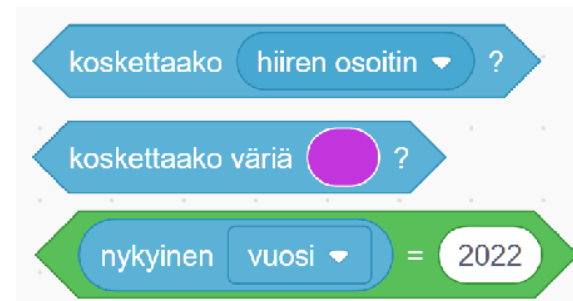


Pelejä, sovelluksia ja robotteja ohjelmoitaessa täytyy ymmärtää, miten **ehtolauseet** ja **totuusarvot** toimivat. Logiikan avulla saadaan aikaan älykkyyttä!

Ehdot Scratchissa

Scratchissa ehdoilla ovat muodoltaan **päistä teräviä**. Ne eivät ole komentoja. Niitä voidaan vain käyttää osana **ehtolauseita**.

Scratchissa on valmiita ehtoja. Esimerkiksi **koskettaako väriä ()** -ehdolla voidaan testata, koskettaako hahmo tiettyä väriä. Scratchissa voi myös tehdä omia ehtoja. Esimerkiksi voidaan testata, onko nyt vuosi 2022 vai ei.



Luetaan - Ehtolauseet ja totuusarvo

Ajankäyttö

30 min

Työtapa

Työskennellään **2-3** hengen yksiköissä, lukien ja keskustellen

Avainsanat

Ehto, totuusarvo, logiikka, tosi, epätosi, ehtolause, jos, jos-muuten, toisto, odotus.

Laitteettomia harjoituksia 1 - Tee kuten käsketään

Tunnin voi aloittaa tällä laitteettomalla harjoituksella.

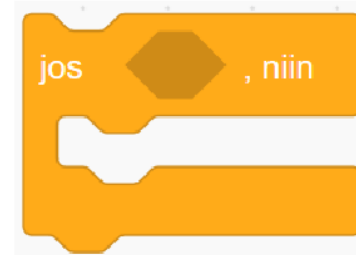
- Oppilaat ovat robotteja ja opettaja käskee heitä.
- Opettaja antaa erilaisia ehtolauseita sisältäviä toimintaohjeita. Tässä esimerkkejä:
 - **Jos** (nyt on maanantai) **niin** taputa käsiä
 - **Jos** (nyt on maanantai) **niin** taputa käsiä **muuten** viittaa oikealla kädellä
 - **Jos** (nyt on maanantai) **TAI** (nyt on tiistai) **TAI** (nyt on perjantai) **niin** taputa käsiä **muuten** viittaa oikealla kädellä
 - **Jos** (näet punaista) **TAI** (näet keltaista) **niin** sulje silmäsi **muuten** älä tee mitään
 - **Jos** (ulkona sataa) **JA** (ulkona paistaa aurinko) **niin** mökötä
- Oppilaat voivat myös keksiä itse vastaavia toimintaohjeita pienissä ryhmissä.

Alla esimerkkejä eri ehtolauseista:

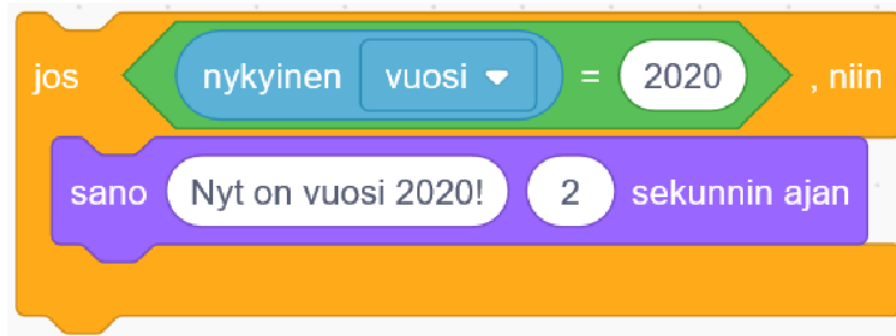
Jos -ehto...



Tarkistaa onko ehto tosi ja suorittaa lohkon sisään rakennetun koodin **ainoastaan jos ehto täyttyy** eli on **tos**.



Esimerkki:



Jos nykyinen **vuosi on 2020**, hahmo sanoo "Nyt on vuosi 2020!".

Jos **vuosi ei ole 2020**, ei tapahdu mitään.

Jos--muuten -ehto...

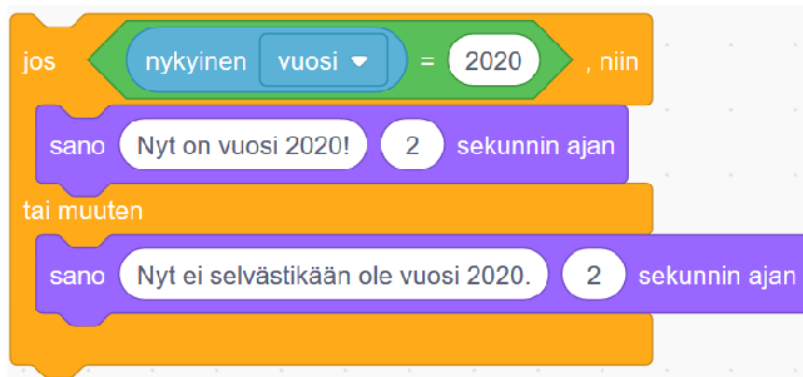


Tarkistaa onko ehto tosi.

Jos ehto **on tosi**, **ylempänä oleva lohko suoritetaan**. Jos ehto on **epätosi**, **alempana oleva lohko suoritetaan**.



Esimerkki:



Jos nykyinen **vuosi on 2020**, hahmo sanoo "Nyt on vuosi 2020!".

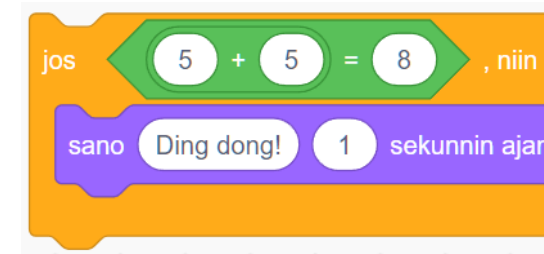
Jos **vuosi ei ole 2020**, hahmo sanoo "Nyt ei selvästikään ole vuosi 2020."

Oppilaat saavat testata Scratchissa

Luetaan-osiossa on paljon esimerkkejä. Opettaja voi halutessaan ohjata oppilaita rakentamaan ja kokeilemaan niitä itse Scratchissa.

Tosi ja epätosi

Matematiikkaan sidottu esimerkki saattaa helpottaa oppilaita ymmärtämään mitä *tos* ja *epätosi* todella tarkoittavat. Tätä opettaja voi demonstroida esimerkiksi tekemällä ehdon **5+5=8**, joka ei voi ikinä olla tosi eli **sano**-lohko ei koskaan suoriteta. Jos muutamme ehdon muotoon **4+4=8**, se on aina tosi.



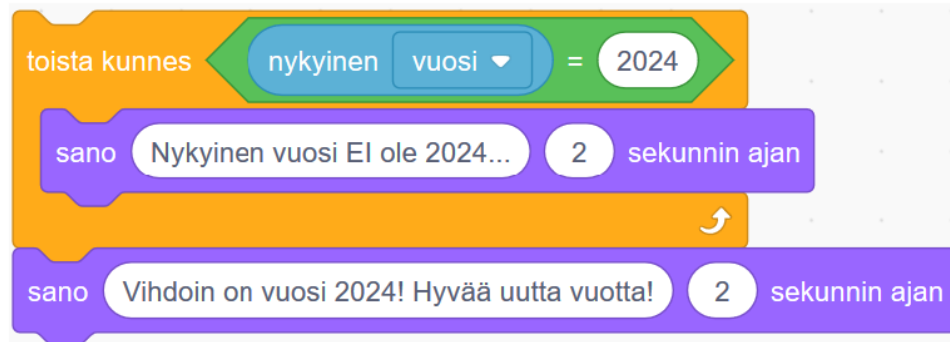
Toista kunnes -ehto...



Toistaa lohkon sisään asetettuja lohkoja **kunnes ehto täyttyy**.



Esimerkki:

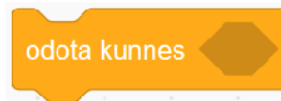


Hahmo sanoo **jatkuvasti** lausetta "Nykyinen vuosi EI ole 2024...", siihen asti, että vuosi vaihtuu vuodeksi **2024**. Kun vuosi on **2024**, asetettu ehto täyttyy jonka jälkeen **toisto loppuu** ja hahmo sanoo "Vihdoinkin on vuosi 2024! Hyvää uutta vuotta!".

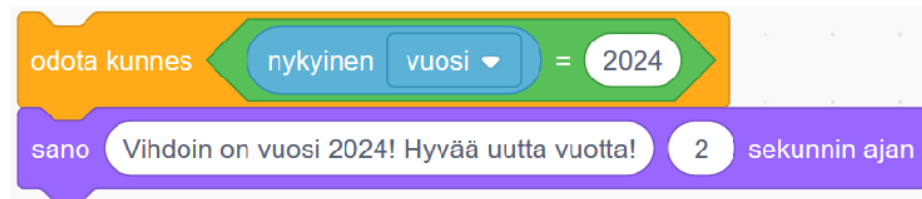
Odota kunnes -ehto...



Odottaa kunnes asetettu ehto täyttyy.



Esimerkki:



Lohko **odottaa** pysäyttää ohjelman etenemisen, **kunnes vuosi on 2024**.

Sen jälkeen seuraava lohko suoritetaan, ja hahmo sanoo "Vihdoinkin on vuosi 2024! Hyvää uutta vuotta".

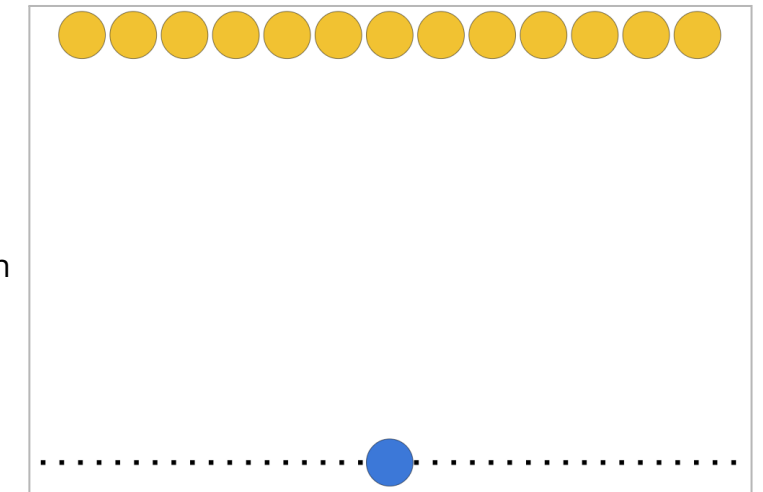
Seuraavaksi pääset kokeilemaan eri ehtolauseita Scratchissa!

Laitteettomia harjoituksia 2 - Kysymyspeli

Toinen esimerkki laitteettomasta ehtolauseiden harjoittelusta on **Kysymyspeli**, joka saattaa olla oppilaille entuudestaan tuttu eri nimellä.

Ohjeet:

- Oppilaat seisovat rivissä kasvotusten opettajan kanssa noin 10 metrin päässä opettajasta
- Opettaja sanoo yhden ehtolauseen kerralla. Esimerkiksi, "Jos sinulla on keltaiset sukat, ota kaksi askelta eteenpäin".
- Kuten ohjelmoinnissa, opettaja voi myös antaa erilaisia komentoja ehtolauseiden kanssa. Esimerkiksi "askel taaksepäin" tai "käänny 180-astetta". Oppilaiden vain täytyy noudattaa ohjeita sanasta sanaan.
- Oppilas, joka pääsee opettajan tasalle ensimmäisenä, ottaa opettajan roolin pelissä.



Testaa ehtolauseita

Testaa ja tutki!

1. a) Mene selaimella tähän osoitteeseen: <https://codeschool.fi/ea6> ja
b) Klikkaa **Katso sisälle** -nappia.

2. **Kokeile klikata näppäimiä A, B, C, D and E, yksi kerrallaan.**
Ennen kuin kokeilet uutta näppäintä, resetoï ohjelma klikkaamalla kuvaketta.

3. **Keskustele** parisi kanssa. Selittäkää eroavaisuudet...
a) A ja B -näppäinten skriptien välillä
b) C ja D -näppäinten skriptien välillä
c) A ja E -näppäinten skriptien välillä

4. **Kuvaile**, mitä hahmo tekee kun **painetaan E -näppäintä**.

Testataan ehtolauseita

Tässä vaiheessa oppilaat viimeistään pääsevät itse testaamaan erilaisia ehtolauseita.

Tehtävässä 3 ohjataan keskustelemaan. Tämä keskustelu voidaan käydä parin kesken, parien kesken tai koko luokan kesken.

Esimerkkivastaus tehtävään 4:

*Kun E-näppäintä on painettu, hahmo ryhtyy jatkuvasti (ja loputtomasti) testaamaan, osuuko hiiren osoitin siihen. Jos osuu (**tos**) niin se reagoi, jos ei (**epätosi**) niin se palaa normaaliksi.*

JA, TAI, EI

Tässä on esitelty loogiset operaattorit, joita ei kirjassa varsinaisesti ohjata käyttämään, mutta opettaja voi ne ottaa halutessaan esille.

- **JA** yhdistää ehdot: Vain jos molemmat ovat tosia, koko ehto on tosi.

- **TAI** yhdistää ehdot: Jos edes toinen on tosi, koko ehto on tosi.

Huomio: Päivä 7 on Scratchissa lauantai ja päivä 1 on sunnuntai.


- **EI** kääntää ehdon. Jos sisään asetettu ehto ei ole tosi NOT, siitä tulee tosi, ja toisinpäin.

Ehtolause peli

Seuraa ohjeita ja ratkaise haaste.

Ohjeet

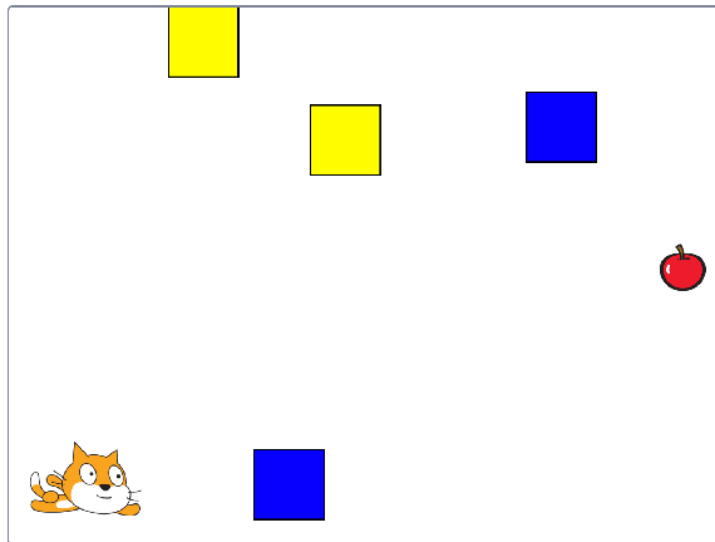
1. Mene sivulle: <https://codeschool.fi/ea7> ja
2. Klikkaa **Katso sisälle** -painiketta.

 Katso sisälle

H Haaste!

Haaste: Saatko kissan sanomaan "I found the apple!" pelkästään keltaisia ja sinisiä laatikoita siirtelemällä?

Sinun pitää lukea **kissan koodia, jotta osaat käyttää laatikoita oikealla tavalla.**

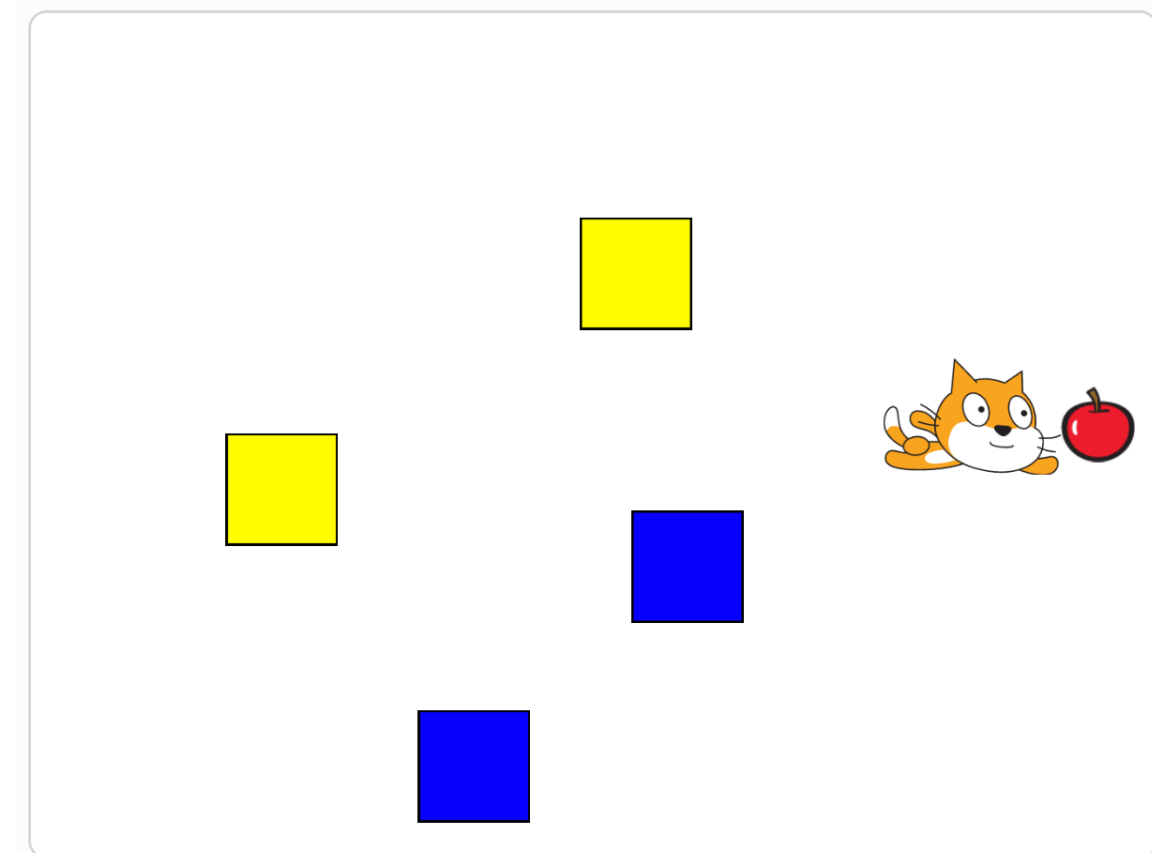


MUISTA: Älä muuta koodia! Ratkaise haaste pelkästään laatikkoja siirtelemällä.

Ehtolause peli

Tämä on valmis ohjelma, jossa oppilaiden tulee tutkia koodia, muttei muokata sitä. Sen sijaan ajatuksena on vain liikuttaa hiirellä laatikot sopiville paikoille niin, että kissa pitää päätyä omenan luo. Kissan tulee myös koskea jokaiseen laatikkoon matkalla.

Esimerkkiratkaisut - Sivun 45 haastetehtävät



Kuva esimerkkiratkaisusta

Ohjelmoidaan - Ehtolauseiden käyttäminen

Nyt pääset sukeltamaan ehtolauseiden maailmaan ratkaisemalla haasteita.

Mr. Nosey -haasteita

Ohjeet

1. Mene sivulle: <https://codeschool.fi/ea8> ja Katso sisälle
2. Klikkaa **Katso sisälle** -painiketta.
3. Tässä on **Mr. Nosey!** Hän tarvitsee apuasi. Hän haluaa **piirtää kuvioita sinisen ympyrän sisään kuvioita** . Sinun tehtäväsi on ohjelmoida häntä.



H Haaste!

Haaste 1: Ohjelmoi Mr. Nosey aloittamaan piirtäminen kun A näppäintä on kerran painettu. Käytä näitä lohkoja:



Oikeat lohkot on valmiina ohjelmointialueella.

MUISTA: Mr. Nosey ei saa poistua sinisestä ympyrästä.

Ohjelmoidaan - Ehtolauseiden käyttäminen

Ajankäyttö

45 min

Työtapa

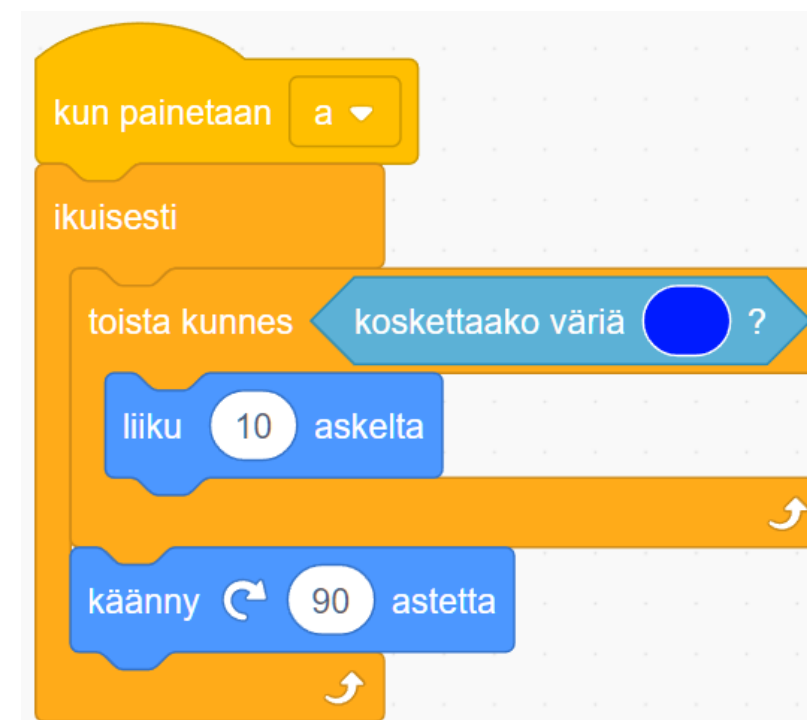
Työkennellään **1-3** hengen yksiköissä

Avainsanat

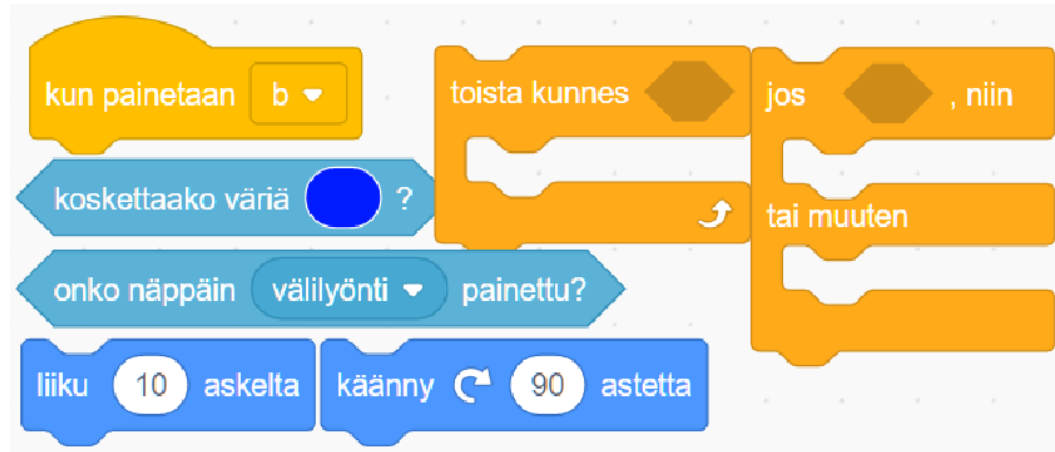
Ehto, tosi, epätosi, ehtolause, jos, jos-muuten, toisto, kunnes, ikuisesti, tuntoaisti, koskettaa.

Esimerkkiratkaisut - Sivujen 46-48 haastetehtävät

Haaste 1:

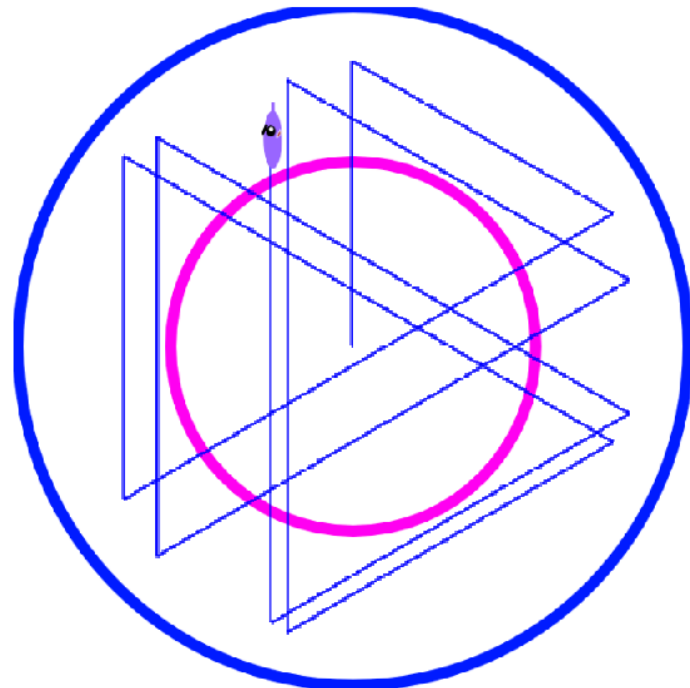


Haaste 2: Ohjelmoi Mr. Nosey **piirtämään kun B näppäintä painetaan**. Mutta **kun välilyöntiä painetaan kerran, Mr. Nosey pysähtyy**. Käytä näitä lohkoja:



Oikeat lohkot on valmiina ohjelmointialueella.

Haaste 3: Muuta äsken tehtyä skriptiä niin, että Mr. Nosey piirtää **kolmion muotoista kuviota**. Muuta vain **lohkojen numeroita**.

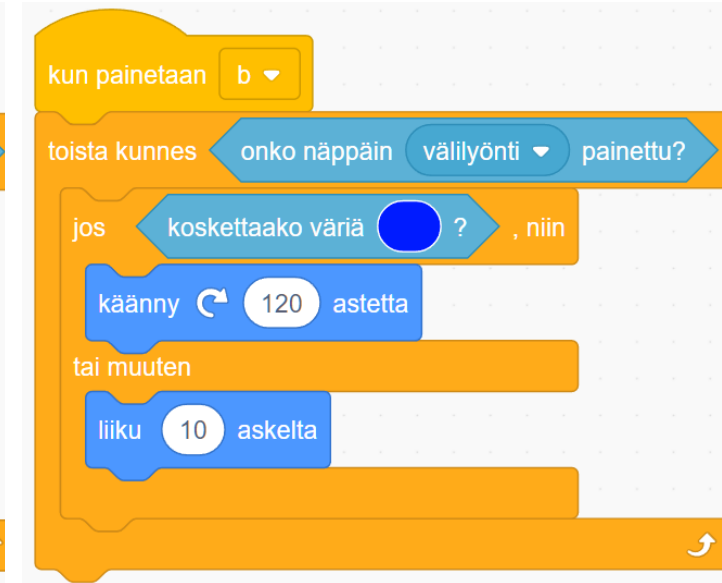


Piirroksen tulisi näyttää kutakuinkin tältä.

Haaste 2:



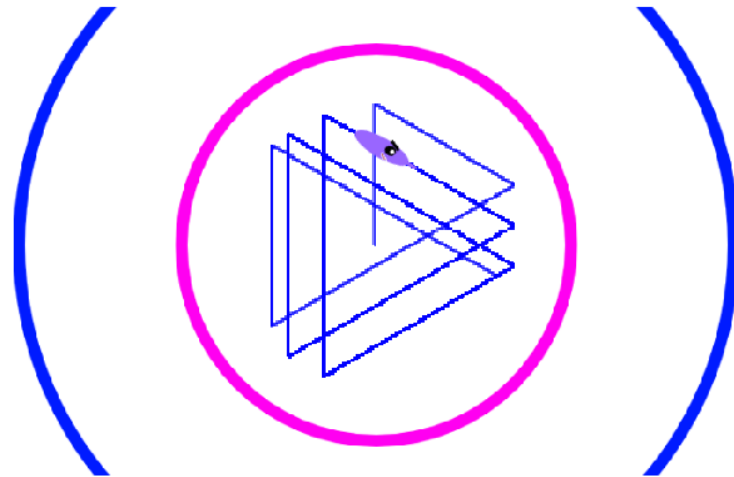
Haaste 3:



Kuten robotin ohjelmointi

Mr. Nosey harjoitukset mukailevat yksinkertaisia robotilla toteutettavia harjoituksia: Liikutaan ja käytetään antureita (robotin "tuntoaisteja"), jotta esimerkiksi robotti ei tipu pöydältä vaan kääntyy sen reunalla.

Haaste 4: Muuta edellistä skriptiä niin, että, **kolmiokuvio piirretäänkin pinkin ympyrän sisään.**

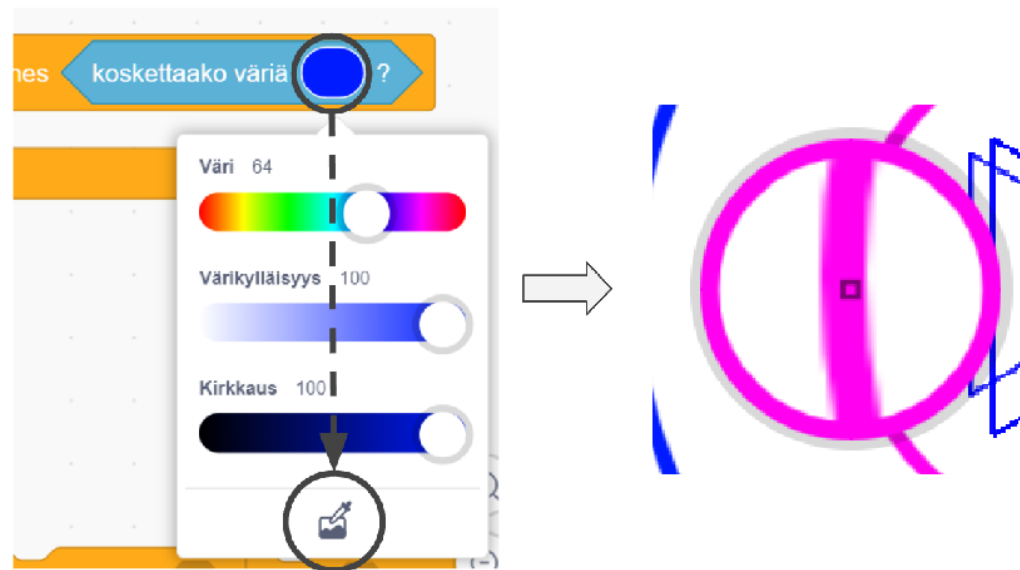


Piirroksen tulisi näyttää kutakuinkin tältä.

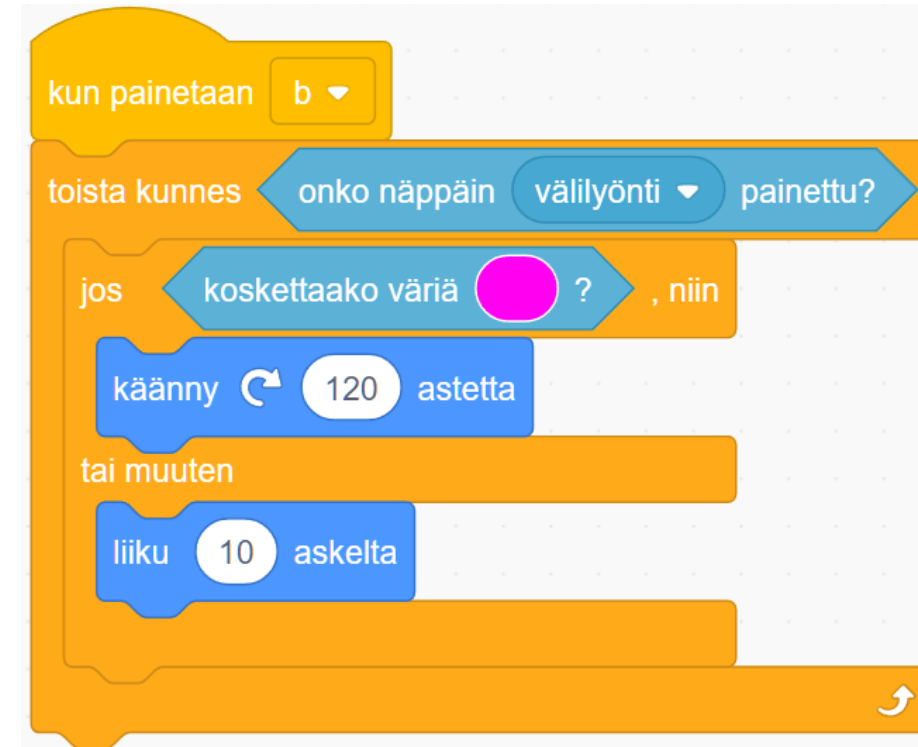
💡 Vinkki haasteeseen 4

Saat ehtoon oikean värin helpoiten **hakemalla sen suoraan näyttämöltä:**

Klikkaa **ehdossa olevaa väriä**, sitten **klikkaa** 🖌️ **-symbolia** ja lopuksi **klikkaa haluttua väriä näyttämöllä.**



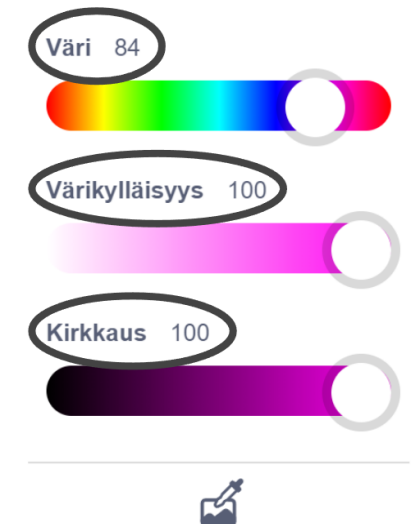
Haaste 4:



Tarkkana värien kanssa

Värin täytyy olla juuri oikea. Helpoin

Väri kannattaa siis hakea ehtoon suoraan taustasta. Värin samanlaisuus voidaan tarkastaa myös lukuja vertaamalla (viivan väri taustassa ja ehtoon asetettu väri).



Haaste 5: Ohjelmoi Mr. Nosey **aloittamaan hiiren osoittimen seuraaminen kun C näppäintä painetaan kerran**. Lisäksi:

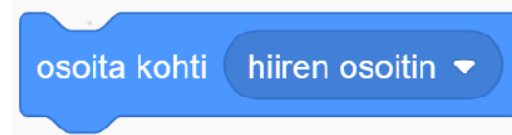
- Seuratessaan hiirtä Mr. Nosey **ei saa pystyä ylittämään sinistä viivaa**.
- Kun hän pääsee **hiiren osoittimelle, skripti pysähtyy**.

→ Tässä video, jolla lopputulos esitetään:

<https://codeschool.fi/ea9>

Vinkkejä haasteeseen 5:

➔ Saadaksesi hahmon seuraamaan hiiren osoitinta, käytä **osoita kohti (hiiren osoitin)** -lohkoa.

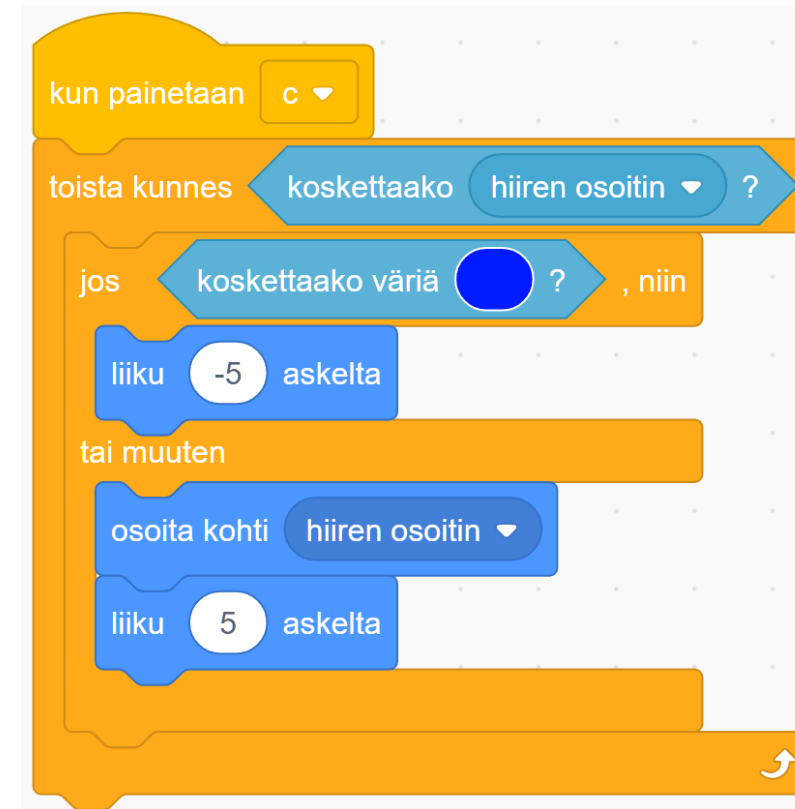


➔ Kosketukseen ja väreihin liittyvät ehdot löydät **Tuntoaisti**-valikosta.



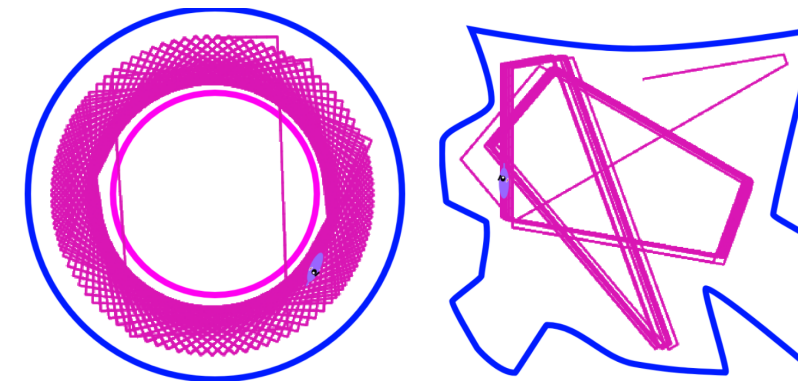
Hienoa työtä! Haasteiden jälkeen on aika aloittaa oman pelin tekeminen →

Haaste 5:

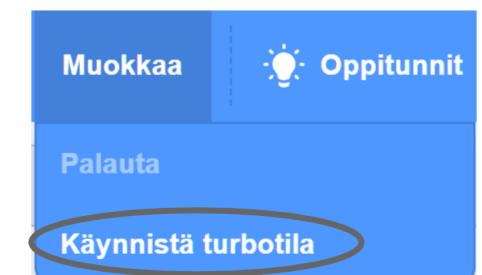


Jos halutaan jatkaa Mr. Noseyn kanssa

Opettaja voi haastaa oppilaat jatkamaan edellisten harjoitusten hengessä. Voidaan esimerkiksi tehdä omia taustoja tai ohjelmoida hahmo piirtämään mielenkiintoisia kuvioita.



Hahmo saadaan kulkemaan äärimmäisen nopeasti käynnistämällä *turbotilan*.



Projektin aloitus - Oma sokkelopeli!

Tästä alkaa **oman sokkelopelin tekeminen**. Ensin rakennetaan pelille pohja seuraamalla ohjeita ja ratkaisemalla haaste tehtäviä. Lopulta pääset rakentamaan peliin itse useita kenttiä, äänimaailmaa ja grafiikkaa. Pelit tehdään mielellään 2-3 hengen ryhmissä, mutta se onnistuu yksinkin.

Tässä **esimerkkiprojekteja**, jotta tiedät paremmin, mistä sokkelopelissä on kyse:

- Perus sokkelopeli: <https://codeschool.fi/ea10>
- Peli 3D vaikutelmalla: <https://codeschool.fi/ea11>

Seuraavalla sivulla aloitat pelin suunnittelun →

Projektin aloitus - Oma sokkelopeli!

Ajankäyttö

2-3 oppituntia (tässä ajassa tehdään pohja pelille)

Työtapa

Työskennellään **1-3** hengen yksiköissä

Avainsanat

Projekti, suunnitelma, peli, ehto, ehtolause, toisto, hahmo, liukuväri, poikittainen, pitkittäinen, säteittäinen

Oppilaiden motivointi projektityössä

Kolme avaintekijää auttavat pitämään oppilaat motivoituneita oppimiskokonaisuuksissa, joissa rakennetaan pitkän ajan saatossa isompia projekteja:

- **Merkityksekkyyt:** Oppilaille osoitetaan, että heidän tuotoksiaan tullaan käyttämään johonkin merkitykselliseen ja ne esitellään toisille. Oppilaat saavat tuotoksistaan palautetta oppimiskokonaisuuden aikana.
- **Itsensä toteuttaminen:** Oppilaat ymmärtävät, että he saavat käyttää luovuutta ja tehdä valintoja oman tuotoksensa suhteen. Oppilailla on riittävästi resursseja oman tuotoksen tekemiseen.
- **Tuki:** Oppilaat eivät tunne olevansa yksin, kun he tarvitsevat ohjausta. Oppilaita rohkaistaan yhteistyöhön ja sosiaalinen ympäristö pidetään turvallisena. Oppilas tuntee, että virheitä ei tarvitse pelätä eikä niistä rankaista.

Ehdotuksia merkityksellisyyden edistämiseksi

- Pelit **esitellään ja toisten pelejä testataan**. Oppilaat antavat positiivista palautetta ja kehittämissuhteita (rakentava palaute).
- Pelit esitellään **rinnakkaisluokalle tai nuoremmille oppilaille**. Oppilaat antavat positiivista palautetta ja kehittämissuhteita (rakentava palaute).
- Oppilaat **äänestävät parhaista tuotoksista** seuraavissa kategorioissa:
 - Paras ulkonäkö
 - Hauskin peli
 - Paras kokonaisuus

Voittajat saavat pienet palkinnot.

Tee pelille alustava tarina

Ryhmäsi jäsenet:

Kuka on pelin **päähenkilö/pelihahmo**?

Mistä **seinät** on tehty? Miksi **pelihahmo** ei saa osua niihin?

Mitä **pelin viholliset** ovat? Miksi **pelihahmo** ei saa osua niihin?

Mikä on pelin **maali** tai **lopputavoite**?

Kuinka varmoja olette **taidoistanne** pelin valmiiksi saamisen suhteen?

Hyvin varma;
Emme usko
tarvitsevamme apua

Melko varma;
Tarvitsemme välillä
hieman apua.

Epävarma;
Tarvitsemme paljon
apua.

Pelin tarina

Pelille suunnitellaan kehyskertomus, jonka ei tarvitse olla tässä vaiheessa kovin tarkka. Kunhan tarina sopii jotenkin pelin teemaan ja siinä on jonkin verran järkeä. Tärkeää on miettiä, mitä hahmot ovat ja miksi.

Esimerkki: **Lapsi on kadottanut ilmapallon.** Ohjaa ilmapallo lapsen luo, mutta varo **piikkejä ja ilkeää siiliä!**

Tarina voidaan myös sitoa tiettyyn oppiaineeseen tai ajankohtaiseen teemaan.

Piirrä oma pelihahmo

Aloitetaan piirtämällä pelihahmo.



Ohjeet

1. **Aloita uusi projekti** klikkaamalla **Luo**.



2. **Poista kissa** klikkaamalla roskakoria. Nyt projekti on tyhjä!



3. Tehdään oma pelihahmo. Saat piirtää sen itse. **Vie hiiri**  **-kuvakkeen päälle. Klikkaa**  **-kuvaketta.**



Eri elementtien tekeminen

Sivuilla 52-58 käsitellään pääosin pelin eri elementtien piirtämistä. Opettajan kirjassa on näille sivuille vähemmän lisätietoa.

Testaa ja tutki!

Nyt voit tehdä **ikioman pelihahmon!**

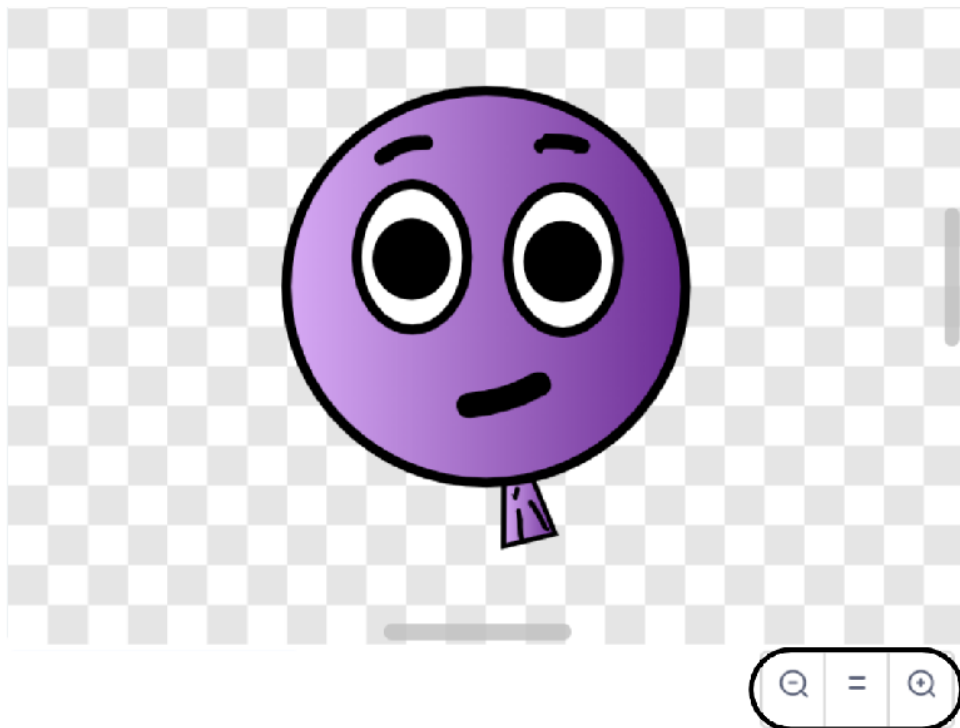
Vinkkejä:

Tee hahmosta riittävän pieni

Sokkelopelin pelaaminen on mielekkäintä **pienellä hahmolla**.

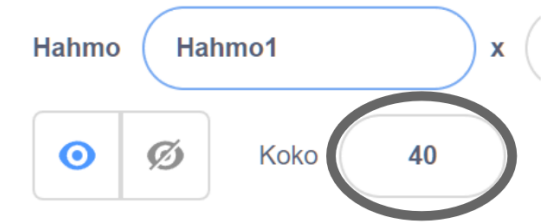


On hankalaa tehdä pieni hahmo ilman **kuvan lähentämistä** ("zoomaus"). Voit lähentää kuvaa klikkaamalla **piirtoalueen alapuolella olevia kuvakkeita**.



Tee pelihahmosta pieni

Kun pelihahmo on pieni, jää enemmän mahdollisuuksia sokkeloiden tekemiseen. Hahmo voidaan piirtää suureksi ja pienentää asetuksista **näyttämön alta** tai **asetta koko arvoon ()**.



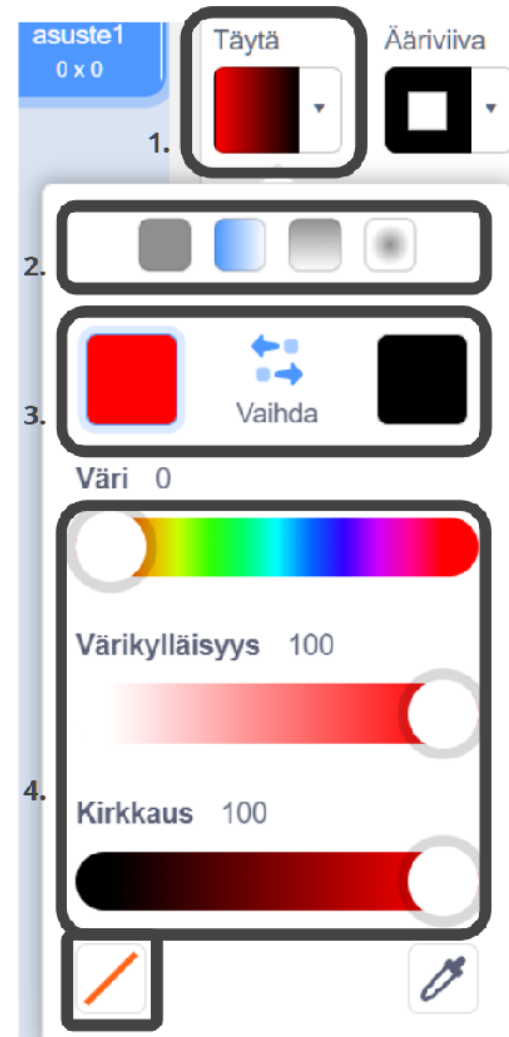
Upeita värejä - Liukuvärit

Piirrosten **täyttämässä** voidaan käyttää **liukuvärejä**:

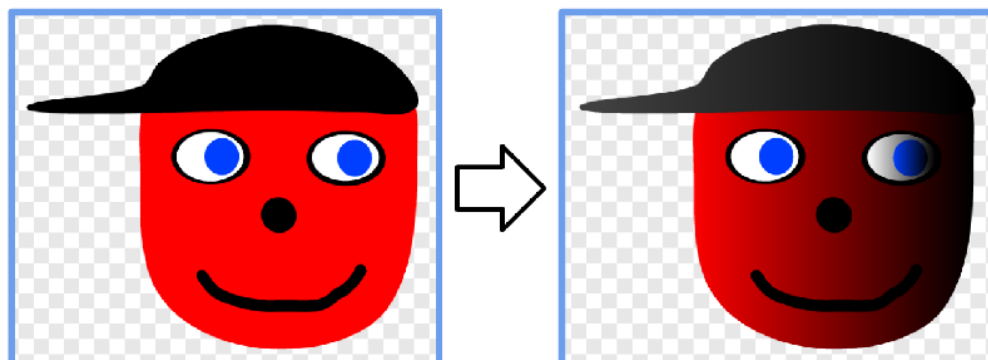
1. Valitse **täyttötyökalu** klikkaamalla **-kuvaketta**.
2. Käytössäsi on neljä täyttötapaa: **yksi väri** , **poikittainen liukuväri** , **pitkittäinen liukuväri** and **säteittäinen liukuväri** .
3. Valitse **kaksi väriä** liukuväriä varten. Liukuväri-asetuksen vaikutus saadaan näkyväksi valitsemalla kaksi **eri väriä**.
4. Värejä voi muuttaa yksi kerrallaan klikkaamalla väriä ja **muuttamalla värejä liukusäätimillä**.

Voit myös asettaa toisen väreistä **läpinäkyväksi**.

Testaa rohkeasti eri asetuksia!



Voit esimerkiksi tehdä kolmiulotteiselta näyttävän hahmon käyttämällä **poikittaista liukuväriä** ja valitsemalla **ensimmäiseksi väriksi haluamasi vaalean värin** ja **toiseksi väriksi mustan**.



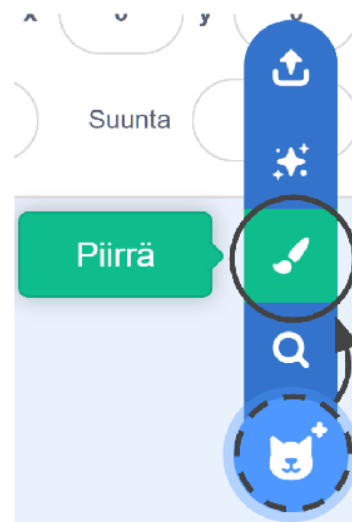
Piirrä maali

Seuraavaksi tehdään **maalialue** kuvaava hahmo

Ohjeet

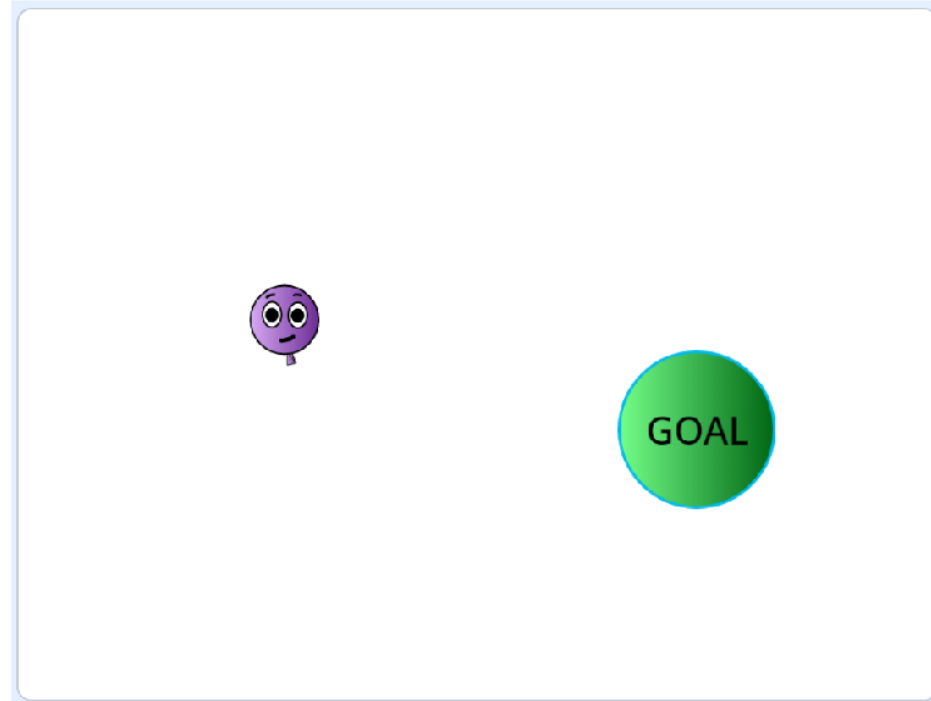
1. Tarvitsemme uuden hahmon. Kuten aikaisemmin, Saat piirtää sen itse.

Vie hiiri  -kuvakkeen päälle. Klikkaa  -kuvaketta.



2. Piirrä pelille **maali**, johon hahmolla pyritään pääsemään. Voit tehdä maalista heti tarinaan sopivan tai jättää maalin toistaiseksi vaikka peläksi palloksi.

Kun maali on piirretty, sinulla on kaksi hahmoa valmiina ohjelmoitavaksi!
Mutta ensin on hyvä tehdä ensimmäinen sokkelo.



Piirrä ensimmäiset kaksi sokkeloa

Nyt on aika piirtää pelille ensimmäiset **kaksi sokkeloa**. Sokkeloiden ei tarvitse olla vielä tässä vaiheessa kovin kummoisia. Kunhan sokkelona on edes laatikko, sitä voidaan hyvin testata.

Ohjeet

1. Sokkelot tehdään esiintymislavan **taustoina**. Vie hiiri  **-kuvakkeelle. Klikkaa  -kuvaketta.**



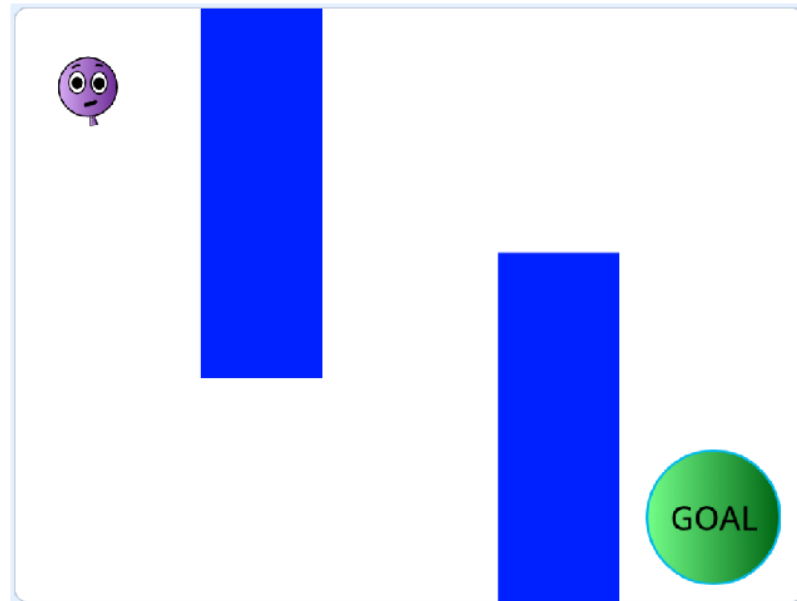
Maalin ja sokkelon värit

Maalin ja sokkelon värien tulisi olla eri, koska pelissä häviäminen tyypillisesti ohjelmoidaan tapahtumaan tiettyyn väriin osumisesta (sokkelo, viholliset). Mikäli maali on sokkeloiden värinen, peli hävitään maaliin osumisesta.

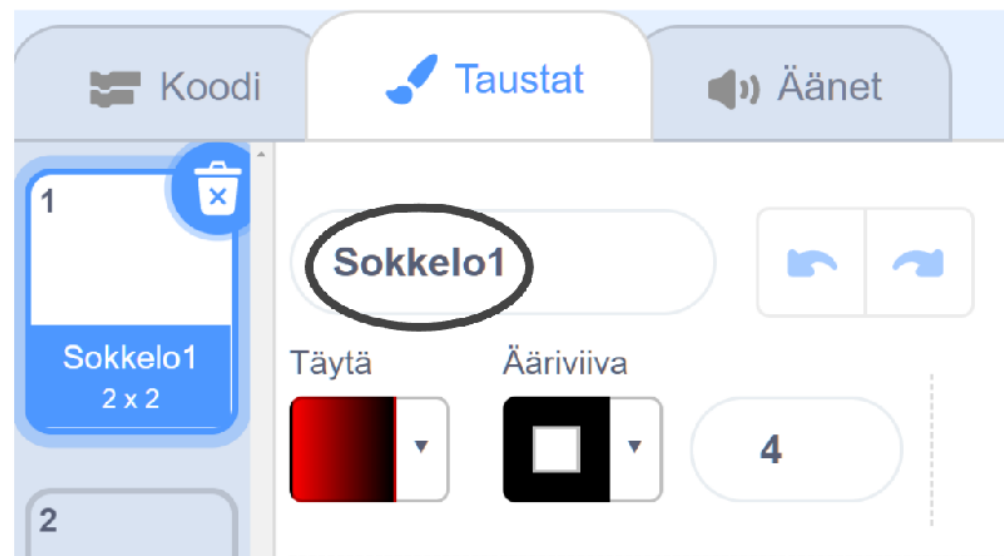
2. Nyt voit aloittaa ensimmäisen sokkelon piirtämisen. **Sokkelon piirtämisessä on tärkeää huomioida seuraavat asiat:**

- **Käytä vain yhtä väriä!** Näin voit käyttää **“koskettaako väriä ()”** -ehtoa peliä ohjelmoidessa.
- Tee ensimmäisestä sokkelosta **helposti läpäistävä**, jotta testaaminen ja pelin aloittaminen on helppoa.

Tässä esimerkki:



3. Taustat ovat pelin eri kenttiä. **Anna ensimmäiselle taustalle kuvaava nimi**, kuten “Sokkelo1” tai “Kenttä1”.



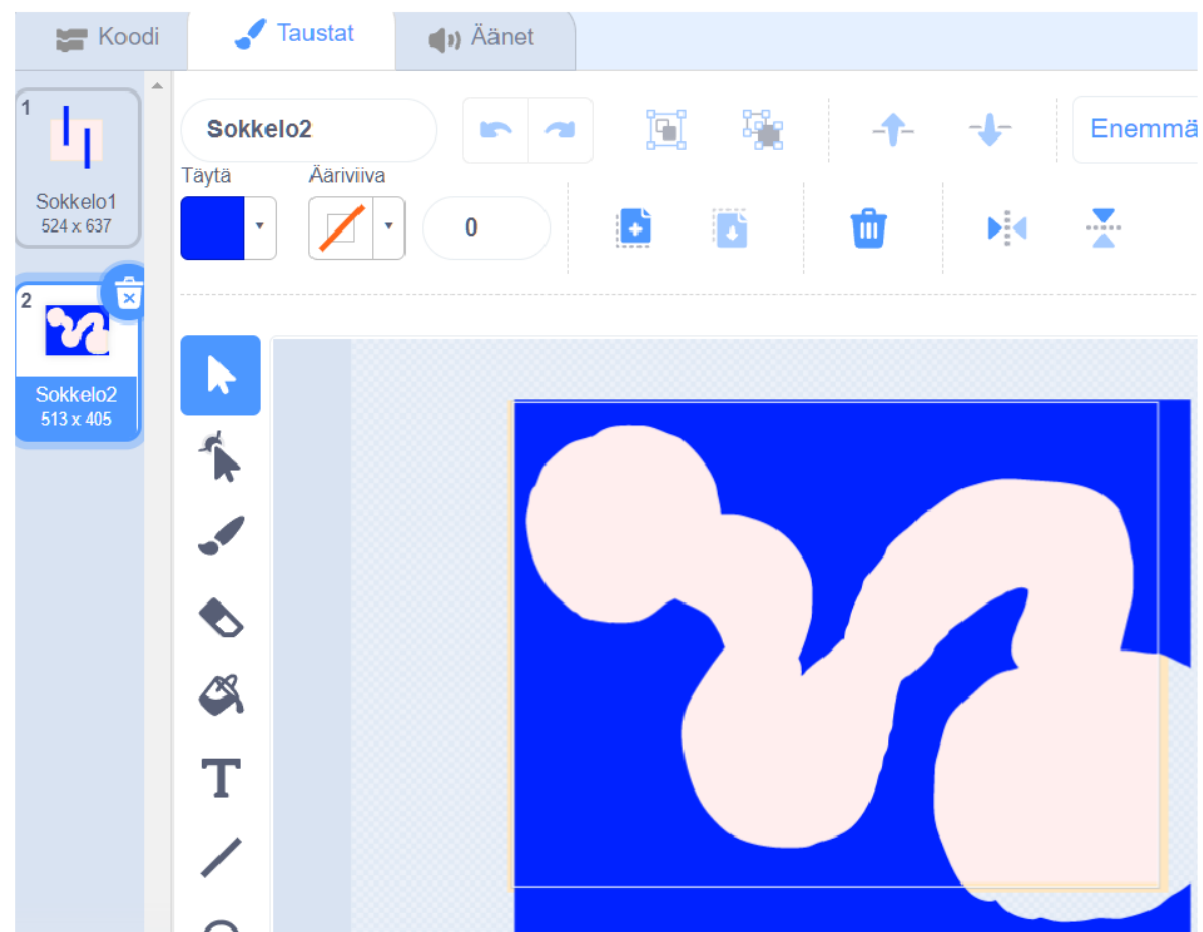
4. Tee toinen tausta. Tästä tulee pelin **toinen kenttä**. Kuten aiemmin, **käytä vain yhtä väriä ja anna taustalle/kentälle sopiva nimi**.

Kuvaava nimeäminen

Jälleen kuvaava nimeäminen ja numerointi on tärkeää jatkokehittämisen kannalta. Scratch automaattisesti nimeää asusteita ja taustoja sopivilla numeroilla, jos ensimmäisen asusteen/taustan nimeksi antaa numeroidun nimen. Esimerkiksi jos ensimmäinen tausta on nimeltään “Sokkelo1”, siitä tehty kopio on automaattisesti “Sokkelo2”.

Hienosäätöä jätetään myöhemmälle

Oppilaita kannattaa muistuttaa, että tässä vaiheessa tehdään vain pohjat peleistä. Jatkokehittäminen ja hienosäätö tehdään myöhemmin.



Kun **pelihahmo**, **maalihahmo** and **kaksi sokkeloa** on valmiina, voit aloittaa ohjelmoinnin!

Ohjelmoi aloitustilanne

Tässä osiossa ohjelmoidaan sokkelopeli alulle. Sinun tehtäväsi on sitten jatkaa itse.


Tässä sokkelopelissä hahmoa liikutellaan hiirellä. Pelihahmon tulisi siis joko **seurata hiiren osoitinta** tai **pysyä hiiren osoittimella**.

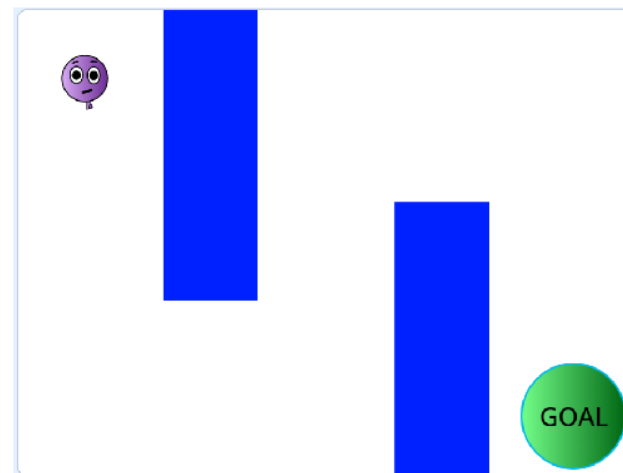
Pelissä on tarkoitus myös varoa seiniä ja tavoitteena päästä maaliin. Peli siis ohjelmoidaan **loppumaan kun kosketaan seinään** ja **kenttä vaihtumaan kun kosketaan maaliin**.



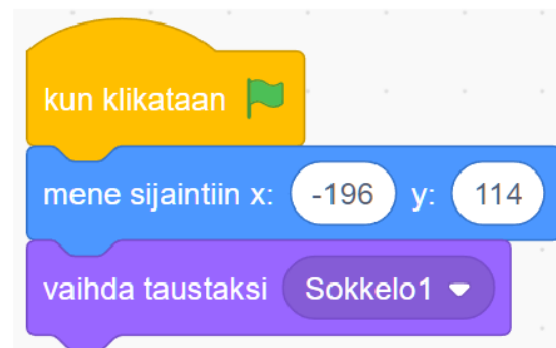
Aloita **pelihahmon** ohjelmoinnilla.

Ohjeet

1. Ohjelmoi pelihahmolle aloituspaikka. Ensin **vie hahmo** halutulle aloituspaikalle. Sitte, hae **mene sijaintiin x:() y:()** -lohko **Liike**-valikosta ja kiinnitä se **kun klikataan**  -lohkoon.



2. Varmistetaan myös, että oikea kenttä on näkyvillä. Hae **vaihda taustaksi (Sokkelo1)** -lohko. Löydät sen **Ulkonäkö**-valikosta.



Alkutilanteen ohjelmointi

Kun tarvittavat elementit on piirretty, oppilaat alkavat ohjelmoida pelin toimintalogiikkaa ohjeiden ja haasteiden kautta.

Nyt voit jatkaa itse! Haasteiden 1 ja 2 ohjelma rakennetaan **kun klikataan** -tapahtumaan. Voit ottaa myös kaksi vihreän lipun lohkoa, jos haluat. On kymmeniä tapoja ratkaista haasteet, joten etsi vain rohkeasti oma tapasi!

H Haaste!

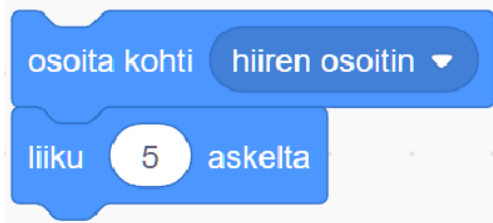
Haaste 1: Ohjelmoi **pelihahmo odottamaan, että hiiren osoitin koskee sitä**. Kun hiiren osoitin koskee pelihahmoon, se lähtee liikkeelle. Käytä sopivaa ehtolausetta!

Haaste 2: Ohjelmoi **pelihahmo seuraamaan hiiren osoitinta. Jos pelihahmo osuu sokkelon seinään, se lakkaa liikkumasta ja peli päättyy**. Käytä sopivaa ehtolausetta!

Vinkkejä haasteeseen 2:

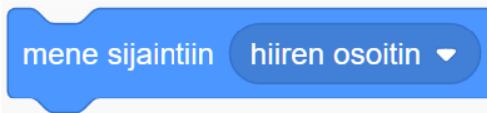
- Voit ohjelmoida pelihahmon seuraamaan hiirtä **tai** pysymään hiiren osoittimella.

Seuraa hiiren osoitinta:



tai

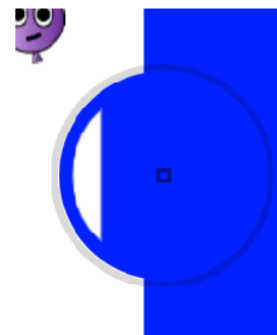
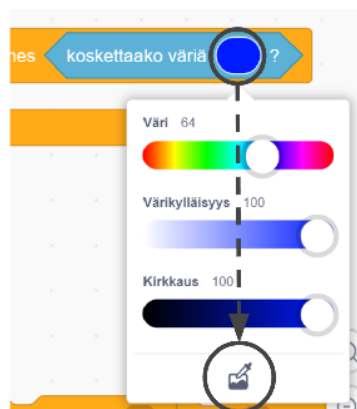
Pysy hiiren osoittimella:



Jotta toiminto tapahtuu jatkuvasti tarvitset myös toistorakenteen.

- Muista, että voit hakea sokkelon seinän värin suoraan näyttämöltä.

Klikkaa **ehdossa olevaa väriä**, sitten **klikkaa** -symbolia ja lopuksi **klikkaa haluttua väriä näyttämöllä**.



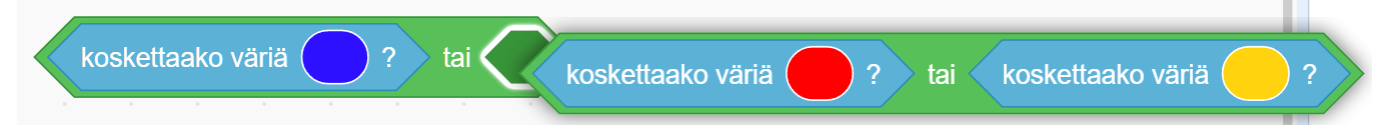
Viimeinen vinkki seuraavalla sivulla →

Useiden värien käyttäminen sokkelossa

Häviämistä ei tarvitse ohjelmoida tapahtumaan ainoastaan yhdestä väristä. Ehtoon voidaan **TAI** operaattorilla lisätä värejä, jotka aiheuttavat pelin häviön.



TAI-lohkoja voidaan myös yhdistellä toisiinsa. Näin saadaan ehtoon niin monta väriä kuin halutaan.



Useiden värien ehtoa ei ole pakko tehdä.

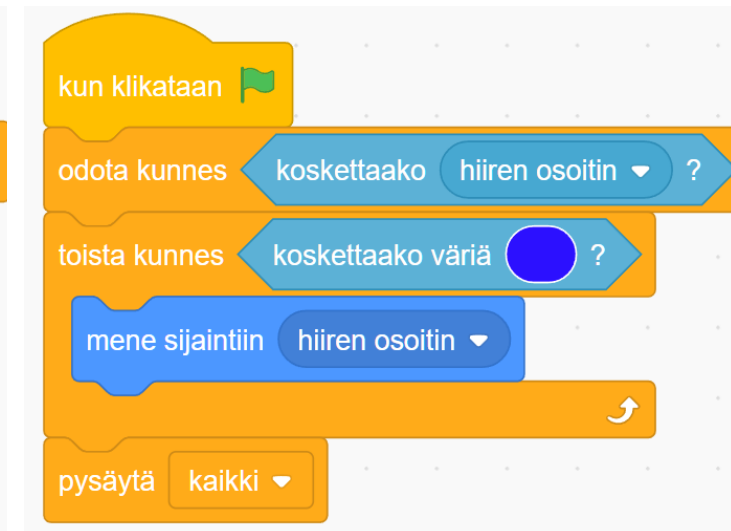
Esimerkkiratkaisut - Sivujen 60-61 haastetehtävät

Huomio: Haastetehtävien ratkaisuja on kymmeniä erilaisia.

Haasteet 1-2 (vaihtoehto 1):



Haasteet 1-2 (vaihtoehto 2):



↪ Koko ohjelma voidaan lopettaa **pysäytä [kaikki]** -lohkolla. Tällä voi pysäyttää pelin. Lohko voidaan myös muuttaa **pysäytä [tämä koodi]** or **pysäytä [hahmon muut koodit]** muotoihin, jos se sopii ohjelmaasi paremmin.

pysäytä kaikki ▼

Haaste 3: Ohjelmoi **toinen sokkelo/kenttä** ilmestymään **kun pelihahmo koskee maaliin**. Tämän jälkeen maalin tulee myös vaihtaa paikkaa.

💡 Vinkkejä haasteeseen 3:

↪ Jotta **maali** saadaan reagoimaan **taustan vaihtoon**, voit käyttää **kun taustaksi vaihtuu []** tapahtumaa. Löydät sen **Tapahtumat**-valikosta.



Tällä videolla näet, miltä pelin tulisi näyttää haastetehtävien jälkeen. Klikkaa linkkiä tai kopioi osoite: <https://codeschool.fi/ea12>

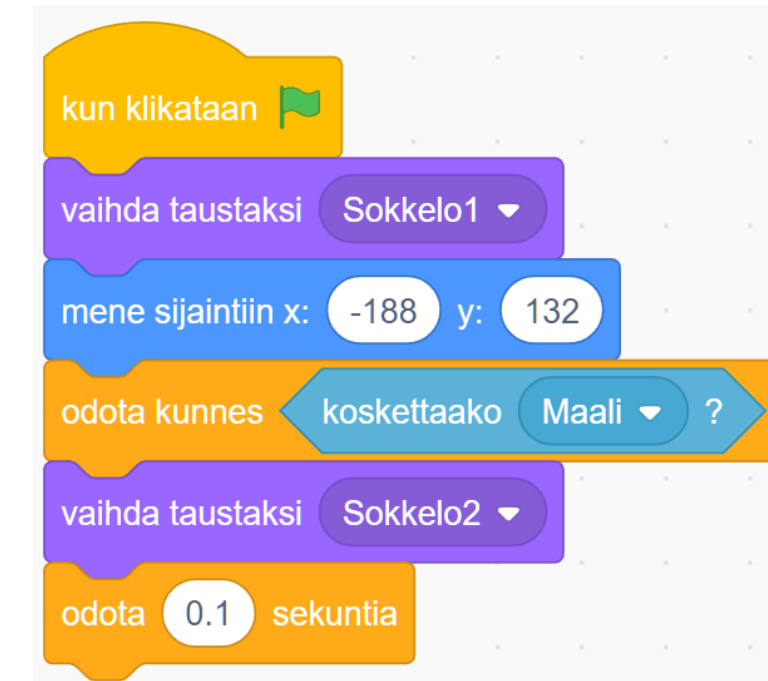
Tästä eteenpäin pelin tekeminen on vain näiden edeltävien vaiheiden toistamista ja omien ideoiden tuomista osaksi pelin ulkonäköä, äänimaailmaa ja toimintaa. Käytännössä sinun täytyy vain tehdä lisää kenttiä ja tehdä pelistä omalaatuinen!

Jos oppilas haluaa ohjata pelihahmoa eri tavalla

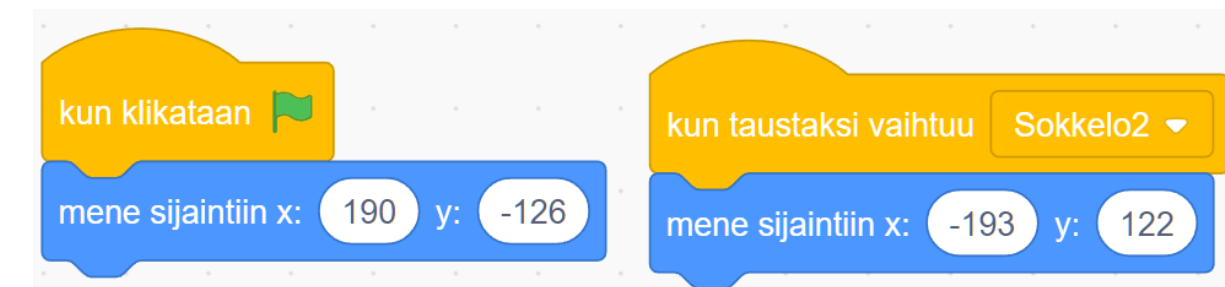
Moni saattaa haluta ohjata pelihahmoa näppäimistöllä. Opettaja voi sallia halutessaan tämän, sillä se ei suuresti vaikuta pelin toimintaan tai ohjelmoinnin haastavuuteen. Pelin toimintalogiikka vain on hieman erilainen.

Haaste 3:

Pelihahmo:



Maali:



Projektityö - Viimeistele sokkelopeli

Nyt on aika jatkaa peliä itse!

On vielä paljon tehtävää. Projektin minimivaatimukset on listattu alle.

Minimivaatimukset projektille

- Tee **vähintään 5 kenttää**
- Pelin tulee **haastavoitua** kenttä kentältä
- Tee **ainakin kaksi liikkuvaa vihollista**
- Tee **useita animaatioita pelihahmolle**
- Tee pelille **voittoruutu**, joka näytetään viimeisen kentän jälkeen
- Ohjelman tulee **resetoitua, kun klikataan vihreää lippua**
- Käytä sopivia **äänitehosteita** ja **taustamusiikkia**
- Pelin tulee loppua **jos pelaaja osuu viholliseen** tai **seinään**

💡 Vinkkejä pelin tekemiseen

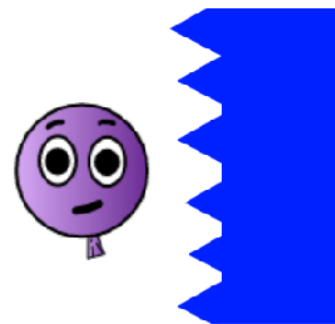
Pelin tarina

➤ Tee yksinkertainen tarina pelille

Suunnittelit pelille aikaisemmin tarinan. Tarina saa muuttua pelin tekemisen aikana paljonkin.

Tarinan ei tarvitse olla suuressa roolissa. Pidä tarina yksinkertaisena. Esimerkiksi pelihahmo voi olla **ilmapallo**, seinät **piikkejä** ja viimeisen kentän maali **surullinen lapsi**, joka on menettänyt ilmapallonsa.

Kunhan tarinassa on hieman **järkeä!**



Projekti - Viimeistele sokkelopeli

Tässä osiossa oppilaat tekevät lopulliset projektinsa. Kyseessä on tämän oppimiskokonaisuuden tärkein oppimistilaisuus ja samalla osoitus oppilaiden taidoista ja tiedoista.

Ajankäyttö

6-8 oppituntia

Työtapa

Työskennellään **1-3** hengen yksiköissä

Avainsanat

Projekti, yhteistyö, vertaisoppiminen, luovuus, avoin ongelmanratkaisu, itsenäinen työskentely

Eri versioita samasta pelistä

Tässä neljä esimerkkiä valmiista pelistä. Jokaisen linkin takaa löytyy täysin sama peli, mutta niissä on käytetty **eri tyyliä ohjelmoida sama logiikka:**

- Perusversio: codeschool.fi/maze-classic
- Tapahtumapohjainen: codeschool.fi/maze-events
- Odotuksiin perustuva: codeschool.fi/maze-wait
- Automatisoitu: codeschool.fi/maze-compact

Tässä sokkelopeli, johon on tehty **hauskoja muutoksia:**

- "Outo" versio: codeschool.fi/maze-weird

3D-tyylinen versio

Tässä vielä esimerkki aivan eri tavalla toteutetusta pelistä, jossa peli vaikuttaa grafiikoiltaan kolmiulotteiselta:

- 3D-tyylinen sokkelopeli: codeschool.fi/maze-3dish

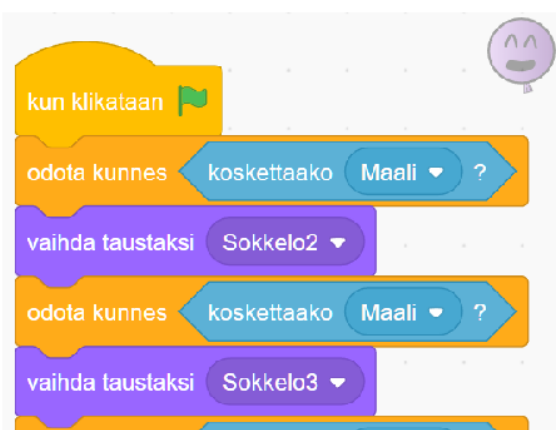
Kentän vaihtumisen ongelmat

⇒ Jos pelisi vaihtaa liian monta kenttää kerralla...

Jos käytit **odota**-lohkoa kentän vaihtoon, peli saattaa hypätä useita kenttiä eteenpäin.

Tämä voidaan estää **lyhyellä odotuksella** kenttien välissä! Näin maali ehtii siirtyä uuteen paikkaan ennen kuin uusi kenttä alkaa.

Kentän vaihto skripti



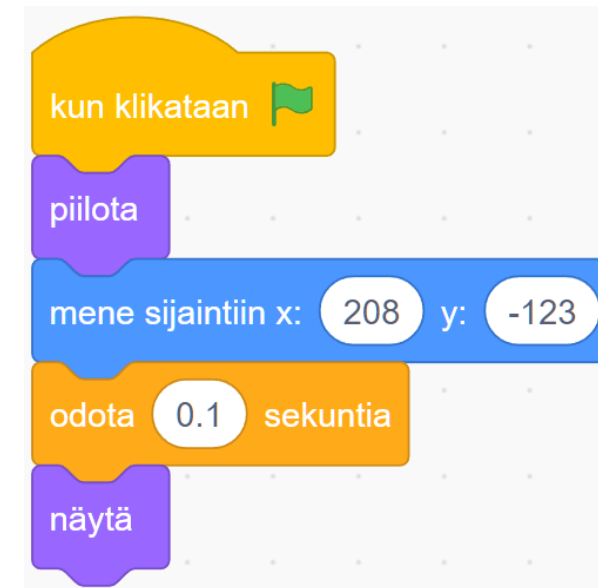
Korjattu versio



Kentän vaihtumisen ongelmat

Oppilaan kirjassa esitettyjen tilanteiden lisäksi oppilaan peli saattaa hyppiä heti alussa kentästä toiseen tai jäädä jumiin. Tämä saattaa johtua siitä, että maalin paikka on heti alussa pelihahmon kohdalla eli kenttä vaihtuu välittömästi pelin alussa. Tämä voidaan korjata helposti piilottamalla maali alussa, siirtämällä se haluttuun paikkaan ja näyttämällä vasta sitten. Väliin voi lisätä halutessa pienen odotuksen, jolla varmistetaan, että pelihahmo ei varmasti ehdi koskea maaliin ennen sen siirtoa aloituspaikalle.

Maalin muokattu skripti sen aloituspaikalle siirtämiseksi:



Projektityö on ongelmanratkaisua

Projekti tarjoaa luonnollisen kontekstin ratkaista ongelmia. Ongelmia ei tarvitse aina osata itse ratkaista. Yhteistyö ja yhdessä pohtiminen ovat tärkeä osa projektityötä.

Oppilaille kannattaa huomauttaa, että ongelmanratkaisu **vie aikaa**. Kun oppilas tuntee, että hän sai itse olla merkittävässä osassa ongelmanratkaisua, hänen itsevarmuus ja halu **ottaa vastuuta** tulevien ongelmien ratkaisussa kasvaa.

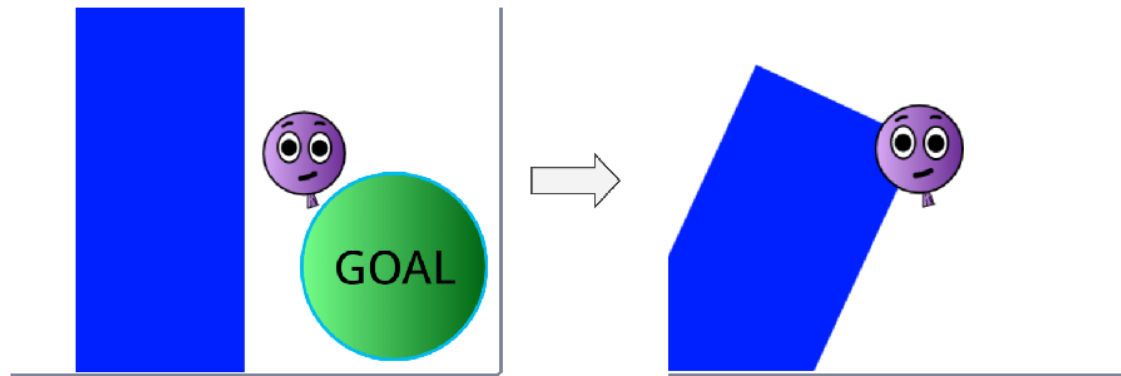
Jos sokkelopeli tehdään opettajan ohjeistamana, se saadaan valmiiksi tunnissa. Kun oppilaat saavat itse suunnitella tuotoksia ja ratkaista ongelmia, heille luodaan mahdollisuus laaja-alaisten taitojen harjoittelulle, luovuudelle ja vastuun ottamiselle omasta oppimisesta. Yksi tunti venyy helposti kymmeneksi tunniksi, mutta kokemuksen arvo on mittaamaton.

Opettajan rooli tässä kuviossa on rohkaista yhteistyöhön, ohjata tiedon ääreen ja motivoida.

Motivoinnista seuraavalla sivulla (kertaus)→

Jos peli häviää maaliin kosketuksesta...

Kun pelihahmo osuu maaliin, uusi kenttä tulee näkyviin. Jos seuraavan kentän seinä on liian lähellä pelihahmoa, **se osuu välittömästi pelihahmoon** kentän vaihdon yhteydessä. Varmista, että **seuraavassa kentässä on riittävästi tilaa** pelihahmolle.



Tässä kuva ongelmasta: Vasemmassa kuvassa pelihahmo pääsee maaliin ja kenttä vaihtuu. Oikeassa kuvassa näkyy, kuinka seuraavan kentän sokkelo on liian lähellä edellisen kentän maalia ja peli päättyy häviöön.

Oppilaiden motivointi projektityössä (kertaus)

Kolme avaintekijää auttavat pitämään oppilaat motivoituneita oppimiskokonaisuuksissa, joissa rakennetaan pitkän ajan saatossa isompia projekteja:

- **Merkityksekkyyys:** Oppilaille osoitetaan, että heidän tuotoksiaan tullaan käyttämään johonkin merkitykselliseen ja ne esitellään toisille. Oppilaat saavat tuotoksistaan palautetta oppimiskokonaisuuden aikana.
- **Itsensä toteuttaminen:** Oppilaat ymmärtävät, että he saavat käyttää luovuutta ja tehdä valintoja oman tuotoksensa suhteen. Oppilailla on riittävästi resursseja oman tuotoksen tekemiseen.
- **Tuki:** Oppilaat eivät tunne olevansa yksin, kun he tarvitsevat ohjausta. Oppilaita rohkaistaan yhteistyöhön ja sosiaalinen ympäristö pidetään turvallisena. Oppilas tuntee, että virheitä ei tarvitse pelätä eikä niistä rankaista.

Ehdotuksia merkityksellisyyden edistämiseksi (kertaus)

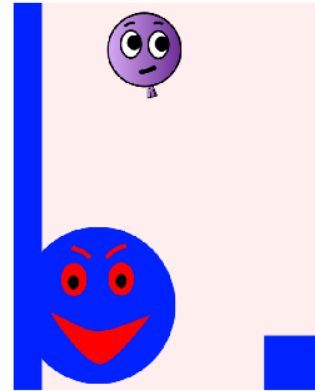
- Pelit **esitellään ja toisten pelejä testataan**. Oppilaat antavat positiivista palautetta ja kehittämissuhteita (rakentava palaute).
- Pelit esitellään **rinnakkaisluokalle tai nuoremmille oppilaille**. Oppilaat antavat positiivista palautetta ja kehittämissuhteita (rakentava palaute).
- Oppilaat **äänestävät parhaista tuotoksista** seuraavissa kategorioissa:
 - Paras ulkonäkö
 - Hauskin peli
 - Paras kokonaisuus

Voittajat saavat pienet palkinnot.

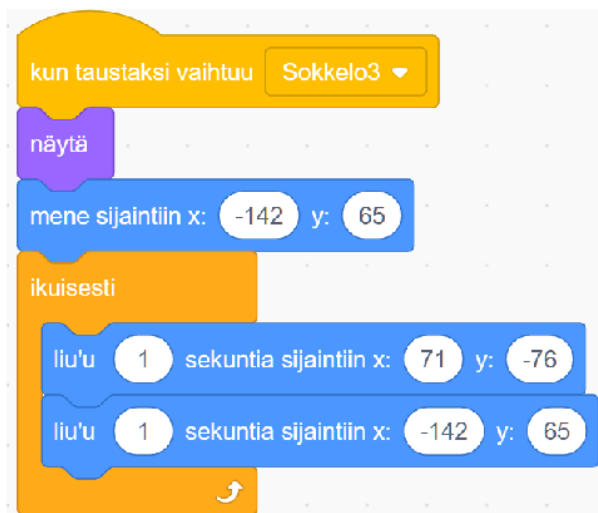
Liikkuvat viholliset

⇒ Yksinkertainen vihollinen

Liikkuvat viholliset tekevät pelistä jännittävän! Kun teet **uuden hahmon**, joka on **väritään sama kuin sokkelon seinät**, se voi toimia pelissä vihollisena. Hahmon voi sitten laittaa liikkumaan halutulla tavalla. Kuten maalille, myös viholliselle pitää ohjelmoida, mitä se tekee kussakin kentässä esimerkiksi **kun taustaksi vaihtuu []** tapahtuman avulla.



Liikkuu kahden pisteen välillä

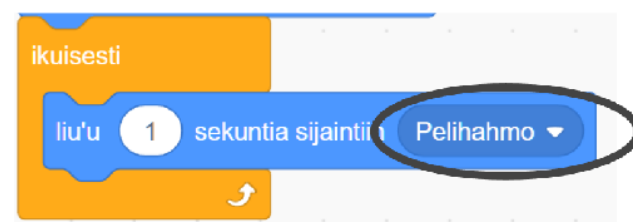


Liikkuu satunnaisiin paikkoihin



⇒ "Älykäs" vihollinen

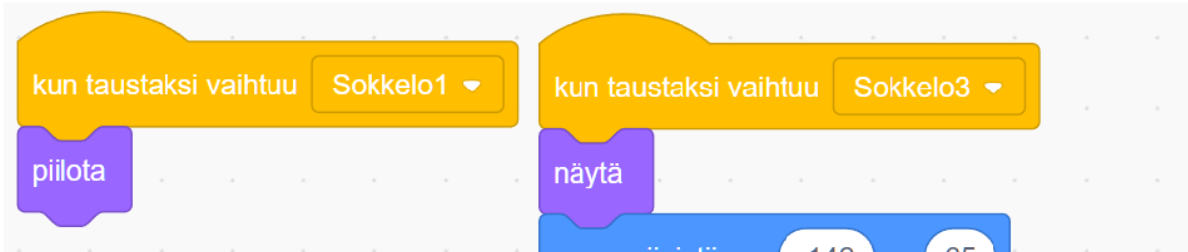
Jos "**satunnainen sijainti**" tilalle vaihdetaan "**Pelihahmo**", vihollinen alkaa jahtaamaan pelihahmoa.



Ohjaa hahmoja tapahtumilla

⇒ Piilota ja näytä

Jos haluat, että vihollinen tulee peliin vasta kolmannessa kentässä, käytä **näytä** ja **piilota** -lohkoja.



⇒ Sama hahmo eri kentissä

Jos haluat, että sama hahmo käyttäytyy eri tavalla eri kentissä, täytyy edellinen skripti lopettaa ennen seuraavan aloittamista. Tähän sopii **pysäytä [hahmon muut koodit]** -lohko.



Animaatiot

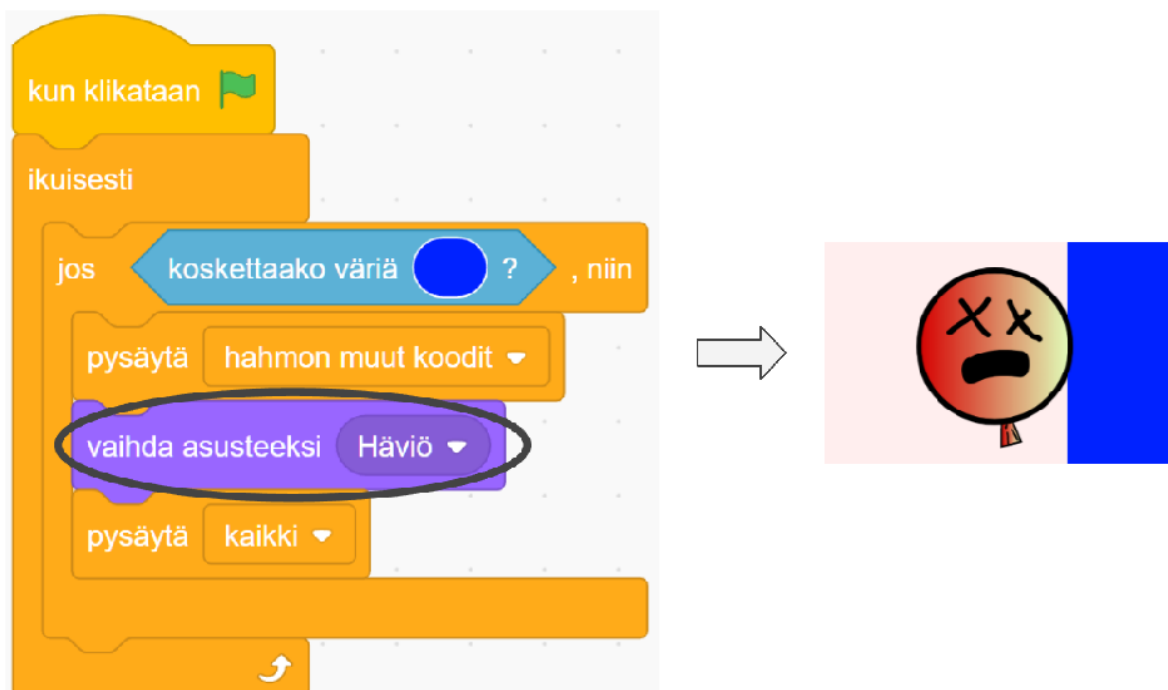
Ennen animointia

Animointi tapahtuu tekemällä kopioita asusteista ja muokkaamalla asusteita halutulla tavalla. Ennen animointia sinun täytyy siis olla varma, että olet **tyytyväinen hahmon ulkonäköön**. Sen jälkeen voit klikata asustetta hiiren oikealla painikkeella, **monistaa** hahmon asusteen ja tehdä uudesta asusteesta halutun näköisiä.



Häviön animointi

Yksinkertainen esimerkki animoinnista on asusteen vaihtaminen häviön yhteydessä. Tämä tapahtuu vaihtamalla hahmon asusteen kun hahmo koskee seinää.



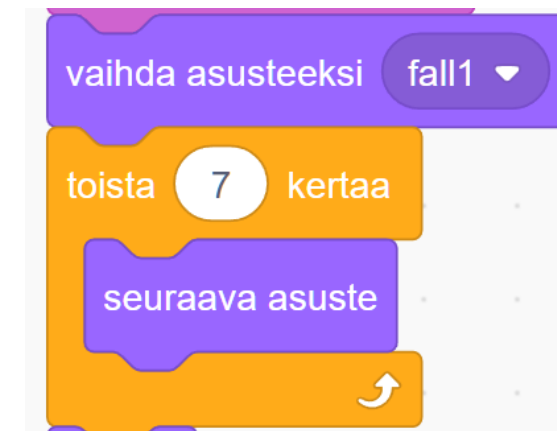
Animaatiot

Mitä enemmän kuvia animaatioissa on aikayksikköä kohden, sitä sulavampi animaatio on. Esimerkiksi elokuvissa on pitkään näytetty 24 kuvaa per sekunti. Kuvataajuudesta käytetään puhekielessä termiä fps (frames per second), joka tyypillisestä halutaan saada moderneissa peleissä lähemmäs lukua 60.

Jos Scratchissa halutaan suhteellisen sulavia animaatioita, kannattaa pyrkiä tekemään isommalle animaatiolle ainakin 12 asustenvaihtoa (12 erilaista asustetta). Alla esimerkki animaatiosta, jossa hyvin nopea tapahtuma esitetään kahdeksalla asusteella. Animaatio tapahtuu kun klikataan jäälauttaa.

- Pingviini putoaa veteen: codeschool.fi/anim-penguin

Mitä vähemmän asusteita, sitä nopeampi animaatio on. Tätä voidaan kompensoida laittamalla lyhyitä, 0.01s odotuksia, asusteen vaihtojen väliin, mutta tämä saattaa helposti saada animaation näyttämään pätkivältä.

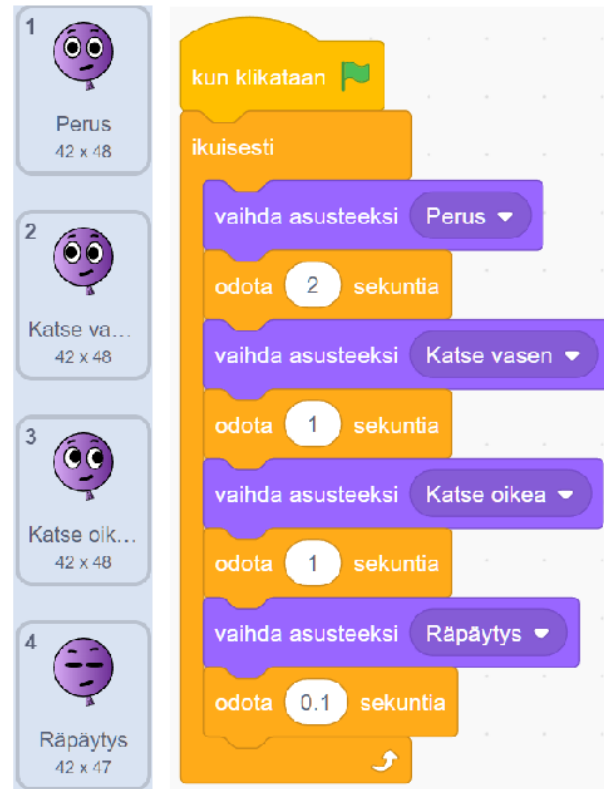


Tietty animaatio saadaan helposti tapahtumaan oikein, kun asetetaan oikea ensimmäinen asuste ja vaihdetaan asustetta niin sopivan monta kertaa toistorakenteella. 8 asusteen animaatio saadaan tapahtumaan kokonaan, kun asetetaan ensimmäinen asuste ja vaihdetaan asustetta 7 kertaa.

Luoppaava animaatio

Peli tuntuu eloisammalta, kun pelihahmolla on **jatkuvasti toistuva animaatio**.

Esimerkiksi neljällä hieman erilaisella asusteella, **vaihda asusteeksi ()** -lohkolla ja **odota** -lohkolla saadaan tehtyä toistuva animaatio, kun käytetään ikuista toistoa.



Graafiset tehosteet

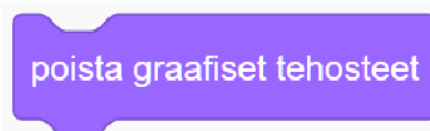
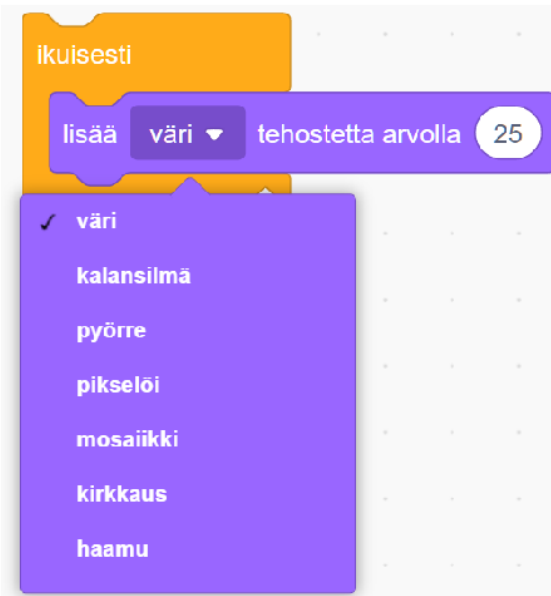
Käytä tehosteita lohkolla

Hahmojen ulkonäköön voi vaikuttaa myös ilman asusteiden vaihtamista käyttämällä graafisia tehosteita **lisää [väri] tehostetta arvolla ()** -lohkolla.

Väri vaihtaminen ei ole ainoa käytössä oleva tehoste. Tehosteita on helppo kokeilla valitsemalla tehosteen vetovalikosta ja klikkaamalla tehostetta. Voit myös laittaa lohkon ikuisen toistoon.

Täällä vielä hauskoja esimerkkejä tehosteiden käytöstä yhdessä äänien kanssa: <https://codeschool.fi/ea13>

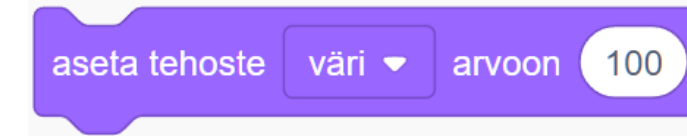
Graafiset tehosteet voi poistaa käyttämällä **poista graafiset tehosteet** -lohkoa.



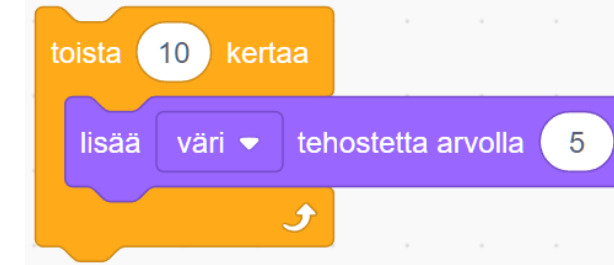
Graafiset tehosteet

Animoinnin lisäksi tehosteiden käyttämisellä voidaan elävöittää peliä:

Aseta tehoste [] -lohkolla asetetaan tehosteen vahvuus suoraan tiettyyn arvoon:



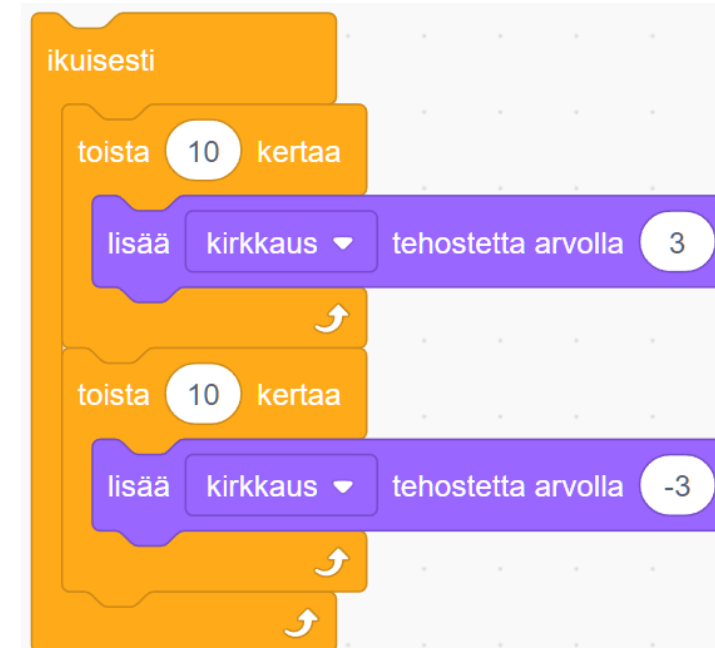
Lisää [] tehostetta -lohkolla voidaan muuttaa arvoa tietyn arvon verran. Yhdistämällä tämän määrättyyn toistoon, saadaan tehosteen muutoksesta liukuvia.



Voidaan myös käyttää ikuista toistoa, jolloin tehosteen muutos on jatkuvaa:



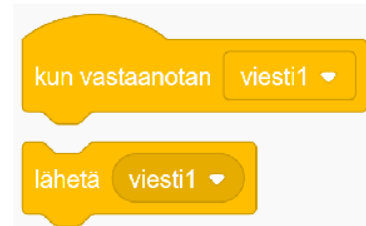
Kun tehostetta liu'utetaan tiettyyn arvoon ja takaisin, saadaan näyttäviä sykkiviä tehosteita:



Viestien käyttö (**haastava**)

⇒ Hahmot voivat kommunikoida viesteillä

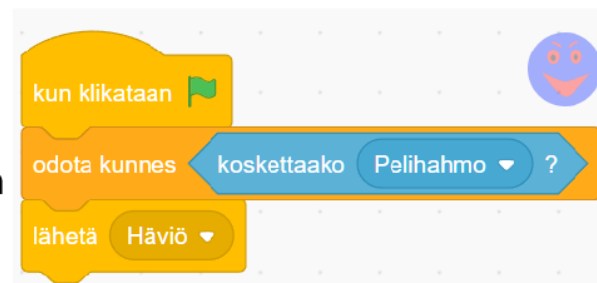
Tähän asti pelin tapahtumia on ohjattu taustan vaihdoilla. Hahmot voivat myös käskää toisiaan lähettämällä viestejä. Viesti-lohkot löydät **Tapahtumat**-valikosta.



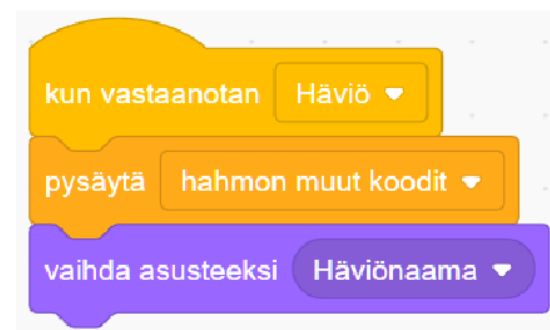
⇒ Pelin hallinta viestejä käyttämällä

Pelin etenemisen voi ohjelmoida myös täysin viesteillä, ilman taustan vaihtoon liittyviä tapahtumia.

Esimerkiksi **vihollinen** voi lähettää viestin "Häviö", kun pelihahmo osuu viholliseen. Tämä viesti lähtee kaikille hahmoille.



Kukin hahmo voi reagoida viestiin **kun vastaanotan [Häviö]** -tapahtumalla. Jos käytämme viestejä tällä tavalla, vihollisen ei tarvi enää olla samanvärinen kuin sokkelon, sillä se reagoi suoraan pelihahmon kosketukseen.

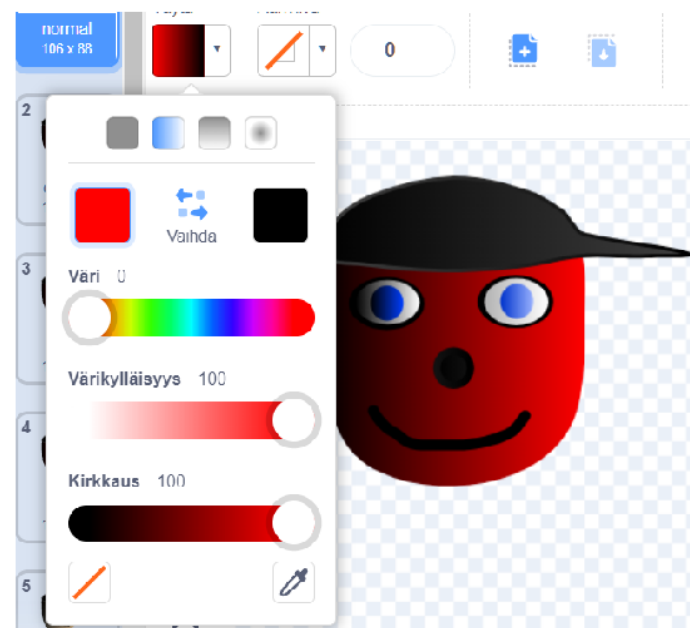


Syvyysvaikutelmaa varjoilla (**haastava**)

⇒ Liukuvärit

Liukuvärien avulla, voimme tehdä peliin varjostuksia. Löydät **tarkemmat ohjeet** liukuvärien käyttöön **sivulta 49**.

Jos teet varjostukset hahmoille, kannattaa ne tehdä myös sokkeloille!



Viestien käyttäminen

Sokkelopelin eteneminen nojaa taustojen vaihtoon. Hahmot voivat kuitenkin kommunikoida myös viestien välityksellä, jolloin kommunikointia varten ei tarvitse vaihtaa taustaa.

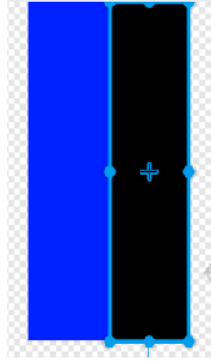
Syvyysvaikutelmaa varjoilla

Opettaja voi esitellä oppilaille valmiin projektin, jossa syvyysvaikutelmaa on haettu varjoja käyttämällä. Varjot on saatu käyttämällä mustaa toisena värinä liukuvärissä.

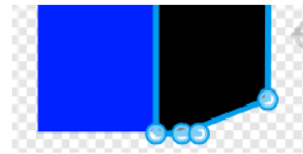
- codeschool.fi/maze-3dish

⇒ Sokkelon varjot

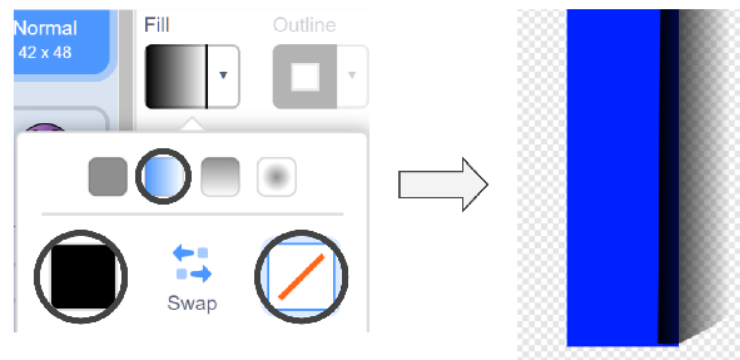
1. Tee **suorakulmio** käyttämällä □ -työkalua



2. **Uudelleenmuotoile** se ↻ -työkalulla



3. Täytä suorakulmio liukuvärillä 🎨 -työkalun avulla. Käytä **mustaa** ja **tyhjää** väriä yhdessä **poikittaisen liukuvärin** (▭) kanssa.



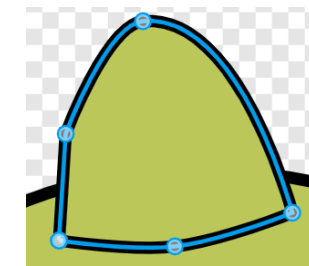
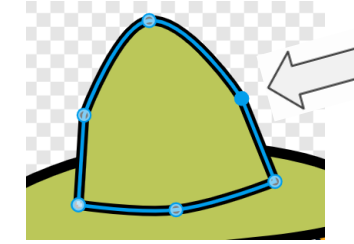
4. Lopuksi, valitse ↻ -työkalulla tekemäsi objekti. Nyt voit viedä objektin seinän alle klikkaamalla **Taaksepäin** painiketta, kunnes tekemäsi varjo on seinän alla. Kyseinen painike on piirtoalueen yläpuolella



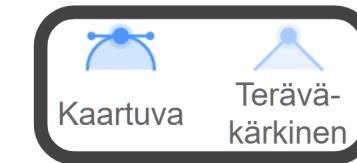
Muotoilutyökalu (kertaus)

Muotoile uudelleen-työkalulla saadaan esille objektien muotoilupisteet. Niiden liikuttamisen lisäksi voit tehdä seuraavaa:

- **Poistaa** pisteen kaksoisklikkauksella



- Valita, miten piste muuttaa muotoa (kaartuva vai terävä)



Valot ja varjot

Voit tehdä todella näyttäviä sokkeloita ja pelimaailmoja, kun vain jaksat rohkeasti kokeilla erilaisia ideoita!

Alla esimerkki kolmiulotteiselta vaikuttavasta sokkelopelistä. Voit pyytää opettajalta linkin kyseiseen peliin.



Jos haluat tehdä sokkelon hahmona, voit tehdä varjot erillisenä hahmona.



 **Valmistaudu loppukokeeseen!** (Opettaja tulostaa)

Tasot näyttämöllä

Aikaisemmin käsiteltiin piirtoalueen tasoja, mutta myös näyttämöllä on tasot.

Hahmo voidaan siirtää suoraan taakse tai eteen **mene [etu/taka] alalle** -lohkolla. Hahmo voidaan siirtää tietyn verran eteen- tai taaksepäin käyttämällä **mene () tasoa [eteenpäin/taaksepäin]** -lohkoa.



Huomio: Vertaisarviointiin siirrytään vasta kun projektit on tehty.

Vertaispalaute

Ajankäyttö

25 min

Työtapa

Vertaispalaute täytetään projektiryhmissä.



Vertaispalaute: Miten meillä meni?

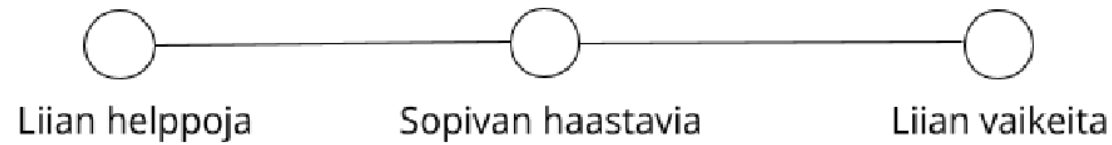
Tulosta tai jaa vertaispalaute lomake oppilaiden kanssa (erillinen dokumentti). Oppilaita rohkaistaan keskusteluun ja rakentavaan palautteeseen.



Lopuksi!

Vastaa seuraaviin kysymyksiin tehtyäsi kaikki kirjan harjoitukset:

1. Ohjelmointitehtävät olivat ... (keskustele vastauksista parin kanssa!)



2. Kirjoita **kolme positiivista** kommenttia omasta työskentelystäsi.

3. a) Työskentelitkö parhaasi mukaan?

Kyllä En

b) Miksi / Miksi et?

4. a) Kuuntele tai tarkista sivulla 35 tekemäsi lupaus. Pystyitkö pitämään sen?

Kyllä En

b) Miksi / Miksi et?

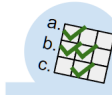
Lopuksi

Ajankäyttö

25-45 min

Työtapa

Työskennellään **2-3** hengen yksiköissä. Rohkaistaan keskustelemaan.



Arviointi

Loppukoe

Opettaja voi halutessaan pitää oppilaille kokeen (erillinen dokumentti), jonka oppilaat voivat ristiin arvioida opettajan ohjeiden mukaisesti.

Kustakin tehtävästä saa maksimissaan 10 pistettä.

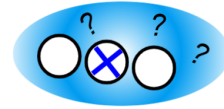
Loppukoe - Oikeat vastaukset

1	Valitse tosi , epätosi tai en tiedä	Tosi	Epätosi	En tiedä
	Scratch ohjelma alkaa klikkaamalla painiketta -----	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Scratchissa ohjelmoidaan antamalla komentoja hahmoille	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Toistorakenteelle tarvitaan aina ehto -----	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Kun mikä tahansa skripti on suoritettu loppuun, koko --- ohjelma päättyy	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Ohjelmoinnissa ehto voi olla vain <i>tosi</i> tai <i>epätosi</i> -----	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

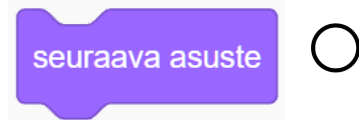
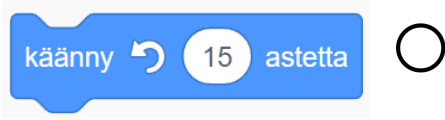
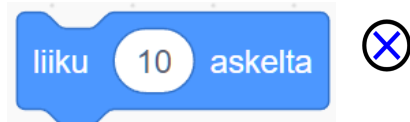
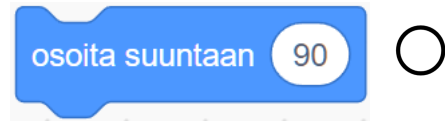
(Oikea vastaus antaa 2 pistettä. Opettaja voi päättää myös, että väärästä vastauksesta vähennetään yksi piste)

2

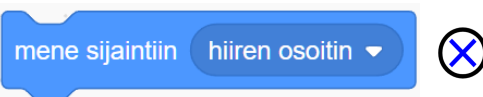
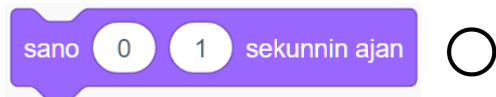
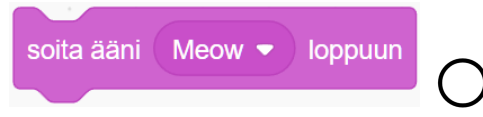
Valitse **oikea vaihtoehto** kuhunkin kysymykseen



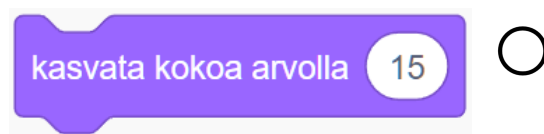
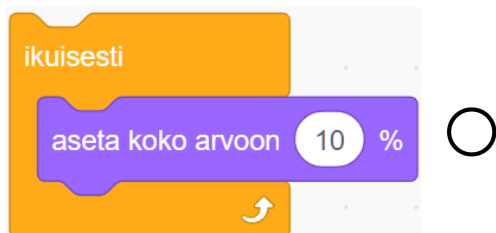
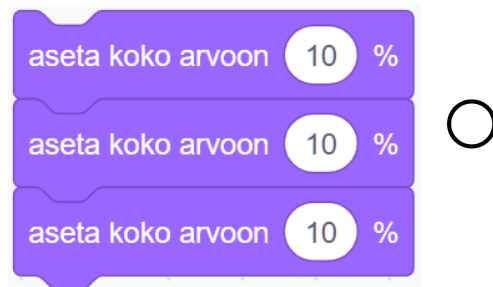
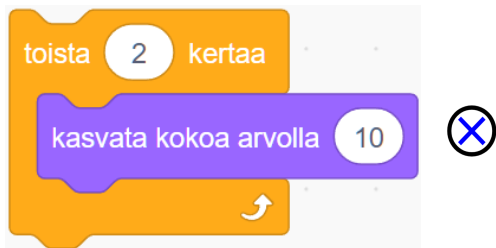
a) Mikä komento muuttaa hahmon paikkaa?



b) Mikä komento suoritetaan loppuun nopeiten?



c) Mikä skripti kasvattaa hahmon kokoa eniten?



(Tehtävissä a ja b oikea vastaus antaa 3 pistettä. Tehtävässä c oikea vastaus antaa 4 pistettä)

3

Ohjelmoi Scratchissa **kissa ja hiiri** -peli.

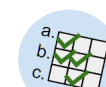
Vaatimukset:

- Tee hiiri, joka **alkaa seuraamaan hiiren osoitinta viiden askeleen nopeudella** sekunnin päästä ohjelman aloittamisesta.
- Tee kissa, joka **alkaa seuraamaan hiirtä viiden askeleen nopeudella** sekunnin päästä ohjelman aloittamisesta.
- **Kun kissa osuu hiireen, hiiri sanoo "Auts!"**.
- Pelissä on näillä säännöillä selvästi parannettavaa. **Tee peliin parannuksia**. Saat muuttaa myös pelin sääntöjä.

Ratkaisu:

- **Esimerkkiohjelma:** <https://codeschool.fi/9n87>
- Kissan nopeus ei voi olla sama kuin hiiren. Muuten peliä ei voi pelata muutamaa sekuntia kauempaa.
- "Resetoitumisella" tarkoitetaan tässä sitä, että hahmoille täytyy antaa aloituspaikat.

(Jos ohjelma toimii ja resetoituu, annetaan 10 pistettä. Jos ohjelma vaatii korjausta, annetaan 5 pistettä. Jos ohjelmassa ei ole yhtään koodia, ei anneta pisteitä)



Arviointi

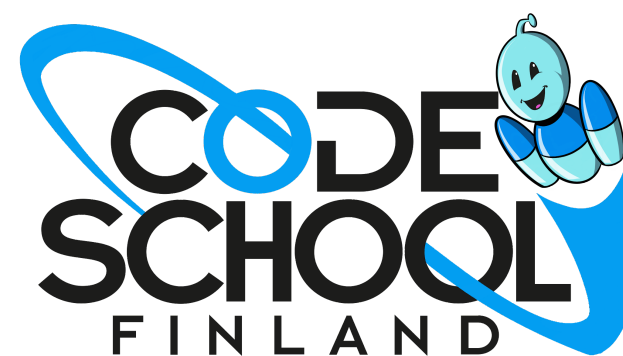
Arvioinnista

On tärkeää, että aloitteleva oppijan kyvykkyyttä ei välttämättä heti ensimmäisessä oppimiskokonaisuudessa arvioida kovin arvosanoin. Tärkeämpää on, että oppilas saa oppimisen ja antamansa panoksen kannalta olennaista palautetta opettajalta ja vertaisilta sekä pohtii omaa oppimistaan.

Mikäli opettaja kuitenkin haluaa antaa asteittaisen arvosanan (esim. numeroarvosanan), voi hän käyttää esimerkiksi alla olevaa arviointisuunnitelmaa.

Esimerkki arviointisuunnitelmasta

Prosenttia kokonaisarviointista	Assessment task
25 %	Tuotoksen arviointi: Interaktiivinen tarina
25 %	Tuotoksen arviointi: Sokkelopeli
25 %	Aktiivisuus ryhmätyöskentelyssä
25 %	Loppukoe



www.codeschool.fi

ISBN: 978-952-7403-29-7

