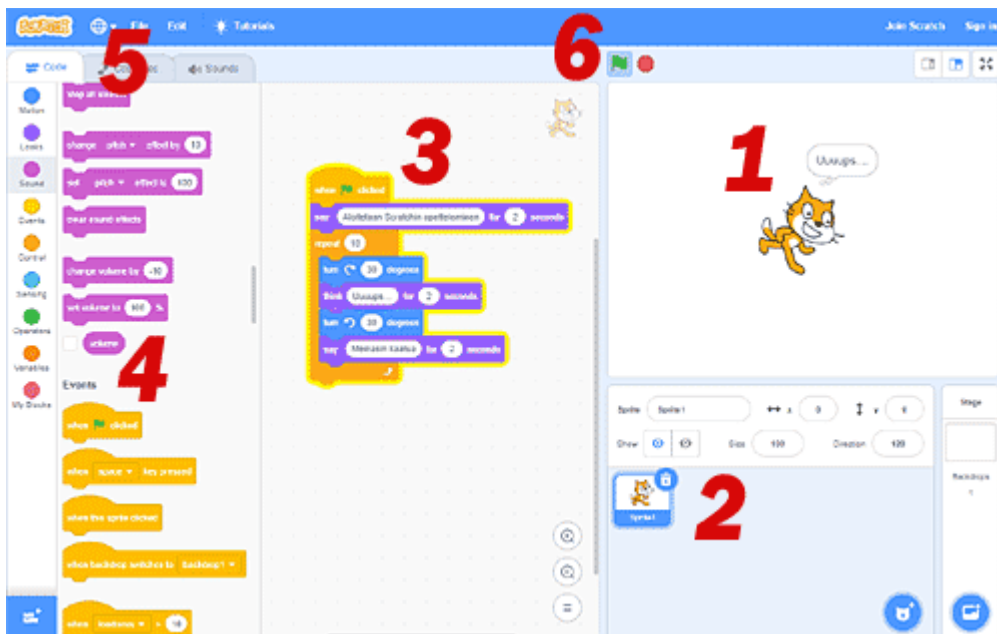


Ohjelmoinnin perusteet – Scratch

Scratch on MIT-yliopistossa kehitetty ohjelmointiympäristö, jonka avulla on helppoa opetella ohjelmoinnin toimintaperiaatteita. Ohjelma on kehitetty ohjelmoinnin tutustumiseen oppitunneilla. Siitä on pyritty tekemään mahdollisimman helppo ohjelmointiympäristö kai ken ikäisille ja ohjelmointi perustuukin graafisten lohkojen siirtelyyn ja järjestelyyn. Ohjelmasta löytyy valmiita kirjastoja, joista voi hyödyntää kuvia, ääniä ja taustoja omiin ohjelmiin ja projekteihin. Jopa lapsetkin oppivat tällä työkalulla tuottamaan pieniä animaatioita ja pelejä. Näitä taitoja voi myöhemmin hyödyntää Pythonin ja Javan opiskelussa. Scratchin Offline Editorin voi ladata Windows-, Mac OS ja Linux-käyttöjärjestelmille.

Scratchista on myös nettisivulla oleva verkkoeditoriversio, jota voi käyttää suoraan asentamatta koneelle. Kirjautumalla voi omia töitä tallentaa myöhempää käyttöä varten. Scratchilla on lisäksi omat kotisivut, jonne on kerätty muiden tekemiä projekteja. Rasbian käyttöjärjestelmässä Raspberry Pi:ssä on Scratch jo valmiiksi asennettu ja tällä laitteella voidaan ohjata myös muita lisälaitteita GPIO-piirien avulla.



1 - Esiintymislava - alue jonka ohjelmaa käyttävä tulee näkemään. Voit nähdä toiminnan ohjelmastasi tällä alueella ja siihen voi sijoitella hahmoja ohjelmoinnin aikana.

2 - Hahmot - Tähän tulee näkyville ohjelmassa käyttämäsi hahmot. Voit nimetä hahmot ja muuttaa niiden asetuksia täältä.

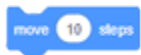
3 - Ohjelmointialue - tällä alueella voit liittää ohjelmapalikoita toisiinsa ja tuottaa tarvittavia toimintoja ohjelmaan. Ohjelmapalikoita voidaan myös muokata haluamallasi tavalla täällä.

4 - Koodikirjasto/lohkot – Esiohjelmoituja ohjelmapalikoita, joita voidaan ottaa käyttöön ohjelmassasi. Raahaa haluamasi toiminto koodit-alueelle ja laita ne haluamaasi järjestykseen.

5 - Välilehdet - voit valita eri näkymiä asusteet, äänet ja koodi-välilehtien avulla. Asusteet-välilehdessä voit luoda uusia hahmoja ja äänet-välilehdessä voit luoda musiikkia ja ääniä ohjelmaan.

6 - Vihreä lippu - Tällä lipulla voit käynnistää ohjelman ajon ja kokeilla koodin toimivuutta.

Koodikirjaston ryhmät:



Liike - näillä palikoilla ohjataan hahmojen liikkumista joko ohjelman mukaan tai hiiren kosketuksesta.



Ulkonäkö - tällä osalla voidaan muokata graafisia toimintoja, muokata liikettä, vaihtaa asusteita ja säätää puhetta.



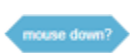
Ääni - ohjelmassa on usein äänet tärkeässä osassa. Tämän avulla voit säätää ääniä oikeaan hetkeen tai vaikka taustamusiikkia ohjelmaasi.



Tapahtumat - tällä palikalla on hyvä aloittaa ohjelmakoodit, koska ohjelmilla on hyvä olla jonkinlainen aloitustoiminto, kun ne käynnistetään.



Ohjaus - Silmukat ovat ohjelmoinnissa tärkeä elementti. Joskus liikettä pitää toistaa useaan kertaan, kunnes toivottu lopputulos saavutetaan. Toistopalikoihin voidaan myös määrittää arvoja, kunnes toiminto pysäytetään, eli ehtoja toiminnan jatkamiselle.



Tuntoaisti - Tämä osio havaitsee hahmojen koskettamiset toisiinsa. Tällä palikalla voidaan havaita myös hiiren liikkuminen ilman klikkaamista.



Toiminnot - Tämän osion avulla voidaan laskea tuloksia yhteen tai luoda jotain satunnaisia elementtejä. Nämä ovat matemaattisia toimintoja. Tämän toiminnan avulla voidaan luoda vaikka arpakuutio peliin.



Muuttujat - Muuttujat ovat myös tärkeä osa ohjelmointia, koska ohjelma joutuu useasti päättelemään toimintaansa jonkun muun tiedon pohjalta. Muuttujan tietoa voidaan muuttaa ohjelman aikana ja tulosta voidaan hyödyntää jossain toisessa ohjelman toiminnassa. Tämä on eräänlainen muistilapputoiminto ohjelmoinnissa.

Lohkoni - Tällä toiminnolla voidaan tallettaa erilaisia ohjelmasarjoja, jota voidaan ottaa käyttöön halutessaan.

Scratch ohjelmistorakenteet

Toistorakenne - tällä toiminnolla toistetaan määrättyä koodirakennetta, jotta sitä ei tarvitse kirjoittaa monta kertaa peräkkäin. Toistojen määrä riippuu käytetystä rakenteesta.

- **For-rakenne** - käytetään, kun tiedetään, kuinka monta kertaa asia toistetaan. Scratchista se löytyy repeat/toisto-rakenteena. Tällä voidaan ilmoittaa, kuinka monta kertaa toistoa tapahtuu ohjelmassa.
- **While-rakenne** - Käytetään silloin kun on epävarmaa tai ei tiedetä etukäteen toistojen määrää. Tämä tarvitsee aina ehdon, milloin toisto keskeytetään. Ehtona käytetään totuusarvoa, eli onko arvo tosi tai epätosi. Jos tosi, suoritus keskeytetään. Totuusarvo asetetaan while-rakenteen sisälle. While-rakenne löytyy komentona repeat until/toista kunnes.
- **Forever-rakenne** - Käytetään silloin kun halutaan tapahtumatoistoja rajattomasti.

Ehtorakenne - sisältävät ehtoja koodin suorittamiseen

- **If-lause** - lauseen sisällä oleva koodi suoritetaan, jos määritelty ehto on tosi. Else-lauseen koodi sisällytetään lopuksi siltä varalta, jos if-lause ei toimi.
- **Vertailu-/loogiset operaattorit** - tällä voidaan yhdistää useita lausekkeita yhdeksi totuusarvoksi. Nämä antavat mahdollisuuden vertailuihin. Vertailun lähtökohtia saadaan

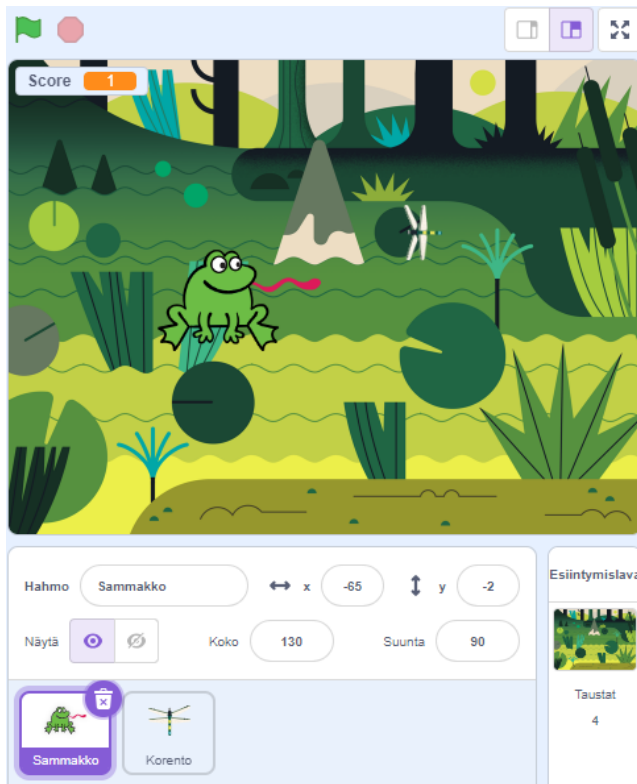
esimerkiksi muuttujien arvoista. Vertailuoperaattorilausekkeina toimivat pienempi kuin, suurempi kuin, yhtä suuri kuin ja erisuuruisuus. Lisäksi loogisina operaattoreina toimivat ja, tai ja ei. Myös kahta eri operaattoria voidaan käyttää verrannossa, kuten suurempi kuin ja yhtä suuri kuin.

Muuttujat– ohjelma varaa muistista paikan muuttujalle, jolloin arvoa voi muuttaa ohjelman edetessä. Tätä arvoa käsitellään muuttujan nimellä. Muuttujia voidaan muuttaa yhteen tai vähennyslaskulla tai tutkia sanojen rakennetta. Muistipaikassa oleva muuttujan arvo voidaan korvata aina uudella tai sitä voidaan muuttaa. Viimeisin muutos jää aina voimaan.

Online Scratch (scratch.mit.edu)

Scratch-ohjelmasta löytyy verkossa online-testausalusta. Mene osoitteeseen scratch.mit.edu ja klikkaa painiketta Aloita luominen. Voit myös halutessasi kirjautua sivuille, niin voit tallentaa töitäsi myöhempää käyttöä varten.

Sammakkopeli



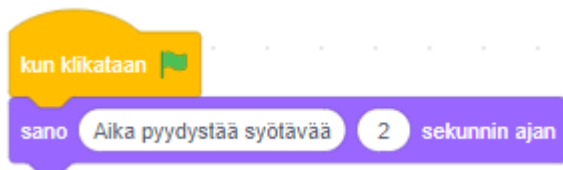
Teemme sammakkopelin, jossa voimme harjoitella ohjelmoinnin eri rakenteita ja koodikirjaston ryhmiä. Hae ohjelman oikealla alareunassa olevista painikkeista haluamasi eläimet, eli hahmot. Voit poistaa alkuperäisen hahmon kuvion yläreunasta.



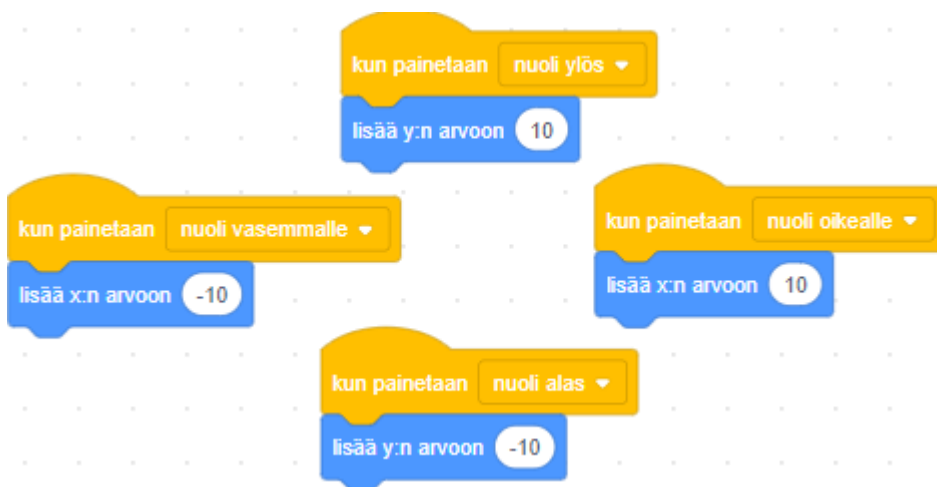
Anna hahmoille haluamasi nimi. Voit myös muuttaa sen kokoa/suuntaa aktivoituasi hahmon klikkaamalla sen kuvaketta. Hae myös pelille sopiva tausta klikkaamalla taustakuvaketta.

Aluksi luomme pelille käynnistyskäskyn, jotta saamme sen käyntiin haluamallamme tavalla. Tähän käytämme Tapahtumat-ohjelmalikkua. Aktivoi sammakko ensimmäiseksi ohjelmoitavaksi

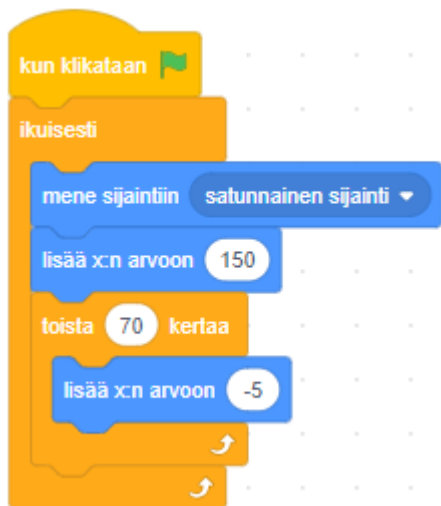
hahmoksi. Valitsemme ”kun klikataan” palikan, niin saamme ohjelman käynnistymään. Vedä kyseinen palikka koodikirjastosta ohjelmointialueelle. Seuraavaksi laitamme Sammakon sanomaan Aika pyydystää syötävää, jotta tiedämme pelin alkaneen. Vedä Ulkonäköpalikka sano Hei! 2 sekunnin ajan. Muuta teksti haluamaasi muotoon.



Seuraavaksi ohjelmoimme sammakon liikkumaan näppäinnuolien avulla. Vedä neljä ”kun painetaan” tapahtumapalikkaa ohjelmointialueelle. Valitse pudotusvalikostavälilyönti-tilalle eri nuolen suunnat. Vedä jokaiselle oma liikepalikka. Vasemmalle ja oikealle ”lisää x:n arvoon”-palikka ja ylös/alas ”lisää y:n arvoon”-palikka. Kirjoita haluamasi liikkeet arvoruutuun.



Nyt tarvitsemme liikkuvan hahmon, mitä sammakko pyydystää. Aktivoi korento-hahmo klikkaamalla hahmon kuvaketta. Myös tämä käynnistetään ”kun klikataan” palikalla. Seuraavaksi laadimme hahmolle liikkeen silmukan avulla. Vedä ohjaus-ryhmästä palikka ”ikuisesti”. Määritellään hahmolle aloituspaikka liike-ryhmästä ”mene sijaintiin satunnainen sijainti”. Näin saamme hahmon ilmaantumaan joka kerralla satunnaisesta paikasta. Annetaan myös käsky aloittaa kentän oikeasta reunasta ”lisää x:n arvoon” palikalla. Muuta arvo haluamaasi lukemaan. Luodaan tämän jälkeen hahmolle liike toistorakenteella. Käytä toista x kertaa-palikkaa. Kirjoita haluamasi toistokerta, joka vastaa hahmon kulkemaa matkaa. Hahmon vauhti voidaan määrittää jokaisen toiston pituudella. Käytä liike-palikkaa ”lisää x:n arvoon”.

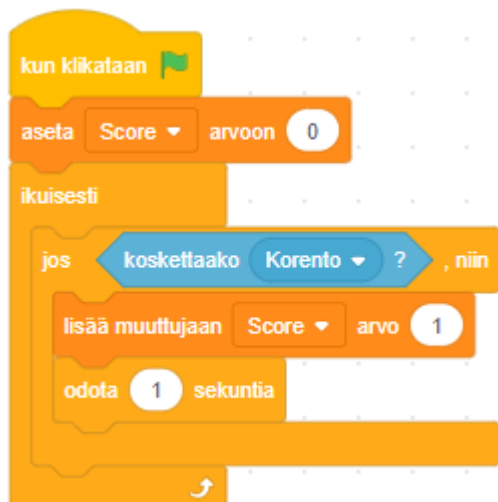


Nyt voit kokeilla pelin toimivuutta käynnistämällä peli lipusta esiintymislavan päältä.



Lopuksi teemme pelille pisteruudun, josta voimme seurata napattujen korentojen määrän. Aktivoi sammakko-hahmo kuvakkeesta. Aloitetaan käynnistämällä pisteenlasku "kun klikataan" palikalla. Luodaan muuttuja, joka voi pitää kirjaa pisteistä. Mene ryhmälistan loppuun kohtaan muuttujat ja klikkaa painiketta "Tee muuttuja". Anna uudelle muuttujalle nimi "Score" ja hyväksy ok-painikkeesta. Nyt pitäisi pelin yläreunaan tulla pisteenlaskuruutu. Vedä samasta ryhmästä "aseta muuttujani arvoon 0" palikka asettaaksesi lähtöpisteet. Valitse pudotusvalikosta "Score" palikkaan muuttujaksi. Seuraavaksi teemme jälleen silmukan "ikuisesti"-palikalla, koska emme halua pistelaskun pysähtyvän. Silmukan sisään laitamme jos-ehtolausekkeen. Valitse ohjausvalikosta palikka "jos xxxx, niin" ja vedä tuntoaisti-valikosta palikka "koskettaako hiiren osoitin?" jotta ohjelma tunnistaa hahmojen kosketuksen. Vaihda "hiiren osoitin" hahmon nimeksi pudotusvalikosta. Näin ohjelmalle kerrotaan milloin koodin pitäisi käynnistyä. Vedä seuraavaksi "lisää muuttujaan muuttujani arvo 1"-palikka, jotta koskettaessaan hahmoa, muuttujan arvo kasvaa yhdellä. Vaihda taas muuttujaksi "Score" tähänkin palikkaan. Liitä myös toiminto "odota 1 sekuntia", jotta pisteitä ei tulisi liikaa yhdellä kerralla.

Tee muuttuja



Nyt peli on valmis pelattavaksi. Jos olet tehnyt koodauksen oikein, sinulla on toimiva peli. Voit hienosäätää peliä ohjelmistokoodin arvoja muuttaen, myös pelin aikana. Hyviä pelihetkiä!