Sphero labyrintissa

Tutustumme ohjelmoinnin termistöön ja opimme,
kuinka robotteja voidaan ohjata labyrintissa JavaScript-ohjelmointikieleen perustuvalla lohko-ohjelmoinnilla.

**Ohjelmoinnin käsitteistä (luokkahuone):**

Tutustutaan ensimmäiseksi hieman ohjelmoinnin eri osa-alueisiin. Vastatkaa ryhmissä oman aiheenne kysymyksiin. Tarvitsette näitä tietoja myöhemmin Sphero-pallon kanssa ratkaistavassa labyrintissa.

Vinkki: Sphero Edu -sovelluksessa voit saada lisätietoja lohkoista ja niiden toiminnoista “lohko-ohje”-toiminnon avulla. Tietokoneella ohje löytyy painamalla lohkosta hiiren oikealla näppäimellä ja sovelluksella pitkään painamalla. Sovelluksen lohko-ohjelmointi perustuu JavaScript-kieleen. Kun teette projektin ohjelmointitehtäviä, voitte kurkata, miltä JavaScript kieli näyttää, painamalla sovelluksessa kolmen pisteen kuvakkeesta ja valitsemalla JavaScript Code.

Kysymyksissä saa käyttää apuna internettiä, puhelimeen ladattavaa Sphero Edu -sovellusta ja tarvittaessa voitte pyytää apua myös opettajalta.

Ryhmä 1: Silmukat

* Mikä on silmukan pääidea?
* Miten silmukka toimii?
* Mistä välilehdestä ja millä värillä silmukat löytyvät lohko-ohjelmoinnissa?
* Milloin kannattaa käyttää ”x kertaa”/ ”silmukka kunnes”/ ”jos \_\_ niin”/ ”jos \_\_ niin, muuten” -silmukkaa? Selitä esimerkiksi silmukan idean kautta.
* Mitä hyötyä silmukoista on koodia kirjoittaessa?
* Mitä voit sijoittaa Sphero Edu ohjelmassa ”Tosi/epätosi” kohdan tilalle?

Ryhmä 2: Muuttujat

* Mikä on muuttuja ohjelmoinnissa ja matematiikassa?
* Miten muuttuja kannattaa nimetä?
* Mitkä ovat Sphero Edu -ohjelman mahdolliset muuttujan tyypit, ja mitä näillä muuttujan tyypeillä tarkoitetaan?
* Miten muuttuja voidaan sijoittaa koodiin Sphero Edu -ohjelmassa?
* Miten muuttujan arvoa voidaan muuttaa myöhemmin?

Ryhmä 3: Funktiot

* Mikä on funktio ohjelmoinnissa ja matematiikassa?
* Miten funktio kannattaa nimetä?
* Miten funktio helpottaa ohjelmointia? (Vinkki: saman asian tekeminen monta kertaa)
* Mitä ovat parametrit?
* Onko tiettyä määrää, jota parametreja pitäisi asettaa?
* Mitä funktioon tulisi lisätä viimeisenä?

Ryhmä 4: Laskutoiminnot

* Millä merkeillä saat suoritettua nämä laskutoimitukset?
	+ Pluslasku
	+ Miinuslasku
	+ Kertolasku
	+ Jakolasku
	+ Potenssi
	+ Prosentti
* Mitä tarvitset, jotta voit suorittaa laskutoimintoja Sphero Edu –lohko-ohjelmoinnissa?
* Mihin voit sijoittaa operaattoreita Sphero Edu -lohko-ohjelmoinnissa?
* Mitä nämä merkinnät tarkoittavat ohjelmoinnissa?
	+ ===
	+ !==
	+ <
	+ <=
	+ >
	+ >=

**Labyrintti (Fablab):**

Jakaudutaan uusiin ryhmiin ja siirrytään labyrinttiin porrastetusti. Labyrintissa ratkotaan seuraavia ohjelmointiongelmia. Labyrintissä on yksi tehtävä per etappi (este). Kunkin ratkaistun tehtävän jälkeen olette yhden etapin lähempänä maalia ja voitte nostaa pallon esteen toiselle puolelle. Tehkää jokaiseen tehtävään oma ohjelmapohja, jonka säästätte lopullista palautekeskustelua ja koontia varten.

**Tehtävä 1:**

Asettakaa pallon ledille haluamanne väri (älkää kuitenkaan valitko punaista). Ohjelmoikaa pallo ensimmäisen esteen (etapin) luo niin, että se törmää siihen. Törmätessään pallon pitää vilkuttaa punaista led-valoa 3 kertaa 2 sekunnin ajan.

**Tehtävä 2:**

Ohjelmoikaa pallo seuraavan esteen luo. Matkan varrella haluamassanne kohtaa pallon pitää kuitenkin tehdä täysi ympyrä. Jotta ympyrä erottuu pallon muusta etenemisestä, ohjelmoikaa pallo vaihtamaan väriään satunnaisesti ympyrää kierrettäessä.

**Tehtävä 3:**

Edellisissä tehtävissä saitte ohjelmoida pallon esteiden luo “vieri”-komennolla. Nyt eteneminen pitää toteuttaa kuitenkin erillisen funktion avulla. Lisätkää funktioon myös toiminto, jolla pallosta saadaan kuulumaan haluamanne ääni sen saapuessa uudelle esteelle.

**Mäen ylitys:**

Mäen ylityksen voitte jälleen toteuttaa “vieri”-komennoilla. Sphero pallo sisältää useita sensoreita, joiden avulla se mittaa mm. sijaintia ja kaltevuutta. Jokaisen ohjelman suorituksen jälkeen näiden sensorien keräämää dataa voi tarkastella painamalla oikeasta yläkulmasta löytyvää ”Sensor Data”-painiketta. Tässä tehtävässä teidän pitää hyödyntää pallon sensorien mittaamaa y-akselin muutosta eli koordinaatistosta ajatellen pallon reitin varrella tapahtuvia korkeuden muutoksia.

Toteuttakaa mäen ylitystä varten ohjelma, jossa pallo ensin vierii kohti mäkeä. Mikäli pallo pääsee mäen huipulle eli sen y-akselin sijainti kasvaa, pallo jatkaa mäkeä alas. Jos taas pallo ei saavuta mäen huippua, se pysähtyy, vilkuttaa punaista led-valoa kolmesti 2 sekunnin ajan ja antaa äänimerkin. Muuttakaa pallon ”vieri”-komennon nopeuksia, kunnes saavutatte mäen huipun.

Vinkki: Jos/muuten-silmukka

* PALLO PÄÄSI MAALIIN 😊 HUIPPUA TYÖTÄ!

Halutessanne voitte jatkaa nyt haastavampiin tehtäviin. Ylittäkää mäki uudelleen, mutta vastakkaiseen suuntaan. Huomioikaa pallon kalibrointi ennen aloitusta.

**Tehtävä 4:**

Saavuttaessa neljännelle esteelle teidän tulee muodostaa yhtälö, joka on tosi, jotta pallo pääsee jatkamaan matkaa. Mikäli yhtälö on muodostettu oikein, pallo suorittaa ”disko”-funktion, jossa se pyörii viidesti ympäri vaihtaen samalla valoaan. Mikäli muuttujille on annettu virheelliset arvot, pallo vilkuttaa punaista valoa ja hälyttää.

Vinkki: Aloita luomalla kaksi muuttujaa (numero), joille ohjelman alussa asetat tietyt arvot (ja joita voit vaihtaa). Tulon tarkistaminen kannattaa toteuttaa jos/muuten-silmukalla, johon ”disko”-funktio voidaan sijoittaa. Saat pallon pyörimään paikallaan kierto -komennolla. Mieti tarkkaan, miten saat ”disko”-funktion luotua mahdollisimman yksinkertaisesti.

**Tehtävä 5:**

Seuraavaksi pallo pitää ohjelmoida seuraavan esteen luo, jonka edessä pallo tarkastelee paikallaan ollessaan jälleen kokonaislukuja. Ideana on miettiä jokin kokonaisluvuista muodostuva väli, jonka kaikki alkiot eli luvut käydään läpi. Jotta varmistetaan kaikkien lukujen läpikäynti, on pallon vaihdettava väriään parillisten lukujen kohdalla. Lukulistan läpikäyminen toteutetaan funktiolla.

Vinkki: Valitse pieni väli, esim. 1-10, sillä muuten ohjelman suoritus kestää kauan. Funktiossa kannattaa hyödyntää seuraavia jo aiemmissa tehtävissä esiin tulleita aiheita luomalla esimerkiksi muuttuja, jonka alkuarvoksi asetetaan 0 ja jonka arvoa kasvatetaan jokaisessa silmukan suorituksessa yhdellä. Parillisten lukujen tarkastelu kannattaa toteuttaa jos/muuten -rakenteella suoraan funktion sisään.

**Tehtävä 6:**

Viimeisenä kohtana pallo pitää ohjelmoida kulkemaan mutkaisen radan osion läpi. Kohdatessaan viimeisen esteen saatte toteuttaa itse suunnittelemanne funktion. Funktiossa pallon pitää jollakin tavalla hyödyntää sen omien sensoreidensa muodostamaa dataa. Tutkikaa eri vaihtoehtoja ja tehkää mieleisenne. Ohjelmoikaa pallo lopuksi ulos labyrintistä.

Vinkki: Funktion toteutuksessa voi hyödyntää esim. pallon sensorien mittaamaa edetyn matkan pituutta. Mikäli pallo törmää seinään eli pallon pujottelu esteissä epäonnistuu, pallon laskema matka nollataan.

* PALLO PÄÄSI TAKAISIN LÄHTÖPISTEESEEN 😊 ONNEKSI OLKOON, OLETTE SUPERHYVÄ TIIMI JA SELVISITTE LABYRINTISTA KAHDESTI!

**Arviointi:**

Kirjoittakaa lyhyt itsearvio omasta työskentelystä sekä vertaisarvio ryhmätyöskentelystä ja projektin onnistumisesta.

Vastatkaa esimerkiksi näihin kysymyksiin:

Itsearviointi (tehdään yksin):

* Opitko jotain uutta ohjelmoinnista?
* Mitä tykkäsit projektista?
* Osallistuitko tasapuolisesti projektin tekoon?
* Kuunneltiinko sinua?
* Mikä oli sinulle helppoa? Entä haastavaa?

Vertaisarviointi (ryhmässä):

* Kuinka toimitte ryhmänä?
* Osallistuivatko kaikki ryhmän jäsenet tasapuolisesti?
* Miten keskustelitte ja delegoitte tehtävänjakoa?
* Kuinka ohjelmointi sujui?
* Onnistuitteko ohjelmoimaan lyhyitä ohjelmia?
* Oliko jokin tehtävä liian vaikea?