



PAIKOITUS

Robotin oikea sijainti ja usein myös oikea asento luovat perustan sille osoitettujen tehtävien onnistuneelle suorittamiselle.

Robotin asennon oikaisu pelialustan merkintöjä, seinää tai muuta kiinteää rakennetta hyödyntäen on kokeneille robottikilpailuihin osallistuville perustoiminto.



EV3-G



EV3 Classroom



Spike Scratch



Spike Python

TEHTÄVÄ

PAIKOITUS

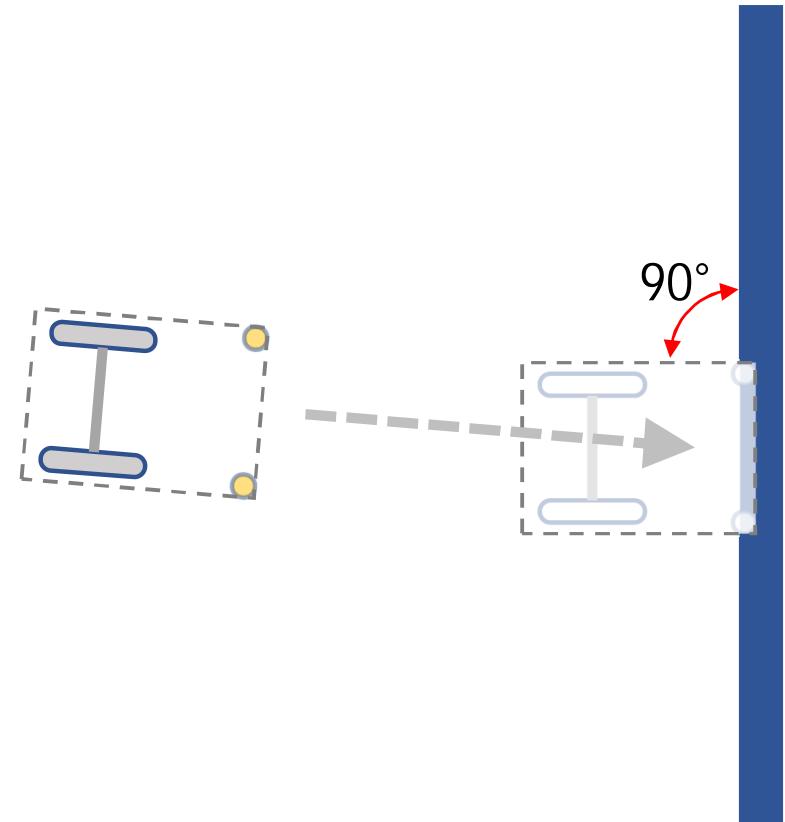
Harjoitellaan robotin asennon oikaisua alustan merkintää hyödyntäen.

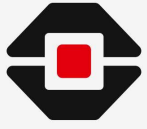
Pitkiä matkoja kuljetaan usein reippaammalla nopeudella mutta kohteen luokse tultaessa on hyvä hidastaa kulkua ajoissa.

Kun vasen värianturi tunnistaa merkinnän, pysäytetään vasen moottori. Vastaavasti oikea moottori pysäytetään kun oikea värianturi tunnistaa merkinnän.

Peruutetaan 5 cm ja toistetaan edellinen hyvin pienellä nopeudella.

Arvioi lopuksi miten hyvin robotti onnistui oikaisemaan asentonsa.

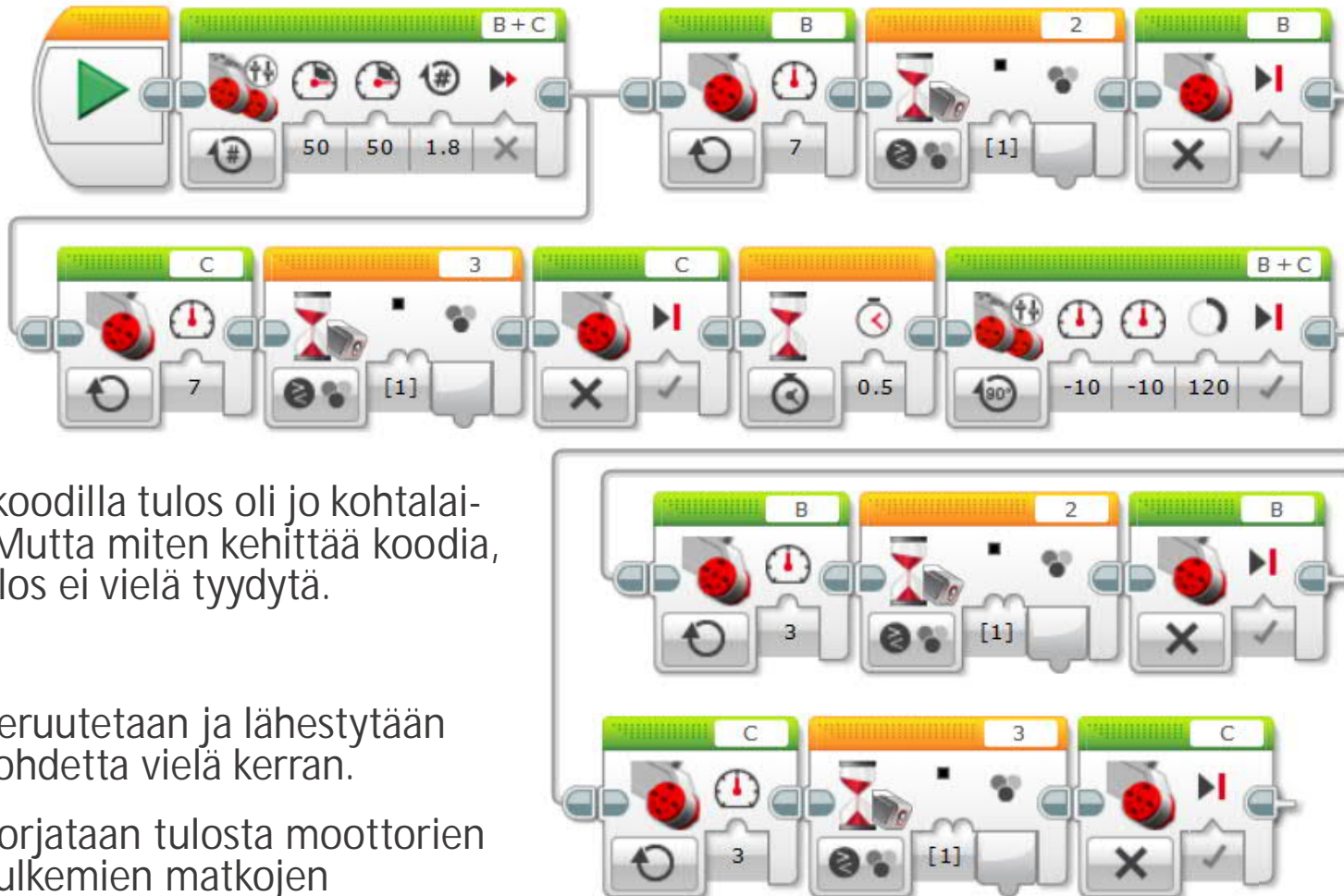




LEGO education

PAIKOITUS

EV3-G



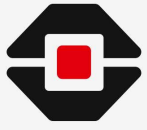
Tällä koodilla tulos oli jo kohtalainen. Mutta miten kehittää koodia, jos tulos ei vielä tyydytä.

Esim:

1. Peruutetaan ja lähestytään kohdetta vielä kerran.
2. Korjataan tulosta moottorien kulkemien matkojen erotuksella.

Kokeillaan jälkimmäistä

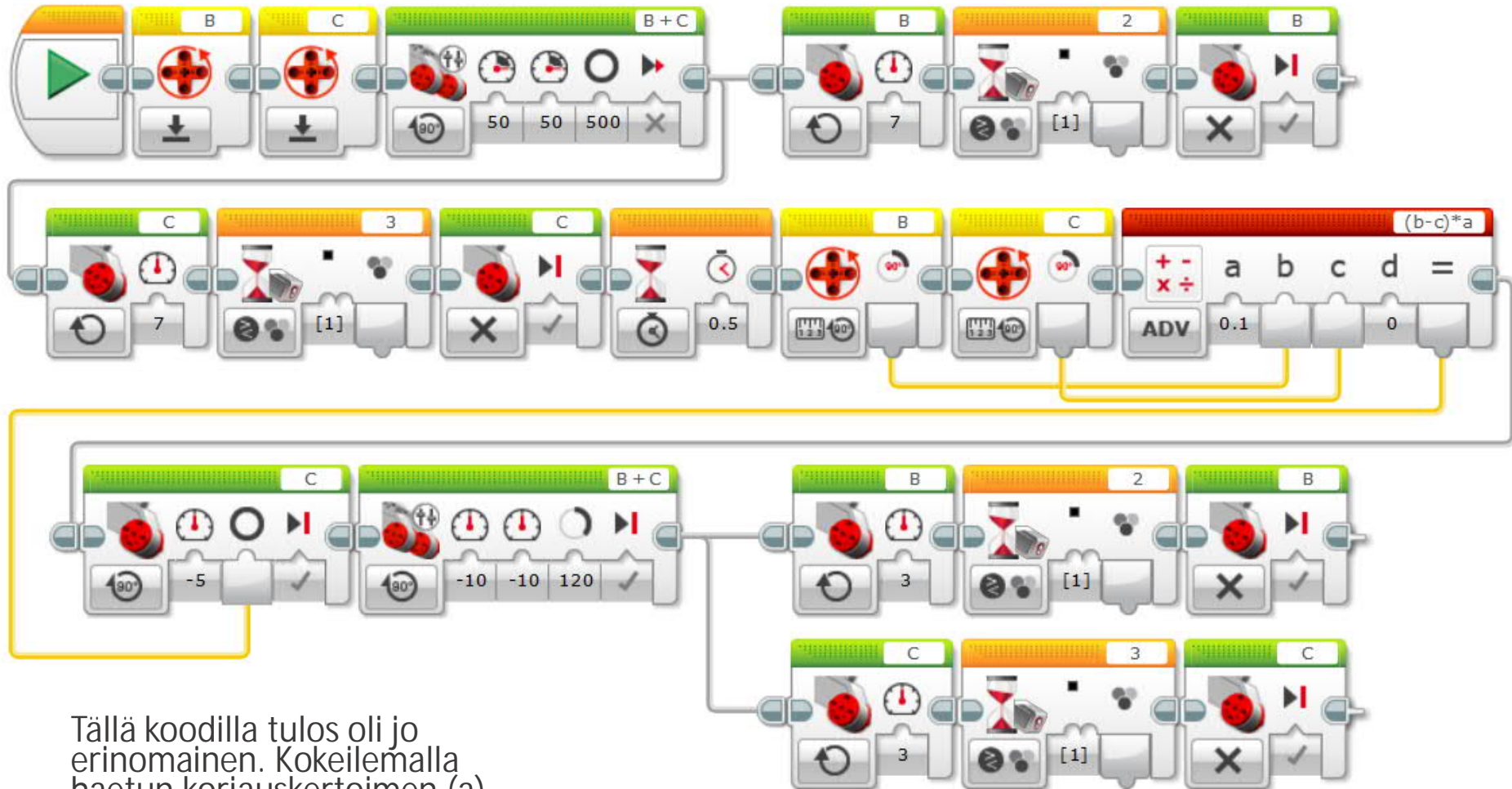




LEGO education

PAIKOITUS

EV3-G



Tällä koodilla tulos oli jo erinomainen. Kokeilemalla haetun korjauskertoimen (a) arvon pienuus yllätti.

My Block -ainesta





LEGO education

EV3 Classroom

PAIKOITUS

```
when program starts
  set movement motors to B and C
  set movement speed to 50 %
  move forward for 500 degrees
  broadcast message1
  B start motor at 10 % speed
  wait until 2 is color black ?
  B stop motor
  wait .5 seconds
  set movement speed to 10 %
  move backward for 120 degrees
  broadcast message2
  B start motor at 3 % speed
  wait until 2 is color black ?
  B stop motor
  wait 1 seconds
  stop and exit program
```

Tällä koodilla tulos oli jo kohtalainen. Mutta miten kehittää koodia, jos tulos ei vielä tyydytä.

Esim:

1. Peruutetaan ja lähestytään kohdetta vielä kerran.

```
when I receive message1
  C start motor at 10 % speed
  wait until 3 is color black ?
  C stop motor
```

2. Korjataan tulosta moottorien kulkemien matkojen erotuksella.

Kokeillaan jälkimmäistä

```
when I receive message2
  C start motor at 3 % speed
  wait until 3 is color black ?
  C stop motor
```





LEGO education

EV3 Classroom

PAIKOITUS

```
when program starts
  B reset degrees counted
  C reset degrees counted
  set korjaus to .1
  set movement motors to B and C
  set movement speed to 50 %
  move forward for 500 degrees
  broadcast message1
  B start motor at 10 % speed
  wait until 2 is color black ?
  B stop motor
  wait .5 seconds
```

Tällä koodilla tulos oli jo erinomainen. Kokeilemalla haetun korjauskertoimen (a) arvon pienuus yllätti.

My Block –ainesta

(HUOM. Ohjelma jatkuu seuraavalla sivulla

```
when I receive message1
  C start motor at 10 % speed
  wait until 3 is color black ?
  C stop motor
```





LEGO education

PAIKOITUS

EV3 Classroom

```
run for [C] degrees counted - [B] degrees counted * korjaus degrees at -5 % speed
set movement speed to 10 %
move backward for 120 degrees
broadcast message2
[B] start motor at 3 % speed
wait until 2 is color black?
[B] stop motor
wait 1 seconds
stop and exit program

when I receive message2
[C] start motor at 3 % speed
wait until 3 is color black?
[C] stop motor
```





PAIKOITUS

Spike Scratch

Tällä koodilla tulos oli jo hyväksyttävä.

My Block –ainesta

(HUOM. Ohjelma jatkuu seuraavalla sivulla





LEGO education

PAIKOITUS

Spike Scratch

```
asetta liikenopeus 10 %iin  
liiku ↓ 120 astetta  
lähetä viesti2  
C aseta nopeus 8 %iin  
C käynnistä moottori ↺  
odota kunnes [camera icon] A onko väri [black] ?  
C pysäytä moottori  
odota 1 sekuntia  
pysäytä kaikki
```

```
kun vastaanotan viesti2  
D aseta nopeus 8 %iin  
D käynnistä moottori ↻  
odota kunnes [camera icon] B onko väri [black] ?  
D pysäytä moottori
```





LEGO education

PAIKOITUS

Spike Python



Tällä koodilla tulos oli jo hyväksyttävä.

My Block –ainesta

```
1 from spike import PrimeHub, ColorSensor, MotorPair, Motor
2 from spike.control import wait_for_seconds
3 hub = PrimeHub() # Tehdään laitteiden alustukset
4 moottorit = MotorPair('C', 'D')
5 vasen = ColorSensor('A')
6 oikea = ColorSensor('B')
7 vasen_m = Motor('C')
8 oikea_m = Motor('D')
9 # Ohjelma
10 moottorit.move(500, 'degrees', 0, 50) # Lähestytään ensin nopeasti
11
12 vasen_m.start(-10) # Siirrytään hitaaseen lähestymiseen
13 oikea_m.start(10) # Huomaa pyörimissuunnat
14 mustalla = 0 # Alustetaan tarvittavat muuttujat.
15 mustalla_v = 0 # Tehdään hiukan piilavetinen ratkaisu rinnakkaisen
16 mustalla_o = 0 # suorittamisen sijaan.
17 # Ensimmäinen hitas lähestymisen
18 while mustalla < 2: # Silmukkaa suoritetaan kunnes molemmat anturit mustalla
19     vasen_väri = vasen.get_color() # Luetaan anturit
20     oikea_väri = oikea.get_color()
21     if vasen_väri == 'black':
22         vasen_m.stop()
23         mustalla_v = 1
24 # Jatkuu seuraavalla sivulla
```





LEGO education

PAIKOITUS

Spike Python



```
25     elif oikea_väri == 'black':
26         oikea_m.stop()
27         mustalla_o = 1
28         mustalla = mustalla_v + mustalla_o
29
30     wait_for_seconds(0.5) # Rauhoitetaan liike paikalleen
31     moottorit.move(-120, 'degrees', 0, 10) # Peruutetaan
32
33     vasen_m.start(-8) # Alustetaan toinen hiidas lähestymisen
34     oikea_m.start(8)
35     mustalla = 0
36     mustalla_v = 0
37     mustalla_o = 0
38     # Toinen hiidas lähestymisen
39     while mustalla < 2:
40         vasen_väri = vasen.get_color()
41         oikea_väri = oikea.get_color()
42         if vasen_väri == 'black':
43             vasen_m.stop()
44             mustalla_v = 1
45         elif oikea_väri == 'black':
46             oikea_m.stop()
47             mustalla_o = 1
48         mustalla = mustalla_v + mustalla_o
49     end
```





LEGO education

PAIKOITUS

Spike Python



Saman koodin voi kirjoittaa myös lyhyemmin

```
1 from spike import PrimeHub, ColorSensor, MotorPair, Motor
2 from spike.control import wait_for_seconds
3 hub = PrimeHub()
4 moottorit = MotorPair('C', 'D')
5 vasen = ColorSensor('A')
6 oikea = ColorSensor('B')
7 vasen_m = Motor('C')
8 oikea_m = Motor('D')
9 moottorit.move(500, 'degrees', 0, 50)
10 def hi das_lähestymi nen(): # Tehdään oma toiminto (MyBlock) nimeltä hi das_lähestymi nen
11     vasen_m.start(-10)
12     oikea_m.start(10)
13     while vasen.get_color() != 'black' or oikea.get_color() != 'black':
14         if vasen.get_color() == 'black':
15             vasen_m.stop()
16         elif oikea.get_color() == 'black':
17             oikea_m.stop()
18 hi das_lähestymi nen() # Suoritetaan hi das_lähestymi nen -toiminto
19 wait_for_seconds(0.5)
20 moottorit.move(-120, 'degrees', 0, 10)
21 hi das_lähestymi nen() # Suoritetaan hi das_lähestymi nen -toiminto toistamiseen
22 end
23
```

