

ICILS 2023: Ohjelmoinnillisen ajattelun esimerkkitehtäviä

Seuraavat tehtävät ovat IEA:n omaisuutta. Niiden käyttö julkaisuissa tai tutkimuksessa vaatii luvan IEA:lta. Lisätietoja osoitteessa <https://www.iea.nl/node/4111>. Englanninkielisiä tehtäviä voi kokeilla osoitteessa <https://www.iea.nl/icils2023-assessment>.

Tehtävät edustavat erilaisia tehtäviä, joita oppilaat saivat tehtäväkseen ICILS 2023 -tutkimuksen ohjelmoinnillisen ajattelun kokeessa. Esimerkkitehtävät ovat kahdesta eri tehtäväkokonaisuudesta: Automaattinen bussi ja Maatilan lennokki. Tehtäviä havainnollistetaan kuvakaappauksin, jotka kuvaavat, miltä tehtävä näytti sen aloittamishetkellä. Muutamien tehtävien yhteydessä on esitetty kriteerejä tietyn pistemäärän saavuttamiseksi sekä pistemäärien prosenttiosuudet.



Ohjelmoinnillisen ajattelun koeosio: Automaattinen bussi

Automaattinen bussi -osiossa arvioitiin pääasiassa ongelmien käsitteellistämisen osaamista ohjelmoinnillisessa ajattelussa. Tehtävät liittyivät tietokoneohjelmien suunnitteluun ja navigointi- ja jarrujärjestelmien säätämiseen kuljettajattoman bussin käyttämiseksi. Tehtävissä hyödynnettiin vuorovaikuttavia suunnattuja graafeja, kaavioita ja simulaatiota. Simulaatiotehtävään sisältyi tiedon (datan) keräämistä havaintoperustaisen analyysin ja päätelmien tekemisen tukena.

JOHDANTO

On rakennettu uusi automaattinen bussi.
Bussilla kuljetetaan oppilaita urheilutapahtumaan.
Sinä autat kehittämään tietokoneohjelman, joka ohjaa bussia.

1

2

3

4

5

5 min

7

8

Napsauta 🔍 luettuasi johdannon.

🔍

➡️

Tehtävä 1: Bussin ohjaaminen käskysarjalla

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu: Algoritmien, ohjelmien ja rajapintojen kehittäminen

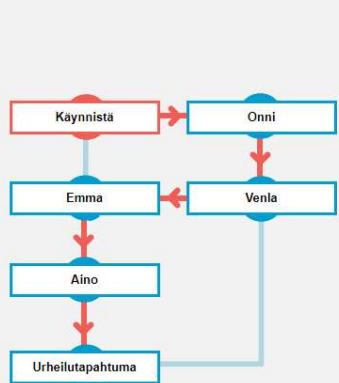
Ensimmäisessä tehtävässä oppilaan tuli ohjata bussi määrättyä reittiä pitkin luomalla käskysarja ohje- ja sijainti/suuntakomentoja hyödyntäen. Määrätty bussireitti esitettiin suunnattuna graafina, jonka sisältämät solmu-/risteyskohdat esittivät pysähdyspaikkoja ja nuolet kohtien välisiä kulkureittejä. Oppilaan tuli täydentää käskysarja lisäämällä bussin ohjausasetuksiin uusia käskyjä (rivejä) ja valitsemalla uusille riveille ilmestyneistä pudotusvalikoista sopiva käsky (*käänny suuntaan* tai *liiku kohtaan*) ja sijainti/suunta (pääilmansuuntien tai reitillä merkittyjen paikkojen nimet). Kaksi ensimmäistä käskyä oli asetettu valmiiksi. Tehtävässä arvioitiin käskyjen loogista järjestämistä ja niiden toteuttamista käyttöliittymää hyödyntäen.

Bussin ohjausasetukset

Ohjeet	Sijainti/Suunta
Käänny suuntaan ▼	Itä ▼
Liiku kohtaan ▼	Onni ▼
▼	▼
+Lisää	

Nollaa

Bussireitti



Pohjoinen
Länsi
Itä
Etelä

Käynnistä → Onni
Onni → Venla
Venla → Emma
Emma → Aino
Aino → Urheilutapahtuma
Urheilutapahtuma → Venla

1

2

3

4

5

5 min

7

8

Bussin tulee seurata punaisilla nuolilla merkittyä reittiä.
Käytä bussin ohjausasetusten pudotusvalikoita, jotta saat bussin seuraamaan reittiä.
Ensimmäiset kaksi on tehty puolestasi.
Napsauta ⌂, kun olet valmis jatkamaan.

🔍

➡

Tehtävä 2: Kulkureitin määrittäminen vuorovaikuteisessa solmukaaviossa

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Digitaalisten järjestelmien tuntemus ja ymmärrys

Toisessa tehtävässä oppilaan tuli käyttää vuorovaikuteista solmukaaviota määrittääkseen suorimman kulkureitin bussille, jonka tehtävänä oli ottaa kaikki ihmiset kyytiin ja kuljettaa heidät urheilutapahtumaan. Lähtöpaikka, ihmiset ja urheilutapahtuma esitettiin suuntaamattomassa graafissa, joka oli tehtävää 1 moniosaisempi ja sisälsi useampia reittivaihtoehtoja. Kaaviota käytettiin napsauttamalla solmu-/risteyskohtia hiirellä, jolloin peräkkäin valittujen solmujen välille piirtyi reitti (kutsuttu tehtävässä nimellä ”siirto”). Kaikki mahdolliset siirrot vierekkäisiin solmuihin esitettiin vihreällä reunuksella. Tehtävän yläreunassa näytettiin, montako siirtoa oppilas oli tehnyt. Siirtoja oli mahdollista tehdä käytännössä loputtomasti, sillä samaa reittiä pitkin oli mahdollista kulkea monta kertaa. Tehtävä liittyi pääasiassa digitaalisten järjestelmien tuntemukseen ja ymmärrykseen, sillä se edellytti kaavion ominaisuuksien, sallittavien siirtojen ja järjestelmän osioiden keskinäisvuorovaikutuksen logiikan ymmärrystä.

Tehtävästä saivat yhden pisteen ne oppilaat, jotka määrittivät muun kuin optimaalisen reitin (ts. reitillä oli ylimääräisiä tai tarpeettomia liikkeitä), jolla bussi pysähtyi jokaisen ihmisen kohdalla ja päättyi urheilutapahtumaan. Kaksi pistettä saivat ne oppilaat, jotka loivat optimaalisen (ts. jokaisen ihmisen kohdalla vain kerran pysähtyvän) reitin, joka päättyi urheilutapahtumaan.

Kansainvälisesti keskimäärin 76 prosenttia oppilaista sai tehtävästä vähintään yhden pisteen. Kaksi pistettä saaneita oli kansainvälisesti keskimäärin 57 prosenttia oppilaista.

Siirtoja: 0

Kumoa

Nollaa

1

2

3

4

5

5 min

7

8

Q

➔

Bussin on kuljetettava kaikki ihmiset urheilutapahtumaan.
Napsauta nimiä ja luo suorin reitti, johon sisältyvät kaikki ihmiset.
Aloita "Lähdöstä" ja lopeta "Urheilutapahtumaan".
Napsauta ☺, kun olet valmis jatkamaan.

Suoritustaso: 1–2

Maa/alue	Vähintään 1/2 pistettä saavuttaneiden oppilaiden %-osuus	2/2 pistettä saavuttaneiden oppilaiden %-osuus
* Tšekki	89 (0.63)	70 (0.92)
Suomi	84 (1.43)	67 (1.67)
* Ruotsi	83 (1.19)	66 (1.62)
* Tanska	83 (1.14)	66 (1.38)
* Belgia (fl)	81 (1.84)	63 (2.07)
Slovakia	82 (1.15)	63 (1.56)
Italia	80 (1.06)	62 (1.43)
Saksa	79 (1.38)	62 (1.41)
Taiwan	76 (1.24)	62 (1.32)
* Itävalta	80 (1.16)	60 (1.44)
* Latvia	77 (1.60)	58 (1.82)
* Norja	74 (1.25)	57 (1.68)
Ranska	75 (1.14)	57 (1.40)
Kv. Keskiarvo	76 (0.29)	57 (0.34)
* Portugali	78 (1.16)	56 (1.36)
Luxemburg	76 (0.86)	56 (1.18)
* Slovenia	76 (0.98)	54 (1.25)
* Kroatia	67 (1.79)	43 (1.91)
* Serbia	64 (1.77)	43 (1.76)
* Etelä-Korea	53 (1.42)	42 (1.50)
* Uruguay	64 (1.77)	41 (1.70)
Malta	66 (1.47)	40 (1.42)

* Katso oppilasotoksiin liittyvät huomautukset kansallisen raportin kuviosta 3.1.

() keskivirhe

Tehtävä 3: Kulkureitin määrittäminen aikatietoja sisältävässä solmukaaviossa

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Oleellisen tiedon kerääminen ja esittäminen

Kolmannessa tehtävässä oppilas käytti tehtävän 2 kaltaista kaaviota, joka sisälsi numeerista aikatietoa (ts. matkustusaika minuutteina) solmujen välillä. Oppilaan tuli etsiä nopein reitti urheilutapahtumasta koululle. Kaaviota käytettiin samaan tapaan kuin edellisessä tehtävässä sillä poikkeuksella, että kutakin reittiä pitkin oli tällä kertaa mahdollista kulkea vain kerran. Tehtyjen siirtojen sijaan oppilaalle näytettiin Tulokset-taulukossa hänen valitsemaansa reittiin kuuluva yhteenlaskettu kesto minuutteina. Oppilaan oli mahdollista tallentaa taulukkoon kulloinkin yhteensä viisi reittiä/yritystä (yritykset oli mahdollista nollata) ja tarkastella niitä uudelleen valitsemalla ne taulukosta. Tehtävässä arvioitiin oppilaan kykyä suunnitella ja arvioida ratkaisuja kokeilemalla erilaisten reittien laatimista systemaattisesti ja reittien välillä olevien erojen tarkastelemista parhaan vaihtoehdon löytämiseksi.

Selitys

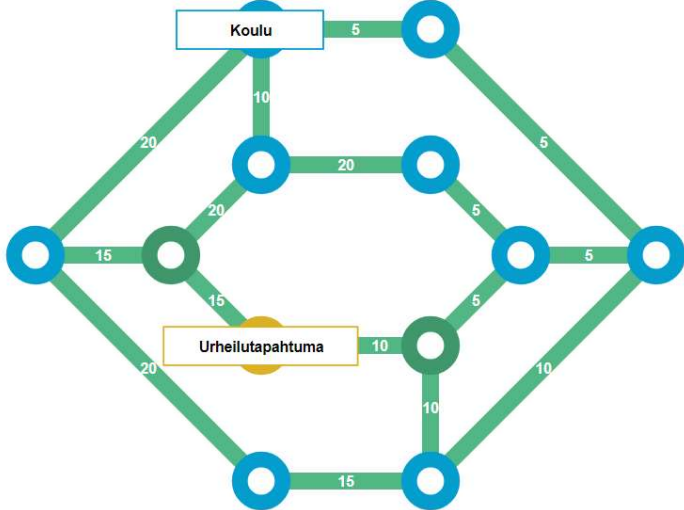

Risteyks

Tulokset

Yritys	Kokonaisaika
1	
2	
3	
4	
5	

Valittu yritys: 1

Nollaa yritys



1

2

3

4

5

5 min

7

8

Etsi nopein reitti "Urheilutapahtumasta" "Koululle".
Luo reitti napsauttamalla risteyskohtia.
Reittikartta näyttää, kuinka kauan kestää kulkea kunkin risteyskohdan välillä. Tuloksesi tallentuvat taulukkoon.
Mikä taulukon yrityksistä näyttää nopeimman reitin?

Yritys





Tehtävä 4: Oikean diagrammin valitseminen

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Oleellisen tiedon kerääminen ja esittäminen

Neljännessä tehtävässä oppilaalle näytettiin tiekartta, johon sisältyi sijainteja välillä A–E. Tehtävän vasemmassa laidassa esitettiin diagrammeja, joissa oli erilaisia pelkistettyjä reittejä näiden kirjainten välillä. Oppilaan piti tunnistaa, mikä diagrammeista esitti kaikki mahdolliset kartalla esitetyt reitit. Tehtävässä painottui oppilaan kyky pelkistää merkityksetön tieto (esim. katujen mutkittelu, talot) ja ymmärtää reitin rakenne kaaviona, jossa solmukohdat esittävät risteyksiä tai pysähdyspaikkoja ja viivat niiden välisiä kulkuväyliä.

Diagrammi 1

Diagrammi 2

Diagrammi 3

Diagrammi 4

Bussin pitää kulkea pisteestä A pisteeseen E.
Bussi voi liikkua vain eteenpäin.
Mikä diagrammi esittää kaikki mahdolliset reitit A:sta E:hen?

- ☐ Diagrammi 1
- ☐ Diagrammi 2
- ☐ Diagrammi 3
- ☐ Diagrammi 4

1
2
3
4
5
5 min
7
8

Q

→

Tehtävä 5: Puukaavion täydentäminen

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Ongelmien määrittely ja analysointi

Viidennessä tehtävässä oppilaalle esitettiin tilanne, jossa kuljettamatonta bussia ohjaavan tietokoneen tulisi päättää, pysäytetäänkö bussi seuraavalla pysäkillä. Tehtävässä oli määritetty kaksi pysähtymisehtoa: se, että matkustaja haluaa poistua bussista tai se, että uusi matkustaja haluaa tulla bussiin sisään. Kaavioon oli määritetty valmiiksi kohdat "Aloita seuraavan pysäkin tarkistus", "Haluaako joku poistua bussista?", "Haluaako joku tulla bussiin sisään?" ja "Lopeta seuraavan pysäkin tarkistus". Oppilaan tuli täydentää puukaavio vetämällä puuttuvat nimikkeet ("Pysähdy seuraavalla pysäkillä", "Kyllä", "Kyllä", "Ei", "Ei") kaavion oikeisiin kohtiin, jotta kuvattu prosessi eteni loogisesti oikein.

Nimikkeet

Pysähdy seuraavalla pysäkillä

Kyllä

Kyllä

Ei

Ei

Puukaavio

```
graph TD; A[Aloita seuraavan pysäkin tarkistus] --> B{Haluaako joku poistua bussista?}; B -- Kyllä --> C{Haluaako joku tulla bussiin sisään?}; B -- Ei --> D[Lopeta seuraavan pysäkin tarkistus]; C -- Kyllä --> D; C -- Ei --> B;
```

1

2

3

4

5

5 min

7

8

Bussin tietokoneen pitää päättää, pysäytetäänkö bussi seuraavalla pysäkillä.

Bussin pitää pysähtyä, jos matkustaja haluaa poistua kyydistä seuraavalla pysäkillä.

Bussin pitää pysähtyä, jos joku odottaa seuraavalla pysäkillä.

Vedä ja pudota nimikkeet puukaavioon osoittaaksesi, miten matkustajatarkistuksen pitäisi toimia.

Napsauta , kun olet valmis jatkamaan.





Tehtävä 6: Simulaattoritehtävä

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Oleellisen tiedon kerääminen ja esittäminen

Kuudennessa tehtävässä oppilas käytti kohteentunnistussimulaattoria määrittääkseen suurimman bussin ja pyöräilijän välisen etäisyyden, jolta bussi voi tunnistaa pyöräilijän oikein annetuissa olosuhteissa (yöaikaan ja sateella).

LAAJAN TEHTÄVÄN TIEDOT

Automaattinen bussi käyttää kameroita tallentaakseen kuvia ympäristöstään.
Sen jälkeen se käyttää tietokoneohjelmaa tunnistaa kuvissa olevia kohteita.
On tärkeää, että ohjelma tunnistaa kuvia oikein.
Vuorokaudenaika ja säätila voivat vaikuttaa siihen, kuinka hyvin ohjelma tunnistaa kohteita.

Tässä tehtävässä käytät simulaattoria. Ensinnäkin on käytettävä puukaaviota, jotta voit asettaa simulaattorille vuorokaudenajan ja säätilan.
Sen jälkeen voit käyttää simulaattoria selvittääksesi, mikä on suurin mahdollinen bussin ja kohteen välinen etäisyys, jolta ohjelma tunnistaa kohteen vielä oikein.

1

2

3

4

5

5 min

7

8

Napsauta , kun olet valmis jatkamaan.

Simulaattoriin kuului puukaavio, jossa oli mahdollista säätää (kyllä/ei) seuraavia kohtia: "Onko kohde havaittu?", "Onko yöaika?" ja "Sataako?". Näiden olosuhteiden asettamisen jälkeen oppilaan tuli säätää simulaattorissa bussille pysähtymisetäisyys (0–1000 metriä) ja suorittaa simulaatio nähdäkseen, tunnistiko bussi pyöräilijän määritetyillä parametreilla oikein. Oppilaan oli mahdollista kokeilla eri etäisyyksiä niin monta kertaa kuin halusi ja etsiä niistä suurin mahdollinen etäisyys, jolta bussi tunnisti pyöräilijän oikein. Tehtävään vastattiin valitsemalla oikea etäisyys vastausvaihtoehdoista.

Kohteentunnistussimulaattori

```

graph TD
    Start([Start]) --> Q1{Onko kohde havaittu?}
    Q1 -- Ei --> Start
    Q1 -- Kyllä --> Q2{Onko yöaika?}
    Q2 -- Ei --> Q1
    Q2 -- Kyllä --> Q3{Sataako?}
    Q3 -- Ei --> Q2
    Q3 -- Kyllä --> Q4{Sataako?}
    Q4 -- Ei --> Q2
    Q4 -- Kyllä --> Sim[Käynnistä simulaatio]
    
```

1

2

3

4

5

5 min

7

8

Aja: etäisyys kohteesta

1000 m

Nollaa bussin sijainti

Ei tulosta

Simulaattorissa näkyy pyöräilijä. On yöaika. Sataa.

Mikä on suurin bussin ja pyöräilijän välinen etäisyys, jolta bussi voi tunnistaa pyöräilijän oikein?

Käytä kohteentunnistussimulaattoria apunasi vastataksesi tehtävään.

Napsauta Q nähdäksesi tehtävän tiedot uudelleen.

☐ 0 m
 ☐ 100 m
 ☐ 200 m
 ☐ 300 m
 ☐ 400 m
 ☐ 500 m
 ☐ 600 m
 ☐ 700 m
 ☐ 800 m
 ☐ 900 m
 ☐ 1 000 m

Tehtävä 7: Simulaattorin käyttöstrategia

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Ongelmien määrittely ja analysointi

Seitsemännessä tehtävässä oppilaalle esitettiin edellisessä tehtävässä hänen antamansa vastaus (etäisyys metreinä). Oppilaan tuli kertoa, miten hän päätyi tähän vastaukseen edellisessä tehtävässä. Oppilas sai tehtävästä pisteitä, jos hän pystyi ymmärrettävästi kuvailemaan simulaattorin käyttöstrategiaansa, esimerkiksi systemaattinen kokeileminen tai satunnainen yrittäminen ja erehtyminen.

Kohteentunnistussimulaattori

```
graph TD; Q1{Onko kohde havaittu?} -- Ei --> Q1; Q1 -- Kyllä --> Q2{Onko yöaika?}; Q2 -- Ei --> Q1; Q2 -- Kyllä --> Q3{Sataako?}; Q3 -- Ei --> Q1; Q3 -- Kyllä --> Q4{Sataako?}; Q4 -- Ei --> Q1; Q4 -- Kyllä --> Start([Käynnistä simulaatio]);
```

Aja: etäisyys kohteesta
1000 m

Nollaa bussin sijainti

Ei tulosta

Vastauksesi mukaan suurin bussin ja pyöräilijän välinen etäisyys, jolta bussi voi tunnistaa pyöräilijän oikein, on
Kerro, miten päädyit tähän vastaukseen.

- 1
- 2
- 3
- 4
- 5
- 5 min
- 7
- 8

Q

→

Tehtävä 8: Tietokonesimulaatioiden hyödyt

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmien käsitteellistäminen: Digitaalisten järjestelmien tuntemus ja ymmärrys

Tehtävässä oppilaat vastasivat tietokonesimulaatioihin liittyvään kysymykseen. Oppilaan tuli kertoa, miksi tosielämässä käytettävien järjestelmien tietokonesimulaatiot ovat hyödyllisiä. Tehtävästä sai yhden pisteen, jos oppilas mainitsi johonkin seuraavista kategorioista kuuluvan hyödyn: turvallisuuden parantaminen, käytännöllisyys (esim. planeetan liikkeen simulointi), hallittu ympäristö (esim. olosuhteiden muuttaminen), resurssitehokkuus (esim. aika, materiaalit, raha). Tehtävästä sai kaksi pistettä, jos mainitsi hyötyjä kahdesta eri kategoriasta.

Kansainvälisesti keskimäärin 27 prosenttia oppilaista sai tehtävästä vähintään yhden pisteen. Kaksi pistettä saaneita oli kansainvälisesti keskimäärin 5 prosenttia oppilaista.

Kohteentunnistussimulaattori

```
graph TD; A{Onko kohde havaittu?} -- Ei --> A; A -- Kyllä --> B{Onko yöaika?}; B -- Ei --> C{Sataako?}; B -- Kyllä --> D{Sataako?}; C -- Ei --> E{Sataako?}; C -- Kyllä --> F(Käynnistä simulaatio); D -- Ei --> E; D -- Kyllä --> F; E -- Ei --> G(Ei tulosta); E -- Kyllä --> F;
```

Aja: etäisyys kohteesta
1000 m

Nollaa bussin sijainti

Ei tulosta

- 1
- 2
- 3
- 4
- 5
- 5 min
- 7
- 8

Miksi tosielämässä käytettävien järjestelmien tietokonesimulaatiot ovat hyödyllisiä?

Mainitse kaksi eri syytä.

Suoritustaso: 3–4

Maa/alue	Vähintään 1/2 pistettä saavuttaneiden oppilaiden %-osuus	2/2 pistettä saavuttaneiden oppilaiden %-osuus
* Etelä-Korea	72 (1.37)	36 (1.51)
* Itävalta	39 (1.74)	11 (0.84)
Taiwan	39 (1.55)	8 (0.67)
Saksa	22 (1.37)	5 (0.70)
* Kroatia	23 (1.54)	5 (0.66)
Kv. keskiarvo	27 (0.29)	5 (0.13)
* Tanska	26 (1.45)	4 (0.55)
* Latvia	26 (1.51)	4 (0.55)
* Uruguay	27 (1.34)	4 (0.53)
* Tšekki	26 (0.76)	4 (0.26)
* Belgia (fl)	23 (1.27)	3 (0.52)
Malta	17 (0.96)	3 (0.47)
* Norja	22 (1.19)	3 (0.45)
Slovakia	31 (1.50)	3 (0.45)
* Ruotsi	11 (0.83)	3 (0.44)
Suomi	31 (1.56)	3 (0.43)
Luxemburg	25 (1.10)	3 (0.42)
Italia	28 (1.35)	3 (0.40)
* Slovenia	19 (1.00)	3 (0.39)
Ranska	18 (0.97)	2 (0.37)
* Portugali	37 (1.66)	2 (0.32)
* Serbia	8 (0.81)	1 (0.21)

* Katso oppilasotoksiin liittyvät huomautukset kansallisen raportin kuviosta 3.1.

() keskivirhe

Ohjelmoinnillisen ajattelun koeosio: Maatilan lennokka

Maatilan lennokka -osio koostui ongelmanratkaisutehtävistä, joissa oppilaat ohjasivat lennokkia visuaalisessa ohjelmointiympäristössä. Oppilaiden tehtävänä oli luoda, muokata, testata ja korjata koodia. Koodilausekkeita ei tarvinnut itse kirjoittaa tekstipohjaisella ohjelmointikielellä, vaan ohjelmakoodi koostettiin valmiista kuvakkeista (lohkoista) vetämällä ja pudottamalla niitä visuaalisen ohjelmointiympäristön työtilaan. Työtilaan koostetun ohjelmakoodin ohjeet lähetettiin lennokille napsauttamalla vihreää suorituspainiketta. Suorittamalla ohjelmakoodin oppilas näki, miten lennokka ohjautui hänen koostamallaan koodilla. Oppilaiden oli mahdollista palata suorittamiinsa tehtäviin. Oppilaiden oli myös mahdollista liputtaa tehtävät, joihin he halusivat vielä myöhemmin palata parantamaan ratkaisujaan. Tehtävät vaikeutuivat sitä mukaa, kun oppilas eteni osion tehtävästä toiseen.

Oppilaiden suorittamista tehtävistä arvioitiin ohjelmakoodin tehokkuutta ja virheettömyyttä. Tehokkuuden osalta verrattiin oppilaan käyttämien koodilohkojen lukumäärää optimaalisimpaan lukumäärään, jolla oikeaan vastaukseen päästään. Virheettömyyden osalta tarkasteltiin sitä, suorittiko lennokka tehtävänannossa annetut toiminnot oikein. Jokaisesta yksittäisestä koodaustehtävästä annettiin yksi pistemäärä, joka johdettiin yhdistämällä virheettömyys- ja tehokkuuspisteet.

JOHDANTO

Tässä osiossa ohjelmoit lennokin suorittamaan toimintoja.

Käytät kuvakepohjaista koodieditoria lennokin ohjelmoimiseen.

Jokaisessa tehtävässä arvioidaan seuraavaa:

- Kuinka monta koodilohkoa käytät (vähemmän on parempi).
- Kuinka tarkasti suoritat tehtävän.

1

2

3

4

5

6

7

8

9

Napsauta  luettuasi johdannon.



Tehtävä 1: Ohjelmointitehtävä (käsky)

(Harjoittelutehtävä)

Osion ensimmäinen tehtävä oli harjoittelutehtävä, jossa oppilaan tuli ohjelmoida lennokki liikkumaan eteenpäin ainoastaan yhden ruudun verran.

The screenshot shows a programming interface with a green grid world on the left, a central workspace, and a right sidebar. The workspace has a status bar at the top that says "Työtila: 0" and "suoritteilla". Below the grid, there is a "Koodilohkot" (Code Blocks) area with two blocks: "liiku eteenpäin" (move forward) and "käänny oikeaan" (turn right). The right sidebar contains a list of numbers 1 through 9, each with a red flag icon. At the bottom, there are instructions in Finnish: "Käytä 'liiku eteenpäin' -koodia siirtääksesi lennokkia yhden ruudun verran eteenpäin." and "Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen. Napsauta ▶ nähdäksesi lopputuloksen. Napsauta ⌂, kun olet valmis jatkamaan." There is also a search icon and a green arrow icon in the bottom right corner.

Työtila: 0

suoritteilla

Koodilohkot

liiku eteenpäin

käänny oikeaan

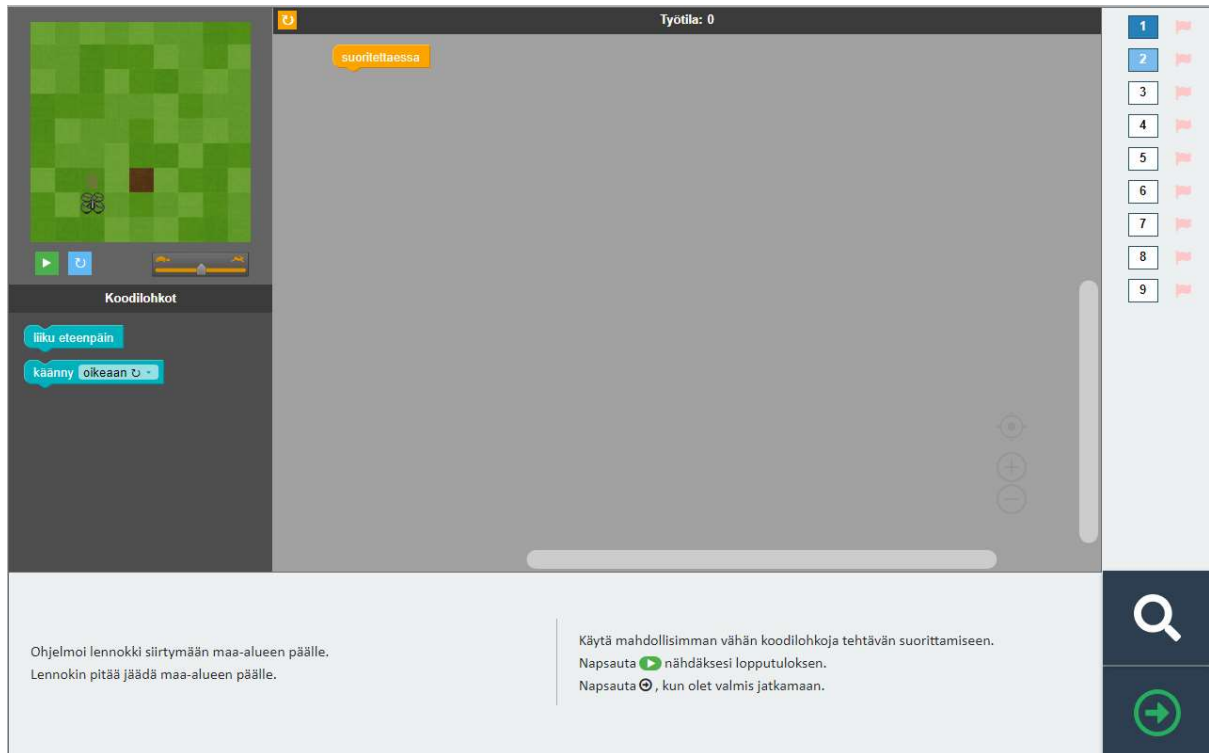
Käytä "liiku eteenpäin" -koodia siirtääksesi lennokkia yhden ruudun verran eteenpäin.

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.
Napsauta ▶ nähdäksesi lopputuloksen.
Napsauta ⌂, kun olet valmis jatkamaan.

Tehtävä 2: Ohjelmointitehtävä (käskysarja)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu: Algoritmien, ohjelmien ja rajapintojen kehittäminen

Osion toinen tehtävä oli ohjelmointitehtävä, jossa oppilaan tuli ohjelmoida lennokki liikkumaan ja pysähtymään maa-alueen päälle *liiku eteenpäin* ja *käänny oikeaan/vasempaan* -lohkoja hyödyntävällä käskysarjalla. Tehtävä oli vaativuustasoltaan helppo, sillä siihen sisältyi ainoastaan yhdentyypisiä komentoja (liikekomentoja) ja vain yksi kohderuutu.



The screenshot shows a programming environment with a central workspace and a left sidebar. The workspace has a title bar with a logo and the text "Työttilä: 0". Below the title bar is a large gray area with a small orange label "suoritettaessa". The left sidebar contains a green grid representing a map, a small airplane icon, and a "Koodilohkot" (Code Blocks) section. The code blocks section has two blocks: "liiku eteenpäin" (move forward) and "käänny oikeaan" (turn right). The bottom of the interface has a light blue area with instructions in Finnish. On the right side, there is a vertical list of numbers 1 through 9, each with a red flag icon. At the bottom right, there is a search icon and a green arrow icon.

Ohjelmoi lennokki siirtymään maa-alueen päälle.
Lennokin pitää jäädä maa-alueen päälle.

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.
Napsauta ▶ nähdäksesi lopputuloksen.
Napsauta ⏮, kun olet valmis jatkamaan.

Tehtävä 3: Virheidenkorjaustehtävä (toistorakenne)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu: Ratkaisuvaihtoehtojen suunnittelu ja arviointi

Kolmannessa tehtävässä oppilaan tuli korjata virheet (ns. "debuggaaminen") koodista, johon sisältyi kaksi uutta koodilohkoa. *Pudota*-koodilohkolla lennokka ohjelmoitiin pudottamaan siemeniä, vettä tai lannoitetta sen kulloiseenkin sijaintiin. *Toista-suorita*-koodilohko oli toistorakenne, jolla lohkon sisään asetettuja käskyjä voitiin toistaa haluttu määrä kertoja. Työtilaan valmiiksi laadittu algoritmi pudotti siemenet neljä kertaa ensimmäisen maa-alueen päälle. Oppilaan tuli korjata koodia siten, että lennokka pudotti siemenet neljän eri maa-alueen päälle. Toistorakenteen vuoksi tehtävää voidaan pitää edellistä tehtävää monimutkaisempana, sillä lennokin kulloistakin sijaintia ohjelman eri suorituskäskyissä ei voida suoraan päätellä ohjelmakoodissa näkyvien *liiku eteenpäin* -lohkojen lukumäärän perusteella (vrt. käskysarja, jossa toistorakenteen sijaan kukin liikkumislohko olisi asetettu koodiin erikseen).

Koodilohkot

- liiku eteenpäin
- käännä oikeaan
- pudota siemenet
- toista 3 kertaa
 - suorita

Työtila: 3

- suoritettaessa
- liiku eteenpäin
- toista 3 kertaa
 - pudota siemenet
- suorita

1 2 3 4 5 6 7 8 9

Koodilohkoja on sijoitettu työtilaan.
Lennokin pitää pudottaa siemenet neljän maa-alueen päälle.
Työtilan koodilohkot eivät tee tätä oikein.
Napsauta ► nähdäksesi ongelman.
Korjaa ongelma muokkaamalla työtilan koodilohkoja.

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.
Napsauta ► nähdäksesi lopputuloksen.
Napsauta ◀, kun olet valmis jatkamaan.

Tehtävä 4: Ohjelmointitehtävä (toistorakenne)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu: Algoritmien, ohjelmien ja rajapintojen kehittäminen

Neljäs tehtävä oli ohjelmointitehtävä, jossa oppilaan oli ohjelmoitava lennokka pudottamaan vettä neljälle kylvetylle maa-alueelle hyödyntämällä toistorakennetta.

The screenshot shows a programming interface with a game area on the left, a code editor in the center, and a task description at the bottom. The game area displays a green grid with a small brown ground patch and a bird icon. The code editor contains a script with the following blocks: "liiku eteenpäin", "kaanny oikeaan", "pudota vesi", and a "toista 3 kertaa" loop around the "pudota vesi" block. The task description at the bottom explains the goal: to program the bird to drop water on four specific ground tiles using a loop.

Työtia: 0

suorittamassa

Koodilohkot

- liiku eteenpäin
- kaanny oikeaan
- pudota vesi
- toista 3 kertaa
- suorita

Ohjelmoi lennokka pudottamaan vettä neljälle kylvetylle maa-alueelle "toista suorita" -koodilohkon avulla.
Lennokka ei saa pudottaa vettä ruohoalueille.

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.
Napsauta ▶ nähdäksesi lopputuloksen.
Napsauta ⏏, kun olet valmis jatkamaan.

Tehtävä 5: Ohjelmointitehtävä (kaksi toistorakennetta)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu: Algoritmien, ohjelmien ja rajapintojen kehittäminen

Viidennessä tehtävässä oppilaan tuli ohjelmoida lennokki pudottamaan siemenet kaikille kylvämättömille maa-alueille, mutta ei millekään muille alueille. Tätä tehtävää voidaan pitää edellistä tehtävää vaikeampana, sillä maa-alueita oli yhden rivistön sijaan kaksi, mikä edellytti toistorakenteiden hyödyntämistä, kun tavoitteena oli hyödyntää mahdollisimman vähän koodilohkoja algoritmin laatimisessa. Algoritmit, joissa toistorakennetta ei hyödynnetty, edellyttivät vähintään 22 koodilohkon käyttämistä, kun taas toistorakennetta käyttämällä tarvittavien koodilohkojen vähimmäismäärä oli 12 lohkoa.

The screenshot shows a programming interface with a game area on the left, a code editor in the center, and a right sidebar. The game area displays a green field with a brown path and a small character. The code editor contains a script with the following blocks: 'liiku eteenpäin', 'käänny oikeaan', 'pudota vesi', and a 'toista 3 kertaa' loop containing a 'suorita' block. The right sidebar shows a list of 9 steps, with the first five steps highlighted in blue. At the bottom, there is a text box with instructions in Finnish and a search icon.

Työtila: 0

suoritteissa

Koodilohkot

- liiku eteenpäin
- käänny oikeaan
- pudota vesi
- toista 3 kertaa
- suorita

Ohjelmoi lennokki pudottamaan siemenet kaikille kylvämättömille maa-alueille. Siemeniä ei saa pudottaa millekään muille alueille.

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen. Napsauta ▶ nähdäksesi lopputuloksen. Napsauta 🔍, kun olet valmis jatkamaan.

Tehtävä 6: Virheidenkorjaustehtävä (ehtolause toistorakenteen sisässä)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu, Algoritmien, ohjelmien ja rajapintojen kehittäminen

Kuudes tehtävä oli vaativa virheidenkorjaustehtävä. Lennokin oli pudotettava vettä sekä isoille että pienille viljelyskasveille, mutta lannoitetta tuli pudottaa vain pienille viljelyskasveille. Tehtävässä työtilaan oli sijoitettu valmiiksi viisi koodilohkoa, mukaan lukien toistorakenteen sisällä oleva ehtolause. Koska toistorakenne ei toiminut oikein, oli se muutettava toimivaksi ja tehokkaaksi siten, että siihen siirrettävä *liiku eteenpäin* -komento suoritettaisiin useita kertoja lennokin edetessä viljelyskasvien yllä. Toistorakenteen sisässä olevaa *jos-niin*-ehtolauseetta tuli niin ikään muokata siten, että lennokka pudottaisi vettä ja/tai lannoitetta oikein kulloinkin alla olevan viljelyskasvin koon perusteella. Oppilaita kehoitettiin käyttämään mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.

Tehtävästä sai 0–3 pistettä. Oppilaat, joiden laatimassa ratkaisussa vesi ja lannoite pudotettiin oikeisiin ruutuihin virheettömästi ja optimaalisen tehokkaasti (5 komentoa), saivat tehtävästä kolme pistettä. Kaksi pistettä sai, kun vesi ja lannoite pudotettiin virheettömästi oikeisiin viljelyskasveihin, mutta keskinkertaisen tehokkaasti. Yhden pisteen sai, kun vesi ja lannoite pudotettiin oikeisiin ruutuihin, mutta ratkaisuun sisältyi virheitä tai se oli tehoton (yli 13 komentoa).

Kansainvälisesti keskimäärin 56 prosenttia oppilaista sai tehtävästä vähintään yhden pisteen. Kaksi pistettä saaneita oli kansainvälisesti keskimäärin 28 prosenttia oppilaista ja kolme pistettä saaneita 4 prosenttia.

Koodilohkot

- liiku eteenpäin
- käänny oikeaan
- pudota vesi
- toista 3 kertaa
- suorita
- jos iso viljelyskasvi
- niin

Koodilohkoja on sijoitettu työtilaan.
Lennokin pitää

- pudottaa vettä kaikille viljelyskasveille (isot ja pienet)
- pudottaa lannoitetta vain pienille viljelyskasveille.

Työtilan koodilohkot eivät tee tätä oikein.
Napsauta ► nähdäksesi ongelman.
Korjaa ongelma muokkaamalla työtilan koodilohkoja.

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.
Napsauta ► nähdäksesi lopputuloksen.
Napsauta ►, kun olet valmis jatkamaan.

Suoritustaso: 2–4

Maa/alue	Vähintään 1/3 pistettä saavuttaneiden oppilaiden %-osuus	Vähintään 2/3 pistettä saavuttaneiden oppilaiden %-osuus	3/3 pistettä saavuttaneiden oppilaiden %-osuus
* Etelä-Korea	71 (1.27)	46 (1.47)	8 (0.67)
Suomi	64 (1.72)	35 (1.54)	6 (0.67)
Ranska	61 (1.49)	34 (1.27)	5 (0.65)
* Norja	53 (1.89)	32 (1.42)	5 (0.57)
Taiwan	70 (1.52)	43 (1.52)	5 (0.50)
* Tšekki	71 (1.03)	37 (1.16)	5 (0.45)
* Tanska	65 (1.47)	29 (1.32)	4 (0.62)
* Latvia	59 (2.09)	27 (1.90)	4 (0.56)
* Belgia (fl)	63 (1.87)	32 (1.75)	4 (0.53)
Saksa	59 (1.57)	27 (1.39)	4 (0.49)
Luxemburg	53 (1.31)	26 (1.03)	4 (0.48)
* Ruotsi	56 (1.84)	31 (1.42)	4 (0.47)
Slovakia	63 (1.44)	30 (1.21)	4 (0.46)
Kv. keskiarvo	56 (0.36)	28 (0.29)	4 (0.11)
* Itävalta	55 (1.81)	25 (1.43)	3 (0.36)
Italia	60 (1.32)	22 (1.04)	3 (0.47)
Malta	46 (1.52)	22 (1.19)	3 (0.42)
* Portugali	59 (1.78)	22 (1.11)	2 (0.41)
* Kroatia	36 (1.56)	15 (1.13)	2 (0.40)
* Uruguay	37 (1.74)	17 (1.19)	2 (0.38)
* Serbia	38 (2.02)	15 (1.15)	2 (0.37)
* Slovenia	47 (1.86)	20 (1.35)	2 (0.34)

* Katso oppilasotoksiin liittyvät huomautukset kansallisen raportin kuviosta 3.1.

() keskiarvo

Tehtävä 7: Ohjelmointitehtävä (ehtolause toistorakenteen sisällä)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu, Algoritmien, ohjelmien ja rajapintojen kehittäminen

Seitsemäs tehtävä oli vaativa ohjelmointitehtävä, jossa lennokki oli ohjelmoitava toimimaan samaan tapaan kuin edellisessä tehtävässä, mutta kohderuutuja (viljelyskasveja) oli tällä kertaa useampia ja ne olivat yhden sijaan kahdella rivistöllä. Tehtävässä oppilaiden tuli ohjelmoida lennokki pudottamaan vettä sekä isoille että pienille viljelykasveille, mutta lannoitetta tuli pudottaa vain pienille viljelykasveille (yhteensä 12 pudotustoimintoa kahdeksan eri viljelyskasvin päällä). Lennokki ei saanut pudottaa vettä tai lannoitetta ruohoalueille. Tehtävässä työtilassa ei ollut valmiiksi sijoitettuja koodilohkoja. Tehokkaimpaan mahdolliseen ratkaisuun sisältyi toistorakenteita ja sisäkkäisiä ehtolauseita (19 lohkoa). Ilman toistorakennetta ratkaisu oli mahdollista saada aikaan 24 koodilohkolla.

Tehtävästä sai 0–3 pistettä. Oppilaat, joiden laatimassa ratkaisussa vesi ja lannoite pudotettiin oikeisiin ruutuihin virheettömästi ja optimaalisen tehokkaasti saivat tehtävästä kolme pistettä. Kaksi pistettä sai, kun vesi ja lannoite pudotettiin oikeisiin viljelyskasveihin, mutta keskinkertaisen tehokkaasti. Yhden pisteen sai, kun vesi ja lannoite pudotettiin oikeisiin ruutuihin, mutta ratkaisuun sisältyi virheitä tai se oli tehoton.

Kansainvälisesti keskimäärin 48 prosenttia oppilaista sai tehtävästä vähintään yhden pisteen. Kaksi pistettä saaneita oli kansainvälisesti keskimäärin 24 prosenttia oppilaista ja kolme pistettä saaneita 12 prosenttia.

The screenshot shows a programming environment with a game interface on the left and a code editor on the right. The game interface displays a green field with yellow blocks and a small robot. The code editor shows a script with the following blocks: 'liiku eteenpäin', 'käänny oikeaan', 'pudota vesi', 'toista 3 kertaa', 'suorita', 'jos iso viljelykasvi', 'niin'. The right side of the interface has a score display showing 'Työtia: 0' and a list of numbers 1 through 9, each with a small icon. At the bottom, there is a text box with instructions: 'Ohjelmoi lennokki' followed by bullet points: '• pudottamaan vettä kaikille viljelykasveille (isot ja pienet)' and '• pudottamaan lannoitetta vain pienille viljelykasveille.' Below this, it says 'Lennokki ei saa pudottaa vettä tai lannoitetta ruohoalueille.' To the right of this text box, there is a note: 'Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen. Napsauta [play icon] nähdäksesi lopputuloksen. Napsauta [stop icon], kun olet valmis jatkamaan.'

Suoritustaso: 2–4

Maa/alue	Vähintään 1/3 pistettä saavuttaneiden oppilaiden %-osuus	Vähintään 2/3 pistettä saavuttaneiden oppilaiden %-osuus	3/3 pistettä saavuttaneiden oppilaiden %-osuus
* Etelä-Korea	67 (1.34)	45 (1.34)	24 (1.11)
Taiwan	69 (1.64)	43 (1.51)	21 (1.03)
Suomi	57 (1.95)	30 (1.26)	17 (1.13)
* Tšekki	64 (1.38)	32 (1.15)	17 (0.87)
* Norja	49 (1.86)	28 (1.42)	15 (1.11)
Ranska	52 (1.54)	28 (1.10)	15 (0.95)
* Tanska	56 (1.65)	27 (1.28)	14 (1.29)
* Ruotsi	49 (2.07)	29 (1.50)	14 (0.82)
* Latvia	54 (2.21)	24 (1.90)	13 (1.33)
* Belgia (fl)	56 (2.07)	27 (1.89)	13 (1.11)
Slovakia	56 (1.45)	26 (1.28)	13 (0.95)
Luxemburg	45 (1.22)	23 (1.15)	12 (0.84)
Kv. keskiarvo	48 (0.37)	24 (0.28)	12 (0.20)
Saksa	47 (1.53)	21 (1.07)	10 (0.79)
* Itävalta	45 (1.77)	22 (1.38)	9 (0.86)
Italia	47 (1.64)	18 (0.99)	9 (0.73)
Malta	38 (1.14)	17 (0.91)	9 (0.68)
* Slovenia	36 (1.88)	17 (1.47)	7 (0.81)
* Portugali	49 (1.82)	17 (1.00)	7 (0.67)
* Kroatia	28 (1.86)	12 (1.27)	6 (0.89)
* Uruguay	28 (1.63)	12 (0.98)	6 (0.60)
* Serbia	27 (1.87)	11 (0.93)	5 (0.65)

* Katso oppilasotoksiin liittyvät huomautukset kansallisen raportin kuviosta 3.1.

() keskivirhe

Tehtävä 8: Virheidenkorjaustehtävä (sisäkkäiset toistorakenteet)

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu: Ratkaisuvaihtoehtojen suunnittelu ja arviointi

Kahdeksas tehtävä oli virheidenkorjaustehtävä, jossa oppilaan tuli ohjelmoida lennokki pudottamaan vettä vain viljelykasveille. Korjattavaan algoritmiin sisältyi sisäkkäisiä toistorakenteita.

The screenshot shows a programming environment with a game area on the left and a code editor on the right. The game area displays a drone on a grid with yellow blocks. The code editor contains the following script:

```
suorittaessa
toista 2 kertaa
  suorita
    toista 4 kertaa
      liiku eteenpäin
      pudota vesi
    käännä vasempaan
    liiku eteenpäin
    liiku eteenpäin
    käännä vasempaan
  toista 4 kertaa
    suorita
      liiku eteenpäin
      pudota vesi
    käännä oikeaan
    liiku eteenpäin
    liiku eteenpäin
    käännä oikeaan
```

Below the code editor, there is a list of code blocks (Koodilohkot) and a set of instructions.

Koodilohkot

- liiku eteenpäin
- käännä oikeaan
- pudota vesi
- toista 3 kertaa
- suorita
- jos iso viljelykasvi
- niin

Instructions:

Koodilohkoja on sijoitettu työtilaan.
Lennokin pitää pudottaa vettä vain viljelykasveille.
Lennokki ei saa pudottaa vettä ruohoalueille.
Työttilan koodilohkot eivät tee tätä oikein.
Napsauta ▶ nähdäksesi ongelman.
Korjaa ongelma muokkaamalla työttilan koodilohkoja.

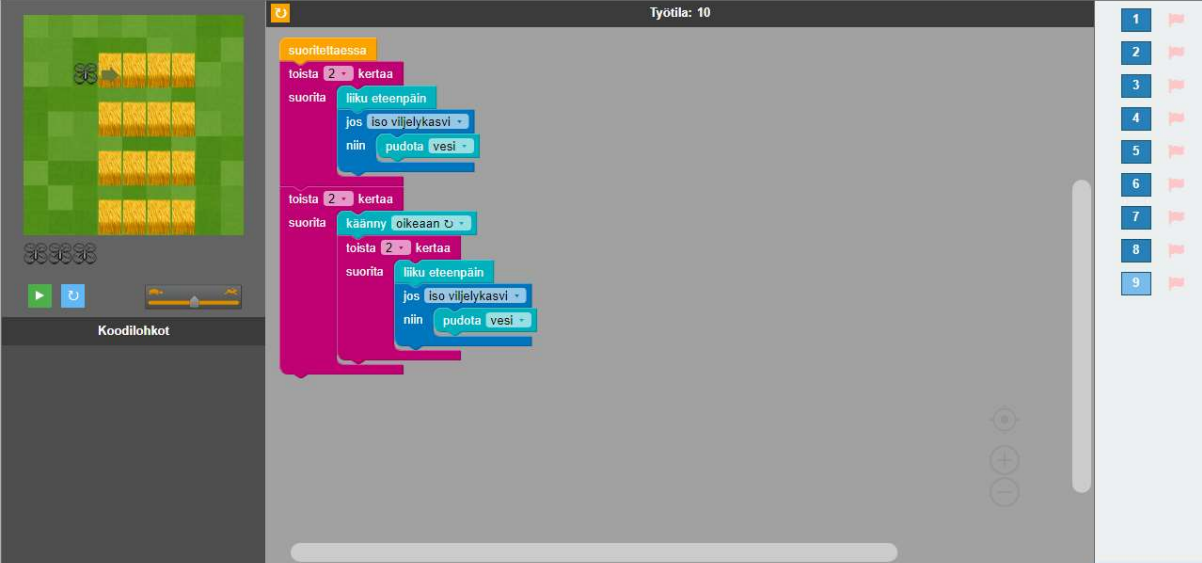
Additional Instructions:

Käytä mahdollisimman vähän koodilohkoja tehtävän suorittamiseen.
Napsauta ▶ nähdäksesi lopputuloksen.
Napsauta ⏮, kun olet valmis jatkamaan.

Tehtävä 9: Lennokkien asettaminen

Ohjelmoinnillisen ajattelun arviointikategoria: Ongelmanratkaisu, Algoritmien, ohjelmien ja rajapintojen kehittäminen

Viimeisessä tehtävässä ei edellytetty koodilohkojen asettamista tai korjaamista, vaan työtilaan laadittu ohjelmakoodi ohjasi neljää lennokkia, jotka oppilaan tuli vetää ja pudottaa maatilalle. Maatilalle sijoitettua lennokkia oli mahdollista siirtää eri aloituskohtaan napsauttamalla. Neljä lennokkia oli sijoitettava siten, että ne pudottivat vettä kaikille viljelykasveille. Tehtävässä arvioitiin rinnakkaislaskentaan liittyviä taitoja.



The screenshot shows a programming environment with a stage on the left and a code editor on the right. The stage displays a farm scene with a green field, a yellow sun, and a small house. A 'Koodilohkot' (Code Blocks) panel is visible at the bottom left. The code editor contains the following script:

```
suorittaessa  
toista 2 kertaa  
  suorita  
    liiku eteenpäin  
    jos iso viljelykasvi  
    niin pudota vesi  
toista 2 kertaa  
  suorita  
    käänny oikeaan  
    toista 2 kertaa  
      suorita  
        liiku eteenpäin  
        jos iso viljelykasvi  
        niin pudota vesi
```

On the right side of the code editor, there is a vertical list of numbers 1 through 9, each with a small red 'x' icon next to it. At the bottom of the interface, there is a light blue area with instructions in Finnish:

Työtilän koodilohkoja ei voi muuttaa.
Maatilalle sijoitettuja lennokkeja ohjataan koodilohkoilla.
Voit napsauttaa lennokkia ja vaihtaa sen suuntaa.
Vedä ja pudota lennokit siten, että vesi putoaa kaikille viljelykasveille.

Napsauta ▶ nähdäksesi, miten maatilalla oleva lennokki pudottaa vettä viljelykasveille.
Napsauta ⏹, kun olet valmis jatkamaan.

At the bottom right, there is a search icon (magnifying glass) and a green arrow icon.