Informatika pradiniame ugdyme

Edukacinis žaidimas „Scottie Go!“

Pasirengimas darbui su žaidimu *Scottie Go!*

Pasirengimui reikalingi arba kompiuteriai, arba planšetiniai kompiuteriai, būtinas interneto ryšys. Pasirengimas gali būti skirstomas į 4–6 kartus po 20 min. Svarbu, kad mokiniai suprastų esminius veiksmus ir tai, jog tie veiksmai gali būti vaizduojami įvairiais blokeliais ir komandomis. Pavyzdžiui, komandos „dešinėn“ ir „kairėn“ gali būti užrašomos „Rytai, Vakarai, Pietūs, Šiaurė“. Mokiniai gali sėdėti po du prie vieno kompiuterio.

Pradedant naudoti mokomąjį žaidimą *Scottie Go!*, rekomenduojame panagrinėti kelis paprastus blokelių kalbos pradmenis įtvirtinančius uždavinius. Juos atveriame, užrašę šį adresą: https://studio.code.org/hoc/1.
Atvėrę pirmą uždavinį matome pagrindinių laukelių paaiškinimus. Juos naikiname spustelėdami viršutiniame kampe esantį kryželį.
Pirmieji du uždaviniai suformuoja robotuko (šiuo atveju paukštuko) elgesį, kai tikslas pasiekiamas keliais žingsniais. Mokytojas kartu su mokiniais nagrinėja uždavinius ir aptaria atliekamus veiksmus. Nuo 16 uždavinio galima pasinaudoti ir programos atlikimu pažingsniui. Tai labai naudinga mokiniams, kuriems kyla sunkumų. Mokytojas pademonstruoja paprasčiausius veiksmus, parodo, kaip netinkamas blokelis ar keli blokeliai yra sunaikinami, juos numetant atgal į šiukšlinę. Nors visi uždaviniai turi patarimų laukelį, būtų labai naudinga, kad mokytojas susitartų su mokiniais į patarimų laukelį nežiūrėti.
Vaikai turi suvokti, kad komanda „ženk į priekį“ yra universali ir nepriklauso nuo paukštuko orientacijos: ji gali būti žemyn arba aukštyn, dešinėn arba kairėn. Žengiama ta kryptimi, kuria žvelgia paukštukas. Komanda „ženk į priekį“ atlieka tik vieną žingsnį.

1 ir 2 uždaviniai

3 ir 4 uždaviniai įtvirtina komandą, kuri pasuka paukštuką 90 laipsnių kampu. Svarbu, kad mokiniai atidžiai išsiaiškintų, į kurią pusę turi pasisukti paukštukas. Uždaviniuose komandos „ženk į priekį“ ir „pasisuk į kairę / dešinę“ atliekamos nuosekliai viena paskui kitą. Mokiniai panagrinėja ir blokelio „pasisuk į kairę / dešinę“ nustatymą atitinkamam veiksmui: kairėn arba dešinėn.

3 ir 4 uždaviniai

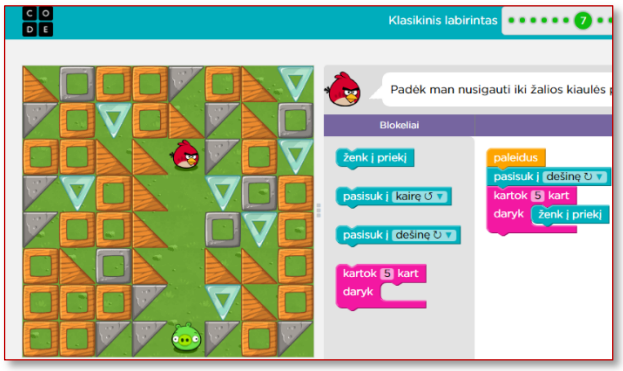
Analogiškas ir 5 uždavinys, tik jame atliekama daugiau veiksmų, kurių kiekis verčia susimąstyti apie kartojamų veiksmų programavimą.

5 uždavinys

6 uždavinys supažindina mokinius su kartojamų veiksmų automatizavimu. Įvedamas sudėtingesnis blokelis „kartok *n* kartų“. Mokiniai gali įrašyti reikiamą kartojimų skaičių, o į kartojimo blokelio vidų įterpti teisingą komandą. Šiuos atveju – komandą „ženk į priekį“, kuri atliekama 5 kartus.

6 uždavinys

7 uždavinys vaizduoja kartojimo derinimą su kitais veiksmais. Mokiniai savarankiškai ar dirbdami po du turi išsiaiškinti, kad neteisinga paukštuko orientacija, todėl, prieš žengdamas, jis turi pasisukti dešinėn. 8 uždavinys sudėtingesnis – kartojimo komanda naudojama du kartus, nes nuoseklų veiksmų kartojimą nutraukia posūkis kairėn.



7 ir 8 uždaviniai

9 uždavinys įtvirtina kelių iš eilės einančių veiksmų kartojimą. Tai pakankamai sudėtingas algoritmas, ir mokytojui gali tekti išsamiau mokiniams paaiškinti veiksmų seką. Norint pasirinkti sėkmingą sprendimą, reikia detaliai išsinagrinėti paukštuko judėjimą.

9 uždavinys

10 uždavinys sutampa su 6 uždaviniu. Mokiniai tai gali ir pasitikrinti. Šio sprendimo pagrindinė mintis – tai naujas būdas automatizuoti pasikartojančius veiksmus. Tačiau dabar kartojama kiekvieną kartą tikrinant, ar nepatenkinta sąlyga. Ant avietinio blokelio nupiešta kiaulytė ir yra kartojimo sąlyga.

10 uždavinys

Analogiškai 11 uždavinys sutampa su 9 uždaviniu, todėl labai svarbu formuoti mokinių nuomonę, kad uždavinį dažniausiai galima spręsti keliais būdais. Svarbiausia, kad būtų teisingai pasiektas tikslas ir kad būtų daroma kuo mažiau veiksmų.

11 uždavinys

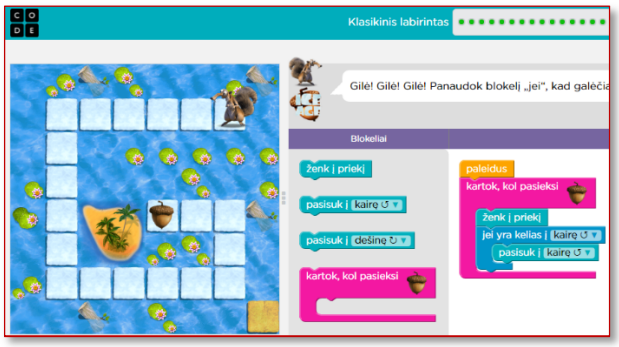
12 uždavinys taip pat yra sudėtingas. Mokiniai labai atidžiai išsinagrinėja zombio maršruto dėsningumus.
13 uždavinys įtvirtina 12 uždavinio idėjas, tik zombis keliauja priešinga kryptimi – žemyn.

12 ir 13 uždaviniai

14 uždavinys supažindina su vienu svarbiausių programavimo elementų – veiksmų šakojimu arba pasirinkimu, kurį veiksmą atlikti. Veiksmas atliekamas tik tada, jei patenkinta sąlyga – „yra kelias kairėn“. Šis uždavinys vaizduoja veiksmą, jei sąlyga patenkinta. Priešingu atveju neatliekamas joks kitas veiksmas, o grįžtama prie kartojimo veiksmo.

14 uždavinys

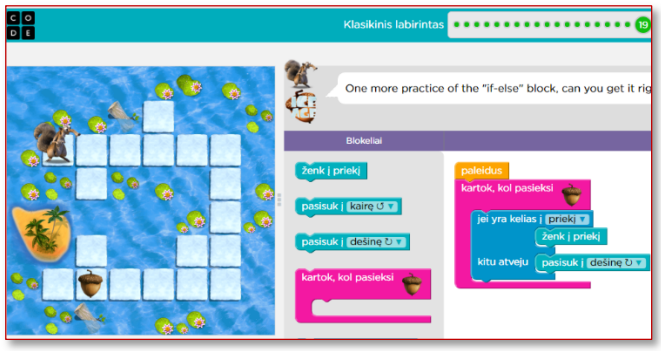
15, 16 ir 17 uždaviniai įtvirtina veiksmų šakojimą.



15 ir 16 uždaviniai

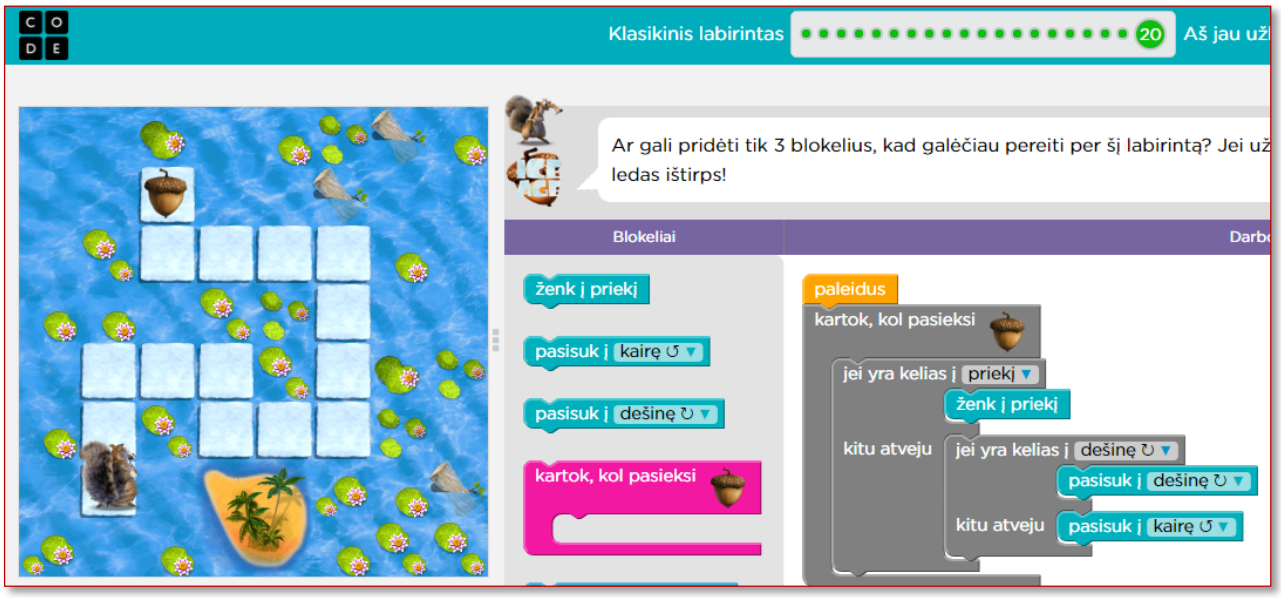
17 uždavinys

18 ir 19 uždaviniai atskleidžia visą šakojimo veiksmą: jei sąlyga patenkinama, atliekamas pirmas veiksmas, o jei ne (bet kuriuo kitu atveju), – atliekamas antrasis veiksmas. Mokiniams išaiškinama, kad būtinai bus veiksmas, o koks jis bus, priklauso nuo sąlygos patenkinimo.



18 ir 19 uždaviniai

Paskutinis uždavinys demonstruoja sudėtinį šakojimo veiksmą, kai pirma sąlyga nėra patenkina, dar kartą vyksta naujos sąlygos tikrinimas. Tai pakankamai sudėtinga ir suvokiama aukštesnių pasiekimų mokiniams.



20 uždavinys

Taigi mokiniai, išnagrinėję šiuos uždavinius, suvokia robotuko (šiuo atveju paukštuko ar zombio) valdymo principus: komandas priekin, pasisukti, kartoti veiksmus, kai tiksliai žinoma, kiek kartų juos reikia kartoti, kartoti veiksmus, kai kiekvieną kartą prieš kartojant yra tikrinama sąlyga, ar dar nepasiektas tikslas, veiksmų atlikimą, atsižvelgiant į tai, ar sąlyga patenkinama, ar ne, taip pat šių komandų kombinacijas.

Mokytojas su mokiniais apibendrina šiuos uždavinius pasinaudodamas toliau pateikiamais tinklalapiais.

1. Tinklalapiu (kairėn ir dešinėn čia bus pakeista į pasaulio šalių kryptis): https://studio.code.org/s/course1/stage/7/puzzle/1

2. Tinklalapiu (žinomo kartų skaičiaus kartojami veiksmai) https://studio.code.org/s/course3/stage/2/puzzle/4, https://studio.code.org/s/course3/stage/2/puzzle/5, https://studio.code.org/s/course3/stage/2/puzzle/6, https://studio.code.org/s/course3/stage/2/puzzle/7, https://studio.code.org/s/course3/stage/2/puzzle/8, galima naudoti ir tinklalapį https://studio.code.org/s/course2/stage/8/puzzle/1 (visus pratimus).
3. Tinklalapiu (kartojimas, kol bus patenkinta sąlyga) https://studio.code.org/s/course3/stage/2/puzzle/9, https://studio.code.org/s/course3/stage/2/puzzle/10, https://studio.code.org/s/course3/stage/2/puzzle/11, https://studio.code.org/s/course3/stage/2/puzzle/12, https://studio.code.org/s/course3/stage/2/puzzle/13.
4. Tinklalapiu (šakojimo uždaviniai) https://studio.code.org/s/course2/stage/13/puzzle/3.

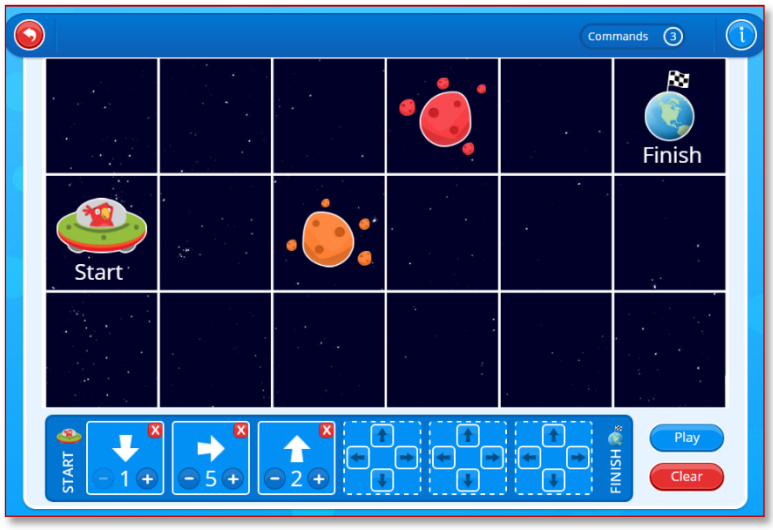
Daug kartojimo pratimų yra čia: https://studio.code.org/s/course2

Taip pat labai praverstų išnagrinėti „Bebro“ konkurso uždavinius (http://www.upc.smm.lt/projektai/bebras/2016/; http://www.upc.smm.lt/projektai/bebras/2015). Pavyzdžiui, uždavinys „Robotas-rutulys“: http://www.upc.smm.lt/projektai/bebras/2016/rodyti.php?id=248.

Šis uždavinys yra interaktyvus. Visi interaktyvūs uždaviniai pažymėti smagratuku.

Pastaba. Dar galima pasinaudoti http://lithuania.dbprimary.com/. Prisijungę prie 2 kl. aplinkos, galime išbandyti algoritmo kūrimą, kai naudojamos krypties rodyklės ir nustatytas žingsnių skaičius. Tai labai panašu į veiklas su „Blue-Bot“ robotukais.

Pavyzdys, kai „rašoma“ paprasčiausia programėlė papūgiuko aparatui, kuris turi sėkmingai aplenkti kliūtis ir pasiekti Žemę.



https://studio.code.org/

Mokinių gebėjimai

Orientuojasi paprasčiausioje programavimo aplinkoje:

- *suvokia užduotį;*
- *planuoja jos vykdymo etapus;*
- *geba konstruoti nuoseklius veiksmus atitinkantį algoritmą;*
- *geba robotuką nukreipti reikiama kryptimi – į priekį, pasisukti į kairę (į dešinę);*
- *geba konstruoti ciklą, kai žinomas kartojimų skaičius;*
- *į žinomo kartojimų skaičiaus konstruktą įterpia kelis nuoseklius veiksmus;*
- *geba konstruoti ciklą, kai kartoja tiek kartų, kol yra patenkinama sąlyga;*
- *atpažįsta ir šakojimą ir jį panaudoja, spręsdamas užduotį.*

Žaidimas *Scottie Go!*

Scottie Go! – tai edukacinis žaidimas, skatinantis vaikus logiškai ir algoritmiškai mąstyti, ugdyti erdvinį mąstymą, kūrybiškumą, mokytis dirbti komandoje.

Reikalingas ne tik žaidimas *Scottie Go!* (179 blokeliai), bet ir planšetiniai kompiuteriai, kuriuose įdiegta programėlė. Programėlė yra pritaikyta *Android, Windows* ir *IOS* operacinėms sistemoms.

Atliekant kiekvieną užduotį, paprastai būna specialūs nurodymai, kaip ji turi būti atlikta. Tačiau mokiniai gali visada pasirinkti paprastesnius atlikimo būdus. Žinoma, tokiu atveju, užduoties rezultatas negaus aukščiausio įvertinimo (trijų žvaigždučių

, o tik vieną

arba dvi).

Pirmuosius tris etapus rekomenduotina atlikti 1 ar 2 klasėse, kitus šešis – 3 ir 4 klasėse. Mokiniai turėtų kurti algoritmus įvairiais būdais ir juos aptarti. Svarbu, kad jie nebijotų klysti, analizuotų klaidas ir ugdytųsi kantrybę, taip pat pajustų džiaugsmą, kai robotukas pasiekia tikslą.

Dešimtąjį etapą, kuriame naudojamos funkcijos, rekomenduotina atlikti tik su pačiais gabiausiais ketvirtokais.

Pastaba. TIK įveikus visas etapo užduotis galima pereiti į kitą etapą.

Pirmasis etapas

Pirmajame etape mokiniai susipažįsta su žaidimu: pagrindiniais blokeliais, jų dėliojimo tvarka, programėle, esančia planšetiniame kompiuteryje (naudotis išmaniuoju telefonu nepatogu). Klasė susitaria dėl darbo tvarkos, nes šis žaidimas bus žaidžiamas ne vieną pamoką. Nuo pat pirmos pamokos mokytojui svarbu pabrėžti įvairias robotuko „maršrutų“ galimybes – tikslas gali būti pasiekiamas įvairiais keliais, kuriant skirtingus algoritmus. Svarbu, kad mokiniai neskubėtų, išbandytų įvairias galimybes, diskutuotų, pakartotų.

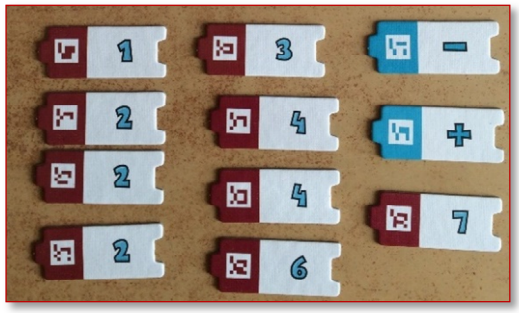
Pastaba. Dėžutėje yra net 179 blokeliai (detalės), iš pradžių visi blokeliai nėra reikalingi. Ant blokelių užrašyta įvairi informacija. Kai kurie blokeliai sutapatinami su komandomis ar veiksmais, kai kurie reiškia skaičius, ciklo pradžią ar pabaigą.

Pirmajam žaidimo etapui tikrai užteks šių blokelių:

,

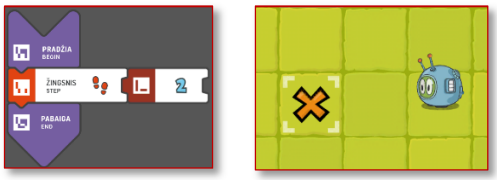
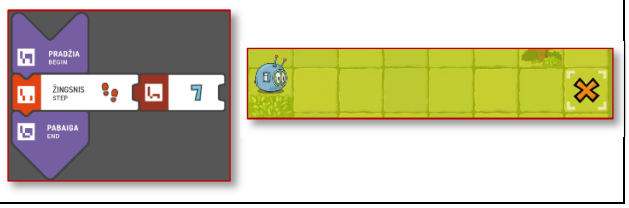
,

,

ir blokelių su skaičiais 

Kiekviena programėlė pradedama blokeliu *PRADŽIA* (*BEGIN*) ir užbaigiama blokeliu *PABAIGA* (*END*). Atsarginių šių blokelių nėra! Blokelis *ŽINGSNIS* (*STEP*) naudojamas, kai reikia žengti nuo vieno iki *N* žingsnių. Prie blokelio būtina pridėti blokelius su skaičiais, kurie parodo, kiek žingsnių robotukas turi žengti.

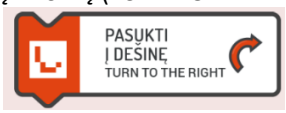
Jei žengiamas tik vienas žingsnis, būtina pridėti skaičių 1. Dešiniajame paveiksliuke parodyta programėlė, kai bus žengiami 7 žingsniai (7 langeliai).

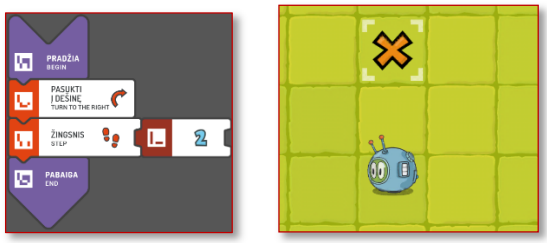
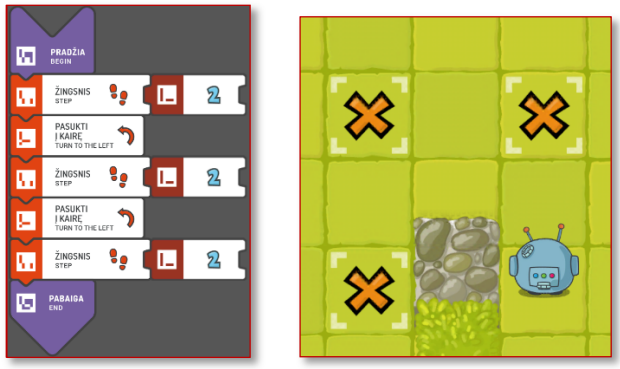
Robotukas žengs du žingsnius	Robotukas žengs septynis žingsnius
	

Klausimas. Kuo skiriasi šio žaidimo komandos *ŽINGSNIS* užrašymas nuo naudojamo analogiško *studio.code.org*? Ar atliekami tokie patys veiksmai?

Atsakymas. *studio.code.org* žengiama po vieną žingsnį

be užrašo.

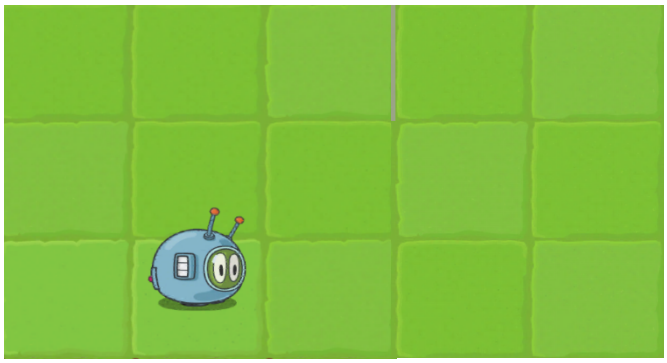
Šiame pavyzdyje komanda, kurios blokelis *PASISUKTI Į DEŠINĘ* (*TURN TO THE RIGHT*),  pasuka robotuką dešinėn 90 laipsnių kampu. Toliau komanda, kurios blokelis *ŽINGSNIS* (STEP), nurodo robotukui žengti du žingsnius (langelius).	Šiame pavyzdyje blokelis *PASISUKTI Į KAIRĘ* (*TURN TO THE LEFT*) pasuka robotuką kairėn per 90 laipsnių. Toliau komanda, kurios blokelis *ŽINGSNIS* (*STEP*), nurodo robotukui žengti du žingsnius (langelius). Toliau vėl kartojami blokeliuose vaizduojami veiksmai.

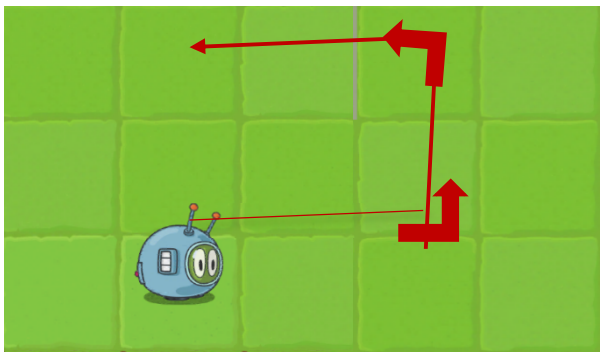
Klausimas. Kuo skiriasi šio žaidimo komandos *PASISUKTI (Į KAIRĘ / DEŠINĘ)* užrašymas nuo naudojamo analogiško *studio.code.org*? Ar atliekami tokie patys veiksmai?

Atsakymas. Naudojamas tas pats blokelis , nes jame galima pakeisti kryptį, spustelėjus pele trikampiuką.

Klausimas. Kuriame langelyje atsidurs robotukas, kai bus įvykdyta ši programa? Į kurią pusę robotukas tada žiūrės?



Atsakymas. Robotukas žiūrės priešinga pradinei padėčiai kryptimi (kairėn).

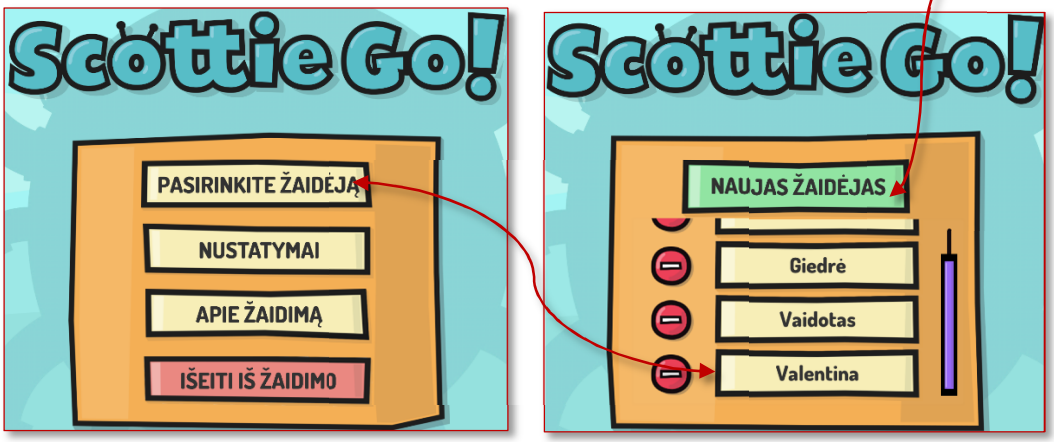


Žaidimas pradedamas atvėrus programėlę ***Scottie Go! Edu*** planšetiniame kompiuteryje.

Mokytojas paaiškina žaidimo idėją. Vaikai gali peržiūrėti trumpą žaidimo pradžios filmuką.

Iš kitos planetos keliavęs robotukas **Skotis** (Scottie) nusileido Žemėje nevisiškai sėkmingai. Norėdamas tęsti kelionę, robotukas visoje Žemėje ieško pamestų savo erdvėlaivio detalių. Mokiniai gali padėti robotukui tik spręsdami užduotis ir jas patikrindami mobiliaisiais įrenginiais.
Atvėrus programėlę pasirenkama kalba.

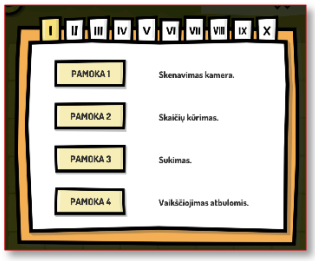
Atvėrus programėlę, kiekvienam mokiniui galima priskirti vardą (spustelimas užrašas *NAUJAS ŽAIDĖJAS*). Ištrynus vardą (spustelint raudoną mygtuką su minuso ženklu), visi žaidėjo duomenys sunaikinami.



Žaidimas prasideda Europoje, paskui bus keliaujama ir per kitus žemynus.

Paveikslėlyje matomų mygtukų reikšmės

Pagrindinis meniu	Pagalba / Pamokėlės	Surinktų detalių skaičius

Pirmasis etapas

Pirmojo etapo užduotys labai paprastos. Svarbu, kad mokiniai susipažintų su žaidimu, jo aplinka planšetiniame kompiuteryje, pagrindiniais aplinkos meniu punktais. Klasėje svarbu susikurti darbo ir vertinimo taisykles. Žaidime yra daug įvairių blokelių, todėl svarbu dirbti tvarkingai.

Pirmajame etape robotuko tikslas – aplankyti visus langelius, pažymėtus oranžiniu kryželiu. Žinoma, reikia išvengti kliūčių. Jei mokiniai jau yra susipažinę su *Blue-Bot* valdymu naudojant planšetinius kompiuterius arba https://studio.code.org aplinka, tvarkytis su užduotimis bus paprasta. Norint gauti pirmąją erdvėlaivio detalę, teks išspręsti 10 paprastų užduočių. Užduotys sprendžiamos iš eilės. Galima bet kada sugrįžti tik į jau išspręstą užduotį. Neišspręstos užduotys žymimos pilkais rutuliukais.

1 užduotis. Robotukas turi žengti du žingsnius, kad atsistotų ant kryželio . Atsistojus ant kryželio, šis pavirsta tokiu ženkliuku .

Sudėliojus programėlę, reikia ją nufotografuoti. Spustelimas fotoaparato mygtukas. Kai ekrane atsiranda blokelių vaizdas, dar kartą spustelime fotoaparato mygtuką. Po to reikia spustelti žalią mygtuką, kuriuo pavirsta fotoaparato mygtukas. Jei vaizdo nepavyksta gauti, spustelima oranžinė rodyklytė ir pakartojamas fotografavimas. Spustelėjus kryžiuką, fotografavimas nutraukiamas. Vėl iš naujo įkeliama ta pati užduotis ir galima vėl kartoti sudėliotos programėlės fotografavimą.

Jei robotukas atliko užduotį teisingai, atsiranda sveikinimo pranešimas:

. Trys žvaigždutės užsidega tik tada, jei, žaidimo autorių nuomone, rastas trumpiausias sprendimas.
Jei užduoties lauke yra langelių, kuriuose vaizduojami akmenys, medžiai ir pan., tai robotukas negali keliauti tokiais langeliais. Pavyzdžiui, jei yra tokie langeliai, tai viršutinėje langelių linijoje robotukas negalės žengti į vandenį.

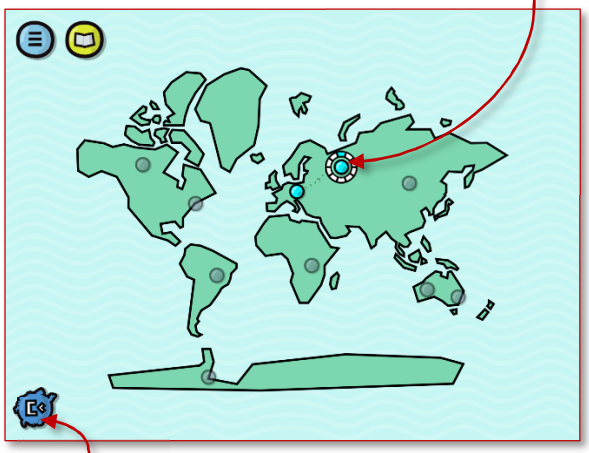
Tačiau pirmieji du langeliai ne iki galo yra užimti, todėl tokiais langeliais robotukas gali keliauti. Robotukas gali keliauti ir truputį pakeltais langeliais.

Užduotis. Išsiaiškinti, kokius pranešimus pateikia programa, jei sudėliotoje programoje yra klaida.

8 užduotyje panaudojamas naujas veiksmas. Robotukas gali judėti ir atbulas.
Tada prie blokelio *ŽINGSNIS (STEP)* pridedamas blokelis su minuso ženklu.

Pirmojo etapo (EUROPA 1) paskutinė 10 užduotis (reikia atsistoti ant 4 kryžiukų) gali būti sprendžiama taip:

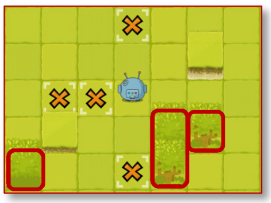
Užbaigus pirmąjį etapą, naujas etapas pažymimas ryškiu mėlynu burbuliuku su žiedu aplinkui. Užbaigtas etapas kairiajame paveikslėlyje yra su mėlynu burbuliuku, nes ne visi uždaviniai gavo trijų žvaigždučių įvertinimą. Dešiniajame paveikslėlyje jau patobulinti pirmojo etapo uždaviniai, todėl burbuliukas čia jau geltonos spalvos.



Spustelėję šį mėlyną paveikslėlį, matome, kad jau surasta viena detalė iš dešimt, kurių reikia robotukui, kad pataisytų savo erdvėlaivį.

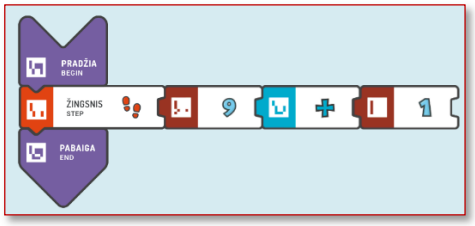
Klausimai

1. Kuriais langeliais **negali** keliauti robotukas?

Atsakymas. 

2. Kada patogu naudotis robotuko galimybe vaikščioti atbulomis?

3. Kiek žingsnių žengs robotukas? Kiek žingsnių žengtų robotukas, jei vietoje blokelio su pliuso ženklu būtų blokelis su minuso ženklu?



Kai kurių užduočių rezultatų vaizdai

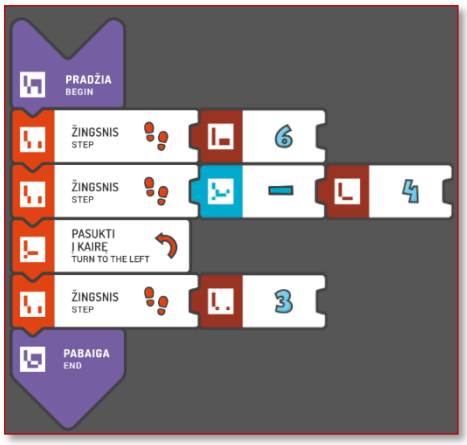
6 užduotis

7 užduotis

Atlikę 7 užduotį matome tokį vaizdą:

9 užduotis

0 / 3
Užduotis Nr. 9.


PRADŽIA
BEGIN
ŽINGSNIS
STEP
6
ŽINGSNIS
STEP
4
PASUKTI
Į KAIRĘ
TURN TO THE LEFT
ŽINGSNIS
STEP
3
PABAIGA
END

Antras etapas

Antrame etape mokiniai išmoksta naujų komandų: PAIMK (PICK UP) ir ŠOK (JUMP). Pirmoji komanda leidžia paimti reikalingus objektus (čia bus renkamos sraigės), o antroji – peršokti tam tikras kliūtis, kai trumpam galima atsispirti nuo kliūties. Mokiniai išmoksta savais žodžiais nupasakoti užduočių formuluotes, robotuko veiksmus. Mokiniai įpranta prie žaidimo aplinkos, geba pasiskirstyti vaidmenimis, patikrinti programą, ją tobulinti (jei gaunamos ne visos trys žvaigždutės), susitvarko savo darbo vietą.

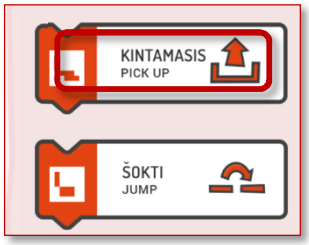
Atvėrę programėlę, pasirenkame žaidėją, kuris jau įveikė pirmąjį etapą. Primename, kad į antrąjį etapą gali pereiti mokinys, kuris teisingai išsprendė visus pirmojo etapo uždavinius. Jie gali būti įvertinti nebūtinai visomis trimis žvaigždutėmis, svarbu, kad būtų išspręsti teisingai.

Galima grįžti į bet kurį jau išspręstą etapą ir pasirinkti norimą uždavinį. Jei norime išeiti iš esamo etapo, spustelime viršuje meniu ir, atsivėrus kortelei, spustelime *IŠEITI IŠ LYGIO (LEAVE LEVEL)* .

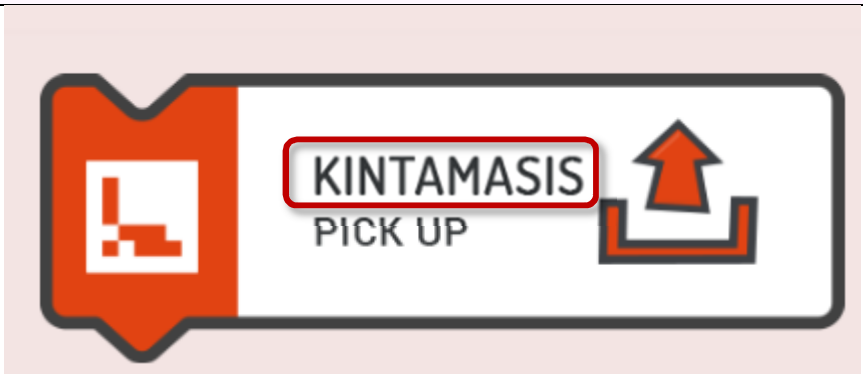
Į neišspręstus etapus keliaujama iš eilės. Etapo užduotys taip pat sprendžiamos iš eilės.

Antrajame (Europa 2) etape teks rinkti objektus (sraiges) ir šokinėti per kliūtis.

Antrojo etapo užduotyse bus naudojami du nauji blokeliai. Vienas iš jų atsakingas už komandą *PAIMK (PICK UP)*, o kitas – už *ŠOKTI (JUMP)*.



DĖMESIO!
Sulietuvintuose blokeliuose yra didelė klaida.
Vietoje komandos *PAIMK* yra užrašas *KINTAMASIS*. Reikia mokiniams tai atidžiai išaiškinti ir kaip nors sužymėti šiuos blokelius.



Pirmajame etape robotukui užtekdavo tik užlipti ant kryžiuko, o dabar jis turi surinkti sraiges. Norint surinkti sraigę, reikia panaudoti komandą *PAIMK (PICK UP)*. Komanda veikia, kai robotukas yra tame pačiame langelyje, kuriame yra paimtinas objektas.

Pirmoje užduotyje robotukas ne tik pakelia sraigę, bet ir ją nuneša į rudos aikštelės (vadinamos bėgimo takeliu) bet kurį langelį.

Parašyta programa nurodo, kad robotukas turi žengti du žingsnius, tada paimti sraigę ir dar tiesiai nueiti 4 žingsnius, kad patektų į rudos spalvos aikštelę.

Šeštoje užduotyje jau atsiranda kliūtys – vandenyje plaukiojančios salelės. Blokelis su komanda *ŠOKTI (JUMP)* atlieka tokius veiksmus: robotukas užšoka ant langelio su kliūtimi ir šoka tiesiai ant kito langelio. Robotukas turi surinkti tris sraiges ir atlikti du šuolius. Jei robotukas šuolių išvengtų, sukurta programa būtų labai ilga ir gremėzdiška. Užduoties programa vaizduoja tokius robotuko veiksmus: robotukas žengia vieną žingsnį, pasisuka kairėn, tada žengia į langelį, kuriame yra sraigė, sraigę pasiima ir šoka per salelę į langelį su kita sraige, vėl šuolis į paskutinės sraigės langelį ir du žingsnius – į rudos spalvos aikštelę.

Nors 10 užduotyje yra net penkios kliūtys, tačiau visiškai nebūtina ant jų robotukui šokinėti, kad paimtų vieną sraigę ir nuneštų ant rudos aikštelės. Robotukas žengia du žingsnius, pasiima sraigę, pasisuka kairėn, žengia 6 žingsnius ir, pasisukęs kairėn, žengia vieną žingsnį į rudos spalvos aikštelę.

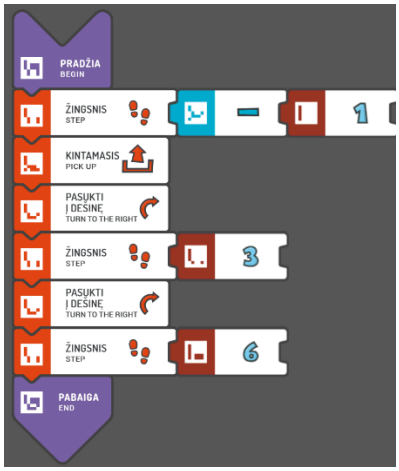
Užbaigę etapą, matome, kad 5 ir 7 užduotys nėra optimaliai išspręstos, kaip kad norėtų žaidimo kūrėjai. Kiekvieną išspręstą užduotį galima tobulinti.

Užduočių vaizdai ir jų algoritmai

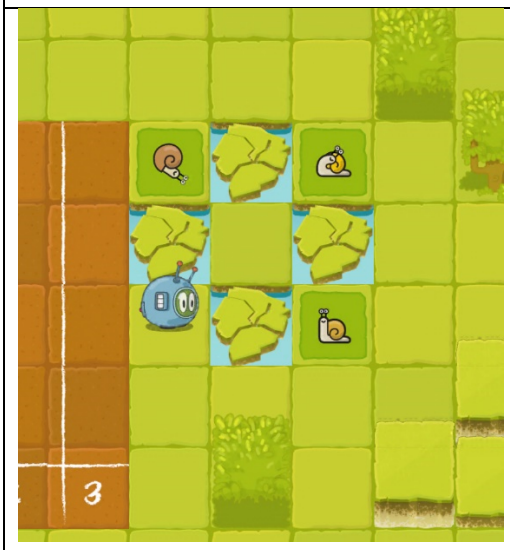
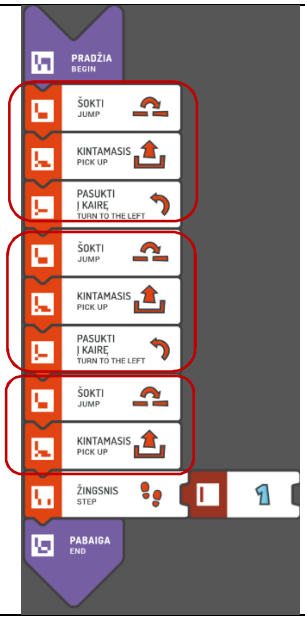
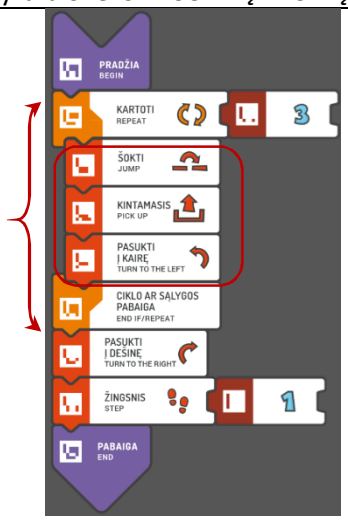
5 užduotis

8 užduotis

9 užduotis



7 užduotis

Šioje užduotyje reikia surinkti tris sraiges, šokinėjant per kliūtį.	Matome, kad tris kartus kartojami tie patys veiksmai, tik trečią kartą nebėra *PASUKTI Į KAIRĘ.*	Šią užduotį galima pakartoti po trečio etapo, kai mokiniai jau moka spręsti naudodami ciklą *KARTOTI (REPEAT)*. Tačiau jį atlikus, robotukui vėl reikia pasisukti atgal, todėl po ciklo yra blokelis *PASUKTI Į DEŠINĘ*.
		

Trečias etapas

Trečiajame etape mokiniai išmoksta naudotis žinomo kartojimų skaičiaus ciklu KARTOTI (REPEAT). Jie nagrinėja šio ciklo pateikimo skirtumus, lygindami su studio.code.org/hoc/6 . Mokiniai moka sudėlioti ciklo pradžios ir pabaigos blokelius, suskaičiuoti, kiek kartų bus atliekamas ciklas. Mokiniai mokosi sudėlioti vidinio ciklo veiksmus (komandas), atrinkdami pasikartojančius veiksmus.

Dabar keliaujama į Pietų Ameriką ir robotukui teks rinkti įvairias plunksnas. Atlikus visas aštuonias užduotis, robotukas gaus dar vieną trūkstamą erdvėlaivio detalę.

Šis etapas skirtas labai svarbiai programavimo struktūrai – žinomo kartojimo skaičiaus ciklui – įvaldyti. Ši struktūra apima du blokelius *KARTOK (REPEAT)* ir pabaigos blokelį *CIKLO AR SĄLYGOS PABAIGA /Jeigu/Kartok (END/IF/REPEAT)*.
Kol kas į *JEIGU (IF)* nekreipiame dėmesio. Su žinomo skaičiaus kartojimo ciklu mokiniai jau susidūrė atlikdami ir kurdami užduotis *Blue-Bot*. Mokiniai kas kartą vis geriau supranta, kad programuoti kartojimą galima įvairiais būdais, svarbu, kad naudojamos technologijos „suprastų“ ir „teisingai perskaitytų“ sukurtus ar sukonstruotus algoritmus.

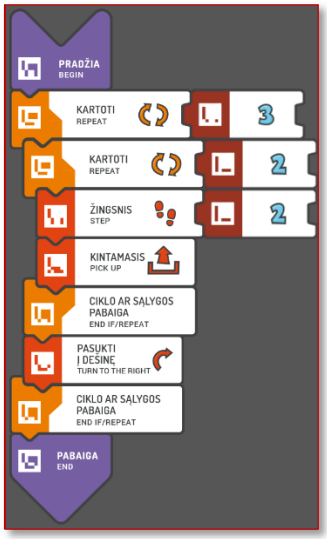
Failo pabaigoje yra pasikartojančių sekų pavyzdžių. Rekomenduotina išspręsti uždavinius su pasikartojančiais elementais, kad vaikai geriau suprastų, kur ciklas prasideda, kur baigiasi, kiek kartų kartojami veiksmai (elementai).

Pirmojoje užduotyje reikia du kartus kartoti tuos pačius veiksmus: nueiti du žingsnius ir pakelti plunksną. Primename, kad paveikslėlyje matome blokelį *PAIMK (PICK UP)* su **NETEISINGAI** išverstu pavadinimu.

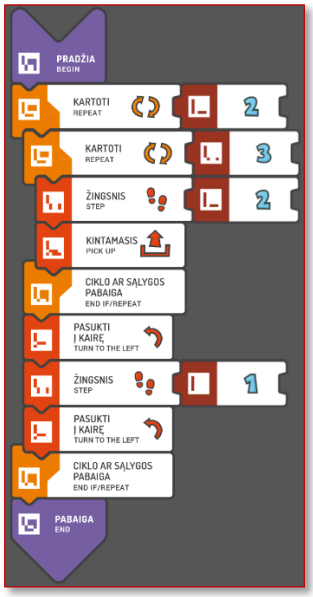
Trečioje užduotyje yra jau keturios plunksnos. Tačiau patogiau pirmiausia pasiimti pirmą plunksną, o jau po to tris kartus kartoti tuos pačius veiksmus.

6 užduotį pirmiausia reikia gerai išnagrinėti ir pamatyti tris pasikartojančių veiksmų serijas: žengti du žingsnius, paimti plunksną, pasisukti kairėn, vėl žengti du žingsnius ir, paėmus kitą plunksną, pasisukti dešinėn.

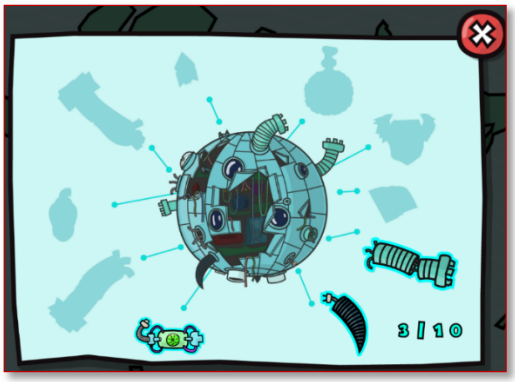
7 užduotis gana sudėtinga. Jei norime gauti optimalų rezultatą, tenka sukurti ciklą cikle. Nebūtinai visi mokiniai taip išspręs šią užduotį, nes galimi įvairūs sprendimo variantai.



Paskutinė trečiojo etapo užduotis reikalauja išsamios analizės. Kaip surinkus tris pirmąsias plunksnas robotukui elgtis, kad jis galėtų antrą kartą kartoti tuos pačius veiksmus, kai rinks ketvirtą, penktą ir šeštą plunksnas. (Robotukui, surinkus 3 plunksnas, reikia pasisukti kairėn, žengti du žingsnius ir dar kartą pasisukti kairėn).



Įveikęs tris etapus žaidėjas jau padėjo robotukui surinkti 3 detales erdvėlaivio taisymui.



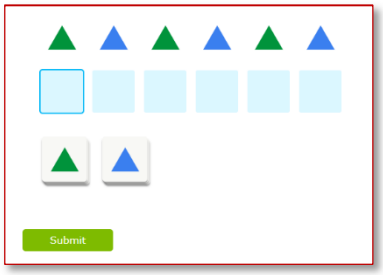
Priedas

Sekų nagrinėjimo pavyzdžiai

Rekomenduojama 1 klasei.

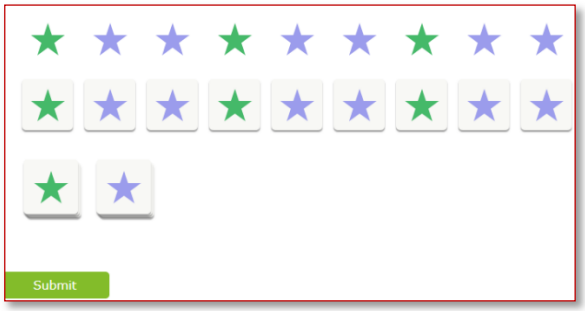
Sekų kopijavimas, kai sekos narių forma yra tokia pati.

Paprastas uždavinys, kai mokiniai kuria sekas pagal pavyzdį. Kartu nagrinėjama, kai tris kartus iš eilės kartojami du veiksmai (dedamas žalias ir mėlynas trikampiukai).

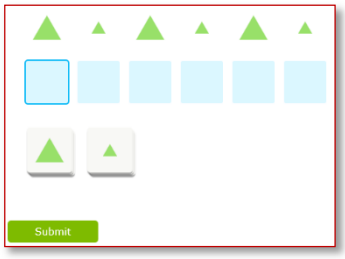


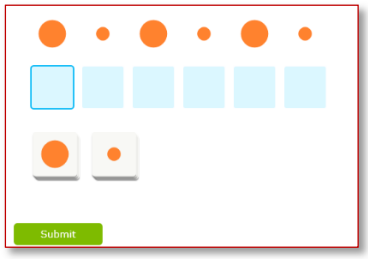
https://ca.ixl.com/math/grade-1/colour-patterns.

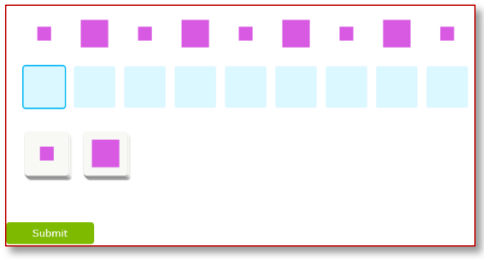
Atvėrus šią nuorodą, atliekamos ir kitos užduotys, kuriose pabrėžiamos skirtingos spalvos, o figūros lieka tokios pačios. Spalvų išdėstymas įvairus: keičiasi po vieną dvi vienodos spalvos figūras, toliau viena figūra kitos spalvos ir t. t.

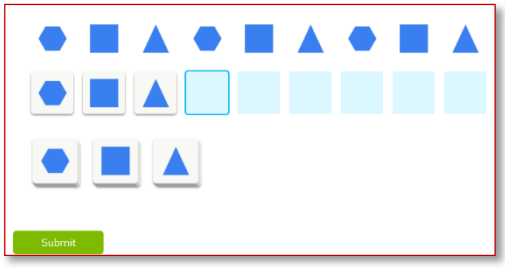


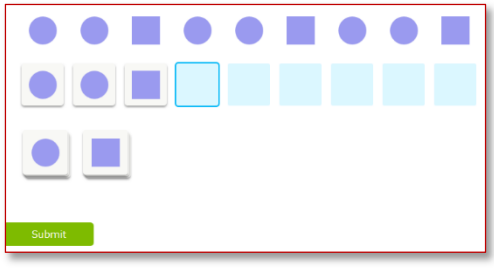
Sekų kopijavimas, kai sekos narių spalva vienoda, bet skiriasi forma.

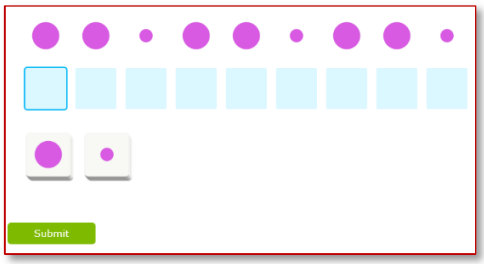






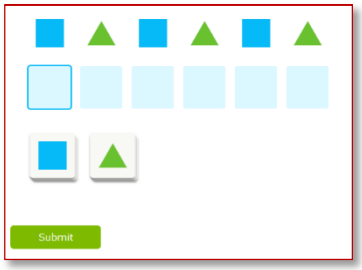


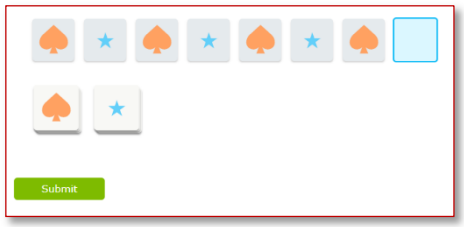


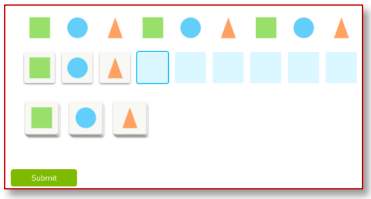


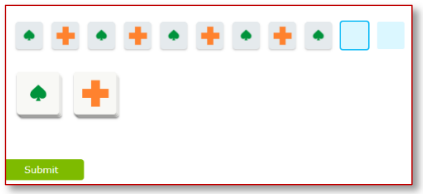
Trečia sekų grupė supažindina su jų savybėmis, kai skirtinga objektų forma ir spalva.

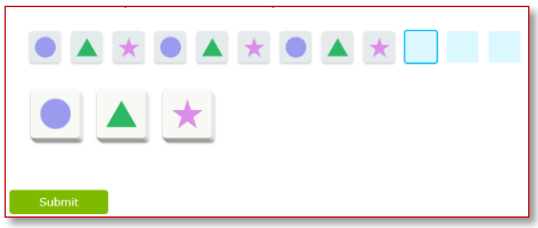
Mokiniai, parinkdami tinkamą objektų tvarką, suvokia sekos sudarymo principus.







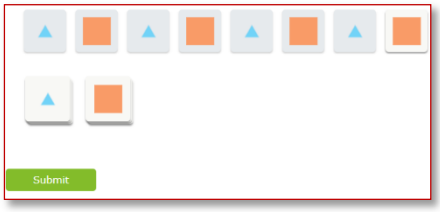


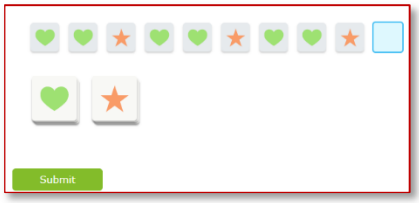


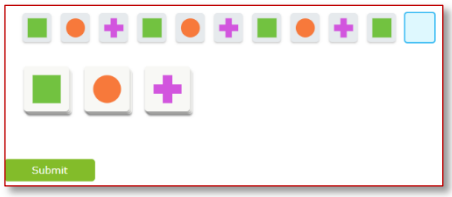
https://ca.ixl.com/math/grade-1/shape-patterns

Kita užduočių grupė – reikia užbaigti seką. Mokiniai taiko įgytas žinias ir randa tinkamą sekos narį.

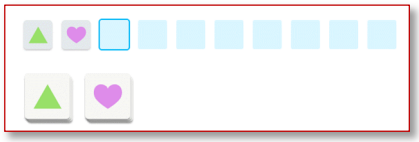
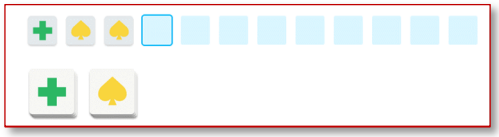
https://ca.ixl.com/math/grade-1/complete-a-pattern

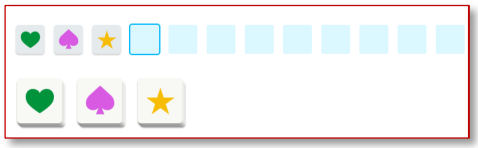
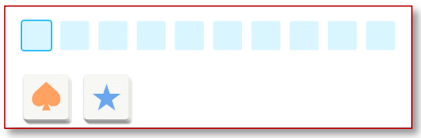
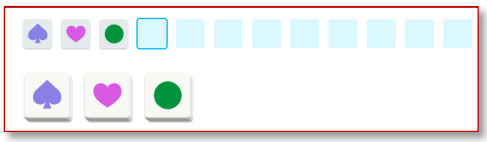




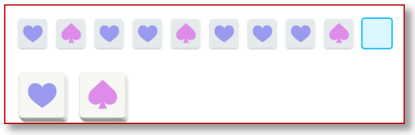
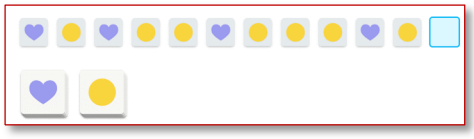


Sukurti seką.

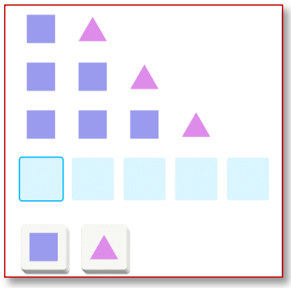



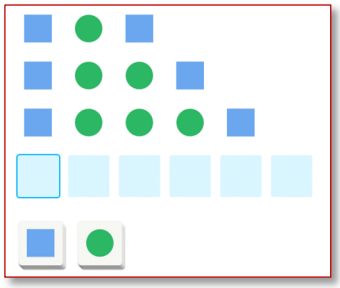




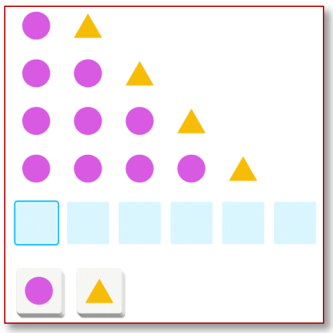
Didėjančioje sekoje rasti paskutinį narį.

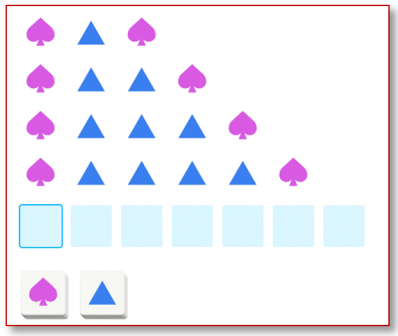



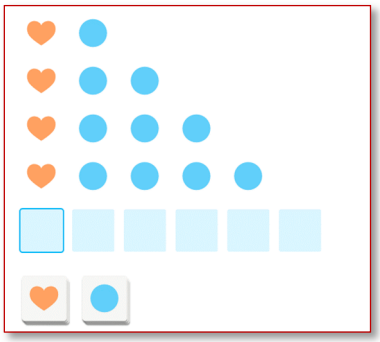
Paskutinė užduočių grupė, kai reikia sukurti tolesnę eilutę.

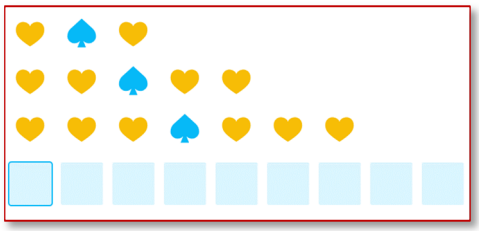


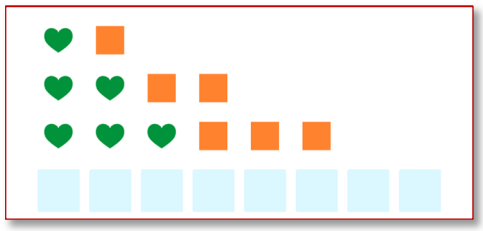




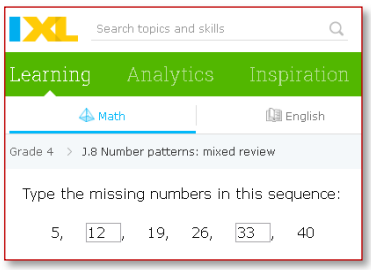


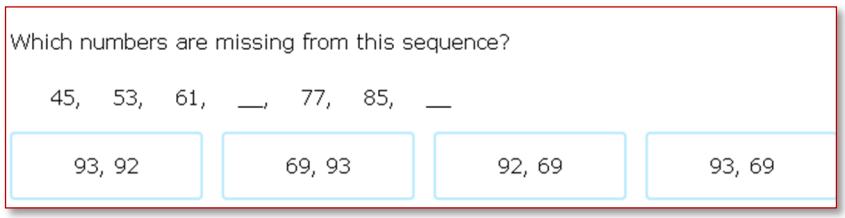




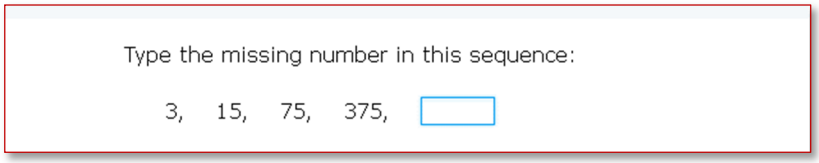


Visi sekų (*Patterns*) uždaviniai 1 klasei yra čia:
https://ca.ixl.com/math/grade-1
Sekų pavyzdžiai vyresniems mokiniams.
Reikia įrašyti praleistus skaičius jau dviejuose langeliuose.





Sudėtingesnė seka



Ketvirtasis etapas

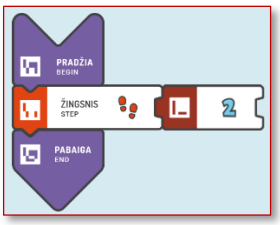
Ketvirtajame etape mokiniai išmoksta valdyti jau kelis robotus. Jie moka įjungti robotus bebrus (skiriasi jų kepuraičių spalva), nurodo programoje, kada pradeda atlikti veiksmus bebras, kada robotukas Skotis ir kada robotukas ir bebras juda kartu.

Robotukas keliauja į Šiaurės Ameriką1. Čia robotukas ras daug aukso luitų ir sėkmingai juos surinks.

Ketvirtajame etape atsiranda nauji veikėjai – bebrai. Sprendžiant užduotis, reikia labai atidžiai stebėti, kokia bebro kepurėlės spalva užduoties laukuose.

Šios kortelės programoje reiškia, kad į programos vykdymą įsitraukia nauji veikėjai (trijų spalvų kepurėles dėvintys bebrai), taip pat robotukas.

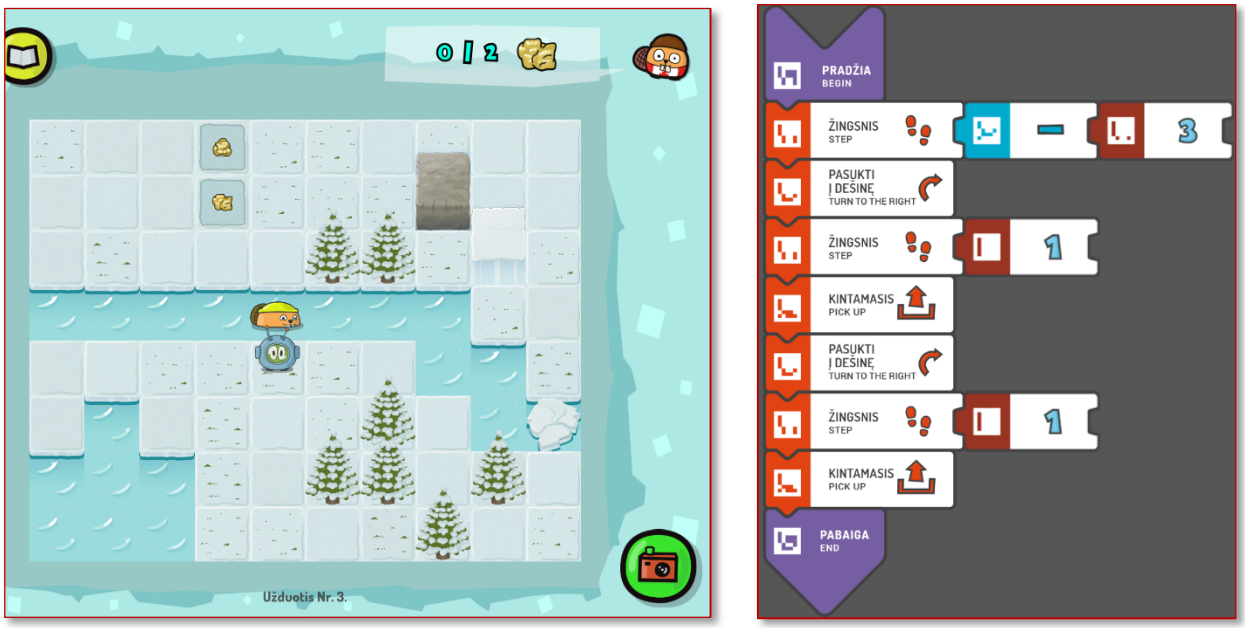
Vandenyje plaukiojantys bebrai gali padėti robotukui pereiti upelį. Tuomet bebrai yra tarsi tilteliai. Robotukas tiesiog peržengia langelį, kuriame yra bebras.



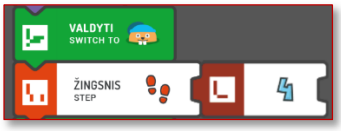
1 užduotyje bebras yra tiksliai tame langelyje, per kurį turi žengti robotukas, todėl ir programa labai paprasta, nes bebras atlieka tiltelio vaidmenį.

2 užduotis analogiška, tik reikia paimti du aukso luitus.

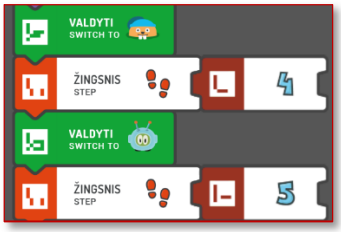
3 užduotyje robotukas dar turi pasisukioti.

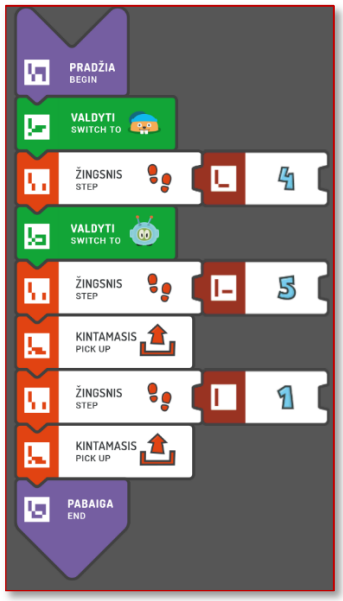


Sprendžiant 4 užduotį, susiduriama su nauja konstrukcija – reikia, kad judėtų ne tik robotukas, bet ir bebras arba net keli bebrai. Paveikslėlyje matome, kad bebras taps tilteliu tik tada, jeigu jis nuplauks keturis langelius. Kad bebras galėtų judėti, reikia padėti žalią blokelį *Bebras VALDYTI* (*SWITCH TO*), kuriame nupiešta bebro galvutė su mėlyna kepuraite, ir padėti kitą blokelį *ŽINGSNIS* (*STEP*) su keturiais žingsniais.



Kai jau bebras atsiduria tinkamame langelyje, reikia, kad pradėtų judėti robotukas. Tai labai svarbu įsidėmėti: reikia padėti blokelį *VALDYTI* (*SWITCH TO*) ir toliau jau kuriama programa, kurios žinomi visi elementai.

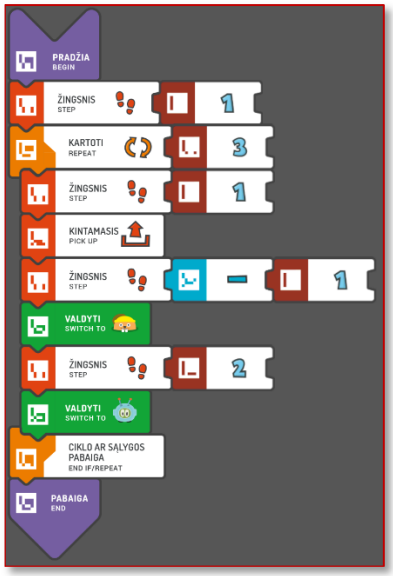




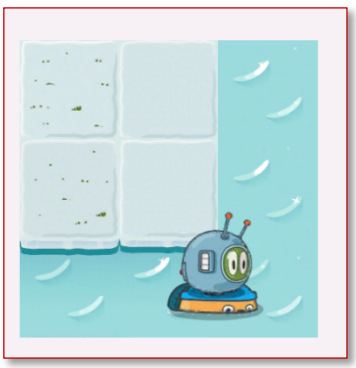
4 užduotis

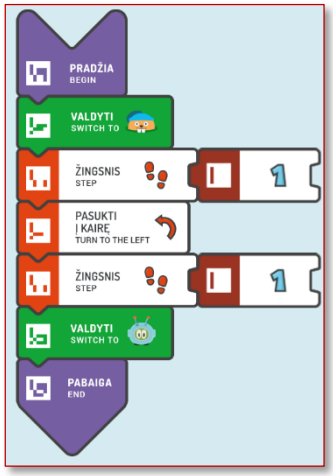
5 užduotis įtvirtina mokinių gebėjimą kontroliuoti bebro judėjimą kartu su robotuku. Kai pasiėmęs pirmąjį aukso luitą robotukas užlipa ant bebro, dedamas bebro su raudona kepuraite blokelis ir kitas blokelis, kuris liepia bebrui nunešti ant jo užlipusį robotuką per du langelius. Toliau jau programuojamas savarankiškas robotuko judėjimas, todėl dedamas blokelis *VALDYTI* (*SWITCH TO*) robotuką.

6 užduotyje paimami iš eilės trys aukso luitai, todėl prasminga panaudoti kartojimo ciklą tris kartus: nuo bebro robotukas žengia į aukso luito langelį, pasiimamas aukso luitas; grįžtama atbulomis ant „bebro tiltelio“, tada įjungiamas bebras, jo paprašoma nuplaukti du langelius ir vėl įjungiamas robotukas.

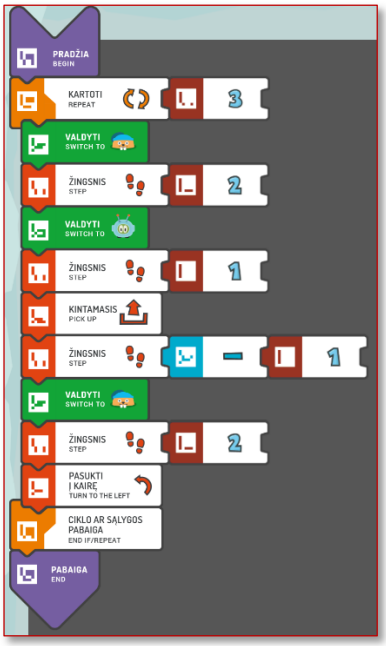


Bebras gali ne tik plaukti pirmyn ir atgal, bet taip pat gali daryti posūkius. Visus veiksmus jis gali atlikti ir vienas, ir tuomet, kai ant jo galvos tupi robotukas.

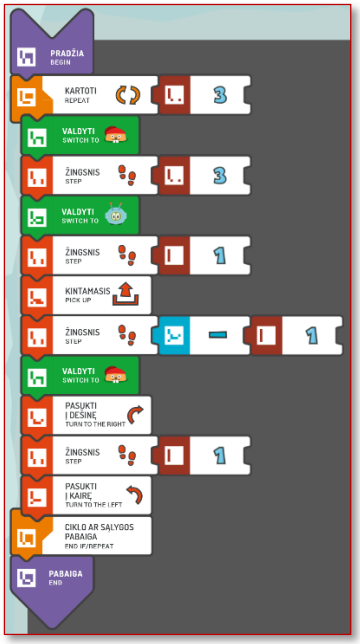




7 užduotyje robotukas turi pasiimti aukso luitus tris kartus. Kadangi užduoties pradžioje robotukas jau tupi ant bebro mėlyna kepuraite, jis atlieka tokius veiksmus tris kartus: plaukia su bebru du langelius, žengia į aukso luito langelį (vienas žingsnis), pasiima luitą, žengia vieną langelį atgal, kuriame yra bebras, tada plaukia kartu du langelius ir pasisuka kairėn. Programuojant reikia nepamiršti, kad pirmiausia rašant ciklo viduje esančius veiksmus, reikia įjungti bebrą, nes kitaip robotukas nušoks į vandenį, nepamiršti įjungti robotuko, kai jis žengia pasiimti aukso luitą, o paskui vėl nepamiršti įjungto bebro, kad jis toliau galėtų plukdyti robotuką prie aukso luitų.

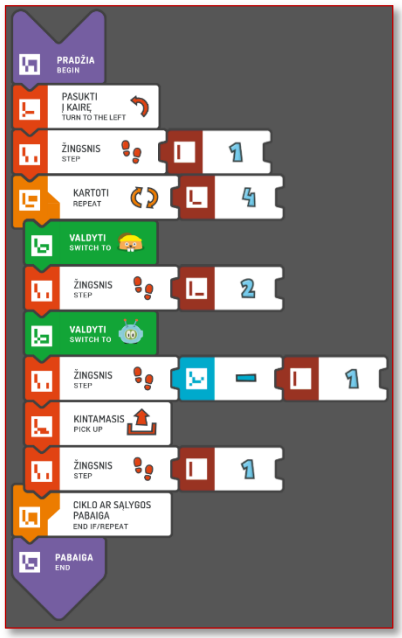


8 uždavinyje reikia panagrinėti vieno ciklo veiksmus. Jų seka būtų tokia: įjungiamas bebras raudona kepurėle, tada jis nuplaukia tris langelius, įjungiamas robotukas, jis žengia vieną žingsnį, tada paima aukso luitą ir atbulas grįžta ant bebro langelio; vėl įjungiamas bebras, jis pasisuka dešinėn, plaukia vieną langelį ir pasisuka kairėn. Šie veiksmai dar kartojami du kartus, tad iš viso gaunami trys ciklai.

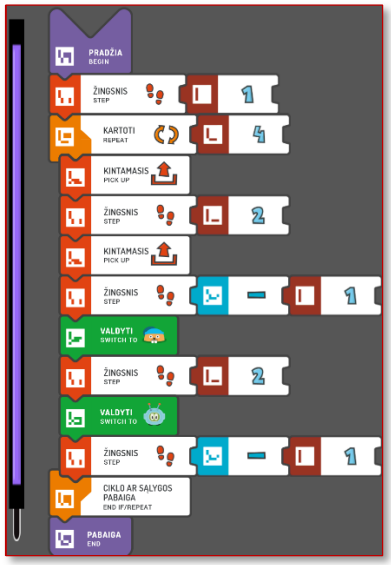


8 užduotis

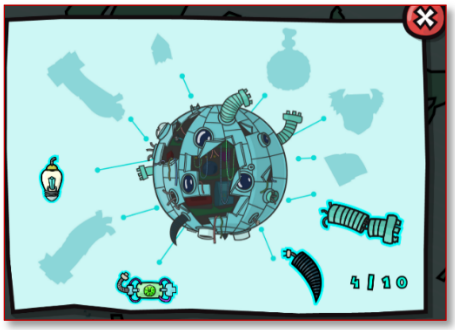
9 užduotyje robotukas pirmiausia pasisuka į kairę, nueina į langelį, kuriame yra bebras geltona kepurėle. Tada jau keturis kartus kartojami šie veiksmai: įjungiamas bebras geltona kepurėle, jis kartu su ant jo užlipusiu robotuku nuplaukia du langelius, tada įjungiamas robotukas, jis atbulas žengia į langelį, kuriame yra aukso luitas, jį pasiima, tada vėl žengia į langelį, kuriame yra bebras geltona kepurėle.



10 užduotis analogiška 9 užduočiai, tik joje aukso luitai išdėstyti abiejuose krantuose, todėl programa pailgėja, nes kiekvieno ciklo metu reikia paimti po du aukso luitus.



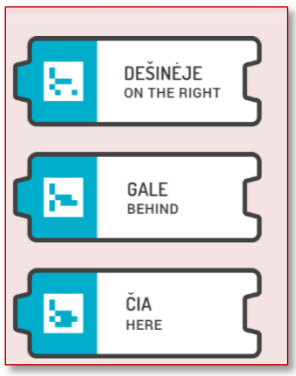
Įveikus keturis etapus, robotukas jau turi keturias reikiamas erdvėlaivio detales.



Penktasis etapas

Penktajame etape mokiniai išmoksta naujų veiksmų, kurie atliekami su įvairiais objektais. Atlikdami kai kurias užduotis, mokiniai išmoksta pasirinkti tinkamiausius objektus ir juos valdyti.
Robotukas keliauja į Antarktidą. Čia jis užsiims nauja veikla – užžiebs švyturius, esančius ledynuose. Jam padės ištikimi draugai bebrai.

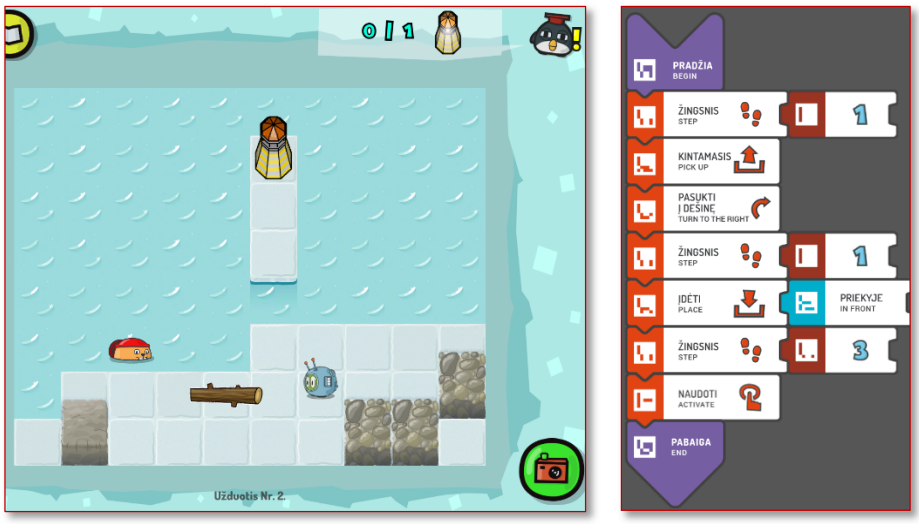
Robotukas naudos naujus blokelius: *ĮDĖTI* (*PLACE*), *PRIEKYJE* (*IN FRONT*), *KAIRĖJE* (*ON THE LEFT*), *DEŠINĖJE* (*ON THE RIGHT*), *GALE* (*BEHIND*), *ČIA* (*HERE*).



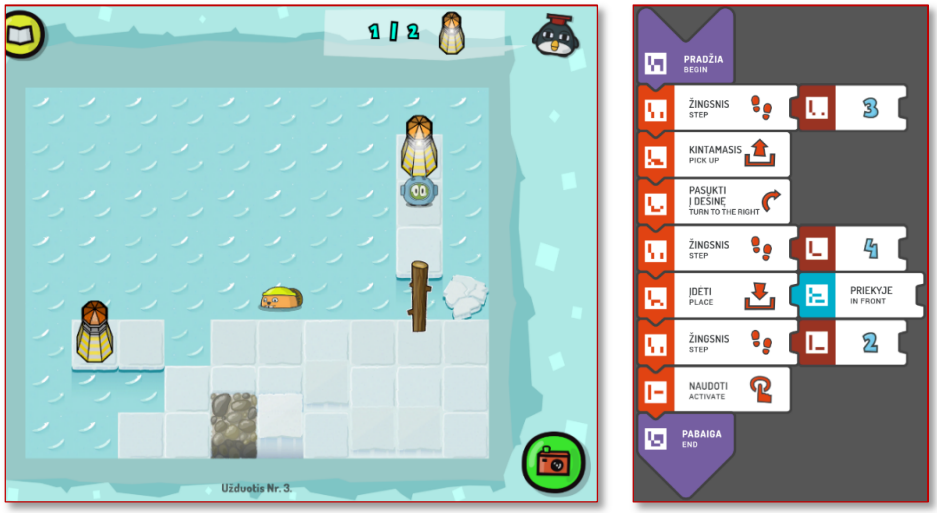
Robotukas užžiebs švyturį, naudodamas komandą *NAUDOTI* (*ACTIVATE*)

Šia komanda įjungiami arba išjungiami įrenginiai.
Pirmoje užduotyje robotukas atlieka paprastus veiksmus: žengia vieną žingsnį, pasisuka dešinėn, žengia keturis žingsnius (bebras naudojamas kaip tiltas) ir užžiebia švyturį. Kai reikėjo naudoti komandą *PAIMK*, reikėjo atstoti į langelį, kuriame buvo plunksna ar aukso luitas. Dabar, norint užžiebti švyturį, reikia stovėti gretimame langelyje.

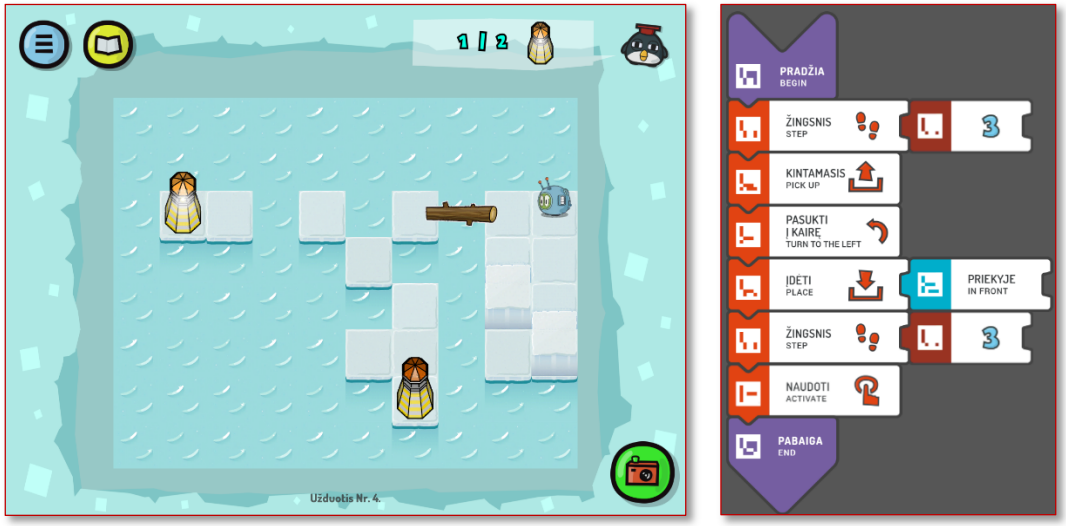
2 užduotyje robotukas gali pasinaudoti rąstu. Bet jam reikia rąstą pasiimti *(ŽINGSNIS 1, PICK UP)*, tada jį padėti priešais *(PASISUKTI Į DEŠINĘ, ŽINGSNIS 1, ĮDĖTI, PRIEKYJE)*, tik tada žengti tris žingsnius švyturio link ir jį užžiebti.



3 užduotyje vienas švyturys jau užžiebtas, todėl robotukas keliauja ant kito ledyno. Bet ten yra properša, todėl jis turi pasiimti rąstą. Rąstą jis pasiima, neatsižvelgdamas į jo poziciją (jam nereikia sukiotis).



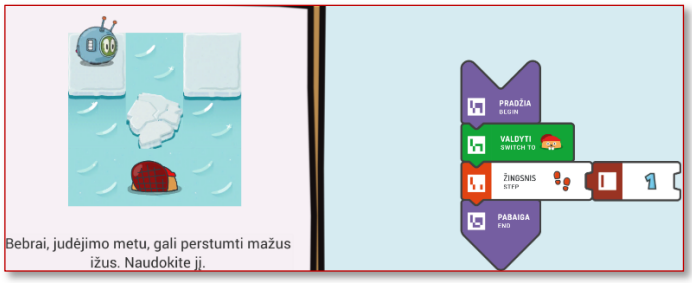
4 užduotis panaši į 3 užduotį.



5 užduotyje jau reikia du kartus dėti prieš save rąstą, kad pasiektum reikiamą švyturį. Todėl prasminga naudotis du kartus ciklu *KARTOTI (REPEAT)*. Ciklo veiksmai tokie: žengti du žingsnius, pasiimti rąstą, jį padėti priešais robotuką. Pakartojus du kartus, belieka žengti per rąstą ir dar vieną langelį (iš viso du) ir užžiebti švyturį.

6 užduotyje rąstų nebėra, bet robotukui gali padėti bebrai. Norint užžiebti visus keturis švyturius, pirmiausia reikia nurodyti bebrui geltona kepurėle nuplaukti du langelius, o bebrui mėlyna kepurėle – vieną langelį. Dabar jau robotukas įjungiamas ir keturis kartus jis kartoja tokius veiksmus: žengia du žingsnius ir užžiebia švyturį.

7 užduotyje atsiranda plaukiojančios ledo lytys, kurias ištikimieji bebrai gali stumdyti.
Šis pavyzdys rodo, kad nusisukęs nuo skaitytojo bebras raudona kepurėle gali pastumti ledo lytį per vieną langelį ir atsidurti langelyje, kuriame dabar yra ledo lytis.



Jei robotuko kelyje pasitaiko lyčių, jis gali jas peršokti – naudojama žinoma komanda *ŠOK (JUMP).* Pirmiausia reikia nurodyti bebrui pastumti ledo lytį. Kadangi bebras yra ne prie lyties, tai jam reikia dviejų žingsnių: vieno – kad priartėtų prie lyties, kito – kad lytis pasislinktų per vieną langelį. Aktyvuojamas robotukas, jis pasukamas dešinėn, norint pritaikyti dviejų kartojimų ciklą, nes ledo lytys yra dvi, tad dar reikia žengti vieną žingsnį. Tada ciklo veiksmai tokie: žingsnis ir peršokimas per lytį. Tuomet belieka tik užžiebti švyturį.

8 užduotis jau gerokai sudėtingesnė. Norint taikyti ciklą, reikia labai gerai išsiaiškinti, kaip turi būti išsidėstę bebrai ir ledo lytys. Norint pamatyti pasikartojimą, reikia susitvarkyti su bebrų judėjimu. Paveiksle parodytas vienas iš galimų algoritmų.

9 užduotyje bebras geltona kepurėle nereikalingas, nes tarp kiekvieno švyturio yra po dvi plaukiojančias lytis, tai paprasčiausia tiltelius pastatyti kaip mėlyną ir raudoną kepurėles dėvinčius bebrus.

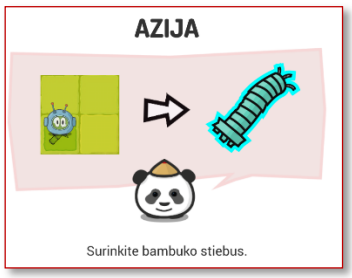
10 užduotyje svarbu pasukti bebrą mėlyna kepurėle dešinėn, kad jis galėtų nustumti ledo lytį ir reikiamą langelį.

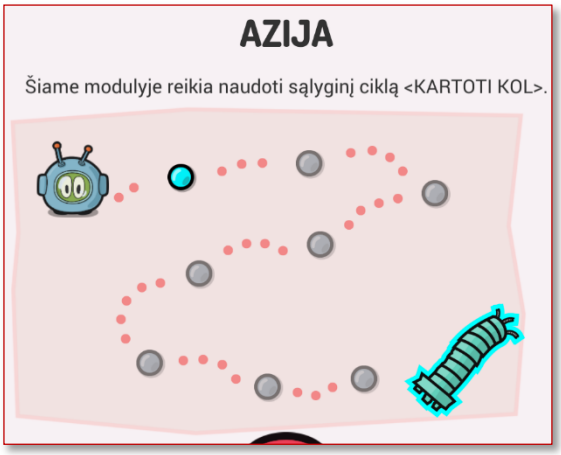
Užbaigus penktąjį etapą, robotukas jau turi penkias detales, reikalingas erdvėlaivio remontui.

Rekomenduojama mokiniams prisiminti užduotis. Mokytojas paskirsto mokinius po du ir jiems išrenka iš kiekvieno etapo po kelias užduotis. Mokiniai užduotis atlieka (sudėlioja blokelius į programą) ir jas nufotografuoja paprastai, tada nufotografuoja *Scottie Go*! programėle, o rastas klaidas ištaiso. Nufotografuoja ištaisytas blokelių programas ir jas palygina su pirmomis nuotraukomis. Grupelėmis aptaria užduotis.

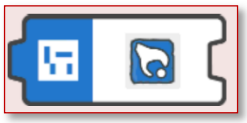
Šeštasis etapas

Šeštasis etapas skirtas nežinomo kartojimo ciklui KARTOTI KOL (WHILE). Tokio tipo ciklas programavime vadinamas sąlyginiu ciklu. Kai kurias užduotis galima atlikti ir su mokiniams jau žinomais veiksmais.
Robotukas keliauja į Aziją. Čia jis užsiims nauja veikla – rinks bambukų stiebus.

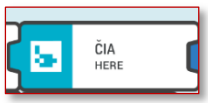




Robotukas naudos naujus blokelius: nežinomo kartojimo ciklo *KARTOTI KOL (WHILE)*, kliūties, robotuko paimamo objekto, tuščio langelio (*EMPTY*).

			
Nežinomo kartojimo ciklas *KARTOTI KOL (WHILE)*	*KLIŪTIS*	*PAIMAMAS OBJEKTAS*	*TUŠČIAS LANGELIS*

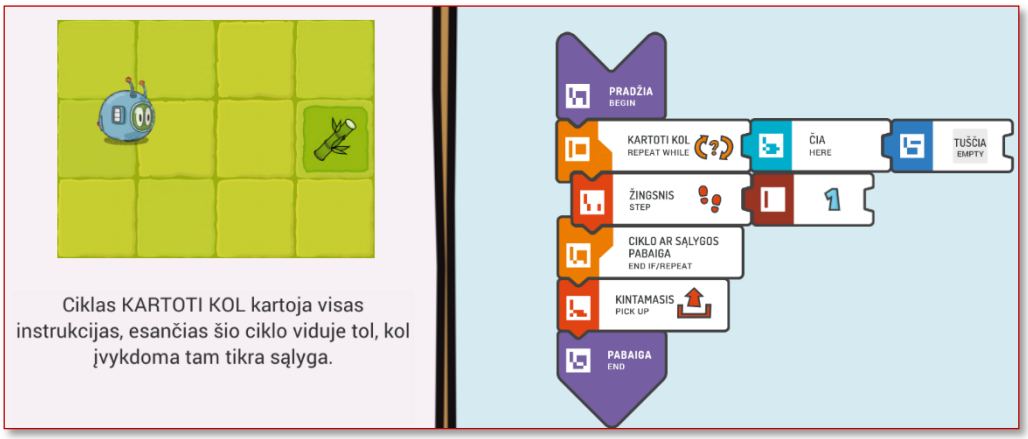
Primename mokiniams **svarbų** blokelių *PRIEKYJE* (*IN FRONT*) ir *ČIA* (*HERE*) skirtumą. Pirmasis blokelis

parodo, kad prieš robotuką esantis langelis tuščias, o antrasis 

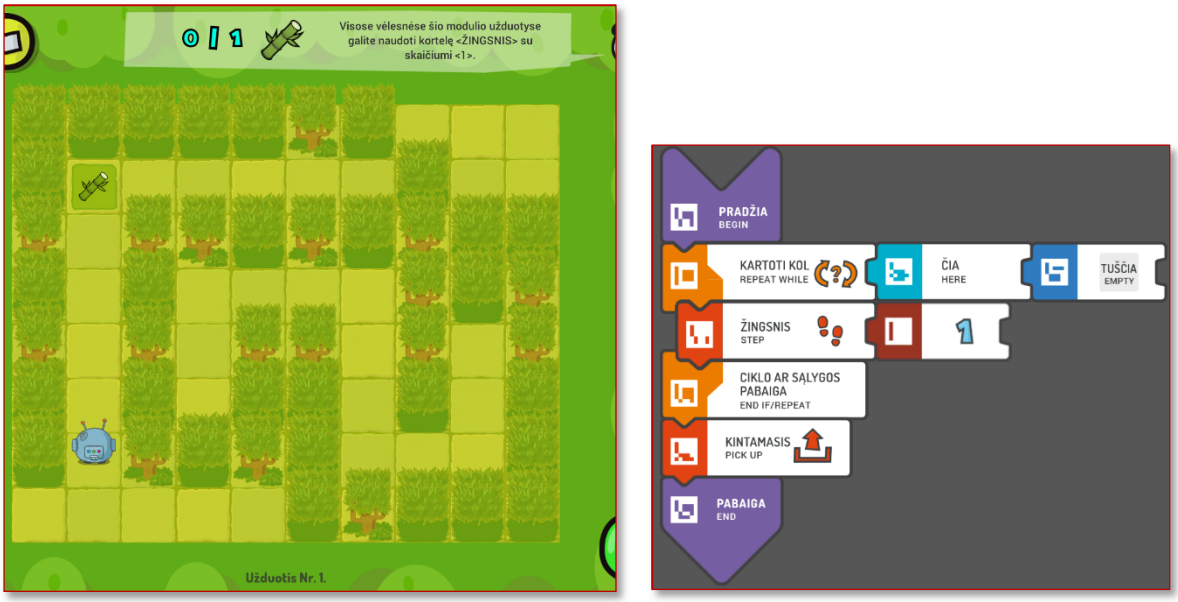
rodo langelį, kuriame robotukas šiuo metu yra.
Greta teks naudoti abu ciklus: jau žinomą KARTOTI (*REPEAT*) ir visiškai naują KARTOTI KOL (*REPEAT WHILE*). Abu ciklai baigiami tokiu pačiu CIKLO AR SĄLYGOS pabaigos blokeliu *(END IF/ REPEAT).*

Ciklas *KARTOTI KOL (REPEAT WHILE)* naudojamas tuomet, kai nežinoma, kiek kartų robotukui teks atlikti vieną ar kitą veiksmą. Iš pradžių užduotys lengvai sprendžiamos ir nenaudojant jokių ciklų, nes šiame žaidime nurodoma, kiek žingsnių turi žengti robotukas. Rekomenduotina nuo to ir pradėti spręsti užduotis. Tik jas atlikus žinomais būdais reikėtų bandyti aiškintis ciklo galimybes, nes 5-oje užduotyje jos jau tampa būtinos.

Paveikslėlyje vaizduojamas ciklas, kuris tikrina, ar langelis, kuriame stovi robotukas, yra tuščias. Jei taip, tada robotukas žengia pirmyn. Baigiantis ciklui, jis stovės tame langelyje, kuriame yra bambuko stiebas. Jei vietoje *ČIA (HERE)* būtų blokelis *PRIEKYJE (IN FRONT)*, tada robotukas po ciklo stovėtų langelyje prieš bambuko stiebą.



Klausimas. Kaip reikėtų papildyti šeštojo etapo 1 užduoties programą, jei vietoje blokelio *ČIA (HERE)* būtų blokelis *PRIEKYJE (IN FRONT)*?



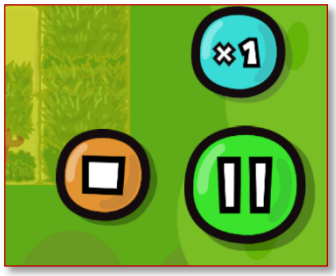
Atsakymas. Prieš blokelį *PICK UP* (*primename, kad blokelio pavadinimas išverstas **neteisingai***) reikėtų įterpti blokelį *ŽINGSNIS (STEP)* su žingsniu 1.

Klausimas. Kaip galima 1 užduočiai parašyti programą, kad nereikėtų naudotis jokiu ciklu?

Atsakymas. Užtektų dviejų blokelių eilučių: *ŽINGSNIS (STEP) 5* ir *PICK UP*.

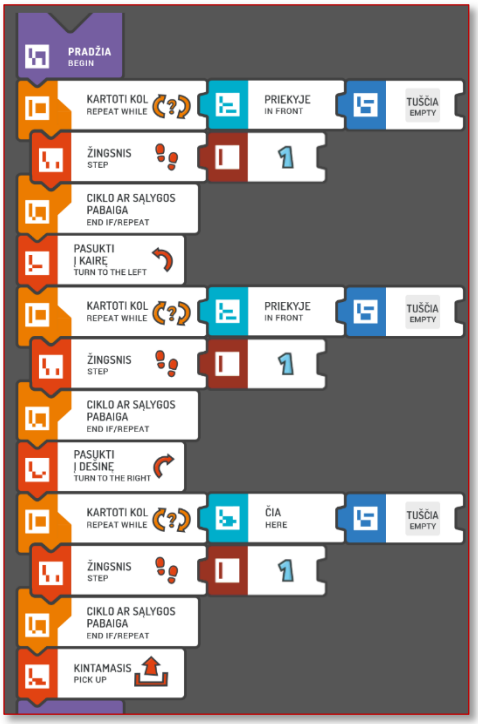
2 užduoties atlikimo pavyzdys.

Klausimas. Kaip pasikeistų programa, jei paveikslėlyje būtų kitas langelių skaičius. Pavyzdžiui, pirmiausia robotukas turėtų nužygiuoti 10 langelių, o po to – 12?
Atsakymas. Programa nepasikeistų. Čia ir yra ciklo *KARTOTI KOL (REPEAT WHILE)* privalumas.
Klausimas. Kam reikalingas atsirandantis mėlynas mygtukas?



Atsakymas. Spustelint mygtuką, galima padidinti 2 arba 3 kartus robotuko judėjimo greitį.

3 užduotis su programa, kuri sudėliota, atsižvelgiant į žaidimo reikalavimus.

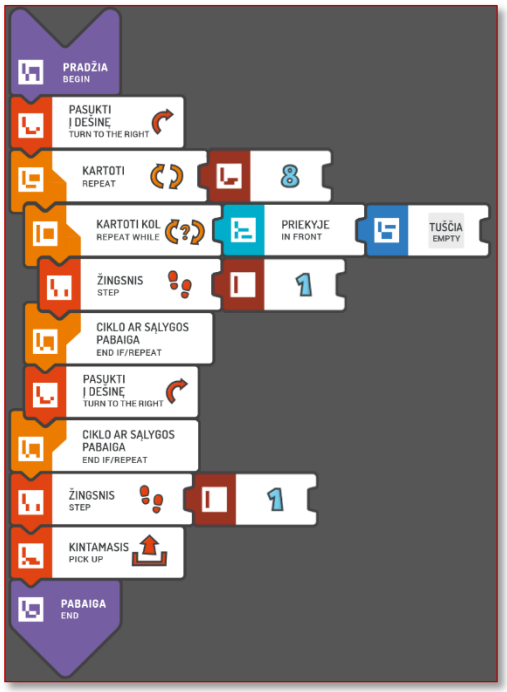


Sprendžiant 4 užduotį, rekomenduotina ją pirmiausia atlikti su paprastu *KARTOTI (REPEAT)* ciklu.

4 užduotis su programa, kuri sudėliota, atsižvelgiant į žaidimo reikalavimus.

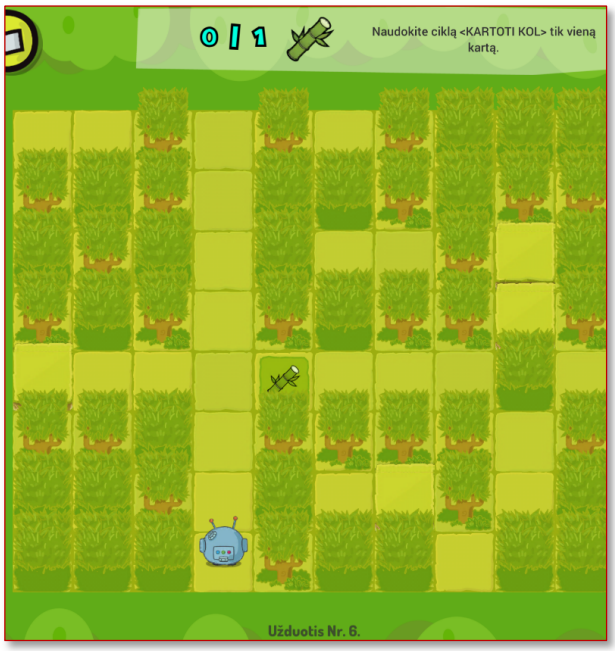
Klausimas. Kaip pasikeistų programa, jei robotukui tektų surinkti 6 ar 16 bambuko stiebų?
Atsakymas. Su ciklu *KARTOTI KOL (REPEAT WHILE)* parašyta programa nepasikeistų.

5 užduotis yra ypatinga. Jos sėkminga programa jau privalo naudoti ciklą *KARTOTI KOL (REPEAT WHILE)*, nes naudojantis kitomis priemonėmis gali nepakakti blokelių. Pateikiamas atsakymas. Mokiniams gali tekti padėti spręsti šią užduotį, nes kitos užduotys yra paprastesnės.

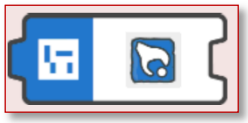


6 užduotis reikalauja naudoti vieną kartą ciklą *KARTOTI KOL (REPEAT WHILE)*. Žinoma, galima šią užduotį spręsti gerokai paprasčiau. Sprendžiant pagal reikalavimus, reikia prisiminti blokelį *KLIŪTIS*

ir blokelį, žinomą iš kitų etapų, *DEŠINĖJE (ON THE RIGHT)*. Programa kiekvieną kartą, atlikdama ciklą, tikrina, ar yra kliūtis (krūmas) dešinėje. Jei kliūties nėra (šiuo atveju langelyje bambuko stiebas), tai ciklas sustabdomas.

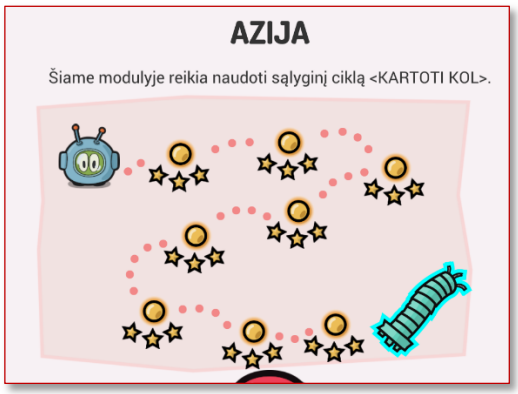


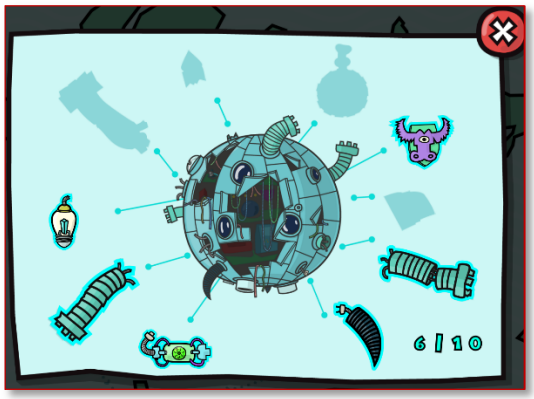
7 užduotis analogiška 6 užduočiai, tik reikia ciklą *KARTOTI KOL (REPEAT WHILE)* kartoti du kartus, nes reikia nukeliauti nevienodą langelių skaičių. Kadangi du kartai, tai ciklas įvelkamas į ciklą *KARTOTI (REPEAT)*. Žinoma, kad vaikai gali kurti ir kitokias, paprastesnes programas.



8 užduotis reikalauja naudoti blokelį *PAIMAMAS OBJEKTAS* . Tai labai panašu į 7 užduotį, tik čia robotukui nebereikia sukiotis.

Įveiktas šeštasis etapas.



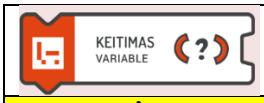


Rekomenduojama mokiniams prisiminti užduotis. Mokytojas paskirsto mokinius po du ir jiems išrenka iš kiekvieno etapo po kelias užduotis. Mokiniai užduotis atlieka (sudėlioja blokelius į programą) ir jas nufotografuoja paprastai, tada nufotografuoja *Scottie Go!* programėle, o rastas klaidas ištaiso. Nufotografuoja ištaisytas blokelių programas ir jas palygina su pirmomis nuotraukomis. Grupelėmis aptaria užduotis.

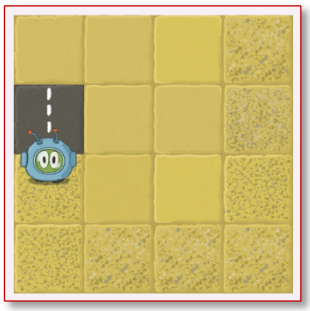
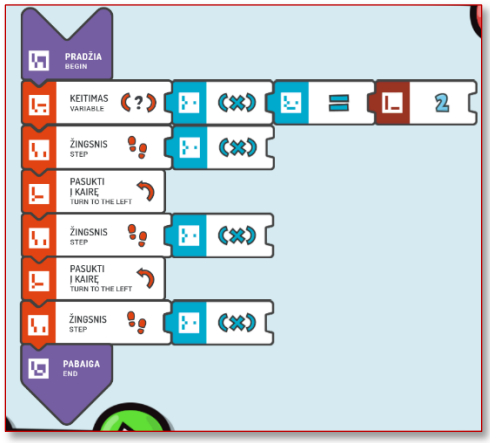
Septintasis etapas

Septintasis etapas skirtas kintamiesiems. Tai pradinukams sudėtinga tema, tačiau šio etapo užduotis galima atlikti ir su jau mokiniams žinomais veiksmais (nenaudojant kintamųjų).
Pastaba. Su kintamaisiais rekomenduojama dirbti tik aukštesniųjų gebėjimų 4 kl. mokiniams, kitiems reikėtų spręsti uždavinius tradiciniais būdais.
Robotukas keliauja į Australiją (AUSTRALIJA 1). Čia jis užsiims nauja veikla – tobulins kelius. Robotukas asfaltuos nepatogius kelius.

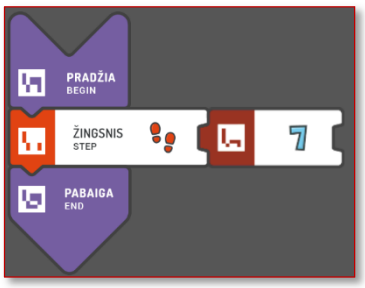
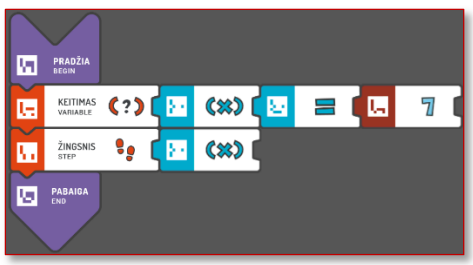
Nauji blokeliai aprašo kintamuosius. Kintamieji naudojami nusakyti programai robotuko žingsnių skaičių, ciklo kartojimo skaičių ir t. t.

	Kintamojo *X* arba *Y* įvardijimas ir reikšmės priskyrimo pradžios blokelis
DĖMESIO!	**Kortelės pavadinimas turi būti *KINTAMASIS***
	X – kintamojo vardas
	Y – kintamojo vardas
	LYGU – priskiria kintamajam reikšmę

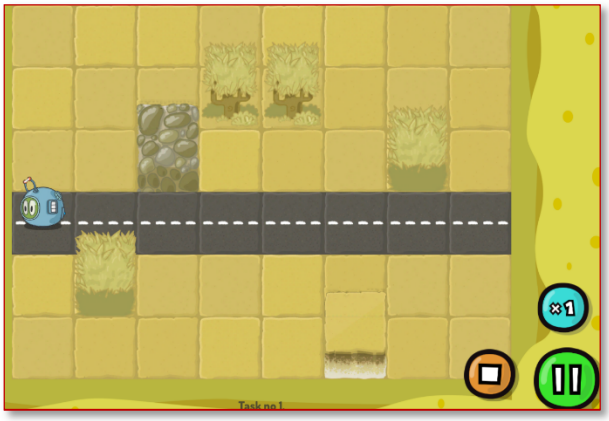
Abu paveikslėliai vaizduoja to paties uždavinio sprendimą: robotukas žengia tris žingsnius, pasisuka kairėn ir šiuos du veiksmus atlieka dar du kartus – iš viso tris kartus. Tik sprendžiant sudėtingesnius uždavinius išaiškėja kintamųjų privalumai. (Analogiškai ir ciklų naudojimo privalumai išryškėja tik sudėtingesnėse situacijose.)

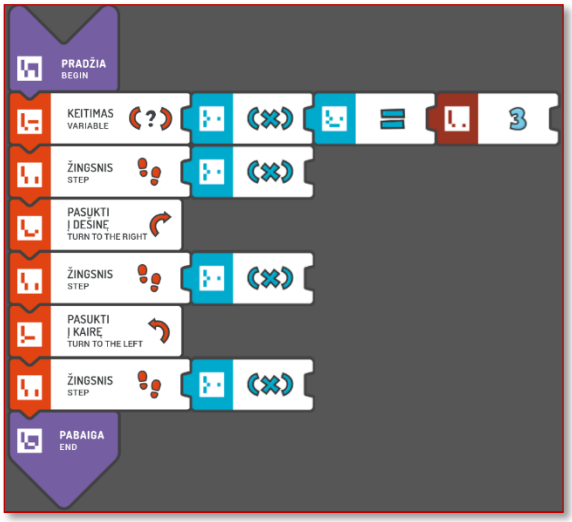
Spręsdami užduotis paprastesniais būdais, vaikai negaus trijų žvaigždučių, nes šie sprendiniai neatitinka uždavinio reikalavimų. Pavyzdžiui, 1 užduotis: robotukas tiesia asfaltuotus kelius vietoje nepatogių žvyrkelių. Vaikams liepiama sukurti programą, naudojant kintamuosius.

Šią užduotį galima atlikti pačiu paprasčiausiu būdu.	Atlikę kairiąją programą, vaikai gaus tik tokį užduoties įvertinimą.	Trys žvaigždutės užsidegs, jei programa bus atliekama, panaudojat kintamąjį *X*.
		

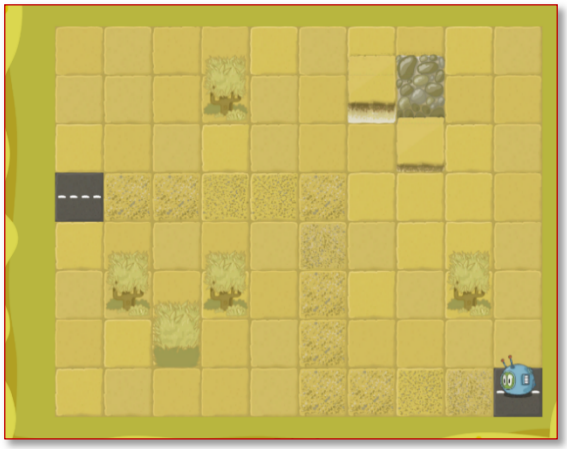
Atlikus programą, matomas išasfaltuotas kelias:

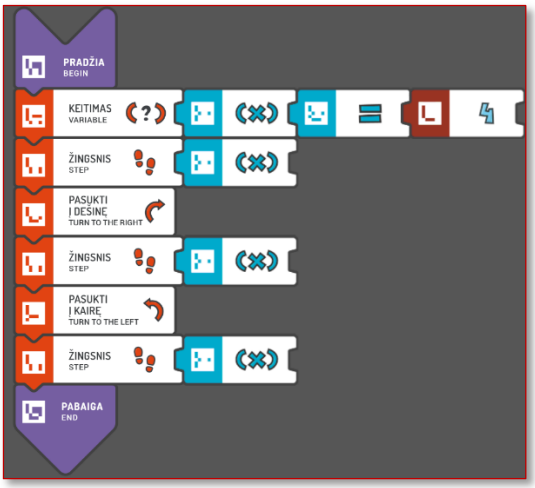


2 užduotis sutampa su nagrinėtu lentelėje pavyzdžiui, tik čia reikia pirmą kartą pasisukti į dešinę, o antrą kartą – į kairę.

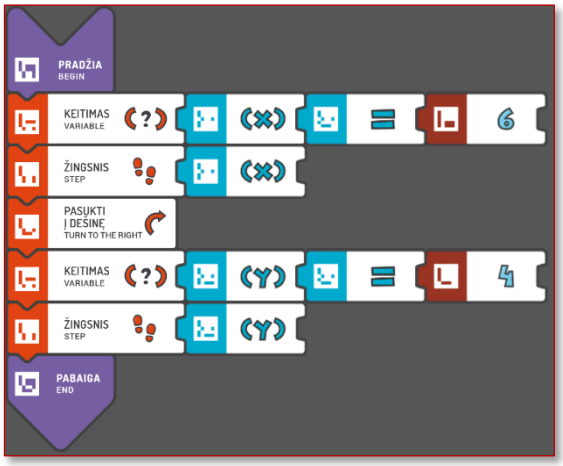


3 užduotyje reikia kiekvieną kartą žengti po keturis žingsnius vietoje trijų žengtų žingsnių 2 užduotyje.

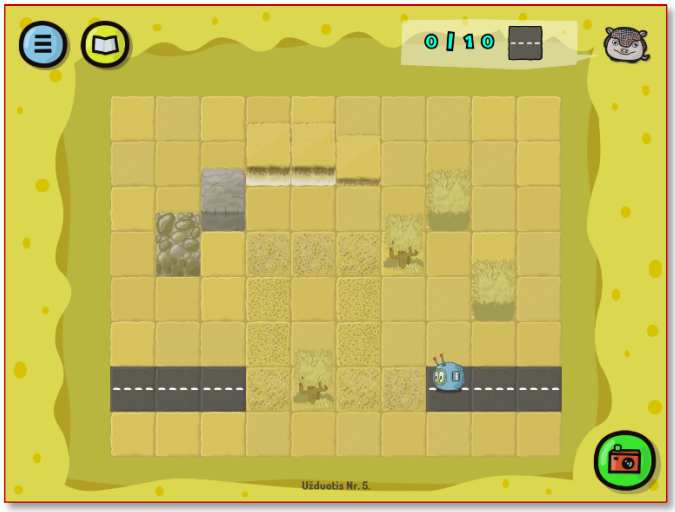




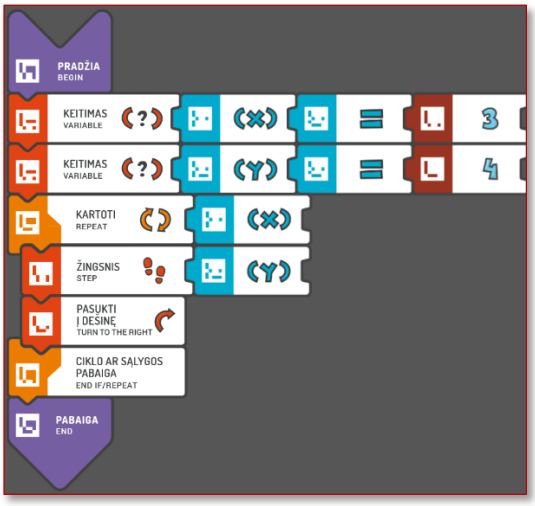
4 užduotyje prašoma atlikti veiksmus naudojant du kintamuosius – *X* ir *Y*. Robotukas žengia iš pradžių šešis žingsnius (kintamasis *X*), o pasisukęs į dešinę – keturis žingsnius (kintamasis *Y*).



5 užduotyje įtvirtinamas dviejų kintamųjų naudojimas.

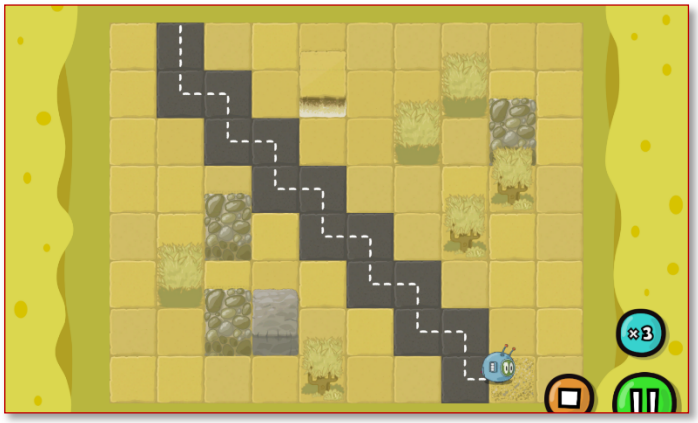


6 užduotyje kintamasis *X* naudojamas ciklo *KARTOTI (REPEAT)* kartojimo skaičiui aprašyti, o *Y* – robotuko žingsnių skaičiui.

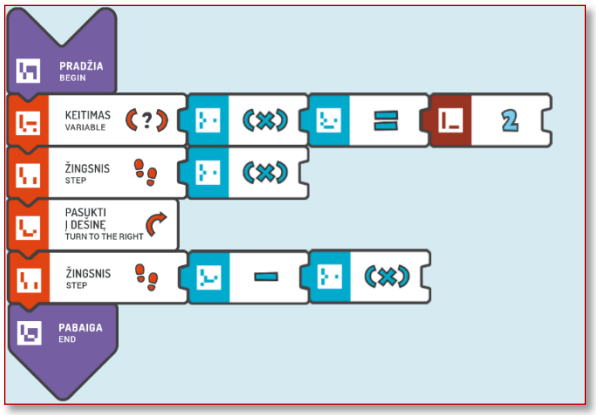


7 užduotis reikalauja mokinių susikaupimo. Reikia labai atidžiai išsiaiškinti, kaip juda robotukas, kokie jo veiksmai kartojasi.

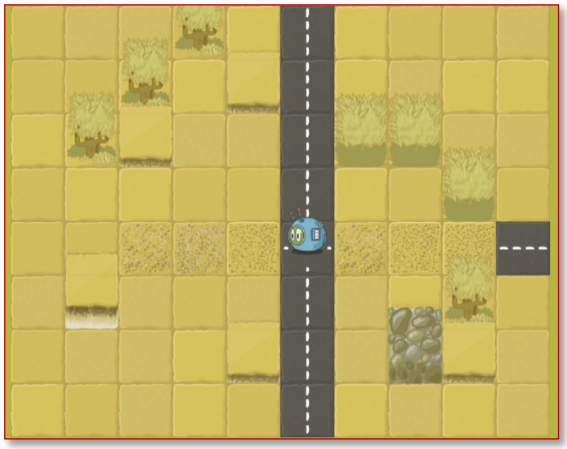
Atlikus užduotį, matome robotuko išasfaltuotą kelią

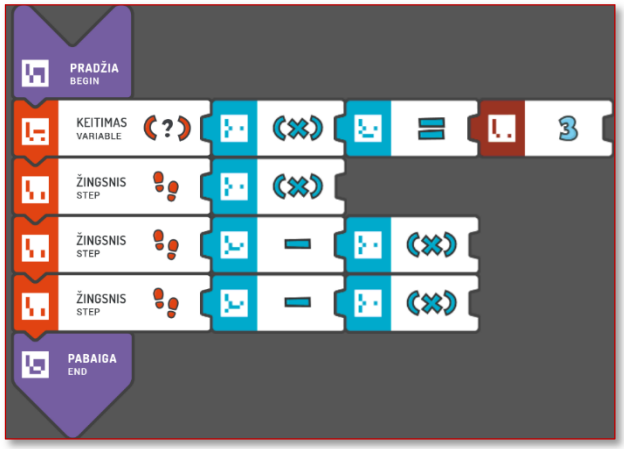


Robotukas žengs atbulas, jei prieš kintamojo vardą *(X)* padėsime minuso blokelį.



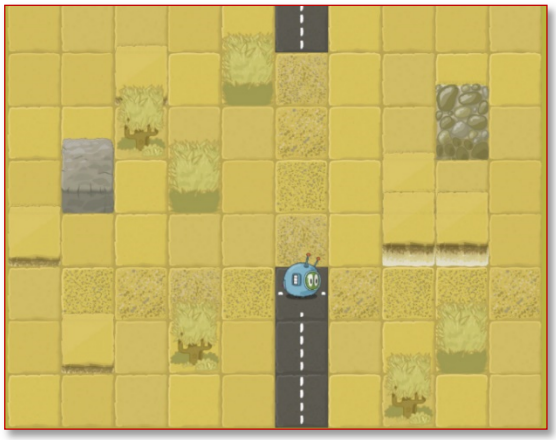
8 užduoties programos pavyzdyje matome, kad robotukas du kartus eina atbulas po tris žingsnius.

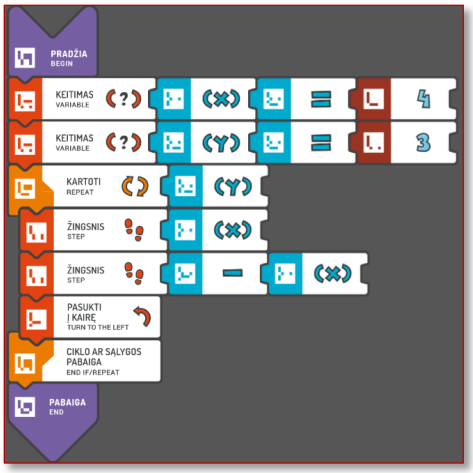




8 užduoties rezultatas:

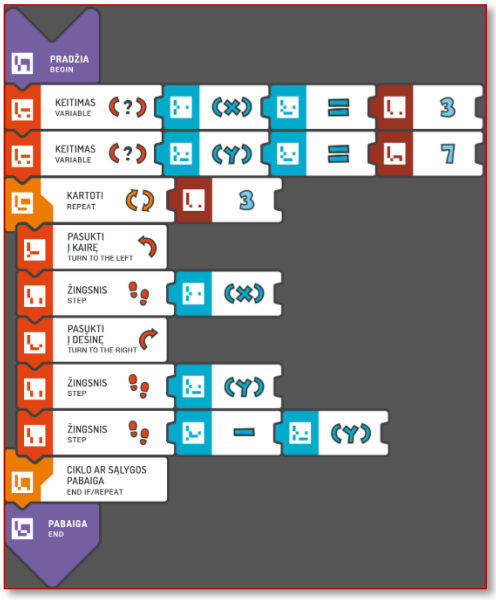
9 užduotis





9 užduoties rezultatas:

10 užduotis analogiška 9 užduočiai, tik truputį sudėtingesnis algoritmas. Kintamasis *X* naudojamas ir ciklo skaičiui aprašyti programoje, ir trims žingsniams žengti.



10 užduoties rezultatas:

Atlikus užduotis pagal reikalavimus, užsidega visos septintojo etapo žvaigždutės.

Rekomenduojama mokiniams prisiminti užduotis. Mokytojas paskirsto mokinius po du ir jiems išrenka iš kiekvieno etapo po kelias užduotis. Mokiniai užduotis atlieka (sudėlioja blokelius į programą) ir jas nufotografuoja paprastai, tada nufotografuoja *Scottie Go!* Programėle, o rastas klaidas ištaiso. Nufotografuoja ištaisytas blokelių programas ir jas palygina su pirmomis nuotraukomis. Grupelėmis aptaria užduotis.

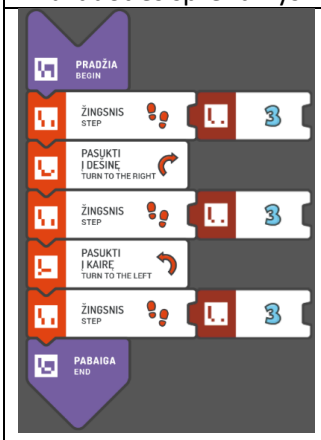
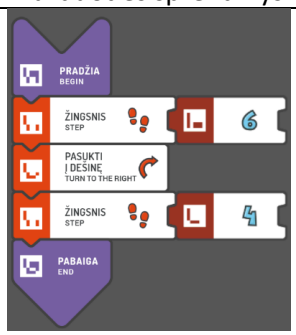
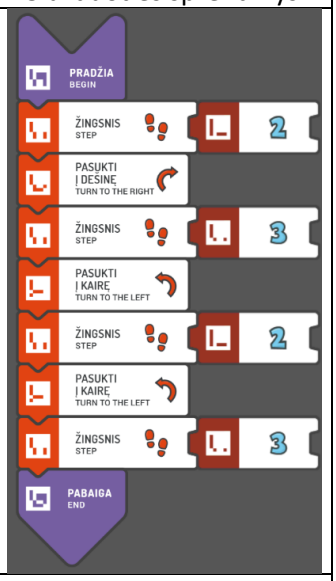
Priedas

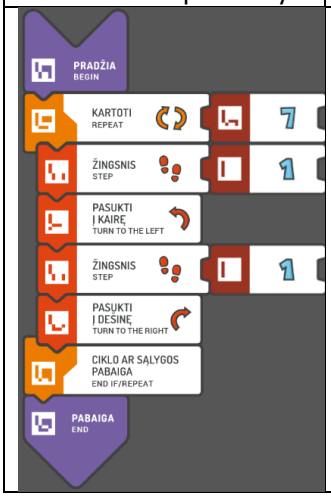
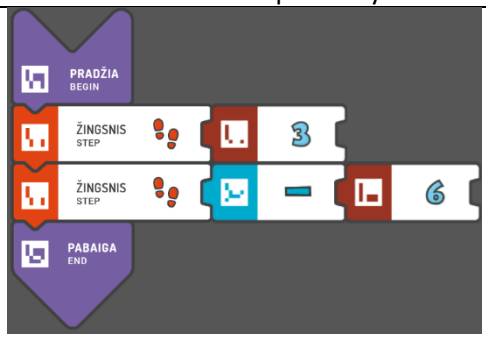
Septintojo etapo 9 ir 10 užduočių sprendiniai paprastuoju būdu – ciklas *KARTOTI (REPEAT)*.

9 užduotis, nenaudojant kintamųjų

10 užduotis, nenaudojant kintamųjų

Kitų septintojo etapo užduočių sprendiniai, nenaudojant kintamųjų

2 užduoties sprendinys	3 užduoties sprendinys	4 užduoties sprendinys	5 užduoties sprendinys
			

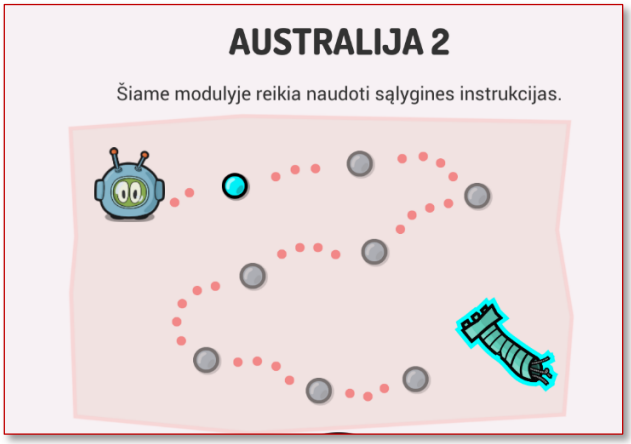
6 užduoties sprendinys	7 užduoties sprendinys	8 užduoties sprendinys
		

Aštuntasis etapas

Aštuntasis etapas skirtas šakojimo užduotims. Dažnai šakojimas vadinamas sąlyginiu sakiniu (jeigu /ar – IF). Tai sudėtinga pradinukams tema, todėl šį etapą gali įveikti tik labai gerai pasirengę mokiniai.

Programavime sąlyga – tai loginis reiškinys, kurio reikšmė lemia tam tikrų veiksmų atlikimą: jei reiškinio reikšmė yra *TIESA* – atliekami vieni veiksmai, jei *NETIESA* – kiti. Sąlyginis sakinys – tai programavimo kalbos valdymo konstrukcija, apibrėžianti sąlygą ir veiksmus, kurie turi būti atliekami, atsižvelgiant į tai, ar sąlyga tenkinama.

Robotukas tęsia kelionę Australijoje (AUSTRALIJA 2). Čia robotukas užsiims nauja veikla – nurodytose vietose sodins augalus. Kai kuriuos augalus robotukas turės su savimi, kai kuriuos turės pasiimti, kad juos pasodintų.



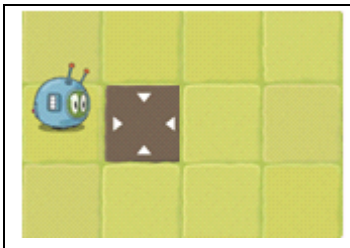
Naujas blokelis naudojamas žymėti žemės sklypeliams, kuriuose reikia sodinti augalus.

Sąlygos užrašymas pradedamas blokeliu *JEIGU (IF)*

.

Apie sąlygų naudojimą mokiniai turėtų būti supažindinti anksčiau. Jei ne, mokytojas skiria papildomą laiką ir nagrinėja įvarius atvejus. Pavyzdžiui, jei lyja, tai reikia pasiimti skėtį. Jei klausysiu mamos, tai gausiu saldainių.

Jei priešais langelyje robutokas mato paruoštą žemę, tai jis sodina augalą.

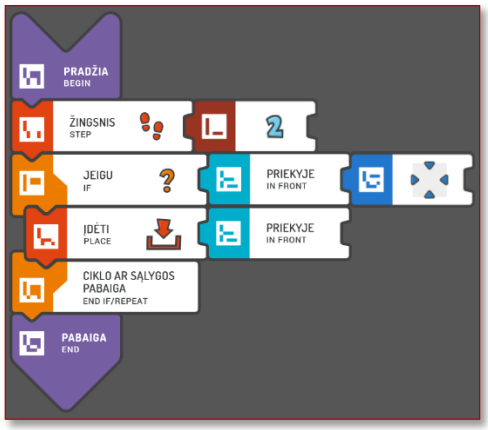


Sodina augalą

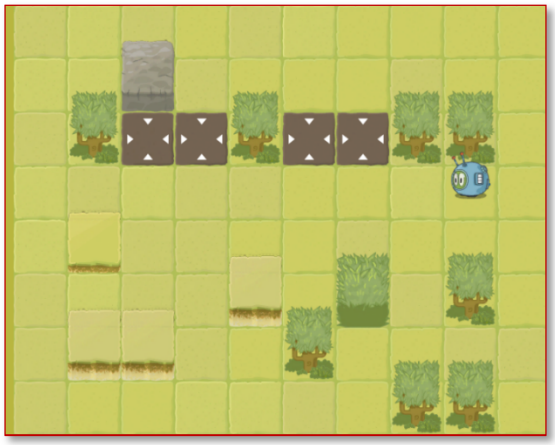
Ši konstrukcija baigiama blokeliu

, kuris naudojamas ir užbaigiant *KARTOTI KOL (WHILE)* ir *KARTOTI (REPEAT)* ciklus.

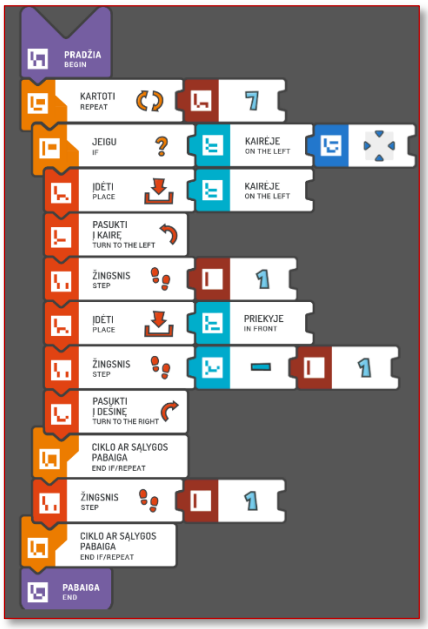
1 užduotis labai paprasta. Jis skirta įsidėmėti žemės sklypelį žymintį blokelį, suvokti, kad augalą robotukas nešasi, taip pat svarbu įtvirtinti sąlyginio sakinio blokelių išdėstymą. (Pabaigos blokelis reikalingas ir naudojant kartojimo ciklą *KARTOTI (REPEAT)*.)



2 užduotyje augalus tenka sodinti surastuose žemės sklypeliuose, kurie yra robotukui iš dešinės. Jis septynis kartus atlieka tokius veiksmus: patikrina, ar yra jam iš dešinės žemės sklypelis. Jei taip, tada šiame sklypelyje pasodina augalą. Jei ne, nieko nedaro ir toliau žengia vieną žingsnį. Tokius veiksmus jis kartoja septynis kartus.



3 užduotis sudėtingesnė. Robotukas tikrina septynis kartus, ar yra paruošta sodinimui žemė, jei taip, sodina augalą, pasisuka ir žengia prie antro šalia esančio žemės sklypelio ir vėl sodina. Pasodinęs grįžta atgal. Toliau vėl viskas kartojama. Jei žemė neparengta, robotukas žengia toliau.

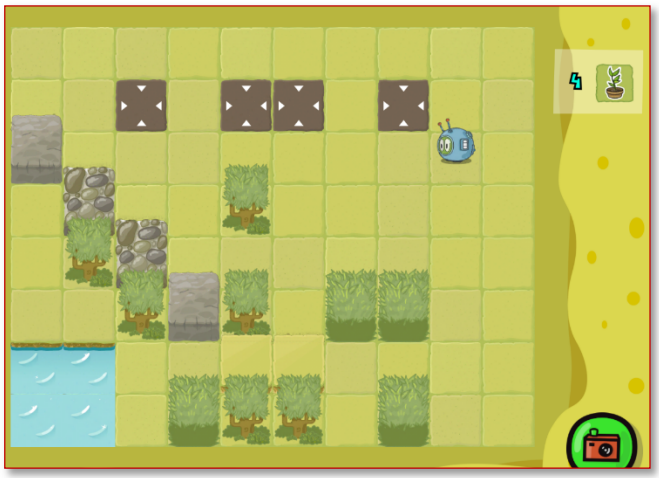


Atlikta 3 užduotis.

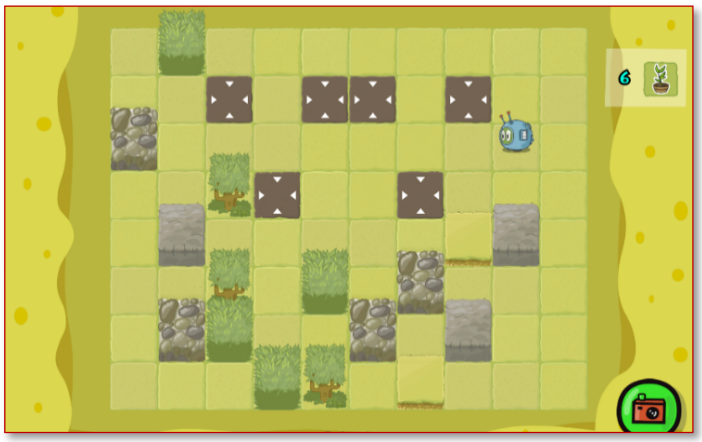
4 užduotis labai panaši, tik robotukas eina laužyta linija: žengia žingsnį, pasisuka kairėn, tada dar kartą žengia ir pasisuka dešinėn. Dėl tokios jo orientacijos reikia augalą sodinti iš kairės.

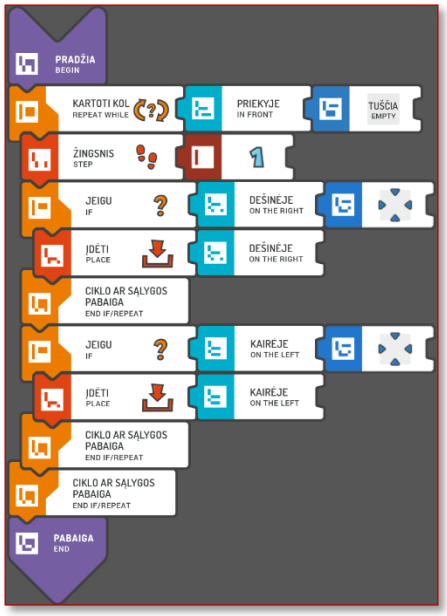
Atliktos 4 užduoties vaizdas:

5 užduotį prašoma atlikti naudojant ciklą *KARTOTI KOL (WHILE)*, nes robotuko kelionė baigiasi ties kliūtimi. Todėl jis keliaudamas kiekvieną kartą pasitikrina, ar artimiausias langelis yra laisvas (tuščias, t. y. nėra kliūties). Sąlyginis sakinys tikrina, ar robotukui iš dešinės nėra parengto žemės sklypelio. Jei taip, jis sodina augalą.

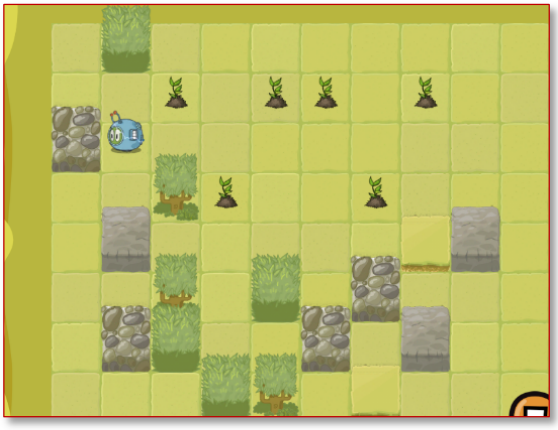


6 užduotyje ciklo *KARTOTI KOL* viduje prisideda antras sąlyginis sakinys, nes robotukas turi apsidairyti ir dešinėje pusėje, ar nėra paruošto žemės sklypo.

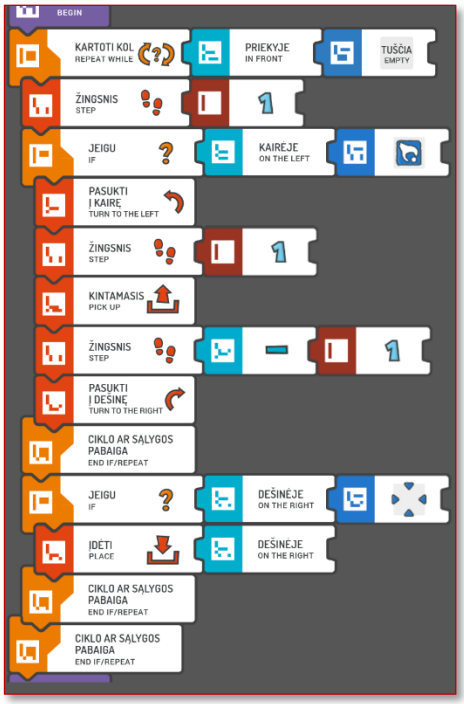




Atliktos 6 užduoties vaizdas:

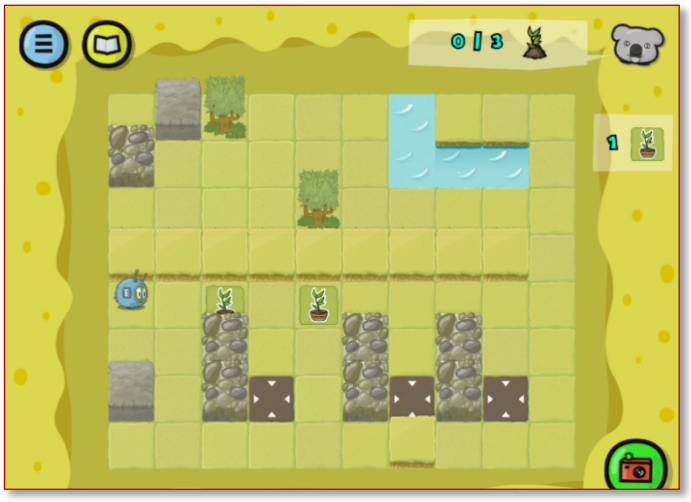
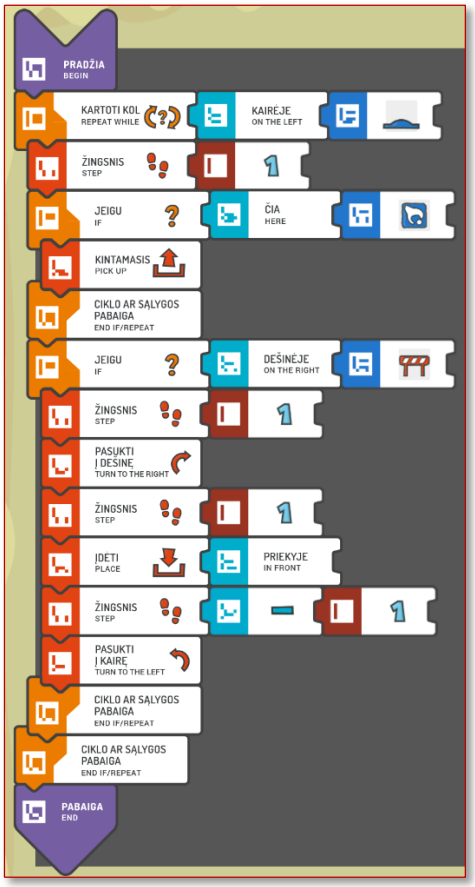


7 užduotyje robotukas neturi pasiėmęs augalų, todėl juos pasiima, kai randa. Norint, kad robotukas pasiimtų augalą, reikia pasisukti, įžengti į langelį, kuriame augalas yra, apsisukti ir vėl grįžti į pagrindinį taką.

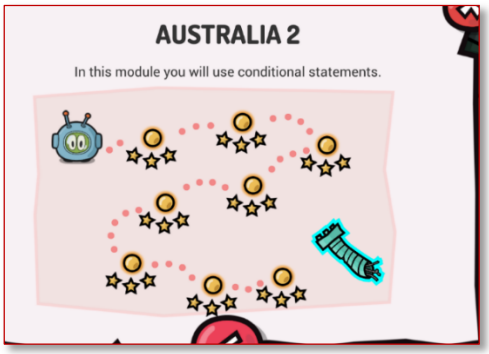


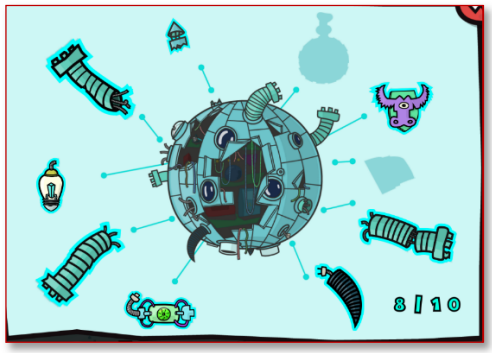
Atliktos 7 užduoties vaizdas:

Paskutinėje užduotyje robotukas turi tik vieną augalą, o kitus jam reikia pasiimti. Kiekvieną kartą, atliekant ciklo veiksmus, pirmiausia tikrinama, ar yra iš kairės kiūtis. Jei langelyje yra augalas, jis paimamas; jei dešinėje yra siena, žengiama į kitą langelį, pasisukama ir dar kitame langelyje sodinamas augalas. Pasodinęs augalą, robotukas grįžta į savo taką ir tęsia kelionę.

8 užduotis	8 užduoties programa
	
Atliktos 8 užduoties vaizdas	

Užduotys atliktos, tad robotukas gali keliauti toliau.
Robotukui betrūksta dviejų detalių, ir erdvėlaivis bus sutaisytas.





Rekomenduojama mokiniams prisiminti užduotis. Mokytojas paskirsto mokinius po du ir jiems išrenka iš kiekvieno etapo po kelias užduotis. Mokiniai užduotis atlieka (sudėlioja blokelius į programą) ir jas nufotografuoja paprastai, tada nufotografuoja *Scottie Go!* programėle, rastas klaidas ištaiso. Nufotografuoja ištaisytas blokelių programas ir jas palygina su pirmomis nuotraukomis. Grupelėmis aptaria užduotis.

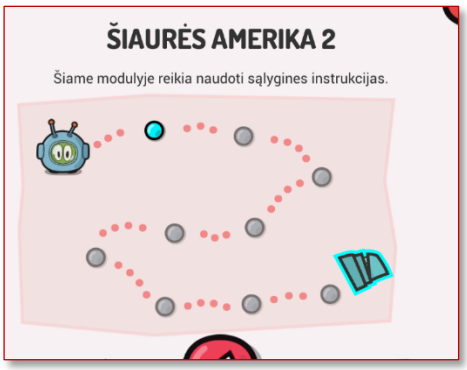
Devintasis etapas

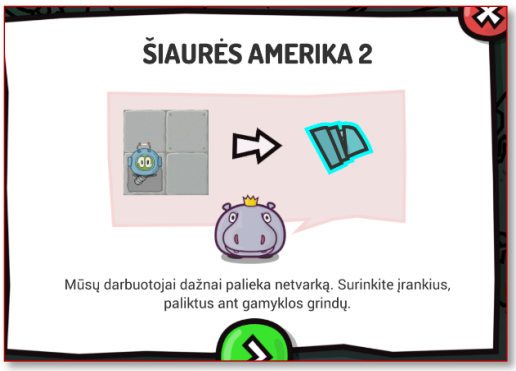
Devintasis etapas skirtas sudėtingesnėms šakojimo užduotims. Dažnai šakojimas vadinamas sąlyginiu sakiniu (jeigu /ar – IF). Tai YPAČ sudėtinga pradinukams tema, todėl šį etapą gali įveikti tik labai gerai pasirengę mokiniai.

Programavime sąlyga – tai loginis reiškinys, kurio reikšmė lemia tam tikrų veiksmų atlikimą: jei reiškinio reikšmė yra *TIESA* – atliekami vieni veiksmai, jei *NETIESA* – kiti. Sąlyginis sakinys – tai programavimo kalbos valdymo konstrukcija, apibrėžianti sąlygą ir veiksmus, kurie turi būti atliekami, atsižvelgiant į tai, ar sąlyga tenkinama.

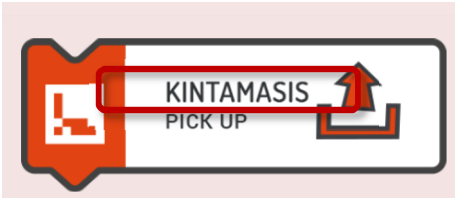
Robotukas tęsia kelionę Šiaurės Amerikoje 2.

Čia robotukas užsiims nauja veikla – rinks įrankius fabrikuose.

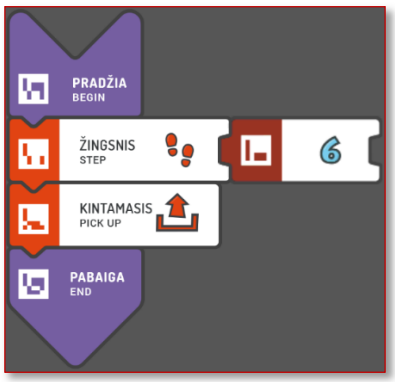




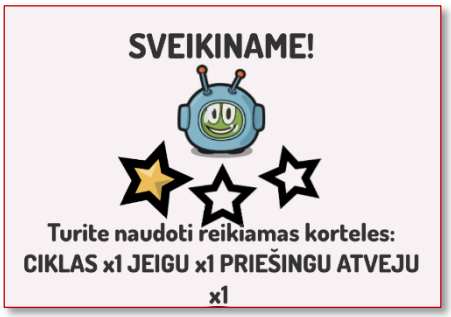
Primename vertimo klaidą, nes kiekvienoje šio etapo užduotyje reikės naudoti blokelį *PAIMK*, nors ant jo lietuviškai parašyta *KINTAMASIS*.



1 užduotis atrodo labai paprasta. Robotukas pereina šešis langelius ir šeštajame pasiima detalę.



Tačiau taip išsprendę užduotį gausime tik vieną žvaigždutę. Bus paaiškinta, kad, norint teisingai ją atlikti, reikia naudoti sąlygos blokelius.



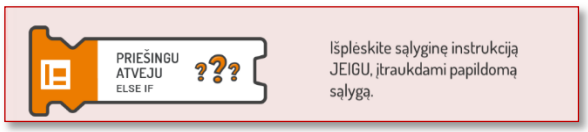
Atlikdami aštuntojo etapo užduotis, išsamiai susipažinome su sąlygos (šakojimo veiksmu) *Jeigu (IF)*

ir atlikdavome atitinkamus veiksmus, jei ta sąlyga būdavo patenkinama. Pavyzdžiui, pasakome – jei rytoj švies Saulė, tai eisiu maudytis. Tačiau šiuo atveju nutylime, ką darysime, jei rytoj lis, ar bus šalta ir vėjuota.

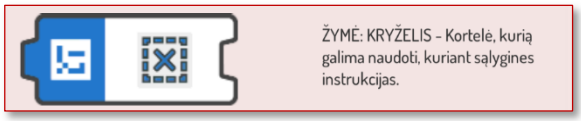
Devintajame etape jau turime naudotis sudėtingesne konstrukcija – ką darysiu, jei sąlyga nebus patenkinta. Pavyzdžiui, jei rytoj švies Saulė, tai eisiu maudytis, priešingu atveju (blokelis *PRIEŠINGU ATVEJU (ELSE)*) skaitysiu knygą.

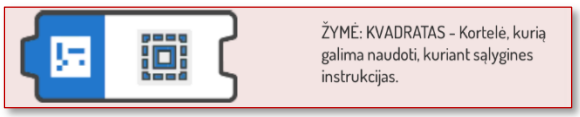
Pati sudėtingiausia konstrukcija, kai po elemento *PRIEŠINGU ATVEJU* yra klausiama, ar išpildomos naujos sąlygos. Pavyzdžiui, jei rytoj švies Saulė, tai eisiu maudytis, priešingu atveju, jei draugas ateis (blokelis *PRIEŠINGU ATVEJU (ELSE IF)*), žaisime šachmatais, o jei draugas neateis, skaitysiu knygą.

Sąlyginio sakinio konstrukcijos:



Devintojo etapo užduotyse bus naudojami du nauji blokeliai:





1 užduoties tikslas – padirbėti su sąlyginiu sakiniu, kai reikia kažką padaryti, kai sąlyga patenkinta, ir taip pat kažką padaryti, kai sąlyga nepatenkinta (blokelis *PRIEŠINGU ATVEJU (ELSE)*).

....

1 užduotyje robotuko kelias turi būti aprašomas tokiais principais: robotukas turi nukeliauti šešis langelius, tačiau kiekvieną kartą jis turi tikrinti, ar prieš jį esantis langelis yra tuščias. Jeigu jis tuščias

, robotukas žengia į naują langelį ir daugiau nieko nedaro. Tačiau priešingu atveju

(langelis nėra tuščias (jame yra veržlė), robotukas žengia į langelį ir paima veržlę. Kad langelis nėra tuščias, aišku iš užduoties, nes gali būti tik dvi langelio būsenos: langelis tuščias arba ne.

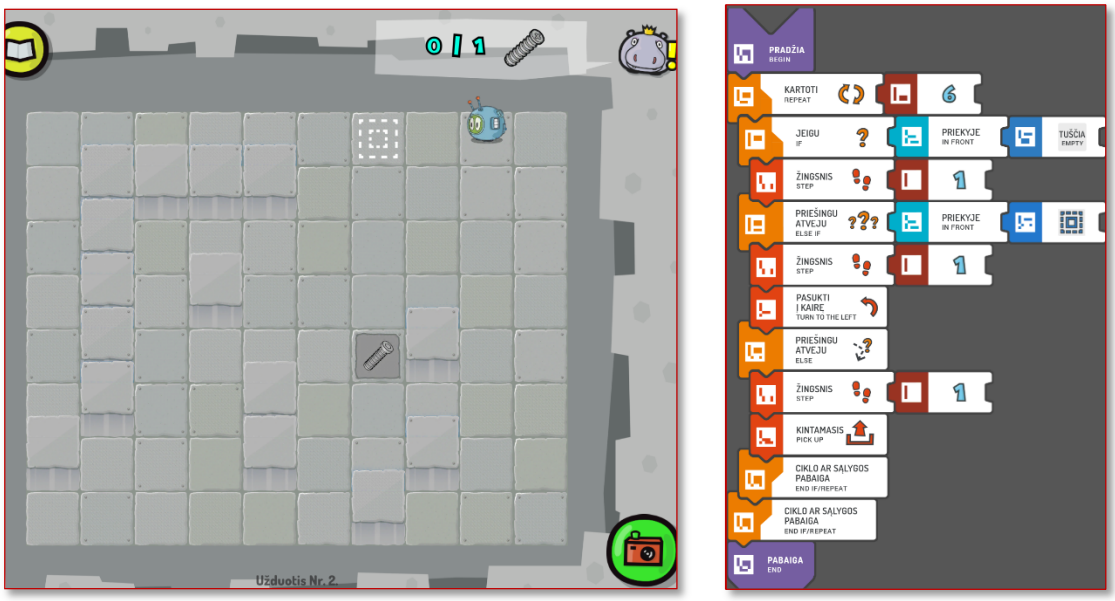
2 užduotyje yra langelis langelis su dviem baltų punktyrų kvadratukais. Tai reiškia, kad jau neužtenka patikrinti, ar langelis tuščias, – dar reikia įsitikinti, ar priešais robotuką nėra langelio

, kuriame robotukas turi pasisukti į kairę ir kuris nustatomas naudojant tokią kortelę

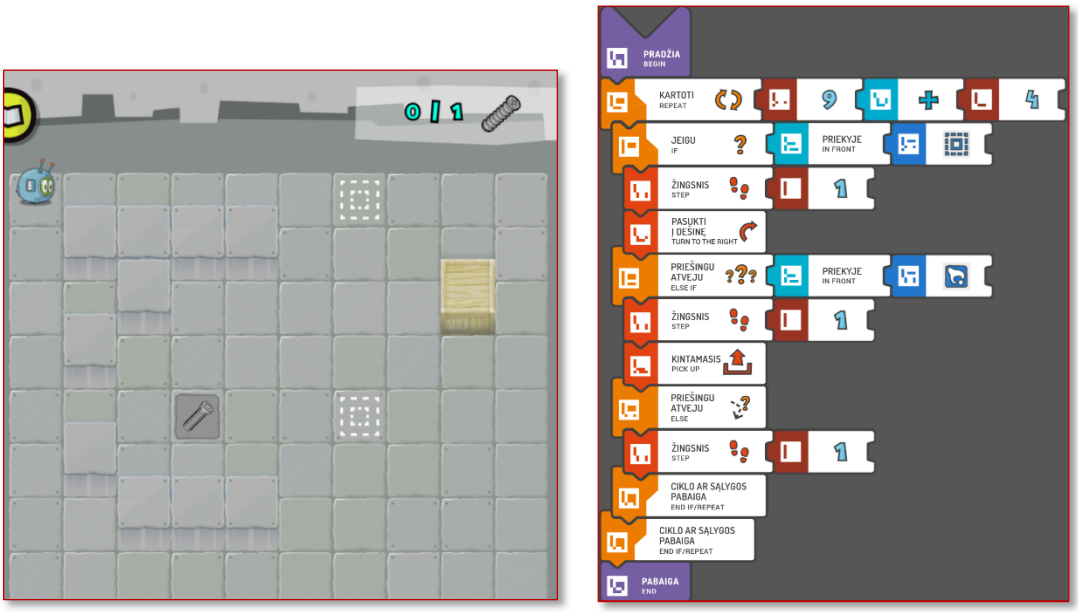
.

Šioje užduotyje, atsižvelgiant į sąlygos patenkinimą, atliekami net trys skirtingi veiksmai. Jei priešais esantis langelis yra tuščias, reikia tik žengti į jį, o jei ne, tada tikrinama, ar yra posūkio langelis, į kurį įžengus reikia pasisukti į kairę

. Jei taip, tada žengiama į langelį ir pasisukama. Jei minėtos dvi sąlygos nepatenkinamos, tai lieka vienintelė trečioji, priešais robotuką esančiame langelyje yra detalė. Robotukas žengia į langelį ir pasiima detalę.

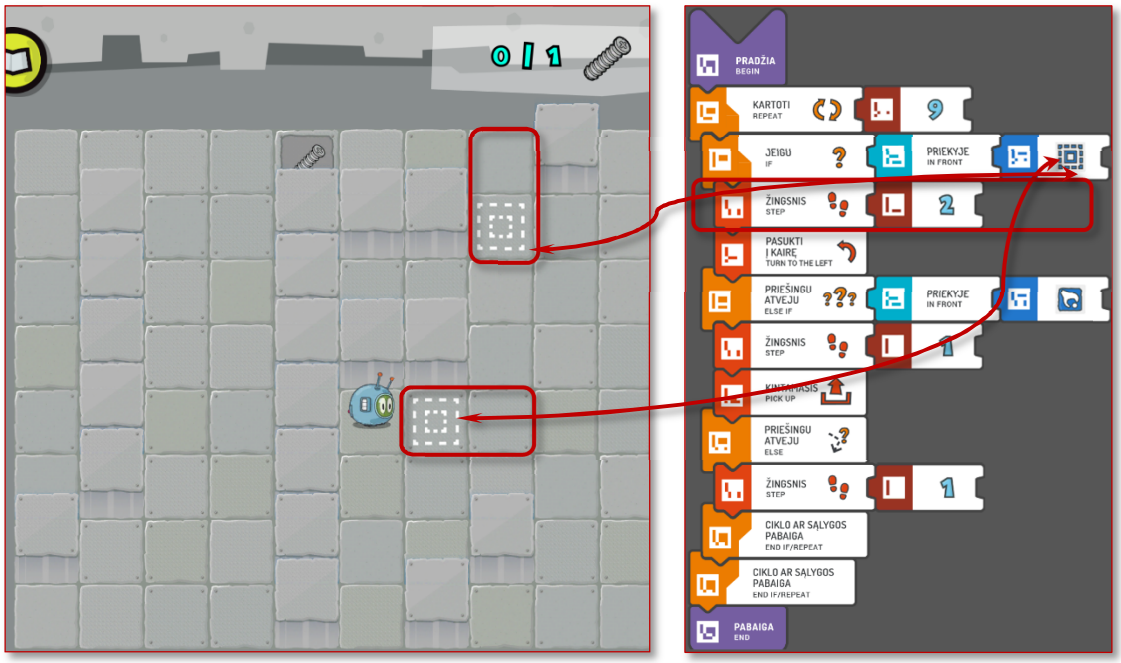


Sudėtingas sąlyginis sakinys naudojamas taip pat 3 ir 4 užduotyse.



3 užduotis ir jos programa

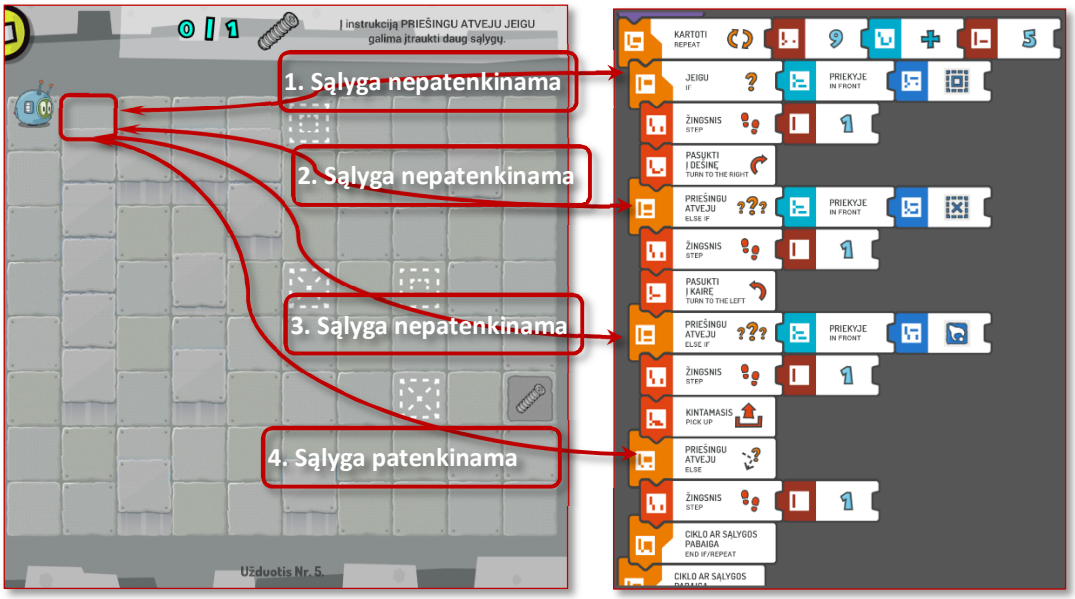
Kuriant 4 užduotį, prasminga įsidėmėti, kad pasisukti į kairę robotukui reikia tik apsilankius langelyje su dviem baltų punktyrų kvadratukais ir įžengus į kitą tuščią langelį.



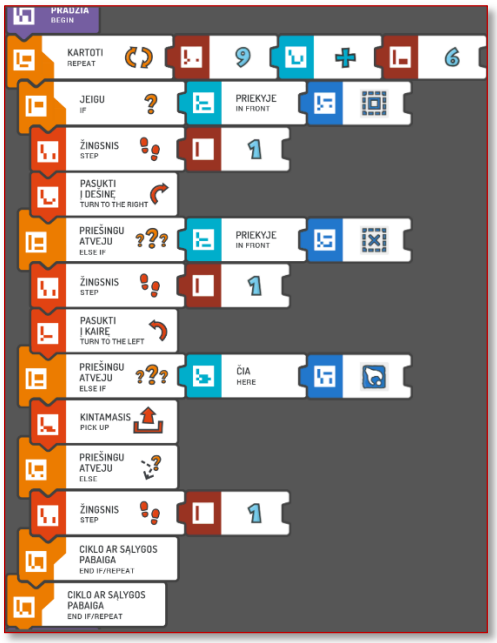
4 užduotis ir jos programa

5 ir 6 užduotyse dar atsiranda kito tipo langeliai , kuriuose turi robotukas apsilankyti.

Su kryžiuku centre esantys kvadratiniai langeliai algoritme žymimi tokiu blokeliu .
Todėl sąlyginis sakinys dar sudėtingėja (4 skirtingi atvejai).



5 užduotis ir jos sprendimas

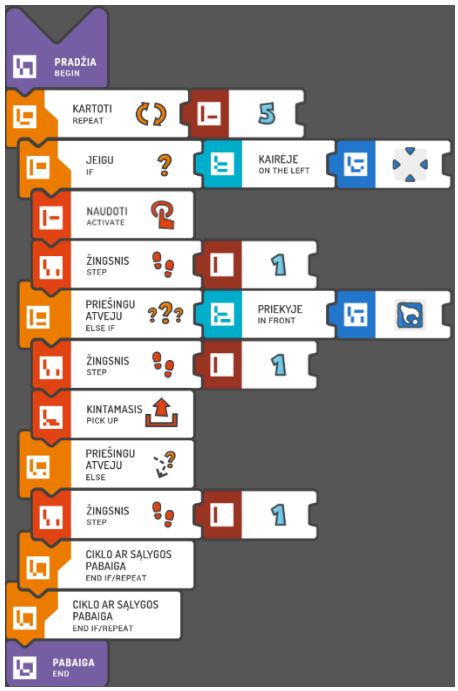


6 užduotis ir jos sprendimas

7 užduotyje yra judanti juosta (konvejeris), kuri robotuką perkelia per kelis langelius.

Prie juostos yra specialus langelis su jungikliu. Aptikus jungiklį, reikia jį įjungti, todėl naudojamas blokelis *NAUDOTI* (*ACTIVATE*)

.



8 užduotis analogiška 7 užduočiai, tik čia reikia paimti dvi detales.

9 užduotyje robotukui dar reikia ir pasisukti, todėl prisideda viena sąlyga.

Įveiktas devintasis etapas! Lieka tik viena detalė, ir erdvėlaivis bus parengtas skrydžiui.

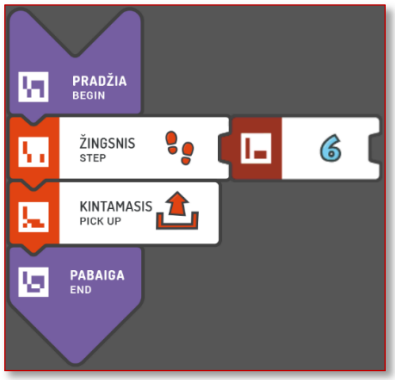
Rekomenduojama mokiniams prisiminti užduotis. Mokytojas paskirsto mokinius po du ir jiems išrenka iš kiekvieno etapo po kelias užduotis. Mokiniai užduotis atlieka (sudėlioja blokelius į programą) ir jas nufotografuoja paprastai, tada nufotografuoja *Scottie Go!* programėle, o rastas klaidas ištaiso.

Nufotografuoja ištaisytas blokelių programas ir jas palygina su pirmomis nuotraukomis. Grupelėmis aptaria užduotis.

Priedas

Užduotys ir jų algoritmai, kai nenaudojamos sąlyginio sakinio konstrukcijos. Atlikus tokiu būdu, gaunama tik viena žvaigždutė. Pavyzdžiui, po 2 užduoties:

1 užduotis ir jos algoritmas:

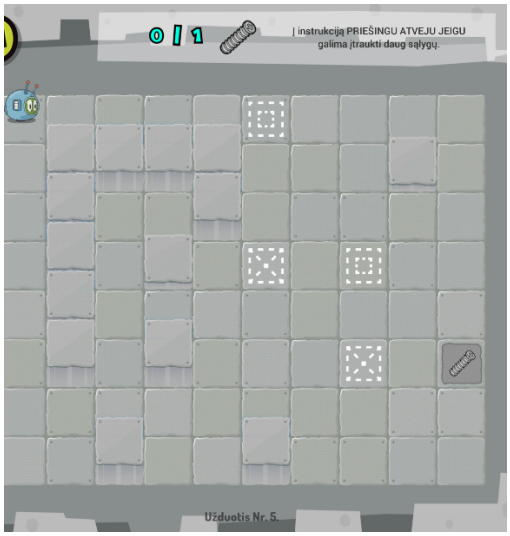


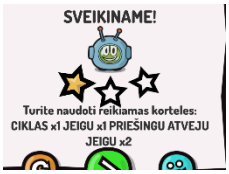
2 užduotis ir jos algoritmas:

3 užduotis ir jos algoritmas:

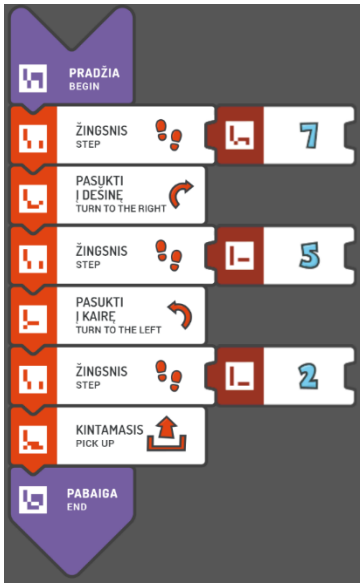
4 užduotis ir jos algoritmas:

5 užduotis ir jos algoritmas:

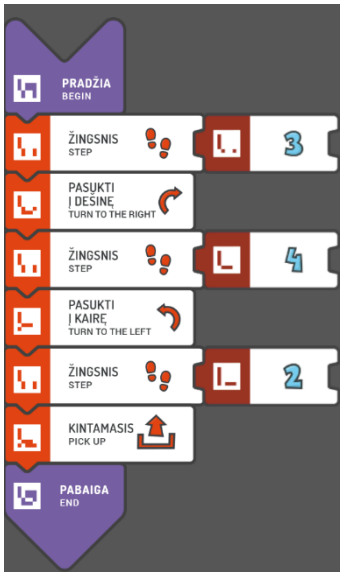




6 užduotis ir jos algoritmas:



7 užduotis ir jos algoritmas:



8 užduotis ir jos algoritmas:

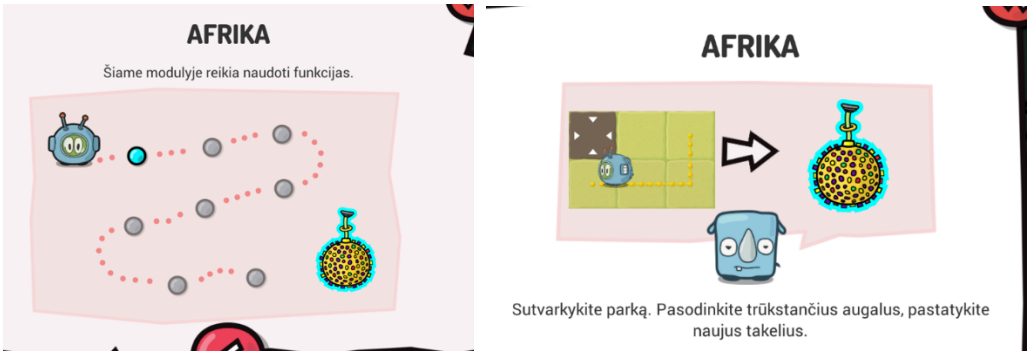
9 užduotis ir jos algoritmas:

Dešimtasis etapas

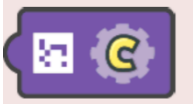
Dešimtasis etapas skirtas susipažinti su funkcijomis – programos savarankiška konstrukcija, į kurią, vykdant programą, galima kreiptis kiek tik norima kartų.

Robotukas tęsia kelionę Afrikoje.

Čia robotukas užsiims nauja veikla – tvarkys parką: sodins augalus, ties naujus takelius.



Žaidime galima naudotis trimis funkcijomis *A, B* ir *C*. Raidės reiškia funkcijos pavadinimą.

Funkcija *A*	Funkcija *B*	Funkcija *C*
		

Šiame žaidime funkcijos kuriamos kaip savarankiškos programos, tik truputį skiriasi jų konstrukcija.

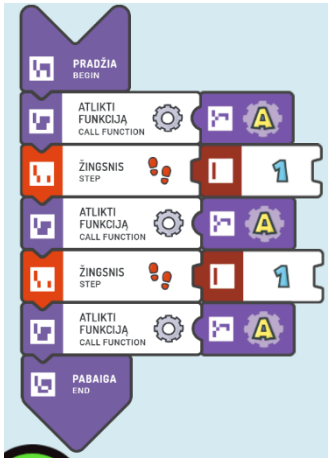
Čia yra funkcijos pradžios ir pabaigos blokeliai:

Prie viršutinio blokelio prijungiamas blokelis su raide *A, B* ir *C*. Tai reiškia, kad galima kurti tris skirtingas funkcijas *A, B* ir *C*. Funkcijų blokeliai dėliojami atskirai. Taip pat funkcijos fotografuojamos atskirai.

Veiksmai funkcijos viduje niekuo nesiskiria nuo programos veiksmų.

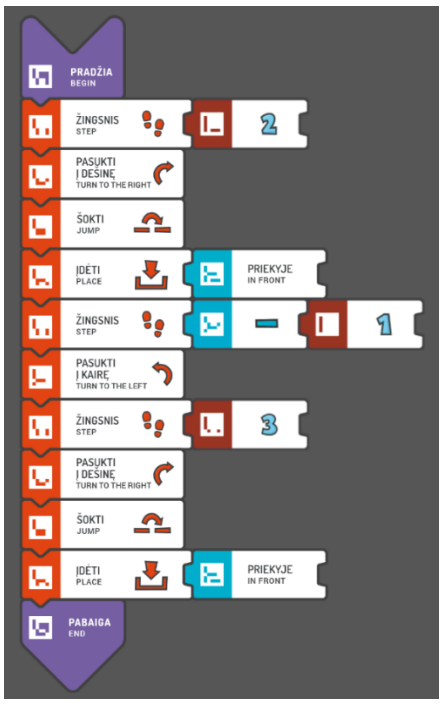
Naudojant funkcijas programoje, reikia įterpti blokelį „Atlikti funkciją“ ir prie jo prijungti blokelį su atitinkama funkcijos raide.

. Funkciją galima įterpti kelis kartus. Paveikslėlyje funkcija *A* įterpta į programą tris kartus:

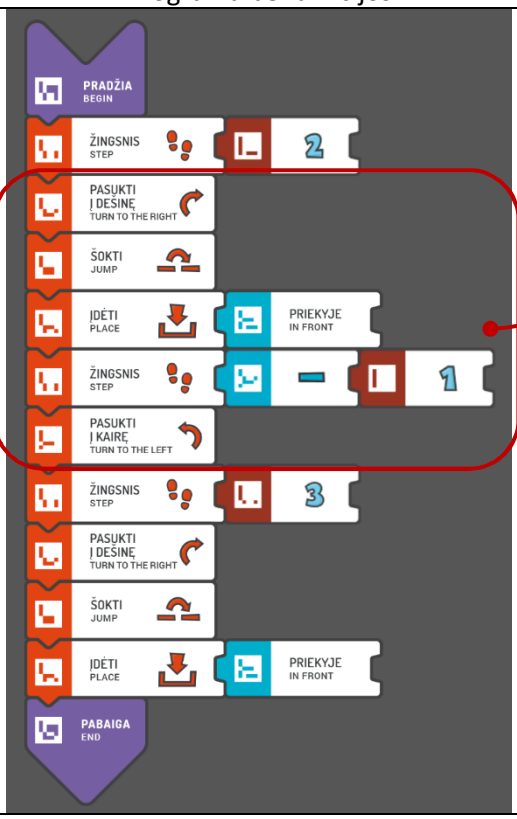
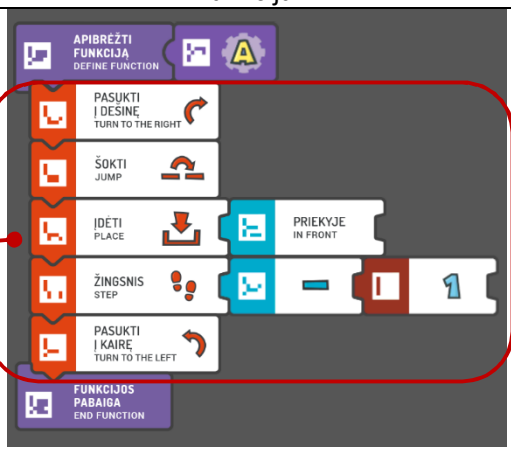


1 užduotis reikalauja pasodinti du augalus.

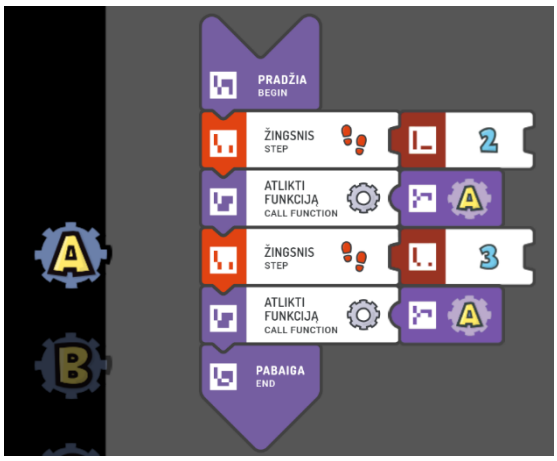
Galima sukurti teisingą programą, kurioje nebus naudojama funkcija.



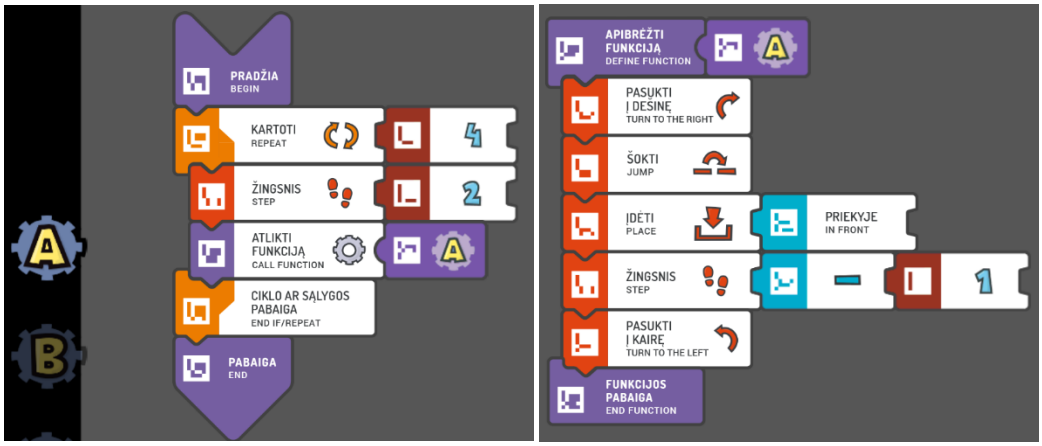
Tačiau atlikus tokią programą, bus skirta tik viena žvaigždutė. Norint surinkti visas tris žvaigždutes, tenka kurti pirmąją funkciją.

Programa be funkcijos	Funkcija *A*
	

Programa sukurta pagal reikalavimą naudoti funkciją, nes du kartus panaudota funkcija *A*. Todėl ir dega visos trys žvaigždutės.

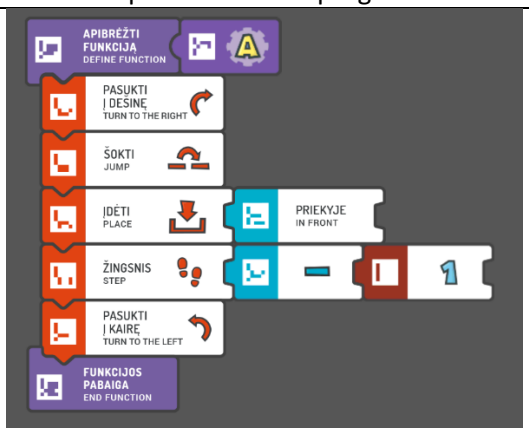


2 užduotis įrodo funkcijos naudojimo patogumą. Nebereikia kurti naujos funkcijos, nes puikiai tinka senoji. Reikia keturis kartus robotukui žengti į priekį ir pasisukus į dešinę užšokti ant pakelto langelio, įdėti priekyje augalą, grįžti vienu žingsniu atgal ir pasisukti kairėn. Ir tokius veiksmus kartoti dar likusius tris kartus.



2 užduoties funkcija ir programa (algoritmas)

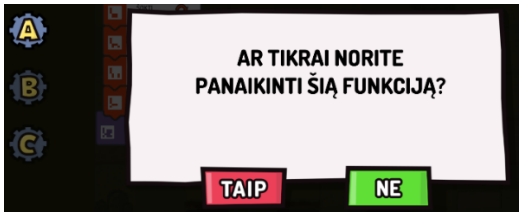
3 užduotis

3 užduoties programa	3 užduoties programa naudoja tą pačią funkciją *A*, kaip ir 1 užduoties programa
	

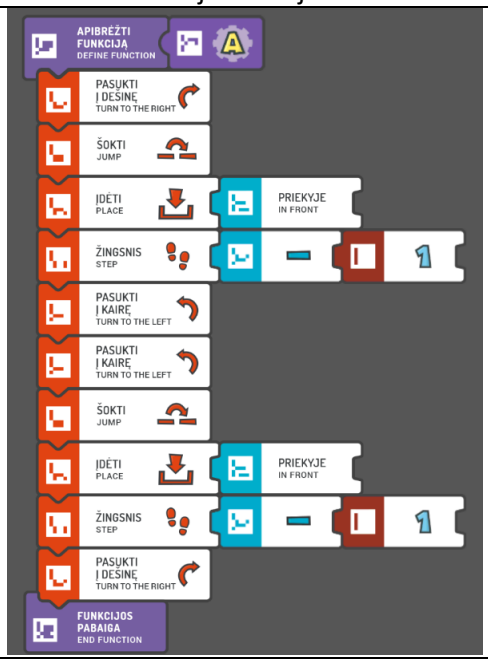
Atliekant šią programą sodinami šeši augalai. Ciklas atliekamas tris kartus, nes kiekvieną kartą atliekant ciklą, programa kreipiasi į funkciją du kartus.

Ekrano vaizdas, kai baigiama vykdyti 3 užduotis (jau pasodinti šeši augalai):

Jei esame įsitikinę, kad funkcija nebereikalinga, galime ją ištrinti. Prieš trynimą bus paklausiama, ar tikrai norime ištrinti, nes galima vienu metu turėti tris sukurtas funkcijas *A, B* ir *C*.

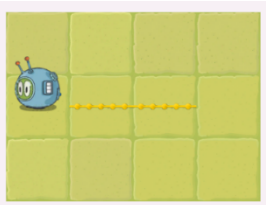


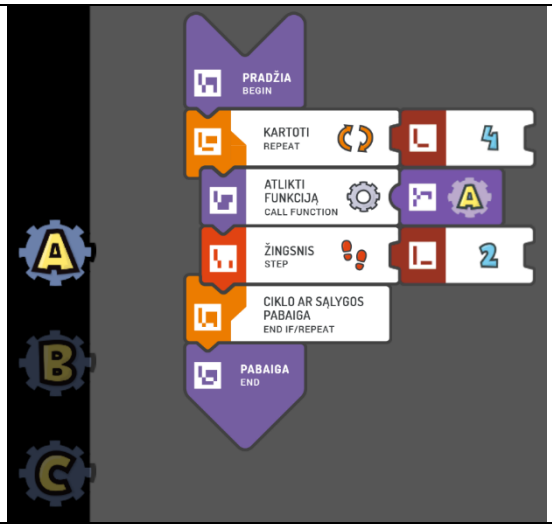
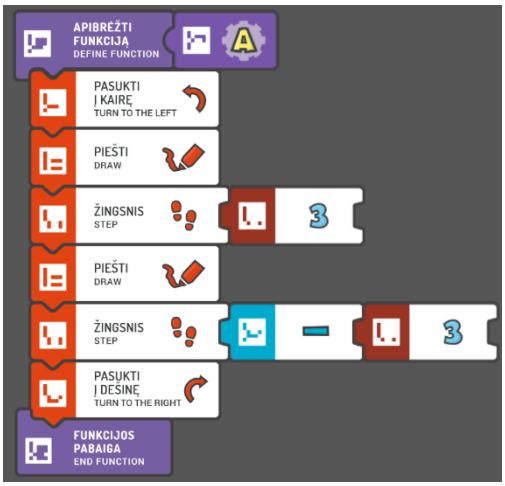
Pateikiame kitą pavyzdį, kaip galima atlikti 3 užduotį

Programa jau gerokai trumpesnė	Nauja funkcija *A*
	

Atlikdamas 4 užduotį robotukas piešia. Naudojamas blokelis *PIEŠTI (DRAW)*:

. Šis blokelis įjungia, o panaudotas antrą kartą išjungia piešimą. Piešimas langeliuose be taškų yra klaida.

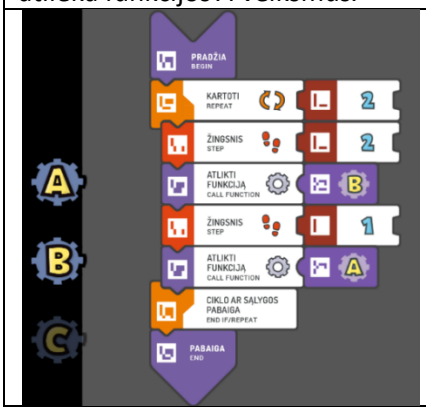
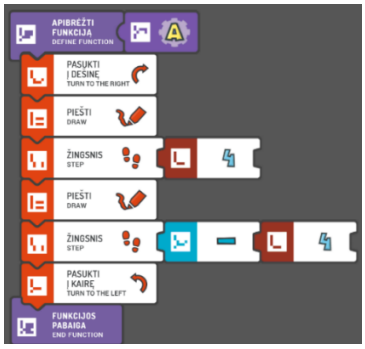
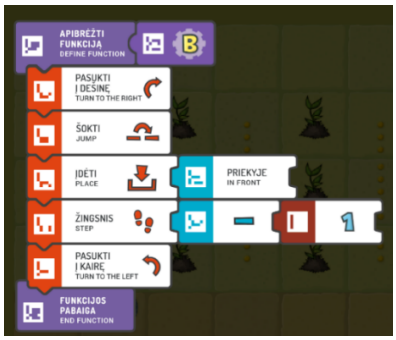


Programa kreipiasi į funkciją *A* keturis kartus, nes reikia nupiešti keturias linijas. Kiekvieną kartą atlikęs *A* funkciją, robotukas turi žengti per du langelius pirmyn.	Nauja funkcija *A* robotuką pasuka kairėn, tada įjungiamas piešimas ir piešiama trijuose langeliuose (trys žingsniai). Tada piešimas išjungiamas ir robotukas grįžta tris langelius atgal ir pasisuka dešinėn.
	

Paveikslėlyje vaizduojamas robotukas, kuris piešia jau ketvirtą liniją. Pasirinktas antras atlikimo greitis.

5 užduotis reikalauja jau dviejų funkcijų: viena funkcija piešia, o kita sodina augalą. Piešti kiekvieną kartą reikia jau keturiuose langeliuose.

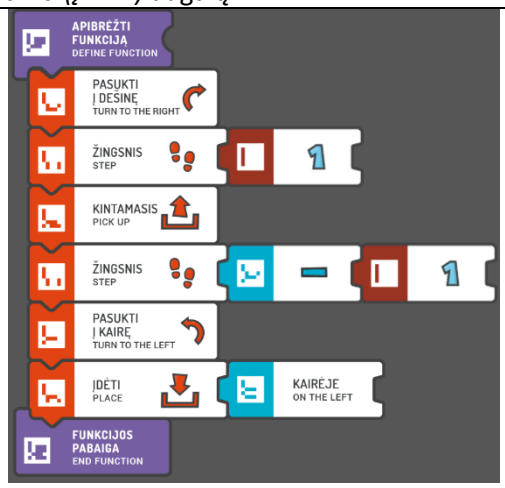
Robotukas turi žengti į priekį per du langelius, tada pasisukti dešinėn, užšokti ant pakilusio langelio, pasodinti augalą, grįžti per vieną langelį atgal, pasisukti į kairę (funkcija *B*). Tokius veiksmus jam reikės dar kartą atlikti vėliau. Dabar jis turi žengti vieną žingsnį pirmyn, tada pasisukti dešinėn ir piešti keturis langelius, nustoti piešti, grįžti per keturis langelius atgal ir pasisukti kairėn (funkcija *A*).

Atliekant programą, pirmiausia kreipiamasi į *B* funkciją, o paskui – į *A*. Funkcijų vardų eilė nesvarbu, svarbu, kad funkcija atliktų teisingus veiksmus. Veiksmai kartojami du kartus. Ciklo viduje atlikus *B* funkciją, robotukas žengia per vieną langelį, o po to atlieka funkcijos *A* veiksmus.	Funkcija *A* piešia	Funkcija *B* sodina augalą
		

Paveikslėlyje vaizduojama beveik atlikta 5 užduotis – nupiešti iš viso aštuoni langeliai ir pasodinti du augalai.

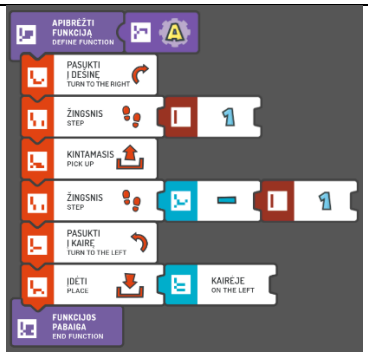
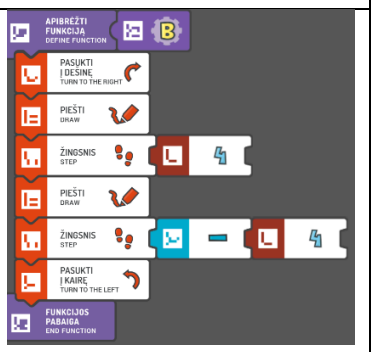
6 užduotis

6 užduotis reikalauja pasodinti keturis augalus, tačiau robotukas turi eiti per daugiau langelių. Jam reikia kiekvieną kartą tikrinti, ar yra langelis, kuriame galima sodinti. Ši programa reikalauja jau sudėtingesnio ciklo naudojimo. Užduotis reikalauja naudoti nežinomo kartojimo skaičiaus ciklą *KARTOTI KOL (WHILE)*.

Programa naudoja ciklą, kuris tikrina, ar langelis, esantis prieš robotuką, yra tuščias. Jei taip, tada atliekami tolesni veiksmai; jei ne, tada programa pabaigiama.	Funkcija *A* liepia robotukui pasisukti į dešinę, eiti per vieną langelį, paimti augalą, žengti vieną žingsnį atgal. Tada pasisukti į kairę ir kairėje pusėje pasodinti *(ĮDĖTI)* augalą.
	

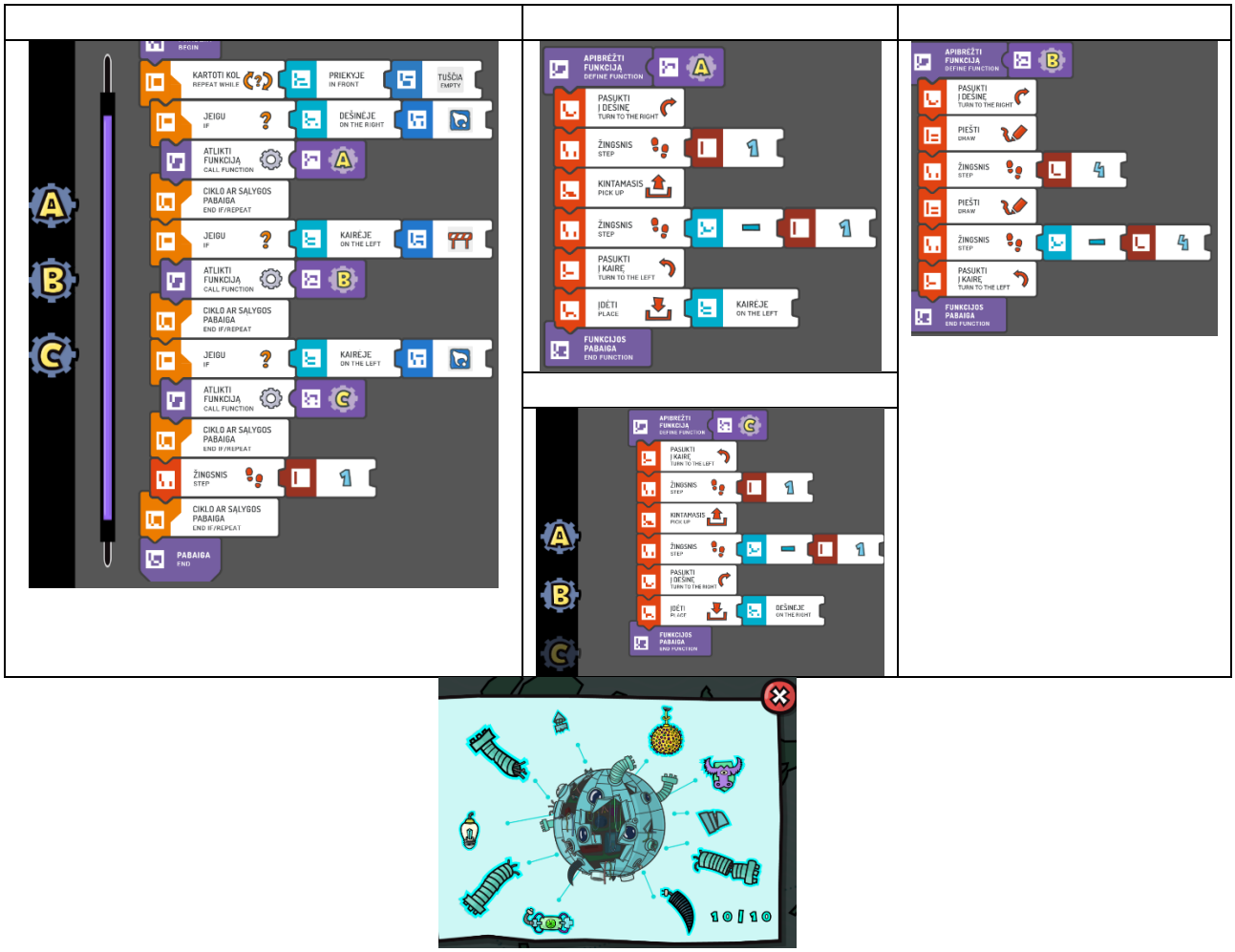
Atlikta 6 užduotis

7 užduotyje galima pasinaudoti jau sukurta funkcija *A*, taip pat vėl reikės piešti. Svarbu įsidėmėti, kad užvėrus žaidimą, dingsta sukurtos funkcijos.

Programa naudojasi funkcija *A* ir funkcija *B*. Funkcija *A* paima augalą ir pasodina į reikiamą vietą, o funkcija *B* piešia. Naudojamas nežinomų kartojimų skaičiaus ciklas *KARTOTI KOL (WHILE)* ir du sąlygos sakiniai. Jei dešinėje yra augalas, tada atliekami funkcijos *A* veiksmai; jei kliūtis yra kairėje – *B* veiksmai.	Funkcija *A* jau žinoma.	Funkcijos *B* atliekami veiksmai taip pat jau nagrinėti.
		

Atlikta 7 užduotis

8 užduotis



Žaidimas baigtas, mokiniai pagelbėjo robotukui surinkti visas 10 detalių. Robotukas sėkmingai iškeliauja.

Pastabos

Primename, kad atlikus visas užduotis galima grįžti ir kurti, fotografuoti bet kurią užduotį. Tai labai svarbu, įvirtinant mokinių gebėjimą planuoti ir kurti robotuko maršrutus įvairiais būdais. Rekomenduojama mokiniams prisiminti užduotis. Mokytojas paskirsto mokinius po du ir jiems išrenka iš kiekvieno etapo po kelias užduotis. Mokiniai užduotis atlieka (sudėlioja blokelius į programą) ir jas nufotografuoja paprastai, tada nufotografuoja *Scottie Go!* programėle, o rastas klaidas ištaiso. Nufotografuoja ištaisytas blokelių programas ir jas palygina su pirmomis nuotraukomis. Grupelėmis aptaria užduotis.

Žaidime yra keli nenaudotini blokeliai *NUTRAUKTI*

, *NUTRAUKTI FUNKCIJĄ*

ir *NUOLAT*

. Šioje medžiagoje šie blokeliai buvo nenaudojami, nes jie susiję su pasenusiomis programavimo technologijomis.