



**Vilniaus
universitetas**

Informatikos ir informatinio mąstymo mokomoji veikla
pagrindiniam ugdymui

Algoritmovimo pradžia: priemonės ir metodika

Mokyklos pedagogika



Kuriame
Lietuvos ateitį
2014–2020 metų
Europos Sąjungos
fondų investicijų
veiksmų programa



**Vilniaus
universitetas**

Informatikos ir informatinio mąstymo mokomosios veiklos sukurtos įgyvendinant projektą „Aukštųjų mokyklų tinklo optimizavimas ir studijų kokybės gerinimas Šiaulių universitetą prijungiant prie Vilniaus universiteto“ (Nr. 09.3.1-ESFA-V-738-03-0001), finansuojamą iš Europos socialinio fondo lėšų pagal 2014–2020 metų Europos Sąjungos fondų investicijų veiksmų programos 9 prioriteto „Visuomenės švietimas ir žmogiškųjų išteklių potencialo didinimas“ įgyvendinimo priemonę Nr. 09.3.1-ESFA-V-738 „Aukštųjų mokyklų tinklo tobulinimas“.

Metodinė medžiaga „Algoritmavimo pradžia: priemonės ir metodika“, skirta Mokyklos pedagogikos studijų programos moduliui „Informatikos didaktika“. Tikslinė grupė – būsimi informatikos pagrindinio ugdymo mokytojai. Medžiaga siejasi su Bendrosiomis informatikos ir matematikos programomis, ypač su algoritmų pažintimi, jų atlikimu, pateikiami teoriniai samprotavimai būsimiems mokytojams, aptariamos pagrindinės temos sąvokos. Atlikdami veikloje numatytas užduotis studentai išsiaiškins, kaip pradėti kurti įvairius algoritmus, atliks praktines algoritmavimo užduotis. Pateikiamas teorinis temos pagrindimas būsimam mokytojui, aptariamos pagrindinės srities sąvokos.

Šioje informatikos ir informatinio mąstymo veikloje dalis medžiagos panaudota iš tarptautinio informatikos mokymo tinklo „Informatika be kompiuterio“ (*Computer Science Unplugged*: csunplugged.org), taip pat tarptautinio informatinio ir informatikos iššūkio „Bebras“. Visi naudojami išteklių (įskaitant ir paveikslus) yra sukurti taikant *Creative Common* licenciją.

Šios veiklos rengėjai: prof. dr. Valentina Dagienė ir dr. Eglė Jasutė

Redagavo: Viktoras Dagys

Vilnius, 2022

Įvadas

Algoritmavimas – svarbi veikla, su kuria turi būti supažindinami mokiniai. Todėl mokytojai turi gerai suprasti algoritmų paskirtį, sandarą, užrašymą, vykdymą, testavimą. Algoritmavimas glaudžiai siejamas su programavimu. Pradedant mokytį programavimo daugiausiai naudojamas blokinis programavimas naudojant **XLogo** ar **Scratch**. Naudojamos ir kitos žaidybinės programavimo aplinkos.

Informatikos bendrojo ugdymo programoje algoritmų ir programavimo srityje išskiriami tokie mokinių pasiekimai:

- B1. Įžvelgia algoritmų, programų naudą, atpažįsta ir vartoja pagrindines sąvokas.
- B2. Naudojasi algoritmavimo, programavimo kalbos konstrukcijomis, programavimo aplinkomis.
- B3. Kuria ir vykdo algoritmus, programas.
- B4. Testuoja, derina, tobulina programas.

Diegiant šių pasiekimų ugdymą praktiškai svarbu pasirinkti tinkamas priemones ir metodiką.

Priemonių algoritmavimui mokytis pasirinkimo principai

Kaip pasirinkti priemonę mokinių algoritmavimo (taip pat ir programavimo) gebėjimams ugdyti? Be abejo, pasirinkimas priklauso nuo daugelio veiksnių, išimčių, tad vieno bendro recepto negali būti.

Galima remtis slovakų mokslininko Ivan Kalaš suformuluotais principais (Kalaš I. 2018).

Mokant algoritmavimo, turi būti ugdomi daiktų valdymo, manipuliavimo įgūdžiai. Mokytojai turi atkreipti dėmesį į vaikų gebėjimus valdyti objektus. Labai svarbu atsižvelgti į tai, kaip vaikas gali valdyti objektą, sakoma, koks atstumas tarp vaiko ir jo valdomo objekto. Pagal tai objektų valdymą galima skirstyti į kelias grupes:

- Tiesioginė manipuliacija
- Netiesioginė manipuliacija
- Tiesioginis valdymas
- Programuojamas valdymas

I. Kalaš vadovaudamasis daugeliu atliktų tyrimų teigia, kad kuo mažesnis vaikas, tuo mažesnis atstumas turėtų būti tarp jo ir valdomo objekto: pradėti reikėtų nuo tiesioginės manipuliacijos ir judėti link programuojamojo valdymo. Pirmieji du žingsniai tinkami net ankstyvo ikimokyklinio amžiaus vaikams.

Tiesioginė manipuliacija – tai pats paprasčiausias valdymo būdas, kai vaikas tiesiogiai valdo objektą, pavyzdžiui, ima žaislą, nuneša jį į nurodytą vietą ir padeda. Šiuo atveju atstumas mažiausias – vaikas tiesiogiai manipuliuoja objektu, pavyzdžiui, koku nors žaislu.

Su *netiesiogine manipuliacija* susiduriame, pavyzdžiui, planšetinių kompiuterių ar telefonų programėlėse, kai spustelime tą vietą, kurioje turi atsidurti valdomas objektas. Pavyzdžiui, žemiau esančioje iliustracijoje – spustelime langelį su saga, iki kurios turi ateiti robotas.



Iliustracija iš I. Kalaš straipsnio (Kalaš, 2018)

Tiesioginis valdymas būna tada, kai objektas (veikėjas, robotas) valdomas kryptinių mygtukais. Ilustracijoje – kryptinių rodyklėmis apačioje dešinėje.



Ilustracija iš I. Kalaš straipsnio (Kalaš, 2018)

Programuojamasis valdymas – komandų seka surašoma atskirai, naudojant sutartinius ženklus. Vykdoma ne po vieną komandą, o visa seka iš karto.



Ilustracija iš I. Kalaš straipsnio (Kalaš, 2018)

Mokant algoritavimo ir programavimo gebėjimų natūraliai realizuojami *konstrukcionistinio mokymosi* principai, suformuluoti S. Paperto, žymaus matematiko, edukologo, informatiko, dirbtinio intelekto teorijos vieno iš pradininkų, Logo programavimo kalbos autoriaus. Anot Paperto, mokymasis geriausiai vyksta tada, kai besimokantysis aktyviai kuria išorinio pasaulio objektus (o ne tik žinių struktūras, kaip mentalinius objektus) (Papert, 1995). Konstrukcionistinį mokymąsi gerai apibūdina frazė *mokymasis kuriant* (angl. *learning by making*). Tokį mokymąsi realizuoti padeda informacinės ir komunikacinės technologijos, o „išorinio pasaulio objektai“ (artefaktai) – tai mokinių kuriamas turinys, sąveikaujant su kuriuo vyksta mokymosi procesas: programos, žaidimai, projektavimo proceso (plakatai, pateiktys, nuotraukos) ir kt. darbai. Mokymas(is), paremtas konstrukcionizmu, nėra žinių ir įgūdžių pasyvus perteikimas, tai aktyvus supratimo, reikšmių ir įgūdžių konstravimo procesas, ankstesnių ir naujų žinių ir įgūdžių susiejimas. Vaikas turi mokėti organizuoti, planuoti ir kontroliuoti savo mokymosi procesą, kuris turi tikslą ir yra susijęs su kontekstu bei

aplinkybėmis. Kompiuterinės konstravimo priemonės vaikui gali padėti įsitraukti į aktyvų objektų kūrimą.

Panagrinėkite, aptarkite. Paskaitykite S. Paperto knygą „Minčių audros“ (1995), raskite, ką šis mokslininkas rašo apie vaikų algoritmavimo mokymąsi. Išrašykite keletą jums patikusių citatų, aptarkite su kolegomis.

Kitas svarbus mokymo metodas – žaidimas, žaidybinio veiklos. Tai holistinis būdas patirti ir tyrinėti pasaulį. Jis remiasi motyvacija, laisvu pasirinkimu, teikia malonumą, turi netiesioginę prasmę ir aktyviai įtraukia mokymosi proceso dalyvius. Pagrindinė žaidimu paremta mokymo ir mokymosi esmė yra mokymas aktyviai įsitraukus, atliekant veiksmus, kurie įtraukia, todėl nejučia kartojami ir išmokstami nemaža dalis rutininių veiksmų, mokymasis iš klaidų, jos taisomos, nes siekiama įgyvendinti žaidimo tikslus. Žaidimai paprastai turi ir kūrybiškumo elementų, leidžia rinktis, apmąstyti situacijas.

Ugdant algoritmų ir programavimo srities gebėjimus žaidimo metodas įgyja daugialypę prasmę, pavyzdžiui:

- Žaidimas kaip mokymo(si) metodika.
- Žaidimas kaip algoritmas (žaidimo taisyklės).
- Žaidimas kaip mokinių kuriamas skaitmeninis turinys (paprasčiausių žaidimų programavimas).

Natūraliai integruoti algoritmų ir programavimo gebėjimų ugdymą padeda *projektinis mokymasis*, kuris suteikia mokiniams galimybę ir savarankiškai mokytis, ir mokytis bendradarbiaujant, tenkina mokinių smalsumą, skatina ugdyti kūrybiškumą, mokėjimą mokytis, sieti savo darbą su savo aplinka, kraštu, interesais.

Užduotis. Panagrinėkite svetainę „Kompetencijų ugdymas. Metodinė svetainė. Projektai. Kaip galima mokytis kitaip.“

<http://www.ugdome.lt/kompetencijos5-8/pagrindinis/kompetenciju-ugdymo-praktika/projektai/> .

KOMPETENCIJŲ UGDYMO PRAKTIKA

→ PAGRINDINIAI KOMPETENCIJŲ UGDYMO ASPEKTAI

Pamoka

Metodai

Projektai

Projektai: galima mokytis kitaip
Didesnės integravimo galimybės
Projektas kaip metodas
Projektinio darbo etapai
Projektinės veiklos pavyzdžiai

Filmuota medžiaga

Projektai: galima mokytis kitaip

Siekiant sukurti sąlygas mokiniams įgyti bendrųjų ir dalykinių kompetencijų pagrindus, svarbu į mokymąsi įtraukti pačius mokinius, kad šie ne tik aktyviai atliktų mokymosi užduotis, bet ir patys kurtų ugdymo turinį: keltų jiems aktualias problemas ir ieškotų atsakymų į jiems rūpimus klausimus. Vienas iš veiksmingų būdų tai daryti – projektinė veikla, nes:

- kiekvienas projektas turi tikslą, apibrėžtą trukmę, vadinasi, mokiniams suteikiama galimybė drauge su mokytoju ar savarankiškai mokytis planuoti mokymosi laiką, dirbti su įvairiais informacijos šaltiniais, pritaikyti žinias praktikoje, dirbti ir grupėje, ir savarankiškai;
- projektai tenkina mokinių smalsumą, nes dažniausiai susiję su tyrimais ar kūrybine raiška, teikia galimybių pasireikšti ir nuojautai, ir intelektui. Projektų dalyviai skatinami teikti kuo daugiau idėjų ir jas įgyvendinti. Projektinė veikla mokiniui suteikia galimybių stebėti savo pažangą, ja džiaugtis;

Svetainė „Kompetencijų ugdymas“

Mokant algoritmavimo ir programavimo svarbus *įvairių medijų integravimas* (programuojant manipuluojama garsais, nuotraukomis, vaizdo įrašais ir pan.). Visa tai leidžia atlikti žaidybinės programavimo aplinkos vaikams, nagrinėjamos tolesnėse šios veiklos dalyse.

Tą patį algoritmą galima pateikti įvairiais būdais, įvairiomis „kalbomis“. Štai tik keletas „algoritmavimo kalbų“ pavyzdžių:

- Schemos sudarymas.
- Veiksmų kortelių dėliojimas.
- Piešimas.
- Šokis.
- Programa kompiuterinėje programavimo aplinkoje.
- Roboto valdymas.
- Pasakojimas.

Algoritmų ir programavimo srities veiklas galima organizuoti įvairiais būdais:

Be kompiuterio:

- Kompiuterio / roboto vaidmenį atlieka vaikai, mokytojas, žaislai ir pan.

Naudojant išmaniuosius žaislus, robotukus:

- Tiesioginis robotuko valdymas.

- Robotuko valdymas naudojant kompiuterį.

Kompiuteriu:

- Virtualių veikėjų programavimas interaktyvioje aplinkoje.
- Sudėtingesnių programų rašymas.

Mokydami algoritmavimo turime aptarti algoritmo sąvoką ir pateikti pavyzdžių. Algoritmai gali būti skirstomi į dvi stambiausias rūšis:

- Algoritmai, kuriami konkrečiai užduočiai atlikti (pvz., kaip rasti lobį duotame labirinte).
- Algoritmai tam tikrai uždavinių klasei spręsti. Tai dažniausiai jau sukurti, žinomi algoritmai, pavyzdžiui, duomenų rikiavimo algoritmas, netgi galime įvardinti konkrečius algoritmų pavadinimus, burbulo rikiavimo metodas arba greitas rikiavimas.

Antrojo tipo algoritmai gana sudėtingi, pradedant mokyti algoritmavimo jie nenagrinėjami, tačiau galimas tokių algoritmų konkrečių atvejų nagrinėjimas, veikimo suvokimas nenaudojant kompiuterio (pvz., „Bebro kortelių uždaviniai“, „Informatika be kompiuterio“).

Algoritmai ir programavimas be kompiuterio

Yra daugybė būdų, kaip algoritmų ir programavimo srities gebėjimus ugdyti ir be kompiuterio. Aptarsime „Informatiką be kompiuterio“ ir „Bebro“ uždavinius.



Informatika be kompiuterio

Naujosios Zelandijos mokslininkas Tim Bell inicijavo veiklas algoritmams mokytis. Jo su kolegomis publikuotą knygą „Informatika be kompiuterio“ pamėgo viso pasaulio mokytojai, ji išversta ir į lietuvių kalbą (Bell, 2015). Šioje knygoje yra nemažai veiklų, skirtų 8-12 metų vaikams. Dar daugiau veiklų skelbiama svetainėje <https://www.csunplugged.org/en/> (anglų k.).

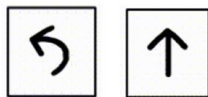
Vienas iš pavyzdžių, kuris padėtų mokiniams suprasti algoritmo pagrindus, komandas, kodėl kompiuteriui svarbu pateikti vienareikšmiškas komandas, yra **Gelbėjimo misija**.

Žaidime yra trys vaidmenys:

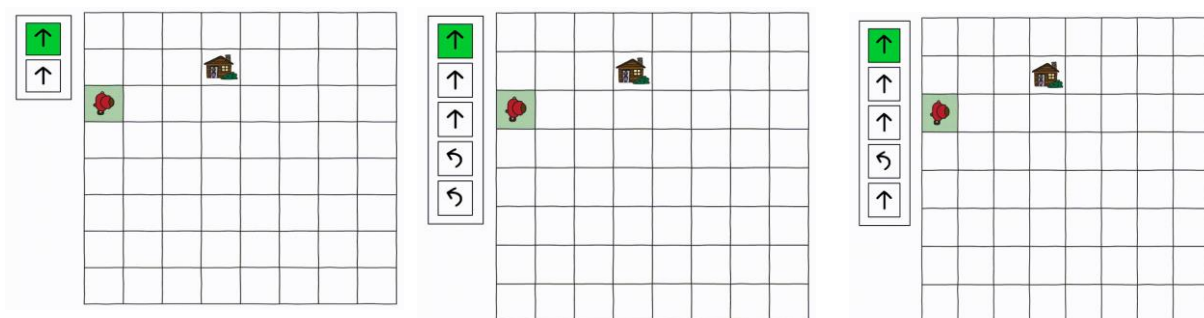
- Programuotojas (pradžioje šį vaidmenį gali atlikti mokytojas) – formuluoja komandas, rašo komandų sekas (komandų sekas galima rašyti lentoje, jei veikla organizuojama klasėje, dideliame lape, jei veikla organizuojama lauke).
- Robotas – tiksliai vykdo pasakytas komandas.
- Testuotojas – tikrina, ar pasiektas rezultatas, siūlo taisymus.

Veiklai atlikti reikalingas tinklėlis (grindų plytelės, kreida nupieštas tinklėlis ant grindų, kilimėlis ir pan.). Veikla gali vykti klasėje arba lauke. Viena iš langelių pažymėtas tikslas (paveiksle – namas, realiai vykstančioje veikloje galima padėti kokį nors žaislą, nuspelvinti langelį). Langelių paviršiuje (ant grindų plytelių, tinklėlio, kilimėlio) juda robotas, kurio vaidmenį atlieka pasirinktas grupės narys. Programuotojas (pradžioje tai gali būti mokytojas)

rašo komandas, žymėdamas jas rodyklėmis. Pradžioje galima naudoti tik dvi komandas, pavyzdžiui, „pasisuk į kairę“ ir „ženklą žingsnį į priekį“. Testuotojas prašo roboto vaidmenį atliekančio grupės nario vykdyti komandas, jei gaunamas neteisingas rezultatas, taiso programą.



Žemiau vaizduojamas bandymų, testavimo, ir pataisyto programos rezultato pavyzdžiai:

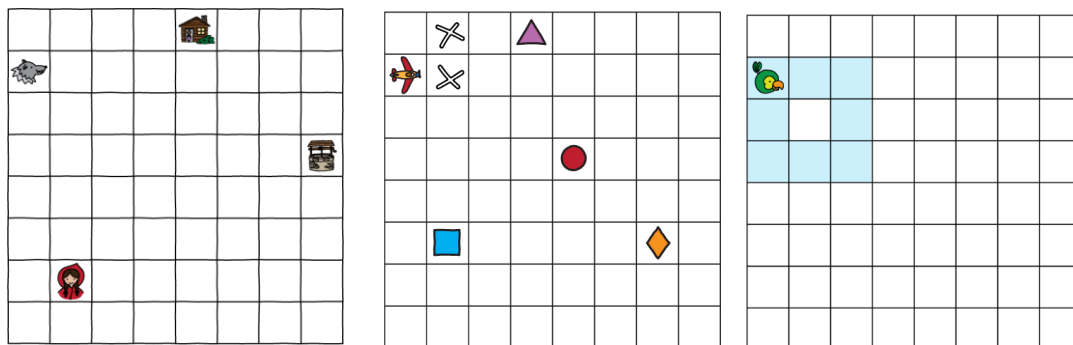


Šaltinis: <https://csunplugged.org/en/topics/kidbots/unit-plan/rescue-mission/>

Tęsiant tokio tipo veiklas, galima formuluoti papildomas sąlygas, pavyzdžiui:

- Kliūtys kelyje – kaip apeiti kliūtis?
- Kaip pasiekti tikslą naudojant tik posūkį į dešinę?

Kaip galima pritaikyti tokio tipo veiklas integruojant su kitais dalykais, pavyzdžiui, kuriant pasakojimą, mokantis geometrinių figūrų?



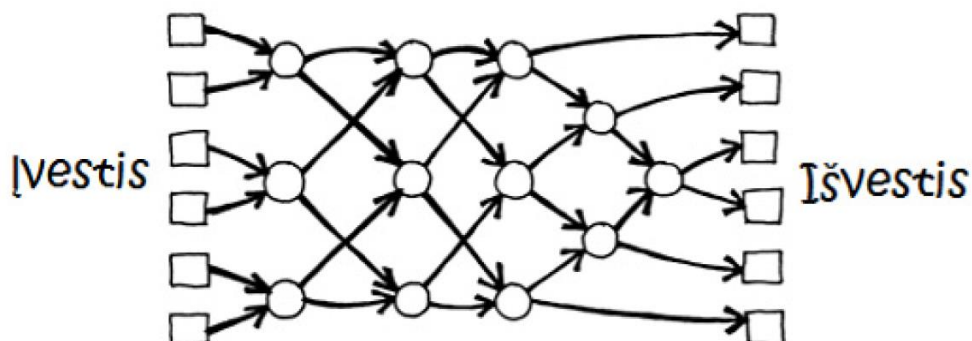
Šaltinis: <https://cs-unplugged.appspot.com>

Savarankiško darbo užduotis

Atlikite ir išsiaiškinkite vieną iš rikiavimo algoritmo metodų – tinklinį rikiavimą.

Surikiuokite skaičių seką naudodami tinklinio rikiavimo algoritmą. Veiklos scenarijus pateikiamas svetainėje „Informatika pradiniam ugdyme“

(https://informatika.ugdome.lt/wp-content/uploads/2018/08/05-inf-be-kompiuterio_1-2.pdf).



Nuotraukos. Klaipėdos Martyno Mažvydo progimnazija

Kokius objektus galime rikiuoti? (Atkreipkite dėmesį į siūlomus pavyzdžius – drugelio gyvenimo ciklas, augalo gyvenimo ciklas, pasakos įvykiai ir jų eiliškumas ir kt.).

Bebro uždaviniai

„Bebro“ informatikos ir informatinio mąstymo ugdymo konkurso-iššūkio tikslas – atskleisti mokiniams (ir mokytojams) informatikos mokslo esmę, sudominti juo, supažindinti su pagrindinėmis informatikos ir informacinių technologijų sąvokomis, jų platesniu kontekstu, motyvuoti gilintis į modernius informatikos ir inžinerijos sprendimų metodus. Įdomūs, žaidybiniais principais besiremiantys uždaviniai apima visas fundamentalias informatikos sritis, parodo jų svarbą ir dermę su kitais mokslais.

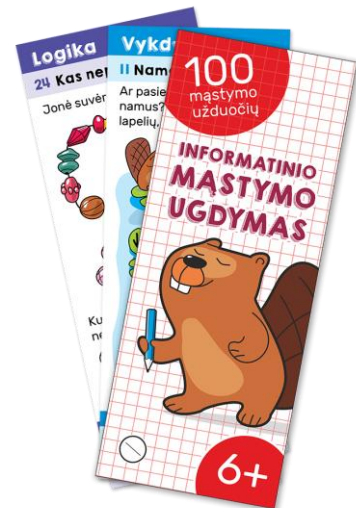
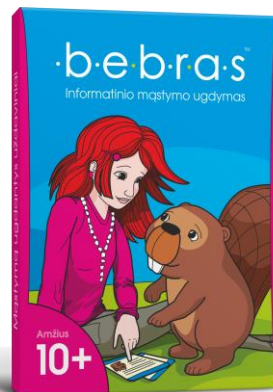
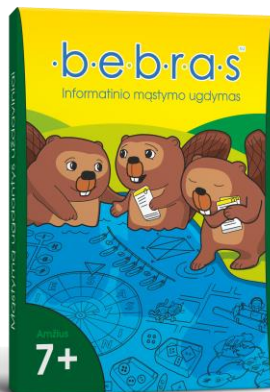
Tačiau „Bebras“ – tai ne tik konkursas, tai informatikos ir informatinio mąstymo ugdymo modelis. Mokiniai, sprenddami įdomius informatikos uždavinius, susipažįsta su informatikos sąvokomis, pagilina jų sampratą, išsiugdo gebėjimą taikyti jas praktiškai, uždavinių

sprendimą aptaria su bendraamžiais ir mokytojais. Didelė dalis uždavinių yra iš algoritmavimo srities.

„Bebro“ uždaviniai pateikiami įvairiai: kasmet parengiamos „Bebro“ uždavinių knygelės, galima įsigyti „Bebro“ kortelių pagal vaikų amžių, publikuojama „Bebro“ ir kitose svetainėse:

Informatika pradiniame ugdyme. <https://informatika.ugdome.lt/lt/biblioteka/leidiniai/>

„Bebro“ konkurso užduotys <https://bebras.lt>



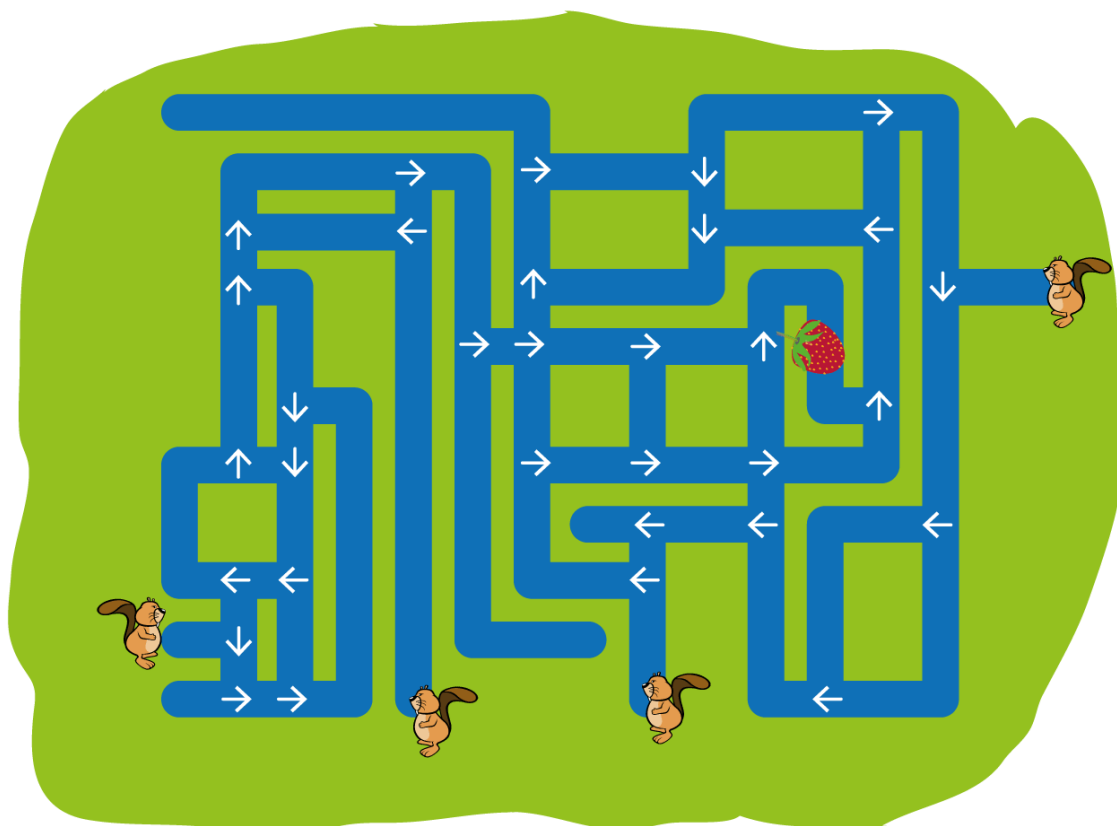
„Bebro“ kortelių rinkiniai

Užduotis. Išspręskite ir aptarkite uždavinių pavyzdžių. Kurio amžiaus vaikams skirtumėte šias užduotis? Kaip aptartumėte? Ką labiau akcentuotumėte?

Viliojanti braškė

Keturi bebrai pradeda plaukti kanalais iš skirtingų vietų ir plaukia tik rodyklių kryptimis.

Kiek bebrų nuplauks iki braškės?



Kairèn

Bebriukas turi žaislinį robotą, kuris klauso dviejų komandų:



Paeina 10 cm ģ priekģ.



Pasisuka į dešinę ketvirtadalį apskritimo, t. y. 90 laipsnių.

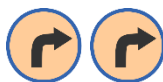
Kaip priverstum robotą pasisukti į kairę 90 laipsnių?



A.



B.



C.



D.

Katinas ir pelė

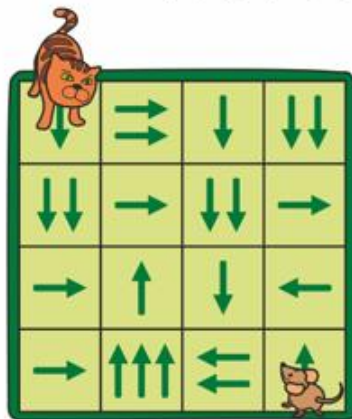
24

KATINAS IR PELĖ



Bebras sukonstravo robotukus katiną ir pelę, kurie gali judėti iš langelio į langelį pagal rodyklės. Katinas turi sumedžioti pelę.

- Pirmas pradeda katinas.
- Ėjimai daromi paeiliui.
- Einama pagal rodyklių kryptis per tiek langelių, kiek rodyklių nupiešta (pavyzdžiui, per vieną langelį, jei yra viena rodyklė, per du langelius, jei dvi, ir t. t.).
- Langelių, per kuriuos žengiama, rodyklės ignoruojamos.
- Pelė sugaunama, kai katinas atsistoja į tą patį langelį.

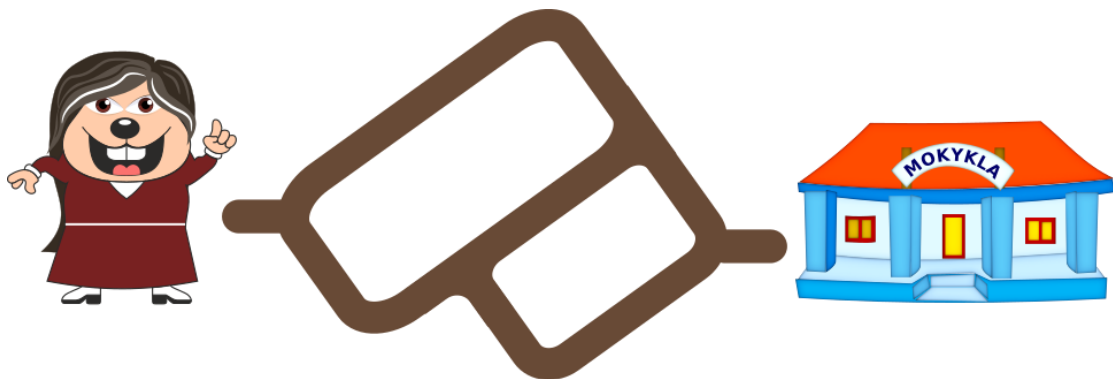


Ar pavyks pelei pabėgti?

Uždavinio žaidimo aikštelę galima laikyti programa: yra komandos, nusakantios veiksmus, nurodyta pradžia ir pabaiga. Rodyklės langeliuose – tai tikrų tikriausios komandos, jos aiškios ir nedviprasmiškos, nesvarbu, kad užrašytos ne raidėmis, o rodyklėmis. Svarbu tik sukonstruoti įrenginius, kurie atpažintų, suprastų ir mokėtų vykdyti šias komandas. Čia matome netgi du įrenginius – robotus katiną ir pelę, kurie gali judėti pagal rodyklės.

Kiek skirtingų maršrutų?

Bebriukė Jonė dažnai eina pėsčiomis į mokyklą.



Ji mėgsta kasdien keisti savo maršrutą. Jonė renkasi tuos kelius, kuriais galima greičiau pasiekti mokyklą. Kiek skirtingų maršrutų Jonė gali pasirinkti?

Pagrindinės algoritmavimo komandos

Algoritmų, programų kūrėjams ir vartotojams reikalingos kelios pagrindinės struktūros. Tai – *komandų seka, šakojimo ir kartojimas*. Su visais šiais konstruktais galima supažindinti mokinius pradedančius programuoti. Tam, kad mokiniai suvoktų šias esmines sąvokas, reikia pateikti daug praktinių užduočių, kurias turi atlikti mokiniai. Mokytojas gali taikyti įvairias priemones: spręsti trumpus uždavinius, komandomis valdyti edukacinius robotus, žaisti lauko ar klasės žaidimus, ir, žinoma, pasitelkti kompiuterio programas ar programėles, vaikams skirtas žaidybines programavimo aplinkas.

Su **komandos** sąvoka jau susipažinome atlikdami įvairias veiklas. Tam, kad galėtume užrašyti algoritmą vykdytoji suprantama kalba, komandą (veiksmą-nurodymą) vaizduojame sutartiniu ženklu (pavyzdžiui, rodykle, programavimo kalbos bloku, žodžiu, garsu ir pan.). Kompiuteriams skirtose programose operuojame komandas vaizduojančiais ženklais, dažniausiai, grafinais (blokai, piktogramos) arba tekstiniais (rašytinis sutartinis žodis arba ženklas). Pradedantiesiems skirtos programavimo aplinkos dažniausiai vaizduoja komandas grafiškai. Komandas atitinkantys sutartiniai ženklai gali būti skirtingų lygių: nuo mažiau formalizuotų (pvz., piešinių, garsų) iki formalių, vienareikšmiškai suprantamų kompiuteriu (programavimo kalbos struktūriniai elementai).

Kalba instrukcijose

Skaitant įvairių instrukcijų aprašymus, pastebėta, kad buvo panaudoti skirtingi kalbos lygiai, kaip pateikta lentelėje. Formuluoiant instrukcijas žmogui, pateikiamos priežastys, išvados ir tolesni paaiškinimai. Kompiuteriui tokių išsamių paaiškinimų nereikia. Lektorius skaidrėse pristato skirtingus instrukcijų kalbos lygius. Instrukcijų kalba gali apimti abu lygius. Iš vienos pusės ji aprašo naudojimo procedūrą, iš kitos – pateikia priežastinius paaiškinimus.

1 lentelė. Ryšys tarp aprašymo lygio ir tikslinio vartotojo

		Žmogus (pvz. pokalbis)	Formalus (naudojimo instrukcija)	Kodas (pvz. kompiuteris)
Aprašymas	Procedūra (kas ir kaip)	Apima abu lygius	Apima abu lygius, bet dėmesys procedūrai	Apima tik procedūros žingsnių aprašymą
Paiškinimai	Priežastys (kodėl)			Tik komentaruose (nebūtina)

Išbandykime programėlę „LightBot“, ją galima įsidiegti į savo išmaniuosius įrenginius arba kompiuterius. Programėlėje robotas valdomas taip, kad pasiektų savo tikslą įvairiomis aplinkybėmis. Robotas programuojamas dėliojant programavimo blokelius. Po pirmojo lygio, kuriame naudojamas tiesinis programavimas, užduotys tampa sunkesnės ir vartotojas turi naudoti procedūras ir ciklus. Programėlę galima atsisiųsti: <https://lightbot.com/hour-of-code.html>

Peržiūrėti pagrindinį tinklalapį su mokymo pavyzdžiais

Atverkime ir peržiūrėkime tinklalapį su mokymo pavyzdžiais, kur naudojama programėlė „LightBot“. Kokios mokomųjų robotų priemonės naudojamos, nėra svarbu, nes studentai turi pateikti kritišką ir konstruktyvų požiūrį į gausius pasiūlymus, susijusius su informatikos mokymu, randamus internete. Pateikite tinklalapio ir jame esančio darbo lapo apžvalgą ir įvertinimą.



Pirmasis programavimo žingsnis – susipažinimas su „LightBot“

Antrąją užduotį sudaro penki pratimai. Pabandykite atlikti pratimus programėle „LightBot“, kad galėtume planuoti programavimo pamokas.

Apgalvokite ir aptarkite

1. Įvertinkite „LightBot“ sudėtingumą.
2. Ar „LightBot“ gali būti naudojamas 5 ar 6 klasės mokinių savarankiškam mokymuisi?
3. Išanalizuokite pateikto darbo lapo tinklalapyje struktūrą.
4. Kokius informatinio mąstymo konceptus ugdo programėlė?
5. Aprašykite savo žodžiais, kas yra „programavimas“.

Veiksmų nuoseklumas

Sudarant algoritmą veiksmai išdėstomi nuosekliai (veiksmų seka), o užrašant algoritmą programavimo kalba, kiekviena komanda vykdoma viena po kitos – tai **veiksmų nuoseklumo** principas, kurį svarbu suprasti atliekant įvairias algoritmų sudarymo ir vykdymo užduotis, paprasčiausių programų kūrimo užduotis naudojant įvairias priemones (su kompiuteriu ar be jo). Ankstesniuose pavyzdžiuose („Gelbėjimo misija“ ir kt. uždavinių pavyzdžiai) jau susidūrėme su veiksmų nuoseklumu.

Mokyti veiksmų nuoseklumo galima nagrinėjant įvairias mokinių daromas užduotis, pvz., kaip padaryti nuotrauką skaitmeniniu įrenginiu, kaip sudaryti veiksmazodžių grandinę rašinėliui rašyti, kaip pagaminti patiekalą, kaip sudėti ir atimti skaičius mintiniu ir rašytiniu būdu, kaip patikrinti sudėtį atimtimi, o atimtį – sudėti.

Pavyzdžiui, „Bebro“ uždavinys „Drabužių krūva“ iliustruoja veiksmų nuoseklumą apsirengimo pavyzdžiu.

Drabužių krūva

Mama bebrė kruopščiai išdėsto savo mažojo sūnaus Beno drabužius į krūvą ant stalo.

Marškinėliai	Apatiniai marškiniai	Kelnės	Apatiniai	Petnešos	Kojinės	Batai
						



Benas drabužius rengiasi, imdamas juos nuo krūvos viršaus iš eilės, bet nenori dėvėti petnešų po marškinėliais.

Kurias drabužių krūvas gali naudoti Benas?

A)


B)


C)

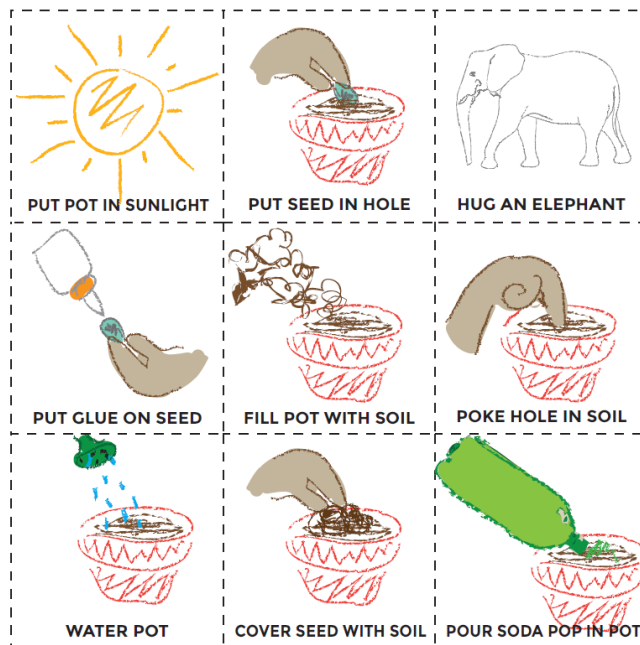

D)


Veiksmų nuoseklumui mokytis su mokiniais galima atlikti algoritmo veiksmų (pavaizduotų grafiškai) sekos dėliojimo užduotis. Vienas iš pavyzdžių pateiktas žemiau.

Sėklos sodinimas

Sudėliokite sėklos sodinimo algoritmą iš „Code.org“ pateiktų paveikslėlių.

<https://code.org/curriculum/course1/6/Activity6-RealLifeAlgorithms.pdf> (Code.org 1 kurso 6 etapas „Algoritmai realiame gyvenime. Sėklos sodinimas“.)



Beje, šiame pavyzdyje yra daugiau veiksmų paveikslėlių negu reikia panaudoti sėklos sodinimo algoritmui užrašyti.

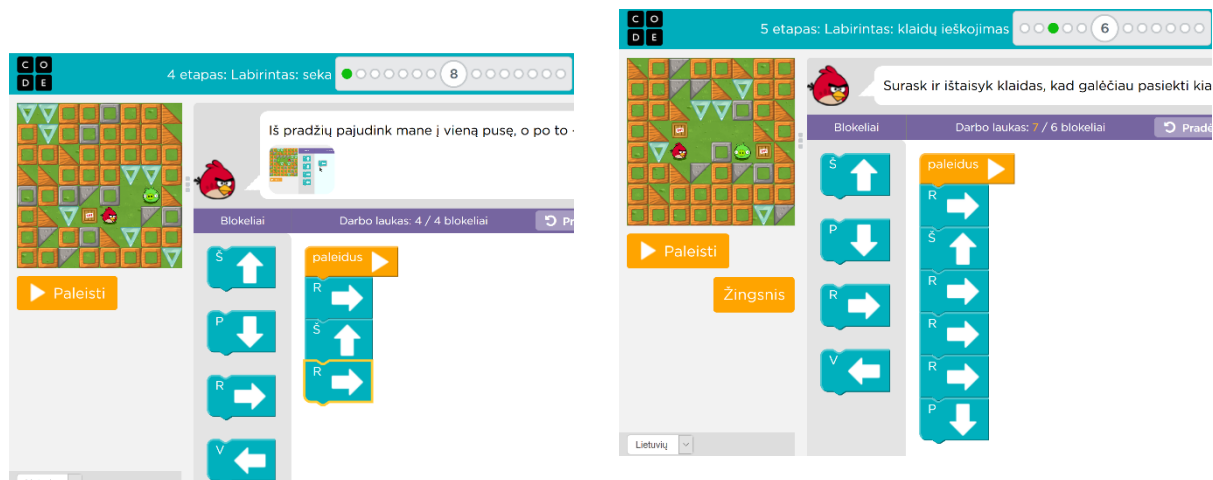
Veiksmų nuoseklumo galima mokytis dėliojant komandų blokus. Pavyzdžiui, „ScratchJr“ veikėjo programoje Pingvinas iš eilės atlieka judėjimo komandas (žydros spalvos blokai): eiti 5 žingsnius į dešinę, pašokti į viršų, eiti 5 žingsnius į kairę.



Veiksmų nuoseklumo (pvz., judėjimo įvairiomis kryptimis) galima mokytis atliekant internetinės priemonės „Code.org“ interaktyvias užduotis.

Atlikite ir aptarkite „Code.org“ 1 kurso (4–6 metai) 4 etapo „Labirinto sekos“ užduotis.

Kuo skiriasi čia naudojamos judėjimo komandos nuo anksčiau naudotų iš „Informatikos be kompiuterio“ užduočių, robotų judėjimo komandų, „ScratchJr“ komandų ir kitų jums žinomų priemonių komandų?



Labirinto sekos užduočių rinkinys: <https://studio.code.org/s/course1/lessons/4/levels/1>

Veiksmų kartojimas

Su veiksmų kartojimu susiduriame dažnai. Pavyzdžiui, gaminant maistą, maišymas – tai to paties judesio kartojimas tam tikrą laiką, šokant, kartojami keletą kartų pagrindiniai judesio elementai.

Vykdant įvairius algoritmus natūraliai gali kilti klausimas, ar galima trumpiau užrašyti komandų seką, pavyzdžiui, tokią (10 kartų iš eilės kartojama komanda „ženklą žingsnį į priekį“):



Ar būtų tas pats, jei susitartume tokius kartojamus veiksmus užrašyti trumpiau, pavyzdžiui:



Kartok 10:

Pavyzdžiui, norint suprogramuoti laiptelių piešinį, reikia kartoti dviejų skirtingų komandų seką („Code.org“ užduočių rinkinio pavyzdys):



Programavimo konstrukcija **kartojamiems veiksams** užrašyti vadinama **ciklu**.

Aukščiau pateiktuose pavyzdžiuose, tokią konstrukciją išreiškėme sutartiniais žodžiais (Kartok 10) arba programavimo kalbos blokais.

Žaidybinėse programavimo aplinkose dažniausiai būna dviejų rūšių ciklai:

- **Žinomo kartojimo skaičiaus ciklas** – programavimo konstrukcija veiksmų sekai kartoti žinomą kartų skaičių. Aukščiau pateikti pavyzdžiai yra žinomo kartojimo skaičiaus ciklai (kartojami 10 ir 4 kartus).
- **Begalinis ciklais** – niekada nesibaigiantis ciklas. T. y. vykstantis tol, kol vykdoma programa. Pavyzdys – groti pastoviai tą pačią melodiją, suktis visą laiką, kol vykdoma programa.

Ciklų (veiksmų kartojimo) mokytis galima nagrinėjant įvairius su kasdieniu gyvenimu pavyzdžius, atliekant programavimo be kompiuterio užduotis, integruojant su kitų dalykų turiniu, pvz., muzikinis ritmas ir jo kartojimas, šokio žingsnių kartojimas, programuojant specialiose programavimo aplinkose, valdant edukacinius robotus.

„Code.org“ pateikia pavyzdį ciklams mokytis be kompiuterį – kartojant sutartinius šokio elementus, pateiktus sutartiniais paveikslais

(<https://code.org/curriculum/course2/5/Activity5-GettingLoopy.pdf>).

„ScratchJr“ aplinkoje žinomo kartojimo skaičiaus ir begalinis ciklas atitinkamai žymimi taip:



Analogiškai ciklai žymimi ir toliau nagrinėjamoje „X-LOGO“ programavimo aplinkoje (žinomo kartojimų skaičiaus ciklas):



Veiksmų šakojimas

Kasdieniam gyvenime nuolat susiduriame su pasirinkimais, kurie priklauso nuo tam tikros sąlygos tenkinimo. Pavyzdžiui: **JEI** bus šilta, **TAI** eisiu pasivaikščioti; **JEI** draugas sutiks, **TAI** darbą atliksime kartu; **JEI** vanduo užvirė, **TAI** reikia įdėti kruopų; **JEI** dega žalia šviesoforo šviesa, **TAI** galiu eiti per sankryžą. Tai konstrukcijos **JEI – TAI**. Tokios konstrukcijos programavime vadinamos **veiksmų šakojimu**. Jei sąlyga tenkinama, vykdoma viena veiksmų seka, priešingu atveju gali būti vykdoma kita seka.

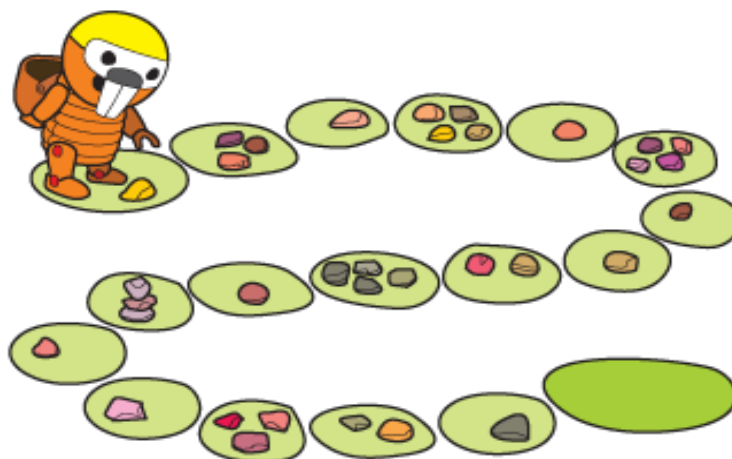
Išnagrinėkite žemiau pateiktą „Bebro“ uždavinį, kuriame taikomas veiksmų šakojimas.

Akmenukai

JEI laukelyje vienas akmenukas, **TAI** žengti į kitą laukelį,

KITAIP – paimti vieną akmenuką ir žengti per tiek laukelių, kiek liko akmenukų.

Kiek Bebras surinks akmenukų? Ar pasieks tikslą?



Veiksmų šakojimas programavimo kalbose užrašomas programavimo kalbos valdymo konstrukcija – **sąlyginiu sakiniu**. Sąlyginis sakiny apibrėžia sąlygą ir veiksmus, kurie turi būti atliekami atsižvelgiant į tai, ar sąlyga tenkinama, ar ne.

Žemiau patiekama sąlyginio sakinio konstrukcija, išreikšta bloku „Scratch“ aplinkoje ir pavyzdys (jei veikėjas liečia objektą žalios spalvos, jis pasisuka 90 laipsnių, eina 10 žingsnių, priešingu atveju keičia savo kaukę (išvaizdą)).



Su veiksmų šakojimu yra glaudžiai susijusi **įvykio** sąvoka.

Kai kuriose programavimo aplinkose vaikams, pvz., „ScratchJr“ aplinkoje, sąlyginių sakinių nerasime, kadangi aplinka pritaikyta jaunesniojo amžiaus mokiniams, o norint užrašyti daugumą sąlygų, reikia žinoti kintamojo ir jo reikšmės sąvokas. Todėl čia sąlyginius sakinius keičia įvykių programavimas.

Įvykis – veiksmas, į kurį programa turi reaguoti, pavyzdžiui:

- klavišo paspaudimas,
- pelės patraukimas,
- duomenų gavimas jutikliu,
- kitų programų pranešimai,
- programos signalas apie klaidą.

Iš esmės, įvykius galime traktuoti kaip sąlygas. Aukščiau pateiktos įvykio sąvokos apibrėžtyje minimus įvykių pavyzdžius galima suprasti kaip sąlygas:

„jei paspaustas klavišas, tai...“,

„jei patraukta pelė, tai...“,

„jei jutiklis gavo duomenų, tai...“,

„jei kita programa pateikė pranešimą, tai...“,

„jei programa pateikė signalą apie klaidą, tai...“.

Veiksmų šakojimo (įvykių) galima pradėti mokytis žaidžiant įvairius žaidimus (be kompiuterio), o įsisavinus šį konstruktorius pereiti prie jo formalaus užrašo programavimo aplinkose.

Svarbu atkreipti mokinių dėmesį į tai, kad apibūdinant ar vaizduojant algoritmus, nurodymai turi būti konkretūs ir aiškūs algoritmo vykdytojui.

„ScratchJr“

Vizualiojo programavimo aplinka „ScratchJr“ skirta vaikams nuo 5–6 metų. Šioje aplinkoje vaikai gali kurti skaitmeninius objektus, animacijas, interaktyvias istorijas programuodami. Taip ugdomi vaikų „projektuotojo“ gebėjimai.

„ScratchJr“ programavimo kalba yra vaizdinė, komandos dėliojamos iš spalvingų blokų.

Naudodami „ScratchJr“ priemonę, vaikai išmoksta programavimo, planavimo, kompiuterio valdymo pradmenų, mokosi spręsti įvairius uždavinius, kūrybiškai išreikšti save. Priemonėje esantys veikėjai, fonai skatina vaikus kurti pasakojimus, animacijas, o integruota piešimo priemonė leidžia piešti naujus veikėjus, fonus ar koreguoti esamus. „ScratchJr“ programavimo kalba pritaikyta jaunesniojo amžiaus vaikams: programavimo konstrukcijos reiškiamos paveikslėliais, todėl net dar neišmokę rašyti vaikai gali kurti nesudėtingas programas. Integruota garso įrašymo priemonė leidžia vaikams įrašyti savo balsą ir taip įgarsinti projekto veikėjų pokalbius. Vaikai, gebantys rašyti tekstus, gali kurti sudėtingesnius pasakojimus, interaktyvius tekstinius dialogus. Sukurtus darbelius vaikai gali įrašyti naudojamame įrenginyje arba jais pasidalinti su kitais (pvz., išsiųsti el. paštu tiesiogiai iš „ScratchJr“ aplinkos). Priemonė teikia plačias galimybes integruoti informatikos konstruktus į įvairius dalykus.

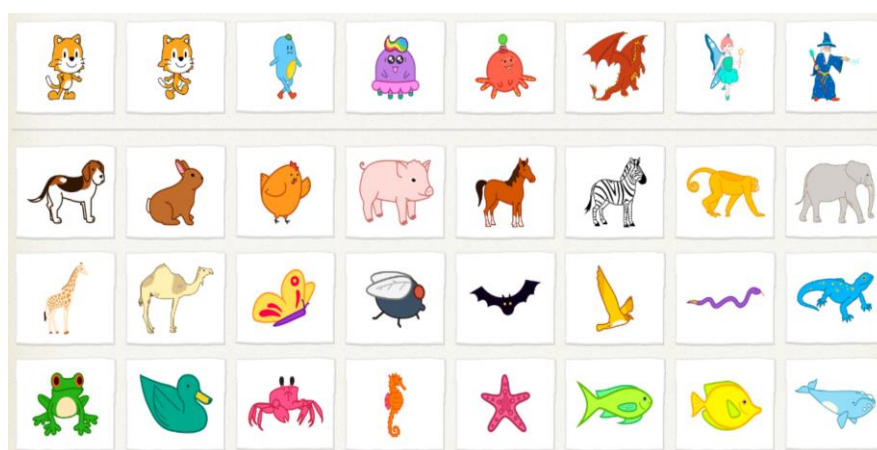
„ScratchJr“ aplinkoje vaikai nurodymą (komandą) pateikia ne tiesiogiai kompiuteriui, o veikėjui, kuris tą komandą atlieka. Komandos reiškiamos „ScratchJr“ blokais. Komandų blokų rinkinys sudaro programą, dar vadinamą veikimo scenarijumi. Iš visų veikėjų scenarijų (programų) visumos susideda viso projekto (tam tikrai temai skirto darbo) programa.

Vaikai gali daryti garso įrašus, fotografuoti ir įtraukti garsus ar nuotraukas į savo kuriamas programas. Visa tai leidžia realiu laiku įrašyti įspūdžius, pavyzdžiui, mokomosios išvykos metu.

Pradėjus naują „ScratchJr“ projektą, reikėtų pasirinkti, kas vykdys komandas (kuris veikėjas ar veikėjai), ir kur vyks programuojami veiksmai (veiksmų laukas).

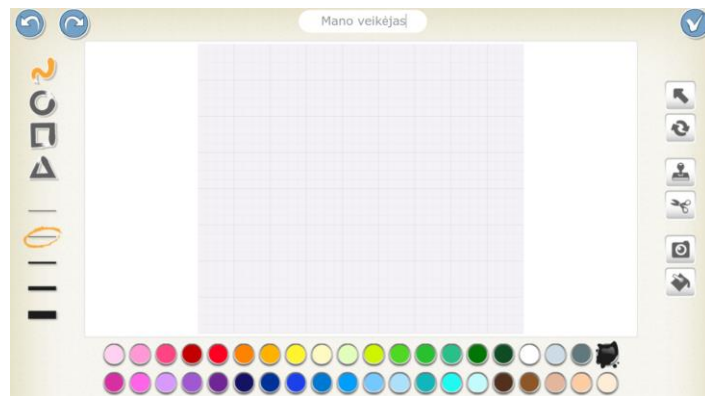


„ScratchJr“ aplinkoje yra didelis pasirinkimas veikėjų ir veiksmų laukų piešinių (paveiksle pateiktas siūlomų veikėjų rinkinio fragmentas), tačiau daugeliui vaikų darbų kūrybiškai idėjų išraiškai prireiks kitokių veikėjų ar veiksmų laukų, kuriuos galima nupiešti arba adaptuoti pakoregavus esamus veikėjus ar fonus.



Jei norima pasirinkti veikėją tokį, koks jis siūlomas „ScratchJr“ aplinkoje, spaudžiamas varnelės mygtukas, jei norma jį koreguoti, pažymimas veikėjas ir pasirenkamas teptuko mygtukas, jei norima atsisakyti veikėjo pasirinkimo, spaudžiamas kryžiukas. Jei norima pašalinti esamą veikėją iš veiksmų lauko (pvz., numatytąjį veikėją Katiną), reikia jį spustelėti ir palaikyti, ir paspausti atsiradusį raudoną kryžiuką.

Pasirinkus redaguoti arba piešti naują veikėją, naudojama integruota „ScratchJr“ piešimo priemonė.



Viršuje yra trys mygtukai: kairėje – atšaukti paskutinį veiksmą, grąžinti paskutinį atšauktą veiksmą, dešinėje – patvirtinti visus veiksmus ir įrašyti piešinį.

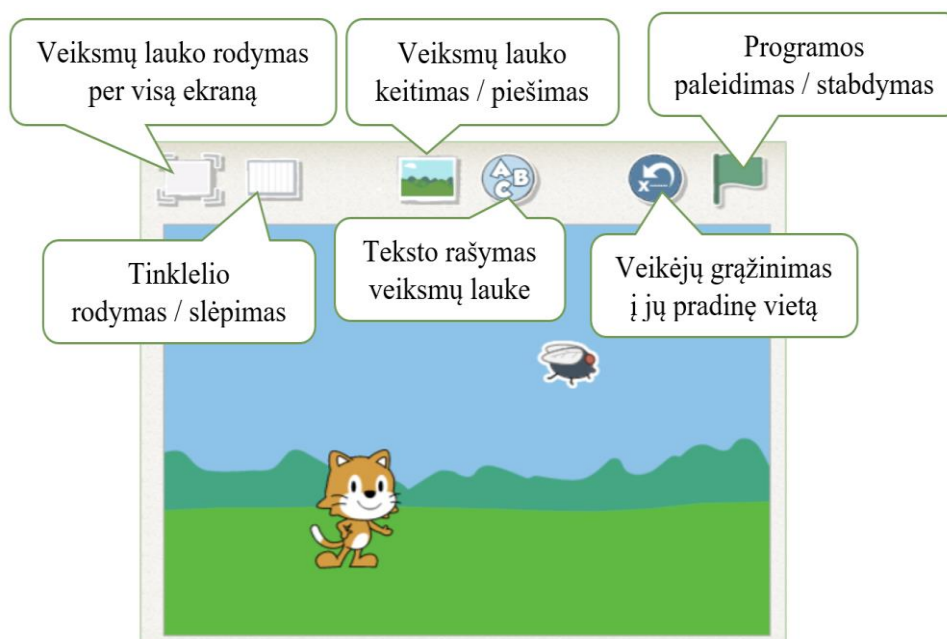
Kairėje vertikalčiai išdėstyti piešimo objektų įrankiai (linija, apskritimas, stačiakampis, trikampis) ir jų linijų storių pasirinkimo mygtukai, apačioje – spalvų paletė, o dešinėje vertikalčiai – veiksmų su piešiamais objektais įrankiai. Trumpai juos apibūdinsime iš eilės, iš viršaus žemyn:

	Objekto kontūro taisymo ir objekto tempimo priemonė. Pasirenkama priemonė, o tada objektas, kurio padėtį piešimo lauke arba kontūrą koreguojant taškus norima keisti.
	Objekto posūkio priemonė. Pasirenkama priemonė, o tada objektas, kurį norima pasukti.
	Kopijavimo priemonė. Pasirinkus kopijavimo priemonę ir pažymėjus objektą, sukurama jo kopija.
	Iškirpimo priemonė. Pažymėjus mygtuką su nupaišytomis žirklėmis, o tada pažymėjus objektą, jis pašalinamas iš piešimo lauko.
	Integruota fotografavimo priemonė. Pasirinkus fotografavimo priemonę ir spustelėjus bet kurią piešinio sritį, fotografuojama, o pasirinkta sritis užpildoma padaryta nuotrauka. Gali būti naudojama praturtinant fonus nuotraukomis ir kuriant individualizuotus veikėjus, pavyzdžiui, veikėjus, kurių veidas – vaiko veido nuotrauka.
	Spalvinimo priemonė. Pasirinkus priemonę ir spustelėjus piešinio sritį, ta sritis nuspalvinama paletėje pažymėta spalva.

Integruota fotografavimo priemonė leidžia individualizuoti veikėjus. Pažymėta piešinio sritis užpildoma nufotografuotu vaizdu. Atkreipsime dėmesį, kad veikėjo ar veiksmų lauko piešinį

negalima papildyti nuotrauka ar paveikslu iš failo. Yra galimybė įkelti tik realiu laiku, piešimo priemonėje esant padarytą nuotrauką.

Veikėjų programuojami veiksmai atliekami veiksmų lauke.



Veiksmų lauką (foną) galima pasirinkti iš esamo rinkinio, nupiešti naują arba pakoreguoti esamą. Naudojama ta pati piešimo priemonė, kurią ką tik aptarėme. Fonui kurti ji paleidžiama pasirinkus fono piktogramą viršutinėje „ScratchJr“ mygtukų juostoje, pažymėjus tuščią foną ir pasirinkus piešimo mygtuką su pavaizduotu teptuku.

„ScratchJr“ blokų apžvalga

„ScratchJr“ programavimo kalbos blokai suskirstyti į šešias dalis ir, kad juos būtų lengviau atpažinti, nuspalvinti skirtingomis spalvomis. Kuriant tam tikrą projektą, paprastai naudojami įvairių grupių veiksmai.

Blokų grupė	Aprašas
Veiksmų perjungimas ir aktyvavimas 	Geltonos spalvos blokai, skirti paleisti parašytą programą, valdyti veikėjų sąveiką.
Judesiai 	Žydros spalvos blokai, skirti veikėjo judesiams (į kairę, į dešinę, į viršų, apsisukimą, grįžimą į pradinę vietą ir pan.) programuoti.

Išvaizda 	Purpurinės spalvos blokai, skirti veikėjų išvaizdai keisti (pasislėpti, pasirodyti, keisti dydį, rodyti teksto burbulą).
Garsai 	Žalios spalvos blokai, skirti garsui įrašyti, įrašytam ir numatytajam garsui groti.
Valdymas 	Oranžinės spalvos blokai, skirti veiksmams kartoti, stabdyti, atidėti, keisti veiksmų atlikimo spartą.
Baigimas 	Raudonos spalvos blokai, kuriais nurodoma, kas vyksta programos atlikimo pabaigoje: baigiama, visada kartojama ar pereinama į naują puslapį – veiksmų lauką.

Keletas užduočių

1. Pasirinkite ir pakoreguokite vieną iš esamų veikėjų: pakeiskite spalvas, pridėkite naujų detalių, į pasirinktą sritį įdėkite nuotrauką ir pan. Nupieškite šiam veikėjui skirtą veiksmų lauką, panaudodami kuo daugiau įvairių piešimo priemonių.
2. Kuo skiriasi veikėjas nuo objekto, nupiešto veiksmų lauko piešinyje?
3. Parsisiųskite blokų piešinius spausdinimui iš „ScratchJr“ kūrėjų svetainės: <http://scratchjr.org/pdfs/blocks.pdf>. Parenkite popierinių blokų ruošinius mokinių pirmiesiems žingsniams su „ScratchJr“.
4. Parenkite keletą paprastų komandų sekų ugdomajai veiklai „Atspėk komandas“ atlikti (https://informatika.ugdome.lt/wp-content/uploads/2018/08/07-Scratchjr_1-2.pdf). Išbandykite veiklą praktiškai (žr. siūlomą veiklos eigą).

Patarimas. Pradėti algoritmavimo ir programavimo mokymąsi su „ScratchJr“ rekomenduotina po to, kai mokiniai be kompiuterio išmoko vykdyti paprastas komandas. Rekomenduotina pradžia – su „ScratchJr“ atspausdintais (popieriniais) blokais, prieš tai išsiaiškinus judėjimo krypčių komandų skirtumus nuo anksčiau naudotų. Pavyzdžiui, galima pradėti nuo gerųjų patirčių skyriuje aprašyto veiklos scenarijaus „Atspėk komandas“. Reikia turėti atspausdintus komandų blokus arba jų poaibį.

Šaltiniai

„Scratch“ – blokinio programavimo aplinka. <https://scratch.mit.edu/>

„XLogo“ – blokinio programavimo aplinka su pratimais.

<https://xlogo.inf.ethz.ch/release/latest/#/>

Bell, T., Witten, I. H., Fellows, M. (2015). Informatika be kompiuterio. Vertė į lietuvių kalbą V. Dagienė ir E. Jasutė. Prieiga per internetą: <https://informatika.ugdome.lt/wp-content/uploads/2017/05/KNYGA-Informatika-be-kompiuterio-2015-09-03.pdf>

Code.org – programavimo mokymosi aplinka. <https://code.org/>

Kalas I., Blaho A., Moravcik M. (2018) Exploring Control in Early Computing Education. In: Pozdniakov S., Dagienė V. (eds) Informatics in Schools. Fundamentals of Computer Science and Software Engineering. ISSEP 2018. Lecture Notes in Computer Science, vol 11169. Springer, Cham

LightBot – programėlė algoritmavimui mokytis. <https://lightbot.com/hour-of-code.html>

Papertas S. (1995). Minčių audros. Vaikai, kompiuteriai ir veiksmingos idėjos. Vilnius „Žara“ 1995 m.