

Co-funded by the
Erasmus+ Programme
of the European Union



2 modulis

Bendras įvadas į informatinį mąstymą

Pagrindinis modulis,
tinkamas visiems mokytojams

Autoriai: Radboud University (Olandija)

Maria Kallia,
Sjaak Smetsers,
Erik Barendsen,
Christos Chytas

Recenzantai:

Arnold Pears (KTH),
Valentina Dagienė (VU)

Išoriniai recenzantai:

Piret Luik (Estija),
Renate Motschnig (Austrija)

Pilotavimas:

Ankara University (Turkija), KTH Royal Institute of Technology (Švedija), Radboud University (Olandija), University of Paderborn (Vokietija), Vienna University of Technology (Austrija)

Vertimas: Vilniaus universitetas

Prof. Dr. Valentina Dagienė, Dr. Eglė Jasutė

Dizainas (piktogramos):

Vaidotas Kinčius (Lietuva)

Modulis sukurtas įgyvendinant Erasmus+ KA2 projektą „Būsimų mokytojų rengimas: Informatinis mąstymas ir STEM“ (Future Teachers Education: Computational Thinking and STEAM - TeaEdu4CT). Projektą koordinuoja Prof. Valentina Dagienė, Vilniaus universitetas. Partneriai: Vienos technologijų universitetas (Austrija), CARDET (Kipras), Talino universitetas (Estija), Turku universitetas (Suomija), Paderborno universitetas (Vokietija), CESIE (Italija), Radboudo universitetas (Nyderlandai), KTH technologijų karališkasis institutas (Švedija), Ankara universitetas (Turkija).

© „TeaEdu4CT“ projektas (nr. 2019-1-LT01-KA203-060767), „Creative Commons“ licencija (CC BY-4.0), 2019–2021



Turinys

	Bendra apžvalga ir tikslas	3
	Tikslinės grupės ir išankstinės sąlygos	4
	Mokymosi rezultatai ir vertinimo metodai	4
	Modulio planas ir didaktiniai metodai	4
	Dalys ir veiklos	4
	1 dalis. Įvadas į informatinį mąstymą	5
	2 dalis. Informatinio mąstymo priemonės	8
	3 dalis: Patirkite kompiuterinį mąstymą programuodami	24
	4 dalis: informatinio mąstymo mokymas ir mokymasis	39
	O2 modulio mokymosi išteklių apžvalga	46



Bendra apžvalga ir tikslas

Šio modulio tikslas – suteikti būsimiems mokytojams konkretų supratimą apie informatinį mąstymą (IM), praktinių žinių apie IM mokymo ir mokymosi principus ir supažindinti su priemonėmis, kurios gali padėti mokyti ir mokytis.

Šiame modulyje besimokantieji:

- išnagrinės informatinio mąstymo sąvoką ir palygins įvairius būdus, kaip ją apibūdinti, atsižvelgiant į problemų sprendimo veiklą.
- susipažins su informatinio mąstymo aspektais, praktika ir priemonėmis bei supras informatinio mąstymo vaidmenį mokomuosiuose dalykuose.
- susipažins su programavimo pagrindais ir praktiškai išbandys algoritminį mąstymą pasakojimo ir žaidimų kontekste.
- išbandys veiklas su kompiuteriu ir be kompiuterio, susipažins su informatinio mąstymo mokymo strategijų kūrimo principais.

Šiame modulyje IM vertinamas kaip tarpdalykinių įgūdžių ir kompetencijų ugdymo sistema, tinkama bet kurio dalyko mokytojiui.

Modulio struktūra

Šiuo tikslu modulis suskirstytas į keturias dalis: 1 dalyje pristatoma IM sąvoka ir lyginami būdai, kaip ją apibūdinti problemų sprendimo veikloje; 2 dalyje IM pristatoma naudojant įvairias skaitmenines priemones (Ngrams, NetLogo, Excel) ir parodoma, kaip šias priemones galima panaudoti sprendžiant įvairių dalykų (istorijos, biologijos, geografijos) problemas; 3 dalyje daugiausia dėmesio skiriama programavimo pagrindams ir algoritminio mąstymo praktikai pasakojimo ir žaidimų kūrimo su Scratch kontekste. Modulį užbaigia 4 dalis, kurioje aptariamos mokymosi veiklos su kompiuteriu ir be kompiuterio, IM mokymo strategijų ir vertinimo elementai.

1 dalis: Įvadas į informatinį mąstymą bei jo apibūdinimas problemų sprendimo kontekste

2 dalis: Įvadas į informatinį mąstymą naudojant įvairias skaitmenines priemones įvairių dalykų problemoms spręsti

3 dalis: Pagrindiniai programavimo ir algoritminio mąstymo aspektai pasakose ir žaidimų su Scratch kūrime

4 dalis: IM veiklų su kompiuteriu ir be kompiuterio mokymo strategijos ir vertinimas



Tikslinės grupės ir išankstinės sąlygos

Šis modulis skirtas būsimiems mokytojams, besimokantiems pagal mokytojų rengimo programą, taip pat dirbančių mokytojų profesiniam tobulėjimui. Modulis yra skirtas kontaktiniam mokymuisi, tačiau jį galima lengvai pritaikyti kaip nuotolinio mokymosi modulį.

Šiam moduliui studijuoti nėra jokių konkrečių išankstinių sąlygų. Patartina, kad studentai turėtų pagrindinių žinių apie naudojamas priemones ir būtų išklause ankstesnį modulį „01: Bendrasis modulis: informatinis mąstymas ir STEAM būsimiems mokytojams“.



Mokymosi rezultatai ir vertinimo metodai

Sėkmingai baigęs visą modulį studentas gebės:

- paaiškinti informatinio mąstymo sąvoką ir palyginti įvairius būdus, kaip ją apibūdinti problemų sprendimo veikloje;
- suprasti problemų sprendimo uždavinius, taikant informatinio mąstymo sąvokas, praktiką ir priemones, ir paaiškinti IM vaidmenį dalykuose;
- kurti paprastas programas Scratch ir taikyti algoritminį mąstymą pasakojimo ir žaidimų kontekste;
- taikyti mokymosi su kompiuteriu ir be kompiuterio strategijas ir veiklas gilinant IM, apibūdinti pagrindinius strategijų kūrimo principus.

Vertinimas

Visuose moduluose atliekamos užduotys suteikia galimybę atlikti formuojamąjį vertinimą ir gauti grįžtamąjį ryšį. Yra neprivalomas galutinis (bendras) vertinimas, kurį galima rasti atskirame dokumente (žr. mokymosi išteklius: Galutinio vertinimo modulis).



Modulio planas ir didaktiniai metodai

Modulį sudaro 4 dalys tiesioginiam mokymuisi. Kiekvieną modulį sudaro kelios užduotys, kurios paprastai prasideda apšilimo užduotimi ir baigiasi refleksijos užduotimi. Studentai taiko įvairius mokymosi metodus, įskaitant straipsnių skaitymą arba literatūros apžvalgas, grupines diskusijas, problemų sprendimo užduotis ir apmąstymų veiklas.



Dalys ir veiklos

1 dalis. Įvadas į informatinį mąstymą

1.1 veikla. Įvadas į informatinį mąstymą

- Informatinis mąstymas kaip problemų sprendimo veikla
- Informatinio mąstymo elementų apibūdinimas
- Išvada

Iš viso: 2 valandos

2 dalis. Informatinio mąstymo lavinimo priemonės

2.1 veikla. Naudojimas „Google Ngrams“

2.2 veikla. Modeliavimas ir imitavimas naudojant „NetLogo“

2.3 veikla. „Microsoft Excel“ naudojimas

Iš viso valandų: 8 valandos

3 dalis. Patirtis informatinio mąstymo srityje per programavimą

3.1 veikla. Istorijų pasakojimas „Scratch“ programa

3.2 veikla. Labirinto žaidimo kūrimas

Iš viso valandų: 10 valandų

4 dalis. Informatinio mąstymo mokymo ir mokymosi strategijos

4.1 veikla. Bebro užduotys



1 dalis. Įvadas į informtinį mąstymą

Šiame skyriuje nagrinėsite informatinio mąstymo sąvoką ir palyginsite įvairius būdus, kaip ją apibūdinti problemų sprendimo veiklose.



Indėlis į mokymosi rezultatus

Mokymosi rezultatai

- gebėti apibūdinti informatinį mąstymą kaip jungiamąjį dalykinės srities ir IT mechanizmą.
- apibūdinti informatinį mąstymą kaip problemos sprendimo etapus ir veiksmus, kuriais pasiekiamas optimalus rezultatas, įgyvendinamas sprendimas.
- atpažinti informatinio mąstymo elementus mokinių veiklos scenarijuose.



1.1 veikla. Įvadas į kompiuterinį mąstymą

Norint išnaudoti visas kompiuterių galimybes įvairiuose dalykuose ir veiklose, reikia daugiau skaitmeninių įgūdžių nei mokėti dirbti su „Word“ ar tvarkyti socialinės žiniasklaidos paskyrą. Tokiam dalyko ir informacinių technologijų ryšiui užmegzti reikia specifinių problemų sprendimo įgūdžių. Šie įgūdžiai priklauso informatinio mąstymo sričiai

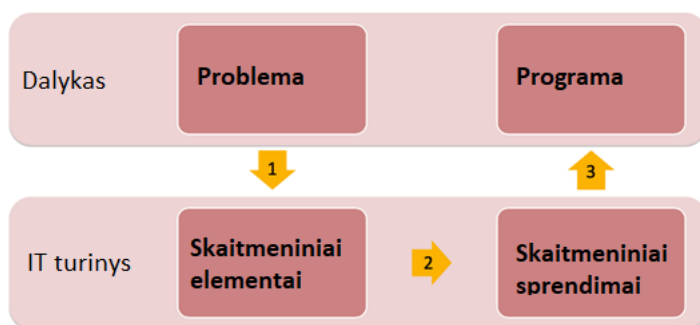


Informatinis mąstymas kaip problemų sprendimo veikla



Skaitymo užduotis

Perskaitykite dokumentą „Trumpas įvadas į kompiuterinį mąstymą“, kurį rasite mokymosi ištekliuose. Šiame tekste išskiriami trys pagrindiniai informatinio mąstymo etapai: schemeje rodyklės (1), (2) ir (3).



Diskusija porose

Aptarkite toliau pateiktus du klasės scenarijus, paimtus iš Yadav et al. (2018, p. 381). Kurią veiklą laikytumėte informatiniu mąstymu? Kaip jos susijusios su A medžiagoje pateiktais (1), (2) ir (3) žingsniais?

1 scenarijus:

Kitus mokslo metus Vestvudo pradinė mokykla pradės nuo „iPad“ 1:1 iniciatyvos. Ponas Novakas nusprendė, kad jo 2 klasės mokiniai savaitę naudosis savo „iPad“ ir prognozuos orus (temperatūrą, kritulius ir vėją). Kiekvienas mokinys nupiešia paveikslėlį, kaip, jo manymu, atrodys orai. Mokinė Sara taip pat norėjo sekti, kokią temperatūrą visi prognozuoja. Ponas Novakas sukūrė „Google“ dokumete lentelę, į kurią kiekvienas mokinys įvedė savo prognozuojamą temperatūrą. Kitą dieną jie užfiksavo tikruosius orus naudodami „Accuweather“ programėlę savo „iPad“ ir įvedė informaciją į „Google“ dokumento lentelę. Olivija taip pat įrašė tikrąją temperatūrą Saros lentelėje, kad galėtų palyginti, kaip jų prognozės atitiko tikrąjį orą. Po savaitės jie projektavo „Google“ skaičiuoklę išmaniojoje lentoje ir atėmė skirtumus tarp stebėtos ir prognozuotos temperatūros. Ponas Novakas parodė, kaip sudaryti šių skirtumų stulpelinę diagramą.

2 scenarijus:

Visos antros klasės važiuoja į ekskursiją! Mokyklos valgykloje visiems supakuoti pietūs iš PB&J į vienodus popierinius maišelius, išskyrus Sarą ir Oliviją, kurios yra alergiškos žemės riešutams. Popieriniai pietų maišeliai paženklinėti visų mokinių vardais ir suskirstyti į 10 dėžučių po 10 pakuočių dėžutėje. Pietūs buvo sudėti į dėžutes abėcėlės tvarka pagal pavardes. Ponas Novakas nori patikrinti, ar Sara ir Olivija gauna pietus be žemės riešutų. Jos padeda jam ieškoti dėžutėse. Olivija Velazquez žino, kad jos pietūs tikriausiai bus netoli pabaigos, todėl ji žiūri į pirmuosius pietus kiekvienoje dėžėje, kol randa tokius, kurie prasideda raide, artima abėcėlės pabaigai. Kai ji randa dėžutę, kuri prasideda Jemal Summer, pažiūri į paskutinius pietus toje dėžutėje. Tai Bilio Vagnerio pietūs, todėl ji žino, kad jie turi būti arti! Ji pažvelgia į pietus, esančius visai šalia Bilio, ir jie yra jos. Laimei, ji mato, kad valgykla nepamiršo jai supakuoti sumuštinio su sūriu ir morkų.



Informacinio mąstymo elementų apibūdinimas

Informacinių problemų sprendimo etapai:

1. dekontekstualizavimas: dalykinės srities problemos ar klausimo (konteksto) vertimas į kompiuterinius terminus;
2. kompiuterinių problemų sprendimas: vykdytino sprendimo kūrimas;
3. (pakartotinis) kontekstualizavimas: sprendimo perkėlimas atgal į dalykinę sritį.

Kompiuterinio mąstymo apibrėžimai skiriasi tuo, kaip jose pabrėžiama veikla, atliekama 1, 2 ir 3 etapuose. Pavyzdžiui, Selby ir Woollard (2013, p. 5) informacinį mąstymą apibūdina kaip „veiklą, dažnai orientuotą į produktą, susijusią su problemų sprendimu, bet neapsiribojančią vien tik juo. Tai pažintinis arba mąstymo procesas, atspindintis

- gebėjimą mąstyti abstrakcijomis,
- gebėjimą skaityti problemą,
- gebėjimą mąstyti algoritmiškai,
- gebėjimą vertinti,
- gebėjimą apibendrinti.“

Nors kai kurie elementai iš dalies sutampa, pirmiau minėtus elementus galime bendrai susieti su mūsų modelio etapais. Pirmieji du elementai daugiausia susiję su 1 žingsniu: problemos ar situacijos dėsningumą analizė (abstrakcija) ir problemos skaidymas į mažesnes (dekompozicija). Algoritminis mąstymas dažniausiai naudojamas 2 žingsnyje, o sprendimo vertinimas ir tyrimas, kaip jį galima apibendrinti, susieja kompiuterinį sprendimą su dalykine sritimi, o tai vyksta 3 žingsnyje. Tai galime apibendrinti lentelėje:

	(1)	(2)	(3)
Selby ir Woollard (2013)	abstrakcija skaidymas	algoritminis mąstymas	vertinimas apibendrinimas



Užduotis porose

Dokumentas „B. Definitions of CT“. Papildykite pirmiau pateiktą lentelę dar trimis eilutėmis. Suklasifikuokite elementus, kuriuos rasite trijose skiltyse. Ar atpažįstate kokius nors anksčiau pateiktų 1 ir 2 klasės scenarijų elementus?



Išvada



Bendra diskusija

Palyginkite savo išvadas su kitų besimokančiųjų. Kartu sukurkite trumpą darbinį informatinio mąstymo apibrėžimą, apimančią etapus ir veiklas.



Mokymosi ištekliai

A. Trumpas įvadas į kompiuterinę tomografiją

B. CT apibrėžtys



Nuorodos

Selby, C. & Woollard, J. (2013) Computational thinking: the developing definition, prieiga per internetą: <http://eprints.soton.ac.uk/356481>, žiūrėta: 2021 m. vasario 10 d.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Kompiuterinis mąstymas pradinėse klasėse: mokytojų supratimo apie gamtos mokslų mokymui skirtas kompiuterines idėjas matavimas. *Computer Science Education*, 28(4), 371-400.



2 dalis. Informatinio mąstymo priemonės

Šioje dalyje įgysite praktinės informatinio mąstymo mokymosi patirties, naudodami paprastas skaitmenines priemones, skirtas realioms problemoms spręsti įvairiuose dalykuose ir nereikalaujančias papildomo programavimo.

Šioje dalyje pristatomos trys priemonės: NetLogo, Microsoft Excel ir Google Ngrams, ir atliekamos veiklos, parodančios, kaip šios priemonės gali būti naudojamos informatiniam mąstymui įtraukti į įvairius mokyklinius dalykus. Išsamiai aprašoma kiekviena priemonė ir jos taikymas įvairiuose dalykuose. Atlikdami šią veiklą susipažinsite su informatinio mąstymo aspektais, praktika ir priemonėmis bei suprasite informatinio mąstymo vaidmenį įvairiuose dalykuose.



Indėlis į mokymosi rezultatus

Mokymosi rezultatai

- Ugdyti informatinį mąstymą naudojant scenarijų pavyzdžius.
- Klasifikuoti problemas, išsprendžiamas skaitmeniniu būdu
- Naudoti skaitmenines priemones problemoms spręsti.
- Projektuoti išsamius struktūruotus problemų sprendimus
- Mąstyti apie duomenis, juos interpretuoti ir vizualizuoti
- Naudokite duomenų analizę geresniam gamtinių sistemų supratimui
- Įvertinti, kokias problemas galima išspręsti naudojant modeliavimą ir imitavimą
- Suprasti ir apibūdinti, kaip modeliavimas ir imitavimas gali būti naudojami problemai spręsti
- Analizuoti duomenis ir projektuoti modelius modeliuojant ir imituojant
- Bendradarbiauti ir bendrauti su kolegomis sprendžiant problemą



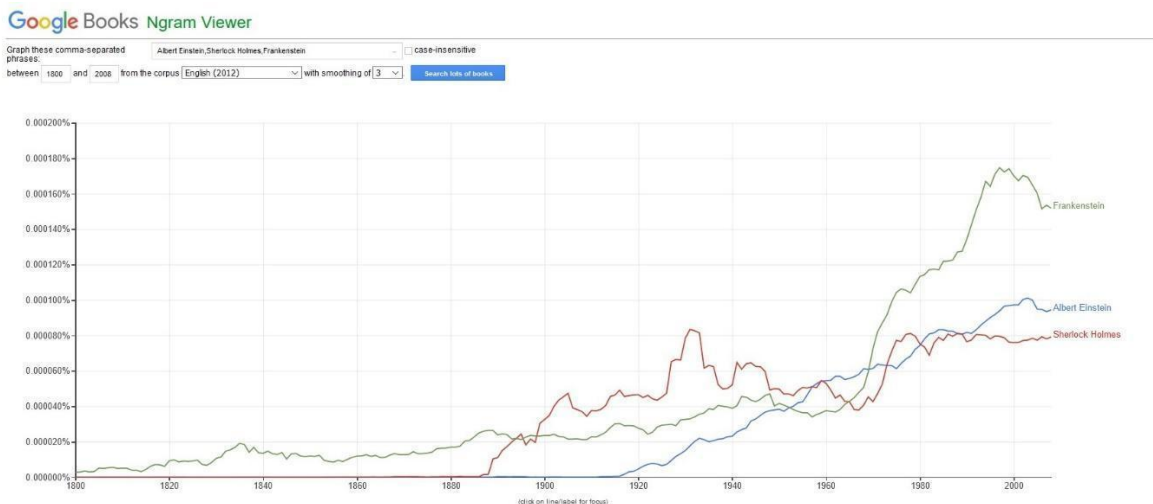
2.1 veikla: „Google Ngrams“ naudojimas

„Google Ngram“ naršyklė – tai grafinis įrankis, kuriame vaizduojami žodžių dažniai iš didelio knygų korpuso, parodantis termino ar frazės vartojimo kitimą laikui bėgant. Įrankis naudoja literatūros šaltinius, išspausdintus 1500-2008 m. amerikiečių anglų, britų anglų, prancūzų, vokiečių, italų, ispanų, rusų, hebrajų ir kinų kalbomis, todėl jį galima naudoti tiriant kalbos pokyčius bėgant metams. Kalbos vartojimo svyravimai gali parodyti socialinius-kultūrinius pokyčius ir taip padėti suprasti, kaip įvairios socialinės-kultūrinės transformacijos vystosi istorijos eigoje. Šioje užduotyje naudodamiesi „Google Ngrams“ nagrinėsime kultūrinius ir socialinius pokyčius laikui bėgant, kuriuos atspindi konkrečių terminų vartojimas knygose.



Diskusija

Prieš pradėdami dirbti su šia užduotimi, spustelėkite šią nuorodą, kad atvertumėte „Google Ngram“ naršyklę (<https://books.google.com/ngrams>). Įrankis atveriamas su pavyzdžiu, kuris pavaizduotas toliau pateiktame paveikslėlyje. Šiame pavyzdyje matome trijų frazių: „Albert Einstein“, „Sherlock Holmes“ ir „Frankenstein“ rezultatus. Įrankis nustato šių frazių dažnumą, nes jos pasirodė 1800-2008 m. knygose. Kaip matote, apie „Frankenstein“ literatūroje pranešama nuo 1800 m. pradžios, palyginti su kitais dviem terminais, kurie literatūroje pasirodo vėliau.



1 pav. „Google Ngram“ frazių pavyzdys: „Albertas Einšteinas, Šerlokas Holmsas ir Frankenstein“.



Diskusijų užduotis

Kokias kitas tendencijas pastebite 1 paveiksle pateiktų terminų dažnumo atžvilgiu?



1 veikla: problemos formulavimas

Klausimas, kurį nagrinėsime šioje užduotyje, susijęs su kultūros pokyčiais Jungtinėse Amerikos Valstijose per pastaruosius du šimtmečius. Socialinis-kultūrinis teorinis aspektas, kurį ketiname naudoti, aprašytas tolesnėje pastraipoje, po kurios pateikiamas mūsų klausimas.

Socialinės kaitos ir žmogaus raidos teorija numato pasaulinį perėjimą nuo sąvokos „Gemeinschaft“ prie „Gesellschaft“, pagrįstą sociodemografiniais pokyčiais. Šiuos du žodžius 1887 m. įvedė Toenniesas ir jie buvo naudojami socialiniams pokyčiams tirti. „Gemeinschaft“ apibūdina privalomus, pirminius, jausmais pagrįstus tarpusavio santykius, kuriems būdinga kaimo aplinka, paprasti santykiai akis į akį, žemas technologijų lygis, ribotas išsilavinimas ir poreikiai, o ne turtas. Kita vertus, „Gesellschaft“ apibūdina sąveikų sistemą, kuriai būdingi savitarpio interesai, konkurencija ir derybų būdu pasiektas prisitaikymas, būdinga miesto aplinka, modernizuota visuomenė, aukštas technologijų ir turto lygis." (Šaltinis: Younes ir Reips, 2018, p. 1; Christenson, 1984, p. 160)

Per pastaruosius šimtmečius urbanizacijos tempai visame pasaulyje smarkiai išaugo. Šioje atvejo analizėje tiriame, ar šis pokytis susijęs su judėjimu nuo „Gemeinschaft“ prie „Gesellschaft“ sistemos. Keliame klausimą:

Ar Jungtinės Valstijos per pastaruosius du šimtmečius perėjo nuo „Gemeinschaft“ prie „Gesellschaft“ ir ar tai sutampa su perėjimu iš kaimo į miesto visuomenę?"

Norėdami atsakyti į šį klausimą, nagrinėsime sociokultūrinių pokyčių dėsningumus, atspindinčius žodžių dažnume per pastaruosius du šimtmečius Jungtinėse Amerikos Valstijose. Šiuo tikslu ypač naudinga „Google Ngram“ naršyklė, nes kultūrinės vertybės atspindi raštuose vartojami žodžiai ir (arba) terminai, todėl, nagrinėdami žodžių dažnumą knygose, galime pateikti įrodymų apie ilgalaikius kultūrinius vertybių pokyčius.



2 veikla: paieškos terminų parinkimas

Pirmasis žingsnis atsakant į mūsų klausimą – nustatyti tinkamas paieškos terminus, susijusius su sąvokomis „Gemeinschaft“ ir „Gesellschaft“ bei galinčias atspindėti kultūrinius vertybių pokyčius Jungtinėse Amerikos Valstijose. Šiame pavyzdyje atrenkant reprezentatyvius paieškos terminus svarbūs šie kriterijai: a) didelis termino dažnumas ir b) siauras semantinių interpretacijų diapazonas.



Diskusijų užduotis

Kaip manote, kodėl minėti kriterijai yra svarbūs šioje veikloje?

Labai svarbu parinkti dažnai vartojamus terminus, kad grafiko linijos iš tikrųjų vaizduotų kultūrinius pokyčius laike ir todėl, kad dažnai vartojami žodžiai yra šalies kultūros bruožai. Ne mažiau svarbu šiame pavyzdyje pasirinkti žodžius, turinčius siaurą semantinių interpretacijų diapazoną. Taip yra todėl, kad žodžiai, turintys plačias reikšmes, gali būti vartojami įvairiuose kontekstuose, kurie ne visada susiję su kultūrinėmis vertybėmis. Toliau pateiktoje lentelėje matote žodžius, atitinkančius pirmiau nurodytus kriterijus ir atspindinčius „Gemeinschaft“ ir „Gesellschaft“ sistemą.

1 lentelė: Terminai, susiję su Gemeinschaft ir Gesellschaft

„Gemeinschaft“	obliged, duty, give, benevolence, act, deed (liet. įpareigotas, pareiga, duoti, geranoriškumas, aktas, veiksmas)
„Gesellschaft“	choose, decision, get, acquisition, feel, emotion (liet. pasirinkti, sprendimas, gauti, įsigyti, jausti, emocijos)



Diskusijų užduotis

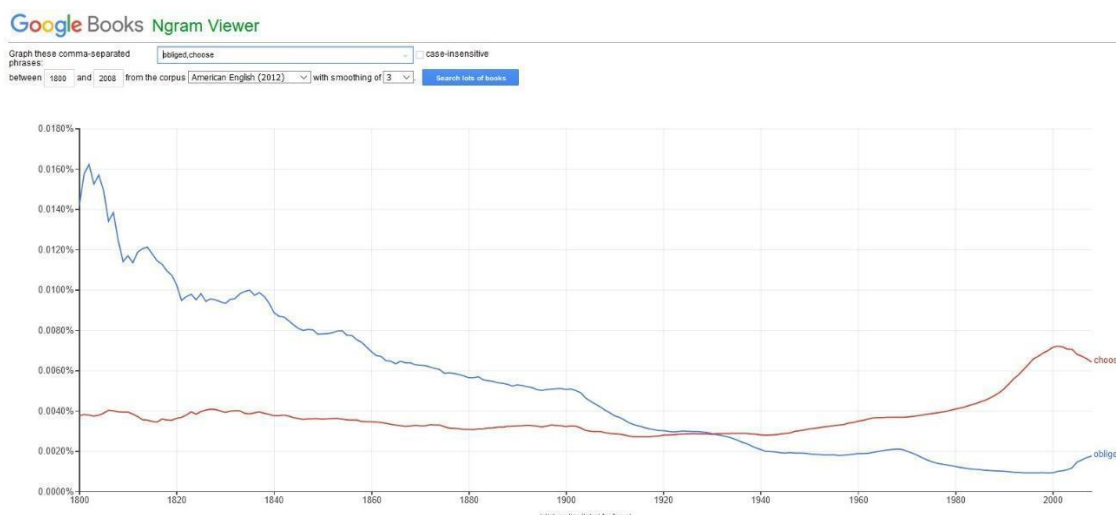
Kaip galime pasinaudoti pirmiau pateiktu sąrašu ir „Google Ngram“ naršykle, kad atsakytume į anksčiau iškeltą klausimą?

Aptarėme, kad „Google Ngram“ gali būti naudojama literatūroje pateikiamų žodžių dažnumui per tam tikrą laiką atvaizduoti. Mūsų pavyzdyje parodoma, kad galime naudoti ir lyginti terminus, atspindinčius priešingus požiūrius, bruožus ir savybes, ir stebėti, kaip jų dažnumas kinta laikui bėgant.



Diskusijų užduotis

Toliau pateiktame paveikslėlyje pavaizduotas žodžių „obliged“ (liet. įpareigoti) (susietas su Gemeinschaft sistema) ir „choose“ (liet. pasirinkti) (susietas su Gesellschaft sistema) dažnumas. Ką pastebėjote apie šių dviejų terminų dažnumą laikui bėgant?



2 pav. Žodžių „obliged“ ir „choose“ palyginimas



3 veikla: terminų palyginimas ir aiškinimas

Šiame etape, naudodamiesi „Google Ngrams“, palyginsime daugiau terminų, susijusių su „Gemeinschaft“ ir „Gesellschaft“ sistemomis, ir stebėsime, kaip keitėsi jų dažnumas per pastaruosius du šimtmečius.



Grupės užduotis

Spustelėkite šią nuorodą, jei norite pasiekti „Google Ngrams“ (<https://books.google.com/ngrams>).

1. Naudokite paieškos lauką (ištrinkite visus lauke jau esančius paieškos žodžius) ir įveskite šiuos du kableliu atskirtus žodžius: „obliged“, „choose“.
2. Iš išplečiamojo meniu „from the corpus“ pasirinkite American English (2012) ir spustelėkite mygtuką „search lots of books“.
3. Ką pastebėjote apie šių dviejų terminų dažnumą laikui bėgant?
4. Taip pat ištrinkite šių žodžių dažnumą (pasirinkite bent du):
 - a. „Give“ ir „Get“,
 - b. „Benevolence“ ir „Acquisition“,
 - c. „Act“ ir „Feel“,
 - d. „Deed“ ir „Emotion“
5. Kaip kinta šių žodžių dažnumas?

Ar jūsų pastebėjimai patvirtina per pastaruosius du šimtmečius vykusį perėjimą nuo „Gemeinschaft“ prie „Gesellschaft“?

*

Iš tiesų, kaip ir numatė socialinių pokyčių ir žmogaus raidos teorija, per pastaruosius du šimtmečius susilpnėjo tokios sąvokos kaip išsipareigojimas, pareiga, duoti, geranoriškumas, veiksmas ir poelgis, kurios atspindi kaimo ir „Gemeinschaft“ aplinką. Per tą patį laikotarpį padaugėjo tokių terminų kaip pasirinkimas, sprendimas, gauti, išgyti, jausti ir emocija, kurie atspindi miesto ir „Gesellschaft“ aplinką.



Apibendrinimas (neprivaloma)

Atliekant šią veiklą paaiškėjo, kad Jungtinėms Amerikos Valstijoms judant „Gesellschaft“ kryptimi, „Gesellschaft“ kultūros bruožų (kuriuos atspindi atitinkami žodžiai amerikiečių knygoje) kiekybiškai daugėjo, o „Gemeinschaft“ kultūros bruožų (kuriuos atspindi atitinkami žodžiai amerikiečių knygoje) kiekybiškai mažėjo. Ar šiuos ryšius ir tendencijas galima apibendrinti kitose pasaulio dalyse, kaip tai numato socialinių pokyčių ir žmogaus raidos teorija?



Grupės užduotis

Pakartokite 3 etape aprašytus veiksmus; šį kartą pasirinkite pasirinktą literatūrą (pvz., britų arba vokiečių) ir atsakykite į tą patį klausimą apie pasirinktą šalį.

Pastaba: vokiečių literatūrai galima vartoti šiuos terminus (šaltinis: Younes & Reips, 2018):

1. Versprechen ir Auswählen
2. Pflicht ir Entscheidung
3. Geben ir Bekommen
4. Güte ir Kauf
5. Handeln ir Spüren
6. Handlung ir Emotion



Mokymosi ištekliai

Gemeinschaft to Gesellschaft.pdf (neprivaloma)

B Kultūros psichologijos pokyčiai 1800-2000 m. (pasirinktinai)

C Kintanti kultūros psichologija vokiškai kalbančiose šalyse (pasirinktinai)



Nuorodos

Christenson, J.A., (1984). Gemeinschaft ir gesellschaft: J.: Gesellschaft (Gesellschaft): erdvinės ir bendruomeninės hipotezės tikrinimas. *Social Forces*, 63(1), p. 160-168.

Greenfield, P.M., (2013). Kintanti kultūros psichologija nuo 1800 iki 2000 metų. *Psichologijos mokslas*, 24(9), p. 1722-1731.

Younes, N. ir Reips, U.D., (2018). Kintanti kultūros psichologija vokiškai kalbančiose šalyse: Google Ngram tyrimas. *International Journal of Psychology*, 53, p. 53-62.



2.2 veikla Modeliavimas ir imitavimas naudojant „NetLogo“

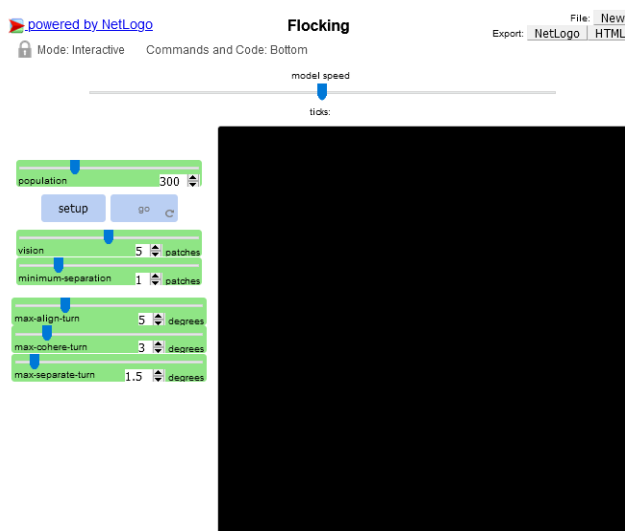
Bendras laikas: 2,30-3 valandos

"NetLogo" yra daugiaagentė programuojama modeliavimo aplinka, skirta modeliuoti gamtos ir socialinius reiškinius ir parodyti, kaip jie vystosi laikui bėgant. Naudodami šią priemonę galite sukurti pasaulį, sudarytą iš stačiakampių arba dėmių, ir nustatyti parametrus agentams (arba vėžliams), kurie juda ir sąveikauja tarpusavyje bei su aplinka. Šiame užsiėmime naudosime NetLogo, atlikdami epidemiologijos atvejo tyrimą ir paleisime modelio simuliacijas, kad ištirtume, kaip dideli ir maži pokyčiai gali paveikti aplinką.



Diskusija

Prieš pradėdami dirbti su šia užduotimi, spustelėkite šią nuorodą, kad atvertumėte įrankį <http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/Sample%20Models/Art/Fireworks.nlogo>. Iš išskleidžiamojo meniu „Search the Models Library“ („Ieškoti modelių bibliotekoje“) pasirinkite parinktį `sample models/Biology/Flocking`. Atveriamas pavyzdys, kaip pateiktame 3 paveikslėlyje.



3 pav. „Flocking“ pavyzdys programoje „NetLogo“

Šiuo modeliu bandoma imituoti paukščių būriavimąsi. Kiekvienas paukštis laikosi lygiai tų pačių taisyklių: suderinimas, atsiskyrimas ir sanglauda. Suderinimas valdomas slankikliu „max-align-turn“, reiškia, kad paukštis stengiasi pasisukti taip, kad judėtų ta pačia kryptimi, kuria juda netoliese esantys paukščiai. Atsiskyrimas valdomas „max-separate-turn“ slankikliu, reiškia, kad paukštis pasisuks taip, kad išvengtų kito paukščio, kuris priartėja per arti. Sanglauda valdoma slankikliu „max-coherence-turn“, reiškia, kad paukštis judės link kitų netoliese esančių paukščių (nebent kitas paukštis būtų per arti). Kai du paukščiai yra per arti, atsiskyrimo taisyklė turi viršenybę prieš kitas dvi, kurios išjungiamos, kol pasiekiamas minimalus atstumas. Regėjimo slankiklis yra atstumas, kuriuo kiekvienas paukštis gali matyti 360 laipsnių aplink save.

Kairėje pusėje esantys žali diapazono slankikliai (pvz., max-fireworks) naudojami modeliui parametrizuoti. Mygtukas SETUP (nustatyti), nustato modelį pagal visų slankiklių nurodytas vertes. Mygtuku GO (paleisti) modelis paleidžiamas.



Poros užduotis

Pirmiausia nustatykite simuliacijoje dalyvaujančių paukščių skaičių nustatydami populiacijos slankiklį į norimą vertę. Paspauskite SETUP, tada GO, kad pradėtumėte modeliavimą (Numatytieji slankiklių nustatymai leis pasiekti pakankamai gerą paukščių susitelkimo elgseną. Tačiau galite su jais eksperimentuoti). Reguliokite slankinkius, kad pamatytumėte, kaip galima gauti glaudesnius, laisvesnius, mažesnius ar didesnius pulkus.



1 veikla: problemos formulavimas

Klausimas, kurį nagrinėsime šioje užduotyje, susijęs su epidemiologija, ypač su Zikos virusu.

Zikos virusas yra uodų platinamas flavivirusas, kuris pirmą kartą buvo nustatytas 1947 m. Ugandoje tarp beždžionių. Apskaičiuota, kad inkubacinis laikotarpis (laikas nuo užsikrėtimo iki ligos simptomų atsiradimo) yra 3-14 dienų. Daugumai žmonių, užsikrėtusių Zikos virusu, simptomai nepasireiškia. Paprastai simptomai būna lengvi, įskaitant karščiavimą, bėrimą, konjunktyvitą, raumenų ir sąnarių skausmą, blogą savijautą ir galvos skausmą, ir paprastai trunka 2-7 dienas. Virusas pirmiausia perduodamas įkandus užsikrėtusiam *Aedes* genties uodui, daugiausia *Aedes aegypti*, tropiniuose ir subtropiniuose regionuose. *Aedes* uodai dažniausiai įkanda dienos metu, anksti ryte ir vėlyvą popietę arba vakare. Tai tie patys uodai, kurie perneša dengę, čikungunją ir geltonąjį drugį. Virusas taip pat perduodamas iš motinos vaisiui nėštumo metu, per lytinius santykius, perpilant kraują ir kraujo produktus bei persodinant organus (šaltinis: Pasaulio sveikatos organizacija ¹).

Šiame atvejo tyrime nagrinėsime hipotetinį epidemiologinį scenarijų ir bandysime suprasti bei įvertinti viruso plitimo riziką. Scenarijus, kurį ketiname nagrinėti, yra toks:

Nedideliame Pietų Afrikos Respublikos mieste dviem žmonėms nustatytas Zikos viruso teigiamas rezultatas. Vyriausybė nori nuspėti, kokia rimta yra situacija, ir taip imtis būtinų

¹ <https://www.who.int/news-room/fact-sheets/detail/zika-virus>

prevencijos priemonių. Iš viso šiame mieste gyvena 800 gyventojų, o atlikus pirminį vertinimą nustatyta, kad uodų skaičius vietovėje yra 1040, o plėšrūnų - 21.

Mums reikia atsakyti į šį klausimą:

Kiek žmonių užsikrės ir kokia būtų geriausia viruso prevencijos strategija?

Norėdami atsakyti į šį klausimą, naudosisime „NetLogo“ ir imituosime Zikos viruso modelį.



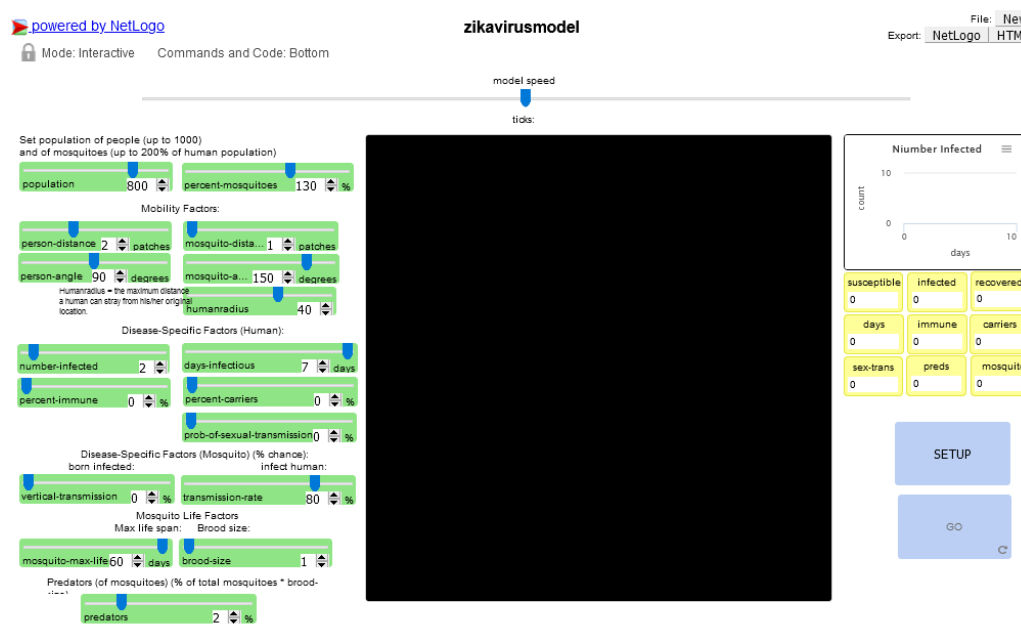
2 veikla: modeliavimas

Šiame etape dirbsime su jau įgyvendintu modeliu, kuris padės mums suprasti, kaip modeliavimas ir imitavimas gali būti naudojami siekiant sukurti naują supratimą ir žinias apie reiškinį ir kaip jie gali paveikti sprendimų priėmimą.

Norėdami atverti „NetLogo“, spustelėkite šią nuorodą:

<http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/Sample%20Models/Art/Fireworks.nlogo>

Spustelėkite „browse“ ir suraskite bei įkelkite modelį „zikavirusmodel.nlogo“. Sėkmingai įkėlus modelį, turėtų būti matomas toliau pateiktas pavyzdys.



4 pav. Zikos virusas „NetLogo“ programoje

Pagal šį modelį uodai įkando Zika virusu užsikrėtusiam žmogui ir toliau platino ligą įkandę kitiems, neužsikrėtusiems žmonėms. Šiame modelyje yra daug kintamųjų, kuriais galima manipuluoti. Veikloje daugiausia išskirsime šiuos kintamuosius:

- *žmogaus atstumai*, kuris parodo žmonių mobilumą – kaip toli jie gali pasukti ir kiek kvadratų (lopinėlių) gali nueiti per dieną.

- *infekuotųjų skaičius*, kuris parodo pradžioje užsikrėtusių žmonių skaičių.
- *užkrečiamųjų dienų skaičius*, kuris parodo, kiek laiko tipiškas žmogus yra užsikrėtęs.
- *imuniteto procentas* – tai populiacijos procentinė dalis, kuri turi imunitetą (= paskiepyta).
- *plėšrūnų*, kurie rodo plėšrūnų skaičių.
- *uodų procentas*, kuris parodo, kokią dalį žmonių populiacijos sudaro uodai.

Kiekvieną kartą paleisdami modelį turite atlikti šiuos tris veiksmus iš eilės: pirma, nustatyti slankiklius (jei norite juos koreguoti), antra, spustelėkite SETUP (apatiniam dešiniajame ekrano kampe) ir, trečia, spustelėkite GO, kad paleistumėte modelį.

Viršutiniame dešiniajame kampe esančiame grafike ir po juo esančiuose laukeliuose matysite parametrus, kurie rodo, kaip visuomenei sekasi kovoti su virusu pagal nurodytus slankiklių duomenis.



Darbas porose

Paleiskite simulatorių su numatytosiomis reikšmėmis, kurios atitinka 1 etape pateiktą problemą formuluotę. Pasibaigus modeliavimui, atsakykite į šį klausimą: Kiek žmonių iš viso užsikrėtė ir per kiek dienų?

Paleidę simulatorių galėsite apskaičiuoti, kiek žmonių užsikrės virusu tam tikromis aplinkybėmis. Šiame pavyzdyje šis skaičius yra tarp 740 ir 775. Taigi, naudodami savo modelį, galėjome atsakyti į vieną klausimo dalį dėl žmonių, kurie užsikrės Žikos virusu, skaičiaus. Dar neatsakėme į klausimą, kokia strategija būtų geriausia užkirsti kelią plitimui. Į tai sutelksime dėmesį kitame etape.



3 veikla: tolesnis tyrimas – aiškinimas

Šiame modelyje yra keli kintamieji, kuriuos galima koreguoti ir ištirti, kaip liga išplis tarp gyventojų. Norėdami ištirti geriausią strategiją, kaip užkirsti kelią plitimui, atsižvelgsime į tris kintamuosius: plėšrūnus, atstumą tarp asmenų ir žmonių su imunitetu procentą.



Grupės užduotis

Šioje užduotyje ketiname ištirti, kaip toliau išvardyti pakeitimai veikia viruso plitimą ir kuris iš jų yra veiksmingesnis.

1. Plėšrūnų skaičių padidinkite iki 6 %. Tada spustelėkite SETUP ir GO. Užsirašykite pasveikusiuųjų (recovered) skaičių. Prieš pereidami prie 2 punkto, nustatykite plėšrūnų skaičių į 2 % (numatytoji vertė).

2. Padidinkite imunitetą turinčių žmonių skaičių iki 6 %. Spustelėkite SETUP ir GO. Užsirašykite pasveikusiųjų (recovered) skaičių. Prieš pereidami prie 3 punkto, nustatykite imunitetą turinčių žmonių skaičių į 0 (numatytoji vertė).
3. Sumažinkite žmogaus judėjimą iki 1 lopinėlio. Tada spustelėkite SETUP ir GO. Užsirašykite pasveikusiųjų žmonių skaičių.

*

Pastebėsite, kad, apribojus žmogaus judėjimą iki 1 lopinėlio, užkrėstų žmonių skaičius bus maždaug 400-680. Jei plėšrūnų skaičių padidinsite 6 %, užkrėstų žmonių skaičius bus tarp 140-300, o padidinus imunitetą turinčių žmonių skaičių iki 6 %, užkrėstų žmonių skaičius bus tarp 630-740. Todėl dabartinėmis sąlygomis, kurias nagrinėja mūsų modelis, geriausia strategija, siekiant užkirsti kelią plitimui visoje populiacijoje, yra plėšrūnų skaičiaus padidinimas 6 %.



4 veikla: baigiamasis klausimas

Šioje užduotyje nagrinėjome hipotetinę epidemiologijos scenarijų ir, naudodami modeliavimą ir imitavimą, analizavome viruso plitimo dinamiką, keisdami kintamuosius ir vertindami rezultatus.



Diskusijų užduotis

Kaip modeliavimas ir imitavimas gali paveikti sprendimų priėmimą kituose dalykuose, išskyrus epidemiologiją?



Mokymosi ištekliai

zikavirusmodel.nlogo



Veikla 2.3 „Microsoft Excel“ naudojimas duomenims tvarkyti ir atvaizduoti

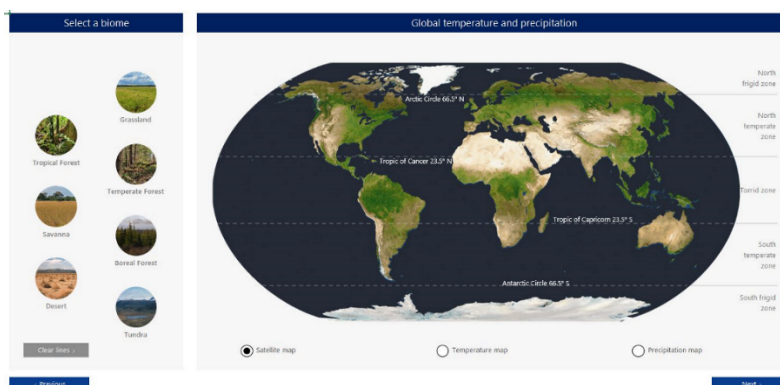
Bendras laikas: 2,30-3 valandos

„Excel“ yra skaičiuoklės įrankis, skirtas duomenims tvarkyti, atlikti skaičiavimus, analizuoti ir pateikti duomenis diagramomis ar grafikais. Šioje užduotyje naudosime duomenis pateiktus „Excel“, juos analizuosime, vaizduosime, nagrinėsime grafikuose pavaizduotas tendencijas ir susiesime informaciją, kad atsakytume į klausimą apie klimato skirtumus biomuose ir kaip jie veikia jų paviršiaus spalvas.



Diskusija

Prieš pradėdami dirbti su šia užduotimi, atsiverskite [Žemės spalvų sąsiuvinį \(Activity 2.4 - Earth colour workbook.xlsx\)](#) ir pereikite prie darbalapio „Biomų žemėlapis“ (Mapping biomes), kad pamatytumėte iš palydovų nuotraukų sukurtą Žemės paviršiaus paveikslą (5 pav.).



5 pav. 1 Biomų žemėlapis darbalapis



Užduotis porose

Spustelėkite Žemės palydovo nuotraukos kairėje pusėje esančius biomų mygtukus, kad pažymėtumėte skirtingų biomų regionų vietas.

- Kaip manote, kokia spalva geriausiai apibūdina kiekvieną bioma?
- Kuo paaiškinamos skirtingos spalvos, susijusios su skirtingais biomais?

Patarimas: pasirinkite temperatūros ir kritulių žemėlapis mygtuką, esantį po palydovo vaizdu, kad galėtumėte susieti spalvas ir fizines sąlygas.

*

Kaip pastebėjote, regionai, kuriuose iškrenta daug kritulių, yra žalesni, o aukštesnė temperatūra ir mažesnis kritulių kiekis atitinka regionus, kurie gali atrodyti rudesni. Ašigaliuose vidutinė temperatūra yra labai žema, todėl sniegas ir ledas yra baltos spalvos.



1 veikla: problemos aprašymas

Atlikdami šią užduotį naudosime duomenis, kad geriau suprastume, kodėl biomuose vyksta spalvų pokyčiai. Nagrinėsime septynis sausumos biomus ir jiems būdingas spalvas, matomas iš kosmoso, kurios susijusios su tokiais veiksniais kaip temperatūra ir krituliai. Klausimas, kurį spręsimė, yra toks:

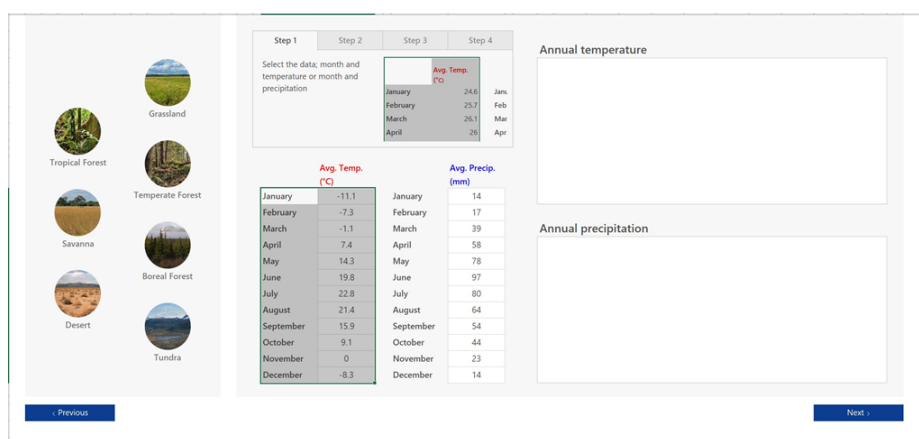
Kaip keičiasi temperatūra ir kritulių kiekis skirtinguose biomuose per metus ir kaip ši kaita atsispindi spalvose?



2 veikla: Temperatūros ir kritulių kiekio grafikų sudarymas

Atlikdami užduotį peržiūrėjote vidutinės temperatūros ir kritulių kiekio pasaulyje duomenis ir pradėjote ieškoti sąsajų tarp palydovo nuotraukoje pavaizduotų Žemės paviršiaus spalvų ir vidutinio metinio kritulių kiekio bei temperatūros. Šiame etape nagrinėsime mėnesinius kritulių ir temperatūros pokyčius, vykstančius per metus biomuose.

Atlikdami šią užduotį pereikite į toliau pateiktame paveikslėlyje pavaizduotą darbalapį „Sukurti biomo klimato diagramą“.



6 pav. 2 darbalapis „Biomų klimato diagramos sukūrimas“



Užduotis porose

Atlikdami šią užduotį pasirinkite vieną biomą, kurį norėtumėte išsamiau iširti. Spustelėję biomo mygtukus kairėje, pamatysite, kad jo mėnesio temperatūros ir kritulių duomenys atvaizduojami skiltyse „Avg Temp“ ir „Avg. Precip“ etiketėse.

1. Pasirinkite mėnesio temperatūros duomenis ir spustelėkite „Excel“ meniu juostoje esančią įterpimo parinktį. Spustelėkite rekomenduojamų diagramų parinktį ir pasirinkite diagramą, kurią norėtumėte naudoti biomo mėnesio temperatūros duomenims pavaizduoti.
2. Pakartokite tuos pačius veiksmus su kritulių duomenimis

Kaip per metus kinta jūsų biomo temperatūros ir kritulių kiekio duomenys? Stebėkite, kurį (-iais) mėnesį (-iais) temperatūra ir kritulių kiekis yra didžiausias ir mažiausias.

Pasirinkite kitą biomą ir pakartokite šią užduotį, sukurdami grafiką, kuriame pavaizduokite jo mėnesio temperatūrą ir kritulių kiekį. Palyginkite šiuos du biomus, stebėdami, kuriame iš jų temperatūros ir kritulių kiekio svyravimai per metus yra didesni ar mažesni.



Nepivaloma užduotis

Pakartokite pirmiau pateiktą užduotį visiems biomasams ir pasirinkite skirtingus diagramų tipus. Stebėkite, kurios diagramos yra naudingesnės vaizduojant temperatūros ir kritulių kiekio pokyčius ir nustatant tendencijas, skirtumus ir panašumus tarp dviejų ar daugiau biomų.

*

Šiuo metu atsakėme į pirmąją klausimo dalį, susijusią su temperatūros ir kritulių kiekiu kitimu per metus biome. Naudodami grafikus, kuriuose pavaizduoti biomo temperatūros ir kritulių kiekio duomenys, galėjome stebėti šiuos svyravimus ir palyginti biomas tarpusavyje. Tačiau turime toliau tirti, kaip šie temperatūros ir kritulių duomenų pokyčiai atsispindi spalvose.



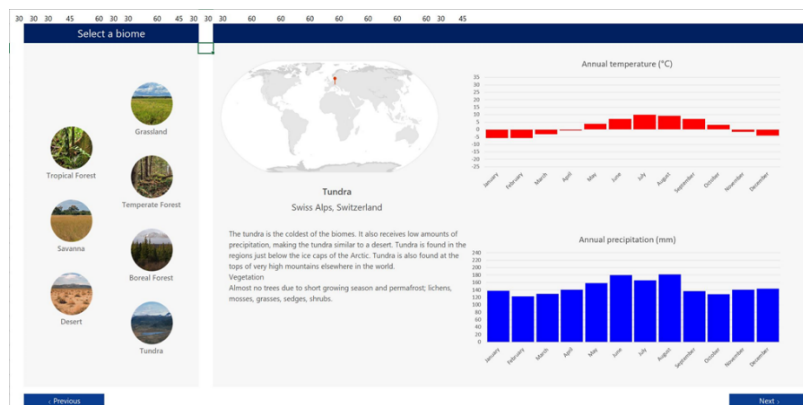
3 veikla: spalvų palyginimas su temperatūra ir kritulių kiekiu biomuose

Ankstesniame etape naudojome grafikus, kad pavaizduotume ir palygintume dviejų biomų mėnesio temperatūros ir kritulių kiekio pokyčius. Šiame etape ketiname susieti biomo charakteristikas, temperatūros ir kritulių kiekio duomenis ir iširti, kaip sezoninės augmenijos pokyčius atspindi ryškiausia biomo kraštovaizdžio spalva tam tikrą metų mėnesį.



Užduotis porose

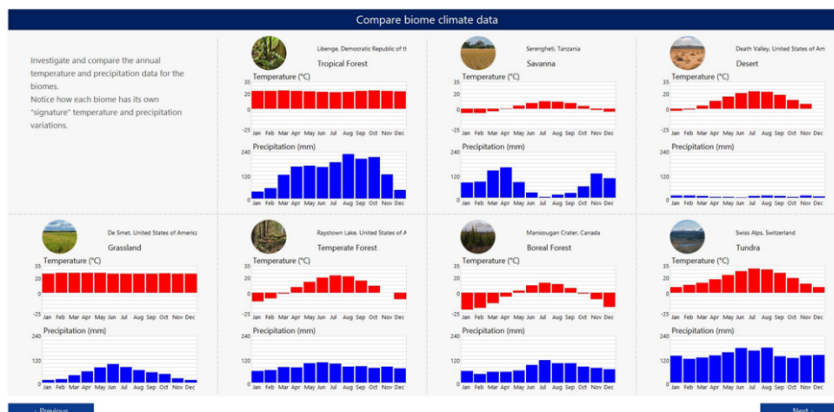
Pereikite prie toliau pateiktame paveikslėlyje pavaizduoto darbo lapo „Informacija apie biomas“. Dar kartą pasirinkite biomas, kuriuos pasirinkote ankstesniame etape, ir perskaitykite pateiktą informaciją apie jų augaliją. Ką pastebėjote apie biomo temperatūrą, kritulių kiekį ir augmeniją?



7 pav. Informacija apie biomas darbo lapas

Lygindami biomo temperatūrą, kritulių kiekį ir augmeniją pastebėsite, kad tam tikro regiono augmenija yra specialiai pritaikyta to regiono klimato sąlygoms. Taip pat pastebėsite, kad net jei vieno regiono temperatūra gali būti panaši į kito, kritulių kiekio skirtumai daro didelę įtaką biomo augalijos tipui, taigi ir jo paviršiaus spalvoms.

Norėdami atsakyti į šį klausimą, pereikite į toliau pateiktame paveikslėlyje pavaizduotą darbo lapą „Biomų klimato duomenų palyginimas“.



8 pav. Biomų klimato duomenų palyginimo darbalapis



Užduotis porose

Remiantis temperatūros ir kritulių kiekio duomenimis,

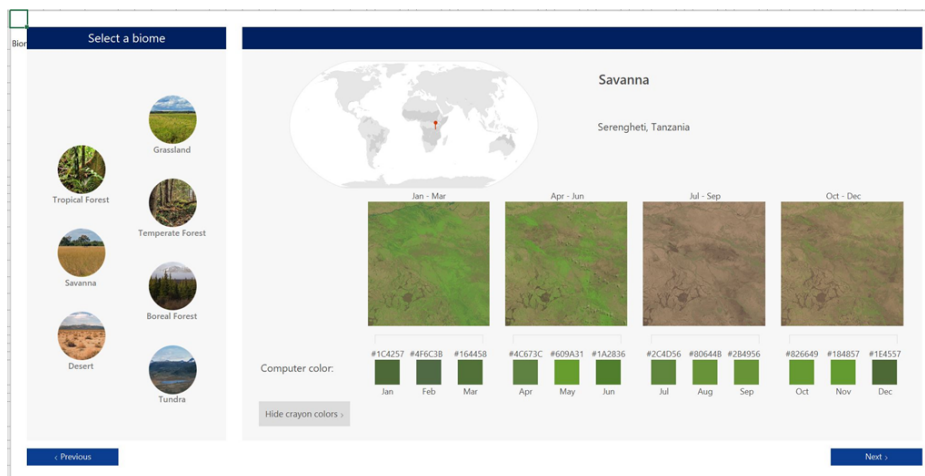
- Kuriuose biomų regionuose, jūsų manymu, paviršiaus spalvos per metus bus pastoviausios?
- Kaip manote, kurios iš paviršiaus spalvų per kalendorinius metus svyruos labiausiai?

Paaiškinkite savo argumentus.

*

Spalva gali būti naudojama biomų pokyčiams per metus apibūdinti; besikeičianti biomo augmenija lemia jo paviršiaus spalvą, kurią atspindi ryškiausia biomo kraštovaizdžio spalva tam tikrą metų mėnesį.

Norėdami atsakyti į šį klausimą, pereikite prie toliau pateiktame paveikslėlyje pavaizduoto darbo lapo „Biomų spalvos per metus“.



9 pav. 5 darbalapis „Biomų spalvos per metus“



Užduotis porose

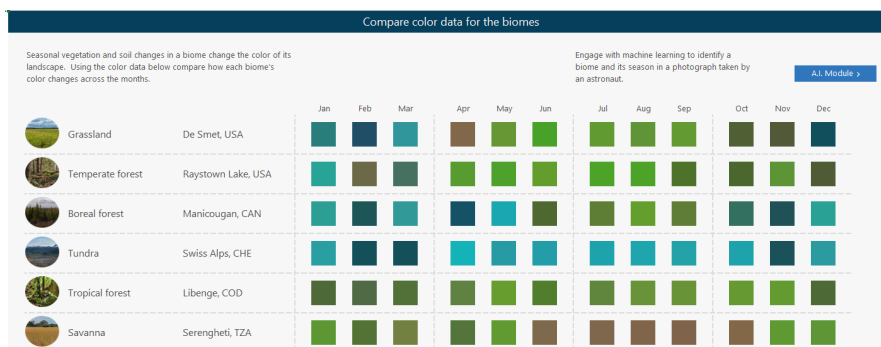
Remiantis temperatūros ir kritulių kiekio duomenimis,

- Kuriuose biomų regionuose, jūsų manymu, paviršiaus spalvos per metus bus pastoviausios?
- Kaip manote, kurios iš jų paviršiaus spalvos per kalendorinius metus svyruos labiausiai?

Papaišinkite savo argumentus.

*

Norėdami atsakyti į šį klausimą, pereikite į toliau pateiktame paveikslėlyje pavaizduotą darbalapį „Biomų spalvų duomenų palyginimas“.



10 pav. Biomų spalvų duomenų palyginimo darbalapis



Užduotis porose

Kaip skirtingų biomų regionų mėnesio paraščių spalvos susijusios su temperatūra ir kritulių kiekiu? Pateikite duomenų pavyzdžių.

Patarimas: Atkreipkite dėmesį, kaip skirtingos spalvos skirtinguose biomuose susijusios su kritulių kiekiu ir temperatūra.

*

Galbūt pastebėjote, kad arbatžolių spalva žymi vietoves, kuriose yra sniego, o vietovės, kuriose iškrenta daug kritulių, yra žalesnės, o aukštesnė temperatūra ir mažesnis kritulių kiekis atitinka regionus, kurie gali atrodyti rudesni.

Atsakydami į antrąją klausimo dalį – *kaip kritulių ir temperatūros pokyčiai atsispindi spalvose* – pastebėjome, kad dėl temperatūros ir kritulių pokyčių besikeičiančią biomo augmeniją atspindi ryškiausia biomo kraštovaizdžio spalva tam tikrą metų mėnesį, taigi, tai lemia biomo paviršiaus spalvą.



4 veikla: Klimato kaitos prognozavimas - išplėstinė veikla (pasirinktinai)

Ankstesniuose etapuose atliktos užduotys ir klausimai padėjo jums atsakyti į klausimą, kaip temperatūra ir kritulių kiekis įvairiuose biomuose keičiasi per metus ir kaip ši kaita atsispindi spalvose. Šiame etape nagrinėsite, kaip klimato kaitos problema veikia pasirinktą biomo regioną. Šiuo tikslu rinksite ir analizuosite šio biomo kritulių ir temperatūros duomenis ir tirsite, kaip gali pasikeisti jo būdingos spalvos, jei klimato kaitos problema nebus sprendžiama.



Grupės užduotis

Ištirkite klimato kaitos problemą, kuri daro poveikį pasirinktam biomo regionui.

1. Surinkite pastarųjų 10 metų mėnesio temperatūros ir kritulių kiekio duomenis ir išsiaiškinkite, ar yra kokių nors tendencijų ar įdomių dėsningumų. Duomenis įrašykite „Excel“ lentelėje.
2. Sukurkite biomo regiono temperatūros ir kritulių diagramas, kurios padės nustatyti tendencijas.
3. Pagalvokite apie klimato kaitos problemas jūsų biomo regione ir kaip jos atsispindi jūsų surinktuose duomenyse. Aptarkite, kaip galėjo pasikeisti šio biomo regiono spalvos dėl klimato kaitos problemų, ir pateikite prognozes, kaip gali pasikeisti šio regiono spalvos per ateinančius 10 metų, jei klimato kaitos problemos nebus sprendžiamos.



Mokymosi ištekliai

Darbo sąsiuviniai: [Activity 2.4 - Earth colour workbook.xlsx](#)



3 dalis: Patirkite kompiuterinį mąstymą programuodami

Bendras laikas: 10 valandų

Šiame skyriuje susipažinsite su programavimo pagrindais ir praktiškai išbandysite algoritminį mąstymą pasakojimo ir žaidimų kontekste. Taip pat išmoksite naudoti srauto diagramas algoritmams vaizduoti.



Indėlis į mokymosi rezultatus

Mokymosi rezultatai

- standartizuotai kurti ir užrašyti istorijos idėjas
- nustatyti pagrindinius elektroninių interaktyvių istorijų ir tradicinių popierinių knygų kūrimo skirtumus
- nustatyti ir paaiškinti esamos programos algoritmą
- atlikti esamos programos pakeitimus
- derinti ir taisyti esamą programą
- suplanuoti ir sukurti naują programą interaktyviai istorijai kurti
- sukurti ir plėtoti interaktyvią istoriją naudojant programuojamus elementus
- naudoti ciklus, kintamuosius, transliuojamus pranešimus, IF teiginius ir nuosekliąsias instrukcijas programoje
- suprasti teisingų nurodymų svarbą
- suprasti, kas yra algoritmas
- pavaizduoti algoritmą srautų diagramoje
- įvertinti algoritmo veiksmingumą
- įgyvendinti iš anksto parašytą algoritmą arba srauto diagramą programoje „Scratch“
- paaiškinti, ką reiškia sąvoka kintamasis
- kurti ir naudoti kintamuosius programoje
- paaiškinti, ką reiškia sąvokos „pasirinkimas“ ir „ciklas“.
- algoritme ar programoje naudoti pasirinkimo ir ciklo teiginius.



3.1 veikla: istorijų pasakojimas su „Scratch“

Skaitmeninių istorijų pasakojimui naudojama skaitmeninė medija (vaizdai, balsas, muzika, judesys), kad būtų papasakota istorija. Per pastaruosius kelerius metus skaitmeninis pasakojimas tapo vis populiariesne ir veiksmingesne mokymosi veikla, padedančia siekti įvairių mokymosi tikslų, ypač su informatiniu mąstymu susijusių mokymosi tikslų.

Šiame skyriuje praktiškai išbandysite skaitmeninį pasakojimą „Scratch“ programa. [Scratch](#) – tai nemokama mokomoji, vaizdinė ir blokais pagrįsta programavimo aplinka. [Scratch](#) programoje mokiniai gali plėtoti savo idėjas projektų pavidalu, naudodami programuojamąją terpę, pavyzdžiui, vaizdo įrašus, paveikslėlius, žaidimus ir animaciją.



Poros užduotis

Žiūrėkite vaizdo įrašą „[Scratch](#)“ įvadas

Peržiūrėkite šį aštuntos klasės mokinių, nagrinėjančių periodinę elementų lentelę, sukurtą [projekto pavyzdį](#).

*

Naudodami programą „Scratch“

- Suformuluokite problemą, kad nustatytumėte, kaip naudoti „Scratch“ programos elementus istorijai kurti – kurti siužetą, aplinką, seką ir perspektyvą.
- Logiškai tvarkyti ir analizuoti duomenis kuriant programas, veikėjų ir jų aplinkos valdymui.
- Atvaizduokite istorijos turinį judėdami „Scratch“ veikėjais.
- Naudokite algoritminį mąstymą, kad sukurtumėte kodą, kuris priverčia veikėjus judėti ir bendrauti.



Pasiruošimas

Darome prielaidą, kad lektorius jums suteikė „CS First“ vartotojo vardus ir slaptažodžius. Prieš pradėdami darbą, atlikite šiuos veiksmus:

- Atidarykite naują naršyklės langą ir eikite į g.co/CSFirst
- Viršutiniame dešiniajame kampe spustelėkite „Prisijungti“.
- Spustelėkite „Aš esu studentas“
- Spustelėkite „Prisijungti su CS First“
- Spustelėkite „Įveskite klasės kodą“.
- Įveskite klasės kodą 3h24s3
- Įveskite savo vartotojo vardą ir slaptažodį



Dialogas

Sužinokite apie „CS First“ ir sukurkite istoriją, kurioje du veikėjai kalbasi nenaudodami klausimų.



1. CS First Survey
2. Introduction to Dialogue and Sequencing
3. Setting the Scene
4. Speaking and Responding
5. Add-Ons
6. Reflection
7. Wrap-Up: Dialogue
8. Wrap-Up: Next Steps



Užduotis porose

- Paspauskite pradžios mygtuką.
- Praleiskite apklausą ir pasirinkite 2: Įvadas į dialogą ir eiliškumą

- Į naršyklės langą įtraukite šį skirtuką: <https://scratch.mit.edu/>
- Jei dar neturite „Scratch“ paskyros, užsiregistruokite.
 - Patarimas: skirtukus galite greitai perjungti naudodami ctrl tab
- Pradėkite vaizdo įrašą nuo 2:30, vykdykite pabaigoje pateiktus nurodymus.
- Paspauskite Next
- Scenos kūrimas (3)
 - Šiuo metu turėtumėte būti sukūrę naują projektą, tarkime, pavadinimu „myFirstStory“.
- Paleiskite vaizdo įrašą ir vykdykite pabaigoje pateiktus nurodymus: nenutraukite vaizdo įrašo.
- Paspauskite Next
 - Kalbėjimas ir atsakinėjimas (4)
- Paleiskite vaizdo įrašą ir vadovaukitės pabaigoje pateiktais nurodymais.
- Paspauskite Next
 - Papildomi priedai (5)
- Puslapio apačioje pasirinkite "Pridėti judesį".
 - Patarimas: jums nereikės patiems įvesti koordinacių, jei prieš pasirinkdami atitinkamą judesio bloką iš pradžių pastatysite veikėją į norimą padėtį.
- Pasirinkite "Pridėti trečią simbolį".



Patikrinkite

Start

Papasakokite istoriją, kurioje veikėjas eina per sceną ir aprašo, ką mato.



Užduotis porose

- Paspauskite pradžios mygtuką.
 - Pirmojo vaizdo įrašo žiūrėti nebūtina, bet vietoj to spustelėkite nuorodą su Starter Project 1. Ji atvers „Scratch“ su pradiniu projektu, kuriame jau yra tam tikras kodas.

- ▶ 1. What is Computer Science?
- ▶ 2. Unexpected Encounter
- ▶ 3. Add-Ons
- ▶ 4. Reflection
- ▶ 5. Wrap-Up: Check It Out
- ▶ 6. Wrap-Up: Next Steps

Instructions

Choose a story starter by clicking a starter project link next to this video.

Links



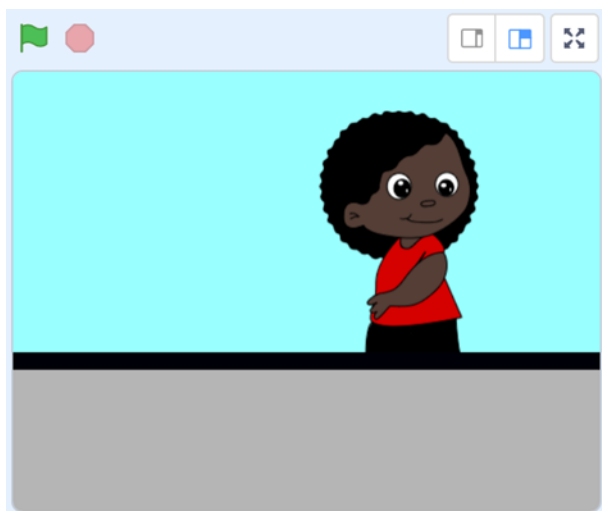
Starter Project 1



Starter Project 2



Starter Project 3



- Paspauskite Next
 - Netikėtas susitikimas (2)
- Vadovaukitės vaizdo įrašė pateiktais nurodymais
- Paspauskite Next
 - Papildomi priedai (3)
- Pasirinkite „Pridėti garsą“ ir vadovaukitės vaizdo įrašė pateiktais nurodymais.



Nustatymas

Start

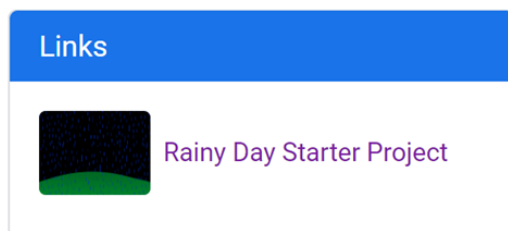
Sukurkite dinamišką audringos dienos aplinką su lietumi ir žaibais.



Užduotis

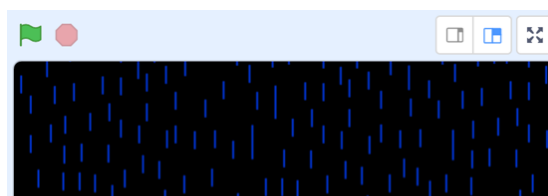
porose

- Paspauskite pradžios mygtuką.
- Atidarykite „Rainy Day Starter“ projektą



- Eikite į
 - Įvadas į nustatymą ir atsitiktinumą (1)
- Žiūrėkite vaizdo įrašą

- ▶ 1. Introduction to Setting and Randomness
- ▶ 2. Make it Rain
- ▶ 3. Lightning Flash
- ▶ 4. Random Lightning
- ▶ 5. Making Your “Stormy Day” Setting into a Story
- ▶ 6. Add-Ons
- ▶ 7. Reflection
- ▶ 8. Wrap-Up: Setting
- ▶ 9. Wrap-Up: Next Steps



- Paspauskite Next
 - Padaryti lietu (2)
- Vaizdo įrašė paaiškinta, kaip į scenarijų įtraukti lietu. Tai daroma naudojant specialų vieną lietaus veikėją, kuriame yra visi lietaus lašai ir kuris užprogramuotas judėti iš viršaus į apačią.
- Vadovaukitės vaizdo įrašė pateiktais nurodymais
- Paspauskite Next
 - Žaibo blyksnis (3)
- Vadovaukitės vaizdo įrašė pateiktais nurodymais
- Paspauskite Next
 - Atsitiktinis žaibas (4)
- Vadovaukitės vaizdo įrašė pateiktais nurodymais
- Eikite į
 - Papildomi priedai (6)
- Peržiūrėkite trumpą vaizdo įrašą ir puslapio apačioje pasirinkite vieną ar du galimus priedus.



3.2 veikla Labirinto žaidimo kūrimas



Įvadas: Algoritmai

Algoritmas – tai instrukcijų seka. Jei algoritmas suformuluotas pakankamai tiksliai, kas nors kitas turėtų sugebėti atlikti žingsnius tiksliai taip, kaip norėjote. Visų pirma algoritmai, skirti kompiuteriui atlikti, formuluojami labai tikslia kalba – programos kodu, kuriame kompiuteriui žingsnis po žingsnio tiksliai nurodoma, ką jis turėtų daryti.



Užduotis porose

Žiūrėkite šį vaizdo įrašą: [Kas yra algoritmas ir kodėl jums tai turėtų rūpėti?](#)



1 veikla: algoritmų nagrinėjimas naudojant srauto diagramas

Panagrinėkime tokį algoritmą.

1. Nubrėžkite vertikalią liniją
2. Nubrėžkite ją kertančią horizontalią liniją
3. Nubrėžkite įstrižą liniją nuo vertikaliosios linijos viršaus iki horizontaliosios linijos kairiojo (arba dešiniojo) galo.
4. Pakartokite 3 instrukciją visiems likusiems linijų galams.
5. Nubrėžkite banguotą liniją nuo apatinio galo.



Grupės užduotis

Pagal šiuos nurodymus nupieškite paveikslėlį.

Palyginkite savo piešinį su kitų piešiniais. Aptarkite, kodėl tai buvo ar nebuvo lengva.

*

Numatytas algoritmo rezultatas buvo aitvaro piešinys (žr. A priedą, kuriame pateiktas galimas rezultatas). Pavyzdys rodo, kad kai instrukcijos yra neaiškios arba neišsamios, rezultatas gali būti dviprasmiškas. Kartais tikimasi, kad atlikėjas pats išspręs visus neaiškumus. Pagalvokite, kas nutiktų, jei visada tiksliai laikytumėmės pateiktų instrukcijų.

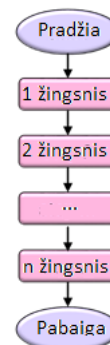


Poros užduotis

Žiūrėkite vaizdo įrašą [tikslias instrukcijas iššūkis](#).

Jei norime aprašyti algoritmą, turime susitarti dėl kalbos, kurią naudosime. Tokia kalba turi būti pakankamai tiksli: turi būti aišku, ką daryti, jei vadovausimės ta kalba parašytu algoritmu. Algoritmas gali būti vizualiai pavaizduotas srautų diagramoje. Tai padeda išlaikyti bendrą vaizdą, o tai savo ruožtu palengvina instrukcijų rašymą, skaitymą ir analizę.

Paprasčiausius algoritmus sudaro viena po kitos vykdomų instrukcijų serija. Tokią instrukcijų seką vaizduojame sekos schema, kaip parodyta dešinėje. Pradedama nuo „Pradžia“ viršuje. Dalyje tarp „Pradžia“ ir „Pabaiga“ (sekos schemas žingsniai) aprašoma, kas turi būti atlikta.



Savarankiškos užduotys

Kvadrato piešimas

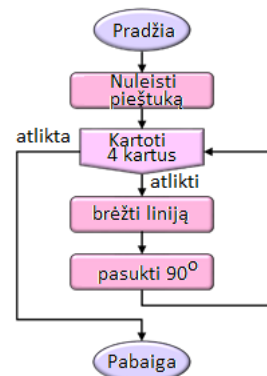
Tarkime, kad galite naudoti šias pagrindines komandas:

- Nubrėškite liniją
- Pasukite 90 laipsnių kampu
- Padėkite rašiklį ant popieriaus

Pasirinkite tinkamus žingsnius ir aprašykite juos sekos schemeje, kad nubrėžtumėte kvadratą.

*

Jūsų sprendimas tikriausiai susideda iš ilgos instrukcijų sekos, kurioje yra pasikartojančių komandų. Tokią seką galime užrašyti kompaktiškiau ir todėl aiškiau, naudodami kartojimo (repeat/while) konstrukta: kol dar neatlikome 4 pakartojimų, laikomasi šakos „atlikti“. Po ketvirtos iteracijos sekame šaką „atlikta“ ir algoritmas baigiamas.



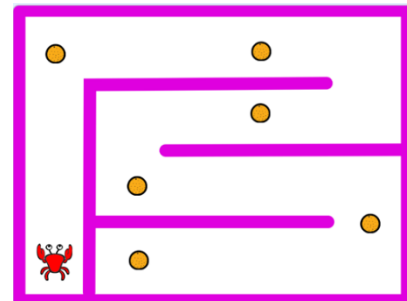
Poros užduotis

- vienas besimokantysis pasirenka paveikslėlį ir parašo instrukcijas, kaip nupiešti paveikslėlį.
- iš anksto susitaria dėl pagrindinių komandų, kurios gali būti naudojamos aprašyme, rinkinio.
- kitas besimokantysis kruopščiai laikosi šių nurodymų.
- aptaria sprendimą ir ištaiso klaidas.



2 veikla: sukurkite labirinto žaidimą naudodami „Scratch“ veikėjus

Šiame etape „Scratch“ programoje sukursite žaidimą, kuriame žaidėjas rodyklių klavišais valdys krabą. Žaidimo tikslas – rasti kelią į kitą labirinto pusę neliečiant sienų. Labirinte išbarstyti apelsinai. Žaidėjas renka taškus priversdamas krabą rinkti apelsinus.



Mokymosi ištekliai

Įbrėžimų failas: crab-in-maze-start.sb3



Poros užduotis

Pabandykite išsiaiškinti, kaip galėtų vykti jūsų žaidimas. Padarykite apytikslį žaidimo planą. Pagalvokite, kaip žaidimas atrodys ir kaip žaidėjas valdys pagrindinį veikėją. Pabandykite suformuluoti sąlygas, kurios parodytų, ar žaidimas baigėsi. Kokių „lobių“ pridėsite į savo žaidimą? Kur jie yra ir kaip žaidėjas juos pasiima?

*

Programuoti šį žaidimą, išmoksite

- pasirinkti veikėjo kaukes, padėtį ir apibrėžti judėjimą.

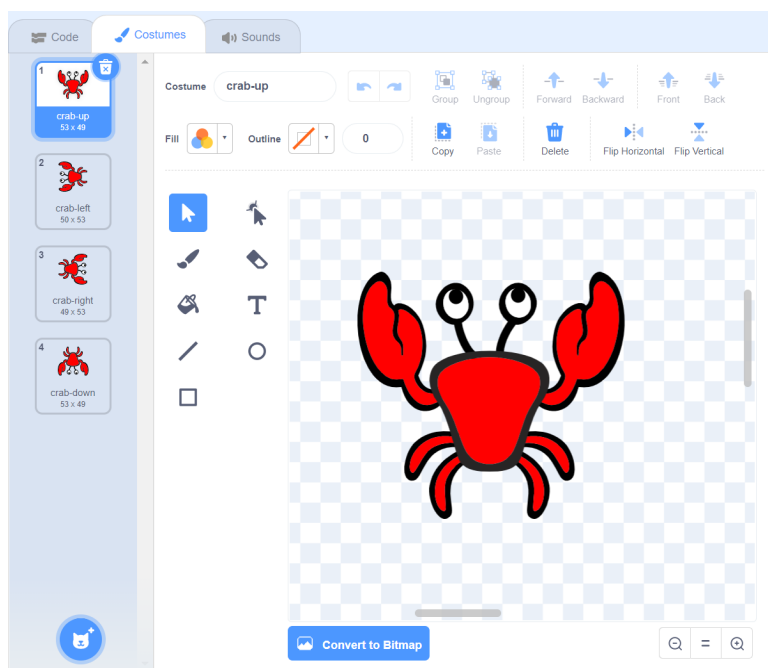
- valdyti veikėją klaviatūros klavišais
- valdyto veikėjo sąveiką su aplinka
- susieti veikėjų valdymo programas tarpusavyje.
- naudoti kintamuosius

Pradėsime nuo pagrindinio žaidimo veikėjo – krabo („Scratch“ veikėjas) – kaukės pakeitimo.



Poros užduotis

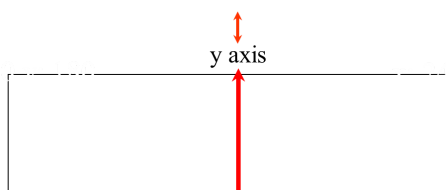
- Eikite į „Scratch“ ir atidarykite failą „crab-in-maze-start.sb3“ (iš meniu „File“ pasirinkę „load from your computer“).
- Pasirinkite krabo veikėją ir spustelėkite skirtuką Kaukės.
- Pašalinkite antrąjį paveikslėlį, pasirinkite pirmąjį paveikslėlį, šiek tiek sumažinkite krabo nagus ir visą paveikslėlį sumažinkite taip, kad krabas patogiai tilptų labirinte (kad sienos nebūtų paliestos).
- Dubliuokite paveikslėlį 3 kartus (dešiniuoju pelės klavišu spustelėkite paveikslėlį ir pasirinkite „dubliuoti“) ir pasukite šias kopijas, kad krabas būtų matomas iš paukščio skrydžio: į kairę, į dešinę, į viršų ir į apačią.



- Atitinkamai pakeiskite pavadinimus.

*

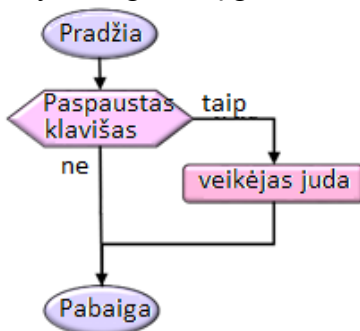
Norint užprogramuoti krabo judėjimą, reikia žinoti, kaip „Scratch“ programoje išdėstomi veikėjai. Scratch kiekvieną žaidimo lauko tašką atitinka koordinatės. Koordinačių plokštumos pradžia yra žaidimo lauko viduryje. Kiekvienas taškas žymimas pora (x, y), kur x yra horizontalus atstumas nuo pradžios, o y - vertikalus atstumas.



Visas žaidimo laukas yra 480 pikselių pločio ir 360 pikselių aukščio. Viršutinio kairiojo kampo ir viršutinio dešiniojo kampo koordinatės yra atitinkamai (-240,180) ir (240,180).

Grįžkime prie mūsų žaidimo. Žaidėjas krabą valdo rodyklių klavišais. Dabar kyla klausimas, kaip savo programoje galime nustatyti, ar žaidėjas paspaudė klavišą, ir kaip tada pagal paspaustą klavišą reguliuoti krabo padėtį,

Paspaustų rodyklių klavišų tvarkymo algoritmą galima nurodyti srauto diagramoje taip:



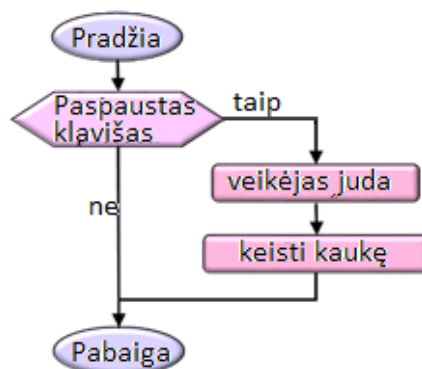
Čia naudojome sąlygą, dar vadinamą konstrukcija „if-then“. Tokioje atrankoje, remiantis sąlygine išraiška (pažymėta šešiakampiame bloke), nustatoma, ar turi būti įvykdyta šaka „tiesas“. Pastarasis veiksmas atliekamas tik tuo atveju, jei sąlyga yra teisinga.



Poros užduotis

- Pabandykite išsiaiškinti, kurį *įvykių bloką* galite naudoti reaguodami į paspaustus rodyklių klavišus
- Tinkamai sureguliuokite krabo padėtį. Kokio *judėjimo bloko* jums reikia?

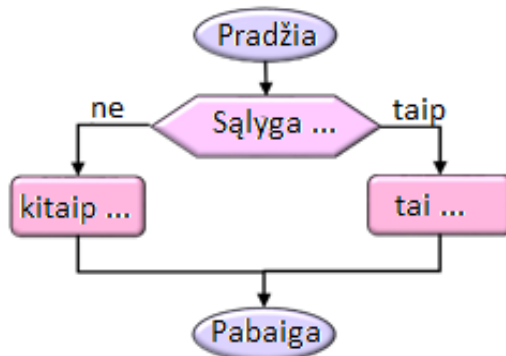
Kad žaidimas būtų tikroviškesnis, padarysime taip, kad krabo vaizdas atitiktų jo judėjimo kryptį. Tai padarysime visada pasirinkdami tinkamą kaukę:



- Pasirinkite, kokio *išvaizdos bloko* jums reikia.
- Paleiskite ir išbandykite savo kodą.

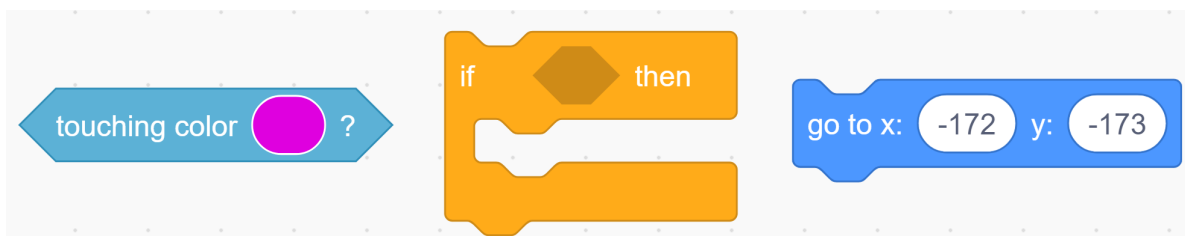
*

Bendras sąlygos sakinio variantas suteikia galimybę atlikti veiksmą ir tuo atveju, jei sąlyga yra klaidinga. Ankstesniame pavyzdyje, esant klaidingai sąlygai, nieko neįvyko. Srautinėje schemoje teiginys „if-then-else“ nurodomas taip.



Jei sąlyga yra teisinga, vykdoma „tai ...“ šaka (pažymėta „taip“). Priešingu atveju vykdoma „kitaip ...“ šaka (pažymėta „ne“).

„Scratch“ turi šiuos komandų blokus, skirtus pelei judinti ir tikrinti, ar krabas pataiko į labirinto sienas.



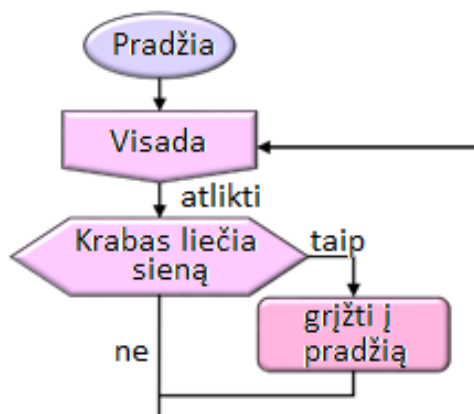
Užduotis porose

Nustatykite, kokių blokų jums reikia ir kokia turi būti blokų eilės tvarka.

- Jei krabas paliečia labirinto šonus, kaip perkelti krabą atgal į pradžią.
- Į savo kodą įtraukite atitinkamus komandų blokus
- Paleiskite ir išbandykite savo kodą.
 - Ar tai veikia? Jei ne, pabandykite išsiaiškinti, kas negerai, ir ištaisykite tai.

*

Jau matėme, kaip sekų schema galima išreikšti, kad tam tikrą algoritmo dalį reikia pakartoti keletą kartų. Gali būti, kad norime nurodyti, jog algoritmas turi būti kartojamas visada ir todėl niekada nesibaigia. Savo žaidime su krabais tai galime panaudoti norėdami vis iš naujo patikrinti, ar krabas paliečia sienas ir, jei reikia, turi būti grąžintas į pradžią. Schema:



Atkreipkite dėmesį, kad ši schema neturi pabaigos.



Užduotis porose

- Krabo judėjimui valdyti naudokite nesibaigiantį ciklą (iš *valdymo bloko*).
- Sužinokite, ką reikia daryti, jei krabas atsitrenkia į vieną iš sienų.
- Paleiskite ir išbandykite savo kodą.

*

Natūralus būdas spręsti dideles problemas – išskaidyti jas į keletą uždavinių, kuriuos galima spręsti daugiau ar mažiau nepriklausomai, o paskui sujungti ir rasti bendrą sprendimą. Algoritmo projektavimo etape, kai problema suskaidoma į dalinius uždavinius, problemos sprendėjas turėtų atsižvelgti tik į tai, ką dalinis algoritmas turi daryti, ir nesirūpinti jo detalėmis. Toks sprendimo etapų atskyrimas vadinamas dekompozicija, o algoritmo apjungimas nesigilinant į mažesnių dalių struktūrą – abstrakcija. Abstrakcijos proceso metu programuotojas gali paslėpti visas subalgoritmų detales, kad sumažintų sudėtingumą ir taip padarytų programą suprantamesnę.

„Scratch“ naudoja abstrakcijas, kad palengvintų programavimą, siūlydama daugybę pagrindinių komandų bloką, kurių sudėtingos pagrindinės realizacijos lieka paslėptos. „Scratch“ programuotojai gali įvesti abstrakcijas patys į programą įtraukdami naujus komandų blokus.

Pavyzdžiui, panagrinėkite toliau pateiktas programų ištraukas, kurios abi gali būti ankstesnės problemos sprendimas. Kairėje pusėje matote sprendimą, kuriame viskas išspręsta paprastai. Dešinėje naudojama abstrakcija, įvedant krabo perkėlimą ir kaukės keitimą kaip naują komandos bloką (su pavadinimu „Start“) ir tada jį naudojant pagrindinėje programoje.

```
when clicked
  go to x: -172 y: -173
  switch costume to crab-up
  forever
    if touching color pink ? then
      go to x: -172 y: -173
      switch costume to crab-up
```

```
when clicked
  start
  set Score to 0
  forever
    if touching color pink ? then
      start
```



Užduotis porose

Peržiūrėkite „Scratch“ scenarijų ir taikykite abstrakciją, pakeisdami (išsamius) kodo fragmentus savarankiškai apibrėžtais komandų blokais.

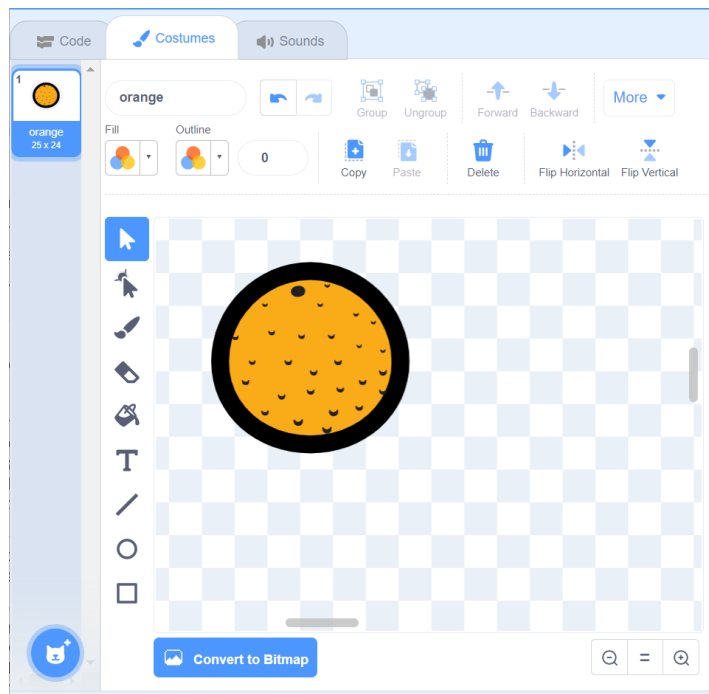
*

Dabar žaidėjas gali valdyti krabą rodyklių klavišais, o krabas teisingai reaguoja, kai pataiko į labirinto sieną. Dabar žaidimą padarysime patrauklesnį pridėdami iššūkių. Iš pradžių apsiribosime apelsiniais, kuriuos paskleisime po labirintą ir kuriuos suvalgys krabas. Tai įvyksta, kai tik krabas juos paliečia. Suvalgius apelsiną, gaunamas papildomas taškas. Dabar žaidimo tikslas – surinkti kuo daugiau papildomų taškų.



Užduotis porose

- Pasirinkite oranžinį veikėją ir padidinkite jį taip, kad jo dydis būtų proporcingas krabo



dydžiui.

Dabar susiduriame su įdomia problema: kas turi būti atsakingas už krabų ir apelsinų sąveikos nustatymą ir tvarkymą? Realiame pasaulyje tai visada bus krabas, tačiau čia turime galimybę rinktis. Apelsinui galime suteikti daugiau atsakomybės nei tikrovėje. Dabar pasinaudosime šia įdomia galimybe: priversime apelsiną aptikti, ar jį surado krabas, ir leisime save suvalgyti. Pastarąjį tikslą pasieksime paslėpdami apelsiną.

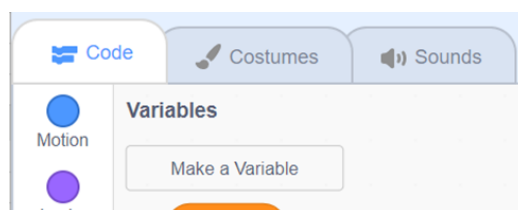
- Žemiau matote blokus, kurių (tikriausiai) prireiks apelsinams programuoti. Naudokite šiuos blokus, kad gautumėte norimą apelsino elgesį.



- Paleiskite ir išbandykite savo kodą!
 - Ar šiuo metu reikėtų dar ką nors daryti?

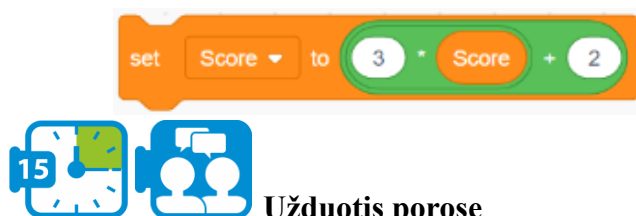
*

Taip pat norime į programą įtraukti kokį nors būdą, kaip sekti rezultatą. Šiam tikslui naudosime kintamąjį. Programavimo kalbose kintamasis – tai talpykla, kurioje vienu metu gali būti laikoma viena informacija, pavyzdžiui, žodis arba skaičius. Galimybė laikyti šią informacijos dalį leidžia mums į ją atsiremti ir ją manipuluoti įvairiose programos vietose. Dėl šios savybės kintamieji yra nepaprastai naudingi.



Kaip „Scratch“ programoje sukurti kintamąjį? Prieš pradėdami naudoti kintamąjį, pirmiausia turime jį sukurti naudodami komandų blokų mygtuką „Make a Variable“ (sukurti kintamąjį). „Scratch“ kintamojo reikšmė gali būti tekstas arba skaičius. Be to, yra blokų, skirtų kintamajam suteikti (pradinę) reikšmę ir ją keisti vykdant programą. Dešinėje matote, kokius blokus galite naudoti kintamiesiems tvarkyti. Beje, reikšmės laukas, pavyzdžiui, bloke „set“ (nustatyti), nebūtinai turi būti skaičius: čia galima naudoti ir sudėtingesnes išraiškas, sudarytas naudojant operatorius iš operatorių paletės. Pavyzdžiui, tarkime, kad parametro „Score“ reikšmė yra 4. Įvykdžius

Rezultato reikšmė bus $3 * 4 + 2 = 14$.



Užduotis porose

- Į savo programą įtraukite kintamąjį, kurio pavadinimas „Taškai“.
- Pagalvokite, kokią reikšmę šis kintamasis turi pradžioje, ir apsvarstykite, kaip galite nustatyti šią reikšmę.
- Taip pat apsvarstykite, kur ir kaip reikėtų koreguoti taškų skaičių ir kokį bloką tam galima naudoti.
- Paleiskite ir išbandykite savo kodą!
- Dubliuokite oranžinį veikėją („sprites lange“) keletą kartų (pvz., 6) ir padėkite šias kopijas kur nors labirinte.



4 dalis: Informatinio mąstymo mokymas ir mokymasis

Bendras laikas: 8 valandos

Informatinio mąstymo mokymas ir mokymasis gali vykti iš esmės dviem būdais: be kompiuterio ir su kompiuteriu. Šiame skyriuje išbandysite abu būdus ir sužinosite apie kompiuterinio mąstymo mokymo strategijų projektavimo principus.



Indėlis į mokymosi rezultatus

Mokymosi rezultatai

- apibūdinti mokymo ir mokymosi veiklų be kompiuterio ir su kompiuteriu, susijusių su informatiniu mąstymu, ypatumus
- apibūdinti esminius informatinio mąstymo mokymo strategijų elementus
- atpažinti tokius elementus konkrečioje mokymo medžiagoje ir veikloje.



4.1 veikla „Bebras“ užduotys



Bebrai

"Bebras" – tai tarptautinės bendruomenės organizuojamas internetinis konkursas ir iššūkis. Konkursų elementai yra vadinamosios „Bebras“ užduotys, kurių kiekviena apima vieną ar daugiau informatikos sąvokų. Šioje veikloje nagrinėsite „Bebras“ užduotis kaip konkurso vienetus ir kaip mokymosi veiklą, skirtą informatiniam mąstymui lavinti.



International Challenge on Informatics
and Computational Thinking



Užduotis porose

- Eikite į „Bebras“ svetainę ir pasirinkite iki penkių pavyzdinių „Bebras“ užduočių (pasistenkite, kad jos būtų įvairios).
- Atlikite užduotis individualiai, o tada palyginkite ir aptarkite jas su partneriu.

- Ar galite suskirstyti užduotis pagal šiame modulyje išmokus informatinio mąstymo etapus ir veiklas?



„Bebras“ užduotys kaip mokymosi veikla

„Bebras“ užduotys suskirstytos į kategorijas pagal nagrinėjamas skaičiavimo sąvokas, o prie kiekvienos užduoties yra paaiškinimas apie užduoties ryšį su informatikos mokslu.

Valentina Dagienė ir Sue Sentance tyrinėjo "Bebras" užduočių naudojimą kaip IM mokymosi veiklą:

Dagienė, V., & Sentance, S. (2016). It's Computational Thinking! Bebras tasks in the curriculum. In International conference on informatics in schools: Situation, evolution, and perspectives (p. 28-39).



Užduotis porose

- Skaityti straipsnį.
- Peržiūrėkite sąvokas ir užduotis, su kuriomis susidūrėte 2 ir 3 skyriuose. Kiekvienam iš šių modulių raskite informatikos sąvokos pavyzdį ir atitinkamą „Bebras“ užduotį.



4.2 veikla Kelio ieškojimas: keliai labirinte

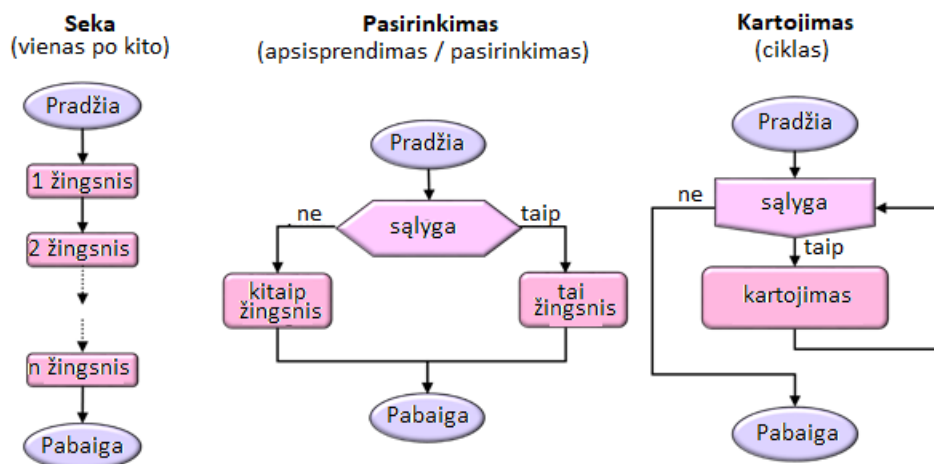
Labirinto sprendimas – tai procesas, kurio metu surandamas kelias per labirintą nuo pradžios iki galo. Kai kurie labirinto sprendimo metodai yra skirti naudoti labirinto viduje keliautojui, neturinčiam jokių išankstinių žinių apie labirintą, o kiti – asmeniui (arba kompiuterinei programai), kuris gali matyti visą labirintą iš karto. Labirintų sprendimas yra praktiškesnės klasės problemų, kurios žinomos kelio paieškos pavadinimu, variantas. Turint kelias vietas, kurios gali būti tarpusavyje susijusios (pavyzdžiui, keliais sujungti miestai), kelio ieškojimo algoritmas ieško maršruto, jungiančio šias vietas, siekdamas nustatyti geriausią maršrutą (pvz., trumpiausią arba pigiausią maršrutą).

Šiais kelio paieškos uždaviniais siekiame įgyti pirmųjų įžvalgų apie problemų analizę ir algoritmų kūrimą be išsamių kompiuterių algoritmovimo ir programavimo kalbų žinių. Tačiau tam, kad galėtume pakankamai tiksliai suformuluoti galimus sprendimus, pirmiausia vėl naudosisime algoritmų schemas.



Algoritmo schemas elementai

Toliau matote tris pagrindinius blokus, iš kurių gali būti sudaryta algoritmo schema.



Kiekviena algoritmo schema sudaroma naudojant šiuos tris pagrindinius elementus

1. **Seką:** viena po kitos vykdomų instrukcijų seka.
2. **Pasirinkimas:** remiantis sąlygine išraiška nustatoma, ar turi būti vykdoma tai-šakos (šakos su etikete „taip“), ar kitaip-šakos (šakos su etikete „ne“).
3. **Kartojimas:** elemento (šaka su etikete „taip“) kartojamas tol, kol sąlyga vertinama kaip „taip“. Tada programa tęsiama pagal šaką, pažymėtą „ne“.



Keliaujančio pirklio uždavinys

Vadinamasis keliaujančio pirklio uždavinys yra klasikinis uždavinio, kuriam ieškoma optimalaus sprendimo, pavyzdys.

Koks yra trumpiausias įmanomas maršrutas tarp miestų, su nurodytais atstumais tarp kiekvienos miestų poros, kai norima aplankyti kiekvienas miestą po vieną kartą ir grįžti į pradinį miestą?

Dabar apžvelgsime šio uždavinio sprendimą, kuris, nors ne visada randa geriausią maršrutą, yra intuityvus ir lengvai suprantamas, t. y. *artimiausio kaimyno algoritmas*. Artimiausio kaimyno algoritmas (NNA) leidžia pikliui pasirinkti artimiausią neaplankytą miestą kaip kitą žingsnį.



Užduotis porose

Bebrai plaukia iš A miesto, jie aplanko B, C ir D miestus ir grįžta atgal į A. Jie nori rasti trumpiausią kelią.



1. Kokį maršrutą jie ras, jei naudosis *artimiausio kaimyno algoritmą*?
2. Ar *artimiausio kaimyno algoritmas* visada pateikia teisingą atsakymą (t. y. bet kokio miestų ir jungčių tarp jų sąrašo atveju)? Jei ne, pateikite priešingą pavyzdį

*

Nors *artimiausio kaimyno algoritmas* intuityviai aiškus, gali būti sunku tiksliai suformuluoti šį algoritmą. Šį algoritmą pabandykite pavaizduoti schema, kurioje pirmiausia turime nustatyti, kokias pradines instrukcijas galime naudoti. Šios instrukcijos, viena vertus, turi būti pakankamai galingi, bet, kita vertus, ir pakankamai abstraktūs, kad nepasiklystume įvairiose detalėse.



Užduotis porose

Pavaizduokite *artimiausio kaimyno algoritmą* schema.

Patarimas: pirmiausia pagalvokite apie pagrindines instrukcijas ir (arba) pradinius duomenis, kuriuos leidžiama naudoti.



Kelio paieška labirinte

Dar vienas pavyzdys, kai užduotį lengva suprasti, bet taip pat pakankamai sudėtinga išspręsti, - rasti kelią labirinte. Yra keli šio uždavinio variantai: rasti kelią iš labirinto, rasti kelią per labirintą arba rasti kelią į tam tikrą vietą labirinte. Pirmiausia galime pastebėti, kad šiuos variantus galima apibendrinti taip: rasti kelią iš padėties A į padėtį B.

Taip pat svarbu, ką mes žinome apie labirintą, kol ieškome tikslo B. Šioje užduotyje darome prielaidą, kad nieko nežinome apie labirinto dydį ir struktūrą. Galime vaikščioti labirinto koridoriais, žvelgdami po vieną žingsnį į priekį. Todėl darome prielaidą, kad bet kuriame

labirinto taške galime patikrinti, ar galime eiti tiesiai (taigi nestovėti priešais sieną), ir galbūt pakoreguoti savo judėjimo kryptį.

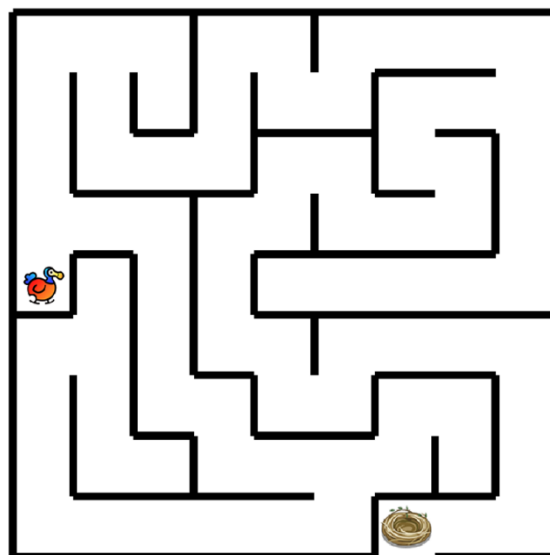
Geriausiai žinoma labirinto įveikimo taisyklė yra sienos sekimo taisyklė (dar vadinama kairės arba dešinės rankos taisykle). Jei visos labirinto sienos yra sujungtos tarpusavyje, tuomet keliautojas, viena ranka prisiliedamas prie vienos labirinto sienos, garantuotai nepasiklys ir pasieks išėjimą.

Kad problema būtų konkretesnė, naudojame tokį scenarijų, kuriame labirinto keliautoją vaizduoja antis Dodo, kuris yra kažkur labirinte ir bando rasti savo lizdą, paslėptą kitoje vietoje.

Prieš aprašydami paieškos algoritmą, turime dar kartą nurodyti, kokiomis pradinėmis instrukcijomis galime valdyti Dodo.

Skiriame *komandas* (kad Dodo žengtų žingsnį į priekį arba pakeistų judėjimo kryptį) ir *užklausas* (jomis Dodo gali pateikti vietinę informaciją apie aplinką).

- Komandos:
 - *pasukti į kairę*: pasukti 90 laipsnių kampu prieš laikrodžio rodyklę.
 - *pasukti į dešinę*: pasukti 90 laipsnių kampu pagal laikrodžio rodyklę.
 - *žingsnis*: žingsnis vienu kvadratu į priekį
 - *dėti kiaušinį*: dėti kiaušinį (dabartinėje vietoje).
- Užklauskos / bandymai:
 - *ar galite judėti?* ar galite žengti žingsnį į priekį?
 - *ar radote lizdą?* ar radote lizdą?



Užduotis porose

Schema pavaizduokite sekimo kairiaja siena strategiją.



Kelio ieškojimas be kompiuterio

Paruošimas

Pavyzdžiui, naudodami dailininko juostą, ant grindų nupieškite labirintą. Pasirūpinkite, kad koridoriai būtų pakankamai platūs, kad žmogus galėtų lengvai praeiti. Labirintas gali būti šiek tiek paprastesnis nei pavaizduotas paveikslėlyje. Pasikeiskite ankstesnėje užduotyje pateiktomis

schemomis. Nurodykite vietą labirinte kaip galutinę paskirties vietą, o vienas iš jūsų bus kažkur labirinte pakankamai toli nuo šios galutinės paskirties vietos.



Užduotis porose

Vienas iš jūsų stovės kur nors labirinte, pakankamai toli nuo galutinio tikslo.

- Kitas asmuo perskaito schemoje pateiktus nurodymus, o tas, kuris yra labirinte, kruopščiai jų laikosi.
- Ar pasiektas galutinis tikslas? Jei taip, taip pat išbandykite, ar tai galioja ir kitoms pradinėms labirinto vietoms. Jei ne, kas negerai su algoritmu? Patobulinkite algoritmą ir patikrinkite, ar patobulinimas duoda norimą rezultatą.
- Gražinkite patobulintą schemos versiją jos savininkams.



Mokymosi ištekliai

- Labirinto failas: PathThroughMaze.sb3



Kelio ieškojimas programoje „Scratch“

Galiausiai ankstesnėje užduotyje pateiktą algoritmą įgyvendinsite „Scratch“ programoje. Sukūrėme pradinį scenarijų, kurį galite naudoti šiam tikslui. Atidarykite pradinį scenarijų „PathThroughMaze.sb3“.

Toliau pateikiamas trumpas paaiškinimas. Pradinės Dodo instrukcijos buvo įtrauktos į scenarijų kaip atskiri blokai. Instrukcijos *layEgg* nėra, nes ji nereikalinga šiai užduočiai atlikti. Visos kitos komandos veikia taip, kaip nurodyta anksčiau. Užklausa buvo įgyvendinta specialiu būdu, taip pat ir todėl, kad savarankiškai apibrėžti blokai „Scratch“ programoje negali duoti jokių rezultatų. Testą, skirtą patikrinti, ar Dodo rado lizdą, galima lengvai realizuoti naudojant iš anksto apibrėžtą bloką iš jutimo komandų grupės. Įvedėme naują bloką *canMove* testui. Rezultatui gražinti naudojame kintamąjį, kurį, kaip ir patį bloką, taip pat pavadiname *canMove*. Taigi įvykdę *canMove* bloką galite patikrinti atitinkamo kintamojo reikšmę ir sužinoti, ar Dodo gali žengti žingsnį, ar ne. Kintamajame *canMove* yra reikšmė 0 arba 1. Reikšmė 0 reiškia, kad žingsnio žengti negalima, o reikšmė 1 rodo, kad žengti žingsnį galima.



Užduotis porose

Pradiniame scenarijuje yra blokas, vadinamas *tryToFindExit*. Šis blokas vis dar nedaug ką daro: ekrane tik pateikia pranešimą, kad išėjimas rastas.

- Įgyvendinkite šį bloką konvertuodami schemą į „Scratch“ kodą.
- Paleiskite ir išbandykite savo programą ir kitoms Dodo pradinėms padėtims. Jei reikia, pakoreguokite kodą.



4.3 veikla Mokymo strategijos

Šioje veikloje nagrinėsite skaičiavimo mąstymo strategijas ir pedagoginius principus.



Žvilgsnis atgal



Užduotis porose

Aptarkite savo patirtį, įgytą atliekant užduotį „Kelio paieška“:

- Kokie yra esminiai veiklų mokymosi be kompiuterio ir su kompiuteriu skirtumai?
- Kokių iššūkių būsimiems mokiniams tikėtės kiekvienoje iš tokių veiklų?



Informatinio mąstymo mokymo principai

Lye ir Koh (2014) apžvalginiame straipsnyje analizuoja 27 empirinius kompiuterinio mąstymo tyrimus, ypač tyrimus, susijusius su programavimo veikla klasėje. Be kita ko, Lye ir Koh nurodo mokymo metodus, skatinančius informatinį mąstymą.

Lye ir Koh sugrupavo metodus į keturias kategorijas, kurios dažnai būna derinamos: *informatikos sąvokų įtvirtinimas*, *apmąstymas*, *informacijos apdorojimas* ir *programų konstravimas*.



Užduotis porose

1. Perskaitykite dokumento „Kompiuterinio mąstymo mokymo strategijos“ (jį galite rasti mokymosi šaltiniuose) keturių kategorijų santrauką.
2. Išanalizuokite kompiuterinio mąstymo veiklą, kurią patys atlikote šiame modulyje. Kurias strategijas atpažįstate? Aptarkite.



Užduotis porose

1. Perskaitykite vertinimo informatinio mąstymo srityje santrauką (žr. dokumentą „Vertinimas informatinio mąstymo srityje“).
2. Aptarkite, kokie vertinimo metodai būtų tinkami 2 arba 3 skyriuje atliktai veiklai. Kiekvienai veiklai pasirinkite po vieną galimą metodą. Kaip taikytumėte šiuos metodus, kad įvertintumėte mokinių informatinio mąstymo įgūdžius?



Palyginkite savo išvadas klasėje. Kurie aspektai jums pasirodė lengvi, o kurie sudėtingi? Ko išmokote?



- A. Kompiuterinio mąstymo mokymo strategijos (Kompiuterinio mąstymo mokymo ir mokymosi strategijų santrauka, parengta pagal Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- B. Vertinimas kompiuterinėje tomografijoje



- 1.1 veikla. – A. Trumpas IM pristatymas
- 1.1 veikla. – B. IM apibrėžtys
- 2.1 užduotis – A. Bendruomenė į visuomenę.pdf
- Veikla 2.1 – B. Kultūros psichologijos pokyčiai 1800-2000 m.
- 2.1 – C. Kintanti kultūros psichologija vokiškai kalbančiose šalyse
- 2.2 veikla – zikavirusmodel.nlogo
- Užduotis 2.4 – Žemės spalvų sąsiuvinis.xlsx
- 3.2 veikla – crab-in-maze-start.sb3
- 4.1 veikla – Dagiene and Sentance 2016 (Dagienė, V., & Sentance, S. (2016). It's Computational Thinking! Bebras tasks in the curriculum. In *International conference on informatics in schools: Situation, evolution, and perspectives* (pp. 28–39).
- 4.2 veikla. Kelias per labirintą.sb3
- 4.3 – A. Kompiuterinio mąstymo mokymo strategijos (Kompiuterinio mąstymo mokymo ir mokymosi strategijų santrauka, parengta pagal Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- 4.3 veikla – B. Vertinimas IM