



**Vilniaus
universitetas**

Metodinė medžiaga Vaizdų kodavimas

Mokyklos pedagogika



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

Metodinė medžiaga. Vaizdų kodavimas

Informatikos ir informatinio mąstymo veiklos, metodinė medžiaga sukurta įgyvendinant projektą „Aukštųjų mokyklų tinklo optimizavimas ir studijų kokybės gerinimas Šiaulių universitetą prijungiant prie Vilniaus universiteto“, projekto Nr. 09.3.1-ESFA-V-738-03-0001, vykdomą pagal 2014–2020 metų Europos Sąjungos fondų investicijų veiksmų programos 9 prioriteto „Visuomenės švietimas ir žmogiškųjų išteklių potencialo didinimas“ 09.3.1-ESFA-V-738 įgyvendinimo priemonę „Aukštųjų mokyklų tinklo tobulinimas“, finansuojamą Europos Sąjungos fondų ir Lietuvos Respublikos valstybės biudžeto lėšomis.

Metodinės medžiagos **Vaizdų kodavimas** tikslas yra supažindinti pedagogus su vaizdų kodavimo ir pateikimo kompiuteriu pagrindiniais principais bei padėti pasirengti mokomosioms veikloms šia tema klasėje. Veikla skirta Mokyklos pedagogikos studijų programos moduliui „Informatikos didaktika“. Veikla ugdomi būsimųjų (arba dirbančiųjų) pedagogų vaizdų (paveikslų) pateikimo kompiuteriuose principų suvokimo, taškinės grafikos vaizdų kodavimo, dvejetainių skaičių ir dvejetainio kodavimo taikymo, vaizdų glaudinimo, spalvų kodavimo gebėjimai ir pasirengimas šioms mokomosioms veikloms klasėje, dirbant su 1–8 klasės mokiniais. Veikloje numatomi ryšiai su informatikos ir matematikos Bendrosiomis programomis, pateikiamas teorinis temos pagrindimas mokytojui, aptariamos pagrindinės srities sąvokos. Išsamiai aptariamas vaizdų kodavimas skaitmeniniuose įrenginiuose ir pateikiamos praktinės veiklos mokiniams darbo lapai.

Šios veiklos autoriai: dr. Tatjana Jevsikova

Redagavo: Viktoras Dagys

Projekto vykdytojas: Vilniaus universitetas.

2022, Vilnius

TEMA: VAIZDŲ KODAVIMAS

Tikslas

Susipažinti su vaizdų kodavimo ir pateikimo kompiuteriu pagrindiniais principais bei pasirengti mokomosioms veikloms šia tema klasėje.

Ryšys su bendrosiomis programomis

(remiantis bendrųjų programų projektais)

Informatika. Duomenų tyryba ir informacija

C1. Įžvelgia duomenų ryšį su algoritmais, išmano ir vartoja šių sričių sąvokas

1–2 klasė: Pateikia duomenų pavyzdžių.

3–4 klasė: Nagrinėja duomenų vaizdavimo kompiuteriuose pavyzdžius.

5–6 klasė: Aptaria duomenų kodavimą, dvejetainius skaičius.

7–8 klasė: Tyrinėja įvairaus tipo duomenų kodavimą kompiuteriuose.

C2. Tyrinėja duomenis ir atlieka veiksmus su jais

3–4 klasė: Pastebi ir tyrinėja dėsningumus duomenyse.

5–6 klasė: Įžvelgia glaudinimo prasmę, glaudina grafinius duomenis.

7–8 klasė: Aptaria duomenų glaudinimo problemas, sprendžia įvairaus tipo duomenų glaudinimo uždavinius.

Matematika

1. Skaičiai ir skaičiavimai

1–6 klasė: 1.1. Natūralieji ir sveikieji skaičiai

2. Modeliai ir sąryšiai

1–5 klasė: 2.1. Dėsningumai

3. Geometrija ir matavimai

1–8 klasė: 3.2. Konstravimas

1–8 klasė: 3.3. Figūros

Ugdomi įgūdžiai

Vaizdų (paveikslų) pateikimo kompiuteriuose principų suvokimo

Taškinės grafikos vaizdų kodavimo

Dvejetainių skaičių ir dvejetainio kodavimo taikymo

Vaizdų glaudinimo pradmenų

Spalvų kodavimo pradmenų

Mokinių amžius

1–8 klasės

TEORINIS PAGRINDIMAS MOKYTOJUI

Pagrindinės sąvokos

Vaizdas, vaizdai – tai, kas rodoma kompiuterio ar kito skaitmeninio įrenginio ekrane. Apima viską, kas matoma ekrane, pvz., vaizdo įrašus, svetaines, programas, tekstą, o ne tik statines nuotraukas ar piešinius.

Paveikslas – grafinės iliustracijos (brėžinio, paveikslo, schemos, fotografijos, piešinio, portreto) bendras pavadinimas nekonkretizuojant jos tipo.

Taškinė grafika – tai grafika, kurios paveikslo sudaryti iš mažų taškų. Dar vadinama *rastrine grafika*.

Taškas – mažiausias taškinės grafikos elementas. Dar vadinamas *pikseliu*.

Raiška (skiriamoji geba) – rodomo arba spausdinamo vaizdo taškų skaičius, tenkantis vienam ilgio vienetui. Kuo didesnis taškų skaičius (smulkesni taškai), tuo didesnė skiriamoji geba bei dailesnis ir tikslesnis vaizdas.

Duomenys – kompiuterio apdorojami objektai – visa, kas laikoma kompiuterinėse laikmenose. Duomenimis išreiškiama apdorojama informacija.

Duomenų apdorojimas – tai duomenų tvarkymas, pertvarkymas ir kitokių veiksmų su jais atlikimas. Tai visi taikomieji veiksmai, kuriuos atlieka kompiuteris ar kitos skaitmeninės technologijos, nes jų paskirtis ir yra duomenų apdorojimas.

Kodavimas – veiksmas, kuriuo objektui suteikiamas kodas (objekto ar jo savybės žymėjimas skaičiumi arba tam tikros abėcėlės ženklais).

Glaudvinimas – duomenų pertvarkymas taip, kad jie užimtų mažiau vietos atmintyje ir jiems persiųsti reikėtų siauresnės dažnių juostos arba persiuntimas būtų spartesnis. Duomenyse būna vienodų arba panašių fragmentų ir kitokio informacijos pertekliaus. Glaudinant pavyksta rasti dalį informacijos pertekliaus ir jį sumažinti ekonomiškiau koduojant duomenis. Dėl to sutaupoma jiems reikalingos atminties.

Metaduomenys – duomenys apie kitus duomenis. Pavyzdžiui, skaitmeninį mokymosi išteklių aprašantys duomenys (autorius, išleidimo data, mokinių amžiaus grupė, failo tipas, naudojimo trukmė ir pan.).

Įvadas į vaizdų kodavimą kompiuteriuose

Su vaizdais kompiuteriuose ir kituose skaitmeniniuose įrenginiuose susiduriame nuolat. Didžioji dauguma mūsų naudojamų skaitmeninių įrenginių turi tam tikrą ekraną. Viskas, ką matome skaitmeninių įrenginių ekranuose (nuotraukos, vaizdo įrašai, svetainės, net tekstas ir t. t.) yra grafinis vaizdas, kurį skaitmeninis įrenginys buvo suprogramuotas rodyti. Kadangi duomenys kompiuteriuose saugomi skaitiniu pavidalu (prisiminkime dvejetainį kodavimą), vaizdai taip pat galiausiai pateikiami kompiuteryje naudojant tik 0 ir 1 sekas.

Šioje metodinėje medžiagoje žodžiai *vaizdas*, *vaizdai* vartojami nurodant tai, kas rodoma kompiuterio ar kito skaitmeninio įrenginio ekrane. Tai apima viską, kas matoma ekrane, pvz., vaizdo įrašus, svetaines, programas, tekstą, o ne tik statines nuotraukas ar piešinius.

Kalbėdami apie nejudančius (statinius) vaizdus (nuotraukas, piešinius, schemas ir t. t.) vartosime bendrą sąvoką *paveikslas*.

Kompiuterių **ekranai** suskirstyti į tinklę mažų kvadratėlių (pikselių, taškų). Kiekvienas iš **pikselių** gali būti skirtingos spalvos, o kadangi pikseliai maži, atskirų taškų ekrane nematome. Jie susilieja ir formuoja rodomą vaizdą. Jei labai atidžiai pažvelgtume į kompiuterio ar televizoriaus ekraną (ypač senesnio modelio), galėtume įžvelgti atskirus pikselius, iš kurių sudarytas bendras vaizdas.

Įprasta **ekrano raiška** yra apie 1920 x 1080 pikselių ar didesnė. Tai reiškia, kad ekrane yra 2 073 600 pikselių – daugiau nei 2 milijonai.

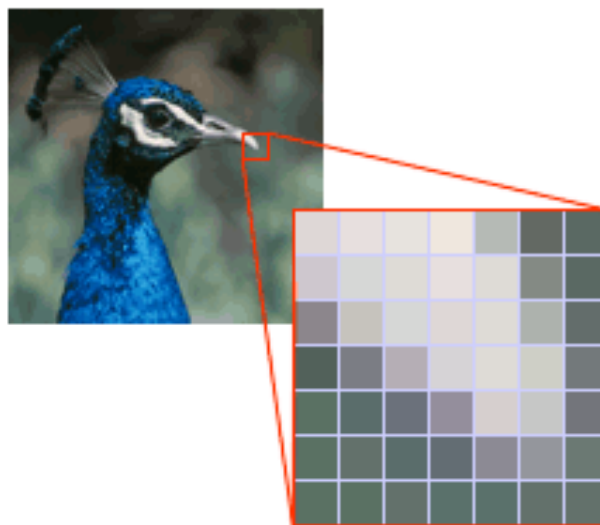
Grafiką, kurios vaizdai, paveikslai sudaryti iš mažų taškų, vadinama *taškine*.

Kiekvienas iš pikselių gali būti tam tikros **spalvos**. Pikselio spalva nurodoma vadinamaisiais *metaduomenimis* – skaitmenų seka, nurodančių kiekvieno pikselio spalvą ar spalvos pasikeitimą. Taip kompiuteriai paverčia skaičių sekas prasmingais vaizdais, kuriuos matome savo ekranuose. Šioje metodinės medžiagos dalyje siūloma praktinė veikla, iliustruojanti spalvų kodavimo principus. Tačiau spalvų modeliai ir jų kodavimo kompiuteriuose mokymo būdai išsamiau nagrinėjami atskiroje metodinės medžiagos dalyje.

Egzistuoja daugybė metodų **vaizdams paversti skaitiniais duomenimis** ir atvirkščiai. Metodas, kurį pasirenkame, priklauso nuo to, į kokio tipo vaizdus reikia paversti duomenis. Nespalvotą vaizdą, kuriame kiekvienas pikselis gali būti tik juodas arba baltas, daug paprasčiau atvaizduoti skaitiniais duomenimis nei spalvotą didelės raiškos grafinį vaizdą.

Kiekvienas taškinės grafikos paveikslas apibūdinamas taškų skaičiumi, pavyzdžiui, 800 taškų pločio ir 600 taškų ilgio. Kiekvienas taškas vaizduojamas vienu arba keliais bitais. Vieno bito

pakanka, kai paveikslas sudarytas vien iš baltų ir juodų taškų (be pustonų ir spalvų). Keturiais bitais galima užkoduoti 16 pilkio tonų arba tiek pat spalvų, 8 bitais – 256 spalvas. Didinant paveikslą, didėja ir jį sudarantys taškai, todėl blogėja paveikslo kokybė.

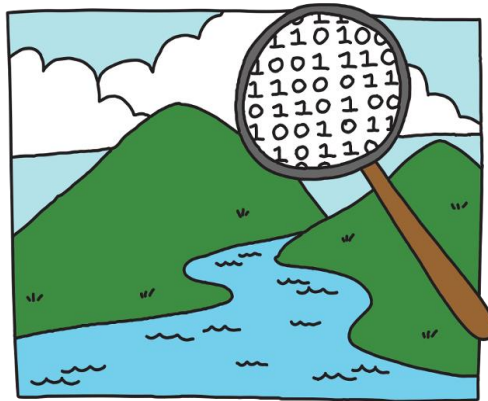


Taškinės grafikos paveikslai dažniausiai gaunami perkeliant realaus pasaulio objektų atvaizdžius į kompiuterį skaitmeniniais fotoaparatais, skeneriais ir kitais skaitmeniniais įrenginiais. Jų failai turi prievardžius BMP, PNG, GIF, TIF, JPEG ir kt.

Kai kurių tipų vaizdų užimamos vietos kiekiui atmintyje sumažinti galime naudoti **glaudinimo algoritmus**. Taip sumažiname paveikslo failo dydį. Tai galima padaryti neprarandant vaizdo kokybės (vad. nenuostolinguoju glaudinimu) arba prarandant nedidelę dalį taškų duomenų (vad. nuostolinguoju glaudinimu). Vaizdų glaudinimo algoritmai leidžia sparčiau saugoti, apdoroti, persiųsti ir kurti vaizdus.

Šioje medžiagoje aptariama, kaip mokiniai paprastais pavyzdžiais gali būti supažindinami su vaizdų kūrimu ir atvaizdavimu kompiuteryje, kaip vaizdo spalvų skaičius priklauso nuo jam saugoti skiriamų bitų skaičiaus, ir kaip galima glaudinti vaizdus, kad jie užimtų mažiau vietos įrenginio atmintyje.

PAŽINTIS SU VAIZDŲ KODAVIMU SKAITMENINIUOSE ĮRENGINIUOSE



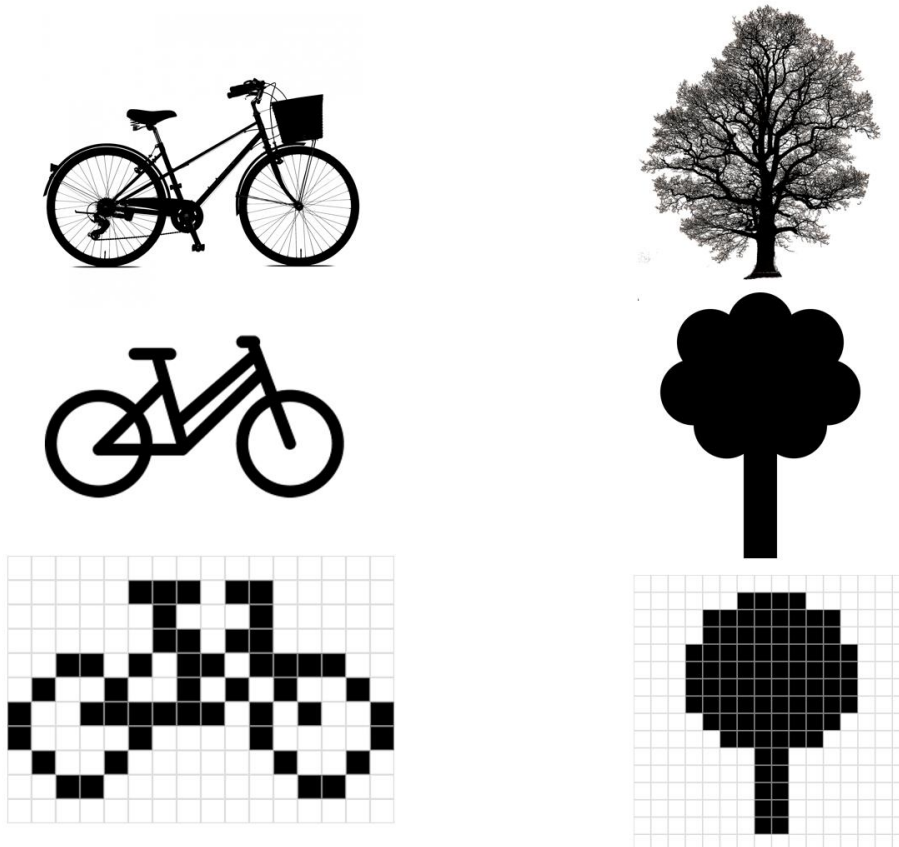
Apšilimo diskusiją galima pradėti rodant mokiniams taškinės grafikos paveikslų (nuotraukų) pavyzdžius.



Galimi klausimai mokiniams:

Pažvelkite į šias dvi nuotraukas. Ką jose būtų sudėtinga kompiuteriui atvaizduoti ekrane? Kaip manote, tokias nuotraukas vaizduoti ekrane lengviau ar sunkiau negu, pavyzdžiui, raides?

Iškėlę hipotezes, parodykite kitus paveikslų rinkinius, kuriuose tie patys objektai vaizduojami supaprastinant jų formą ir parodant vaizdą sudarančius pikselius.



Galimi klausimai mokiniams:

Ar dabar kompiuteriui būtų paprasčiau rodyti paveikslus? Kodėl?

Šie paveikslai skirti pademonstruoti vaizdų **skaidymo į mažesnes dalis** idėją. Mokiniai galėtų pastebėti, kad tai, ką jie mato (paveikslą sudarantys juodi ir balti taškai) atitinka dvejetainę sistemą, todėl prisiminę dvejetainių skaičių taikymo pamokas, jie turėtų sugebėti pavaizduoti tokius paveikslėlius 0 ir 1 sekomis. Nors pastarieji vaizdai yra „prastesnės kokybės“, tas pačias idėjas būtų galima panaudoti kuriant ir sudėtingesnius vaizdus padidinus tinklelio eilučių ir stulpelių skaičių, pavyzdžiui, anksčiau matytus nuotraukų originalus. Juk mes tas nuotraukas jau matome ekrane!

Pademonstruokite mokiniams taškinės grafikos kūrimo principą.

Šiuolaikiniuose ekranuose gali būti sunku įžiūrėti pikselius, nes kokybiškuose ekranuose jie labai maži. Kartais parodyti vaizdo sandarą iš taškų gerai gali tiktį projektorių ekranai,

naudojant didinamąjį stiklą, arba net tiesiog ant kompiuterio ekrano užlašinus vandens lašą, kad būtų sukurtas lęšio efektas. Kad ir ką pasirinktumėte, raskite kokį nors būdą parodyti mokiniams, kad visi vaizdai jų kompiuteriuose iš tikrųjų yra suskaidyti į mažus kvadratėlius.

Demonstravimui taip pat galima naudoti interaktyvią priemonę, pavyzdžiui, „Pixel Viewer“: <http://csfieldguide.org.nz/en/interactives/pixel-viewer/index.html>

Aptarkite su mokiniais, kaip galima būtų nusakyti paveikslą, sudaryto tik iš baltų ir juodų pikselių, piešimą.

Padidinę taškinės grafikos paveikslą matome, kad jis sudarytas iš kvadratėlių. Vienas piešinio kvadratėlis – pikselis, kuris gali turėti 2 būsenas, pavyzdžiui, juodas arba baltas nespaltvotame piešinyje.

Pradėkite nuo diskusijos apie **dvejetainę sistemą**. Priminkite mokiniams, kad viskas, ką rodo ar apdoroja kompiuteris, yra vaizduojama tik vienetais ir nuliais. Prisiminkite su mokiniais tai, ką jie išmoko iš veiklos „Dvejetainiai skaičiai“.

Lentoje nupieškite, pvz., 10×10 langelių tinklę ir parodykite, kaip formuojamas taškinis piešinys. Kiekvienas iš šių langelių būtų paprasčiausiai nuspaltvintas arba ne. Tai atitinka dvejetainės sistemos principus, pvz., galima susitarti, kad kompiuteris tai apdorotų tiesiog kaip „1“, kai langelis nuspaltvintas, arba „0“, kai langelis nenuspaltvintas.

Pavyzdžiui, raidės „c“ vaizdas būtų užkoduotas taip:

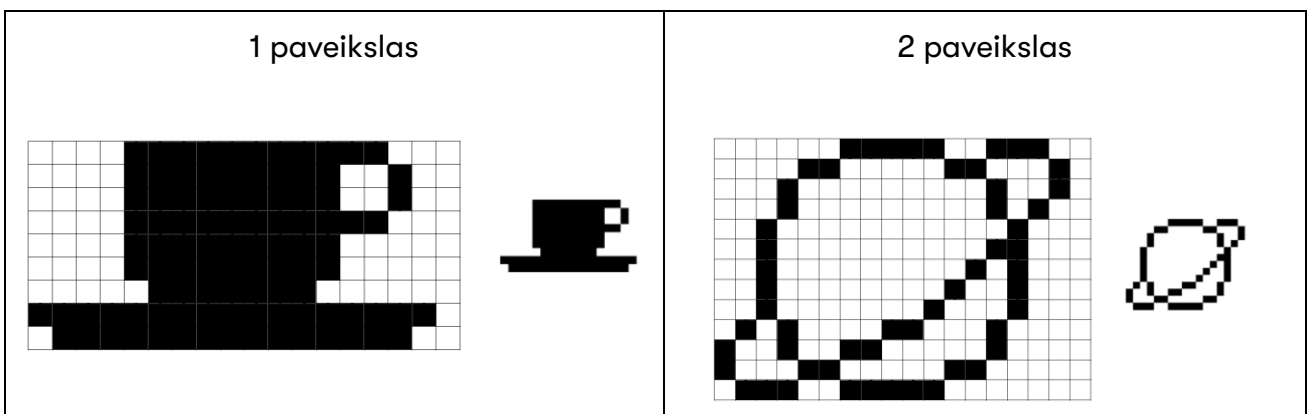
	■	■	■		1 0 0 0 1
■				■	0 1 1 1 0
■					0 1 1 1 1
■					0 1 1 1 1
■				■	0 1 1 1 0
	■	■	■		1 0 0 0 1

Pasiūlykite mokiniams atlikti **piešinių iškodavimo veiklą** remiantis aptartu principu.

Pirmame darbo lape (žr. pridedamą 1-ą darbo lapą) pateiktas dviejų galimų paveikslų iškodavimo ruošinys. Kairėje – tinklė paveikslui piešti pagal kodą, dešinėje – kodas. Nuliai ir vienetai, žymintys langelio spalvinimo pasirinkimą (0 – nespaltvinamas, 1 – spaltvinamas), surašyti kiekvienoje eilutėje skiriant juos kableliais (tai susitarimas, galima kodus rašyti ir be kablelių).

1 paveikslas														Kodas
														0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0
														0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,1,0,0
														0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,1,0,0
														0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0
														0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0
														0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0
														0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0
														0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
														0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0
2 paveikslas														
														0,0,0,0,0,0,1,1,1,1,1,0,0,1,1,1,0,0
														0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,1,0
														0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0
														0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0
														0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
														0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0
														0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0
														0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0
														0,1,0,1,0,0,0,0,1,1,0,0,0,0,1,0,0,0,0
														1,0,0,1,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0
														1,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0
														0,1,1,1,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0

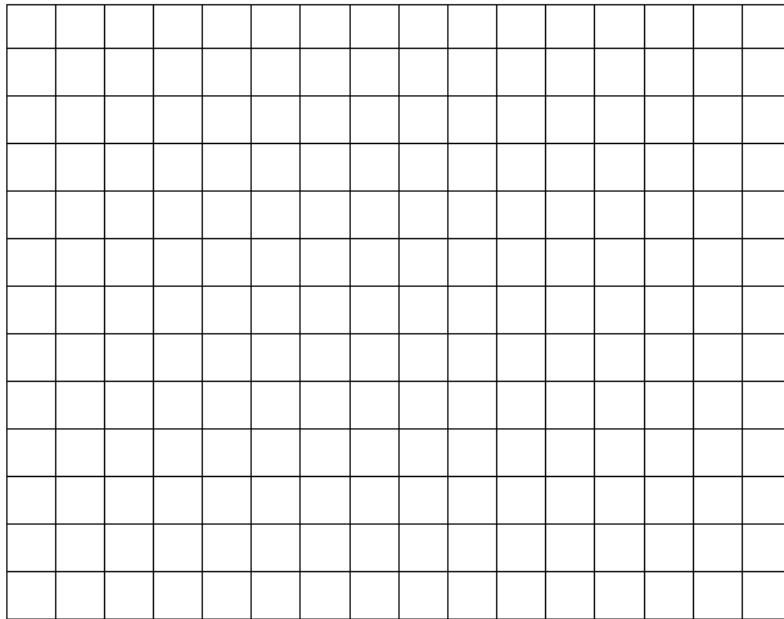
Iškodavę, mokiniai turėtų gauti tokius paveikslus:



Aptarkite su mokiniais šiuos klausimus:

- Kas pasikeistų, jei susitartume koduoti taip: „1“ – nespalvinamas langelis, „0“ – spalvinamas?
- Kiek bitų reikėtų norint užkoduoti nagrinėtus paveikslus?

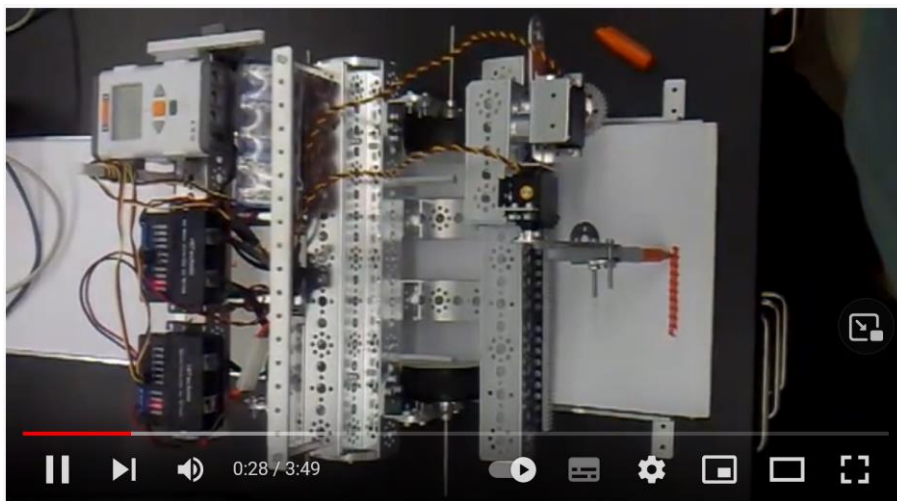
Pasiūlykite mokiniams ant pasirinkto dydžio tinklelio (žr. pridedamą 2-ą darbo lapą) nupiešti patinkantį taškinį piešinį ir užrašyti jo kodą laikantis iš anksto apibrėžtų susitarimų.



Kodas

Kartu su mokiniais peržiūrėkite vaizdo įrašą, kaip robotas piešia paveikslą iš taškų:

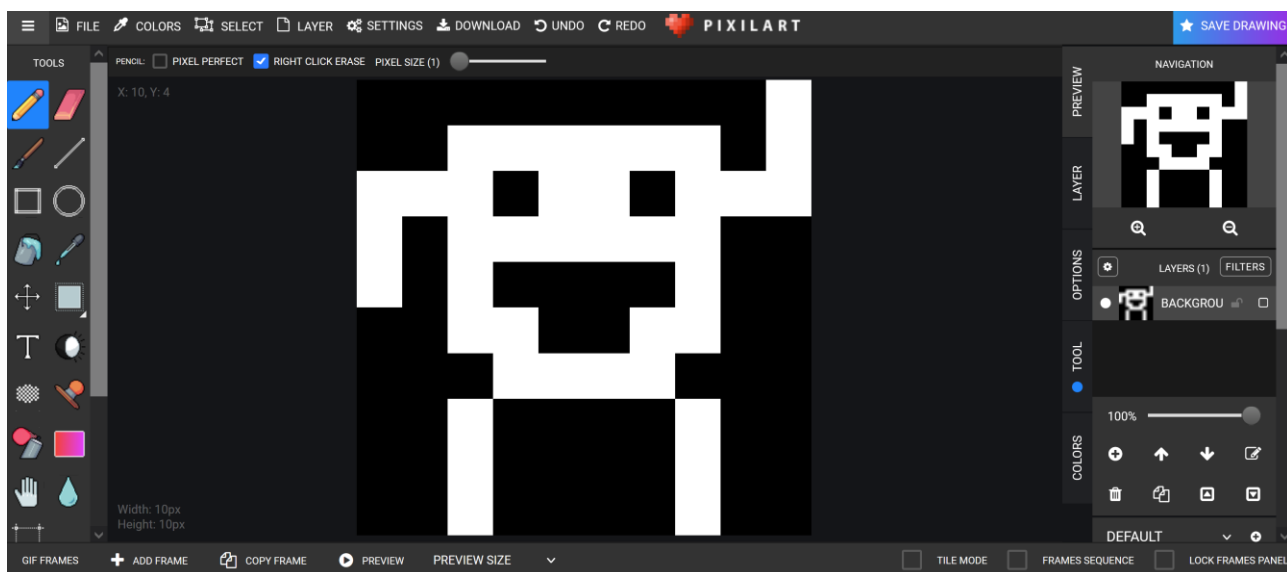
<https://www.youtube.com/watch?v=koBK2glgrQ>



Aptarkite su mokiniais, kaip robotas suprogramuotas piešti.

- Susiekite mokinių ankstesnes žinias apie komandas ir nuoseklų jų vykdymą („pirmyn“, „pasisukti į dešinę“, „pasisukti į kairę“, „sustoti“ ar panašias). Šias komandas mokiniai, tikriausiai, jau taikė robotui (pvz., „Blue-bot“) arba žaidybinės programavimo aplinkos (pvz., „ScratchJr“, „Scratch“) veikėjui programuoti. Išbandykite paprasčiausio taškinio piešinio piešimą (ar piešimo imitaciją) su turimu robotu ar interaktyviu programuojamu veikėju ekrane.

- Vietoje langelių spalvinimo galima naudoti popierinius baltos ir juodos spalvos kvadratėlius (lipnius lapelius ar kitus sutartinius objektus), dedant (klijuojant) juos ant tinkamo langelio.
- Išnagrinėkite su mokiniais 1-ą uždavinį iš šios metodinės medžiagos skyrelio „Vaizdų kodavimas. Bebro uždaviniai“. Šis uždavinys taip pat iliustruoja, kaip linijinį piešinį galima paversti pikseliais.
- Išbandykite taškinio piešimo kompiuterinę priemonę „Pixilart“ (<https://www.pixilart.com>), kuria galima rengti nuo paprastų nespaltvų piešinių pikseliais nurodant tinkamo dydį, iki sudėtingų, realistiškai atrodančių paveikslų, naudojant spalvas, sluoksnius ir efektus.



- Su vyresnių klasių mokiniais paveikslų kodavimo ir iškodavimo veiklas galite papildyti kodo užrašymu šešioliktainiu pavidalu. Darbo lapų pavyzdžiai:

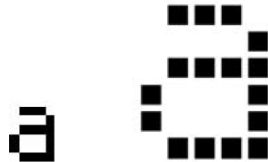
<http://cse+k12.org/binary/bitmaps.html>

	00000000	00
	00011100	1c
	00100010	22
	01000001	41
	01001001	49
	01000001	41
	00100010	22
	00011100	1c

GRAFINIO PRANEŠIMO SIUNTIMAS. VAIZDŲ GLAUDINIMO PRADMENYS

Šia veikla tęsiama pažintis su vaizdų kodavimu, keliame pagrindinį klausimą: „Kaip užkoduoti paveikslą užimant mažiau vietos kompiuterio atmintyje?“.

Prisiminkite su mokiniais vaizdų pateikimo kompiuteriuose principus (ankstesnė veikla). Šį kartą galite naudoti raidės paveikslą kodavimo principui pademonstruoti.



Paveikslui paversti dvejetainiais skaičiais taikoma paprasta schema:

0	1	1	1	0
0	0	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	1

Paklauskite mokinių, ar jų manymu tai gera idėja – taip koduoti vaizdus?

(Atsakymas: toks būdas veikia, bet galime padaryti geriau.)

Jei laikytume kiekvieno paveikslo kiekvieną pikselį kompiuterio atmintyje, koduodami paveikslus tokiu būdu, vietos pritrūktų daug greičiau nei dabar. Mūsų kompiuteriai **suglaudina** paveikslus, kad jų failo dydis būtų mažesnis, tačiau išlaiko visus duomenis, reikalingus vaizdui atkurti. Pristatysime **naują kodavimo schemą**, pagal kurią paveikslui užkoduoti bus naudojama mažiau skaičių.

Įsivaizduokite, kad norėtume išsiųsti grafinį pranešimą kompiuterių tinklais ar net į kitą planetą ar galaktiką. Turėtume sumažinti paveikslo užimamą vietą atmintyje. Kaip tai galėtume padaryti?

Palyginimui paklauskite mokinių, kiek skaičių reikėtų norint pavaizduoti raidę „a“ ir taikant anksčiau pateiktą paprastą metodą.

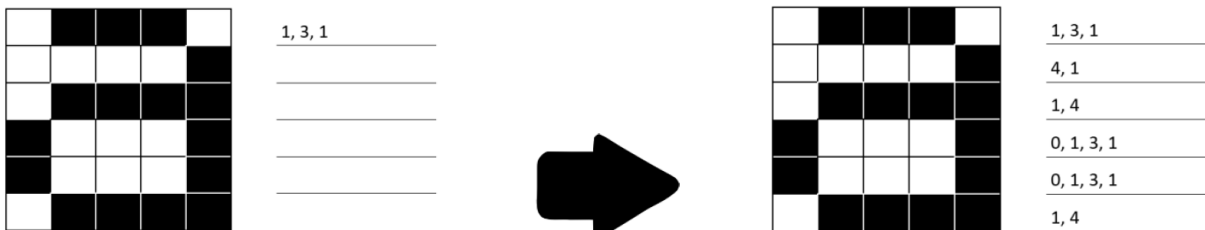
(Atsakymas: reikėtų $6 \times 5 = 30$ skaičių.)

Paklauskite mokinių, ar jie pastebėjo kokių nors paveikslėlyje esančių dėsningumų. Atkreipkite dėmesį, kad pirmoje eilutėje šalia yra 3 juodi pikseliai, o antroje – 4 balti pikseliai. Nukreipkite diskusiją link minties, kad užuot užrašę kiekvieną bitą, galima skaičiais apibūdinti raštą. Atkreipkite mokinių dėmesį, kad pirmoje eilutėje yra juodų pikselių eilutė, o antroje – baltų pikselių eilutė. Mums reikia turėti kokią nors schemą, kuri leistų sužinoti ne tik, kiek yra iš eilės pikselių, bet ir kokios spalvos.

Sutarkite su mokiniais naudoti šias **taisykles**:

1. Jei juodi langeliai paveiksle yra greta, juos sudėsime kartu.
2. Pirmiausia išvardysime, kiek yra baltų pikselių eilutėje. Tada pakaitomis nurodysime, kiek langelių yra juodos ir baltos spalvos.

Dabar pirmoji raidė „a“ piešinio eilutė būtų užkoduojama kaip 131, nes joje yra 1 baltas pikselis, po to – 3 juodi pikseliai, toliau – dar vienas baltas pikselis.



Kaip koduotumėte 4 eilutę (ji prasideda juodu pikseliu)?

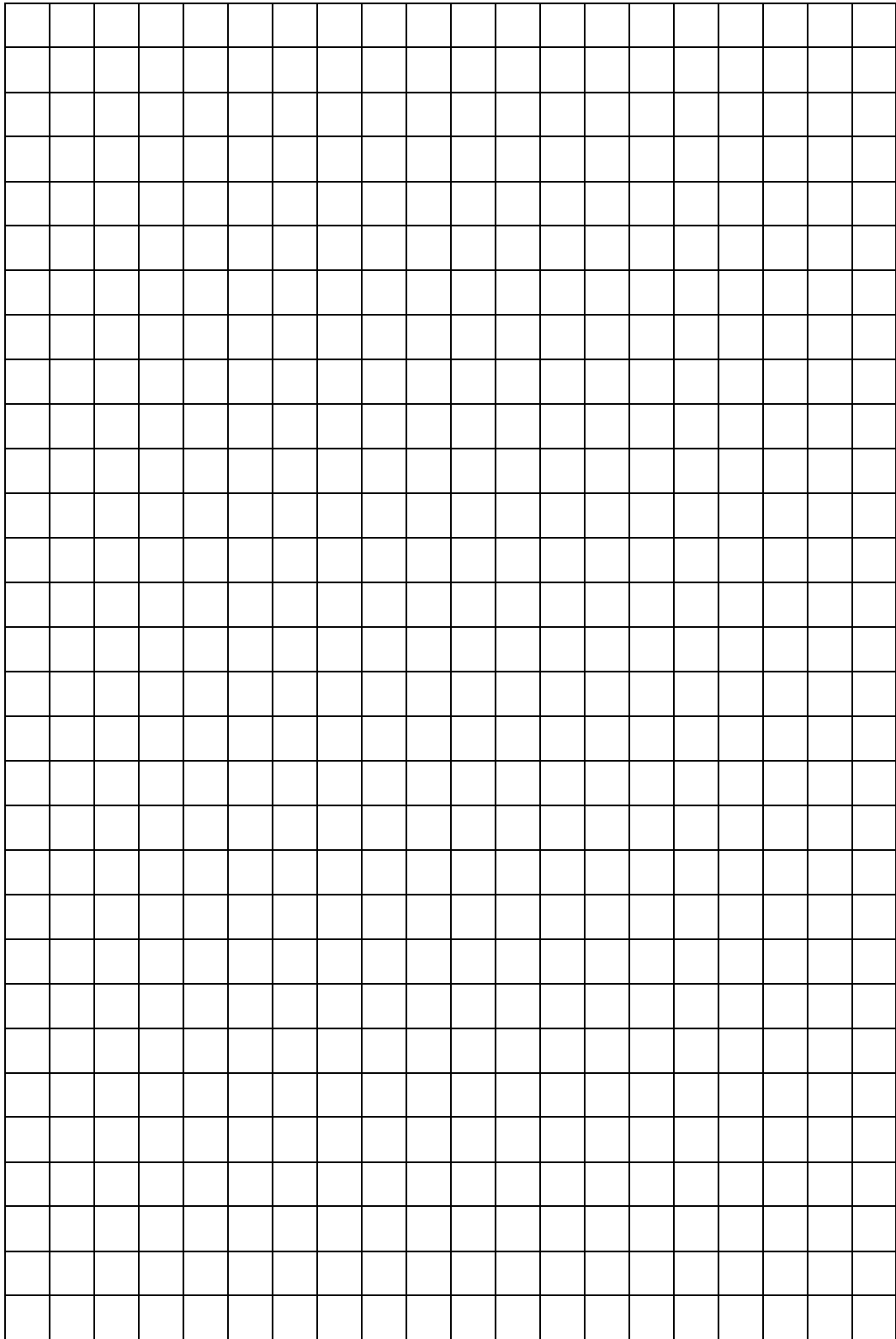
Kadangi pirmiausia reikia nurodyti baltų pikselių skaičių (0), eilutę užkoduotume kaip 0131.

Kodėl? (Atsakymas: kompiuteriui reikia vienodos taisyklės vaizdui atkurti. Jei pirmą nulį praleistume, 4 eilutės spalvos būtų apverstos, nes kompiuteris manytų, kad pradėdame nuo 1 balto pikselio, tada 3 juodų pikselių, tada dar 1 balto pikselio.)

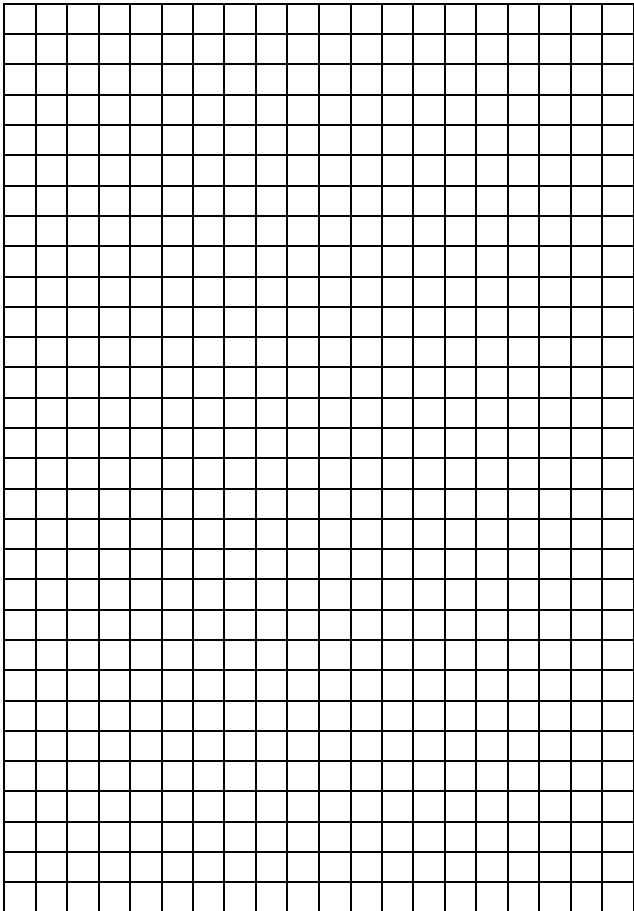
Pabrėžkite mokiniams, kad glaudinimo skaičių suma eilutėje turėtų būti lygi bendram pikselių skaičiui kiekvienoje eilutėje. Šiame pavyzdyje, kadangi kiekvienoje eilutėje yra 5 pikseliai, kiekvieno suglaudinto vaizdo eilutės skaičių suma turėtų būti 5.

Naudodami tinklelius ir paveikslų kodus, mokiniai atlieka suglaudintų paveikslų iškodavimo praktinę veiklą naudodami darbo lapus „Vaizdų kodavimas. Tinkleliai paveikslo iškodavimui“ (failas „Darbo lapai-3.docx“). Galite naudoti didelio arba mažo tinklelio dalį, reikalingą paveikslui nupiešti.

Didelis tinklelis



Mažas tinklelis

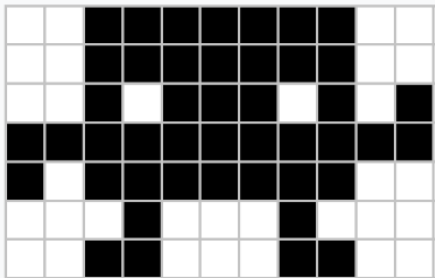


Paveikslų kodai

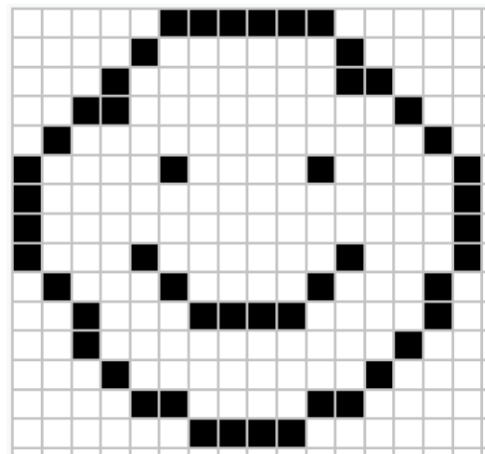
<p>1 paveikslas</p> <p>2, 7 2, 7 2, 1, 1, 3, 1, 1, 1, 1 0, 11 0, 1, 1, 7 3, 1, 3, 1 2, 2, 3, 2</p>	<p>2 paveikslas</p> <p>5, 6 4, 1, 6, 1 3, 1, 7, 2 2, 2, 9, 1 1, 1, 12, 1 0, 1, 4, 1, 4, 1, 4, 1 0, 1, 14, 1 0, 1, 14, 1 0, 1, 3, 1, 6, 1, 3, 1 1, 1, 3, 1, 4, 1, 3, 1 2, 1, 3, 4, 4, 1 2, 1, 10, 1 3, 1, 8, 1 4, 2, 4, 2 6, 4</p>
<p>3 paveikslas</p> <p>4, 1, 5 3, 3, 4 2, 1, 3, 1, 3 1, 2, 1, 1, 1, 2, 2 0, 1, 7, 1, 1 0, 1, 1, 2, 1, 2, 1, 1, 1 0, 1, 1, 2, 1, 2, 1, 1, 1 0, 1, 4, 2, 1, 1, 1 0, 9, 1 4, 6</p>	<p>4 paveikslas</p> <p>1, 11 1, 1, 9, 3 1, 1, 9, 1, 1, 1 1, 2, 7, 2, 1, 1 2, 1, 7, 1, 2, 1 2, 2, 5, 5 3, 2, 3, 2 4, 5 0, 13 2, 9</p>

Atsakymai (iškoduoti paveikslai):

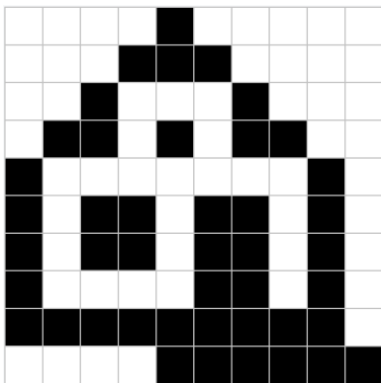
1 paveikslas



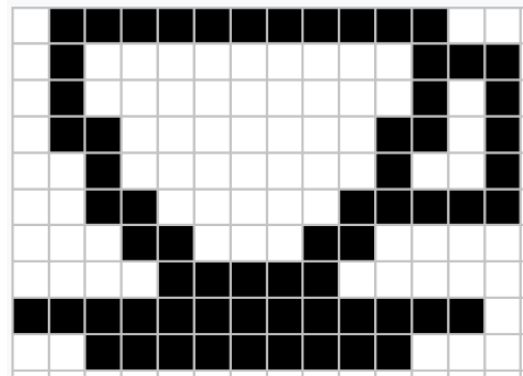
2 paveikslas



3 paveikslas



4 paveikslas



Vaizdų glaudinimo nauda

Kodėl reikia vaizdus glaudinti ir kodėl mums rūpi taupyti vietą? Suglaudinti vaizdai užima žymiai mažiau vietos standžiajame diske, tačiau šiais laikais kompiuterių standieji diskai yra gana dideli.

O kaip dėl vaizdo įrašų, filmų? Filmai ir vaizdo įrašai sudaryti iš daugybės statinių vaizdų, rodomų labai greitai vienas po kito. Tarkime, mūsų filmą sudaro 120 statinių vaizdų, kurie sudaro judančio vaizdo įspūdį.

Jei vienam nesuglaudintam vaizdui atsisiųsti reikia 10 sekundžių, kiek sekundžių užtruktų atsisiųsti visą nesuglaudintą filmą?

(Atsakymas: $120 \times 10 = 1200$ sekundžių. O tai yra $1200/60$ arba 20 minučių.)

Akivaizdu, kad tai nėra tikslus skaičius (vienam vaizdui nebūtinai reikia 10 sekundžių atsisiųsti, o daugumoje filmų yra daug daugiau nei 120 vaizdų), tačiau visi žinome, kiek kartų mums yra tekę laukti, kol bus įkrautas vaizdo įrašas.

Iki šiol mokiniai vaizdavo paveikslą dviem skirtingais būdais: pirmuoju būdu kiekvienas pikselis buvo vaizduojamas kaip 0 arba 1, o antruoju – pikseliai buvo grupuojami, nurodoma pikselių grupės skaičius pradedant nuo baltų pikselių grupės. Įtvirtinkime, kodėl tai geriau, pademonstruodami nesuglaudinto vaizdo (pirmasis metodas) ir suglaudinto vaizdo (antrasis metodas) siuntimą.

Pasiūlykite mokiniams **praktinę veiklą „Paveikslų siuntimas“**.

Mokiniai susiskirsto į grupes po du. Vienas mokinys veiks kaip siunčiantysis kompiuteris (siuntėjas), kuris jau turi siunčiamo paveikslo kodą. Kitas mokinys veiks kaip gaunantysis kompiuteris (gavėjas), gaunantis paveikslo kopiją.

Mokiniai atliks du vaizdų kodų vertimo paveikslais raundus. Svarbu, kad per pirmąjį raundą būtų iškoduojuama nesuglaudinto paveikslo versija, o per antrąjį – suglaudinta versija.

Galima naudoti dviem būdais užkoduotus paveikslus iš ankstesnių veiklų (1 ir 3 darbo lapai, arba 2 darbo lapas laisvam piešimui ir kodo užrašymui). Galite iš anksto parengti mokiniams paveikslų ir jų kodų naudodamiesi piešinių kodavimo priemone (<https://csfieldguide.org.nz/en/interactives/run-length-encoding/>).

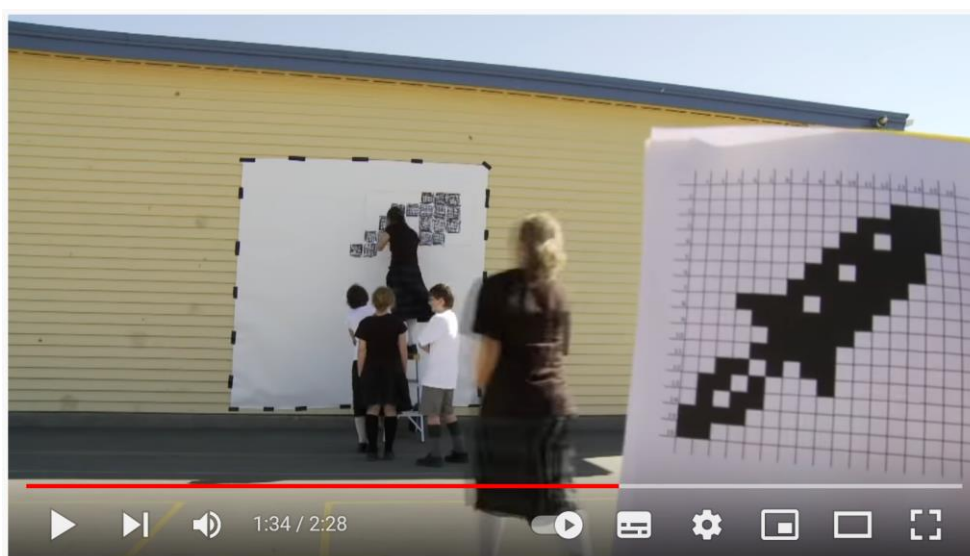
Mokinys, veikiantis kaip siunčiantysis kompiuteris, žodžiu vieną po kitos perduos priimančiajam kompiuteriui reikšmes. Priimančiasis kompiuteris turi pranešti siunčiančiajam kompiuteriui, kad yra pasiruošęs gauti duomenis po to, kai darbo lape nuspelvins reikiamą skaičių langelių. Svarbu pabrėžti, kad abiejuose raunduose siuntėjas gavėjui vienu metu gali

pateikti tik **vieną** skaičių. Pavyzdžiui, nesuglaudinto paveiklo atveju, priimantysis kompiuteris gali gauti pirmąjį skaičių „1“ ir, prieš prašydamas kito skaičiaus, turi baigti spalvinti vieną langelį, o suglaudinto paveiklo atveju, priimantysis kompiuterius gali gauti „3“, ir iš karto gali spalvinti tris langelius iš eilės.

Ijunkite laikmatį, kurį matytų mokiniai. Mokiniai užfiksuoja laiką iš karto, kai baigia kurti paveikslą. Užrašykite lentoje penkis geriausius laikus. Taip darykite abiejų žaidimo rundų metu. Mokiniai turėtų pamatyti, kad suglaudintų paveikslų „siuntimo“ laikas yra daug spartesnis nei nesuglaudintų.

Toliau pateiktas **vaizdo įrašas** demonstruoja vieną iš veiklos organizavimo variantų:

<https://www.youtube.com/watch?v=uaV2RuAJTjQ&t=103s>



Naudodamiesi interaktyvia **piešinių kodavimo priemone**

(<https://csfieldguide.org.nz/en/interactives/run-length-encoding/>) parenkite daugiau piešinių ir jų kodų rinkinių, iš kurių mokiniai pagal pateiktą kodą galėtų nupiešti piešinį arba pagal pateiktą piešinį parašyti jo kodą.

Grid Size: 7

Number of bits for grid: 49

Number of characters for encoding: 47

		■		■		
			■			
			■			
		■		■		

2, 1, 1, 1, 2
 3, 1, 3
 3, 1, 3
 2, 1, 1, 1, 2
 1, 1, 3, 1, 1
 2, 1, 1, 1, 2
 7

Apibendrinimas

Kompiuteriuose beveik viskas vaizduojama naudojant grafinę sąsają. Bet kuris skaitmeninis įrenginys, turintis ekraną, turi gebėti interpretuoti duomenis ir paversti juos vaizdu.

Informatikams rūpi duomenų glaudinimas, nes jis leidžia taupyti vietą atmintyje, atlikti daugiau darbų neatnaujinant techninės įrangos.

Pritaikę aptartus metodus, galime siųsti daugiau informacijos naudodami tą patį žingsnių skaičių (prisiminkite glaudinimo pratimą: gavęs suglaudintą vaizdą mokinys galėjo nuspelvinti kelis pikselius iki kitos duomenų porcijos siuntimo). Kuo geresnė glaudinimo schema, tuo patogiau galime žiūrėti geros kokybės vaizdo įrašus nekeisdami savo interneto ryšio spartos.

Toks glaudinimo algoritmas vadinamas duomenų sekų kodavimu (angl. *run-length encoding*), jis yra duomenų nenuostolingojo glaudinimo forma. Toks glaudinimas veiksmingiausias duomenims, kuriuose yra daug pasikartojančių duomenų grupių, pavyzdžiui, nesudėtingiems grafiniams vaizdams, piktogramos, linijiniams brėžiniams, animacijoms ir pan.

Šis glaudinimo algoritmas realiai taikomas skeneriuose ir fakso aparatuose.

Su vyresnių klasių mokiniais galite papildomai išnagrinėti 3, 4 ir 5 uždavinius iš šios metodinės medžiagos skyrelio „Vaizdų kodavimas. „Bebro“ uždaviniai“.

SPALVOTŲ PAVEIKSLŲ KODAVIMAS

Nuo to, kaip paveikslai pateikiami skaitmeniniu būdu, priklauso, kaip jie atvaizduojami, kuriami, saugomi ir kaip jais manipuluojama. Kuo daugiau bitų (dvejetainių skaitmenų) naudojame pikselio spalvai saugoti, tuo didesnę spalvų įvairovę galime gauti. Jei pikselio spalvai saugoti naudojame vieną bitą, tuomet turime tik dviejų spalvų pikseliui galimybes (atminkite, kad vienu bitu galime išreikšti tik dvi skirtingas reikšmes, nes galime naudoti tik 0 arba 1). Jei pikseliui naudojame 8 bitus (1 baitą), galime atvaizduoti 256 skirtingas spalvas, o jei naudojame 24 bitus (3 baitus), galime atvaizduoti daugiau kaip 16 mln. skirtingų spalvų, t. y. daugiau, nei žmogaus akis gali įžvelgti.

Tačiau naudojant daugiau bitų spalvai padidėja failo dydis. Failas užims daugiau vietos atmintyje, kompiuteris ilgiau jį apdoros ir atvaizduos, o jo perdavimas tarp kompiuterių ir internetu užtruks ilgiau. Filmų ar kompiuterinių žaidimų kūrėjai naudoja galingus kompiuterius savo darbams. Senesni kompiuteriai paprasčiausiai nesugeba pakankamai sparčiai apdoroti žaidimų grafikos.

„Pixelation“ priemonės praktinis taikymas

„Code.org“ mokymosi svetainėje pateikiama interaktyvi priemonė, padedanti mokytis ir leidžianti išbandyti spalvotų paveikslų kodavimo pradmenis. „Pixelation“ priemonė pasiekama adresu: <https://studio.code.org/s/pixelation/lessons/5/levels/1>.

The screenshot shows the 'Pixelation' tool interface. It features a 6x6 grid of pixels. The top row consists of three black pixels followed by three white pixels. The remaining five rows are entirely pink. To the right of the grid, there are three sliders: 'Image width' set to 6, 'Image height' set to 6, and 'Bits per pixel' set to 1. Below these sliders, there are radio buttons for 'Binary' (selected) and 'Hexadecimal'. A 'Pixel format' dropdown menu is set to '1'. A text area on the right contains the following binary code:
0000 0110
0000 0110
0000 0001
101010|

Lauke, kuriame rašomi piešinio kodai, pirmieji 8 bitai apibrėžia tinklelio plotį, kiti 8 bitai – aukštį. Virš šio lauko matomi atitinkami dešimtainiai skaičiai, plotį ir aukštį galime nurodyti interaktyviai. Vaizdavimo patogumo dėlei galime paskirstyti šiuos skaitmenis į dvi eilutes, pridėti tarpą tarp 4 skaitmenų grupių.

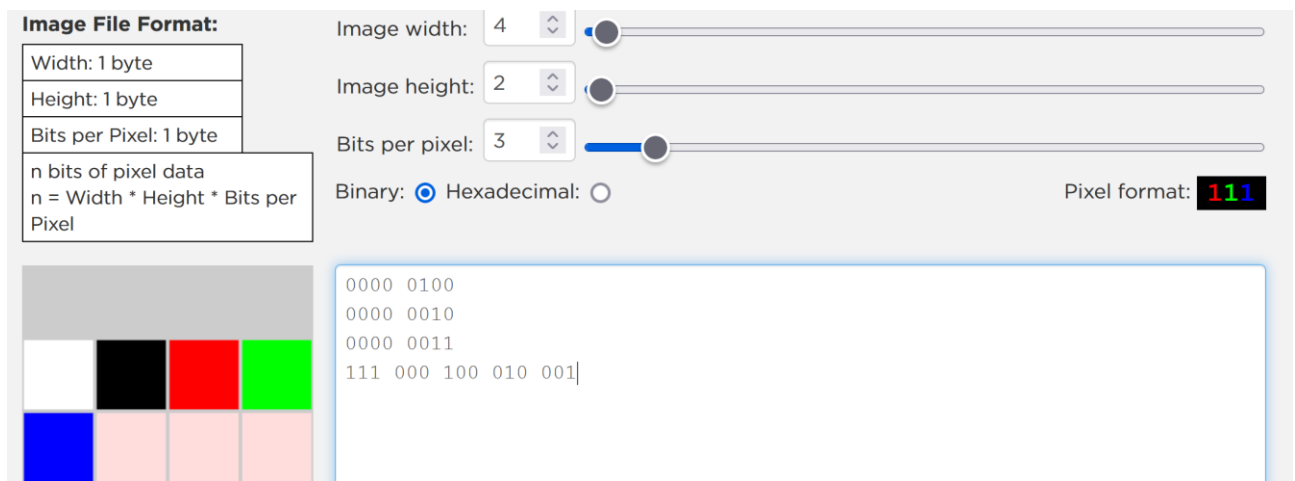
Toliau einantys skaitmenys – dar viena *metaduomenų* dalis, ji nurodo bitų, skirtų pikseliui, kiekį. Tai naudojama pikselio spalvai nurodyti. Jei tai piešinys, kuris susideda tik iš juodos ir baltos spalvų, mums reikia tik vieno bito pikseliui: 1 reiškia juodą pikselį, 0 reiškia baltą pikselį.

Jei norime naudoti daugiau spalvų, turime padidinti pikseliui skiriamų bitų kiekį.

Panagrinėkime paprasčiausią pavyzdį, kai tinklelis yra 4x2, o pikselio spalvai nurodyti skiriami 3 bitai (raudonai, žaliai ir mėlynai spalvai).

Jei visus bitus „įjungsimė“ (111), gausime baltos spalvos pikselį. Jei visus bitus „išjungsimė“ (000), gausime juodos spalvos pikselį.

Jei norėtume nupiešti raudonos spalvos tašką, turėtume „įjungti“ raudonos spalvos bitą, o kitus – „išjungti“, t. y. būtų bitų seka: 100. Atitinkamai žalio pikselio kodas būtų 010, o mėlyno – 001.



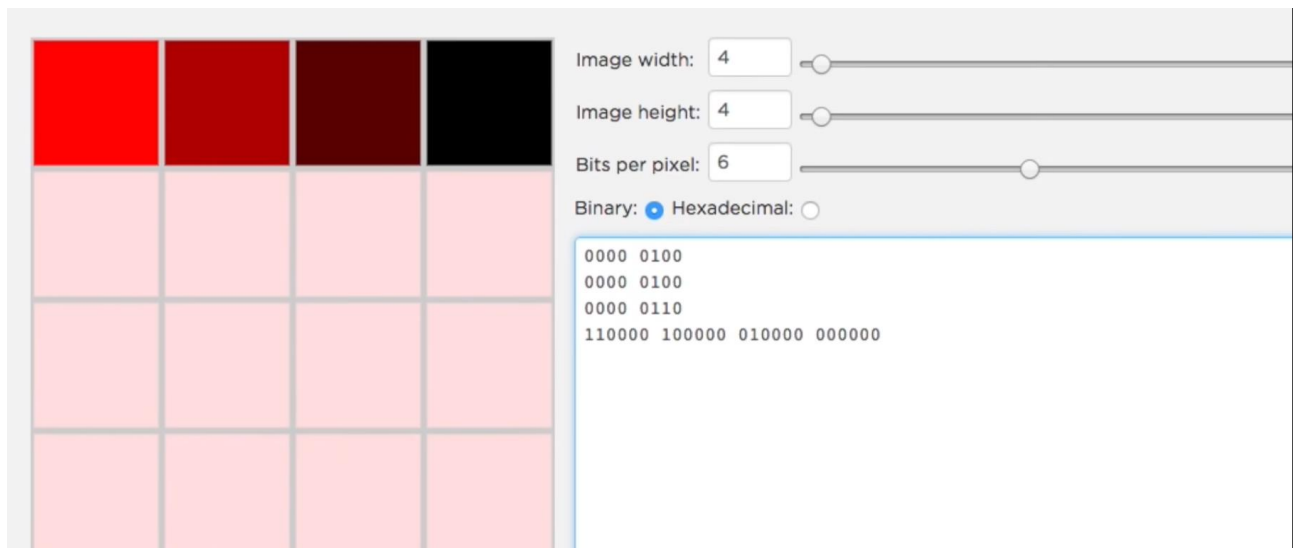
The screenshot shows a configuration window for an image file format. On the left, a box titled "Image File Format:" contains the following information: "Width: 1 byte", "Height: 1 byte", "Bits per Pixel: 1 byte", and a formula "n bits of pixel data n = Width * Height * Bits per Pixel". Below this is a small grid of colored pixels (white, black, red, green, blue, pink). The main configuration area includes: "Image width: 4", "Image height: 2", "Bits per pixel: 3", and "Binary: Binary Hexadecimal". The "Pixel format" is displayed as "111". A text area at the bottom contains the following binary code: "0000 0100", "0000 0010", "0000 0011", "111 000 100 010 001".

Klausimas mokiniam: Kas atsitiktų, jei pikselio trijų bitų kode būtų du vienetai ir vienas nulis? Kokias spalvas gautume?

Leiskite mokiniam paeksperimentuoti su taškinio piešimo priemone ir patikrinti hipotezes. (Rezultatai tikriausiai skirsis nuo tikėtinų, kadangi tai nėra šviesos maišymas).

Tam, kad gautume daugiau atspalvių, turėtume padidinti spalvoms nurodyti skiriamą bitų skaičių. Pavyzdžiui, toliau pateiktame paveiksle naudojami 6 bitai pikseliui: 2 bitai raudonai spalvai, 2 bitai žaliai spalvai ir 2 bitai mėlynai spalvai.

Sudarydami kiekvienos bitų grupės bitų įvairias kombinacijas galime gauti įvairių atspalvių. Čia parodytas raudonos spalvos atspalvių kodavimas:

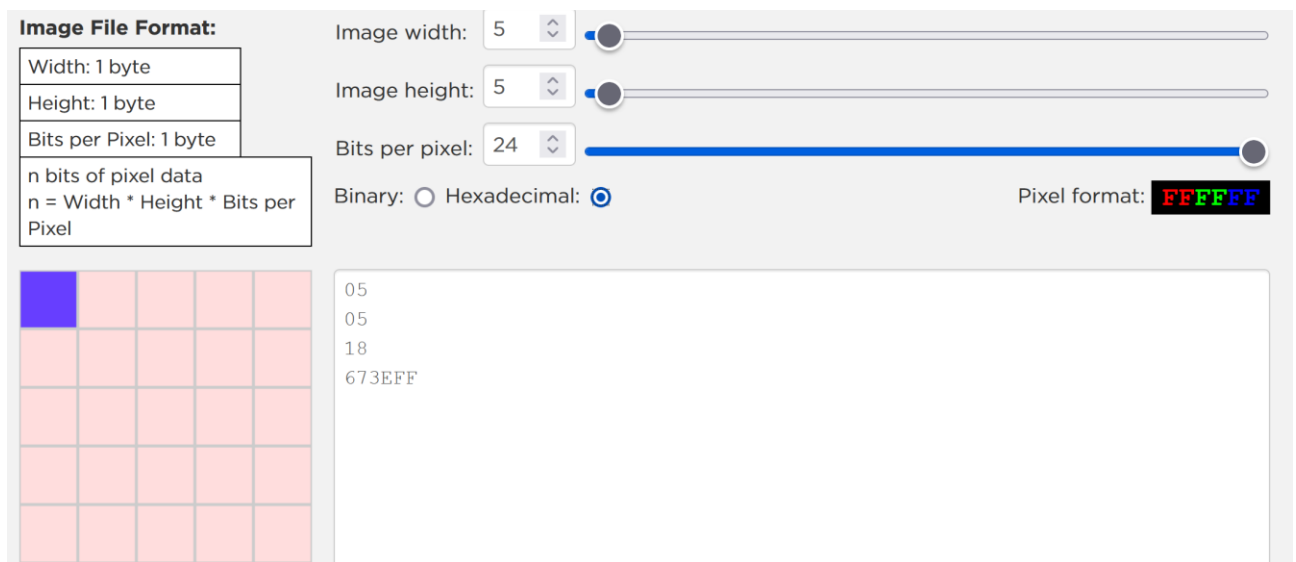


Sukurkite žalios ir mėlynos spalvų atspalvių paletę. Naudokite interaktyvią priemonę „Pixelation widget“ (Code.org).

Išbandykite pikselio kodavimą 9, 12, ..., 24 bitais.

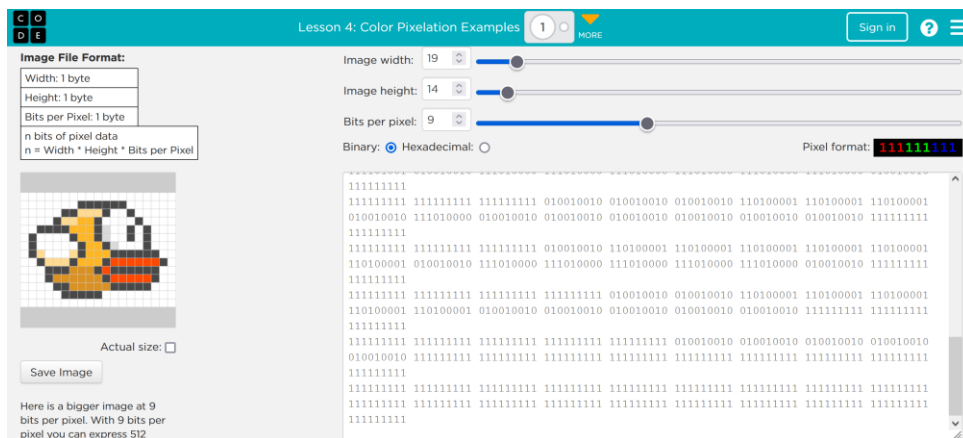
Pakartokite su mokiniais šešioliktainius kodus.

Kai taškui skiriami 24 bitai, jo trims pagrindinėms modelio spalvoms (raudonai, žaliai, mėlynai) skiriama po vieną baitą (8 bitus). Tokius kodus patogiau vaizduoti pavertus juos į šešioliktainę sistemą (pažymėkite priemonėje „hexadecimal“).



Veiklų tęsinys

- Išnagrinėkite su mokiniais 2-ą uždavinį iš šios metodinės medžiagos skyrelio „Vaizdų kodavimas. „Bebro“ uždaviniai“. Šis uždavinys susijęs ne tik su spalvų kodavimu, bet ir šablonų (dėšningumų) atpažinimu. Pasiūlykite mokiniams užkoduoti tokį pat raštą „Pixelation“ priemone (<https://studio.code.org/s/pixelation/lessons/5/levels/1>) ir parsisiųsti į savo kompiuterį.
- Išnagrinėkite taškinių spalvotų paveikslų, sukurtų „Pixelation“ priemone, pavyzdžius: <https://studio.code.org/s/pixelation/lessons/4/levels/1>. Pvz., pavaizduotas piešinio kodavimas, kai vienam pikseliui skiriami 9 bitai:



- Pasiūlykite mokiniams susikurti savo spalvotą piešinį nurodant kiekvieno spalvoto pikselio kodą, operuojant dvejetainiais arba šešioliktainiais kodais.

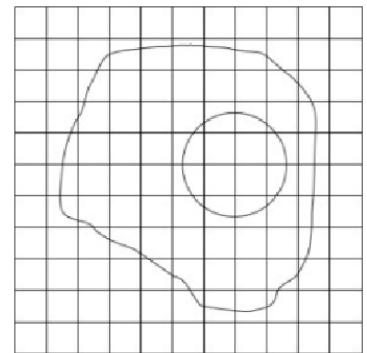
VAIZDŲ KODAVIMAS. „BEBRO“ UŽDAVINIAI

Šiame metodinės medžiagos skyrelyje pateikiame keletą tarptautinio informatikos ir informatinio mąstymo konkurso „Bebras“ uždavinių, kuriuos galėtumėte naudoti per pamokas vaizdų kodavimo ir glaudinimo principams bei jų taikymams mokytis. 1–2 uždaviniai tinka pradinės mokyklos arba pagrindinės mokyklos žemesnių klasių mokiniams, 3–5 uždaviniai – pagrindinės mokyklos aukštesnių klasių mokiniams.

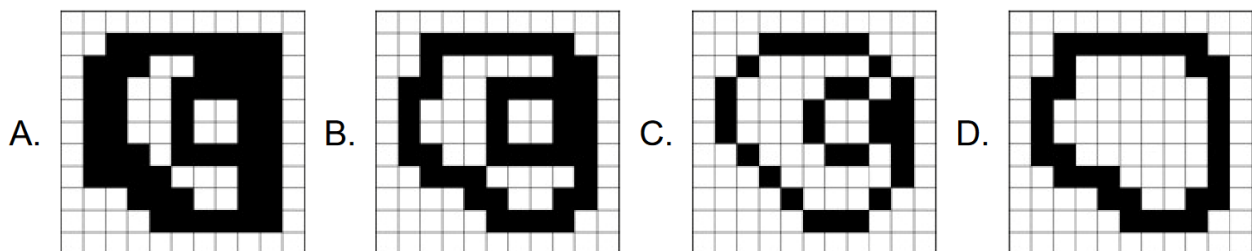
Daugiau uždavinių galite rasti „Bebro“ varžybų lauke, „Bebro“ kortelėse ir knygelėse.

1. Kepamas kiaušinis

Bebriukai juodu pieštuku piešia paveikslėlius. Jiems ypač patinka kepamo kiaušinio vaizdas. Paveikslėlį jie įrašo į kompiuterį – jo dydis 11x11 pikselių. Keista, bet atvėrus įrašytą paveikslėlį gražiai išlenktų linijų nebesimato! Užtat visi langeliai (pikseliai), per kuriuos ėjo linija, visiškai juodi.



Kurį vaizdą mato bebriukai?



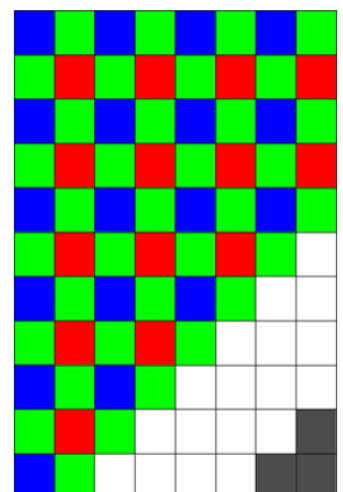
(Atsakymas: B)

2. Spalvų tinklelis

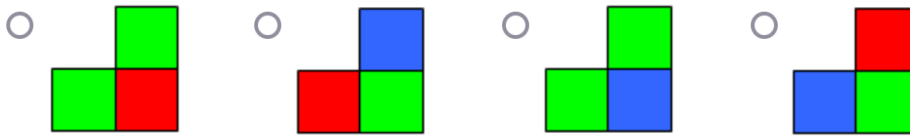
Kiekvienas tinklelio langelis nuspalvintas raudonai, žaliai arba mėlynai.

Tai yra RŽM (Raudona-Žalia-Mėlyna) spalvų modelis, taikomas elektronikoje ir kompiuteriuose.

Dalis nuspalvinto 8x11 tinklelio parodyta kairėje. Pirmoje eilėje pakaitomis eina mėlyni ir žali langeliai, antroje eilėje – žali ir raudoni, trečioje – vėl mėlyni ir žali, ir taip toliau.



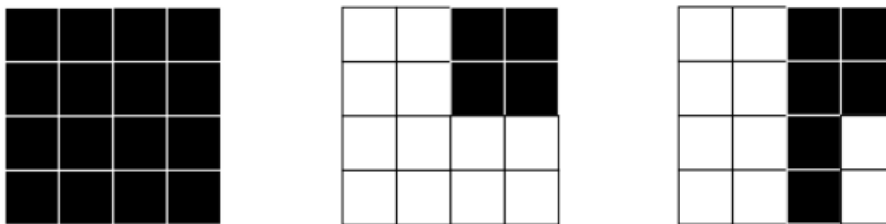
Kokių spalvų langeliai bus apatiniame dešiniajame tinklelio kampe, paveiksle pažymėti pilka spalva, jei šiuo raštu baigtume spalvinti visą tinklelį?



(Atsakymas: dešinėje esantis raštas)

3. Vaizdų glaudinimas (1)

Pažvelkite į 4×4 juodų ir baltų langelių (pikselių) paveikslus:



Paveikslus galime aprašyti dvejetainiais skaitmenimis: „1“ – baltas langelis, „0“ – juodas langelis.

Taigi 4×4 dydžio vaizdai užrašyti reikia 16 skaitmenų. Šiuo glaudinimo metodu galime tai užrašyti taupiau, ypač nesudėtingais atvejais, pavyzdžiui:

0	0	0	0	0	1	1	0	0	1	1	0	0
0	0	0	0	0	1	1	0	0	1	1	0	0
0	0	0	0	0	1	1	1	1	1	1	0	1
0	0	0	0	0	1	1	1	1	1	1	0	1
				0								(10(0110)1)
												(1011)

Dvejetainiai skaitmenys išdėstomi lentele kaip pikseliai vaizduose. Glaudinimo metodas taikomas skaitmenų lentelei kol gaunamas rezultatas – eilutė.

Jei visi skaitmenys lentelėje lygūs 0, rezultatas yra „0“ (kairysis paveikslas). Jei visi skaitmenys lygūs 1, rezultatas yra „1“. Kitais atvejais lentelė dalijama į keturias lygias dalis (kvadratus). Glaudinimo metodas taikomas kiekvienai iš šių dalių atskirai, pradedama nuo kairiausios viršutinės ir einama pagal laikrodžio rodyklę. Rezultatai rašomi iš eilės suskliaudžiant lenktiniais skliaustais (ir). Panagrinėkime vaizdus viduryje ir dešinėje.

Atkreipkime dėmesį, kad lentelė ar analizuojama jos dalis gali būti sudaroma ir iš vieno skaitmens. Šiuo atveju taikant metodą prireiks tik 1-ojo žingsnio.

Pateiktas 8×8 vaizdas – skaitmenų lentelė. Čia pritaikytas glaudinimo metodas.

```
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

Kuri eilutė nusako rezultatą?

A) (1110)

B) (11(1011)1)

C) (111(1(1101)11))

D) (111(1(1011)11))

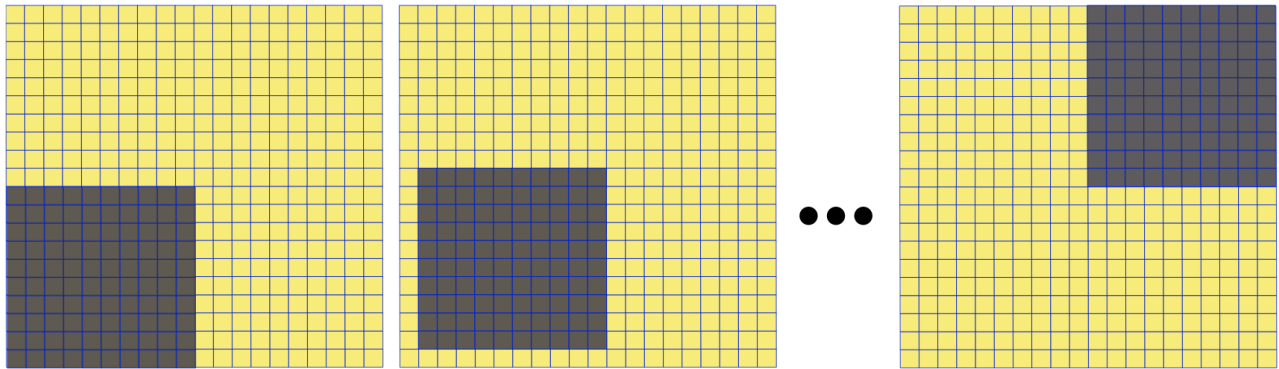
(Atsakymas: D)

4. Vaizdų glaudinimas (2)

Skaitmeninis taškinis paveikslas – tai spalvotų kvadratėlių, vadinamų taškais arba pikseliais, stačiakampis tinklelis. Vaizdo įrašas – tai paveikslų, vadinamų kadrais, seka, kurioje kiekvienas kadras šiek tiek skiriasi nuo ankstesnio.

Paprasčiausias būdas saugoti vaizdo įrašą – tai įrašyti kiekvieno kadro visus pikselius. Efektyvesnis saugojimo būdas – įrašyti visus pirmo kadro pikselius ir tik tuos pikselius, kuriais tolesnis kadras skiriasi nuo ankstesnio.

Pavyzdžio paveiksle pavaizduotas 10×10 dydžio tamsios stačiakampės srities judesys iš kairiojo apatinio kampo į dešinį viršutinį 20×20 dydžio šviesios srities kampą. Kiekviename kadre stačiakampis juda įstrižai (po vieną pikselį horizontaliai ir vertikalčiai). Toks judesys užima 11 kadrų. Jei laikytume šį vaizdo įrašą įprastu formatu, mums reikėtų $(20 \times 20) \times 11 = 4400$ pikselių.



Kiek pikselių reikėtų vaizdo įrašui, jei jį laikytume efektyvesniu formatu?

(Atsakymas: 780)

5. Stebėjimas

Skaitmeninė kamera fotografuoja rotušės vaizdą kas 10 sekundžių (žr. paveikslą). Kompiuterio programa lygina kiekvieną nuotrauką su ankstesne ir formuoja *pokyčių paveikslą*. Mažas raudonas stačiakampis pokyčių paveiksle žymi kiekvieną nuotraukos vietą, besiskiriančią nuo prieš tai darytos nuotraukos.



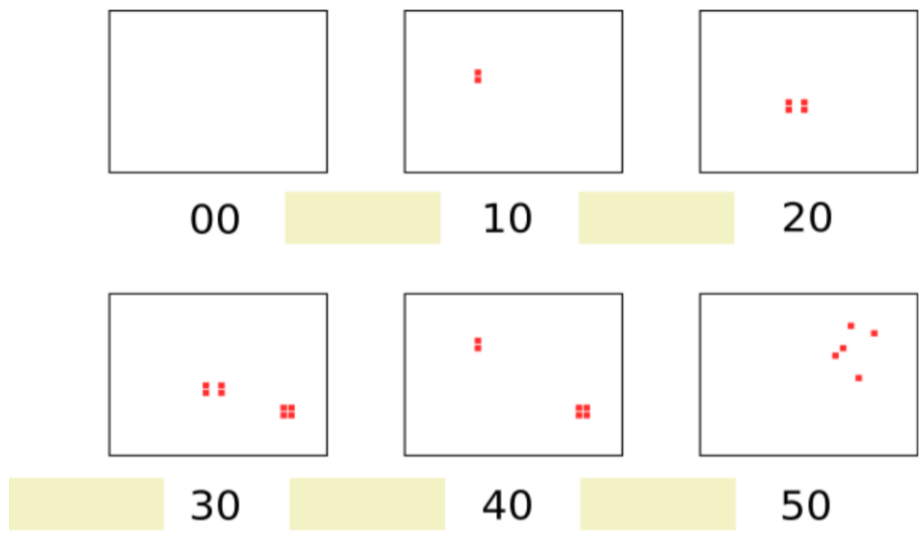
Jei pokyčių paveikslas baltas, tai abi nuotraukos yra vienodos.

Pavyzdžiui, dviem toliau pateiktoms nuotraukoms pokyčių paveikslas (dešinėje) rodo skirtumus tarp pirmos ir antros.

Pateikiama 50 sekundžių intervalo pokyčių paveikslų seka. Per šį laiką įvyko 5 įvykiai.

Susiekite kiekvieną įvykį su jam tinkama vieta.

- A: Tomas sutinka Laurą.
- B: Kažkas atidaro rotušės duris.
- C: Tomas ir Laura eina kartu susikibę rankomis.
- D: Vėjas stiprėja.
- E: Kažkas uždaro rotušės duris.



(Yra du teisingi atsakymai: **BACED** ir **EACBD**)

Šaltiniai

<p>Bell T. Informatika be kompiuterio. Turininga informatikos mokymosi medžiaga pradinukams ir vyresniems mokiniams. 2015. Vertimas į lietuvių kalbą https://informatika.ugdome.lt/wp-content/uploads/2017/05/KNYGA-Informatika-be-kompiuterio-2015-09-03.pdf</p> <p>Image Representation, unit plan, https://www.csunplugged.org/en/topics/image-representation/unit-plan/description/</p> <p>Daugiau veiklų ir papildomų šaltinių „Informatika be kompiuterio“ svetainėje anglų kalba: https://www.csunplugged.org/</p>
<p>Šioje knygoje ir svetainėje sudėtos Naujosios Zelandijos Kenterberio universiteto profesoriaus Timo Bello ir jo kolegų sugalvotos linksmos informatikos veiklos be kompiuterio. Joje, kaip ir svetainėje, aprašomos įdomios ir smagios įvairaus amžiaus mokiniams skirtos užduotys, supažindinama su kompiuterio veikimo pagrindais.</p>
<p>Informatikos ir informatinio mąstymo konkurso „Bebras“ medžiaga</p> <ol style="list-style-type: none">1. Informatika: Bebras 2015, https://bebras.lt/wp-content/uploads/2015/09/Uzdaviniu-2015-m.-knygele.pdf2. Informatika: Bebras 2014, https://informatika.ugdome.lt/wp-content/uploads/2018/08/Bebro-knygele-2014.pdf3. Bebro treniruočių laukas, https://lt.bebas.lt/#!/training
<p>Informatikos ir informatinio mąstymo konkurso „Bebras“ informacija, ankstesnių metų uždaviniai, kortelės. Uždavinių knygelėse pateikiamos uždavinių formulės, jų sprendimai ir paaiškinimai. Kiekvieną uždavinį apibūdina keli informatikos raktažodžiai ir informatinio mąstymo įgūdžiai. „Bebro“ treniruočių lauke galima spręsti ankstesnių metų uždavinius, įskaitant interaktyvius.</p>
<p>Code.org, https://code.org/</p> <p>Taškinių paveikslų kodavimo priemonė https://studio.code.org/s/pixelation/lessons/5/levels/1</p> <p>Representing images, https://studio.code.org/s/csd5-2021/lessons/4</p>
<p>Edukacinė aplinka, kurioje galima mokytis algoritavimo ir programavimo pagrindų, yra pamokų vaizdų kodavimui. Užduotys suskirstytos pagal besimokančiųjų amžių, nuo 4 iki 18 metų. Naudojamos atskiros paskyros mokytojui ir mokiniui. Mokytojas gali sukurti savo klasę ir stebėti kiekvieno mokinio pažangą. Užduotis galima atlikti ir neprisijungus. Svetainėje taip pat yra metodinių patarimų mokytojui anglų kalba.</p>
<p>PIXILART, https://www.pixilart.com</p>
<p>Taškinio piešimo priemonė.</p>
<p>Dagienė V., Grigas G., Jevsikova T. Enciklopedinis kompiuterijos žodynas, http://www.ims.mii.lt/EKŽ/</p>
<p>Žodyne aprašyta apie 4600 terminų ir kitų leksinių vienetų. Pateikiama terminų apibrėžtys, paaiškinimai, iliustracijos.</p>
<p>Informatikos sąvokų žodynelis pradinių klasių pedagogams https://informatika.ugdome.lt/wp-content/uploads/2019/08/Informatikos-sqvokų-žodynelis-pradinių-klasių-mokytojams.pdf</p>

Žodynėlyje pateikiama daugiau kaip 200 svarbiausių informatikos ir informacinių technologijų sąvokų su apibrėžtimis, paaiškinimais, dauguma su pavyzdžiais ir iliustracijomis. Šis žodynėlis padės pradinių klasių pedagogams pasirengti informatikos dalyko integravimui į ugdymo veiklas.

Šiame apraše naudoti paveikslai ir vaizdo įrašai iš šių šaltinių:

- CS unplugged, <https://www.csunplugged.org>. Computer Science Education Research Group, University of Canterbury, New Zealand. Visi „CS unplugged“ svetainės ištekliai platinami pagal „Creative Commons Attribution-ShareAlike 4.0“ licenciją.
- Vaizdo įrašas <https://www.youtube.com/watch?v=uaV2RuAJTjQ&t=4s>. Image Compression - Making Contact. UC Computer Science Education, „Creative Commons“ licencija (leidžiama naudoti pakartotinai).
- Vaizdo įrašas „CS Unplugged Image Representation Activity Part1“ https://www.youtube.com/watch?v=_koBK2glgrQ yra dalis „CS unplugged“ medžiagos, platinamos pagal „Creative Commons Attribution-ShareAlike 4.0“ licenciją.
- Code.org (Representing images, <https://studio.code.org/s/csd5-2021/lessons/4>). All curriculum and tutorial materials developed by Code.org are licensed to you for use under a „Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Unported“ licenciją (<https://code.org/tos>).
- „Bebro“ uždaviniai ir jų paveikslai.
- Tatjanos Jevsikovas nuotraukos.

1 DARBO LAPAS. VAIZDŲ KODAVIMO PRADMENYS. PEIŠIMAS PAGAL KODĄ

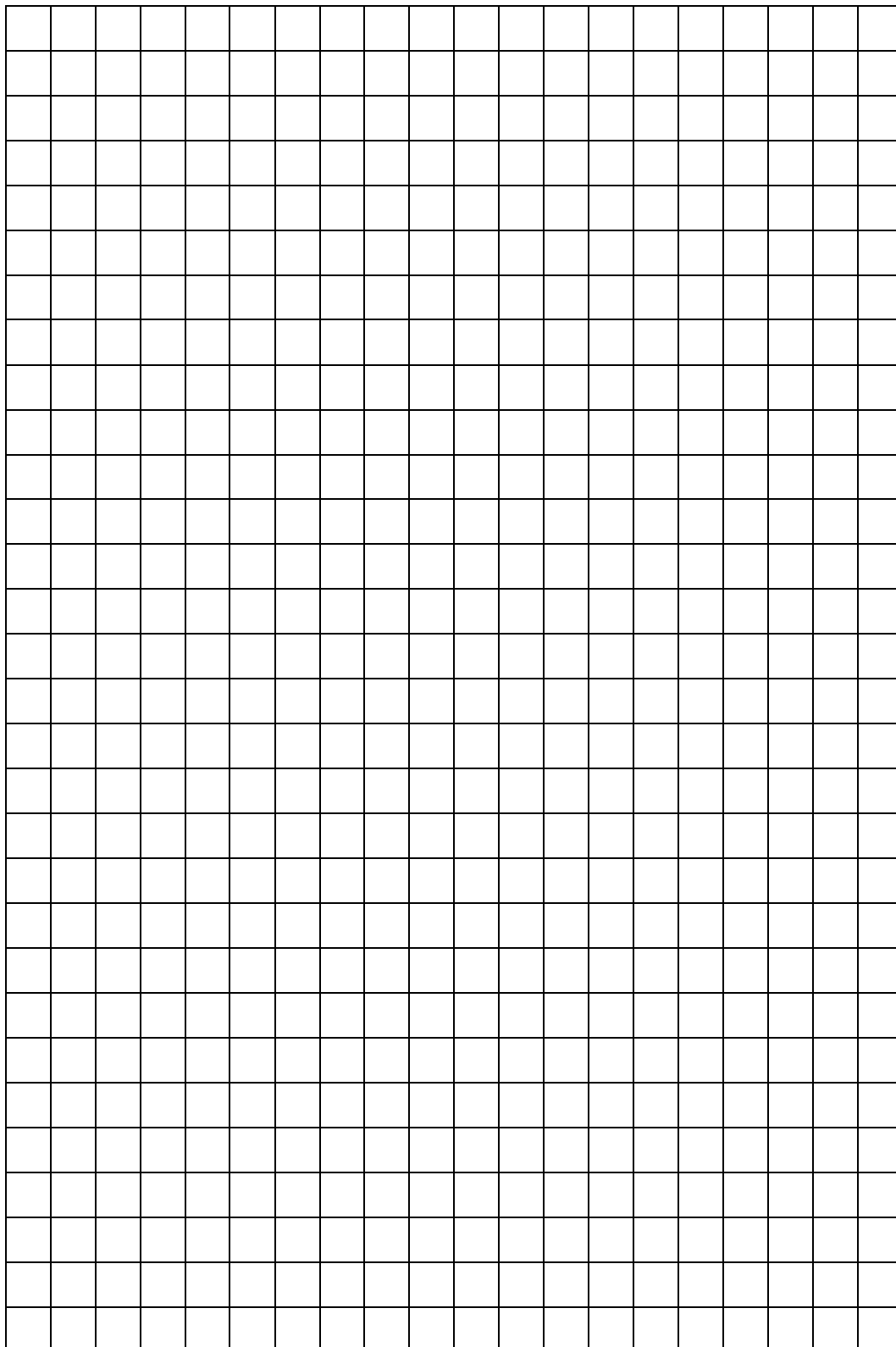
<u>1 paveikslas</u>														Kodas	
															0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0
															0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,1,0,0
															0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,1,0,0
															0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0
															0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0
															0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0
															0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0
															0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
															0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0
<u>2 paveikslas</u>															
															0,0,0,0,0,0,1,1,1,1,1,0,0,1,1,1,0,0
															0,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,1,0
															0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,1,0
															0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,1,0,0
															0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
															0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0
															0,0,1,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0
															0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0
															0,1,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,0
															1,0,0,1,0,0,1,1,0,0,0,0,0,1,0,0,0,0
															1,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0
															0,1,1,1,0,0,1,1,1,1,1,0,0,0,0,0,0,0

2 DARBO LAPAS . VAIZDŲ KODAVIMO PRADMENYS. PEIŠIMAS PAGAL KODĄ

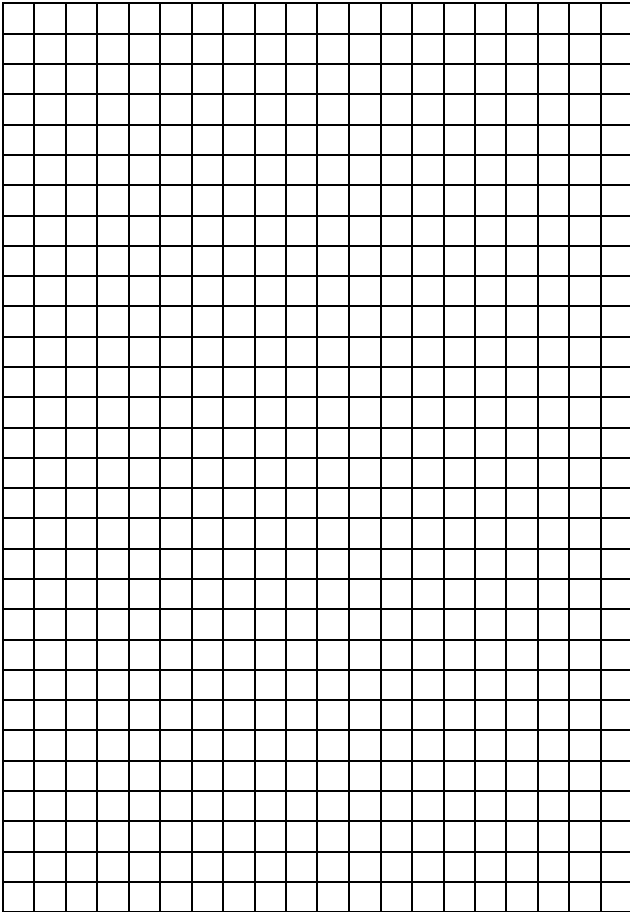
Kodas

DARBO LAPAI. VAIZDŲ KODAVIMAS. TINKLELIAI PAVEIKSLO IŠKODAVIMUI

Didelis tinklelis



Mažas tinklelis



Paveikslų kodai

<p>1 paveikslas</p> <p>2, 7 2, 7 2, 1, 1, 3, 1, 1, 1, 1 0, 11 0, 1, 1, 7 3, 1, 3, 1 2, 2, 3, 2</p>	<p>2 paveikslas</p> <p>5, 6 4, 1, 6, 1 3, 1, 7, 2 2, 2, 9, 1 1, 1, 12, 1 0, 1, 4, 1, 4, 1, 4, 1 0, 1, 14, 1 0, 1, 14, 1 0, 1, 3, 1, 6, 1, 3, 1 1, 1, 3, 1, 4, 1, 3, 1 2, 1, 3, 4, 4, 1 2, 1, 10, 1 3, 1, 8, 1 4, 2, 4, 2 6, 4</p>
<p>3 paveikslas</p> <p>4, 1, 5 3, 3, 4 2, 1, 3, 1, 3 1, 2, 1, 1, 1, 2, 2 0, 1, 7, 1, 1 0, 1, 1, 2, 1, 2, 1, 1, 1 0, 1, 1, 2, 1, 2, 1, 1, 1 0, 1, 4, 2, 1, 1, 1 0, 9, 1 4, 6</p>	<p>4 paveikslas</p> <p>1, 11 1, 1, 9, 3 1, 1, 9, 1, 1, 1 1, 2, 7, 2, 1, 1 2, 1, 7, 1, 2, 1 2, 2, 5, 5 3, 2, 3, 2 4, 5 0, 13 2, 9</p>