



**Vilniaus
universitetas**

Informatikos ir informatinio mąstymo mokomoji veikla

Linijų braižymas

Mokyklos pedagogika



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



**Vilniaus
universitetas**

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

Metodinė medžiaga „Linijų braižymas“ skirta informatikos ir informatinio mąstymo mokymui. Tikslinė grupė – būsimi informatikos pagrindinio ugdymo mokytojai. Medžiaga siejasi su Bendrosiomis informatikos ir matematikos programomis, algoritmų ir programavimo bei technologinių sprendimų pasiekimų sritimi. Atlikdami veikloje numatytas užduotis mokiniai išsiaiškins, kaip veikia kompiuteris, kai grafikos rengyklė braižo linijų ar apskritimų. Pateikiamas teorinis temos pagrindimas mokytojiui, aptariamos pagrindinės srities sąvokos.

Veikla sukurta pagal „Informatika be kompiuterio“ (*Computer Science Unplugged: csunplugged.org*) veiklą „Drawing lines“ pagal Creative Common licenciją.

Veiklą vertė ir adaptavo: dr. Eglė Jasutė

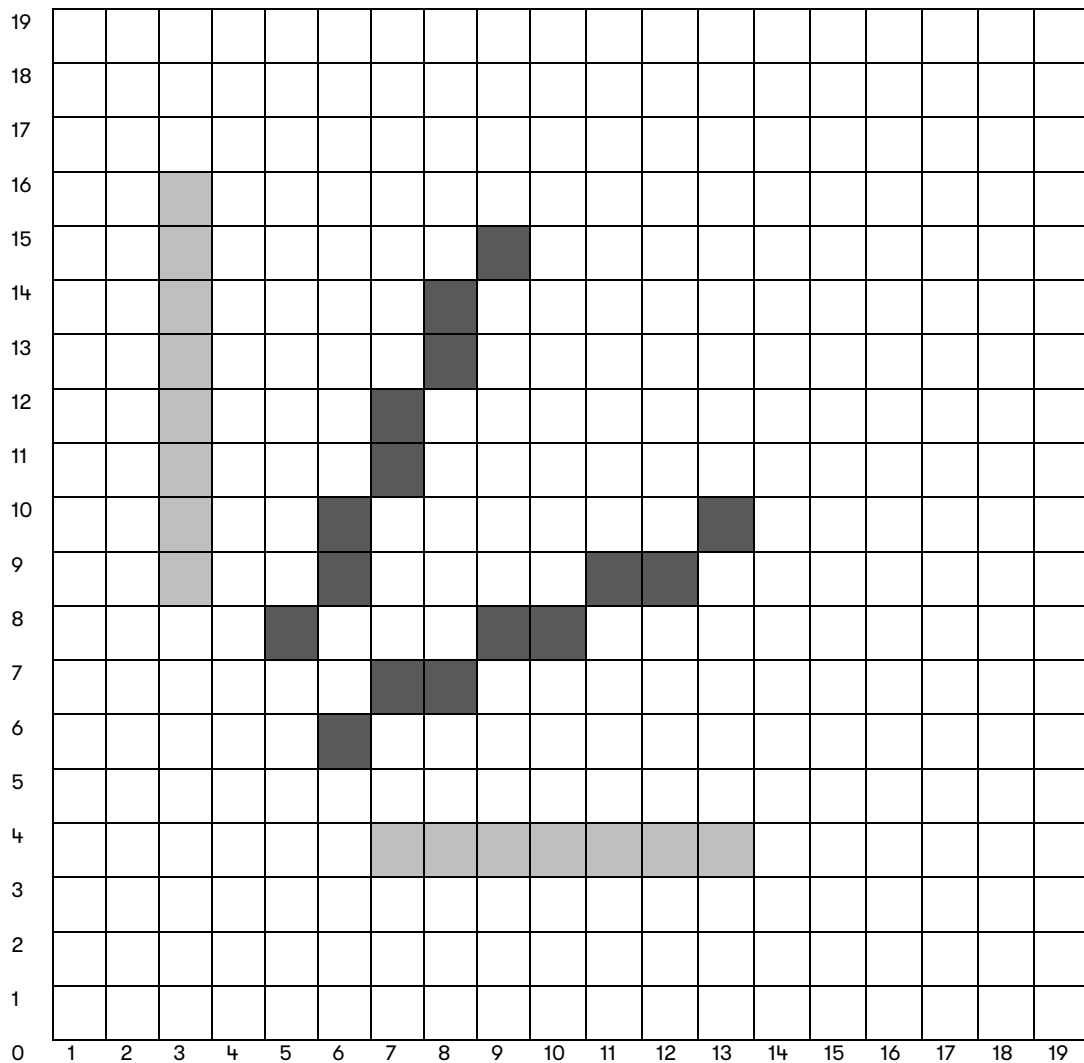
Redagavo: Viktoras Dagys

Apie ką visa tai?

Kompiuteris braižo linijas ir apskritimus atlikdamas nurodytas užduotis. Nuo žaidimų grafikos iki linijų braižymo ir net mažyčio apskritimo, skirto raidei nusakyti – taškui virš i raidės. Derinant linijas ir piešimą su užpildymu ir išlyginimu, kompiuteriai gali piešti tolygiai, nepriklausomai nuo kompiuterio raiškos. Kai vaizdas kompiuteryje aprašomas kaip kontūras ir užpildas, jis vadinamas vektorine grafika. Vektorinę grafiką galima perpiešti bet kokia raiška. Tai reiškia, kad, naudojant vektorinį paveikslėlį, priartinus vaizdą nebus matomi pikseliai, kurie matomi priartinus taškinės grafikos vaizdą. Kontūriniai šriftai yra vienas iš dažniausių vektorinės grafikos naudojimo būdų. Jie leidžia padidinti tekstą iki labai didelių dydžių, neprarandant kokybės.

Vektorinės grafikos programose kiekviena linija turi savo algoritmą, kuris pakeitus linijos padėtį ar jos dydį, kaskart perskaičiuojamas. Todėl vektorinės grafikos paveikslai nepriklauso nuo kompiuterio raiškos. Tačiau kompiuteryje ar spausdinant jie vis tiek turi būti atvaizduoti skaitmenine forma – pikseliais.

Linijų braižymas



Tikslas

Susipažinti su pagrindiniais kompiuterinių įrenginių darbo principais: kaip informacija, duomenys vaizduojami kompiuteriuose

Ryšys su bendrosiomis programomis

Informatika: algoritmo taikymas

Matematika: dvimatė koordinačių sistema, geometrija, liniuotės, skriestuvo naudojimas

Ugdomi įgūdžiai

Stačiakampė koordinačių sistema

Algoritmas

Matavimo ir braižymo priemonių naudojimas

Mokinių amžius

5 ir aukštesnės klasės: grafinis vaizdavimas kompiuteryje, taškinė ir vektorinė grafika, koordinatinių plokštuma, supažindinimas su algoritmais, naudojamais objektams braižyti kompiuterių programose.

Mokymosi priemonės

Pieštukas

Liniuotė, skriestuvai

Veiklų priemonės kiekvienam mokiniui:

Darbo lapas „Paprastoji linija“

Darbo lapas „Sudėtingesnė linija“

Darbo lapas „Bresenhamo linijos brėžimo algoritmas“

Darbo lapas „Pasvirusios tiesės“

Darbo lapas „Apskritimai“

Darbo lapas „Apskritimo centro algoritmas“

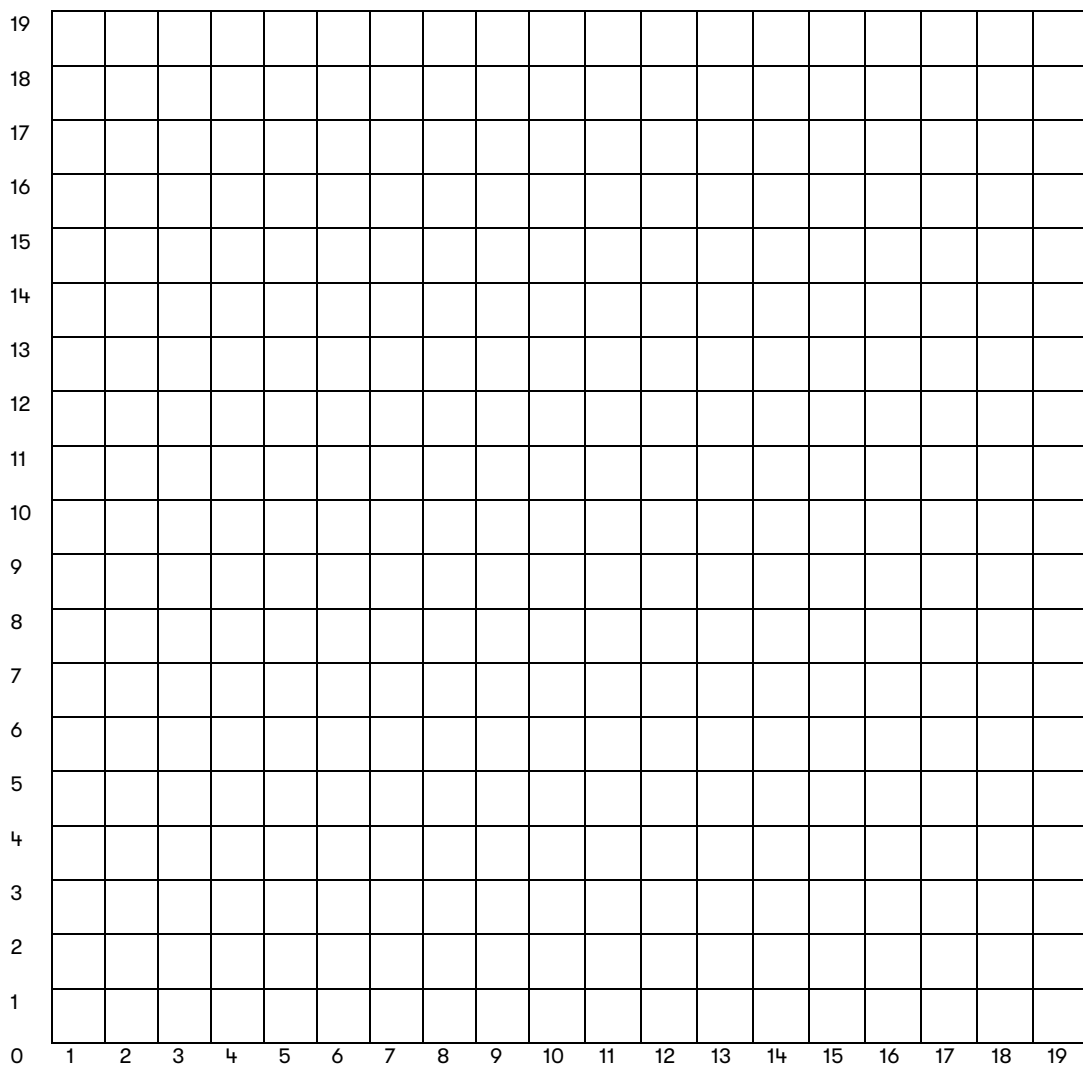
1 darbo lapas. Paprasta linija

Kompiuteriai piešia vaizdus naudodami pikselius. Pikseliai – tai maži kvadratėliai, iš kurių sudarytas kompiuterio monitoriuje matomas vaizdas. Jei atidžiai pažvelgsite į kompiuterio ekraną su didinamuoju stiklu, pamatysite atskirus pikselius.

Kad nubrėžtų liniją, kompiuteris turi išsiaiškinti, kuriuos pikselius reikia užpildyti, kad linija atrodytų tiesi. Tai galime išbandyti spalvindami kvadratėlius tinklėlyje.

Toliau esančiame tinklėlyje pabandykite nubrėžti šias tiesias linijas, užpildydami pikselius tinklėlyje:

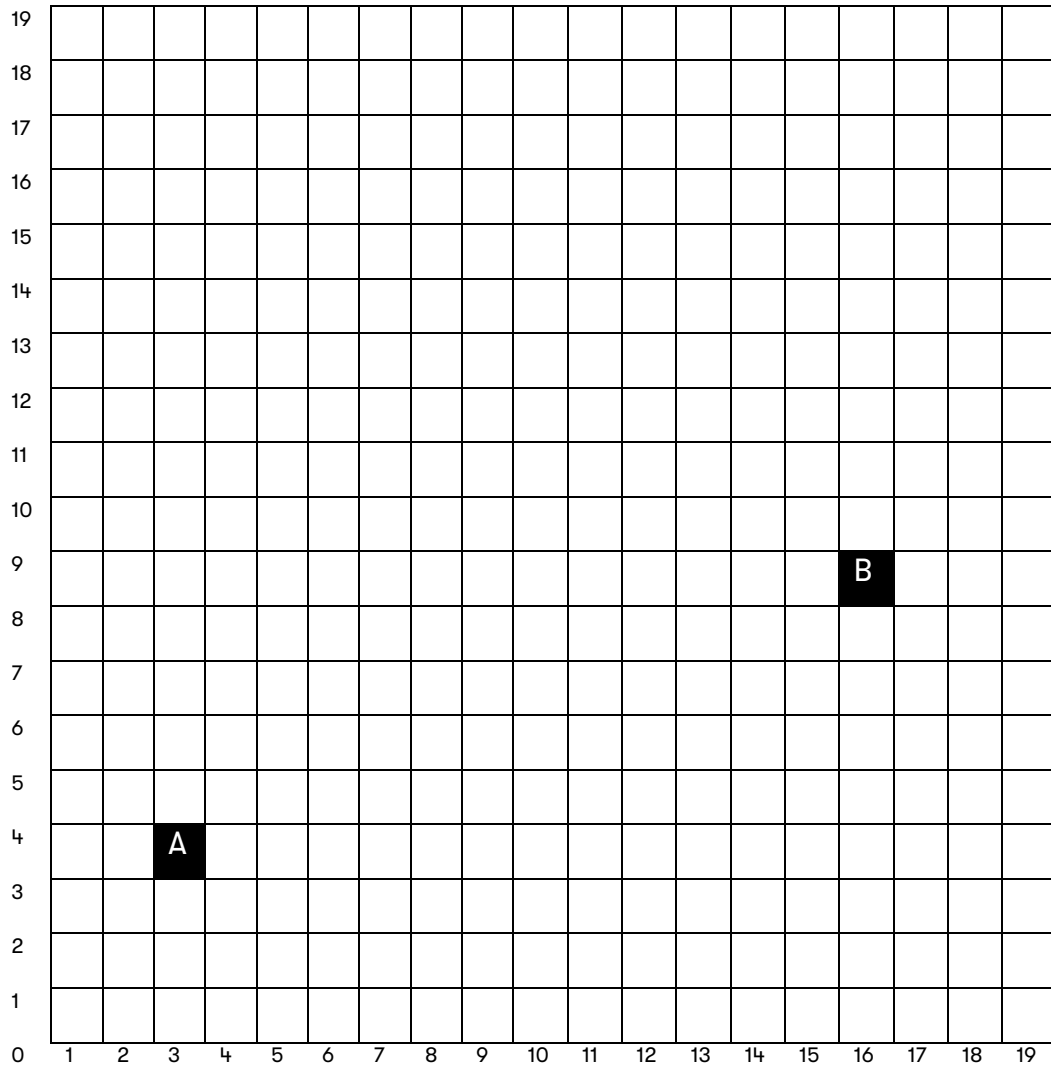
- $(2, 17) \rightarrow (10, 17)$
- $(18, 2) \rightarrow (18, 14)$
- $(1, 5) \rightarrow (8, 12)$



Patikrinkite su liniuote, ar jūsų linija tiesi.

2 darbo lapas. Sudėtingesnė linija

Ar galite nubrėžti tiesią liniją iš taško A į tašką B spalvindami langelius ir nenaudodami liniuotės?



Baigę brėžti liniją, pabandykite ją patikrinti liniuote. Liniuotę dėkite taip, kad ji eitų per A ir B langelių centrus. Ar liniuotė eina per visus langelius, kuriuos nuspalvinote?

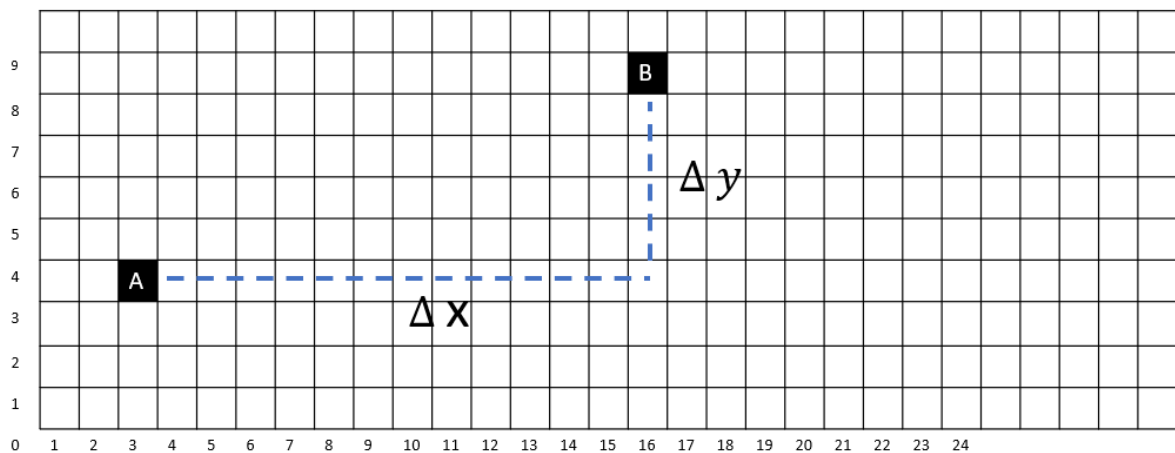
Bresenhamo linijos brėžimo algoritmas

Bresenhamo tiesės algoritmas pavadintas Jacko Eltono Bresenhamo, kuris jį sukūrė 1962 m. IBM bendrovėje, vardu. Bresenhamo algoritmas buvo išplėstas, kad galima būtų gauti apskritimus, elipses, kubines ir kvadratinės Bezerio kreives (<http://members.chello.at/easyfilter/bresenham.html>).

Vienas iš būdų, kaip kompiuteris gali sužinoti, kuriuos pikselius nuspalvinti, yra Bresenhamo algoritmas.

Pabandykite nubrėžti tą pačią atkarpą AB taikydami Bresenhamo algoritmą.

Algoritmas remiasi šiomis paprastomis taisyklėmis:

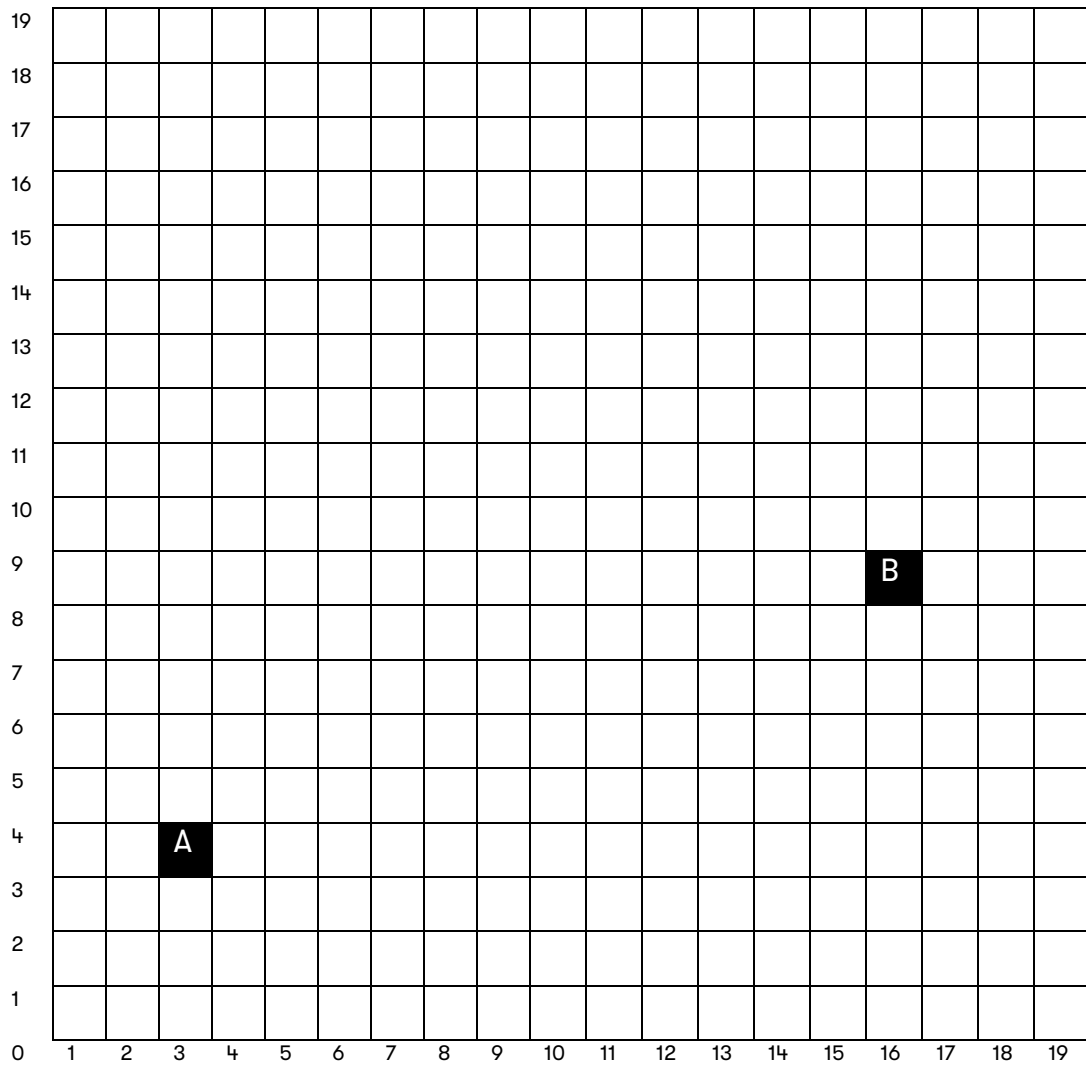


$$A(x_o; y_o) \quad B(x_1; y_1) \quad \Delta x = |x_1 - x_o|, \quad \Delta y = |y_1 - y_o|$$

- $P_o = 2 \cdot \Delta y - \Delta x$
- Nuspalvinkite pradinį tašką. Perskaičiuokite Δy ir Δx .
- Tada kiekvienai padėčiai išilgai X ašies:
 - jei $P < 0$, tai langelį spalvinkite ant tos pačios linijos, kaip ir ankstesnį,
 - ir prie P pridėkite $2 \cdot \Delta y$;
 - jei $P \geq 0$, tai langelį spalvinkite vienu aukščiau nei ankstesnį,
 - ir prie P pridėkite $2 \cdot \Delta y - 2 \cdot \Delta x$.

Tęskite šį procesą, kol pasieksite linijos pabaigą.

Be liniuotės, naudodami Bresenhamo linijos algoritmą, nubrėžkite tiesią liniją iš A į B.



Baigę brėžti liniją, patikrinkite ją liniuote. Palyginkite su pirmuoju savo variantu.

Pasvirusios tiesės

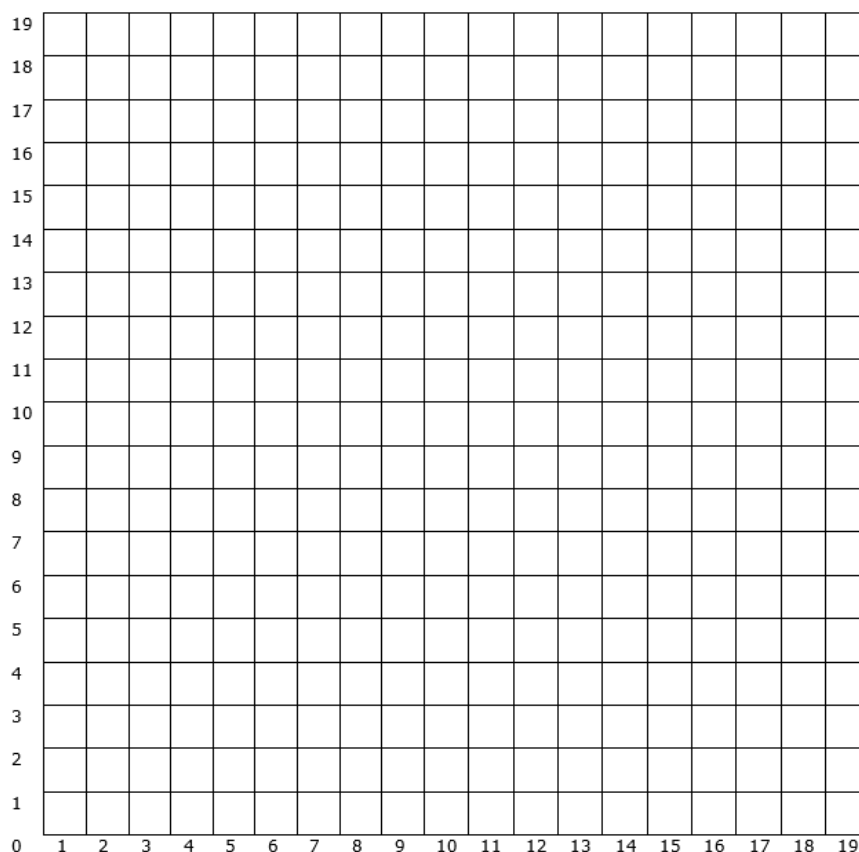
Iki šiol naudota Bresenhamo linijų piešimo algoritmo versija veikia tik toms linijoms, kurių pasvyrimas (nuolydis) yra nuo 0 iki 1. Tam kad algoritmas būtų bendresnis ir jį būtų galima naudoti bet kokiai linijai braižyti, reikia papildomų taisyklių:

- Jei linija yra pasvirusi žemyn, o ne į viršų, tada, kai P reikšmė yra 0 arba didesnė, tai kito stulpelio langelis spalvinamas viena eilute žemiau ankstesnio langelio, o ne virš jo.
- Jei Y vertės pokytis yra didesnis už X vertės pokytį, tada A , B ir pradinės P vertės skaičiavimus reikia keisti.

Apskaičiuodami A , B ir pradinę P reikšmę, naudokite X ten, kur anksčiau naudojote Y , ir atvirkščiai. Spalvindami langelius, užuot ėję per kiekvieną X ašies stulpelį, eikite per kiekvieną Y ašies eilutę, kiekvienoje eilutėje spalvindami po vieną langelį.

- $P_0 = 2 \cdot \Delta x - \Delta y$.
- Nuspalvinkite pradinį tašką. Perskaičiuokite Δy ir Δx .
- Tada kiekvienai padėčiai išilgai Y ašies:
 - jei $P < 0$, tai langelį spalvinkite ant tos pačios linijos, kaip ir ankstesnį,
 - ir prie P pridėkite $2 \cdot \Delta x$.
 - jei $P \geq 0$, tai langelį spalvinkite vienu aukščiau nei ankstesnį
 - ir prie P pridėkite $2 \cdot \Delta x - 2 \cdot \Delta y$.

Tęskite šį procesą, kol pasieksite linijos pabaigą.



Viršuje esančiame tinklelyje nubrėžkite **savo** atkarpą. Atkarpos pradiniam taškui pasirinkite X reikšmę – pirmąją savo vardo raidę, paverstą skaičiumi. (pvz., Barbora būtų 3, nes B yra trečioji abėcėlės raidė). Jei skaičius didesnis už 19, iš jo atimkite 20. Pradžios taško Y reikšmei pasirinkite skaičių, gautą iš antrosios jūsų vardo raidės.

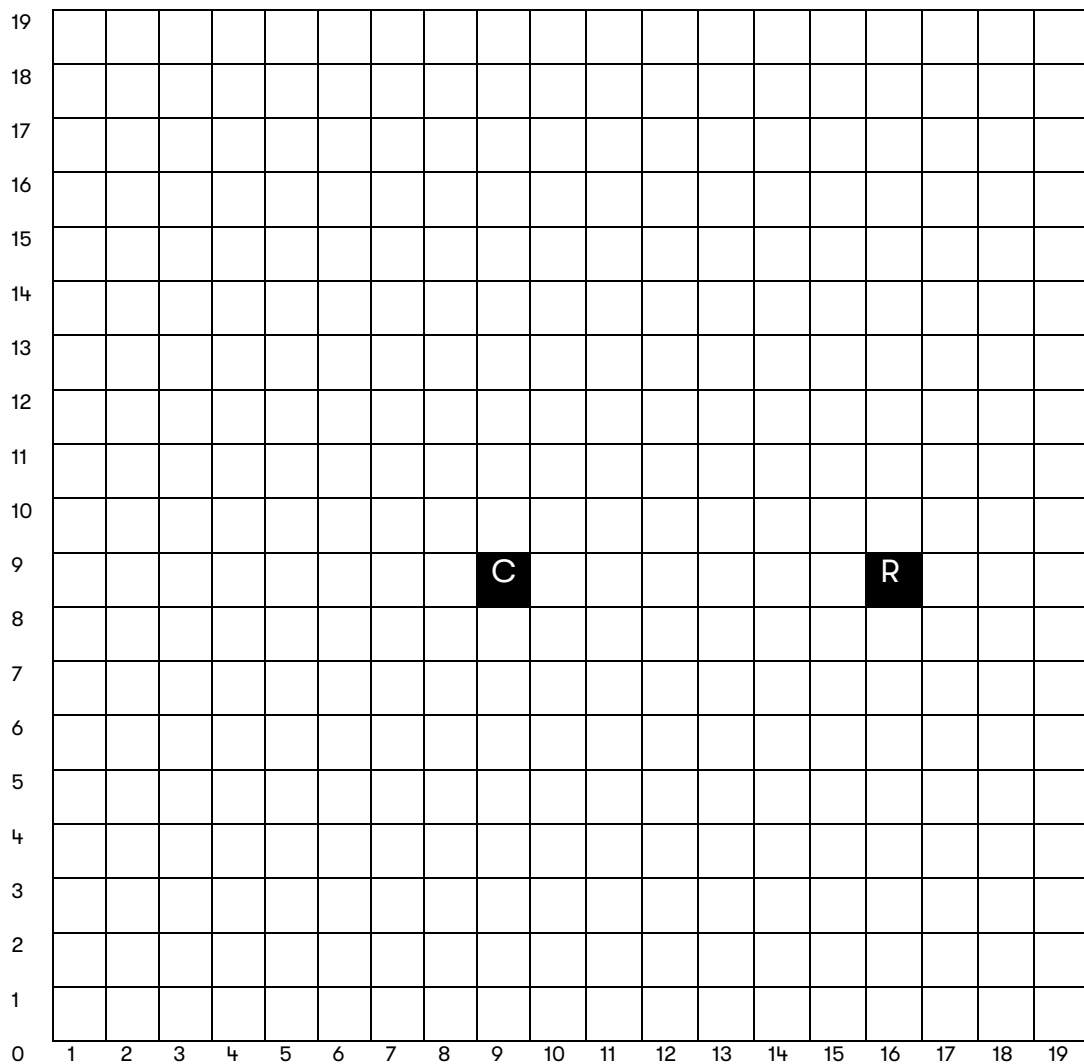
Atkarpos pabaigos taškui naudokite pirmąją ir antrąją savo pavardės raides. Pavyzdžiui, jei Jūsų vardas ir pavardė yra Jonas Smitas, naudokite raides „Jo“, kad pasirinktumėte pradžios tašką (16, 1). Tada naudokite „Sm“ raides, kad nustatytumėte galutinį atkarpos tašką (4, 19). Jei abi x ar y koordinatės sutampa, tai atkarpos galo koordinatėse sukeiskite x ir y vietas (kad užduotis būtų įdomesnė). Atkarpą nubrėžkite naudodami Bresenhamo algoritmą.

1	Aa	9	Ėė	17	Kk	25	Šš
2	Ąą	10	Ff	18	Ll	26	Tt
3	Bb	11	Gg	19	Mm	27	Uu
4	Cc	12	Hh	20	Nn	28	Ųų
5	Čč	13	Ii	21	Oo	29	Ūū
6	Dd	14	Jj	22	Pp	30	Vv
7	Ee	15	Yy	23	Rr	31	Zz
8	Ęę	16	Jj	24	Ss	32	Žž

Apskritimai

Be tiesių, kita įprasta forma, kurią dažnai reikia nubraižyti kompiuteriuose, yra apskritimai. Buvo sukurtas į Bresenhamo tiesių braižymo algoritmą panašus algoritmas, skirtas apskritimams braižyti, vadinamas apskritimo centro algoritmu (angl. *Midpoint Circle Algorithm*).

Apskritimą apibrėžia centro taškas ir spindulys. Visi apskritimo taškai yra spindulio atstumu nutolę nuo apskritimo centro. Centro taškas turi X ir Y koordinatas.



Pabandykite ranka nubrėžti apskritimą užpildydami langelius (nenaudodami liniuotės ar skriestuvo). Atkreipkite dėmesį, kaip sunku tai padaryti.

Apskritimo centro algoritmas

Pateikiame apskritimo centro algoritmą.

E = - spindulys

x = spindulys

$y = 0$

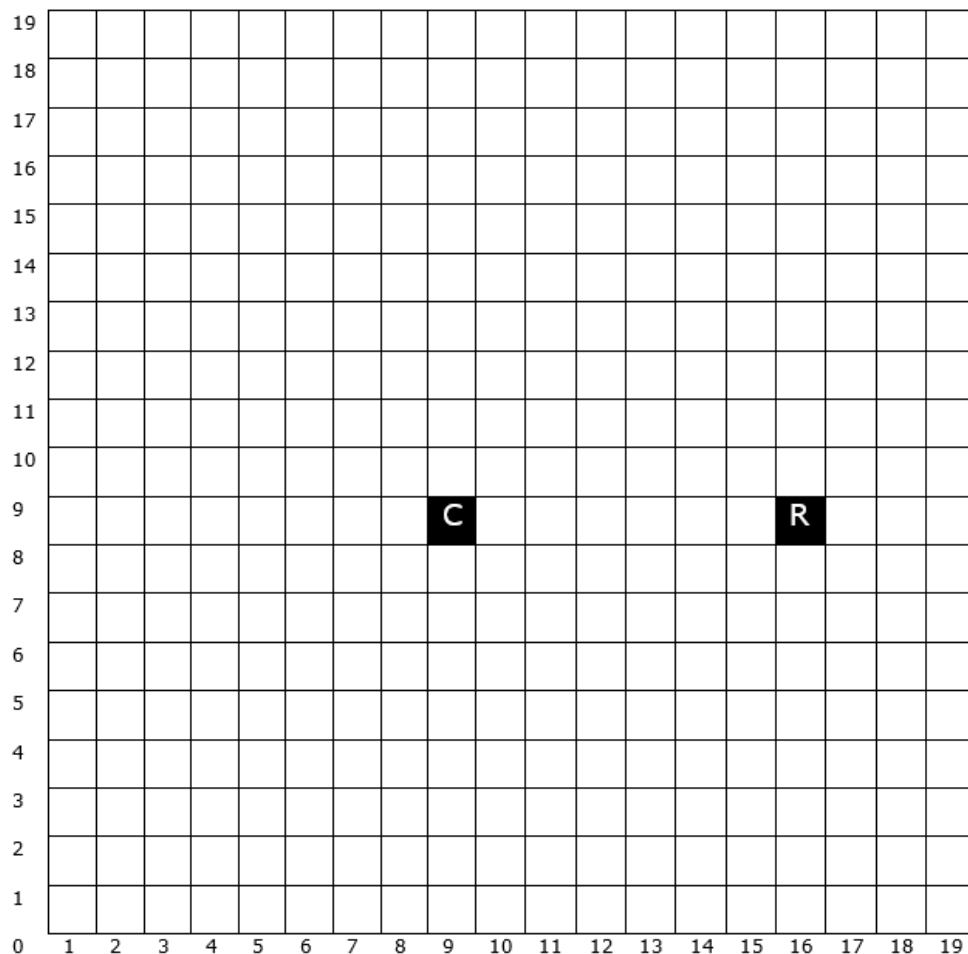
Centro C koordinatės (x_0 ; y_0)

Tol, kol y taps didesnis už x , kartokite šiuos žingsnius iš eilės:

- dėkite (nuspalvinkite) pikselį, kurio koordinatės ($x + x_0$, $y + y_0$)
- E padidinkite $2 \cdot y + 1$
- y padidinkite 1
- Jei E yra didesnis arba lygus 0, tai iš E atimkite $(2x - 1)$, tada iš x atimkite 1.

Vadovaudamiesi taisyklėmis, nubrėžkite apskritimą, kurio centras yra C .

C - apskritimo centras, CR – apskritimo spindulys



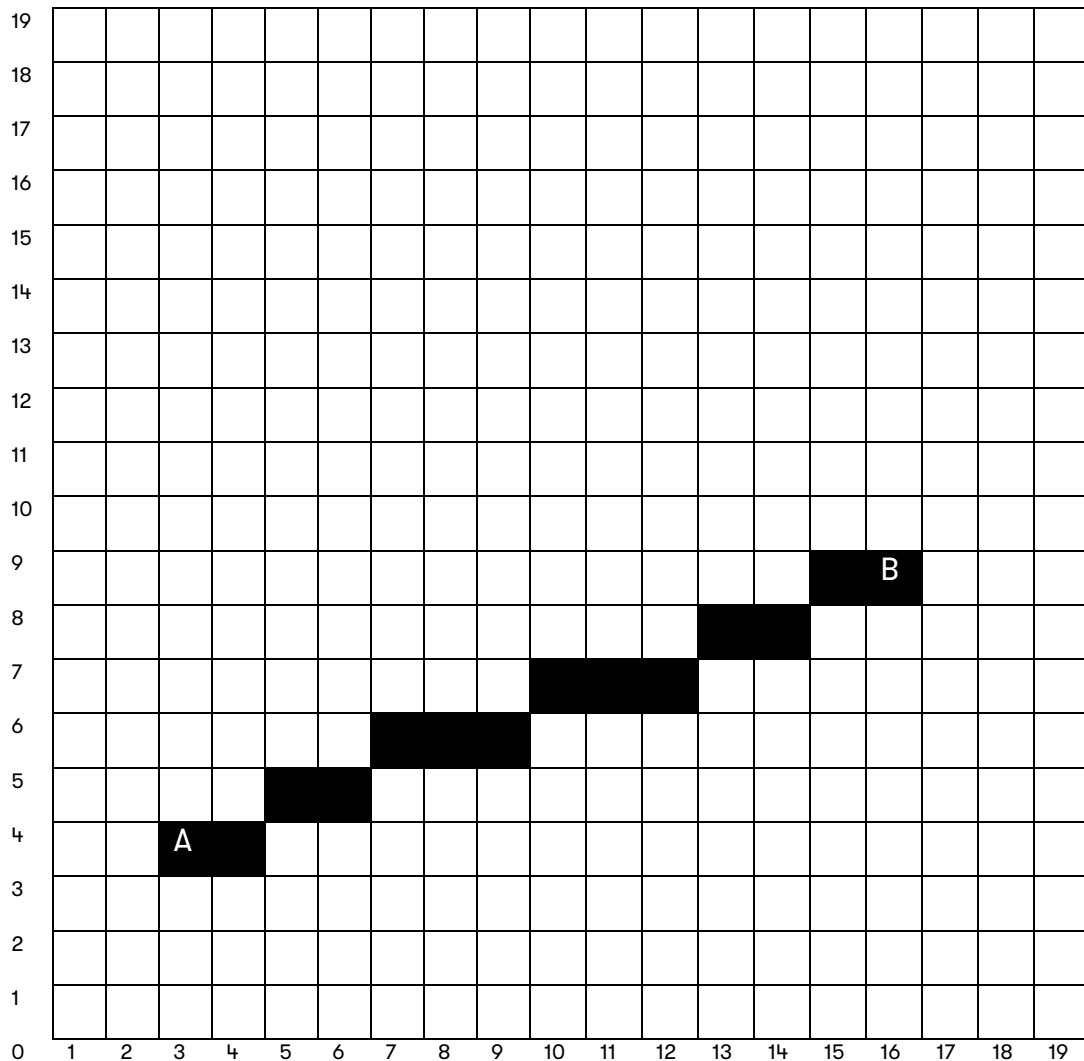
Kai Y tampa didesnis už X , nuspalvinama viena aštuntoji apskritimo dalis. Likusių apskritimo dalių galima nuspalvinti simetriškai jau turimai daliai. Nuspalvinkite langelius išilgai X ir Y ašių taip, kad simetrijos ašis kirstų apskritimo centro langelio vidurį. Dabar nuspalvinta pusė apskritimo – kairioji ir dešinioji pusė. Norint pridėti likusių apskritimo dalių, reikia naudoti kitą simetrijos ašį. Ar galite išsiaiškinti, kuri simetrijos ašis reikalinga apskritimui užbaigti? Prieš atsiversdami puslapį ir norėdami patikrinti atsakymą, pagalvokite, kokią simetriją naudoti.

Norėdami užbaigti apskritimą, turite pamąstyti apie įstrižainę. Simetrijos ašis turėtų būti su 1 arba -1 posvyriu ir kirsti apskritimo centro langelį.

Nors aštuntadalio simetrijos ašies naudojimą lengviau suprasti žmogui, kompiuteris visus simetriškus taškus gali nubraižyti tuo pačiu metu, kai spalvina pirmojo aštuntadalio langelį.

Atsakymų lapas „Bresenhamo linijos brėžimo algoritmas“

Atkarpa nuo A iki B



Teisingos suskaičiuotos reikšmės:

A = 10, B = -16

P0 = -3, dėti pikselį toje pačioje eilutėje kaip ir pradinis langelis.

P1 = 7, dėti pikselį virš ankstesniojo langelio.

P1 = -9, dėti pikselį toje pačioje eilutėje, kaip ir ankstesnįjį pikselį.

P3 = 1, dėti pikselį virš ankstesnio langelio.

P4 = -15, dėti pikselį toje pačioje eilutėje kaip ir ankstesnis langelis.

P5 = -5, dėti pikselį toje pačioje eilutėje kaip ir ankstesnis langelis.

P6 = 5, dėti pikselį virš ankstesnio langelio.

P7 = -11, dėti pikselį toje pačioje eilutėje kaip ir ankstesnis langelis.

P8 = -1, dėti pikselį toje pačioje eilutėje kaip ir ankstesnis langelis.

P9 = 9, dėti pikselį virš ankstesnio langelio.

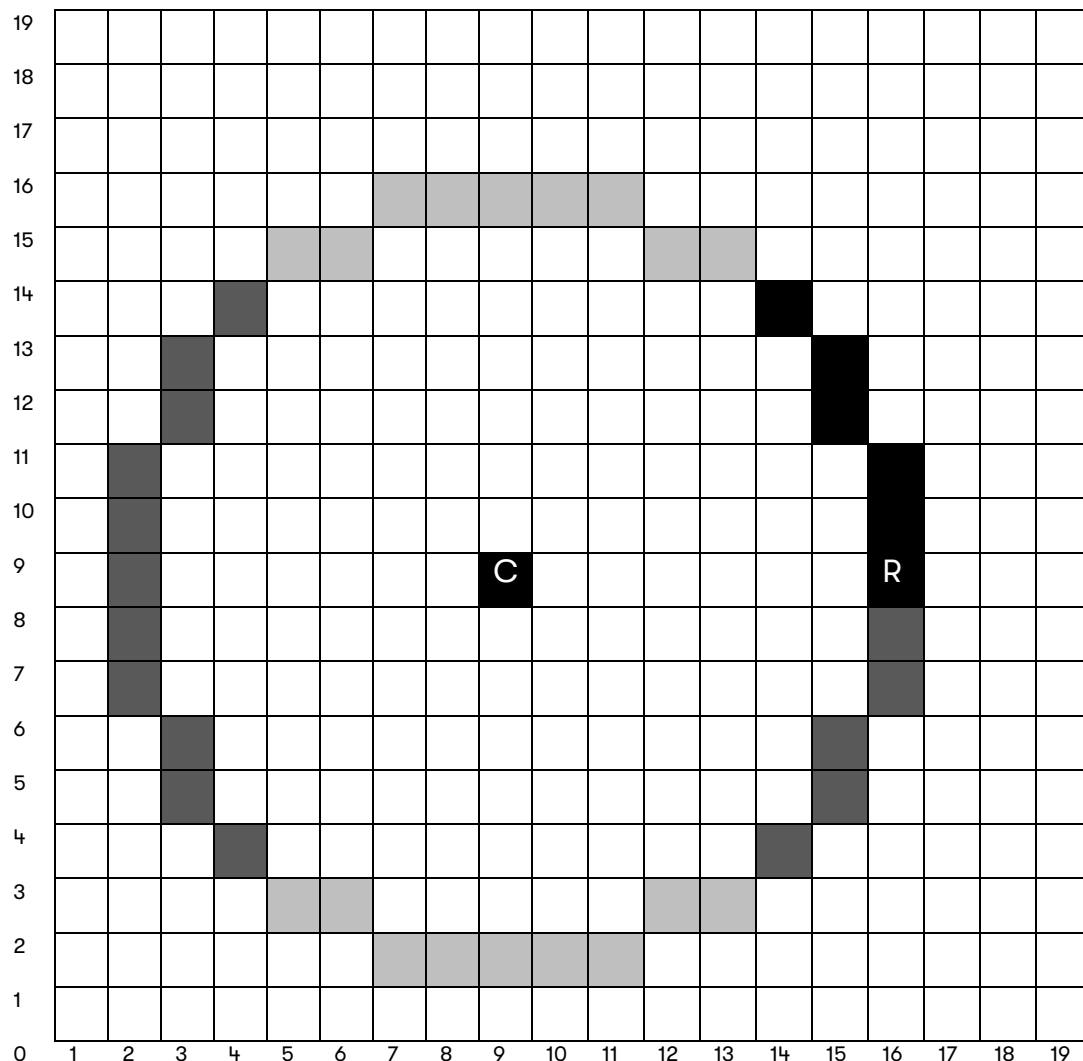
P10 = -7, dėti pikselį toje pačioje eilutėje kaip ir ankstesnis langelis.

P11 = 3, dėti pikselį virš ankstesnio langelio.

P12 = -13, dėti pikselį toje pačioje eilutėje kaip ir ankstesnis langelis.

Atsakymo lapas „Apskritimo centro algoritmas“

Teisingai paskaičiuoti pikseliai:



Juodi langeliai rodo pradinį nubrėžto apskritimo aštuntadalį. Tamsiai pilki langeliai rodo simetriją išilgai X ir Y ašių. Šviesiai pilki langeliai rodo simetriją įstrižos simetrijos ašies atžvilgiu

Algoritmo skaičiavimai:

$E_0 = -7$, $X_0 = 7$, $Y_0 = 0$; dėti pikselį (16, 9).

$E_1 = -6$, $Y_1 = 1$; dėti pikselį (16, 10).

$E_2 = -3$, $Y_2 = 2$; dėti pikselį (16, 11).

$E_3 = 2$, $Y_3 = 3$; $E_4 = -11$, $X_4 = 6$; dėti pikselį (15, 12);

$E_5 = -4$, $Y_5 = 4$; dėti pikselį (15, 13); $E_6 = 5$, $Y_6 = 5$;

$E_7 = -6$, $X_7 = 5$; dėti pikselį (14, 14);

$E_8 = 5$, $Y_8 = 6$; Y didesnis nei X, todėl galime atidėti simetriškai visą aštuntadalį.