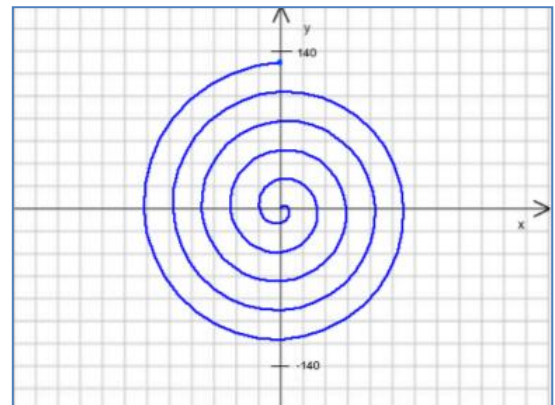


Joonistamine



Joonistamine toimub Scratchis suvalise spraidi skriptiga, milles kasutatakse plokkide gruppidest Pliats ja Liikumine. Joonistavaid objekte võib olla ka mitu, need võivad tegutseda kas kordamööda või üheaegselt.

Sisu

Sirg- ja kõverjoonte joonistamine	3
Koordinaatsüsteemid	5
Scratch'i koordinaatsüsteem.....	5
Parameetrilised koordinaadid	5
Polaarkoordinaadid	6
Joonistamise projektid Scratch'is	7
Projektid siinus.sb ja koosinus.sb	7
Projekt ring_polar.sb	9
Projekt ring_polar_1.sb	10
Soovituslikud ülesanded.....	10
Projekt ellips_1.sb	11
Projekt ellips_2.sb	13
Projekt astroid.sb	14
Soovituslikud ülesanded.....	13
Soovituslikud ülesanded.....	14
Projekt lill.sb	15
Soovituslikud ülesanded.....	15
Projekt a_spiraal.sb	16

Sirg- ja kõverjoonte joonistamine

Joonistada võib suvaline sprait. Iga spraidiga on alati seotud **pliiats**, mis võib olla üleval – spraidi liikumisel joont ei teki, või all – liikumisel tekib lavale joon. Joonistamisel kasutatakse peamiselt käske gruppidest **Pliiats** ja **Liikumine**.

Grupi **Pliiats** käskudega saab määrata joone värvi, varjundi ja suuruse, muuta pliiatsi olekut: üleval või all.

Grupi **Liikumine** käskudega määratakse ja muudetakse joonistava objekti asukohta ja joonte asukoht laval.

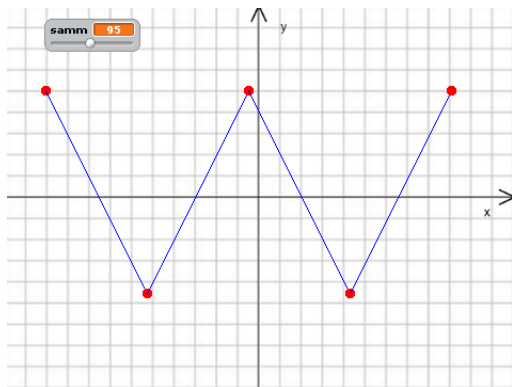
Laval ühest punktist teise liikumiseks võib määrata nende vahelist kaugust ja suunda või punktide koordinaadid x ja y .

Koordinaatide väärtused arvutatakse korduses ja vastav punkt (ning joon) joonistatakse spraiti sellesse punkti liigutades. Käsk **[mine x: ... y: ...]** viib spraidi koheselt antud punkti. Liikumise Käsuga **[liigu t sek. x: ... y: ...]** saab määrata spraidi sujuva liikumise. Ühikuks on piksel.

NB! Spraidi pliiats peab olema joonistamisasendis (all), vt plokk **pliiats all (pen down)** grupis **Pliiats (Pen)**.

Joonistamisplokkid ja nende vastavad tegevused on kirjeldatud järgmises tabelis.

Plokid	Selgitused
	kustutatakse joonis ja spraitide jäljendid laval
	pliiats alla, liikumisel jätab sprait joone
	pliiatsi üles, liikumisel joont ei teki
 	muutub pliiatsi värv (käsu loomisel saab värvi valida pipeti abil)
 	pliiatsi värv muutub antud väärtuse võrra
 	pliiatsi värviks võetakse antud väärtus: kui 0, siis värviringi punane, kui 100, siis sinine jne
 	pliiatsi tooni (varjundit) muudetakse antud väärtuse võrra
 	pliiatsi varjundiks võetakse antud väärtus: kui väärtus on 0, siis väga tume, kui väärtus on 100, siis väga hele
 	pliiatsi suurust muudetakse antud väärtuse võrra
 	pliiatsi suuruseks võetakse antud väärtus
	spraidi aktiivse kostüümi jäljendi jätmine lavale



Pildil olev lihtne joon saadakse skripti abil, mille täidab sprait **Joonistaja** ([zig_zag.sb](#)).

Selles skriptis leitakse viis korda punkti uued koordinaadid (x ja y), viiakse sprait uude asukohta (liikumisel jääb lavale sinine joon) ja tekitatakse jäljend (punktikujuline kostüüm).

Kõigepealt kustutatakse vana joon ning määratakse muutujate algväärtused.

Koordinaatide väärtuste muutmiseks kasutatakse muutujat *samm*, mille väärtust saab määrata lavaliuguri abil. Muutuja *i*

väärtust muutub 1 ja -1 vahel, neid kasutatakse y arvutamisel. Igale kordusele muutub x ühe sammu võrra ja y kahe sammu võrra.

Joonistaja liigutamisel (käsk **[mine x: x y: y]**) jääb lavale joon.

Joonistaja algoritm

protseduur Joonista

kustuta

muutujate algväärtustamine

pliiats üleval

mine x:0 y: 0

pliiats all

jälg

kordus

võta $i = -1 * i$

võta $x = x + samm$

võta $y = y + 2 * i * samm$

mine x: x y: y

jälg

lõpp kordus

lõpp

Joonistaja skript



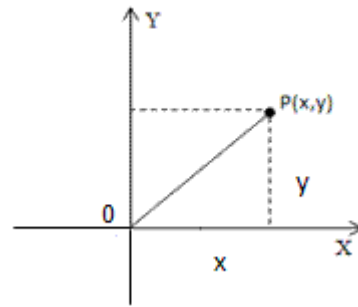
Selleks, et oleks näha selle joone tekkimist on vaja käsu asemel **mine x:x y: y** kasutada käsk **[liigu 1 sek. x: x y: y]** ([zig_zag1.sb](#)).

Koordinaatsüsteemid

Kui joonistamine toimub lava punktide koordinaatide arvutamise abil, võib kasutada mitmeid koordinaatsüsteeme.

Descartes'i koordinaatsüsteemi (ristkoordinaadid) moodustavad tasapinnal kaks teineteise suhtes ristuvat telge X ja Y . Teljed lõikuvad punktis O , mis on koordinaatide alguspunkt. Kummalgi teljel on määratud positiivne suund.

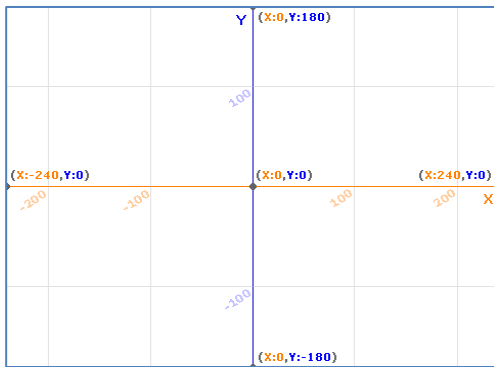
Iga punkt tasapinnal on määratud kahe koordinaadi (x ja y) väärtusega.



Sageli saab keeruliste kujundite (kõverate) võrrandeid esitada lihtsamalt ja lühemalt, kasutades kas polaarvõrrandeid või parameetrilisi võrrandeid (vt. allpool).

Scratch'i koordinaatsüsteem

Lava (*stage*) on koht, kus asuvad ja liiguvad spraidid. See on **480** ühikut lai ja **360** ühikut kõrge. Koordinaatide alguspunkt $(0, 0)$ asub lava keskel.



Hiirekursorit liigutades võib näha selle koordinaate (x, y) lava all paremas nurgas:

x: 84 y: -89

Klõpsates nuppe  lava ülaservas, saab valida väikese või suure lava. Väikese lava valimisel suureneb skriptide ala, see on kasulik mahukamate programmide korral.

Klõpsates esitlusoleku (*Presentation Mode*) nuppu , võib näha ja kasutada projekti tulemust täisekraanil. Sellest väljumiseks vajutada klahvi **Esc** või klõpsata noolt lava ülemises vasakpoolses nurgas.

Sageli on joonistamisel kasulik sisse tuua mastabitegur, et joonis oleks täielikult laval näha (ei väljuks lava piiridest).

Parameetrilised koordinaadid

Parameetiline esitus annab funktsiooni uurimisel olulise eelise nendel juhtudel, kui funktsioon on esitatud ilmutamata kujul ja selle teisendamine lihtsamale kujule on raskendatud ilma parameetriliste või polaarkoordinaatide kasutamiseta.

Parameetrilises võrrandis esitatakse muutujad x ja y lisamuutuja t (nn parameetri) abil.

Funktsiooni $y=f(x)$ esitust kujul $x=\varphi(t)$ ja $y=\psi(t)$, kus $t \in T$, $\{\varphi(t) \mid t \in T\}=X$ ja $\forall t \in T (f(\varphi(t))=\psi(t))$ nimetatakse funktsiooni f parameetriliseks esituseks.

Abstraktselt määrab parameetiline võrrand suhte võrrandite kogumina.

Kõige lihtsam näide on parabooli võrrand $y=x^2$, mille võib esitada parameetrit t kasutades järgmiselt:

$x=t$ ja $y=t^2$, kus $-\infty \leq t < \infty$.

Keerukamaks näiteks võiks olla ringjoone võrrand $x^2+y^2=R^2$, mille võib esitada parameetrilisel kujul:

$x=R \cdot \cos(t)$ ja $y=R \cdot \sin(t)$, kus $0 \leq t < 2\pi$.

Polaarkoordinaadid

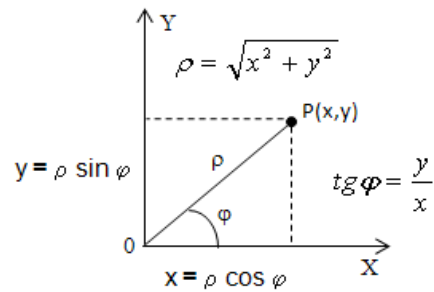
Polaarkoordinaadid on kahemõõtmeline koordinaatsüsteem, kus iga punkt P tasandil on üheselt määratud kaugusega ρ fikseeritud punktist (koordinaatide alguspunktist ehk poolusest) ning nurgaga φ fikseeritud suunast.

Üleminek polaarkoordinaatidelt ristkoordinaatidele toimub järgmiste valemite abil:

$$x = \rho \cos(\varphi) \text{ ja } y = \rho \sin(\varphi)$$

$$\rho = \sqrt{x^2 + y^2}$$

$$\tan \varphi = \frac{y}{x}$$



Joonistamine Scratch'is

Esitame kõigepealt mõned projektid lihtsamate graafikute tekitamiseks:

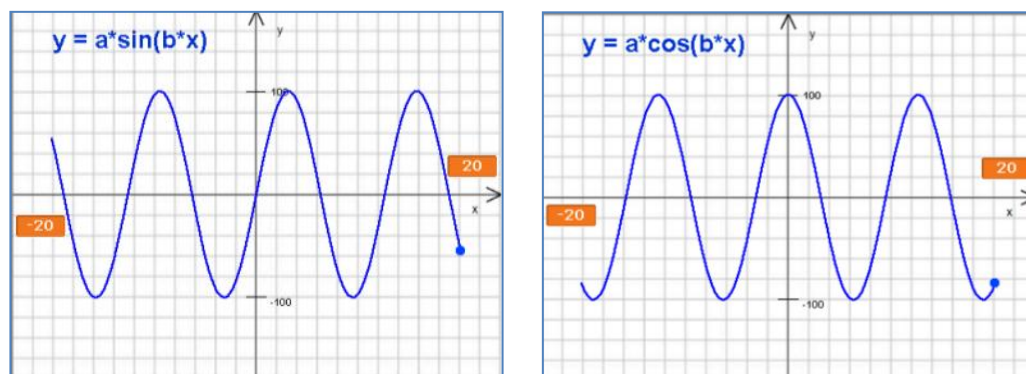
$$y = a * \sin(b * x)$$

$$y = a * \cos(b * x)$$

Projektid graafikute $y = a * \sin(b * x)$ ja $y = a * \cos(b * x)$ joonistamiseks erinevad ainult kasutatava funktsiooni poolest.

Projektid siinus.sb ja koosinus.sb ([siinus.sb](#) ja [koosinus.sb](#))

Nendes projektides on näidatud siinuse ja koosinuse graafikute loomine objekti **Joonistaja** abil.



Korrektse pildi saamiseks on kasutusel mastabitegur m , mille väärtus sõltub argumenti x algus- ja lõppväärtusest ning lava algus- ja lõpp-punktidest x -teljel.

Andmed

a ja **b** – funktsioonide parameetrid, kus a väärtus võib muutuda 5 ... 160, (pildil $a = 100$); b väärtus võib muutuda 0,1 ... 2,5 (pildil $b = 0,5$). Graafik on liiga tihe, kui b väärtus on suurem kui 2,5.

Muutujad: **x**, ja **y** – argumenti ja funktsiooni väärtused; **alg** ja **lõpp** – muutuja x alg- ja lõppväärtused, lõpp = -alg, mõistlikud väärtused 5...50, pildil alg = -20, lõpp = 20;

n – x ja y väärtuste arv (korduste arv), pildil $n = 200$; samm – x muutumise samm, leitakse $(\text{lõpp}-\text{alg})/n$

m – mastabitegur, arvutatakse valemi abil $400/(\text{lõpp}-\text{alg})$. Lava laius on 480 ühikut, graafiku laiuseks laval on valitud 400 ühikut.

Spraidid

Joonistaja – üks **skript** kõikide väärtuste määramiseks ja graafikute joonistamiseks. Tööd alustab vajutamine rohelisele lipule.

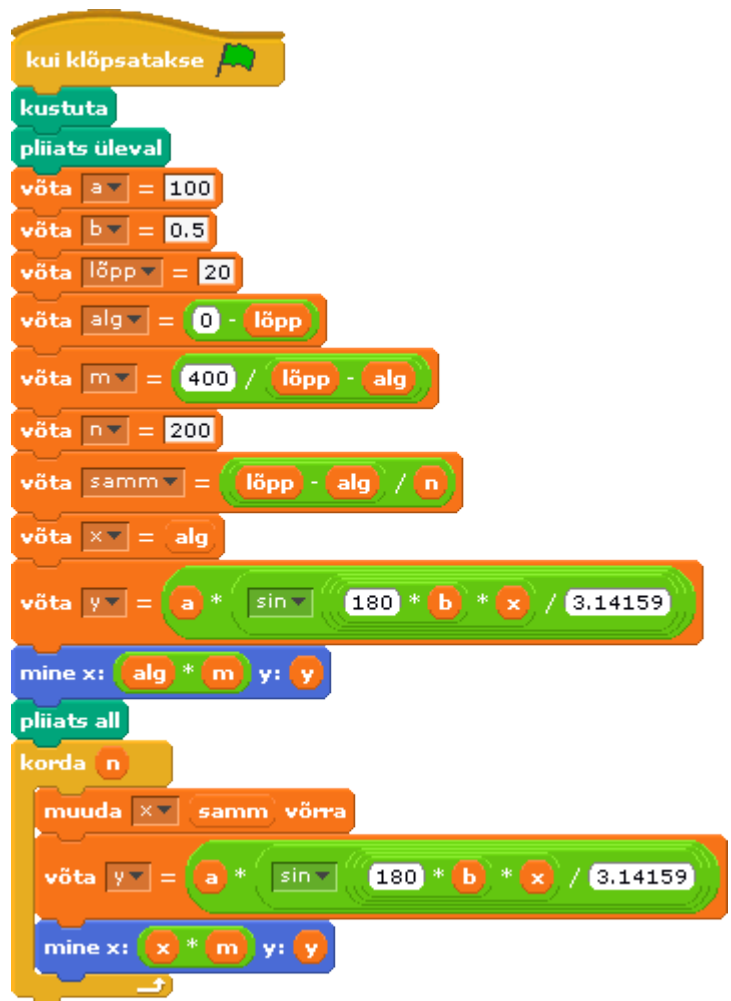
Allpool on toodud siinusfunktsiooni joonistamise algoritm ja skript. Koosinuse joonistamine on praktiliselt sama — y väärtuste arvutamise blokis on siinuse asemel koosinus. Kuna siinuse ja koosinuse argumentid võetakse radiaanides (x -telje väärtused), siis funktsioonid näevad välja järgmiselt:

$$y = a * \sin(180 * b * x / 3.14159) \text{ ja } y = a * \cos(180 * b * x / 3.14159).$$

Joonistaja algoritm

```
protseduur Joonista
  pliiats üleval
  kustuta
  võta a=100
  võta b=0.5
  võta lõpp=20
  võta alg=-lõpp
  võta m=400/(lõpp-alg)
  võta n=200
  võta samm=(lõpp-alg)/n
  võta x=alg
  võta y=a*sin(b*x)
  võta pliiatsi suuruseks 2
  mine x:alg*m y:y
  pliiats all
  kordus n korda
    muuda x samm võrra
    võta y=a*sin(b*x)
    mine x:x*m y:y
  lõpp kordus
lõpp
```

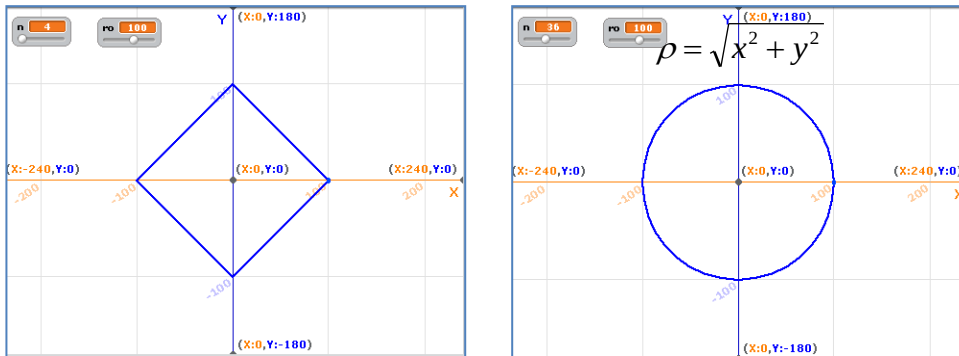
Joonistaja skript



Projekt ring_polar.sb ([ring_polar.sb](#))

Selles projektis on näidatud ringjoone joonistamist objekti **Joonistaja** abil. Kasutatakse ringjoone võrrandit polaarkoordinaatides $\rho = r_o$, kus r_o on ringjoone raadius. Joone keskpunkt asub lava keskpunktis (0,0).

Joonistamine toimub hulknurkade abil, kusjuures muutuja **n** väärtus määrab külgede arvu. Muutujate r_o ja n väärtused määratakse ja muudetakse laval liuguri abil, kusjuures r_o võib muutuda 20 ... 150 ja n 3 ... 75. Kui n on suurem kui **20**, tekib ringjoon, muidu joonistatakse hulknurk. Näiteks kui $n = 4$, saadakse ruut, kui $n = 36$, saadakse ringjoon.



Ringjoone ristkoordinaadid arvutatakse järgmiste võrrandite abil:

$$\rho = r_o, \quad x = \rho \cdot \cos(\varphi), \quad y = \rho \cdot \sin(\varphi)$$

Andmed

Muutujad: x, y – ringjoone punktide koordinaadid; r_o – ringjoone raadius; φ – polaarnurk, mis muutub 0 ... n sammuga h ; n – hulknurga külgede arv; h – nurga muutumise samm, arvutatakse $360/n$.

Sprait

Joonistaja, üks **skript** ringjoone joonistamiseks. Skriptis määratakse muutujate φ ja h väärtused, kusjuures igas korduses suureneb φ väärtus h võrra. Tööd alustab vajutamine rohelisele lipule.

Joonistaja algoritm

```
protseduur Joonista
  pliiats üleval
  kustuta
  võta  $\varphi = 0$ 
  võta  $h = 360/n$ 
  võta pliiatsi suuruseks 2
  mine  $x:0$   $y:0$ 
  pliiats all
  kordus  $n$  korda
    võta  $\varphi = \varphi + h$ 
    võta  $y = r_o \cdot \sin(\varphi)$ 
    võta  $x = r_o \cdot \cos(\varphi)$ 
    mine  $x:x$   $y:y$ 
  lõpp kordus
lõpp
```

Joonistaja skript



Järgmises projektis on algandmete ja muutujate väärtuste määramine ja arvutamine eraldatud joonistamisest ning antud üle lava skriptile. Objektid suhtlevad omavahel teadete abil.

Projekt ring_polar_1.sb ([ring_polar_1.sb](#))

Selles projektis on näidatud ringjoone joonistamine objekti **Joonistaja** abil. Kasutatakse ringjoone võrrandit polaarkoordinaatides $\rho = r_0$, kus r_0 on ringjoone raadius. Vaikimisi r_0 on väärtus 80, kuid seda saab muuta sisestades soovitud raadiuse suuruse vahemikus 20 ... 150.

Joonistamine toimub hulknurkade abil – muutuja n väärtus määrab külgede arvu. Vaikimisi on n väärtus 36, kuid seda saab muuta sisestades soovitud külgede arvu 3 ... 75. Kui n on suurem kui 20, tekib ringjoon, muidu joonistatakse hulknurk. Näiteks kui $n=3$, on tulemuseks kolmnurk.

Ringjoone koordinaadid arvutatakse võrrandite abil: $\rho = r_0$, $x = \rho \cos(\varphi)$, $y = \rho \sin(\varphi)$ ja $\rho = \sqrt{x^2 + y^2}$

Andmed

Muutujad: x , y – ringjoone punktide koordinaadid; r_0 – ringjoone raadius; φ_i – polaarnurk, muutub 0 ... n sammuga h ; n – hulknurga külgede arv; h – nurga muutumise samm, arvutatakse valemi abil $h = 360/n$.

Spraidid

Seletus – kasutusjuhend, kaks **skripti**: roheline lipu skript (ilmub roheline lipu vajutamisel) ja hiireklõpsu skript. Hiireklõps kasutusjuhendil viib tegevused edasi teatega „andmed“.

Lava, üks **skript** algandmete ja muutujate väärtuste määramiseks. Väärtused r_0 ja n valitakse kas vaikimisi või antakse võimalus sisestada andmeid klaviatuurilt. Pärast väärtuste määramist saadetakse teade 'joonista' **Joonistajale**.

Joonistaja, üks **skript** ringjoone joonistamiseks. Käivitub, kui saabub teade 'joonista'. Kordust täidetakse n korda: arvutatakse nurga ja koordinaatide väärtused ning joonistatakse vastav osa ringjoonest.

Seletuse skriptid



Joonistaja skript



Lava skript



Soovituslikud ülesanded

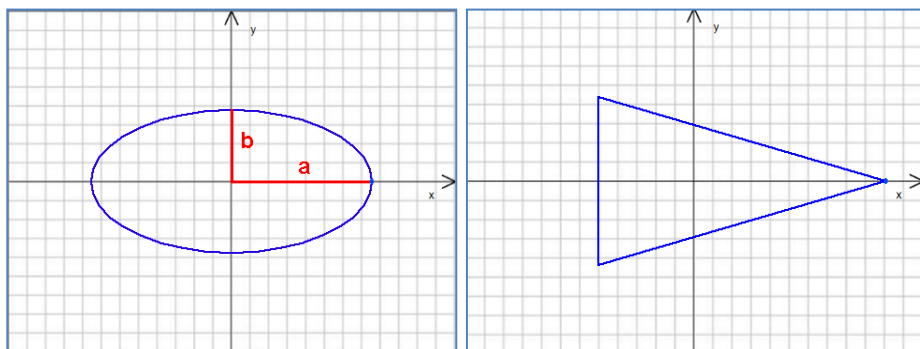
1. Andmete ja muutujate väärtuste määramine toimub juhuarvude abil

Projekt ellips_1.sb ([ellips_1.sb](#))

Selles projektis on näidatud ellipsi joonistamine objekti **Joonistaja** abil. Ellipsi joonistamisel kasutatakse koordinaatide x ja y arvutamiseks parameetrilised võrrandid, milles parameetrik on nurk φ , mis muutub $0 \dots n$. Ellipsi suur pooltelg on a , väike pooltelg b . Vaikimisi on a väärtus 150 ja b väärtus 75; neid saab muuta, sisestades soovitud suurused vahemikest 10 ... 200 ning 10 ... 100.

Joonistamine toimub hulknurkade abil, muutuja n väärtus määrab külgede arvu. Vaikimisi n väärtus on 36, seda saab muuta, sisestades soovitud külgede arvu 3 ... 75.

Kui n on suurem kui 20, tekib ellips, muidu joonistatakse hulknurk. Näiteks kui $n = 3$, saadakse kolmnurk.



Ellipsi koordinaadid arvutatakse järgmiste võrrandite abil:

$$x = a \cos(\varphi), y = b \sin(\varphi)$$

Andmed

Muutujad: x, y – ellipsi punktide koordinaadid; φ – parameeter, mis muutub $0 \dots n$ sammuga h ; n – hulknurga külgede arv; h – nurga muutumise samm, arvutatakse valemi abil $h = 360/n$.

Spraidid

Seletus – kasutusjuhend, kaks **skripti**: rohelise lipu skript ja hiireklõpsu skript. Rohelisele lipule vajutamisel ilmub lavale seletus. Hiireklõps sellel spraidil viib tegevused edasi teate '*andmed*' abil.

Lava, üks **skript** algandmete ja muutujate väärtuste määramiseks. a , b ja n väärtused määratakse vaikimisi või antakse võimalus sisestada need klaviatuurilt. Pärast väärtuste omistamist saadetakse teade '*joonista*' **Joonistajale**.

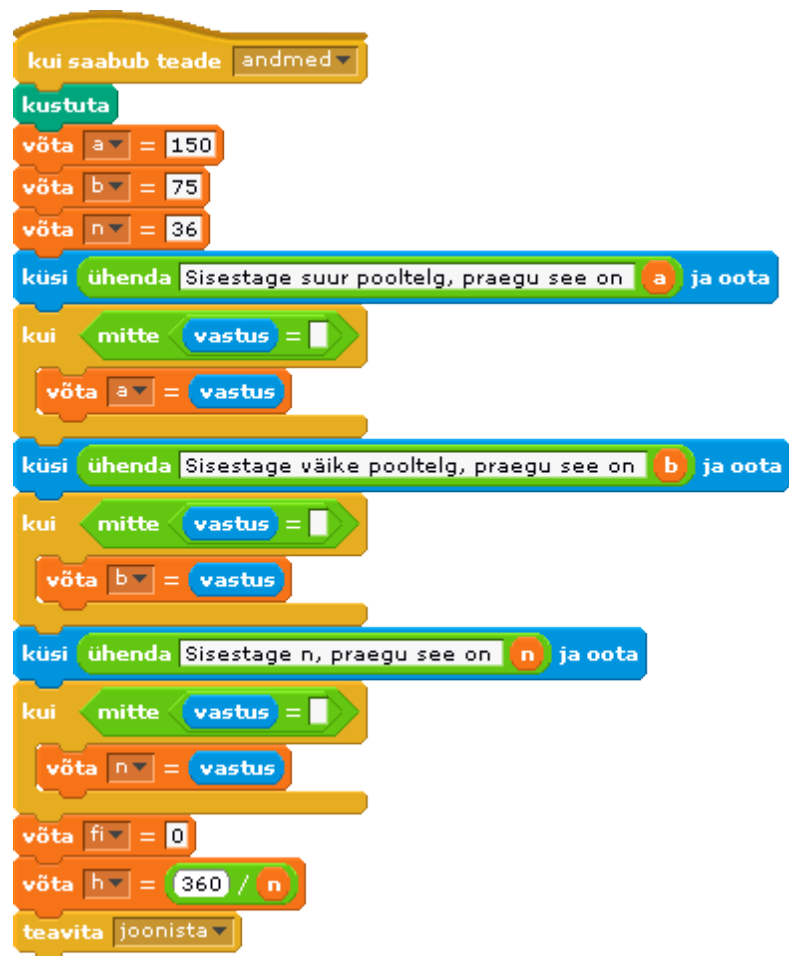
Joonistaja, üks **skript** ellipsi joonistamiseks, käivitub teatega '*joonista*'. Kordust täidetakse n korda: arvutatakse nurga ja koordinaatide väärtused ning joonistatakse vastav osa ellipsist.

Kõikide objektide skriptid on analoogsed eelmise projekti skriptidele.

Joonistaja skript



Lava skript



Projekt ellips_2.sb ([ellips_2.sb](#))

Selles projektis on näidatud ellipsi joonistamine objekti **Joonistaja** abil. Ellipsi joonistamisel kasutatakse koordinaatide x ja y arvutamiseks parameetrilised võrrandid, milles parameetriks on nurk φ , mis muutub $0 \dots n$. Ellipsi suur pooltelg on a , väike pooltelg b . Vaikimisi on a väärtus 150 ja b väärtus 75, neid saab muuta, sisestades soovitud suurused vastavalt vahemikest 10 ... 200 ning 10 ... 100.

Joonistamine toimub hulknurkade abil, kusjuures muutuja n väärtus määrab külgede arvu. Vaikimisi on n väärtus 36, seda saab muuta, sisestades soovitud külgede arvu 3 ... 75 klaviatuurilt. Kui n on suurem kui 20, siis tekib ellips, muidu joonistatakse hulknurk. Näiteks kui $n=3$, saadakse kolmnurk.

Ellipsi koordinaadid arvutatakse võrrandite abil: $x = a \cos(\varphi)$, $y = b \sin(\varphi)$

Andmed

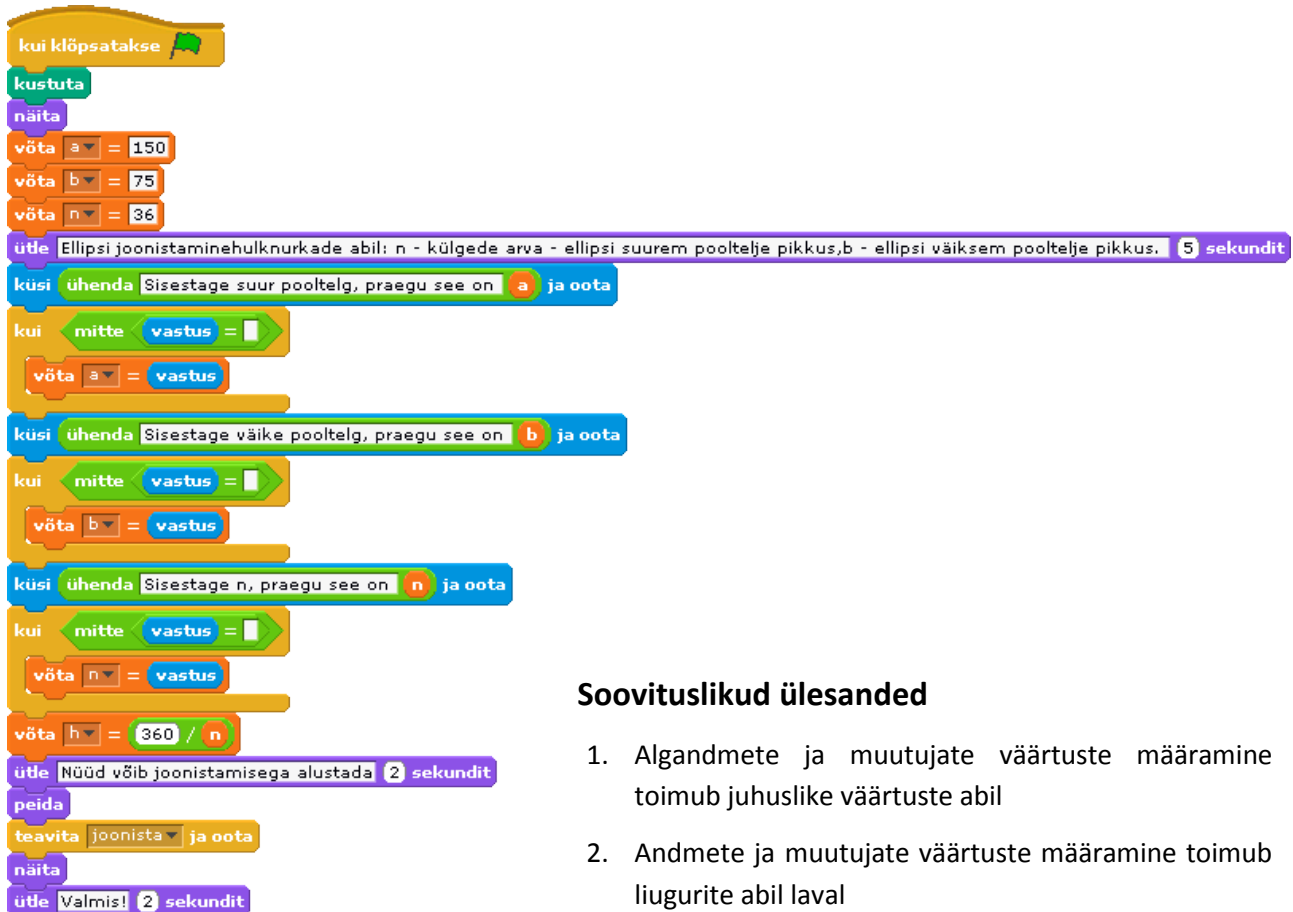
Muutujad: x , y – ellipsi punktide koordinaadid; φ – parameeter, muutub $0 \dots n$ sammuga h ; n – hulknurga külgede arv; h – nurga muutumise samm, arvutatakse $360/n$.

Spraidid

Kraps, üks **skript** seletuste andmiseks ning algandmete ja muutujate väärtuste määramiseks (täidab eelmise projekti kahe objekti skriptide tegevused); käivitub vajutamisel rohelisele lipule. Väärtused a , b ja n valitakse kas vaikimisi või antakse kasutajale võimalus sisestada andmed klaviatuurilt. Pärast väärtuste määramist Kraps saadab teade 'joonista' **Joonistajale**. Kui **Joonestaja** lõpetab oma töö, ütleb **Kraps**: „Valmis!“

Joonistaja, üks **skript** ellipsi joonistamiseks, käivitub teatega 'joonista'. Kordust täidetakse n korda: arvutatakse nurga ja koordinaatide väärtused ning joonistatakse vastav osa ellipsist. **Joonistaja** skript on sama nagu projektis **ellips_1.sb**. **Kraps** ja **Joonistaja** suhtlevad omavahel teadete abil.

Krapsu skript



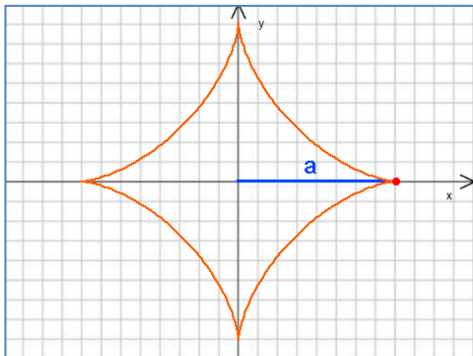
Soovituslikud ülesanded

1. Algandmete ja muutujate väärtuste määramine toimub juhuslike väärtuste abil
2. Andmete ja muutujate väärtuste määramine toimub liugurite abil laval

Projekt astroid.sb ([astroid.sb](#))

Projektis on näidatud astroidi joonistamine objekti **Joonistaja** abil. Koordinaatide x ja y arvutamiseks kasutatakse astroidi punktide **parameetrilised** võrrandid, kus parameetrik on nurk t , mis muutub vahemikus $0^\circ \dots 360^\circ$. Astroidi pooltelg a määrab kujundi suuruse. Näites on muutuja a väärtus 160 , see väärtus võib muutuda vahemikus $10 \dots 160$.

Astroidi punktide koordinaadid arvutatakse võrranditega: $x = a \cos^3(t)$, $y = a \sin^3(t)$ $0 \leq t \leq 360^\circ$



Andmed

Muutujad: x , y – astroidi punktide koordinaadid; t – parameeter, muutub $0 \dots 360$ sammuga 1.

Sprait

Joonistaja. Üks **skript** astroidi joonistamiseks, käivitub vajutamisel rohelisele lipule.

Kordust täidetakse 360 korda: arvutatakse nurga t ja koordinaatide (x ja y) väärtused ning joonistatakse vastav osa astroidist.

Joonistaja algoritm

```
protseduur Joonista
  pliiats üleval
  kustuta
  võta a=160
  võta t=0
  võta pliiatsi suuruseks 2
  võta pliiatsi värviks 12
  mine x:a y: 0
  pliiats all
  kordus 360 korda
    võta t=t + 1
    võta y=ro*sin3(t)
    võta x=ro*cos3(t)
    mine x:x y:y
  lõpp kordus
lõpp
```



Soovituslikud ülesanded

1. Algandmete ja muutujate väärtuste määramine toimub juhuslike väärtuste abil
2. Andmete ja muutujate väärtuste määramine toimub liugurite abil laval
3. Andmete ja muutujate väärtuste määramine on antud üle lavale või mõne teise objektile.

Projekt lill.sb ([Lill.sb](#))

Projektis on näidatud õiekujulise kõvera (lille) joonistamist objekti **Joonistaja** abil. Kasutatakse lille võrrandit polaarkoordinaatides $\rho = a \cdot \sin(b \cdot \varphi)$, kus ρ on polaarraadius ja φ – polaarnurk.

Muutuja a väärtuseks on kroonlehe pikkus ja b väärtuseks kroonlehtede arv.

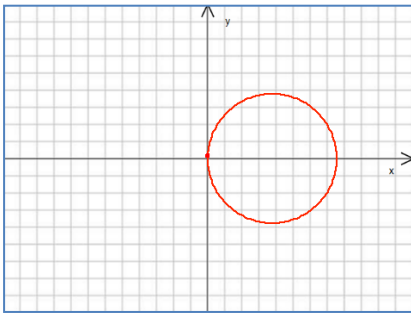
Joone koordinaadid arvutatakse võrranditega: $\rho = a \cdot \sin(b \cdot \varphi)$, $x = \rho \cos(\varphi)$, $y = \rho \sin(\varphi)$

Andmed

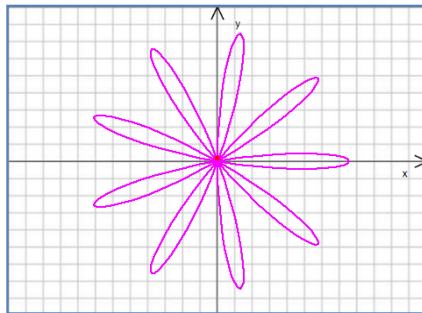
Muutujad: x , y – joone punktide koordinaadid; ro – polaarraadius; fi – polaarnurk (muutub $0^\circ \dots 180^\circ$ sammuga 1°); a – kroonlehe pikkus (võib muutuda 50 ... 160); b – kroonlehtede arv (võib muutuda 1 ... 150).

Kui b on 1, tekib ringjoon. Kui b väärtus on suurem kui 150, muutub kroonlehe laius väga kitsaks.

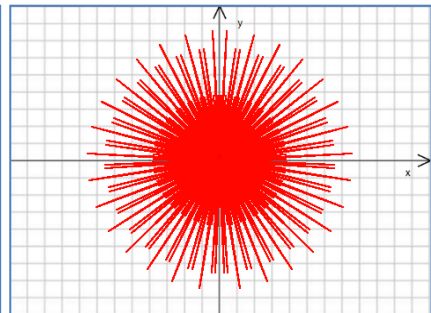
Sprait Joonistaja. Üks skript lille joonistamiseks, käivitub vajutamisel rohelisele lipule.



Pilt 1 Kroonlehtede arv on 1

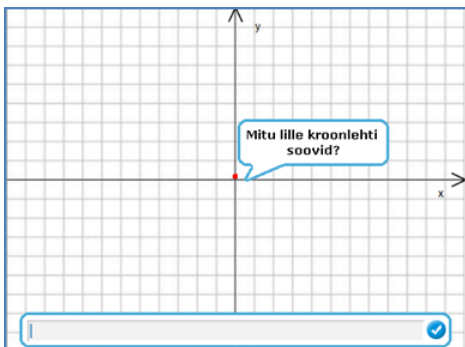
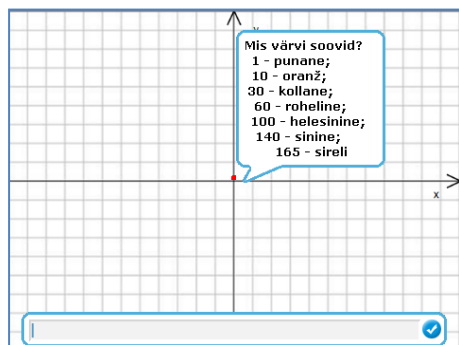


Pilt 2 Kroonlehtede arv on 9



Pilt 3 Kroonlehtede arv on 150

Algul küsitakse kroonlehtede arvu ja lille värvust. Värvu küsimisel pakutakse mõnede värvide väärtused. Kordust täidetakse 180 korda: arvutatakse ro , fi , x ja y väärtused ning joonistatakse vastav osa lillest.



```
kui klõpsatakse
  pliiats üleval
  kustuta
  mine x: 0 y: 0
  küsi Mitu lille kroonlehti soovid? ja oota
  võta b = vastus
  võta a = 150
  võta fi = 0
  küsi Mis värvi soovid? ja oota
  võta pliiatsi värviks vastus
  võta pliiatsi suuruseks 2
  mine x: 0 y: 0
  pliiats all
  korda 180
    muuda fi 1 võrra
    võta ro = a * sin fi * b
    võta x = ro * sin fi
    võta y = ro * cos fi
    mine x: x y: y
```

Soovituslikud ülesanded

1. Algandmete ja muutujate väärtuste määramine toimub kindlates piirides juhuslikult
2. Andmete ja muutujate väärtuste määramine toimub liugurite abil laval
3. Andmete ja muutujate väärtuste määramine on antud üle lavale või mõne teise objektile.

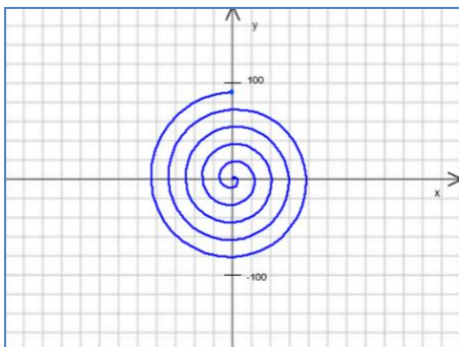
Projekt a_spiraal.sb ([a_spiraal.sb](#))

Projektis on näidatud *Arhimeedese spiraali* joonistamist objekti **Joonistaja** abil. Kasutatakse spiraali võrrandit polaarkoordinaatides $\rho = a * \varphi$, kus ρ on polaarraadius ja φ polaarnurk. Selleks et spiraal mahuks ilusti lavale, kasutatakse mastabitegurit m . Muutuja k määrab keerdude arvu, muutuja a väärtus on k/m ja see peaks olema väiksem kui $1/20$. Muutuja a väärtusest sõltub spiraali keerdude vaheline kaugus.

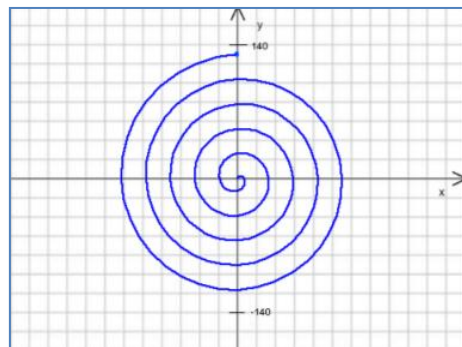
Joone koordinaadid arvutatakse järgmiste võrrandite abil: $\rho = a * \varphi$, $x = \rho \cos(\varphi)$, $y = \rho \sin(\varphi)$

Andmed

Muutujad: x , y – spiraali punktide koordinaadid; ro – polaarraadius; fi – polaarnurk (muutub $0 \dots n$ sammuga h); n – korduste arv, sõltub k väärtusest; h – nurga muutumise samm, arvutatakse $360 * k/n$. k – keerdude arv; mida suurem k , seda suurem peab olema mastabitegur m ; a – keerdudevaheline kaugus, leitakse k/m , võiks olla väiksem kui $0,05$.



Pilt 1 $k = 5$, $m = 100$, $a = 0,05$



Pilt 2 $k = 5$, $m = 70$, $a = 5/70$

Spraidid

Lava, üks **skript** algandmete ja muutujate väärtuste määramiseks. Pärast väärtuste omistamist ja arvutamist annab tekitatakse teade 'joonista' **Joonistajale**.

Joonistaja, üks **skript** spiraali joonistamiseks, käivitub teatega 'joonista'. Kordust täidetakse n korda: arvutatakse nurga ja koordinaatide väärtused ning joonistatakse vastav osa spiraalist.

Lava skript:



Joonistaja skript:



