

Цртање правилних облика уз помоћ

петљи

Многи цртежи су такви у којима се одређени облици понављају по неком утврђеном принципу. Такви програми се реализују уз помоћ петљи. У овом делу ћемо се упознати са тим начином цртања.

Концентрични кругови

Напиши програм који црта концентричне кругове чији је центар у центру екрана, а полупречници су редом 10, 20, ..., 100 пиксела. Сваки круг нацртати црвеном линијом дебљином 5 пиксела.

Центар прозора се једноставно израчунава тако да му је координата x на пола ширине, а координата y на пола висине прозора (обрати пажњу на то да се код кругова морају користити цели бројеви, па стога морамо употребити целобројно дељење). Након тога, можемо навести 10 наредби за цртање кругова.

```
1
2 # bojimo pozadinu prozora u belu
3 prozor.fill(pg.Color("white"))
4 # centar kruga je u centru prozora
5 centar = (sirina // 2, visina // 2)
6 pg.draw.circle(prozor, pg.Color("red"), centar, 10, 5)
7 pg.draw.circle(prozor, pg.Color("red"), centar, 20, 5)
8 pg.draw.circle(prozor, pg.Color("red"), centar, 30, 5)
9 pg.draw.circle(prozor, pg.Color("red"), centar, 40, 5)
10 pg.draw.circle(prozor, pg.Color("red"), centar, 50, 5)
11 pg.draw.circle(prozor, pg.Color("red"), centar, 60, 5)
12 pg.draw.circle(prozor, pg.Color("red"), centar, 70, 5)
13 pg.draw.circle(prozor, pg.Color("red"), centar, 80, 5)
14 pg.draw.circle(prozor, pg.Color("red"), centar, 90, 5)
15 pg.draw.circle(prozor, pg.Color("red"), centar, 100, 5)
16
```



Претходно решење има пуно мана. За почетак, тешко је откуцати оволики број наредби. Даље, ако бисмо се предомислили и променили број кругова или величине полупречника, програм би захтевао доста компликоване измене. Задатак се много боље и једноставније решава када се примети да се полупречници кругова редом мењају од 10 до 100 са кораком 10 (10, 20, 30, ..., 90, 100) и када се подсетимо да у Пајтону такво набрајање може остварити петљом:

```
1 for i in range(pocetak, kraj, korak):
2     ...
3
```

Пошто десни крај није укључен у набрајање (узимају се вредности из полуотвореног интервала [pocetak,kraj]), да би последњи полупречник био 100, за крај је потребно навести вредност 101.

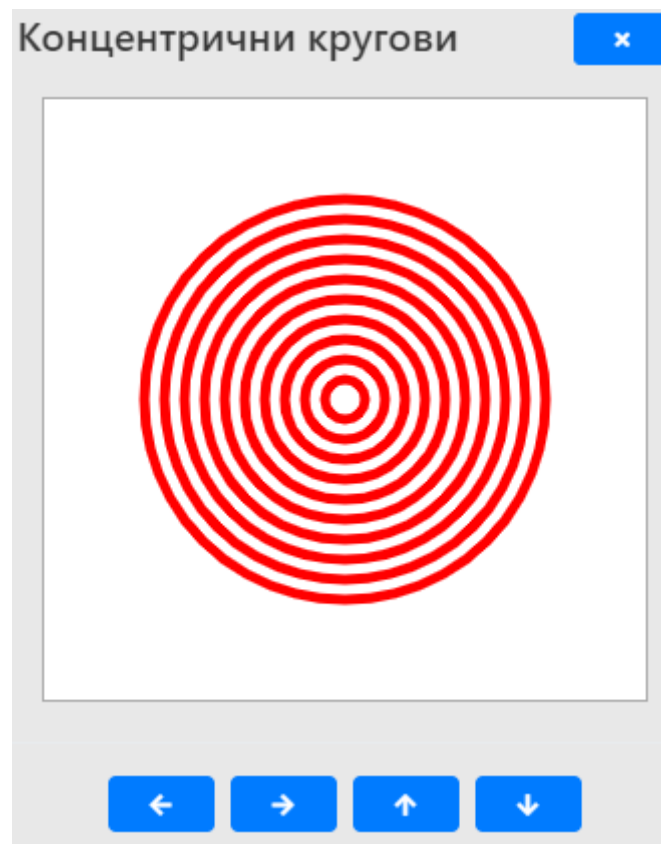
На основу претходне дискусије допуни наредни програм.

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (300, 300) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Концентрични кругови")
6
7 # bojimo pozadinu prozora u belo
8 prozor.fill(pg.Color("white"))
9
10 # centar kruga je u centru prozora - obrati pažnju na tip podataka
11 centar = (???, ???)
12
13 # poluprečnik se menja od 10 do 100, sa korakom 10
14 for r in range(10, ???, 10):
15     # crtamo krug
16     ???
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()
20
```

Једно од могућих решења имало би облик:

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (300, 300) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Концентрични кругови")
6
7 # bojimo pozadinu prozora u belo
8 prozor.fill(pg.Color("white"))
9
10 # centar kruga je u centru prozora - obrati pažnju na tip podataka
11 centar = (150, 150)
12
13 # poluprečnik se menja od 10 do 100, sa korakom 10
14 for r in range(10, 101, 10):
15     pg.draw.circle(prozor, pg.Color("red"), centar, r, 5)
16
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()
20
```

А након покретања програм би нам исцртао следећу слику:

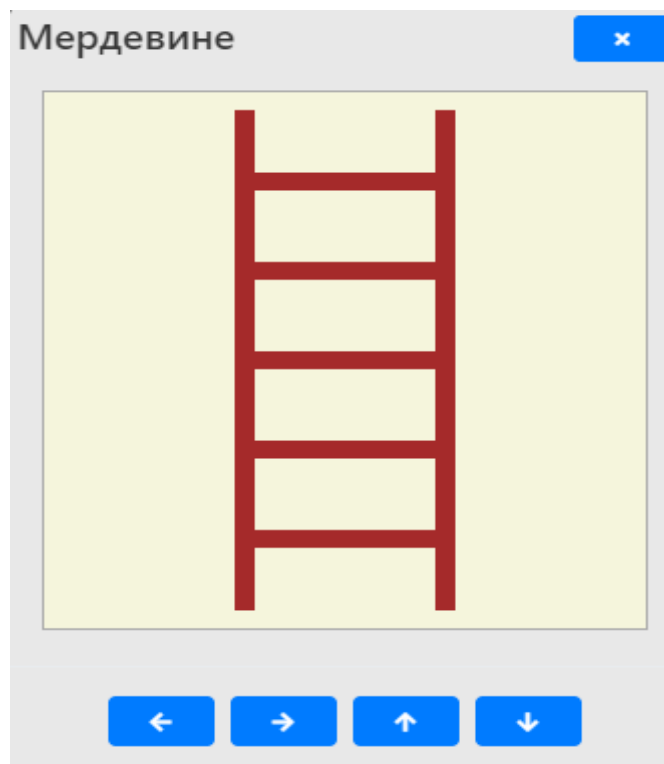


Мердевине

Измените наредни програм тако да се пречаге мердевина цртају у петљи.

Пажљиво проучи које се вредности мењају кроз позиве функције `pg.draw.line`, шта је почетна вредност, шта је крајња вредност и који је корак, па на основу тога позови функцију `range` у оквиру петље `for`.

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (300, 300) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Мердевине")
6
7 prozor.fill(pg.Color("beige")) # bojimo pozadinu ekrana u bež
8
9 # leva strana
10 pg.draw.line(prozor, pg.Color("brown"), (100, 10), (100, visina - 10), 10)
11 # desna strana
12 pg.draw.line(prozor, pg.Color("brown"), (200, 10), (200, visina - 10), 10)
13
14 # ovaj deo prepraviti
15 pg.draw.line(prozor, pg.Color("brown"), (100, 50), (200, 50), 10) # prečaga
16 pg.draw.line(prozor, pg.Color("brown"), (100, 100), (200, 100), 10) # prečaga
17 pg.draw.line(prozor, pg.Color("brown"), (100, 150), (200, 150), 10) # prečaga
18 pg.draw.line(prozor, pg.Color("brown"), (100, 200), (200, 200), 10) # prečaga
19 pg.draw.line(prozor, pg.Color("brown"), (100, 250), (200, 250), 10) # prečaga
20
21 # prikazujemo prozor i čekamo da ga korisnik isključi
22 pygamebg.wait_loop()
```



Решење:

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (300, 300) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Мердевине")
6
7
8 prozor.fill(pg.Color("beige")) # bojimo pozadinu ekrana u bež
9
10 pg.draw.line(prozor, pg.Color("brown"), (100, 10), (100, visina - 10), 10) # leva strana
11 pg.draw.line(prozor, pg.Color("brown"), (200, 10), (200, visina - 10), 10) # desna strana
12
13 for y in range(50, 251, 50):
14     pg.draw.line(prozor, pg.Color("brown"), (100, y), (200, y), 10) # precaga
15
16
17 # prikazujemo prozor i čekamo da ga korisnik isključi
18 pygamebg.wait_loop()
19
```

Правилно распоређени бројеви

У оба претходна примера било је потребно да набројимо неки низ правилно распоређених бројева. У задатку са круговима то су били бројеви 10, 20, ..., 100, а у задатку са мердевинама то су били бројеви 50, 100, 150, 200, 250.

У решењима задатка видели смо да је један начин да се то уради петља облика

```
1 for x in range(x0, xMax+1, dx):
2     ...
3
```

при чему је било потребно обратити пажњу на то да десни крај није укључен у набрајање (узимају се вредности из полуотвореног интервала $[x_0, x_{\text{Max}}+1) = [x_0, x_{\text{Max}}]$).

Да би боље разумео упарићемо за тебе следеће низ бројева са њему одговарајућом петљом:

5, 10, 15, 20, 25, 30

А ————— 1

```
for i in range(5, 30+1, 5)
```

100, 200, 300, 400, 500, 600

Б ————— 2

```
for i in range(100, 600+1, 100)
```

100, 350, 600

Ц ————— 3

```
for i in range(100, 600+1, 250)
```

15, 30, 45, 60, 75

Д ————— 4

```
for i in range(15, 75+1, 15)
```

Још један начин је да променљиву x ажурирамо кроз сваки корак петље, тако што је увећавамо за dx .

```
1 x = x0
2 for i in range(n):
3     ...
4     x += dx
5
```

Видећемо да се велики број задатака са цртањем правилно распоређених облика може решити применом оваквих петљи.

Нагласимо још и да функција `range` са кораком (са три аргумента) прима обавезно целобројне аргументе, па у ситуацијама када корак није целобројан њено коришћење није могуће.

Хоризонтално и вертикално распоређивање облика

Често у применама имамо потребу да распоредимо објекте тако да буду један до другог, тако да су сви објекти равномерно распоређени, тј. тако да су свака два узастопна објекта на истом растојању.

Хоризонтално распоређени кругови

Нацртај 10 кругова пречника 30 пискела тако да буду равномерно распоређени ширином екрана и да се међусобно додирују.

Наредних пар питања ће ти помоћи да решиш овај задатак.

Ако се два круга полупречника r додирују, тада је растојање између њихових центара једнако:

- A. $2 \cdot r$
- B. r
- C. $r / 2$
- D. 100

ПРОВЕРИ

Тачно

Ако круг полупречника r додирује леву ивицу екрана, тада је x координата његовог центра једнака:

ПРОВЕРИ

Тачно!

На основу овога, допуни наредни програм:

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (600, 100) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Хоризонтално распоређени кругови")
6
7 # bojimo pozadinu prozora u belo
8 prozor.fill(pg.Color("white"))
9
10 # crtamo 10 krugova
11 r = 30 # poluprečnik krugova
12 x = ??? # x koordinata centra kruga
13 for i in range(10):
14     # crtamo krug
15     pg.draw.circle(prozor, pg.Color("black"), (x, visina // 2), r, 1)
16     x += ??? # ažuriramo x tako da postane koordinata centra narednog kruga
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()
20
```

Једно од могућих решења је:

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (600, 100) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Хоризонтално распоређени кругови")
6
7 # bojimo pozadinu prozora u belo
8 prozor.fill(pg.Color("white"))
9
10 # crtamo 10 krugova
11 r = 30 # poluprečnik krugova
12 x = 30 # x koordinata centra kruga
13 for i in range(10):
14     # crtamo krug
15     pg.draw.circle(prozor, pg.Color("black"), (x, visina // 2), r, 1)
16     x += 60 # ažuriramo x tako da postane koordinata centra narednog kruga
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()

```

Хоризонтално распоређени кругови



Вертикално распоређени кругови

Прикажимо сада како можемо распоредити кругове вертикално.

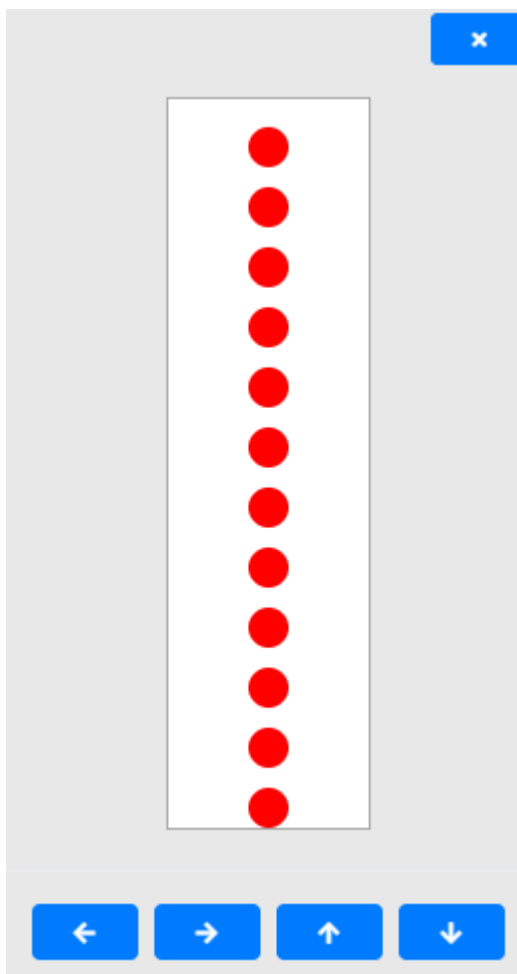
Напиши програм који црта кругове полупречника 10 пиксела равномерно распоређене вертикално средином екрана, тако да су им центри удаљени 50 пиксела (нацртај све кругове који се виде). Висина екрана се мења приликом сваког покретања програма.

Овај задатак је сличан претходном, уз неколико важних разлика. То што су кругови распоређени вертикално уместо хоризонтално не мења пуно - само је потребно заменити улогу x и y координата. Растојање између полупречника је овај пут фиксно (износи 30 пиксела) и не израчунава се на основу полупречника. Кључна разлика је то што број кругова није унапред задат већ је кругове потребно цртати све док се бар неки њихов делић види на екрану. Зато имамо две могућности. Или ћемо некако на основу висине екрана израчунати број кругова који се виде или ћемо уместо бројачке петље `for` употребити условну петљу `while`. Ово друго може бити једноставније.

```
1 import random
2 import pygame as pg
3 import pygamebg
4
5 (sirina, visina) = (100, random.randint(150, 500)) # otvaramo prozor
6 prozor = pygamebg.open_window(sirina, visina, "")
7
8 # bojimo pozadinu prozora u belo
9 prozor.fill(pg.Color("white"))
10
11 r = 10 # poluprečnik krugova
12 dy = 30 # vertikalni razmak između centara dva uzastopna kruga
13 y = ??? # y koordinata centra tekućeg kruga
14 while ???:
15     pg.draw.circle(prozor, pg.Color("red"), (sirina // 2, y), r) # crtamo krug
16     y += ??? # centar narednog kruga je udaljen za dy od centra tekućeg kruga
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()
20
```


Једно од могућих решења је:

```
1 import random
2 import pygame as pg
3 import pygamebg
4
5 (sirina, visina) = (100, random.randint(150, 500)) # otvaramo prozor
6 prozor = pygamebg.open_window(sirina, visina, "")
7
8 # bojimo pozadinu prozora u belo
9 prozor.fill(pg.Color("white"))
10
11 r = 10 # poluprečnik krugova
12 dy = 30 # vertikalni razmak između centara dva uzastopna kruga
13 y = visina-r # y koordinata centra tekućeg kruga
14 while y>=0:
15     pg.draw.circle(prozor, pg.Color("red"), (sirina // 2, y), r) # crtamo krug
16     y += -dy # centar narednog kruga je udaljen za dy od centra tekućeg kruga
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()
```



Штала

Још је лакше ако распоређујемо готове слике учитане из `png` или `jpg` датотека.

У штала живи пет кравица. Напиши програм који их распоређује тако да стоје једна поред друге, на дну прозора. Сliku кравице учитај из датотеке `kravica.png`.

Пре него што решиш задатак, подсети се неколико основних функција у вези са сликама.

Повежи наредбе са оним чему служе.

<code>slika.get_height()</code>	А	1	приказивање слике на екрану
<code>prozor.blit(slika, (x, y))</code>	Б	2	очитавање висине слике
<code>slika = pg.image.load("slika.png")</code>	Ц	3	очитавање ширине слике
<code>slika.get_width()</code>	Д	4	учитавање слике из датотеке

ПРОВЕРИ

Тачан одговор!

Пре слике са редним бројем `i` налази се тачно `i` слика, па се `x` координата леве ивице те слике може добити множењем ширине слике са `i`. Пошто слика треба да буде на дну прозора, у координату њеног горњег левог темена можемо израчунати тако што од висине прозора одузмемо висину слике.

Допуни наредни програм на основу претходне дискусије.



```

1 import random
2 import pygame as pg
3 import pygamebg
4
5 (sirina, visina) = (750, 300) # otvaramo prozor
6 prozor = pygamebg.open_window(sirina, visina, "Кравице")
7
8 # bojimo pozadinu u belo
9 prozor.fill(pg.Color("white"))
10 # učitavamo sliku
11 slika = ???
12 # očitavamo dimenzije slike
13 (sirina_slike, visina_slike) = (slika.get_width(), slika.get_height())
14 # pet puta iscrtavamo sliku
15 for i in range(5):
16     prozor.blit(???)
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()

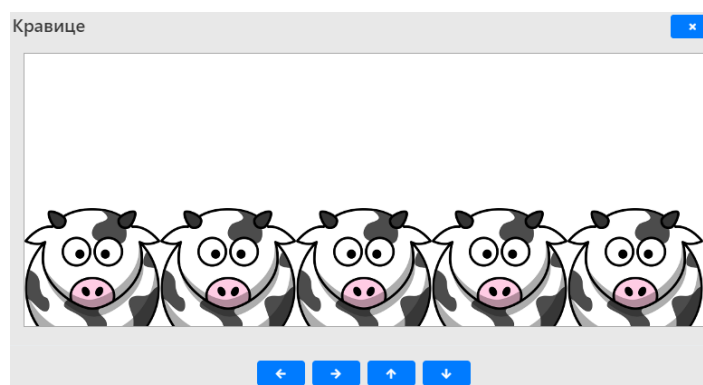
```

Решење:

```

1 import random
2 import pygame as pg
3 import pygamebg
4
5 (sirina, visina) = (750, 300) # otvaramo prozor
6 prozor = pygamebg.open_window(sirina, visina, "Кравице")
7
8 # bojimo pozadinu u belo
9 prozor.fill(pg.Color("white"))
10 # učitavamo sliku
11 slika = pg.image.load("kravica.png")
12 # očitavamo dimenzije slike
13 (sirina_slike, visina_slike) = (slika.get_width(), slika.get_height())
14 # pet puta iscrtavamo sliku
15 for x in range(0, sirina, sirina_slike):
16     prozor.blit(slika, (x, visina_slike))
17
18 # prikazujemo prozor i čekamo da ga korisnik isključi
19 pygamebg.wait_loop()
20

```



Правоугаона мрежа

Напиши програм који исцртава правоугаону мрежу која се састоји од 100 правоугаоних поља, распоређених у 10 врста и 10 колона (исцртати само линије мреже и то хоризонталне линије плавом бојом, а вертикалне црвеном, дебљине 5 пиксела).

Основни задатак је одредити координате x вертикалних линија и координате y хоризонталних линија. Ширину једног правоугаоника можемо одредити дељењем ширине прозора бројем колона (у нашем случају то је 10), док висину једног правоугаоника можемо одредити дељењем висине прозора бројем врста (то је поново 10). Означимо те димензије са dx и dy . Вертикалне линије се онда налазе на растојању $dx, 2dx, 3dx, \dots, 9dx$ пиксела од леве ивице прозора (то су им координате x). Пошто се те линије простиру од врха до дна прозора, координате y су им једнаке нули, односно висини прозора. Понављање цртања линија остварујемо, наравно, употребом петље `for`, при чему је најбоље да се бројач `i` креће од један до девет, јер се тада у кораку `i` црта линија од тачке `(i*dx, 0)` до тачке `(i*dx, visina)`. Цртање хоризонталних линија остварујемо веома слично, у независној петљи `for` у којој се црта линија од тачке `(0, i*dy)` до тачке `(sirina, i*dy)`.

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (400, 300) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Правоугаона мрежа")
6
7 # bojimo pozadinu prozora u belo
8 prozor.fill(pg.Color("white"))
9
10 brojPodeoka = 10
11 dx = sirina / brojPodeoka
12 dy = ??? # izračunaj razmak između podeoka po visini
13
14 # crtamo horizontalne linije
15 for i in range(1, brojPodeoka):
16     pg.draw.line(prozor, pg.Color("blue"), (0, i*dy), (sirina, i*dy), 5)
17
18 # dodaj kod koji crta vertikalne linije crvenom bojom
19 ???
20
21 # prikazujemo prozor i čekamo da ga korisnik isključi
22 pygamebg.wait_loop()
```

Решење:

```
1 import pygame as pg
2 import pygamebg
3
4 (sirina, visina) = (400, 300) # otvaramo prozor
5 prozor = pygamebg.open_window(sirina, visina, "Правоугаона мрежа")
6
7 # bojimo pozadinu prozora u belo
8 prozor.fill(pg.Color("white"))
9
10 brojPodeoka = 10
11 dx = sirina / brojPodeoka
12 dy = visina / brojPodeoka          # izračunaj razmak između podeoka po visini
13
14 # crtamo horizontalne linije
15 for i in range(1, brojPodeoka):
16     pg.draw.line(prozor, pg.Color("blue"), (0, i*dy), (sirina, i*dy), 5)
17
18 # dodaj kod koji crta vertikalne linije crvenom bojom
19 for i in range(1, brojPodeoka):
20     pg.draw.line(prozor, pg.Color("red"), (i*dx, 0), (i*dx, visina), 5)
21
22 # prikazujemo prozor i čekamo da ga korisnik isključi
23 pygamebg.wait_loop()
```

