

30. Цртање правилних облика уз помоћ петљи

Многи цртежи су такви да се у њима одређени облици понављају по неком утврђеном принципу. Такви програми се реализују уз помоћ петљи.

Пример 1 - приказ концентричних кругова - слика мете

Програм се састоји од концентричних кругова чији центар је у центру екрана, са **полупречницима** редом 10, 40, 70, 100, 130, 160 пиксела.

Кругове исцртавамо црном линијом **дебљине** 5 пиксела, на белој површини **прозора величине 350 x 350 пиксела**.

Центар прозора једноставно се израчунава тако да му је координата x на пола ширине, а координата y на пола висине прозора.

Наредба фор **петље са 3 аргмента** у Пајтону:

```
for i in range(pocetak, kraj+1, korak):
```

да би последњи полупречник био 160, за крај је потребно навести вредност 161.

30. Цртање правилних облика уз помоћ петљи 7

primer 1 koncentricni krugovi.py - H:\My Drive\OШ 20. ОКТОБАР\7\II\11 crtanje pravilnih oblika\primer 1 koncentricni kr

File Edit Format Run Options Window Help

```
import pygame as pg
pg.init()
pg.display.set_caption("meta")

(sirina, visina) = (350, 350)
prozor = pg.display.set_mode((sirina, visina))
prozor.fill(pg.Color("white"))

#centar kruga je u centru prozora
centar =(sirina//2, visina //2)

#poluprečnik se menja od 10 do 160, sa korakom 30
for r in range (10, 161, 30):
    pg.draw.circle (prozor, pg.Color("black"), centar, r, 5)

#crtamo krug
pg.display.update()

pg.display.update()
while pg.event.wait().type != pg.QUIT:
    pass
pg.quit()
```



Пример 2 Исцртавање правоугаоне мреже

Мрежа се састоји од 25 правоугаоних поља, распоређених у 5 врста и 5 колона (исцртати само линије мреже, и то хоризонталне линије црвеном бојом, а вертикалне црном, дебљине 3 пиксела).

Основни задатак је одредити **координате x вертикалних линија** и **координате y хоризонталних линија**.

Ширина једног правоугаоника може се одредити **дељењем ширине прозора бројем колона** (у нашем случају то је 5), док се **висина једног правоугаоника** може одредити **дељењем висине прозора бројем врста** (то је поново 5).

Означе се те димензије са **dx** и **dy** . Вертикалне линије се онда налазе на растојању **dx** , **$2dx$** , **$3dx$** , **$4dx$** и **Sdx** пиксела од леве ивице прозора (то су им координатех). Пошто се те линије простиру од врха до дна прозора, координатеу су им једнаке 0, односно висини прозора.

Понављање цртања линија остварује се употребом петље **for** , при чему је најбоље да се бројач **i** креће од 1 до 4, јер се тада у кораку **i** црта линија од тачке (**$i*dx$** , 0) до тачке (**$1*^x$** , **$visina$**).

Цртање хоризонталних линија остварује се веома слично, у независној петљи **for** у којој се црта линија од тачке(0, **$i*dy$**) до тачке (**$sirina$** , **$i*dy$**).

30. Цртање правилних облика уз помоћ петљи 7

```
primer 2 pravougaona mreza.py - H:\My Drive\ОШ 20. ОКТОБАР\7\II\11 crtanje pravilnih oblika\primer 2 pravougaona mreza.py (3.11.1)
File Edit Format Run Options Window Help


import pygame as pg
pg.init()
pg.display.set_caption("pravougaona mreza")
(sirina, visina) = (400, 300)
prozor = pg.display.set_mode((sirina, visina))
prozor.fill(pg.Color("white"))

brojPodeoka = 5
dx = sirina / brojPodeoka
dy = visina / brojPodeoka

# crtamo horizontalne crvene linije
for i in range(1, brojPodeoka):
    pg.draw.line(prozor, pg.Color("red"), (0, i*dy), (sirina, i*dy), 3)

# dodaj kod koji crta vertikalne linije crvnom bojom
for i in range(1, brojPodeoka):
    pg.draw.line(prozor, pg.Color("black"), (i*dx, 0), (i*dx, visina), 3)

pg.display.update()
while pg.event.wait().type != pg.QUIT:
    pass
pg.quit()
```



Пример 3 програм којим се шрафира дијагоналама по прозору у зеленој боји на црној позадини,

Број линија изнад најдужедијагонале (укључујући и њу) јесте $l = 5$ (исто важи и за број линија испод најдуже дијагонале). Прозор је димензија 400 x 300 пиксела.

Напомена: Крајње тачке ових дужи деле сваку ивицу прозора на n једнаких делова. Стога се растојање dx између суседних тачака на горњој (и доњој) ивици прозора може израчунати дељењем ширине, а растојање dy између суседних тачака на левој (и десној) ивици дељењем висине прозора бројем n . Посматрају се дужи које спајају леву и горњу ивицу прозора (последња таква је најдужа дијагонала). Прва дуж спаја тачке са координатама $(dx, 0)$ и $(0, dy)$, Друга дуж тачке са координатама $(2dx, 0)$ и $(0, 2dy)$ итд. Дакле, те дужи могу се нацртати тако што се у петљи у којој бројачка променљива i мења вредности од 1 до n цртају дужи које спајају тачке са координатама $(i \cdot dx, 0)$ и $(0, i \cdot dy)$. Слично, прва дуж испод дијагонале спаја тачке са координатама (w, dy) и (dx, h) , друга тачка са координатама $(w, 2dy)$ и $(2dx, h)$ итд., где w означава ширину, а h висину екрана. Дакле, те дужи могу се нацртати тако што се у петљи у којој бројачка променљива i мења вредност од 1 до $l - 1$ црта дуж која спаја тачке са координатама $(sirina, i \cdot dy)$ и $(i \cdot dx, visina)$.

```

primer 3 dijagonale.py - H:\My Drive\ОШ 20. ОКТОБАР\7\11\11 crtanje pravilnih oblika\primer 3 dijagonale.py (3.11.1)
File Edit Format Run Options Window Help

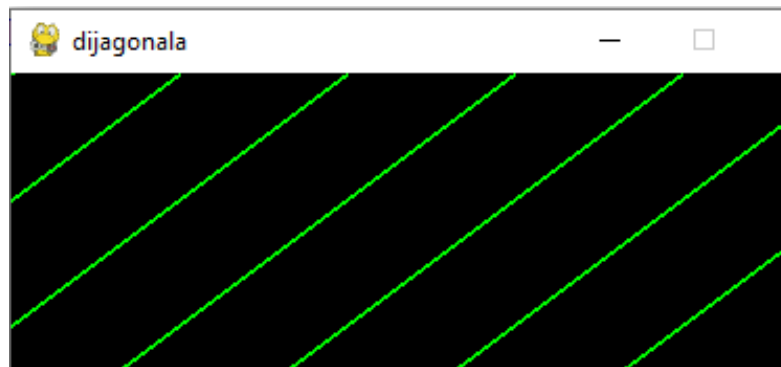
import pygame as pg
pg.init()
pg.display.set_caption("dijagonala")
(sirina, visina) = (400, 300)
prozor = pg.display.set_mode((sirina, visina))
prozor.fill(pg.Color("black"))

n = 5
dx = sirina / n
dy = visina / n
# crtamo n linija iznad dijagonale i nju
for i in range(n+1):
    pg.draw.line(prozor, pg.Color("green"), (i*dx,0), (0, i*dy), 2)

# dodaj n-1 linija ispod dijagonale, bez nje
for i in range(1,n):
    pg.draw.line(prozor, pg.Color("green"), (i*dx, visina), (sirina, i*dy), 2)

pg.display.update()
while pg.event.wait().type != pg.QUIT:
    pass
pg.quit()

```



2. čas

Бирање рандом/насумичног броја

учитавање библиотеке **рандом**

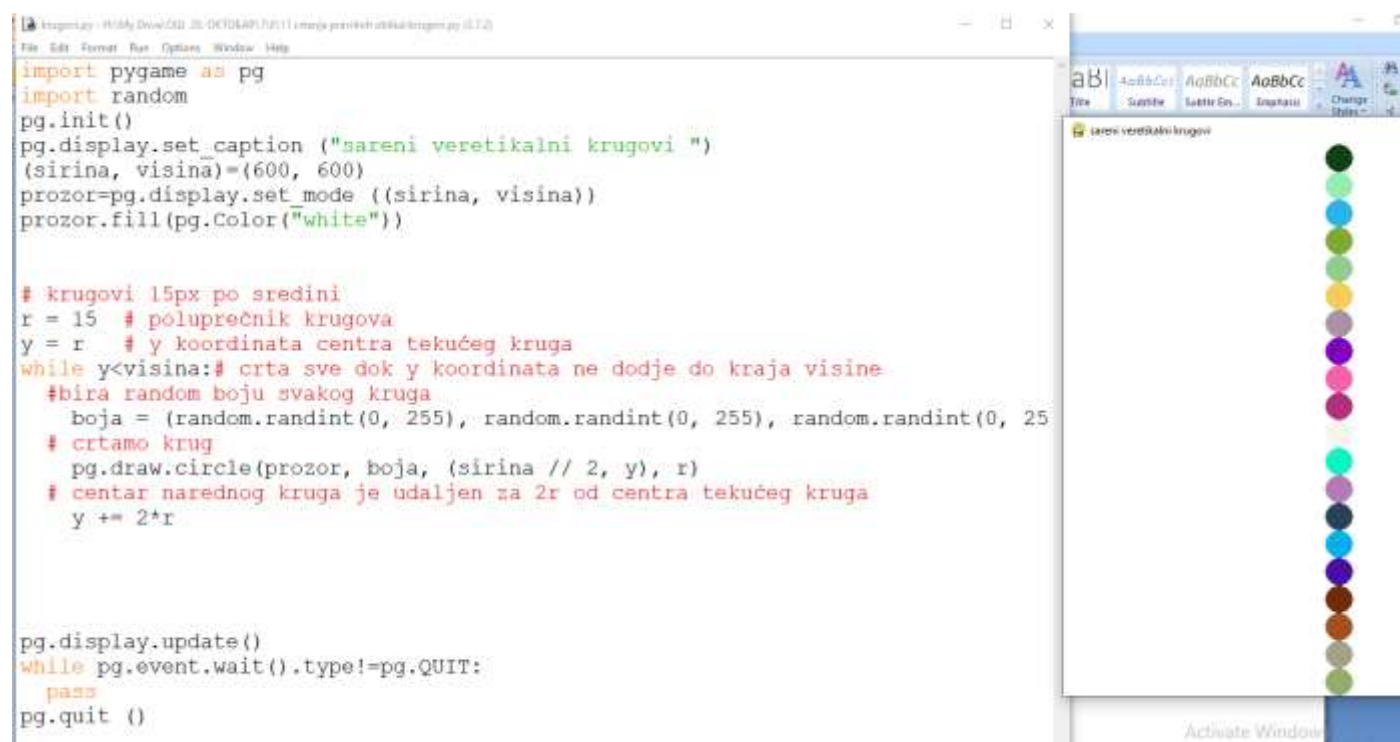
наредба:

random.randint(0, 255)

Y=Y+1

Y=+1

Пример 1. sareni veretikalni krugovi,



пример 2. Шарене хоризонталне линије

```

str191 sarene duzi.py - H:\My Drive\Oli\ 20. OKTOBAR\7\11 crtanje pravilnih oblika\str191 sarene duzi.py (3.7.2)
File Edit Format Run Options Window Help

import pygame
import random
pygame.init()
s = 500
v = 50
prozor = pygame.display.set_mode((s, v))
pygame.display.set_caption("Niz raznobojnih duži sa razmakom")
prozor.fill(pygame.Color("white"))
praznine = 30
sredina = v // 2
x = 0
while x < s:
    boja = (random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    pygame.draw.line(prozor, boja, (x, sredina), (x + praznine, sredina), 5)
    x = x + 2 * praznine
pygame.display.update()
while pygame.event.wait().type != pygame.QUIT:
    pass
pygame.quit()

```



