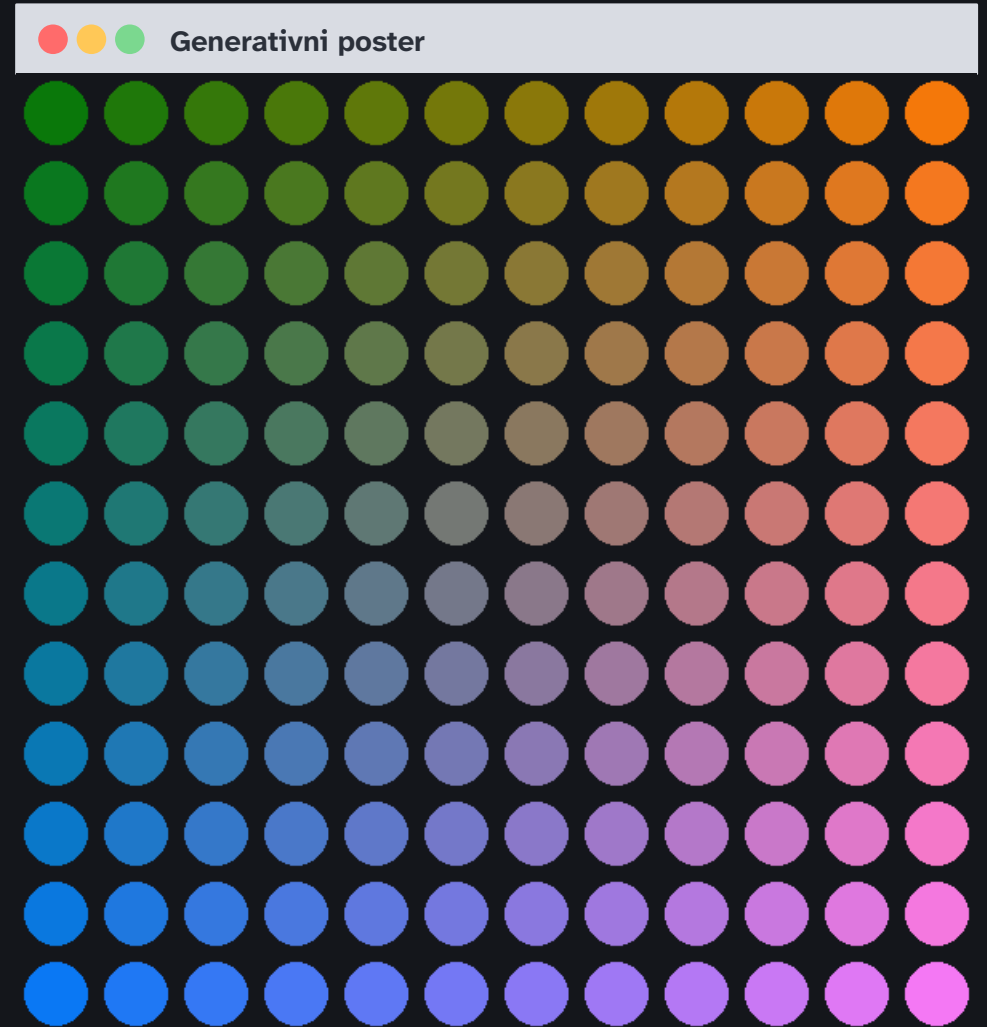


PyGame

Prva programska biblioteka: umesto u konzolu, crtamo u prozor

Danas pravimo: generativni poster, noćno nebo i kolaž od jedne slike



Tri pitanja za zagrevanje

1. prošla tema

```
def dupliraj(x):  
    x * 2  
  
print(dupliraj(4))
```

Ispis: None

2. liste

```
boje = ["siva", "plava"]  
boje.append("žuta")  
print(len(boje), boje[2])
```

Ispis: 3 žuta

3. mešano

```
velicina = (400, 300)  
(s, v) = velicina  
print(s // 3, v / 3)
```

Ispis: 133 100.0

Pet novih reči za danas

biblioteka

gotov kod koji uvozimo i koristimo, a nismo ga sami napisali

pip

program koji preuzima i instalira pakete sa interneta

površina (Surface)

pravougaonik piksela po kome crtamo: prozor, slika ili tekst

blit

lepljenje jedne površine na drugu, na zadato mesto

dogadjaj

poruka programu da se nešto desilo: klik na X, taster, pokret miša

PyGame je biblioteka za crtanje, zvuk i događaje

Standardna biblioteka

Dolazi uz Python, samo je uvezemo:
`import random, import math`

Spoljna biblioteka

Prvo je instaliramo (`pip install pygame-ce`), pa tek onda uvezemo: `import pygame`

MODULI KOJE KORISTIMO

<code>pygame.display</code>	prozor i prikaz
<code>pygame.draw</code>	oblici: krug, pravougaonik, linija
<code>pygame.image</code>	učitavanje i snimanje slika
<code>pygame.font</code>	tekst
<code>pygame.event</code>	događaji (klik na X, tastatura)
<code>pygame.time</code>	čekanje i sat
<code>pygame.transform</code>	rotacija i promena veličine
<code>pygame.mixer.music</code>	muzika
<code>pygame.Rect</code>	pravougaona oblast (klasa)

Instaliramo jednom, a proveravamo uvozom

1. U terminalu (ili u Thonnyju: Tools → Manage packages)

```
$ pip install pygame-ce
```

ako pip nije pronađen: `python -m pip install pygame-ce`

Paket se zove pygame-ce,
a u kodu i dalje pišemo `import pygame`.

2. U programu

```
import pygame  
print(pygame.version.ver)
```

ISPIS (BROJ VERZIJE)

2.6.1

Čudovište: ModuleNotFoundError



Kako izgleda

ModuleNotFoundError: No module named 'pygame' već u redu sa import.

Zašto se pojavljuje

PyGame nije instaliran za Python kojim pokrećemo program, ili je ime pogrešno napisano.

Kako se pobeđuje

`python -m pip install pygame`-ce za isti Python koji koristi editor. Fajl ne nazivamo `pygame.py`.

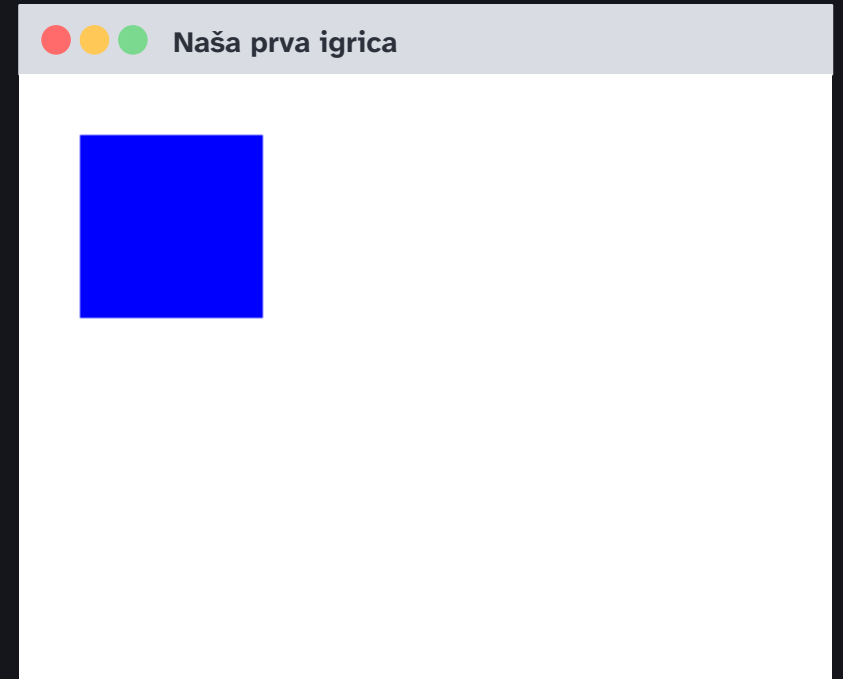
Prvi prozor

Prozor je platno po kome crtamo

Prvi program otvara prozor i crta kvadrat

```
import pygame as pg

pg.init()
ekran = pg.display.set_mode((400, 300))
pg.display.set_caption("Naša prva igrice")
ekran.fill("white")
pg.draw.rect(ekran, "blue", (30, 30, 90, 90))
pg.display.update()
pg.time.wait(3000)
pg.quit()
```



set_mode otvara prozor i vraća površinu

```
import pygame as pg

pg.init()
ekran = pg.display.set_mode((400, 300))
print(ekran.get_size())
print(ekran.get_width(), ekran.get_height())
print(type(ekran))
```

ISPIS

```
(400, 300)
400 300
<class 'pygame.surface.Surface'>
```

pg.init()

pokreće module PyGame-a

set_mode((400, 300))

otvara prozor; vraća površinu

pg.display.update()

prikazuje ono što smo nacrtali

Prozor ostaje otvoren dok korisnik ne klikne na X

```
import pygame as pg

pg.init()
(sirina, visina) = (400, 300)
ekran = pg.display.set_mode((sirina, visina))
pg.display.set_caption("Moj crtež")
ekran.fill("white")

# ovde crtamo

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

Klik na X

operativni sistem šalje programu događaj QUIT

event.wait()

čeka sledeći događaj; dok čeka, prozor reaguje

pg.quit()

gasi PyGame posle izlaska iz petlje

Čudovište: prozor koji ne reaguje



Kako izgleda

Prozor se ne zatvara na X; sistem javlja „ne odgovara“ (Not Responding).

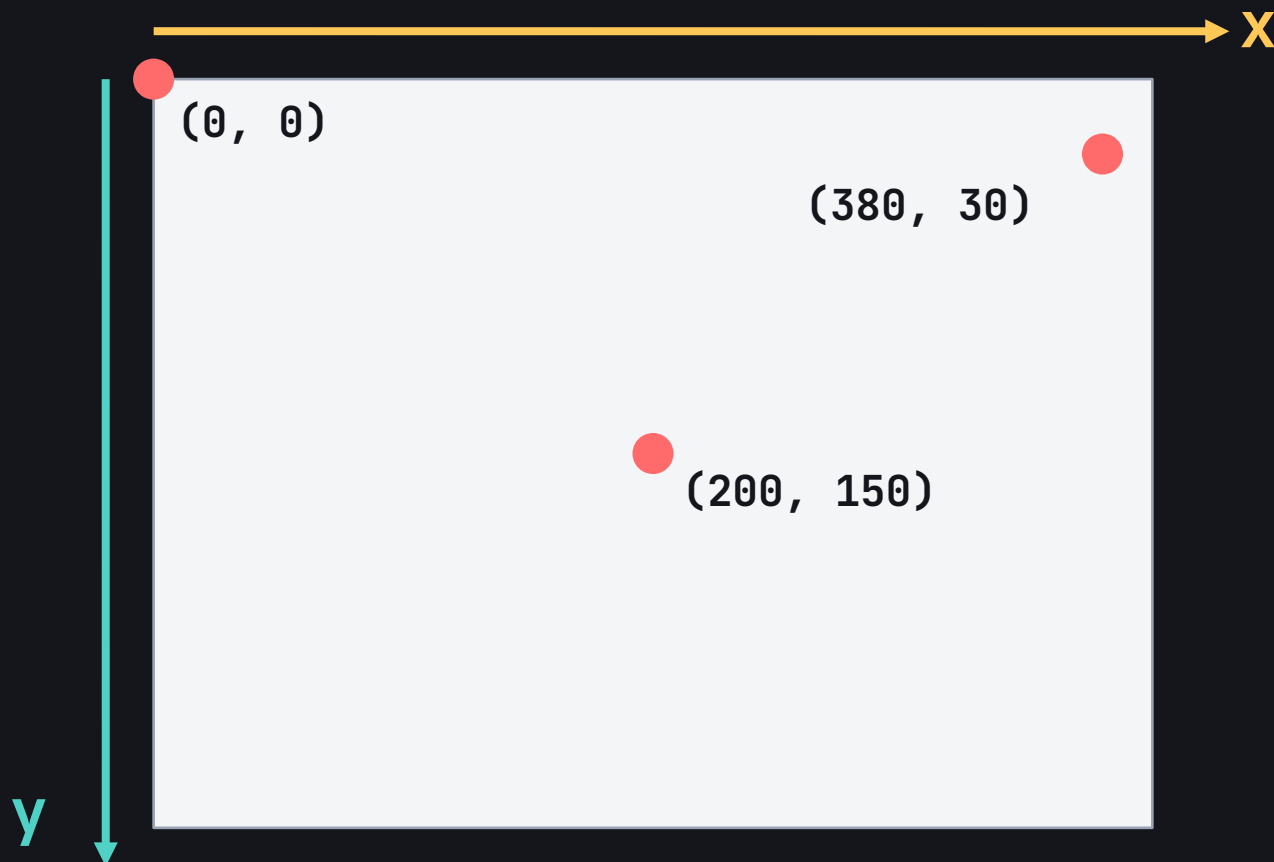
Zašto se pojavljuje

Program ne čita događaje: na primer `while True: pass` ili dugo `pg.time.wait`.

Kako se pobeđuje

Prekid sa `Ctrl+C` ili dugmetom Stop. Zatim dodamo petlju koja čita događaje i čeka `QUIT`.

Početak je gore levo, a y raste nadole



- Jedinica je piksel.
- U prozoru 400 × 300 x ide od 0 do 399, a y od 0 do 299.
- U matematici y raste nagore; na ekranu raste nadole, kao redovi teksta.

Gde se pojavljuje crveni krug?

```
# prozor je 400 x 300  
pg.draw.circle(ekran, "red", (100, 250), 20)
```

Pomoć: početak je u gornjem levom uglu.

A

gore levo

B

tačno

dole levo

C

gore desno

D

dole desno

Boja je ime ili tri broja od 0 do 255

**white**

(255, 255, 255)

**black**

(0, 0, 0)

**red**

(255, 0, 0)

**green**

(0, 255, 0)

**blue**

(0, 0, 255)

**yellow**

(255, 255, 0)

**orange**

(255, 165, 0)

**magenta**

(255, 0, 255)

```
import pygame as pg
```

```
NARANDZASTA = (255, 165, 0)
```

```
print(pg.Color("orange"))
```

```
print(pg.Color("orange") == NARANDZASTA)
```

```
print(pg.Color("#FFA500"))
```

ISPIS

```
(255, 165, 0, 255)
```

```
True
```

```
(255, 165, 0, 255)
```

Koje boje je kvadrat?

```
boja = (255, 255, 0)  
pg.draw.rect(ekran, boja, (0, 0, 50, 50))
```

Svetlost se meša drugačije od boja na paleti.

A

crvene

B

zelene

C

žute

tačno

D

bele

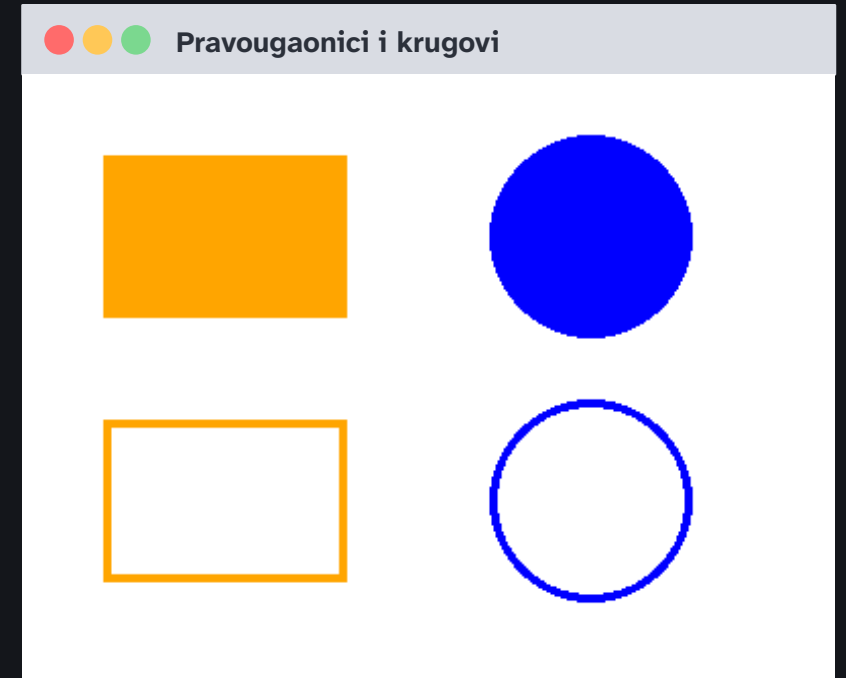
Crtanje oblika

Modul draw: svaka funkcija crta jedan oblik

Bez debljine oblik je popunjen; krug traži poluprečnik

```
# pravougaonik: gornje levo teme, širina, visina  
pg.draw.rect(ekran, "orange", (40, 40, 120, 80))  
pg.draw.rect(ekran, "orange", (40, 170, 120, 80), 4)  
  
# krug: centar i poluprečnik (obavezan)  
pg.draw.circle(ekran, "blue", (280, 80), 50)  
pg.draw.circle(ekran, "blue", (280, 210), 50, 4)
```

- Debljina 4: crta se samo ivica od 4 piksela.
- Bez debljine ili sa 0: oblik je popunjen.

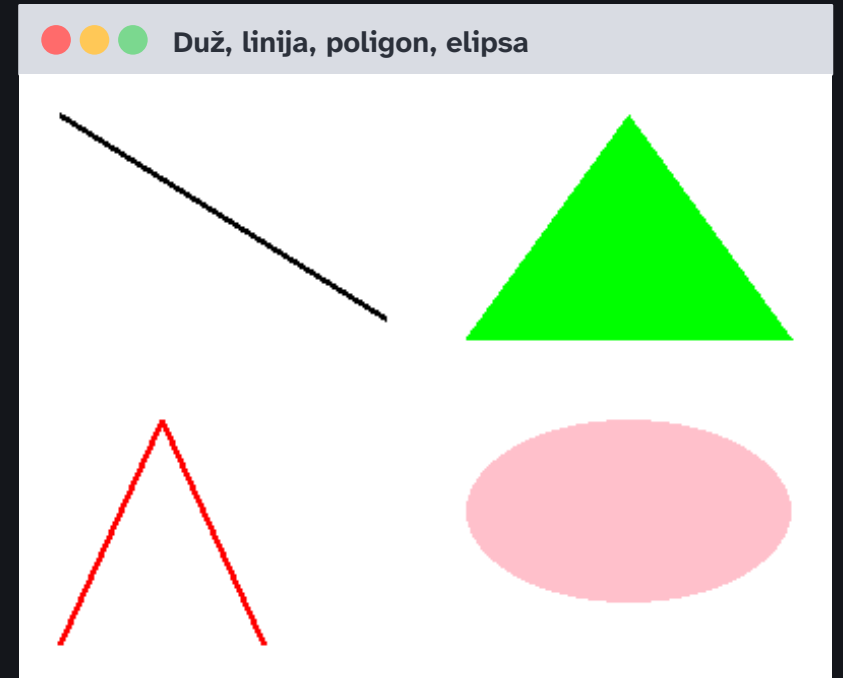


Poligon i izlomljena linija zadaju se listom temena

```
# duž: od tačke do tačke, debljina 3
pg.draw.line(ekran, "black", (20, 20), (180, 120), 3)

# izlomljena linija i poligon: lista temena
cik_cak = [(20, 280), (70, 170), (120, 280)]
pg.draw.lines(ekran, "red", False, cik_cak, 3)
trougao = [(300, 20), (380, 130), (220, 130)]
pg.draw.polygon(ekran, "green", trougao)

# elipsa: upisana u pravougaonik
pg.draw.ellipse(ekran, "pink", (220, 170, 160, 90))
```



- False kod lines: linija nije zatvorena.

Šta se vidi na beloј pozadini?

```
ekran.fill("white")  
c = (200, 150)  
pg.draw.circle(ekran, "red", c, 60, 0)  
pg.draw.circle(ekran, "white", c, 40)
```

Pažljivo: šta znači debljina 0? A šta radi drugi krug?

A

ništa: debljina
0 je nevidljiva
linija

B

tačno

crveni prsten
širok 20
piksela

C

crveni popunjen
krug
poluprečnika 60

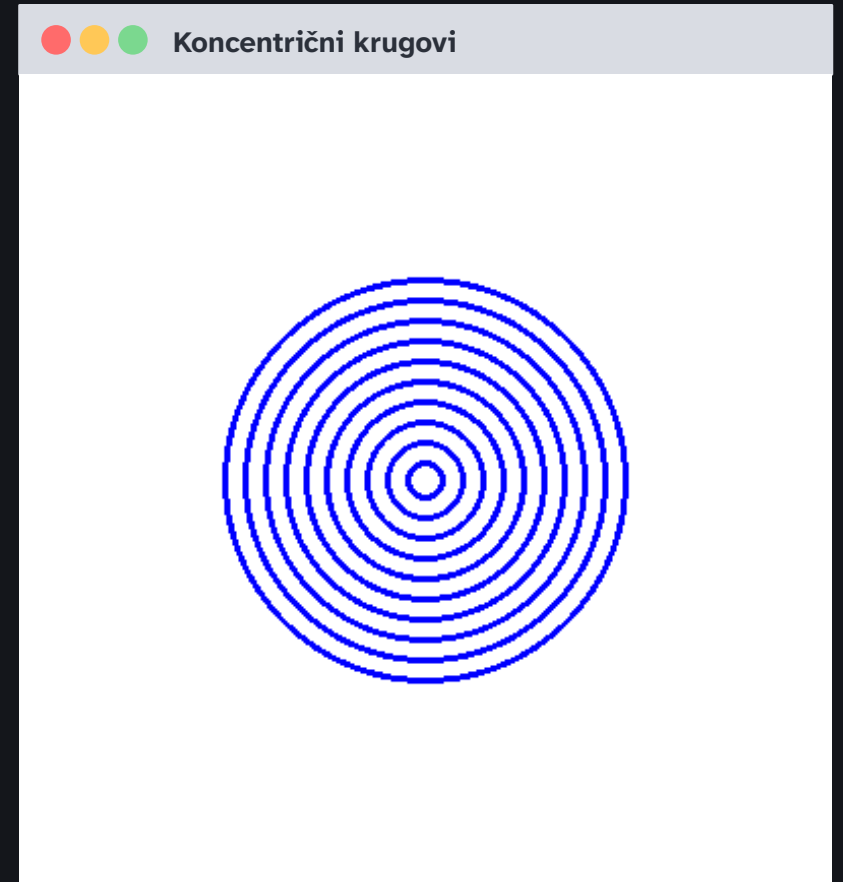
D

tanka crvena
kružnica

Petlja crta deset krugova sa istim centrom

```
# centar svih krugova je u centru prozora  
centar = (sirina // 2, visina // 2)  
  
# r je poluprečnik: 10, 20, ..., 100  
for r in range(10, 101, 10):  
    pg.draw.circle(ekran, "blue", centar, r, 3)
```

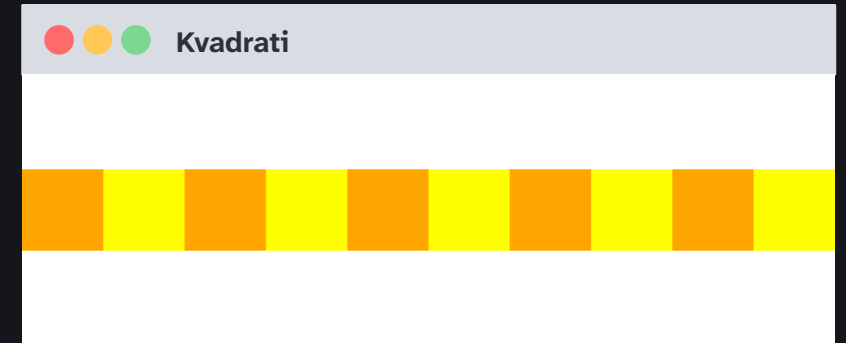
- Gornja granica 101, da bi i 100 bio uključen.
- Isti centar, poluprečnik raste za 10.



Grananje u petlji bira boju svakog kvadrata

```
broj_kvadrata = 10
stranica = sirina // broj_kvadrata
gore = (visina - stranica) // 2

for i in range(broj_kvadrata):
    if i % 2 == 0:
        boja = "orange"
    else:
        boja = "yellow"
    levo = i * stranica
    kvadrat = (levo, gore, stranica, stranica)
    pg.draw.rect(ekran, boja, kvadrat)
```



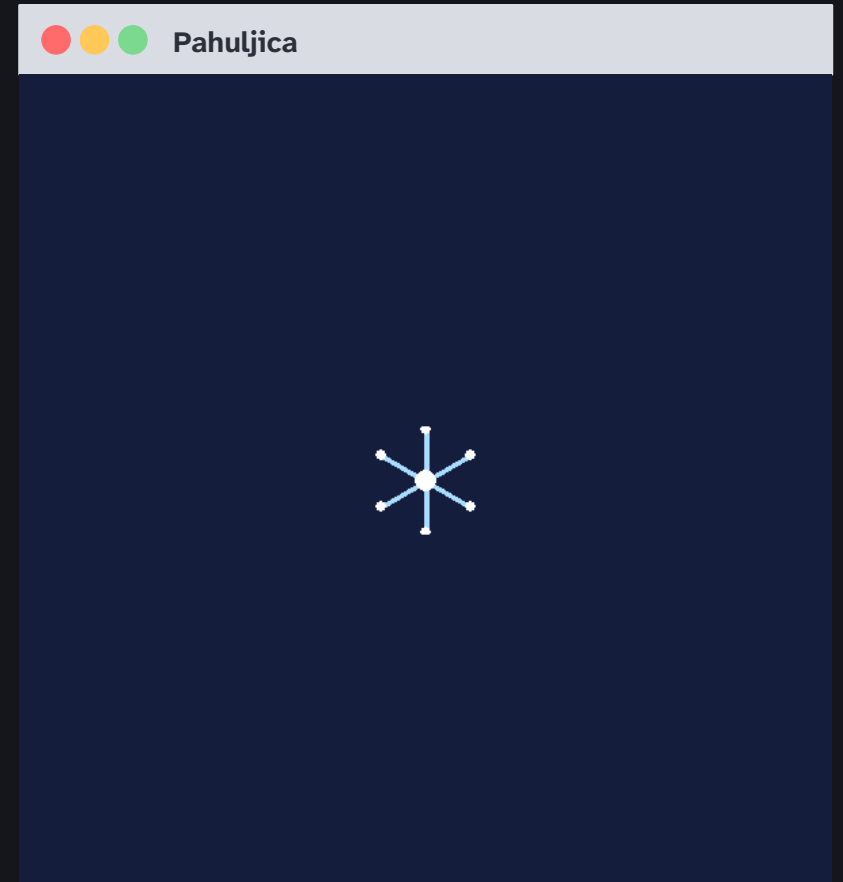
Slike, tekst i slučajnost

Sve što nije oblik, u PyGame-u je sličica

Sliku učitamo jednom, a lepimo je funkcijom blit

```
# učitavamo sličicu iz datoteke pahuljica.png  
slika = pg.image.load("pahuljica.png")  
slika = slika.convert_alpha()  
  
# okvir slike čiji je centar u centru prozora  
centar = (sirina // 2, visina // 2)  
okvir = slika.get_rect(center=centar)  
ekran.blit(slika, okvir)
```

- Fajl sa slikom mora da bude u folderu programa.
- `convert_alpha` čuva providne delove; poziva se posle `set_mode`.



Šta ispisuje ovaj program?

```
import pygame as pg

slika = pg.Surface((100, 60))
okvir = slika.get_rect(center=(200, 150))
print(okvir.topleft)
```

Slika je široka 100, a visoka 60 piksela.

A

(200, 150)

B

tačno

(150, 120)

C

(100, 90)

D

(250, 180)

Tekst je sličica koju pravi font

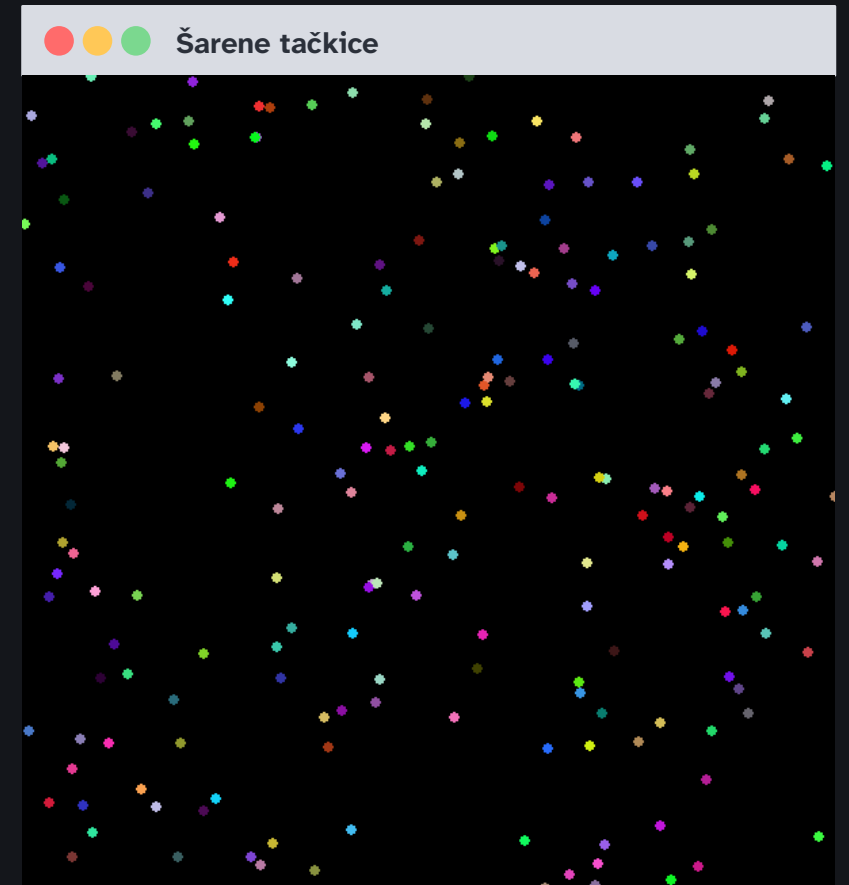
```
# None znači ugrađeni font; 64 je veličina  
font = pg.font.Font(None, 64)  
poruka = "Zdravo svete!"  
natpis = font.render(poruka, True, "magenta")  
  
centar = (sirina // 2, visina // 2)  
okvir = natpis.get_rect(center=centar)  
ekran.blit(natpis, okvir)
```

- render: tekst, glatke ivice (True), boja.
- Natpis centriramo isto kao sliku.



random.randint bira mesto i boju svake tačkice

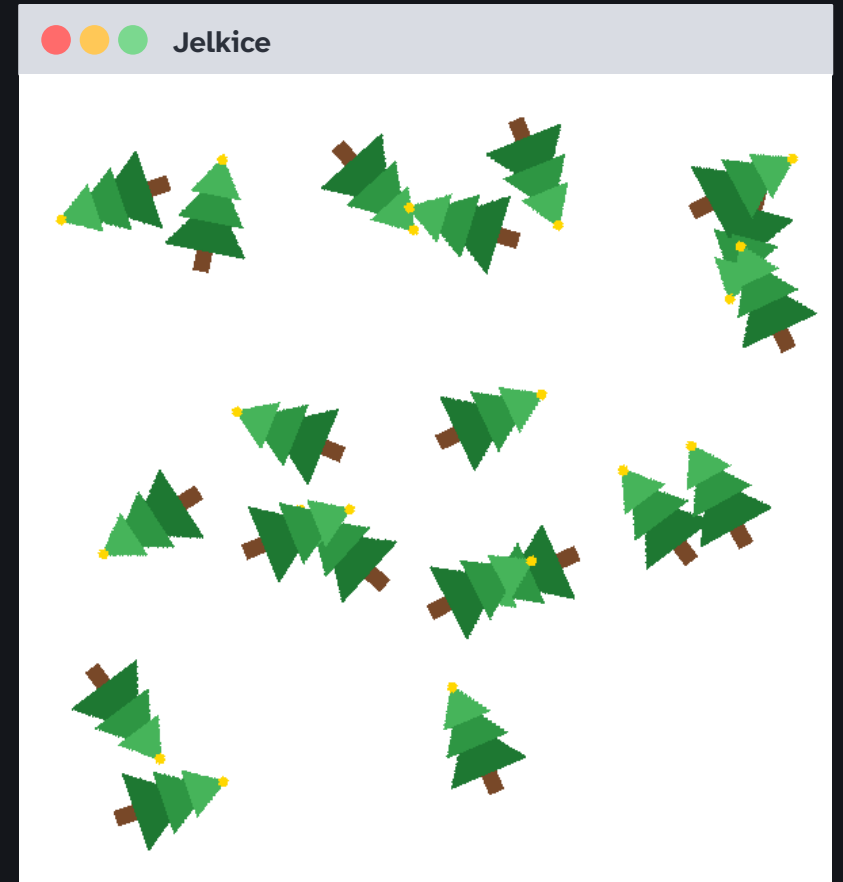
```
def nasumicna_boja():  
    return (random.randint(0, 255),  
            random.randint(0, 255),  
            random.randint(0, 255))  
  
for i in range(200):  
    x = random.randint(0, sirina)  
    y = random.randint(0, visina)  
    boja = nasumicna_boja()  
    pg.draw.circle(ekran, boja, (x, y), 4)
```



Rotiranu sliku postavljamo preko centra

```
slika = pg.image.load("jelkica.png")
slika = slika.convert_alpha()

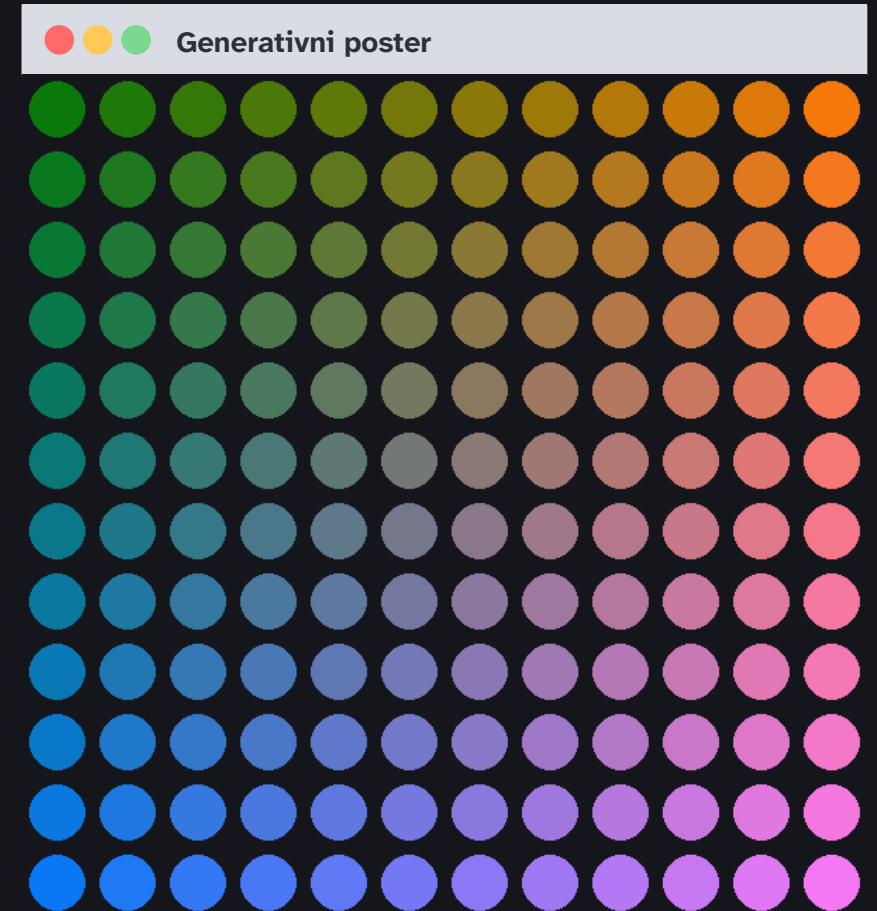
for i in range(20):
    ugao = random.randint(0, 360)
    rotirana = pg.transform.rotate(slika, ugao)
    # centar jelkice biramo nasumično
    x = random.randint(50, sirina - 50)
    y = random.randint(50, visina - 50)
    okvir = rotirana.get_rect(center=(x, y))
    ekran.blit(rotirana, okvir)
```



Pravimo generativni poster

Cilj: za 15 minuta imamo poster u kom boja svakog kruga zavisi od njegovog mesta.

- 1 Okvir programa: prozor 600 × 600, tamna pozadina
- 2 Jedan red krugova: petlja po x sa korakom 50
- 3 Mreža: petlja po y u petlji po x
- 4 Boja: crvena raste udesno, plava nadole
- 5 Dodatak: i poluprečnik zavisi od položaja



Petlja u petlji pravi mrežu, a položaj određuje boju

```
korak = 50
for x in range(korak // 2, sirina, korak):
    for y in range(korak // 2, visina, korak):
        # boja zavisi od položaja kruga
        crvena = x * 255 // sirina
        plava = y * 255 // visina
        boja = (crvena, 120, plava)
        pg.draw.circle(ekran, boja, (x, y), 20)
```

korak = 50

razmak između centara krugova

x * 255 // sirina

od 0 levo do skoro 255 desno

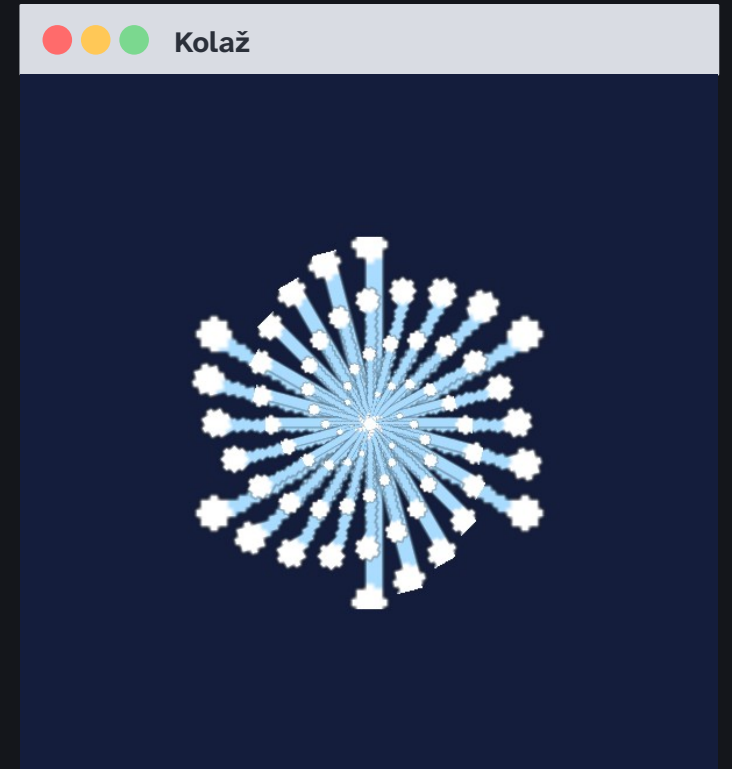
(crvena, 120, plava)

zelena je ista za sve krugove

Kolaž: jedna slika, rotirana i skalirana u petlji

```
slika = pg.image.load("pahuljica.png")
slika = slika.convert_alpha()
centar = (sirina // 2, visina // 2)

for i in range(12):
    ugao = i * 15
    razmera = 4 - i * 0.3
    nova = pg.transform.rotozoom(slika, ugao, razmera)
    okvir = nova.get_rect(center=centar)
    ekran.blit(nova, okvir)
```



Pronađi grešku

```
import pygame as pg

pg.init()
ekran = pg.display.set_mode((400, 300))
ekran.fill("white")
pg.draw.circle(ekran, "red", (200, 150))
pg.display.update()
```

Program se ne pokreće do kraja. Šta nedostaje u šestom redu?

PORUKA O GREŠCI

```
Traceback (most recent call last):
  File "krug.py", line 6, in <module>
    pg.draw.circle(ekran, "red", (200, 150))
TypeError: function missing required
argument 'radius' (pos 4)
```

line 6

red u kom je greška

TypeError

vrsta greške: pogrešni argumenti

'radius' (pos 4)

nedostaje četvrti argument: poluprečnik

Izaberite nivo i počnite

LEVEL 1

Noćno nebo

Izmenite šarene tačkice: tamnoplava pozadina i 200 belih zvezda, svaka poluprečnika od 1 do 3.

Pomoć: i poluprečnik može da bude `random.randint(1, 3)`.

LEVEL 2

Čestitka

Jelkica u centru prozora i natpis „Srećni praznici!“ centriran ispod nje.

Pomoć: `pg.transform.scale` uvećava sliku; `get_rect(center=...)` centrira.

LEVEL 3

Šahovska tabla

Tabla 8×8 od dve boje i jedna figura (krug) na nasumičnom polju.

Pomoć: polje je svetlo kad je $(\text{red} + \text{kolona}) \% 2$ jednako 0.

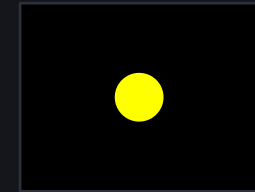
U PZU2 petlja crta 60 slika u sekundi

```
sat = pg.time.Clock()
x = 0
kraj = False
while not kraj:
    for dogadjaj in pg.event.get():
        if dogadjaj.type == pg.QUIT:
            kraj = True
    ekran.fill("black")
    pg.draw.circle(ekran, "yellow", (x, 150), 20)
    pg.display.update()
    x = (x + 3) % sirina
    sat.tick(60) # najviše 60 kadrova u sekundi
pg.quit()
```

1. kadar



2. kadar



3. kadar



Kadar je jedna slika. Svaki prolaz kroz petlju briše prozor, crta krug malo desnije i prikazuje novi kadar. Sat ne dozvoljava više od 60 prolaza u sekundi.

U PZU2: animacija, tastatura i miš, sprajtovi i prva igra.

Rezime

Okvir programa

```
pg.init()
ekran = pg.display.set_mode((400, 300))
pg.display.update()
```

Bez update prozor ostaje prazan.

Koordinate i boje

```
(0, 0)           # gore levo
(255, 165, 0)    # narandžasta
```

y raste nadole; komponente od 0 do 255.

Oblici

```
pg.draw.circle(ekran, boja, (x, y), r)
pg.draw.rect(ekran, boja, (x, y, w, h))
```

Debljina 0 ili izostavljena: popunjen oblik.

Slike i tekst

```
okvir = slika.get_rect(center=c)
ekran.blit(slika, okvir)
```

Tekst je sličica koju pravi font.render.

Pitanje pre odlaska

```
# prozor je 400 × 300  
pg.draw.circle(ekran, "red", (390, 10), 8)
```

U kom uglu prozora se pojavljuje crveni krug?

Odgovor pošaljite pre nego što izađete. Dve reči su dovoljne.



Literatura

- Dokumentacija za pygame-ce (Community Edition): pyga.me/docs
- Dokumentacija za PyGame: pygame.org/docs
- Sweigart, A. Invent Your Own Computer Games with Python, 4. izdanje, No Starch Press (besplatno na inventwithpython.com)
- Kafle, S. Learning Python by Building Games, Packt, 2019

Dodatna literatura i zanimljivi sajtovi

pygame-ce: dokumentacija

pyga.me/docs

Opis svake funkcije biblioteke pygame-ce, sa primerima.

Kenney

kenney.nl/assets

Grafike i zvukovi za igre, besplatni pod licencom CC0.

OpenGameArt

opengameart.org

Grafike, muzika i zvukovi za igre; licenca piše uz svaki rad.

Freesound

freesound.org

Zvučni efekti pod Creative Commons licencama.

Piskel

piskelapp.com

Besplatan editor u pregledaču za piksel-art i animirane sprajtove.

Lospec: palete boja

lospec.com/palette-list

Biblioteka gotovih paleta boja za piksel-art.

Hvala!

Pitanja su uvek dobrodošla: na času, u četvrtu ili međuvremeno.

Slajdovi, primeri i kartica za podsećanje su na sajtu predmeta.

PREDAVANJA

dr Staša Vujičić Stanković

stasa.vujicic.stankovic@math.rs

www.matf.bg.ac.rs/~stasa.vujicic.stankovic

VEŽBE

Nevena Ćirić

poincare.matf.bg.ac.rs/~nevena.ciric/p1uu.html