

Termin XXVI

4. maj 2019.



→ Pristup pygame-u:

◆ <https://repl.it/languages/pygame>

→ Pristup dostupnoj dokumentaciji za pygame:

◆ <https://www.pygame.org/docs/>

→ Jednostavan uvodni tutorijal za pygame:

◆ <https://nerdparadise.com/programming/pygame>

❖ inicijalni_kod.py

```
01. # uvozenje modula pygame koji sadrzi display, event i druge objekte koji
    omogucavaju pravljenje animacija i igrice
02. import pygame
03. # metoda koja inicijalizuje sve uvezene pygame module
04. pygame.init()
05. # kreiranje prozora (deo prikaza koji u gornjem levom uglu sadrzi naziv
    igrice, a u gornjem desnom uglu sadrzi [_] [X] )
06. # display.set_mode((x, y)) - kreira prozor koji je dimenzija 400 x 300
07. pygame.display.set_mode((400, 300))
08. # display.set_caption(title, icontitle=None) - postavlja naziv prozora i
    ikonicu u kornjem levom uglu
09. pygame.display.set_caption('Padajuci kvadrati')
10. # ideja igrice/animacije je da se prikazuju razliciti kadrovi nekom
    brzinom fps (kadrovi po sekundi) sve dok program traje. Kako bi se ovo
    postiglo, cesto imamo promenljivu kraj_igre/kraj_programa koji se
    postavlja na False.
11. kraj_igre = False
12. # Program treba da traje sve dok nije tacno da je kraj igre. Kada igra
    treba da se prekine, kraj_igre se postavlja na True. U tom slucaju, while
    not True ce biti False, te petlja se istog trenutka prekida i glavni deo
    programa se zavrшава.
13. while not kraj_igre:
14. # U svakom trenutku program ceka neku vrstu dogadjaja. Za svaki poznati
    dogadjaj (pritisak na [X] da bi se iskljucio program, pritisak dugmeta,
    klik misem itd.) treba da se izvrši neki kod.
15.     for dogadjaj in pygame.event.get():
16. # Ako je taj dogadjaj poznatog tipa, (konkretno pygame.QUIT) onda znamo da
    je korisnik aplikacije kliknuo na [X] i da tada treba da zatvorimo prozor
    za igricu.
17.     if dogadjaj.type == pygame.QUIT:
18.         kraj_igre = True
19. # Sva izmenjena desavanja nad display treba biti sacuvana
20. pygame.display.flip()
```

1. zuti_pravougaonik.py

Napisati Python program koristeći modul pygame koji na koordinatama (0, 300) iscrtava žuti pravougaonik veličine 50 x 100 i ubacuje ga u beli prozor veličine 500 x 250. Animacija je takva da se pravougaonik na svakih 1.5s pomera udesno za 50. Prozor treba iscrtavati 60 puta u sekundi. Animacija se sama završava kada pravougaonik izađe van prozora.

```
01. import pygame
02.
03. pygame.init()
04.
05. prozor_sirina = 500
06. prozor_visina = 250
07. prozor_boja = (255, 255, 255)
08. prozor = pygame.display.set_mode((prozor_sirina, prozor_visina))
09. pygame.display.set_caption("Zuti pravougaonik")
10. kraj_igre = False
11. clock = pygame.time.Clock()
12.
13. pravougaonik_sirina = 100
14. pravougaonik_visina = 100
15. pravougaonik_x = 0
16. pravougaonik_y = 300
17. pravougaonik_boja = (255, 255, 0)
18.
19. pravougaonik = pygame.Rect(pravougaonik_x, pravougaonik_y,
    pravougaonik_sirina, pravougaonik_visina)
20.
```

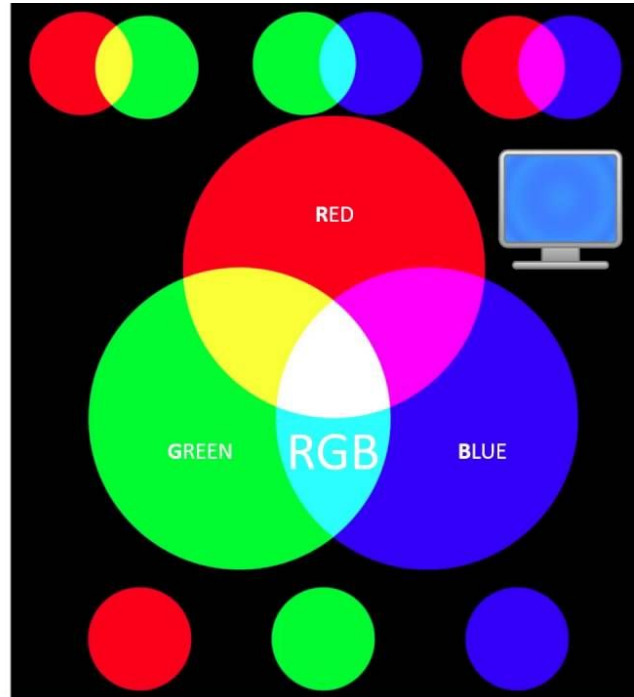
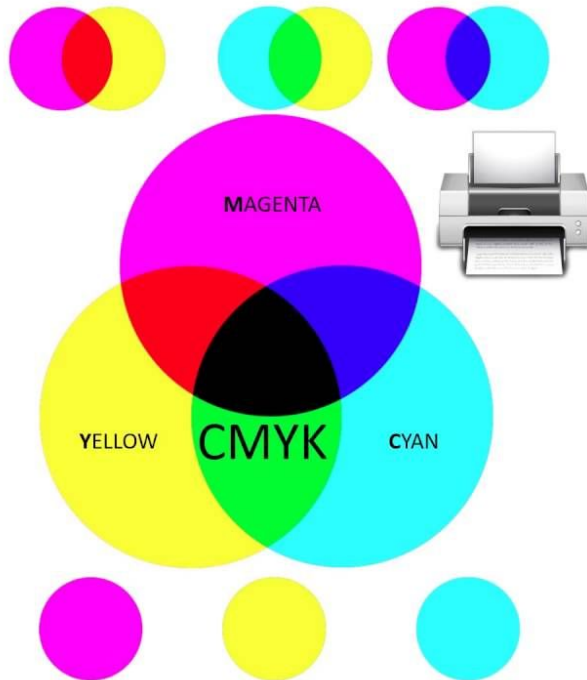
```
21. print("Igra je zapoceta.")
22. while not kraj_igre:
23.     for dogadjaj in pygame.event.get():
24.         if dogadjaj.type == pygame.QUIT:
25.             kraj_igre = True
26.
27.     pravougaonik.x += 50
28.     if pravougaonik.x >= prozor_sirina:
29.         kraj_igre = True
30.
31.     prozor.fill(prozor_boja)
32.     pygame.draw.rect(prozor, pravougaonik_boja, pravougaonik)
33.
34.     pygame.display.flip()
35.     pygame.time.wait(1000)
36.     clock.tick(60)
37. print("Igra je završena.")
```

2. padajuci_sareni_krugovi.py

Napisati Python program koristeći modul turtle koji na svakoj stotoj koordinati na horizontalnoj osi, od nulte tačke na vertikalnoj osi generiše po 1 krug na svakih 15 frame-ova. Prozor je veličine 600 x 600 i treba iscrtavati 60 puta u sekundi.

```
38. import pygame as p
39.
40. p.init()
41. screen = p.display.set_mode((500, 200))
42. game_end = False
43. clock = p.time.Clock()
44. x = [100, 150, 200, 250, 300]
45. y = [15, 30, 45, 60, 75]
46. colours = [(255,0,0), (0,255,0), (0,0,255), (125, 0, 125), (255, 255, 0)]
47.
48. while not game_end:
49.     for event in p.event.get():
50.         if event == p.QUIT:
51.             game_end = True
52.
53.     for i in range(len(y)):
54.         y[i] += 1
55.     screen.fill((255, 255, 255))
56.     for i in range(len(y)):
57.         if y[i] >= (200 - 20):
58.             y[i] = (i + 1) * 15
59.     p.draw.rect(screen, colours[i], p.Rect(x[i], y[i], 20, 20))
60.
61. p.display.flip()
62. clock.tick(120)
```

3. turtle_boje.py



```
01. import turtle as t
02. boje = [
03.     ["crvena", (255, 0, 0)],      # boje[0][1] - crvena
04.     ["zelena", (0, 255, 0)],     # boje[1][1] - zelena
05.     ["plava", (0, 0, 255)],     # boje[2][1] - plava
06.     ["zuta", (255, 255, 0)],    # boje[3][1] - zuta
07.     ["pink", (255, 0, 255)],    # boje[4][1] - pink
08.     ["cyan", (0, 255, 255)],    # boje[5][1] - cyan
09.     ["bela", (255, 255, 255)],  # boje[6][1] - bela
10.     ["crna", (0, 0, 0)]        # boje[7][1] - crna
11. ]
```