

Za radoznale: Sprajтови, grupe i animacija lika

Ovo nije obavezno i neće biti na ispitu. Za one koji žele da vide kako se ista stvar radi naprednije. Svi ispisi su dobijeni pokretanjem u pygame-ce.

1. List sa sprajtovima: sve slike hoda u jednoj slici

Na času je svaka slika hoda bila poseban PNG fajl. Gotove grafike za igre (na primer na sajtovima Kenney i OpenGameArt) često dolaze kao jedna velika slika u kojoj su svi frejmovi poređani jedan do drugog: to se zove list sa sprajtovima (engl. sprite sheet). Metoda subsurface iseca deo površine, pa iz jednog lista dobijamo listu frejmova.

```
import pygame as pg

# pravimo list: četiri frejma 40 × 64 jedan do drugog
SIRINA, VISINA = 40, 64
list_slika = pg.Surface((4 * SIRINA, VISINA))
for i in range(4):
    pg.draw.circle(list_slika, (60 + 60 * i, 90, 200), (20 + i * SIRINA, 32), 15)

# isecamo frejmove iz lista
frejmovi = []
for i in range(4):
    deo = pg.Rect(i * SIRINA, 0, SIRINA, VISINA)
    frejmovi.append(list_slika.subsurface(deo))

print(len(frejmovi), frejmovi[0].get_size())
print(frejmovi[2].get_at((20, 32)))
```

Ispis:

```
4 (40, 64)
Color(180, 90, 200, 255)
```

Svaki frejm je deo velikog lista, širok 40 i visok 64 piksela. Treći frejm (indeks 2) u svojoj sredini ima boju trećeg kruga. U pravoj igri list se učitava sa `image.load`, a ostatak je isti.

Zašto ovo nismo radili na času: naše slike crta program, pa je bilo jednostavnije da svaku sliku snimi posebno; list sa sprajtovima je zgodan kad grafiku preuzimate gotovu.

Fajl: `primeri/za_radoznale/09_r1_list_sa_sprajtovima.py`

2. LayeredUpdates: slojevi i promena sloja

Na času smo `LayeredUpdates` samo pomenuli. Grupa crta sprajtove po slojevima: prvo najniži sloj. Sloj se zadaje pri dodavanju, a može i da se promeni, na primer kad lik prođe iza jelke pa ispred nje.

```

import pygame as pg

class Ime(pg.sprite.Sprite):
    def __init__(self, ime):
        super().__init__()
        self.ime = ime
        self.image = pg.Surface((10, 10))
        self.rect = self.image.get_rect()

sve = pg.sprite.LayeredUpdates()
sve.add(Ime("sneg"), layer=2)
cikica = Ime("Čikica")
sve.add(cikica, layer=1)
sve.add(Ime("jelka"), layer=1)
print([s.ime for s in sve.sprites()])      # redom crtanja

sve.change_layer(cikica, 3)                # Čikica ide ispred svega
print([s.ime for s in sve.sprites()])

```

Ispis:

```

['Čikica', 'jelka', 'sneg']
['jelka', 'sneg', 'Čikica']

```

sprites() vraća sprajtove redom kojim se crtaju: prvo sloj 1, pa sloj 2. Posle change_layer Čikica je u sloju 3, pa se crta poslednji, ispred snega i jelke.

Zašto ovo nismo radili na času: u igri sa dve vrste sprajtova dovoljno je da dve grupe nacrtamo pravim redom.

Fajl: `primeri/za_radoznale/09_r2_slojevi.py`

3. Animacija po satu igre: get_ticks

Na času smo vreme jedne slike sabirali iz dt. Drugi način je da pitamo koliko je milisekundi prošlo od pokretanja programa (pg.time.get_ticks) i iz toga izračunamo koja je slika na redu. Tako ne treba promenljiva koja pamti vreme, ali svi likovi idu u istom ritmu.

```

import pygame as pg

TRAJANJE = 120          # milisekundi po slici
BROJ_SLIKA = 4

for ms in [0, 100, 130, 250, 480, 1000]:
    indeks = (ms // TRAJANJE) % BROJ_SLIKA
    print(ms, "ms -> slika", indeks)

pg.init()
print(pg.time.get_ticks() >= 0)      # milisekunde od pg.init()

```

Ispis:

```

0 ms -> slika 0
100 ms -> slika 0
130 ms -> slika 1
250 ms -> slika 2
480 ms -> slika 0
1000 ms -> slika 0
True

```

Celobrojno deljenje kaže koliko je celih trajanja prošlo, a ostatak pri deljenju sa 4 vrti slike ukrug. Posle 1000 ms prošlo je 8 celih trajanja, pa je slika $8 \% 4 = 0$.

Zašto ovo nismo radili na času: sa `get_ticks` animacija teče i dok je igra na pauzi, a svi likovi mrdaju nogama u istom taktu. Kad svaki lik sam sabira `dt`, nema ni jednog ni drugog problema.

Fajl: `primeri/za_radoznale/09_r3_animacija_po_satu.py`