

Zvuk, transformacije slike i čestice

Kako igra zvuči, kako se slika okreće i kako nastaje rasprskavanje

Danas pravimo: „dzing“, muziku i pahuljice koje se rasprsnu



Uhvaćena pahuljica zazvoni i rasprsne se u dvadeset tačkica

Tri pitanja za zagrevanje

1 • prošla tema: Rect

```
import pygame as pg

a = pg.Rect(0, 0, 40, 40)
b = pg.Rect(30, 30, 40, 40)
t = (40, 40)
print(a.colliderect(b))
print(a.collidepoint(t))
```

True
False

2 • tema 09: slike hoda

```
s = ["a", "b", "c", "d"]
vreme = 0.5
i = int(vreme / 0.12) % 4
print(s[i])
```

a

3 • mešano: max

```
a = 255
for korak in range(4):
    a = max(0, a - 100)
print(a)
```

0

Šest novih reči za danas

zvučni efekat

kratak zvuk uz događaj u igri: skok, udarac, poen

muzika u pozadini

duga melodija koja svira dok traje igra

uzorak

jedan broj u zapisu zvuka; 22 050 uzoraka u sekundi

transformacija

nova slika od stare: uvećana, okrenuta, u ogledalu

čestica

mala tačkica koja kratko živi: iskra, prah, sneg

alfa

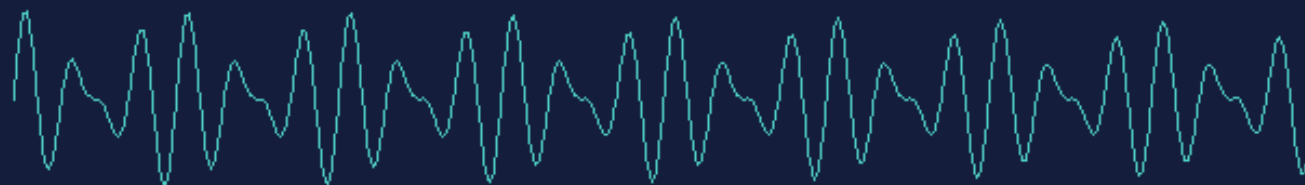
providnost: 255 je neprovidno, 0 je potpuno providno

Zvuk

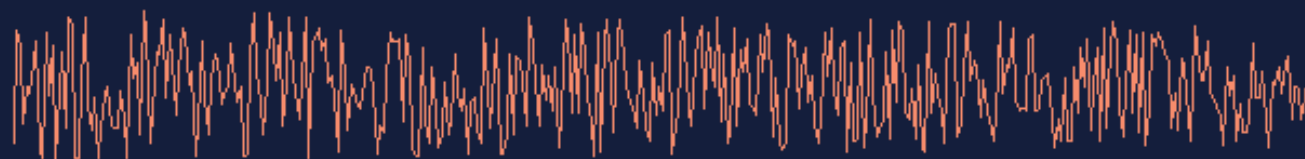
Efekti i muzika, napravljeni kodom

Zvuk je dugačak niz brojeva

dzing.wav: sinusi koji se gase



bum.wav: šum (slučajni brojevi)



prve dve stotinke oba zvuka

- Zvučnik se pomera napred-nazad; svaki broj je jedan položaj.
- 22 050 brojeva (uzoraka) u sekundi.
- Sinus daje čist ton, a slučajni brojevi daju šum.
- Brži talas je viši ton: 1320 talasa u sekundi je visok „dzing“.

„Dzing“ pravimo kodom, modulom wave

```
def ton(visina, trajanje, jacina=0.5, gasenje=6):  
    uzorci = []  
    for i in range(int(FREKVENCIJA * trajanje)):  
        t = i / FREKVENCIJA # vreme u sekundama  
        talas = math.sin(2 * math.pi * visina * t)  
        uzorci.append(jacina * talas * math.exp(-gaseenje * t))  
    return uzorci  
  
dzing = ton(1320, 0.35) # 0,35 s, ton 1320 Hz  
snimi_zvuk("dzing.wav", dzing)
```

`math.sin`

glatki talas od -1 do 1

`math.exp(-6 * t)`

ton se postepeno gasi

`wave`

modul koji snima WAV fajl

Ceo program, sa „bum“ i muzikom: 11_01_pravimo_slicice_i_zvuke.py

Kratak efekat ili duga muzika

zvučni efekat

`pg.mixer.Sound`

- ceo zvuk se učitava u memoriju
- više zvukova svira istovremeno
- „dzing“, „bum“, skok

muzika u pozadini

`pg.mixer.music`

- čita se iz fajla dok svira
- u jednom trenutku samo jedna
- melodija koja se ponavlja

Formati: WAV, OGG, MP3, FLAC. Za igre preporučujemo WAV za kratke efekte i OGG za muziku.

Sound: učitamo jednom, pustimo kad treba

```
import pygame as pg

pg.mixer.init()
dzing = pg.mixer.Sound("dzing.wav")
dzing.set_volume(0.5)          # od 0 do 1
dzing.play()
print(round(dzing.get_length(), 2))
print(dzing.get_volume())
print(pg.mixer.get_num_channels())
```

ISPIS

0.35

0.5

8

get_length()

trajanje u sekundama

8 kanała

do 8 zvukova odjednom

Šta ispisuje ovaj program?

```
import pygame as pg

pg.mixer.init()
dzing = pg.mixer.Sound("dzing.wav")
dzing.set_volume(1.5)
print(dzing.get_volume())
```

Pokušavamo da zvuk pustimo jače od najjačeg.

A
1.5

B tačno
1.0

C
0.5

D
greška:
ValueError

Muzika svira u krug, a M je zaustavlja

```
pg.mixer.music.load("muzika.wav")
pg.mixer.music.set_volume(0.3)
pg.mixer.music.play(-1)          # -1: ponavlja se stalno
svira = True
...
# u obradi događaja:
if dogadjaj.type == pg.KEYDOWN and dogadjaj.key == pg.K_m:
    if svira:
        pg.mixer.music.pause()
    else:
        pg.mixer.music.unpause()
    svira = not svira
```

play(-1)

ponavlja se dok igra traje

pause / unpause

zaustavi i nastavi od istog mesta

fadeout(2000)

utiša se za 2 sekunde

Kad pesma stane, stiže naš događaj

```
KRAJ_PESME = pg.event.custom_type()
pg.mixer.music.load("muzika.wav")
pg.mixer.music.set_endevent(KRAJ_PESME)
pg.mixer.music.play()          # jednom
...
for dogadjaj in pg.event.get():
    if dogadjaj.type == KRAJ_PESME:
        print("Pesma je završena")
```

`custom_type()`

nov, sopstveni tip događaja

`set_endevent`

koji događaj stiže na kraju

Zadatak iz starih beleški:
reprodukujeZvuk.py

Transformacije slike

Uvećaj, okreni, preslikaj

Transformacija pravi novu sliku, a stara ostaje

```
import pygame as pg

lik = pg.Surface((40, 64))
print(pg.transform.scale(lik, (80, 128)).get_size())
print(pg.transform.flip(lik, True, False).get_size())
print(pg.transform.rotozoom(lik, 0, 2).get_size())
print(lik.get_size())
```

ISPIS

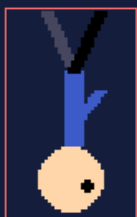
```
(80, 128)
(40, 64)
(80, 128)
(40, 64)
```



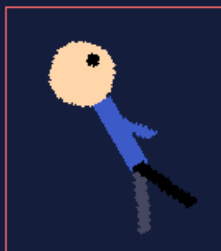
original



flip(True, False)



flip(False, True)



rotate(30)



rotozoom(-30, 0.7)

flip(slika, True, False): ogledalo
levo-desno

flip(slika, False, True): naopako

rotozoom: okreće i menja veličinu

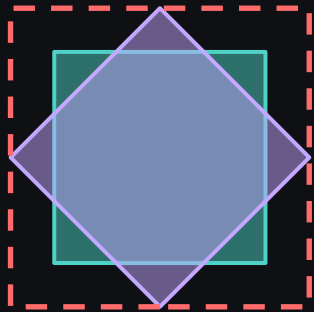
Okrenuta slika je veća od originala

```
import pygame as pg

slika = pg.Surface((24, 24))
for ugao in [0, 30, 45, 90]:
    nova = pg.transform.rotate(slika, ugao)
    print(ugao, nova.get_size())
```

ISPIS

```
0 (24, 24)
30 (32, 32)
45 (33, 33)
90 (24, 24)
```



Uglovi okrenutog kvadrata ne staju u stari okvir, pa nova slika mora da bude veća (crveni okvir). Višak je providan.

Pahuljica se okreće iz originala i ostaje u centru

```
# u __init__:  
self.original = SLIKA_PAHULJICE          # nikad je ne menjamo  
self.ugao = random.uniform(0, 360)  
self.brzina_okretanja = random.uniform(-90, 90)  # °/s  
  
# u update(self, dt):  
self.ugao += self.brzina_okretanja * dt  
self.image = pg.transform.rotate(self.original, self.ugao)  
self.rect = self.image.get_rect(center=self.polozaj)
```

self.original

čuva se nepromenjena

self.ugao

raste za brzina × dt

center=

novi rect, isti centar

Čudovište: slika koja se raspada



crveni okvir: veličina slike

Kako izgleda

Okretana slika postaje mutna i iskrzana, a program sve sporiji.

Zašto se pojavljuje

Okreće se slika iz prošlog frejma. Svako okretanje je doda providan okvir i pokvari piksele.

Kako se pobeđuje

Original se čuva, a u svakom frejmu okreće se on, za ukupan ugao.

Svako okretanje okrenute slike je povećava

```
import pygame as pg

slika = pg.Surface((24, 24))
for i in range(5):
    slika = pg.transform.rotate(slika, 10)
    print(slika.get_size())
```

ISPIS

```
(27, 27)
(31, 31)
(35, 35)
(40, 40)
(46, 46)
```

- Pet puta po 10° je ukupno 50°, ali slika je skoro dvostruko šira.
- Iz originala bi okretanje za 50° dalo sliku oko 34 × 34.

Kolika je slika posle okretanja za 90°?

```
import pygame as pg

slika = pg.Surface((40, 20))
nova = pg.transform.rotate(slika, 90)
print(nova.get_size())
```

Slika je široka 40, a visoka 20 piksela.

A

(40, 20)

B

tačno

(20, 40)

C

(40, 40)

D

(45, 45)

Čestice

Mnogo malih tačkica koje kratko žive

Rasprskavanje je dvadeset malih sprajtova



- Svaka čestica ima položaj, brzinu i preostalo vreme.
- Gravitacija joj svakog frejma povećava brzinu nadole.
- Kako vreme ističe, čestica blede, a na kraju kill.

Čestica kreće iz tačke u slučajnom smeru

```
class Cestica(pg.sprite.Sprite):  
    TRAJANJE = 0.8 # sekundi  
    def __init__(self, pozicija):  
        super().__init__()  
        self.image = pg.Surface((5, 5), pg.SRCALPHA)  
        self.image.fill((120, 190, 255)) # svetloplava  
        self.pozicija = pg.Vector2(pozicija)  
        self.rect = self.image.get_rect(center=self.pozicija)  
        ugao = random.uniform(0, 360)  
        duzina = random.uniform(40, 160) # px/s  
        self.brzina = pg.Vector2(duzina, 0).rotate(ugao)  
        self.preostalo = self.TRAJANJE
```

TRAJANJE

atribut klase, isti za sve

rotate(ugao)

okreće strelicu brzine

SRCALPHA

površina koja može da bleedi

Čestica pada, bleedi i nestaje

```
def update(self, dt):  
    self.brzina.y += GRAVITACIJA * dt      # pada sve brže  
    self.položaj += self.brzina * dt  
    self.rect.center = self.položaj  
    self.preostalo -= dt  
    if self.preostalo <= 0:  
        self.kill()  
    else:  
        alfa = 255 * self.preostalo / self.TRAJANJE  
        self.image.set_alpha(int(alfa))    # bleedi
```

```
def rasprsnj(položaj, grupa):  
    for i in range(20):  
        grupa.add(Cestica(položaj))
```

GRAVITACIJA

300 px/s više svake sekunde

set_alpha

0 providno, 255 neprovidno

rasprsnj

20 čestica iz iste tačke

rotate okreće strelicu oko početka

```
import pygame as pg

v = pg.Vector2(100, 0)
for ugao in [0, 90, 180]:
    w = v.rotate(ugao)
    print(ugao, round(w.x), round(w.y))
```

ISPIS

```
0 100 0
90 0 100
180 -100 0
```

- Dužina ostaje 100; menja se samo smer.
- Na ekranu y raste nadole, pa (0, 100) pokazuje dole: Vector2 se na ekranu okreće u smeru kazaljke na satu.

Koliko je providna čestica pred kraj?

```
TRAJANJE = 0.8  
preostalo = 0.2  
print(int(255 * preostalo / TRAJANJE))
```

Čestica živi 0,8 s, a ostalo joj je još 0,2 s.

A

tačno

63

B

191

C

64

D

0

Zimska šetnja, korak 6: zvuk i čestice

Cilj: igra zvuči, pahuljice se okreću, a uhvaćena pahuljica se rasprsne.

- 1 Zvuci: DZING, BUM i muzika posle pg.init()
- 2 Muzika play(-1); taster M je zaustavlja
- 3 Pahuljica: original, ugao, rotate, centar
- 4 Klasa Cestica i funkcija rasprsnj
- 5 U petlji: dzing i čestice za svaku uhvaćenu

KAKO IZGLEDA NA KRAJU



Sličice i zvuke pravi 11_01; ceo kod je u 11_10.

U petlji igre su samo tri nova mesta

```
uhvacene = pg.sprite.spritecollide(
    igrac, pahuljice, True, pg.sprite.collide_mask)
for p in uhhvacene:
    DZING.play()
    rasprsni(p.rect.center, cestice)
poeni += len(uhhvacene)
...
cestice.update(dt)
...
cestice.draw(ekran)                # posle igrača, pre teksta
```

DZING.play()

jednom za svaku pahuljicu

p.rect.center

pahuljica je nestala, mesto
pamtimo

cestice.draw

iznad igrača, ispod teksta

Pronađi grešku: program stane u prvom redu

```
import pygame as pg

DZING = pg.mixer.Sound("dzing.wav")
pg.init()
ekran = pg.display.set_mode((640, 360))
```

Fajl dzing.wav postoji i ispravan je. Zašto program ne radi?

PORUKA

```
pygame.error: mixer not initialized
```

mixer

deo pygame-a za zvuk

pg.init()

uključuje mikser; mora prvi

Izaberite nivo i počnite

LEVEL 1

Zvučna tabla

D pušta „dzing“, B „bum“, M zaustavlja muziku, a strelice menjaju jačinu.

*Pomoć: KEYDOWN i
pg.mixer.music.set_volume.*

LEVEL 2

Čikica u ogledalu

Slike hoda nalevo napravite od slika nadesno, a Čikicu uvećajte dva puta.

*Pomoć: transform.flip i
transform.scale.*

LEVEL 3

Vatromet

Klik mišem pravi rasprskavanje 40 iskri jedne slučajne boje, uz pucanj.

*Pomoć: klasa Cestica,
dogadjaj.pos, bum.wav.*

Sledeći put igra dobija meni i kraj

```
class Meni:  
    def obradi(self, dogadjaj): ...  
    def azuriraj(self, dt): ...  
    def crtaj(self, ekran): ...  
  
scena = Meni()      # kasnije: Igra(), Pauza(), Kraj()
```

U temi 12 igra se deli na stanja: meni, igra, pauza i kraj. Svako stanje je klasa sa iste tri metode, a dugme „Ponovo“ pokreće novu igru. Tu je i plan za seminarski rad.

Rezime

zvuk

```
z = pg.mixer.Sound("dzing.wav")  
z.set_volume(0.5); z.play()
```

Posle pg.init(); jačina od 0 do 1.

muzika

```
pg.mixer.music.load(...)  
pg.mixer.music.play(-1)
```

Samo jedna; pause, unpause, fadeout.

transformacije

```
scale, flip, rotate, rotozoom  
rotate(original, ugao)
```

Nova slika; centar preko get_rect.

čestice

```
položaj, brzina, preostalo  
set_alpha(...), kill()
```

Gravitacija, bleđenje, nestanak.

Pitanje pre odlaska

```
self.ugao += 90 * dt  
osnova = self._____  
self.image = pg.transform.rotate(osnova, self.ugao)
```

Pahuljica treba da se okreće 90 stepeni u sekundi i da ostane oštra. Šta upisujemo na crtu?

Pošaljite jednu reč pre nego što izađete.



Literatura

- Dokumentacija za pygame-ce, moduli mixer, mixer.music i transform: pyga.me/docs
- Dokumentacija za Python, modul wave: docs.python.org/3/library/wave.html
- Sweigart, A. Invent Your Own Computer Games with Python, 4. izdanje, No Starch Press, poglavlje 20 o slikama i zvuku (besplatno na inventwithpython.com)
- Kafle, S. Learning Python by Building Games, Packt, 2019 (poglavlja o PyGame-u)

Dodatna literatura i zanimljivi sajtovi

pygame-ce: modul mixer

pyga.me/docs/ref/mixer.html

Sound, jačina, kanali i formati koje mikser učitava.

pygame-ce: modul transform

pyga.me/docs/ref/transform.html

scale, flip, rotate, rotozoom i druge transformacije.

Python: modul wave

docs.python.org/3/library/wave.html

Čitanje i pisanje WAV fajlova iz standardne biblioteke.

Freesound

freesound.org

Zvučni efekti pod Creative Commons licencama.

Audacity

audacityteam.org

Besplatan program za snimanje i uređivanje zvuka.

Hvala!

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

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

Milan Kocić

milan.kocic@matf.bg.ac.rs

poincare.matf.bg.ac.rs/~milan.kocic