

Sudari i pravila igre

Kako igra zna da se dve stvari dodiruju i šta se tada desi

Danas pravimo: poene i živote u igri „Zimska šetnja“



Pahuljica donosi poen, grudva odnosi život

Tri pitanja za zagrevanje

1 • prošla tema: grupa

```
import pygame as pg

g = pg.sprite.Group()
for i in range(5):
    s = pg.sprite.Sprite()
    g.add(s)
s.kill()
print(len(g))
```

4

2 • tema 07: brzina * dt

```
brzina = 220      # px/s
x = 100
for dt in [0.05, 0.05]:
    x += brzina * dt
print(x)
```

122.0

3 • mešano: niske

```
zivoti = 3
zivoti -= 1
print("❤️" * zivoti)
```

❤️❤

Šest novih reči za danas

sudar

dva predmeta u igri se dodiruju ili preklapaju

pravougaonik sudara

rect sprajta: nevidljiv okvir po kome se meri sudar

maska

mapa piksela slike: neprovidan (1) ili providan (0)

poluprečnik

atribut radius: veličina kruga za sudar krugovima

pravila igre

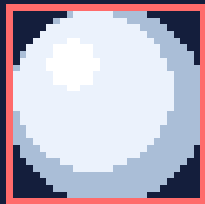
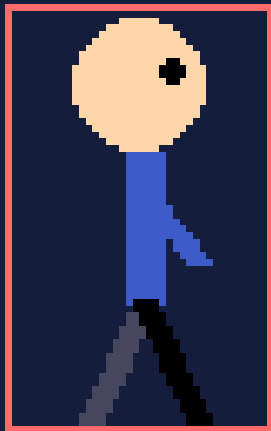
šta donosi poen, šta oduzima život, kad je kraj

neranjivost

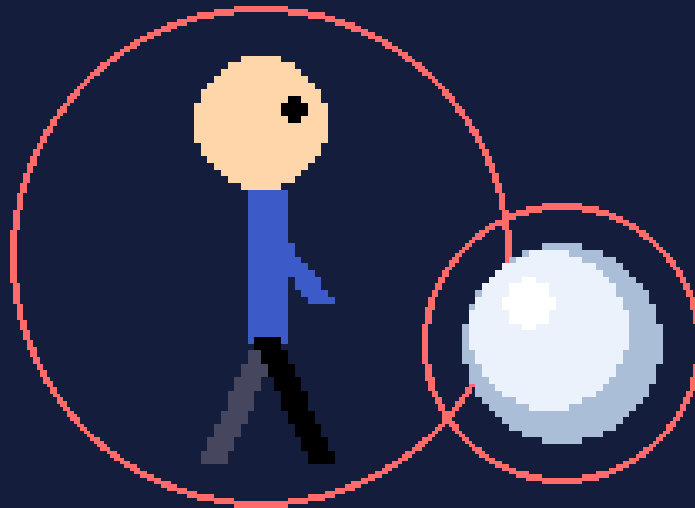
kratko vreme posle udarca kada lik ne gubi život

Igra ne vidi lik, nego njegov oblik za sudar

pravougaonik



krug



maska



Sudar dva pravougaonika

Najjednostavniji i najbrži način

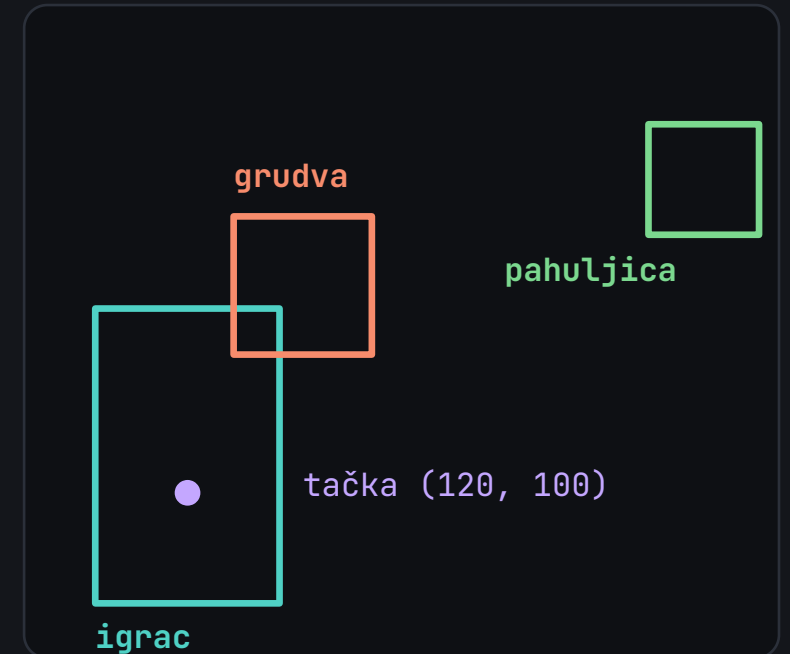
collidect pita: da li se preklapaju?

```
import pygame as pg

igrac = pg.Rect(100, 60, 40, 64)
grudva = pg.Rect(130, 40, 30, 30)
pahuljica = pg.Rect(220, 20, 24, 24)
print(igrac.collidect(grudva))
print(igrac.collidect(pahuljica))
print(igrac.collidepoint(120, 100))
```

ISPIS

```
True
False
True
```



- Ivica na ivicu nije sudar.
- collidepoint: tačka (x, y) unutra?

Da li se ova dva kvadrata sudaraju?

```
import pygame as pg

a = pg.Rect(0, 0, 50, 50)
b = pg.Rect(50, 0, 50, 50)
print(a.collidect(b))
```

Prvi kvadrat zauzima x od 0 do 49, a drugi počinje tačno na 50.

A

True

B

tačno

False

C

50

Dgreška:
TypeError

Sudari sprajtova i grupa

Jedan igrač, mnogo pahuljica

spritecollide vraća listu dodirnutih sprajtova

```
import pygame as pg

def sprajt(x, y, sirina, visina):
    s = pg.sprite.Sprite()
    s.rect = pg.Rect(x, y, sirina, visina)
    return s

igrac = sprajt(100, 100, 40, 64)
pahuljice = pg.sprite.Group(sprajt(110, 105, 24, 24),
                             sprajt(300, 50, 24, 24),
                             sprajt(90, 150, 24, 24))
uhvacene = pg.sprite.spritecollide(igrac, pahuljice, True)
print(len(uhvacene), len(pahuljice))
print(uhvacene[0].alive())
```

ISPIS

```
2 1
False
```

- dokill=True: dodirnite pahuljice nestaju iz svih grupa.
- Treći argument je obavezan: True ili False.

Tri funkcije, tri različita odgovora

```
import pygame as pg

def sprajt(x, y, sirina, visina):
    s = pg.sprite.Sprite()
    s.rect = pg.Rect(x, y, sirina, visina)
    return s

igrac = sprajt(0, 0, 40, 64)
grudve = pg.sprite.Group(sprajt(30, 50, 30, 30))
pahuljice = pg.sprite.Group(sprajt(45, 60, 24, 24))
print(pg.sprite.spritecollideany(igrac, grudve))
print(pg.sprite.spritecollideany(igrac, pahuljice))
sudari = pg.sprite.groupcollide(grudve, pahuljice,
                                False, False)
for grudva in sudari:
    print(len(sudari[grudva]))
```

ISPIS

```
<Sprite Sprite(in 1 groups)>
None
1
```

spritecollide

lista dodirnutih (može prazna)

spritecollideany

jedan sprajt ili None

groupcollide

rečnik: sprajt → lista

Šta ispisuje ovaj program?

```
import pygame as pg

def sprajt(x, y, sirina, visina):
    s = pg.sprite.Sprite()
    s.rect = pg.Rect(x, y, sirina, visina)
    return s

igrac = sprajt(0, 0, 40, 64)
grudve = pg.sprite.Group(sprajt(300, 0, 30, 30))
pogodak = pg.sprite.spritecollideany(igrac, grudve)
print(pogodak)
```

Igrač je kod $x = 0$, a jedina grudva kod $x = 300$.

A

False

B

tačno

None

C

[]

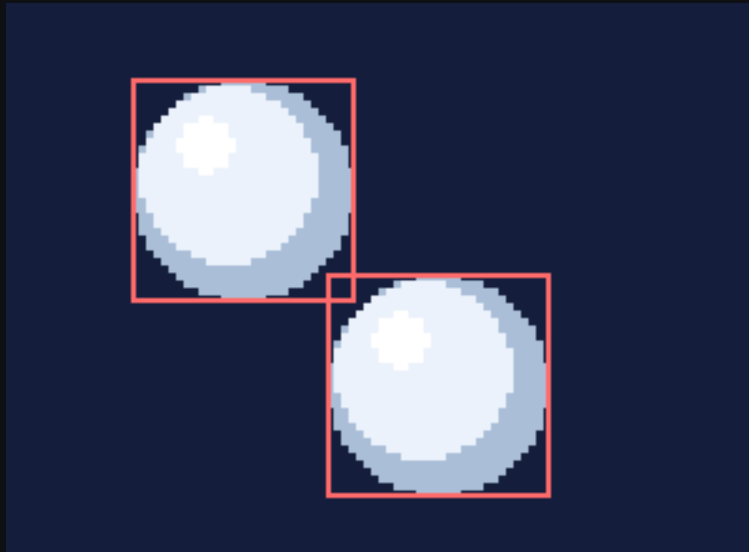
D

0

Krug i maska

Kad pravougaonik nije dovoljno tačan

Čudovište: lažni sudar



dodiruju se samo providni uglovi

Kako izgleda

Grudva prođe pored lika, a igra javi udarac. Igrač kaže: „Nije me ni dotakla!“

Zašto se pojavljuje

Sudar se meri pravougaonicima, a slike imaju providne uglove koji se ne vide.

Kako se pobeđuje

`collide_circle` sa atributom `radius` za okrugle predmete, `collide_mask` za sve ostale.

Krug bez poluprečnika je još veći od okvira

```
import pygame as pg

slika = pg.image.load("grudva.png")
a = pg.sprite.Sprite()
a.image = slika
a.rect = slika.get_rect(topleft=(0, 0))
b = pg.sprite.Sprite()
b.image = slika
b.rect = slika.get_rect(topleft=(26, 26))
print(pg.sprite.collide_rect(a, b))
print(pg.sprite.collide_circle(a, b))
a.radius = 15
b.radius = 15
print(pg.sprite.collide_circle(a, b))
print(pg.sprite.collide_mask(a, b))
```

ISPIS

True	← pravougaonici
True	← krugovi bez radius
False	← krugovi, radius = 15
None	← maske

Bez atributa radius krug obuhvata ceo pravougaonik: poluprečnik je pola dijagonale, oko 21 za grudvu 30 × 30.

Okrugloj prepreci damo poluprečnik

```
class Prepreka(pg.sprite.Sprite):  
    def __init__(self):  
        super().__init__()  
        self.image = SLIKA_GRUDVE          # 30 × 30  
        self.rect = self.image.get_rect()  
        self.radius = 15                    # polovina širine  
  
# u petlji: četvrti argument kaže kako se meri sudar  
pogodjene = pg.sprite.spritecollide(  
    igrac, prepreke, True, pg.sprite.collide_circle)
```

self.radius

polovina širine slike

collide_circle

meri rastojanje centara

četvrti argument

funkcija koja meri sudar

Maska pamti koji pikseli nisu providni

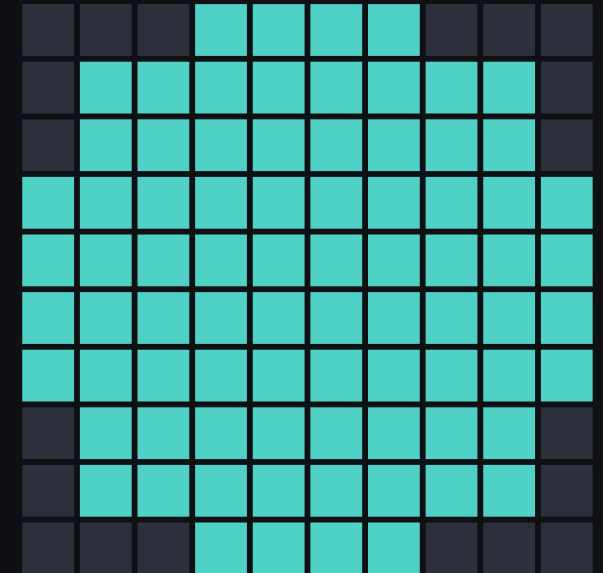
```
import pygame as pg

slika = pg.Surface((10, 10), pg.SRCALPHA)
pg.draw.circle(slika, "white", (5, 5), 5)
maska = pg.mask.from_surface(slika)
print(maska.get_size())
print(maska.count())
print(maska.get_at((5, 5)), maska.get_at((0, 0)))
```

ISPIS

```
(10, 10)
76
1 0
```

MASKA KRUGA 10 × 10



tirkizno: 1 (neprovidno), sivo: 0

`pg.mask.from_surface(slika)` pravi masku iz slike. Pravimo je jednom, u `__init__`.

Pahuljica dobija masku, sudar ide po pikselima

```
class Pahuljica(pg.sprite.Sprite):  
    def __init__(self):  
        super().__init__()  
        self.image = SLIKA_PAHULJICE  
        self.rect = self.image.get_rect()  
        self.mask = pg.mask.from_surface(self.image)  
  
uhvacene = pg.sprite.spritecollide(  
    igrac, pahuljice, True, pg.sprite.collide_mask)
```

`self.mask`

`collide_mask` je koristi; inače pravi novu u svakom frejmu

`collide_mask`

sudar samo ako se dodiruju neprovidni pikseli

Šta ispisuje ovaj program?

```
import pygame as pg

def sprajt(x, y, sirina, visina):
    s = pg.sprite.Sprite()
    s.rect = pg.Rect(x, y, sirina, visina)
    return s

a = sprajt(0, 0, 20, 20)
b = sprajt(22, 0, 20, 20) # bez atributa radius
print(pg.sprite.collide_rect(a, b),
      pg.sprite.collide_circle(a, b))
```

Između dva kvadrata 20×20 je razmak od 2 piksela.

A

True True

B

False False

C

tačno

False True

D

True False

Koji sudar kada: brzo, okruglo ili tačno

pravougaonici

`collide_rect`

Dugmad, zidovi, platforme,
sve što je stvarno
pravougaono.

najbrži

krugovi

`collide_circle`

Lopte, grudve, novčići.
Obavezno postavite
`self.radius`.

brz

maske

`collide_mask`

Likovi i predmeti
nepravilnog oblika: Čikica,
pahuljica.

najtačniji, najsporiji

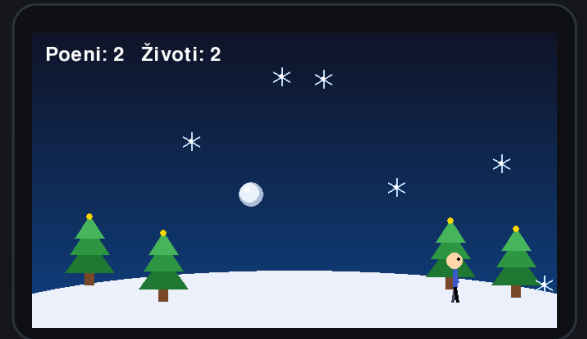
Pravila igre

Poeni, životi i kraj

Rezultat je tekst koji crtamo u svakom frejmu

```
font = pg.font.Font(None, 36)    # jednom, pre petlje  
poeni = 0  
...  
# u petlji, posle crtanja svega ostalog:  
tekst = font.render(f"Poeni: {poeni}", True, "white")  
ekran.blit(tekst, (16, 12))
```

font.render pravi sliku od niske; sliku zatim postavimo sa blit, kao svaku drugu. Font je skup za pravljenje, pa ga pravimo samo jednom.



Tekst se crta poslednji, da bude iznad svega.

Prepreka: grudva koja pada i nestaje

```
class Prepreka(pg.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = SLIKA_GRUDVE
        self.mask = pg.mask.from_surface(self.image)
        x = random.uniform(20, SIRINA - 20)
        self.polozaj = pg.Vector2(x, -30)
        self.rect = self.image.get_rect(center=self.polozaj)
        self.brzina = random.uniform(150, 230) # px/s

    def update(self, dt):
        self.polozaj.y += self.brzina * dt
        self.rect.center = self.polozaj
        if self.rect.top > VISINA:
            self.kill()
```

Vector2

realan položaj; rect ga prati

random.uniform

svaka grudva svojom brzinom

self.kill()

ispala je dole: izbacij je iz grupe

Posle udarca igrač trepće i ne gubi život

```
def udaren(self):  
    self.zivoti -= 1  
    self.neranjiv = 1.5          # sekunde zaštite  
  
# na kraju metode update(self, dt):  
    if self.neranjiv > 0:  
        self.neranjiv -= dt  
        if int(self.neranjiv * 10) % 2 == 1:  
            self.image = PRAZNA # svaka druga desetinka
```

`self.neranjiv`

preostale sekunde zaštite

PRAZNA

potpuno providna slika 40 × 64

`% 2`

smenjuje se svake desetinke

`int(neranjiv * 10) % 2` menja se svake desetinke

```
for neranjiv in [1.5, 1.42, 1.31, 1.25, 1.18]:  
    vidi_se = int(neranjiv * 10) % 2 == 0  
    print(neranjiv, vidi_se)
```

ISPIS

```
1.5 False  
1.42 True  
1.31 False  
1.25 True  
1.18 False
```

- `int` odseca decimale: $1,42 \times 10 = 14,2$ postaje 14.
- Paran broj desetinki: vidi se; neparan: ne vidi se.
- Isto pravilo radi pri svakom FPS, jer zavisi od vremena, a ne od frejmova.

Pravila igre su tri provere u svakom frejmu

```
uhvacene = pg.sprite.spritecollide(  
    igrac, pahuljice, True, pg.sprite.collide_mask)  
poeni += len(uhvacene)  
  
if igrac.neranjiv <= 0:  
    if pg.sprite.spritecollide(  
        igrac, prepreke, True, pg.sprite.collide_mask):  
        igrac.udaren()  
  
if igrac.zivoti == 0:  
    kraj = True
```

`len(uhvacene)`

koliko pahuljica u ovom frejmu

`neranjiv <= 0`

život se gubi samo bez zaštite

`kraj = True`

igra staje, rezultat ostaje

Koliko života ostaje?

```
zivoti = 3
neranjiv = 0
for udar in [True, True, False, True]:
    neranjiv -= 0.5
    if udar and neranjiv <= 0:
        zivoti -= 1
        neranjiv = 1
print(zivoti)
```

Svaki prolaz je frejm od pola sekunde.

A

0

B

tačno

1

C

2

D

3

Zimska šetnja, korak 5: poeni i životi

Cilj: igrač hvata pahuljice za poene, a grudve mu oduzimaju živote.

- 1 Maske za Čikicu i pahuljicu
- 2 `spritecollide`: uhvaćena pahuljica je poen
- 3 Klasa `Prepreka` i tajmer za grudve
- 4 `udaren()`: život manje, 1,5 s zaštite
- 5 Tekst „Poeni“ i „Životi“, kraj na nuli

KAKO IZGLEDA NA KRAJU

Poeni: 2 Životi: 2



Sličice pravi program 10_01; ceo kod je u 10_09.

Pronađi grešku: jedna grudva odnese sve živote

```
import pygame as pg

def sprajt(x, y, sirina, visina):
    s = pg.sprite.Sprite()
    s.rect = pg.Rect(x, y, sirina, visina)
    return s

igrac = sprajt(100, 100, 40, 64)
prepreke = pg.sprite.Group(sprajt(110, 90, 30, 30))
zivoti = 3
for frejm in range(3):          # tri frejma igre
    if pg.sprite.spritecollide(igrac, prepreke, False):
        zivoti -= 1
    print("frejm", frejm, "životi:", zivoti)
```

ISPIS

```
frejm 0 životi: 2
frejm 1 životi: 1
frejm 2 životi: 0
```

False

grudva ostaje u grupi

bez zaštite

udarac u svakom frejmu

Izaberite nivo i počnite

LEVEL 1

Klikni pahuljicu

Pahuljice padaju, a klik mišem na pahuljicu je hvata i daje poen.

Pomoć: MOUSEBUTTONDOWN, dogadjaj.pos i collidepoint.

LEVEL 2

Srca i kraj igre

Životi su srca gore desno. Na nuli scena potamni i piše „Kraj igre“.

Pomoć: srce.png i petlja for po broju života.

LEVEL 3

Grudve i sneg

Grudve koje padaju uništavaju pahuljice koje dodirnu. Brojite uništene.

Pomoć: groupcollide i collide_circle.

Sledeći put uhvaćena pahuljica zazvoni

```
DZING = pg.mixer.Sound("dzing.wav")  
...  
for p in uhvacene:  
    DZING.play()  
    rasprsni(p.rect.center, cestice)
```

U temi 11 hvatanje pahuljice prati zvuk „dzing“, a pahuljica se rasprsne u dvadeset tačkica koje blede i padaju. U pozadini svira muzika, a pahuljice se okreću dok padaju.

Rezime

pravougaonici

```
a.colliderect(b)
r.collidepoint(x, y)
```

Ivica na ivicu nije sudar.

sprajt i grupa

```
spritecollide(s, g, True)
```

Lista; dokill izbacuje iz grupe.

krug i maska

```
self.radius = 15
self.mask = from_surface(...)
```

Tačnije od pravougaonika.

pravila igre

```
poeni += len(uhvacene)
neranjiv, zivoti, kraj
```

Zaštita posle udarca.

Pitanje pre odlaska

```
uhvacene = pg.sprite.spritecollide(  
    igrac, pahuljice, True)
```

U grupi pahuljice je 20 pahuljica, a igrač dodiruje tri. Koliko pahuljica ostaje u grupi posle ovog reda?

Pošaljite jedan broj pre nego što izađete.



Literatura

- Dokumentacija za pygame-ce, moduli sprite i mask i klasa Rect: pyga.me/docs
- Sweigart, A. Invent Your Own Computer Games with Python, 4. izdanje, No Starch Press, poglavlje 19 o sudarima (besplatno na inventwithpython.com)
- Kafle, S. Learning Python by Building Games, Packt, 2019 (poglavlja o PyGame-u)
- MDN Web Docs: 2D collision detection (sudari pravougaonika i krugova, sa primerima koji se pokreću u pregledaču)

Dodatna literatura i zanimljivi sajtovi

pygame-ce: modul sprite

pyga.me/docs/ref/sprite.html

spritecollide, spritecollideany, groupcollide i funkcije collide_*.

pygame-ce: klasa Rect

pyga.me/docs/ref/rect.html

colliderect, collidepoint i sve što pravougaonik ume.

pygame-ce: modul mask

pyga.me/docs/ref/mask.html

Maske: from_surface, count, overlap, outline.

MDN: 2D collision detection

[developer.mozilla.org · 2D collision detection](https://developer.mozilla.org/en-US/docs/Games/Techniques/2D_collision_detection)

Sudari pravougaonika i krugova, sa primerima koji se pokreću u pregledaču.

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