

Programiranje za umetnike 1

~ Pygame II ~

Staša Vujičić Stanković

Teme

Biblioteka Pygame i
njene primene

animacije i događaji

Animacija

- Animacija se ostvaruje u petlji
 - zasnovanoj na događajima – telo petlje se izvršava po jednom za svaki događaj koji nastupi, čim događaj nastupi.
 - zasnovanoj na frejmovima – između svaka dva frejma se čeka određen broj milisekundi ili se uvodi tajmer koji generiše događaje u pravilnim vremenskim intervalima i na svaki „otkucaj“ tajmera se prikazuje naredni frejm.

„Broj“ kadrova

* frame rate *

- Broj kadrova ili frekvencije kadrova (engl. *frame rate*) je termin koji se često koristi kao merilo za merenje performanse igrice.
- Brzina kadrova u video igrici doprinosi kvalitetu rezultujućih kretanja animacije, kao i meri koliko puta su slike koje se posmatraju na ekranu osvežene ili preuzete.
- Brzina kadrova je merenje koje se vrši u frejmovima u sekundi (engl. *frames per second* – FPS)

Animacija zasnovana na frejmovima

čekanje određen broj milisekundi između dva frejma

```
kraj = False
while not(kraj):
    # vrši se crtanje
    ...

    # proverava se da li je korisnik isključio prozor
    for dogadjaj in p.event.get():
        if dogadjaj.type == p.QUIT:
            kraj = True

    # čeka se određeno vreme zadato brojem milisekundi
    p.time.wait(broj_milisekundi)

    # promena podataka koji opisuju stanje objekata na slici
    ...
```

Animacija zasnovana na frejmovima

korišćenje sata

```
sat = p.Clock()
kraj = False
while not(kraj):
    # vrsi se crtanje
    ...

    # proverava se da li je korisnik isključio prozor
    for dogadjaj in p.event.get():
        if dogadjaj.type == p.QUIT:
            kraj = True

    # ogranicava se izvršavanje tela petlje
    # na najviše 20 puta u sekundi
    sat.tick(20)

    # promena podataka koji opisuju stanje objekata na slici
    ...
```

Animacija zasnovana na frejmovima

korišćenje sata

- `Pygame.time.Clock.tick()` treba da se poziva po jednom za frejm.
- Između dva uzastopna poziva metoda `tick()` prati vreme između poziva u milisekundama.
- Na primer, pozivanjem `Clock.tick(20)` jednom po frejmu, program je ograničen na pokretanje u granici do 20 FPS i ne može je prekoračiti, čak i ako je procesorska snaga računara veća. Dakle, to može se koristiti za ograničavanje brzine izvođenja igre. Ovo je važno jer igra treba da „teče“ glatko.
- Smatra se da je u redu svaka vrednost frejmova u sekundi (frekvencija kadrova) između 15 i 40 u Pygame igricama.

Događaji

- Pygame programi vrše interakciju sa korisnikom korišćenjem događaja.
- Svaki put kada korisnik pomeri miša, pritisne neko dugme miša ili taster na tastaturi sistem to beleži u obliku događaja koji se registruje u programu smeštanjem u red događaja na šta se može odreagovati izvršavanjem nekog programskog koda, takozvanom obradom događaja.

Događaji

- `pg.event.wait()` – funkcija kojom se zaustavlja izvršavanje programa dok ne nastupi neki događaj.
- Pomoću ove funkcije se telo glavne petlje izvršava po jednom za svaki događaj koji se registruje. To su takozvane petlje vođene događajima.

```
while p.event.wait().type != p.QUIT:  
    pass
```

Interakcija korisnika upotrebom miša i tastature

- `p.mouse.get_pos()` – vraća uređeni par koordinata na kojoj se trenutno nalazi pokazivač miša.
- `p.mouse.get_pressed()` – vraća uređenu trojku logičkih elemenata koji odgovaraju redom levom, srednjem i desnom tasteru miša.
True – označava da je taster pritisnut, False – označava da taster nije pritisnut.
- `p.key.get_pressed()` - vraća torku elemenata čiji se elementi koriste kao logičke vrednosti koje pokazuju da li je određeni taster pritisnut ili nije.
Da bi im se pristupilo koriste se imenovane konstante. Na primer:
 - `p.K_LEFT`, `p.K_RIGHT`, `p.K_UP`, `p.K_DOWN` odgovaraju strelicama levo, desno, gore, dole
 - `p.K_SPACE` odgovara razmaku
 - `p.K_a`, `p.K_b`, `p.K_c`... odgovaraju slovima a, b, c.

Literatura

- Kafle, Sachin. Learning Python by building games: a beginner's guide to Python programming and game development. 2019.
(posebno značajni odeljci 11, 12 i 13)

[GitHub repozitorijum sa kodovima](#)

- Wentworth, Peter, et al. [How to Think Like a Computer Scientist](#). Free online book.
- Sweigart, Albert. Invent Your Own Computer Games with Python, 4E. No Starch Press, 2016.

Hvala



Staša Vujičić Stanković



stasa@math.rs



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