easygame

Inštalácia

1. Ak nemáte, nainštalujte si Python 3.
2. Nainštalujte si knižnicu `pyglet` (ktorú `easygame` používa pod kapotou). To viete spraviť v príkazovom riadku pomocou:
   ```
   python3 -m pip install pyglet
   ```
3. Skopírujte súbor `easygame.py` do priečinku ku svojej hre.
4. Importujte knižnicu pomocou `from easygame import *`.

S akýmikoľvek problémami s inštaláciou alebo používaním knižnice sa na nás obráťte na Discorde.

Základná kostra hry

```
from easygame import *

open_window('easygame ftw', 800, 600)

should_quit = False
while not should_quit:
    for event in poll_events():
        if type(event) is CloseEvent:
            should_quit = True

    # po načítaní udalostí vykonajte logiku hry
    # a nakreslite veci na obrazovku
    # (kompletne ju prekreslite každý frame)

    next_frame()

close_window()
```

Podrobný popis funkcií

degrees(*d***)**

Prepočíta stupne na radiány. Všetky funkcie akceptujú radiány.

Napríklad: `set_camera(rotation=degrees(30))`

rotate(*vector, angle***)**

Vráti vektor zadaný ako `(x, y)` otočený o zadaný uhol v radiánoch.

open_window(*title, width, height, fps=60***)**

Otvorí okno so zadaným titulkom, šírkou a výškou v pixeloch.

Túto funkciu musíte zavolať predtým než začnete čokoľvek kresliť.

Nepovinný argument `fps` určuje počet snímkov (framov) za sekundu.

Funkcia `next_frame()` sa riadi touto frekvenciou.

close_window()

Zavrie okno. Toto zavolajte pri ukončení programu.

poll_events()

Vráti zoznam udalostí, ktoré sa stali od posledného volania tejto funkcie.

Môžete cez ne prejsť takto: `for event in poll_events(): …`

Je sedem druhov udalostí:

`CloseEvent`,
`KeyDownEvent`, `KeyUpEvent`, `TextEvent`,
`MouseMoveEvent`, `MouseDownEvent`, `MouseUpEvent`.

Druh udalosti môžete zistiť pomocou funkcie `type`, napríklad takto:

```
if type(event) is KeyDownEvent: …
```

Udalosť `CloseEvent` nemá žiadne dáta.

Udalosti `KeyDownEvent` a `KeyUpEvent` majú položku `.key`, ktorá je jedno z:

```
'A' ... 'Z'
'0' ... '9'
'SPACE', 'ENTER', 'BACKSPACE', 'ESCAPE'
'LEFT', 'RIGHT', 'UP, 'DOWN'.
```

Udalosť `TextEvent` má jednu položku `.text`, ktorá obsahuje string,
čo užívateľ napísal. Zvyčajne to je len jeden znak, keďže viac sa nestihne napísať za jeden frame.

Udalosti `MouseDownEvent` a `MouseUpEvent` majú položky `.x`, `.y` a `.button`.
Položky `.x` a `.y` sú aktúálna pozícia myši na obrazovke.
Položka `.button` je jedno z:

```
'LEFT', 'RIGHT', 'MIDDLE'.
```

Udalosť `MouseMoveEvent` má položky `.x`, `.y`, `.dx`, `.dy`.
Položky `.x` a `.y` sú aktuálna pozícia myši na obrazovke.
Položky `.dx` a `.dy` sú rozdiel oproti predchádzajúcej pozícií.

next_frame()

Zobrazí nakreslené vec na obrazovke a počka kým je čas na ďalší frame.

fill(*r, g, b***)**

Vyplní celú obrazovku jednou farbou v RGB.

load_image(*path***)**

Načíta a vráti obrázok. Ak sa cesta nezačína znakom `/` alebo `C:\`
berie sa ako relatívna k ceste programu. Napríklad:

```
sheet = load_image("sheep.png")
```

načíta obrázok `sheep.png` z priečinka programu.

load_sheet(*path, frame_width, frame_height***)**

Načíta obrázok, rozdelí ho na rovnomerné diely o zadanej veľkosti
a vráti zoznam týchto dielov ako individuálne obrázky.

image_data(*image***)**

Vráti zoznam RGBA hodnôt jednotlivých pixelov vo formáte

```
[(r,g,b,a), (r,g,b,a), ...]
```

Jednotlivé pixely sú najprv podľa stĺpcov, potom podľa riadkov. Ak W je šírka obrázka, tak prvých W hodnôt zodpovedá prvému riadku zľava doprava, ďalších W hodnôt zodpovedá druhému riadku, a tak ďalej.

draw_image(*image, position=(0,0), anchor=None, rotation=0, scale=1, scale_x=1, scale_y=1, opacity=1, pixelated=False***)**

Nakreslí obrázok na obrazovku. Riadi sa kamerou.
`position` je pozícia obrázka v svete (bod jeho ukotvenia bude v tejto pozícií).
`anchor` sú súradnice ukotvenia obrázka vzhľadom na ľavý dolný roh obrázka.
Ak ich nezadáte, bude ukotvený za stred.
`rotation` otočí obrázok okolo ukotvenia o uhol v radiánoch.
`scale` zväčší/zmenší obrázok okolo ukotvenia.
`scale_x` dodatočne zväčší/zmenší obrázok okolo ukotvenia po jeho X-ovej osi.
`scale_y` dodatočne zväčší/zmenší obrázok okolo ukotvenia po jeho Y-ovej osi.
`opacity` spriehľadní obrázok. Hodnota `0` je úplna prehľadnosť, 1 nepriehľadnosť.
`pixelated` ak je nastavený na `True`, tak pri zväčšovaní sa obrázok nebude vyhladzovať.

draw_polygon(**points, color=(1,1,1,1)***)**

Nakreslí konvexný mnohouholník jednej farby. Riadi sa kamerou.
Zoznam bodov je dávaný variadicky:

```
draw_polygon((0,0), (100,0), (50,100), color=(1,0,1,1))
```

Farbu treba zadať explicitne pomocou `color=` ako posledný argument.
Farba má štyri komponenty: RGBA. A je alpha, priehľadnosť.

draw_line(**points, thickness=1, color=(1,1,1,1)***)**

Nakreslí lomenú čiaru medzi zadanými bodmi s danou hrúbkou a RGBA farbou.
Riadi sa kamerou.

draw_circle(*center, radius, color=(1, 1, 1, 1)***)**

Nakreslí kruh s daným stredom, polomerom a RGBA farbou. Riadi sa kamerou.

draw_text(*text, font, size, position=(0, 0), color=(1,1,1,1), bold=False, italic=False***)**

Nakreslí text s danou pozíciou ľavého dolného rohu. Napríklad:

```
draw_text('Hey!', 'Times New Roman', 32, position=(20,20))
```

Ďalšie argumenty určujú farbu a formát textu.

set_camera(*center=None, position=None, rotation=None, zoom=None***)**

Nastaví niektoré (alebo všetky) parametre kamery na nové hodnoty.
Vynechané parametre ostanú nezmenené.
Parametre kamery sú:
`center` určuje kde na obrazovke sa bude nachádzať aktuálna pozícia kamery.
Toto sa zvyčajne nastaví raz, napríklad na stred obrazovky.

Meniť má zmysel napríklad pre efekt trasenia obrazovky.
Má tvar `(x,y)`.

`position` určuje pozíciu vo svete kam sa kamera práve pozerá v tvare `(x,y)`.
`rotation` otočí kameru okolo o uhol v radiánoch.
`zoom` priblíži/oddiali kameru.

move_camera(*position=None, rotation=None, zoom=None***)**

Zmení niektoré (alebo všetky) parametre kamery vzhľadom na ich pôvodné hodnoty.
K `position` a `rotation` pripočíta, `zoom` prenásobí.

save_camera()

Uloží aktuálne nastavenia kamery na stack. (Ale nijako ich nezmení.)
Toto je užitočné pokiaľ chcete niektoré veci nakresliť s jednou kamerou a iné s inou.

restore_camera()

Obnoví naposledy uložené a ešte neobnovené nastavenia kamery zo stacku.

reset_camera()

Prepíše aktuálne nastavenia kamery na pôvodné.

load_audio(*path, streaming=False***)**

Načíta a vráti obsah audio súboru. Napríklad:

```
shot = load_audio('gunshot.wav')
```

Ak je nepovinný argument `streaming` nastavený na `True`, súbor nebude načítaný do pamäte, ale bude postupne ťahaný z disku.

play_audio(*audio, channel=0, loop=False, volume=1***)**

Spustí audio na nejakom kanáli (predvolený 0).
Je nekonečno kanálov.
Spustenie zvuku na kanáli automaticky zastaví zvuk, čo tam hral dovtedy (ak hral).
Pre zastavenie hrania na kanáli (napríklad 7) napíšte:

```
play_audio(None, channel=7).
```

Ak je argument `loop` nastavený na `True`, zvuk bude hrať dokola.
Argument `volume` zintenzivní/zoslabí zvuk daným faktorom.

fix_rectangle_overlap(*rect1, rect2***)**

Dostane dva obdĺžniky tvaru `(x0,y0,x1,y1)`.
Vráti najmenší vektor tvaru `(x,y)` o ktorý musíme posunúť prvý obdĺžnik aby sa neprekrýval s druhým.

Cheatsheet

degrees(*d***)**
rotate(*vector, angle***)**
open_window(*title, width, height, fps=60***)**
close_window()
poll_events()
next_frame()
fill(*r, g, b***)**
load_image(*path***)**
load_sheet(*path, frame_width, frame_height***)**
image_data(*image***)**
draw_image(*image, position=(0,0), anchor=None, rotation=0, scale=1, scale_x=1, scale_y=1, opacity=1, pixelated=False***)**
draw_polygon(**points, color=(1,1,1,1)***)**
draw_line(**points, thickness=1, color=(1,1,1,1)***)**
draw_circle(*center, radius, color=(1, 1, 1, 1)***)**
draw_text(*text, font, size, position=(0, 0), color=(1,1,1,1), bold=False, italic=False***)**
set_camera(*center=None, position=None, rotation=None, zoom=None***)**
move_camera(*position=None, rotation=None, zoom=None***)**
save_camera()
restore_camera()
reset_camera()
load_audio(*path, streaming=False***)**
play_audio(*audio, channel=0, loop=False, volume=1***)**
fix_rectangle_overlap(*rect1, rect2***)**