

5.5 Qo'llarni cho'zgan holda sakrashlar sonini sanang - O'zgaruvchi

Entry blokli dasturlashda biz ko'pincha obyektlar animatsiyasidan foydalanganmiz. Pygame Zero-da ham xuddi shunday qilish mumkinmi? Albatta, mumkin. O'yinlardagi animatsiya odatiy holdir. Ushbu misolda biz birinchi marta Actor obyektlari animatsiyasidan foydalanamiz.

```
{% code lineNumbers="true" %}
```

```
from pgzhelper import *

WIDTH = 480
HEIGHT = 270

boy = Actor('boy_1', (WIDTH / 2, HEIGHT / 2))
boy.images = ['boy_1', 'boy_2', 'boy_3']

times = 0
pressed = False

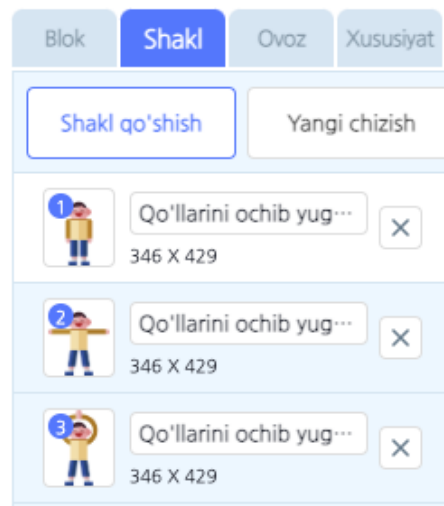
def draw():
    screen.fill('white')
    boy.draw()
    game.time.sleep(0.5)
    screen.draw.text('Times: ' + str(times), (20, 20), color='black')

def update():
    global times, pressed
    if pressed:
        if boy.next_image() == 0:
            times += 1
            pressed = False

def on_mouse_down():
    global pressed
    pressed = True
```

```
{% endcode %}
```

7-qatorda **images** deb nomlangan obyekt xossasi paydo bo'ladi. Nomidan ko'rinib turibdiki, u bir nechta tasvirlarni ro'yxat formatida saqlaydi va uning maqsadi obyekt tasvirlarini jonlantirishdir. Ro'yxatdagi uchta rasm (boy_1, boy_2, boy_3) biz Entry-da ko'rgan sakrayotgan **boy** obyektining uchta tasviri (shakllari) bilan bir xil.



Animatsiya haqida gapirganimizdan kelib chiqib, avval tegishli kodni ko'rib chiqadigan bo'lsak, Entry-da animatsiya yaratish uchun uchta rasmni ketma-ket o'zgartirish orqali animatsiya yaratganimizni eslasangiz kerak. O'shanda rasm o'rtasidagi o'tishni amalga oshirish uchun "**Keyingi shaklga o'zgartirish**" degan blokdan foydalangan edik, va 21-qatorda ishlatilgan **next_image** funksiyasi aynan o'sha blok bilan mos keladigan metod (funksiya) hisoblanadi. Biroq, bu metod **Actor** obyektining tashqi ko'rinishini o'zgartirish bilan bog'liq bo'lgan ekran o'zgarishlari **update()** callback (qayta chaqirish) funksiyasi ichida bajarilishi kerakligiga e'tibor bering.

next_image usuli bajarilgandan so'ng, animatsiya yoqilgan image ro'yxatidagi rasmning indeksini (joylashuv qiymati) qaytariladi. 21-qatorda `boy.next_image() == 0` shart mavjud. Ushbu shart tasvirni almashtirishning to'liq tsikli tugaganligini va animatsiya birinchi rasmga qaytganligini tekshiradi (indeks 0 bilan). Shunday qilib, "qo'llarni ko'tarib sakrash" animatsiyasida uchta rasmning barchasi bajarilganligi tekshiriladi. Aynan shu vaqtda tugallangan animatsiya tsikllarining umumiy sonini hisoblash uchun eng yaxshi vaqt. Ushbu umumiy miqdor **9-qatorda e'lon qilingan times global o'zgaruvchisiga** yoziladi, va o'sha yerda u to'planadi.

10-qatorda e'lon qilingan global o'zgaruvchining maqsadini tushunasizmi? Uning maqsadi animatsiyani faqat sichqoncha tugmasi bosilganda amalga oshirishdir (faqat bosilgan qiymat 27-qatorda True bo'lganda). Shuning uchun, animatsiya tugagandan so'ng, uni 23-qatorda False holatiga qaytarish kerak. Shundan keyingina animatsiya yana sichqoncha tugmachasini bosganda davom etishi mumkin.

Nihoyat, 16-qatorda **text** deb ataladigan usul mavjud. Bu usul **screen** obyektining ichida joylashgan **draw** obyektiga tegishli bo'lgan usul bo'lib, u **screen.draw.text** kabi iyerarxik chaqirish tuzilmasiga ega. Ushbu usuldan foydalanish yo'li topshirilgan argument (o'zgaruvchilar) qiymatlaridan kelib chiqib osonlikcha tushuniladi, ya'ni ekranda qaysi joyda, qanday matn mazmunini va qanday rangda matn chiqarishni belgilash uchun qo'llaniladi. Bu yerda ko'rsatilayotgan qiymat, ya'ni **'Times: ' + str(times)** iborasi yangilik emas va ilgari **Entry-Python** kitobida bir necha bor qo'llangan. Bu yerda **' + '** operatori yordamida ikki matnli qiymatni (string qiymatlarni) birlashtirib, bitta matn hosil qilinadi. O'rtada ishlatilgan **str(times)** esa **str** deb nomlangan ichki funksiyadan foydalanilgan bo'lib, **times** o'zgaruvchisi raqamli qiymat bo'lgani uchun uni matnga qo'shishdan oldin **string** turiga o'zgartirish kerak bo'ladi.

Ma'lumot uchun, 15-qatordagi **game.time.sleep()** usuli(metodi) ushbu ilovaning amalga oshirilishi uchun majburiy bo'lmasa-da, qo'l-oyoqlarni keng yozish harakati animatsiyasining juda tez o'tib ketmasligi uchun qo'llanilgan. Bu usul orqali harakatlarni batafsilroq kuzatish imkoniyatini yaratish maqsadida, tasvir kadrlarining orasida 0.5 soniyalik kutish vaqti qo'shilgan.

Biz oldingi boblardan oldingi bilimlarni to'plaganimiz sababli, biz yangi kiritilgan narsalarni tezda o'rganishga muvaffaq bo'ldik. Nihoyat, biz ushbu misolning bajarilishi natijalarini ko'rib chiqish orqali ushbu bobni yakunlaymiz.

