Анимације

Вероватно већ знаш да цртани филм настаје тако што се на екрану брзо смењују сличице, при чему је свака следећа слика веома слична претходној (ликови на слици су само мало померени у односу на претходни положај).

На пример, од наредних осам сличица лика у различитим положајима:

       

настаје следећа анимација трчања:



Анимације подразумевају брзу промену слике на екрану (на пример, 20 пута у секунди), обично у правилним временским интервалима (на пример, на сваких 50 милисекунди). Свака тако кратко приказана слика назива се *оквир* или *фрејм* анимације (енгл. *frame*). У овом делу приручника видећемо како можемо направити програме у којима се приказују неке анимације.

У програмима које смо до сада сретали слика се није мењала током извршавања и цртање смо вршили само једном, пре главне петље програма у којој смо чекали да корисник искључи прозор (у програмима заснованим на библиотеци PyGameBg, та петља се остварује позивом функције pygamebg.wait\_loop).

У програмима са анимацијом цртање ћемо вршити обично унутар тела те главне петље или, још боље, у засебној функцији коју ћемо на том месту позивати (у програмима заснованим на библиотеци PyGameBg, таква петља се остварује позивом функције pygamebg.frame\_loop).

Програмирање анимација на сајту „Петља”

Све анимације које ћемо у наставку приказати разликоваће се само по томе који подаци одређују оно што се на слици налази, коду који извршава цртање и коду који мења податке када се прелази на наредни фрејм. Да бисмо ти олакшали сналажење са кодом који већ постаје дугачак и компликован, цртање и прелазак са тренутног на наредни фрејм ћемо издвојити у две помоћне функције које ти треба да напишеш, док ћемо главну петљу програма која те две функције позива ми писати уместо тебе (и она ће бити у „сивом делу кода”). Осим те две функције твој задатак ће бити и да дефинишеш променљиве које описују оно што се налази на екрану током анимације. То ће бити обично променљиве које описују положај (координате) објеката тј. ликова који се током анимације померају, њихову брзину, али и неки други подаци који се мењају током анимације.

Постоји неколико начина да се анимације реализују и у зависности од тога „сиви кôд” може бити реализован на различите начине. Ако програмираш анимације само на сајту „Петља”, тада не мораш уопште да читаш и анализираш сиви кôд (он ће увек бити унапред припремљен за тебе). Ако желиш да правиш програме са анимацијама и ван овог сајта, тада ипак треба да разумеш како сиви кôд функционише. За то ти онда препоручујемо да прочиташ [овај текст](https://petlja.org/biblioteka/r/lekcije/pygame-prirucnik/animacije-cas11_nacinianimacije).

Прикажимо сада кроз неколико примера технику коју ће вам олакшати прављење анимација у примерима који следе. Сви ће бити засновани на библиотеци PyGameBg, тако да ће и „сиви кôд” бити прилично једноставан.

Посебна функција за цртање у програму без анимације

За почетак прикажимо како се цртање може издвојити у посебну функцију (и то у прво у програму без анимација, а затим у програму са анимацијама). Кренимо од програма који смо раније већ срели, који црта три круга у разним бојама.



1

import pygame as pg

2

import pygamebg

3

​

4

# otvaramo prozor

5

(sirina, visina) = (200, 200)

6

prozor = pygamebg.open\_window(sirina, visina, "Три круга")

7

​

8

​

9

prozor.fill(pg.Color("white")) # bojimo pozadinu u belo

10

(cx, cy) = (sirina // 2, visina // 2) # centar krugova je u sredni ekrana

11

pg.draw.circle(prozor, pg.Color("red"), (cx, cy), 100) # crveni krug

12

pg.draw.circle(prozor, pg.Color("blue"), (cx, cy), 75) # plavi krug

13

pg.draw.circle(prozor, pg.Color("green"), (cx, cy), 50) # zeleni krug

14

​

15

​

16

# prikazujemo prozor i čekamo da ga korisnik isključi

17

pygamebg.wait\_loop()

18

​

(tri\_kruga)

Цртање кругова је део главног програма. Исти ефекат можемо постићи ако дефинишемо функцију crtaj коју ћемо позвати из главног програма.



1

import pygame as pg

2

import pygamebg

3

​

4

(sirina, visina) = (200, 200) # otvaramo prozor

5

prozor = pygamebg.open\_window(sirina, visina, "Три круга")

6

​

7

​

8

​

9

def crtaj():

10

 prozor.fill(pg.Color("white")) # bojimo pozadinu u belo

11

 (cx, cy) = (sirina // 2, visina // 2) # centar krugova je u sredni ekrana

12

 pg.draw.circle(prozor, pg.Color("red"), (cx, cy), 100) # crveni krug

13

 pg.draw.circle(prozor, pg.Color("blue"), (cx, cy), 75) # plavi krug

14

 pg.draw.circle(prozor, pg.Color("green"), (cx, cy), 50) # zeleni krug

15

​

16

​

17

# pozivamo funkciju za crtanje

18

crtaj()

19

​

20

# prikazujemo prozor i čekamo da ga korisnik isključi

pygamebg.wait\_loop()​

(tri\_kruga\_funkcija)

Насумично одређивање боје позадине током анимације

Наредни програм ће приказивати једноставну анимацију у којој ћемо четири пута у секунди на насумичан начин одређивати боју позадине. Одређивање насумичне боје вршићемо помоћу посебне функције *nasumicna\_boja*. Дефинисаћемо функцију *crtaj* у којој ће се насумично одређивати боја и затим ће се позадина прозора бојити том бојом. Пошто ћемо користити библиотеку *PyGameBg*, анимацију ћемо на крају програма покретати позивом *pygamebg.frame\_loop(4, crtaj)*, чиме ћемо постићи да се функција *crtaj* аутоматски изнова позива 4 пута у секунди.



1

import random

2

import pygame as pg

3

import pygamebg

4

​

5

(sirina, visina) = (200, 200)

6

prozor = pygamebg.open\_window(sirina, visina, "Боја позадине")

7

​

8

​

9

def nasumicna\_boja():

10

 return (random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))

11

​

12

def crtaj():

13

 boja = nasumicna\_boja() # nasumično određujemo boju pozadine

14

 prozor.fill(boja) # bojimo pozadinu prozora

15

16

​

# pokrećemo animaciju tako što funkciju crtaj pozivamo 4 puta u sekundi

pygamebg.frame\_loop(4, crtaj)​