
IGRA PONG - PYTHON

Tehnička dokumentacija

Izradio:

Mario Matić

Rijeka, 8.5.2020.

Sadržaj:

Broj stranice:

1.Uvod	3
O igri	
Ideja	
2.Opis rada	4
Pravila igre	
Opis igre	
Izrada igre	
Cijeli kod	
3.Zaključak	7

1. UVOD

O IGRI

Igra Pong je jednostavna 2D igra izvorno razvijena od američkog studija Atari 1972. godine. Bila je jedna od prvih arkadnih igara, a njezin kreator je Allan Alcorn kojemu je to bio zadatak zadan od suvlasnika tvrtke Atari, Nolana Bushnella. Pong je jedna od prvih igara koja je uspješno ostvarila globalan uspjeh te tako postavila temelje video gaming industriji. S obzirom da u to vrijeme zakon o autorskim pravima nije bio jasno definiran, mnoge su druge kompanije željne uspjeha kopirale igru te ju predstavile kao svoju, što je kasnije rezultiralo velikim brojem većinom loših kopija na tržištu. Iako, s druge strane, kopije su omogućile eksponencijalan rast prodanih primjeraka igre te tako uveli igranje video igara u mnoga kućanstva.

IDEJA

S obzirom da je ovaj projekt prvi moj samostalni projekt, odlučio sam se koristiti programski jezik Python jer sam se njim već donekle upoznao kroz srednjoškolsko obrazovanje. Koristio sam Pygame modul jer on uvelike olakšava izradu jednostavnih video igara. Pygame nisam prije koristio, tako da sam za prvi projekt odabrao jednostavnu igru – Pong. Ideju sam dobio tako što sam se sjetio igre slične Pongu koju sam u djetinjstvu igrao na starom preklopnom mobitelu.

2. OPIS RADA

PRAVILA IGRE

Pravila igre su vrlo jednostavna. U igri sudjeluju dva igrača od kojih svaki predstavlja svoju ekipu – crvena i plava ekipa. Cilj igre je pogoditi lopticu koja se kreće na ekranu tako da loptica ne ode iza nas. Kada igrač uspije pogoditi lopticu tako da ona ode iza protivničkog igrača, ostvaruje bod. Nakon pogotka, igra se nastavlja bez zaustavljanja.

OPIS IGRE

Kada pokrenemo igru ona se otvara u novom prozoru. Vidimo nekoliko elemenata u prozoru: pozadinu, rezultat, mrežu, lopticu i dva reketa. Pozadina je svjetlo zelene boje, a na vrhu prozora se prikazuje rezultat koji ne zaustavlja brojati pogotke sve dok se program ne zatvori. Mreža dijeli prozor na dva jednaka dijela, a u sredini prozora se nalazi loptica koja se automataki pokreće pokretanjem programa. S lijeve i desne strane se nalaze reketi svakog od igrača, jedan je crvene, a drugi plave boje. Igrači upravljaju reketima pritiskom na tipke *W* i *MoveUp* za gore te *S* i *MoveDown* za dolje.

IZRADA IGRE

Glavni dio programa se nalazi u datoteci *pong.py* te zajedno s tim određene su još dvije klase koje olakšavaju izradu programa: *Loptica* i *Reket*. U klasi *Reket* je određeno crtanje reketa te uvjeti da reketi ne idu van okvira prozora. U klasi *Loptica* je određeno crtanje loptice te logika kretanja loptice: kretanje i ubrzanje loptice kada dođe u dodir s reketom. U početnom dijelu glavnog programa su određene potrebne boje te otvaranje prozora s potrebnim glavnim elementima. U redu *carryOn = True* je određeno da se program vrti dok se prozor ne zatvori, a red *clock = pygame.time.Clock()* je potreban za brzinu osvježavanja ekrana.

U glavnom dijelu programa je određeno vrijeme trajanje igre, kontrola reketa, kretanje loptice i uvjeti koji određuju da se odbija od gornjeg i donjeg ruba prozora, pozadina i rezultat koji se prikazuje pri vrhu programa. Na kraju programa je određeno osvježavanje prozora s obzirom na naše kontrole, brzina osvježavanja, te zatvaranje programa.

CIJELI KOD

pong.py

```
import pygame
from reket import Reket
from loptica import Loptica
pygame.init()

# Određivanje potrebnih boja
lzelena = (120,255,135)
bijela = (255,255,255)
zuta = (255,255,0)
crvena = (255,0,0)
plava = (0,10,255)

# Otvaranje prozora
size = (1400, 1000)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Igra Pong")

reket1 = Reket(crvena, 20, 200)
reket1.rect.x = 40
reket1.rect.y = 400

reket2 = Reket(plava, 20, 200)
reket2.rect.x = 1340
reket2.rect.y = 400

loptica = Loptica(zuta,30,30)
loptica.rect.x = 690
loptica.rect.y = 390

all_sprites_list = pygame.sprite.Group()
all_sprites_list.add(reket1)
all_sprites_list.add(reket2)
all_sprites_list.add(loptica)

carryOn = True
clock = pygame.time.Clock()

team1 = 0
team2 = 0
```

```

# Glavni dio programa
while carryOn:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            carryOn = False

    # Kontrola reketa
    keys = pygame.key.get_pressed()
    if keys[pygame.K_w]:
        reket1.moveUp(5)
    if keys[pygame.K_s]:
        reket1.moveDown(5)
    if keys[pygame.K_UP]:
        reket2.moveUp(5)
    if keys[pygame.K_DOWN]:
        reket2.moveDown(5)

    all_sprites_list.update()

    # Kretanje loptice
    if loptica.rect.x >= 1380:
        team1 += 1
        loptica.velocity[0] = -loptica.velocity[0]
    if loptica.rect.x <= 0:
        team2 += 1
        loptica.velocity[0] = -loptica.velocity[0]
    if loptica.rect.y > 980:
        loptica.velocity[1] = -loptica.velocity[1]
    if loptica.rect.y < 0:
        loptica.velocity[1] = -loptica.velocity[1]

    if pygame.sprite.collide_mask(loptica, reket1) or pygame.sprite.collide_mask(loptica, reket2):
        loptica.bounce()

    # Određivanje pozadine, mreže i rezultata
    screen.fill(lzelena)
    pygame.draw.line(screen, bijela, [700, 0], [700, 1000], 15)

    all_sprites_list.draw(screen)

    font = pygame.font.Font(None, 140)
    text = font.render(str(team1), 1, crvena)
    screen.blit(text, (500, 20))
    text = font.render(str(team2), 1, plava)
    screen.blit(text, (840, 20))

    pygame.display.flip()
    clock.tick(60)

pygame.quit()

```

reket.py

```
import pygame
pozadinax = (0,0,0)

# Definiranje klase
class Reket(pygame.sprite.Sprite):
    def __init__(self, color, width, height):
        super().__init__()

        self.image = pygame.Surface([width, height])
        self.image.fill(pozadinax)
        self.image.set_colorkey(pozadinax)

        # Crtanje reketa
        pygame.draw.rect(self.image, color, [0, 0, width, height])
        self.rect = self.image.get_rect()

    def moveUp(self, pixels):
        self.rect.y -= pixels
        # Uvjet da reket ne ide van okvira ekrana
        if self.rect.y < 0:
            self.rect.y = 0

    def moveDown(self, pixels):
        self.rect.y += pixels
        # Uvjet da reket ne ide van okvira ekrana
        if self.rect.y > 800:
            self.rect.y = 800
```

loptica.py

```
import pygame
from random import randint

pozadinax = (0, 0, 0)

# Definiranje klase
class Loptica(pygame.sprite.Sprite):
    def __init__(self, color, width, height):
        super().__init__()

        self.image = pygame.Surface([width, height])
        self.image.fill(pozadinax)
        self.image.set_colorkey(pozadinax)

        # Crtanje loptice
        pygame.draw.rect(self.image, color, [0, 0, width, height])

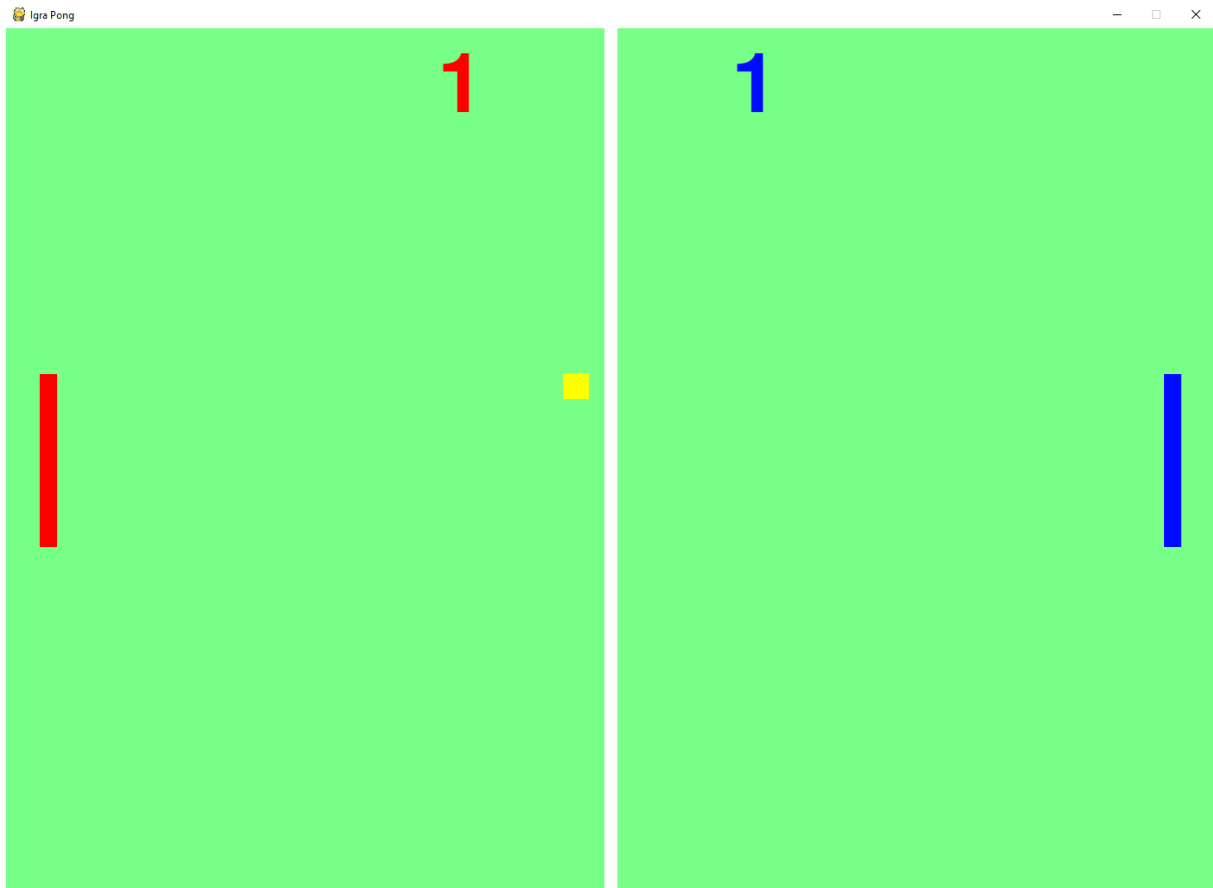
        self.velocity = [randint(6,10),randint(-12,12)]
        self.rect = self.image.get_rect()

    def update(self):
        self.rect.x += self.velocity[0]
        self.rect.y += self.velocity[1]

    def bounce(self):
        self.velocity[0] = -self.velocity[0]
        self.velocity[1] = randint(-8,8)
```

3. ZAKLJUČAK

Nadam se da će vam se igra svidjeti. Iako igra nije previše komplicirana, trebalo je par dana za dovršiti igru jer nisam bio uopće upoznat kako se koristi Pygame modul. Hvala na isprobavanju igre.



screenshot iz igre