SUPER MARIO: PYTHON

TEHNIČKA DOKUMENTACIJA



Autor: Dino Gržinić Mentor: prof. Goran Boneta

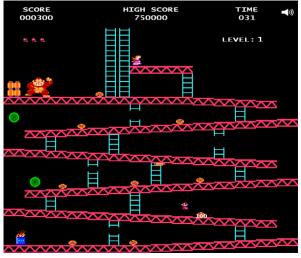
SADRŽAJ

1.	UVOD	2
	1.1 ŠTO JE SUPER MARIO?	2
	1.2 IDEJA PROJEKTA	2
2.	OPIS IZRADE PROGRAMA	3
	2.1 PYGAME	3
	2.2 SPRITES	3
	2.3 POSTAVLJANJE LIKA MARIA	4
	2.4 ZVUČNI EFEKTI	5
	2.5 POSTAVLJANJE NPC-A	5
	2.6 INTERAKCIJA LIKOVA	6
	2.7 OPENING CINEMATIC	6
3.	OBJAŠNJENJE KODA	7
	3.1 VIDEOPLAYER.py	7
	3.2 SUPER MARIO.py	8
4.	UPUTE ZA KORIŠTENJE	. 12
	4.1 CUSTOM MAPA	13
	4.2 PRAVILA IGRE	. 13
	4.3 KONTROLE	. 13
	4.4 TEHNIČKI PODATCI	13

1. UVOD

1.1. ŠTO JE SUPER MARIO?

Super Mario je fiktivni lik koji se pojavljuje u istoimenoj franšizi video igara. U vlasništvu je Nintenda te ga je stvorio japanski dizajner video igara Shigeru



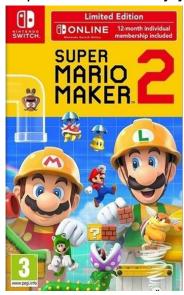
mogu pokušati preći nivoe drugih ljudi.

prvog pojavljivanja do danas, tehnologija video igara te mogućnosti samih računala su se razvijale, a zajedno s njima i lik Maria. Najnovija Super Mario igra je izašla prošle godine pod nazivom Super Mario Maker 2. U njoj se korisnici mogu okušati i izgraditi svoj vlastiti nivo ili

Slika 1.1: Donkey Kong

Miyamoto. Prvi put se pojavljuje na arkadnim strojevima 1981. u igri Donkey Kong. Od tada se pojavljuje u više od 200 igara te raznim animiranim filmovima i stripovima. U svojoj

popularnosti je nadmašio čak i Disneyjevog Mickey Mouse-a. Od njegovog



Slika 1.2: Super Mario Maker 2

1.2 IDEJA PROJEKTA

Inspiriran Super Mario videoigrama, odlučio sam programirati jednu sličnu videoigru te oživjeti jednu od svjetski najpoznatijih videoigara u *retro* izdanju. Igru iz Super Mario franšize koju sam izabrao je Super Mario Bros iz 1985. dostupna na Nintendo Entertainment System-u (NES). Odlučio sam koristiti dva dobro poznata lika iz nje: samog Super Maria te njegovog zloglasnog neprijatelja

– gljiva Goomba.



Slika 1.3: Goomba iz 1985. i Goomba sada



Slika 1.4: Mario iz 1985. i Mario sada

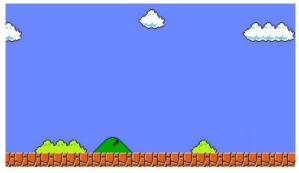
2. OPIS IZRADE PROGRAMA

2.1 PYGAME

Kada sam se odlučio za programiranje video igre, znao sam da će mi Pygame biti nužna Python knjižnica. No u početku nisam znao kako Pygame radi. Stoga sam učio i istraživao o Pygame-u preko raznih internetskih stranica. Kada sam shvatio osnove, počeo sam vježbati na osnovnom programu, na primjer, kako nacrtati pravokutnik i kako se može micati.

2.2 SPRITES

Kada sam naučio kako micati pravokutnik unutar prozora, nisam htio da moja verzija Super Maria ne bude par mičućih pravokutnika na ekranu – trebali su mi pravi likovi. *Sprite* je slika koja se pojavljuje unutar veće grafičke scene (prozora igre). Prvo mi je trebala pozadina. Za nju sam iskoristio sliku Super Mario nivoa s interneta. Zatim je bilo vrijeme da ispunim pozadinu likovima. Na internetu sam pronašao izgled svih originalnih likova korištenih u Super Mario Bros igri poput Goombe i Maria.

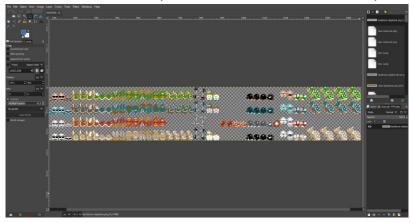


Slika 2.1: Pozadina igre



2.3 POSTAVLJANJE LIKA MARIA

Prvo sam samo sprite Maria trebao 'izrezati' od ostatka slike. Kako bih to učinio, koristio sam se besplatnim softverom za manipulaciju slika GIMP. Prvo sam



Slika 2.2: Program GIMP

dodao Maria, no izgledalo realistično pošto ne izgleda kao da se Mario kreće, već samo slika koja lebdi unutar Trebao prozora. ograničiti njegovo kretanje tako da ne može 'otići' preko ruba prozora

te da se može

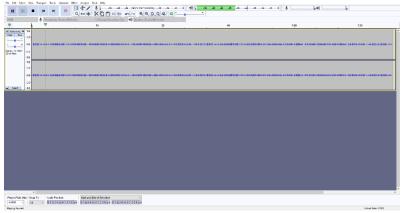
kretati po vodoravnoj liniji. No i dalje ne izgleda realistično pošto nema animacije hodanja. Kako bih to napravio, na ekranu se treba na određeno vrijeme prikazati određeni sprite Maria kako bi se dobio dojam hodanja. Sve potrebne spriteove sam izrezao iz Slika 2.3: Jedan od slike 2.2 te sam ih spremio kao pojedinačne slike. Pošto se Mario

Mario spriteova

kreće ii lijevo i desno, ne mogu koristiti iste *spriteove* za oba kretanja ako želim zadržati realističnost. Zbog toga sam napravio dvije liste sa slikama (jedna se desnim hodom, druga s lijevim). Ovisno koju tipku igrač pritisne (A-lijevo, Ddesno), koristit će se određena lista. Dokle god je tipka pritisnuta, vrijednost tzv. 'brojača hoda' će se povećavati te će se pomoću *for* petlje prikazivati jedan za drugim spriteovi iz liste. Koliko dugo će biti prikazan pojedini sprite ovisi o FPS-u kojega sam postavio na 60 jer igra nema veliki utjecaj na CPU računala. Idući korak je bio omogućiti Mariu da skače. Puta skoka dok je Mario u pokretu treba izgledati kao parabola okrenuta prema dolje. Može ju se napraviti u koordinatnom sustavu koji je unutar prozora igre zapravo širina (x os) i visina (y os) samog prozora. Naknadno sam dodao posebne spriteove Maria kako skače radi realističnosti.

2.4 ZVUČNI EFEKTI

++Također sam kod igrača htio zabaviti ne samo osjetilo vida, već i sluha. Kako bih to postigao, koristio sam se zvučnim efektima iz Super Maria (pozadinski zvuk igre, zvuk skoka, zvuk kraja igre, zvuk pucanja). Za puštanje zvuka koristio sam se s *pygame.mixer.music* funkcijom. No kasnije sam otkrio da ona može puštati samo jednu po jednu .mp3 datoteku. Stoga sam tu funkciju koristio samo za puštanje pozadinskog zvuka igre te sam stavio da se konstantno ponavlja nakon završetka. Ostale zvučne datoteke sam trebao obraditi u programu za



Slika 2.4: Program Audacity

manipulaciju Audacity. Prvo sam trebao datoteke 'izrezati' kako bi se čuo samo dio koji želim čuti te sam ih zatim exportao kao .wav datoteke kako bih mogao zvučnih više efekata istovremeno puštati pomoću funkcije

pygame.mixer.Sound. Zatim sam te zvukove trebao povezati s određenim radnjama poput pritiska tipke (W-skok => puštanje zvučnog efekta skoka).

2.5 POSTAVLJANJE NPC-A

NPC je skraćenica za Non-Player Character. To je lik čije akcije nisu kontrolirane

od strane igrača, već od njegovog koda. NPC-jevi su se također koristili u *Super Mario Bros* igri, poput gljive.

Postupak je sličan izradi Maria. Izrezao sam potrebne slike, napravio da se izmjenjuju kako bi se

Slika 2.6: *Sprite* činilo da hoda, u kodu sam napisao obrazac njenog kretanja. No još leteće ribe uvijek likovi nisu mogli interagirati jedan s drugim. Naknadno sam ubacio i drugi NPC – leteću ribu. Ona prolijeće iznad gljive i Maria, svako toliko. Ubacio sam ju radi ograničavanja Mariovog kretanja, no kao i izvor dodatnih bodova.

2.6 INTERAKCIJA LIKOVA

Prvo sam Mariu htio omogućiti da može pucati gljivu *fireballom*. Izrezao sam *fireball sprite* te sam omogućio da se s Mariovih koordinata ispali *fireball* pritiskom tipke u smjeru u kojem Mario gleda. Ako se koordinate *fireballa* i NPC-a poklapaju, *fireball* će nestati te će program zabilježiti pogodak. Ako igrač pogodi gljivu, dobit će 1 bod.

Ako pogodi ribu, koja se rjeđe pojavljuje na ekranu i brza je te treba skočiti da ju se pogodi, onda dobiva 5 bodova. No kako igrač ne bi konstantno pritiskao paljbu, ograničio sam njegovo pucanje tako da može ispucati sljedeći fireball tek kad se prethodni sudari s NPC-om ili otiđe s ekrana. To znači da što je Mario bliže NPC-u, to će moći ostvariti više pogodaka, no riskirat će sudar s NPC-jem. Tako će ujedno i igra završiti, ako se Mario i NPC dotaknu. Taj trenutak se može identificirati tako da se likovima nacrtaju *hitboxi*. Korištenje okvira samih slika kao *hitboxa* ne bi bilo efikasno jer likovi ne zauzimaju cijelu površinu slike, već treba ručno odrediti položaj, veličinu i oblik *hitboxa*. Kada se bridovi *hitboxa* NPC-a i Maria dotaknu, to će reći programu da je došlo do sudara te da izvrši kod namijenjen za tu situaciju.

2.7 OPENING CINEMATIC

Opening cinematic je film od par sekundi koji se pojavljuje prije početka igre koji služi da pripremi igrača na ono što slijedi. Stoga sam se i ja okušao u izrađivanju jedne. Za to mi je trebao program za uređivanje videozapisa. Za to sam koristio besplatni program *Hitfilm Express*. Kad sam napravio cinematic, trebao sam



Slika 2.8: Program Hitfilm Express

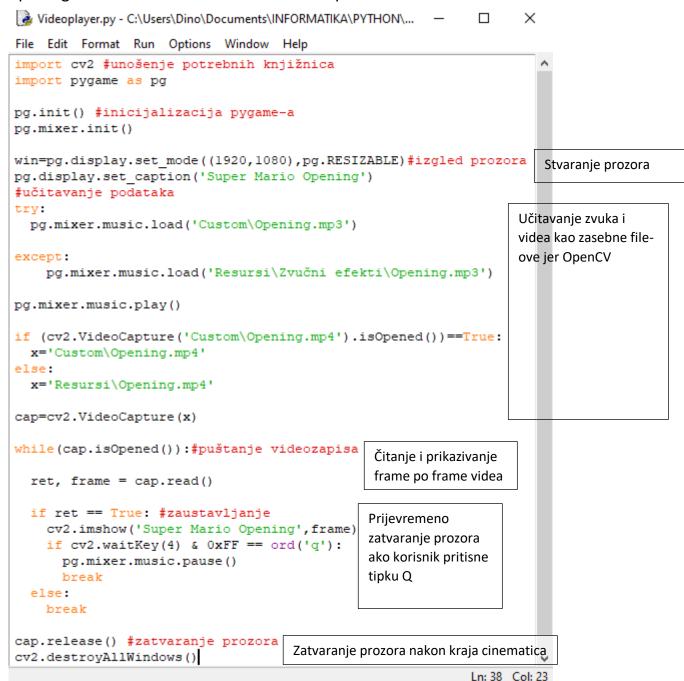
shvatiti kako reproducirati video u Pythonu. Za to su mi trebale knjižnice OpenCV, Numpy i Pygame. Napravio sam zaseban Python file za reprodukciju videa jer se izvodi u zasebnom prozoru te se nakon kraja videa zatvara i igra počinje.

Također sam dodao opciju preskakanja cinematica da ne dosadi igraču nakon svakog ulaženja u igru. Kako bi se igra pokrenula poslije cinematica, importao sam file s video reprodukcijom u glavni.

3. OPIS KODA

3.1 VIDEOPLAYER.py

Videoplayer.py je odvojeni Python file od ostatka koda koji služi za reprodukciju opening cinematica u zasebnom fullscreen prozoru.



Slika 3.1: Kod za Videoplayer.py

3.2 SUPER MARIO.py

```
B Super Mario.py - C:\Users\Dino\Documents\INFORMATIKA\PYTHON\Super Mario\Super... 

□
File Edit Format Run Options Window Help
import pygame
                                         Unošenje Pygamea i Videoplayer.py
import Videoplayer
pygame.init()
win=pygame.display.set mode((960,540),pygame.RESIZABLE) #dimenzije prozora
                                                                               Izgled i
zaslon dužina=960
                                                                               stvaranje
zaslon visina=540
a=pygame.image.load('Resursi\mario ikona.png')
                                                                               prozora i font
pygame.display.set icon(a)
pygame.display.set caption('Super Mario | Python')
font=pygame.font.Font('Resursi\SuperMario256.ttf',25)
pygame.mixer.init() #zvučni efekti
                                                                               Zvučni efekti
   pygame.mixer.music.load('Custom\Mario song.mp3')
                                                                               aktivirani pri
except:
                                                                               određenim
   pygame.mixer.music.load('Resursi\Zvučni efekti\Mario song.mp3')
                                                                               događajima
pygame.mixer.music.play(loops=-1)
   jump=pygame.mixer.Sound('Custom\Mario Jump.wav')
except:
    jump=pygame.mixer.Sound('Resursi\Zvučni efekti\Mario Jump.wav')
    fireball=pygame.mixer.Sound('Custom/fireball.wav')
except:
    fireball=pygame.mixer.Sound('Resursi/Zvučni efekti/fireball.wav')
TIMER EVENT=pygame.USEREVENT+1 #ubrzavanje gljive
                                                                   Ubrzavanje gljive i ribe
pygame.time.set_timer(TIMER_EVENT,10000)
                                                                   svakih 10 sekundi
dužina slike=50 # sprite-ovi
                                                                             Unošenje sprite-ova
visina slike=58
trv:
    bullet=pygame.image.load('Custom\metak.png')
   bullet=pygame.image.load('Resursi\Sprites\metak.png')
bullet=pygame.transform.rotozoom(bullet,-90,1)
trv:
    skok desno=pygame.image.load('Custom\mario.png')
    skok lijevo=pygame.image.load('Custom\mario.png')
except:
    skok desno=pygame.image.load('Resursi\Sprites\mario skače desno.png')
    skok lijevo=pygame.image.load('Resursi\Sprites\mario skače lijevo.png')
try:
    desni hod=[pygame.image.load('Custom\mario.png'),pygame.image.load('Custom
    lijevi_hod=[pygame.image.load('Custom\mario.png'),pygame.image.load('Custom\mario.png')
    idle=pygame.image.load('Custom\mario.png')
   fail=pygame.image.load('Custom\mario.png')
except:
    desni hod=[pygame.image.load('Resursi\Sprites\mario l.png'),pygame.image.
    lijevi hod=[pygame.image.load('Resursi\Sprites\mario 1 obrnuti.png'),pygan
    idle=pygame.image.load('Resursi\Sprites\mario 1.png')
    fail=pygame.image.load('Resursi\Sprites\mario fail.png')
```

```
*Super Mario.py - C:\Users\Dino\Documents\INFORMATIKA\PYTHON\Super Mario\Super M...
                                                                                    Х
File Edit Format Run Options Window Help
                                            Postavljanje FPS-a u glavnoj petlji
clock=pygame.time.Clock()
run=True
                                            Određivanje kada igra traje
class riba(object):
                                                                                     Postavljanje NPC-a o
                                                                                     obliku objekta sa
        riba desno=pygame.image.load('Custom\riba.png')
                                                                                     raznim
        riba_lijevo=pygame.image.load('Custom\riba.png')
                                                                                     podvrijednostima
    except:
        riba desno=[pygame.image.load('Resursi\Sprites/riba 1 obrnuta.png'),pyga
        riba lijevo=[pygame.image.load('Resursi\Sprites/riba l.png'),pygame.imag
                                                                                     Izgled spriteova
    for j in range (len(riba desno)):
        riba desno[j]=pygame.transform.scale(riba desno[j], (43,44))
        riba lijevo[j]=pygame.transform.scale(riba lijevo[j],(43,44))
                                                                                     Vrijednosti za ribu
    def init (self,x,y,width,height,kraj):
        self.x=x
        self.y=y
        self.width=width
        self.height=height
        self.put=[x,kraj]
        self.brojač hoda=0
        self.vel=7
        self.hitbox=(self.x-5, self.y, 55, 45)
                                                                                     Izmjenjivanje
                                                                                     prikazivanja spriteova
    def crtaj(self,win):
                                                                                     radi dojma kretanja
        self.move()
        if self.brojač hoda+1>=24:
            self.brojač hoda=0
        if self.vel>0:
            win.blit(self.riba desno[self.brojač hoda//12],(self.x,self.y))
            self.brojač hoda+=1
            win.blit(self.riba lijevo[self.brojač hoda//12],(self.x,self.y))
            self.brojač hoda+=1
        self.hitbox=(self.x+4,self.y+4,42,42)
        #pygame.draw.rect(win, (255,0,0), self.hitbox,2)
                                                                                     Ograničavanje
                                                                                     kretanja
    def move(self):
        if self.vel>0:
            if self.x+self.vel<self.put[1]:</pre>
                self.x+=self.vel
            else:
                self.vel=self.vel*-1
                self.brojač hoda=0
        else:
            if self.x-self.vel>self.put[0]:
                self.x+=self.vel
            else:
                self.vel=self.vel*-1
                self.x+=self.vel
                self.brojač hoda=0
class gljiva(object):
```

Ln: 84 Col: 17

```
Super Mario.py - C:\Users\Dino\Documents\INFORMATIKA\PYTHON\Super Mario\Super Mar... —
                                                                                   Х
File Edit Format Run Options Window Help
    def hit(self): #u slučaju sudara
                                                                              Zaustavljanje igre,
        font=pygame.font.Font('Resursi\SuperMario256.ttf',30)
                                                                              prikaz rezultata te
        win.blit(pozadina,(move x,0))
        win.blit(neprijatelj.gljiva_hod[1],(neprijatelj.x,neprijatelj.y))
                                                                              zatvaranje programa
        if riba.vel>0:
            win.blit(riba.riba desno[1], (riba.x, riba.y))
        else:
            win.blit(riba.riba lijevo[1], (riba.x, riba.y))
        win.blit(fail, (self.x, self.y-2))
        trv:
            pygame.mixer.music.load('Custom\game over.mp3')
        except:
            pygame.mixer.music.load('Resursi\Zvučni efekti\game over.mp3')
        pygame.mixer.music.play()
        if brojač pogodaka<=20:
            text = font.render('Ukupan broj bodova : {} ... Jadno!'.format(broja
        if brojač pogodaka>20 and brojač pogodaka<=30:
            text = font.render('Ukupan broj bodova : {} ... Meh!' .format(brojač
        if brojač pogodaka>30 and brojač pogodaka<=45:
            text = font.render('Ukupan broj bodova : {} ... Nije lose!' .format(
        if brojač_pogodaka>45 and brojač_pogodaka<=60:
            text = font.render('Ukupan broj bodova : {} ... Opa!' .format(brojač
        if brojač pogodaka>60:
            text = font.render('Ukupan broj bodova : {} ... To legendo!'
        win.blit(text, (185, 180))
        pygame.display.update()
        pygame.time.delay(3000)
        pygame.quit()
class projektil(object): #metci
                                                                           Objekt fireballa i
    def init (self,x,y,radius,boja,smjer):
                                                                           njegova svojstva
        self.x=x
        self.y=y
        self.radius=radius
        self.boja=boja
        self.smjer=smjer
        self.vel=12*smjer
    def crtaj(self,win):
        win.blit(bullet,(self.x,self.y))
def crtanje(): #crtanje objekata
                                                                                    Crtanje svih
    global brojač hoda
    text = font.render('Broj bodova : {}'.format(brojač_pogodaka), True, (255, $\psi$, objekata
    win.blit(pozadina, (move x,0)) #pozadina
                                                                                    unutar
   Mario.crtaj(win)
                                                                                    prozora te
    neprijatelj.crtaj(win)
                                                                                    osvježavanje
    riba.crtaj(win)
    win.blit(text, (7,8))
                                                                                    prozora radi
    for metak in municija:
                                                                                    prikazivanja
        metak.crtaj(win)
                                                                                    promjena
    pygame.display.update()
Mario=igrač(800,435,35,35)
                                                       Određivanie
neprijatelj=gljiva(0,452,32,32,920)
                                                       pozicija objekata
riba=riba(-2550,220,32,32,2470)
municija=[]
                                                                            Ln: 84 Col: 17
```

```
B Super Mario.py - C:\Users\Dino\Documents\INFORMATIKA\PYTHON\Super Mario\Super Mar... — □
File Edit Format Run Options Window Help
shootLoop=1
# glavna petlja
                                 Unutar petlje su operacije koje stalno nadziru kad
while run:
                                 se određeni događaj dogodi
    clock.tick(60)
                      # FPS
                              Postavljanje FPS-a na 60
                                                                                       Identificiranje
    if Mario.hitbox[1]<neprijatelj.hitbox[1]+neprijatelj.hitbox[3] and Mario.hit
                                                                                       sudara
        if Mario.hitbox[0]+Mario.hitbox[2]>neprijatelj.hitbox[0] and Mario.hitbo
            Mario.hit()
    if Mario.hitbox[1]<riba.hitbox[1]+riba.hitbox[3] and Mario.hitbox[1]+Mario.h
        if Mario.hitbox[0]+Mario.hitbox[2]>riba.hitbox[0] and Mario.hitbox[0]<ri
    if shootLoop>0: #ograničavanje pucanja
        shootLoop+=1
    if shootLoop>1:
        shootLoop=0
    for metak in municija: # pucanje
                                                                                    Identificiranie
                                                                                    pogotka
        if metak.y-metak.radius<neprijatelj.hitbox[1]+neprijatelj.hitbox[3] and
            if metak.x+metak.radius>neprijatelj.hitbox[0] and metak.x-metak.radi
                brojač pogodaka+=1
                municija.pop(municija.index(metak))
        if metak.y-metak.radius<riba.hitbox[1]+riba.hitbox[3] and metak.y+metak
            if metak.x+metak.radius>riba.hitbox[0] and metak.x-metak.radius<riba
                brojač pogodaka+=5
                municija.pop(municija.index(metak))
        if metak.x<960 and metak.x>0:
            metak.x+=metak.vel
        else:
            municija.pop (municija.index (metak))
    tipka=pygame.key.get pressed() #kontrole
                                                                                 Događaji koji se
                                                                                 pokreću pri
    for event in pygame.event.get(): #zatvaranje prozora
        if event.type==pygame.QUIT or tipka[pygame.K q]:
                                                                                 pritiskų određene
            run=False
                                                                                 tipke
        if event.type==TIMER EVENT:
            if neprijatelj.vel<0:
                neprijatelj.vel-=4
            else:
                neprijatelj.vel+=4
            riba.put[0]+=200
            riba.put[1]-=200
    if tipka[pygame.K SPACE] and shootLoop==0: # pucanje
        if Mario.lijevo:
            smjer=-1
            bullet=pygame.image.load('Resursi\Sprites\metak-lijevo.png')
            bullet=pygame.image.load('Resursi\Sprites\metak-desno.png')
            smjer=1
        if len(municija)<1:
            fireball.play()
            municija.append(projektil(round(Mario.x+Mario.width//2),round(Mario.
                                                                            Ln: 84 Col: 17
```

```
Super Mario.py - C:\Users\Dino\Documents\INFORMATIKA\PYTHON\Super Mario\Super Mar...
                                                                                    X
File Edit Format Run Options Window Help
            smjer=-1
            bullet=pygame.image.load('Resursi\Sprites\metak-lijevo.png')
            bullet=pygame.image.load('Resursi\Sprites\metak-desno.png')
            smjer=1
        if len(municija)<1:
            fireball.play()
            municija.append(projektil(round(Mario.x+Mario.width//2),round(Mario.
    if tipka[pygame.K m]: # kontrole
                                                          Stopiranje pozadinske glazbe
        pygame.mixer.music.pause()
    if tipka[pygame.K_a] and Mario.x>Mario.vel:
                                                            Kretanje Maria lijevo-
        Mario.x-=Mario.vel
                                                            desno
        Mario.lijevo=True
        Mario.desno=False
        Mario.stajanje=False
    elif tipka[pygame.K_d] and Mario.x<zaslon_dužina-Mario.width-Mario.vel:
        Mario.x+=Mario.vel
        Mario.lijevo=False
        Mario.desno=True
        Mario.stajanje=False
    else:
                                                                Izvođenje skoka
        Mario.stajanje=True
        Mario.brojač hoda=0
    if tipka[pygame.K_s] and Mario.y!=435:
        Mario.y=435
        Mario.skok=False
    if not Mario.skok: #skok
        if tipka[pygame.K w]:
            Mario.skok=True
            Mario.desno=False
            Mario.lijevo=False
            Mario.brojač hoda=0
            Mario.brojač skoka=16
            jump.play()
    else:
        if Mario.brojač skoka>=-16:
            neg=1
            if Mario.brojač_skoka<0:</pre>
                neg=-1
            if Mario.y==262 and Mario.x>=460 and Mario.x<=6
                Mario.y=262
                Mario.y-=(Mario.brojač skoka**2)*0.15*neg
            Mario.brojač_skoka-=1
        else:
            Mario.skok=False
            Mario.bojač skoka=16
    crtanje()
                                                  Stalno prikazivanje objekata
    pygame.display.update() #osvježavanje prozora
                                                     Prikazivanje novih promjena
                                                     vrijednosti u prozoru, ako ih ima,
pygame.quit() #kraj programa
                                                     te završetak programa
                                                                               Ln: 1 Col: 0
```

4. UPUTE ZA KORIŠTENJE

4.1 CUSTOM MAPA

Ako bi igrač htio promijeniti neke resurse koje sam koristio, to može učiniti u Custom mapi. To je prazna mapa u koju korisnik stavi file s određenim nazivom i formatom te će program umjesto zadanog resursa koristiti korisnikov. Mogu se promijeniti: cinematic, Mario, NPC-jevi, pozadina, fireball, zvučni efekti.

4.2 PRAVILA IGRE

Kao Mario, tvoj zadatak je da vatrenim loptama što više puta pogodiš gljivu ili leteću ribu. Radi sprječavanja konstantnog pucanja, novi fireball se neće moći ispaliti dokle god prethodni ne dođe do ruba prozora ili ne pogodi gljivu. To znači, što si bliže gljivi, više pogodaka, ali i veći rizik od sudara. Leteća riba, nakon određenog vremena, prolazi preko ekrana. Pogodak gljive vrijedi 1 bod, dok pogodak leteće ribe vrijedi 5 bodova. Imaš izbor, nastaviti gađati gljivu ili riskirati gađanjem leteće ribe. No ako te gljiva ili leteća riba dotaknu, igra završava. Kako igra prolazi, gljiva se kreće sve brže i brže te se riba sve češće pojavljuje.

4.3 KONTROLE

W – skok

A – lijevo

D – desno

S – brz povratak na tlo iz skoka

SPACE - pucanje vatrenih lopti

M – isključivanje glazbe

Q – preskoči cutscenu/izađi iz igre

4.4 TEHNIČKI PODATCI

Veličina cijele igre: 6,57 MB

Veličina resursa: 6,55 MB

Veličina samog koda: 14,9 KB

Broj linija koda: 448

Korištenje CPU-a: 0,45 GHz

Korištenje RAM-a: 40 MB

Ne preporuča se pokretati na starijim laptopima.