**PODZEMNE BORBE** – *dokumentacija*

Sadržaj

[PYGAME i PYTHON OPĆENITO 4](#_Toc153485557)

[class() 4](#_Toc153485558)

[def \_\_init\_\_(self) 4](#_Toc153485559)

[convert\_alpha() 4](#_Toc153485560)

[display.set\_mode() 4](#_Toc153485561)

[display.set\_caption() 4](#_Toc153485562)

[display.update() 4](#_Toc153485563)

[draw.rect() 4](#_Toc153485564)

[enumerate() 4](#_Toc153485565)

[event.get() 4](#_Toc153485566)

[event.type() 4](#_Toc153485567)

[event.unicode 4](#_Toc153485568)

[font.Font() 4](#_Toc153485569)

[font.Font.render() 5](#_Toc153485570)

[global() 5](#_Toc153485571)

[image.load() 5](#_Toc153485572)

[os. path. join() 5](#_Toc153485573)

[key.get\_pressed() 5](#_Toc153485574)

[map() 5](#_Toc153485575)

[mixer.init() 5](#_Toc153485576)

[mixer.Sound() 5](#_Toc153485577)

[mouse.get\_pos() 5](#_Toc153485578)

[\_\_name\_\_ == “\_\_main\_\_“ 5](#_Toc153485579)

[open() 5](#_Toc153485580)

[pygame.init() 5](#_Toc153485581)

[Rect() 5](#_Toc153485582)

[screen.blit() 5](#_Toc153485583)

[sprite 6](#_Toc153485584)

[sprite.Group() 6](#_Toc153485585)

[sprite.Group.draw() 6](#_Toc153485586)

[sprite.Group.empty() 6](#_Toc153485587)

[sprite.Sprite() 6](#_Toc153485588)

[SRCALPHA 6](#_Toc153485589)

[Super() 6](#_Toc153485590)

[Surface() 6](#_Toc153485591)

[Surface.fill() 6](#_Toc153485592)

[Surface.get\_rect() 6](#_Toc153485593)

[Surface.set\_alpha() 6](#_Toc153485594)

[time.Clock 6](#_Toc153485595)

[time.Clock.tick() 7](#_Toc153485596)

[time.time() 7](#_Toc153485597)

[transform.flip() 7](#_Toc153485598)

[zip() 7](#_Toc153485599)

[PODZEMNE BORBE – FUNKCIJE/KLASE 7](#_Toc153485600)

[def LOADING\_SCREEN() 7](#_Toc153485601)

[Class SlikeGumbi() 7](#_Toc153485602)

[class SpriteRectangle() 7](#_Toc153485603)

[def flip\_image\_horizontally() 7](#_Toc153485604)

[class Andrej() 8](#_Toc153485605)

[class Broz(Andrej) 10](#_Toc153485606)

[def crtanjeHealthaIImena() 10](#_Toc153485607)

[def crtanjeRunde() 11](#_Toc153485608)

[def provjeraPozicijeZaObrnuto() 11](#_Toc153485609)

[def provjeraTrajanjaUdaraca() 11](#_Toc153485610)

[def provjeraCrouchanja() 12](#_Toc153485611)

[def provjeraBlokiranja() 12](#_Toc153485612)

[def provjeraKretanjaIKretanje 13](#_Toc153485613)

[def provjeraSkokaISkakanje 13](#_Toc153485614)

[def provjeraUdarcaIUdaranje 13](#_Toc153485615)

[def provjeraNogeINogatanje 13](#_Toc153485616)

[def provjeraDamagea() 13](#_Toc153485617)

[def provjeraJeLiStunned() 14](#_Toc153485618)

[def provjeraZaRegeneriranjeStamine() 14](#_Toc153485619)

[class Player() 14](#_Toc153485620)

[def read\_data() 15](#_Toc153485621)

[def read\_data\_sve() 15](#_Toc153485622)

[def poredak() 15](#_Toc153485623)

[def update\_achievementa() 15](#_Toc153485624)

[def zadnji\_achievement() 15](#_Toc153485625)

[def update\_score() 15](#_Toc153485626)

[class Button() 15](#_Toc153485627)

[def escape\_screen() 16](#_Toc153485628)

[def main() 16](#_Toc153485629)

[def crtajAchievemente() 16](#_Toc153485630)

[def imali() 16](#_Toc153485631)

[def postignuca\_nema\_profila() 16](#_Toc153485632)

[def draw\_buttons() 17](#_Toc153485633)

[def prije\_postignuca() 17](#_Toc153485634)

[def postignuca() 17](#_Toc153485635)

[def imenovanje\_profila() 17](#_Toc153485636)

[def biranje\_profila() 17](#_Toc153485637)

[def odabir\_borca1() 17](#_Toc153485638)

[def odabir\_borca2() 17](#_Toc153485639)

[def keybind\_screen1() 17](#_Toc153485640)

[def keybind\_screen2() 17](#_Toc153485641)

[def odabir\_rundi() 17](#_Toc153485642)

[def findDamageRectangle() 17](#_Toc153485643)

[def provjeraJeLiTkoDefeated() 17](#_Toc153485644)

[def reset\_igre() 18](#_Toc153485645)

[def igranje() 18](#_Toc153485646)

[def winscreen() 18](#_Toc153485647)

# PYGAME i PYTHON OPĆENITO

## class()

* Izrađuje se nova klasa koja zapravo definira novi tip objekta u Pythonu

## def \_\_init\_\_(self)

* \_\_init\_\_ = ova se metoda poziva kada se objekt kreira iz classa i omogućuje da ima značajke tog classa
* self = predstavlja objekt

## convert\_alpha()

* mijenja format pixela
* Surface.convert\_alpha()
* optimizira Surface za brže iscrtavanje na ekranu, koristeći algoritme koji su prilagođeni za slike s alfa (transparentnim) kanalom.

## display.set\_mode()

* funkcija koja stvara prozor igre (Surface ili površina zaslona)
* pygame.display.set\_mode( širina, visina )
  + u argumentu mora biti barem jedan argument npr. tuple

## display.set\_caption()

* funkcija koja daje naslov trenutnom prozoru
* pygame.display.set\_caption (naslov)

## display.update()

* ažurira površinu zaslona

## draw.rect()

* crta pravokutnik
* pygame.draw.rect(objekt, boja, rect(pravokutnik))

## enumerate()

* uzima nešto iterativno i redom pridjeljuje brojeve od starta te tako kreira enumerate objekt
* enumerate(iterable, start)

## event.get()

* daje pristup svim eventima, u obliku liste, koji su se dogodili tijekom izvršavanja koda

## event.type()

* daje type eventa

## event.unicode

* omogućuje unošenje znakova u igri

## font.Font()

* učitava novi font iz naziva datoteke ili Python datoteke
* veličina je visina fonta u pikselima
* ako ništa nije napisano na mjestu datoteke učitat će se zadani font za Pygame
* nakon što je font kreiran, veličina se ne može mijenjati
* pygame.font.Font(datoteka, veličina)

## font.Font.render()

* ispisuje tekst na novoj površini, odnosno Pygame ne dopušta izravno ispisivanje teksta na postojećoj površini pa se pomoću ove funkcije preslikava postojeća te se ispisuje traženi tekst
* render(tekst, antialias(koeficijent zaobljenosti teksta), boja)

## global()

* radi global varijable unutar funkcije
* omogućuje izmjenu varijable van trenutnog opsega

## image.load()

* učitava sliku iz foldera
* pygame.image.load(ime datoteke)
* svako učitavanje slike stvara novi Surface

### os. path. join()

* olakšava i ubrzava pronalaženje željenog dokumenta
* os. path. join (path do dokumenta)

### key.get\_pressed()

* pamti gumb koji je kliknut na tipkovnici

### map()

* izvrši određenu funkciju nad svim elementima iterativnog objekta
* map(function, iterable)

## mixer.init()

* modul za učitavanje i reprodukciju zvukova

## mixer.Sound()

* učitava zvučni objekt iz datoteke
* pygame.mixer.Sound(ime datoteke)

## mouse.get\_pos()

* vraća položaj miša na zaslonu u odnosu na gornji lijevi kut displayja u obliku (x ,y)

## \_\_name\_\_ == “\_\_main\_\_“

* osigurava da se program počne vrtjeti samo ako je ta skripta, odnosno taj program pokrenuta direktno kao main program, a ne da se pokreće kao importani modul u nekoj drugoj skripti

## open()

* otvara određeni dokument

## pygame.init()

* instalira sve Pygame module
* piše se na početku jer bez njega Pygame ne bi radio

## Rect()

* pamti koordinate objekta i omogućuje da se može micati
* pygame.Rect(prva koordinata, druga koordinata, širina, visina)
* pygame.Rect(objekt)

## screen.blit()

* stavlja jedan Surface u drugi
* screen.blit(surface koji želimo staviti u glavni Surface, koordinate gdje ga želimo)

## sprite

* dvodimenzionalna slika koja je dio veće grafičke cjeline
* mogu se animirati, igrač ih može kontrolirati, a mogu i međusobno komunicirati
* ima Surface i rectangle

## sprite.Group()

* jednostavan “spremnik“ za Spritove
* uzima bilo koji broj Spritova za dodavanje grupi
* podržava sljedeće standardne Python operacije: in, len, bool, iter

## [sprite.Group.draw](https://www.pygame.org/docs/ref/sprite.html" \l "pygame.sprite.Group.empty)()

* [sprite.Group.draw](https://www.pygame.org/docs/ref/sprite.html" \l "pygame.sprite.Group.empty)(Surface)
* crta sve spriteove na Surface

## [sprite.Group.empty](https://www.pygame.org/docs/ref/sprite.html" \l "pygame.sprite.Group.empty)()

* prazni grupu od svih spriteova

## sprite.Sprite()

* osnovan class za vidljive objekte igre

## SRCALPHA

* = source alpha 🡪 surface flag
* omogućuje surfaceu da sadrži alpha kanal, tj. da se može moderirati transparentnost Surfacea

## Super()

* daje pristup prethodno definiranim metodama u nekoj drugoj klasi

## Surface()

* stvara Surface/ površinu zaslona, ali ju naravno ne prikazuje jer nema funkcije display
* kao argument uzima tuple

## Surface.fill()

* ispunjava površinu jednom bojom
* ako se ne zada argument rect, cijela površina će biti ispunjena (rect ograničava ispunu na određeno područje)
* .fill(boja, rect)

## Surface.get\_rect()

* stvara novi pravokutnik koji obuhvaća cijeli Surface
* taj će pravokutnik uvijek počinjati na (0, 0) s jednakim dimenzijama kao slika
* ako se želi promijeniti mjesto počinjanja pravokutnika potrebno je napraviti sljedeće: surface.get\_rect(pozicija(npr.center, top left..)=(koordinate))
* pomaže pri micanju elemenata npr. igrača

## Surface.set\_alpha()

* određuje tranparentnost slike
* set\_alpha(broj ( od 0 do 255))
  + ako je broj 0 slika je potpuno transparentna, odnosno prozirna
  + ako je broj 255 slika je nije nimalo prozirna

## time.Clock

* stvara objekt koji prati vrijeme (poput štoperice)

## time.Clock.tick()

* načelno računa u milisekundama koliko je prošlo od prethodnog pozivanja funkcije
* ograničava brzinu izvođenja igre
  + time.Clock.tick(vrijeme u milisekundama)

## time.time()

* vraća vrijeme u sekundama od početka epohe (1. Siječnja 1970.)

## transform.flip()

* okreće Surface vertikalno i/ili horizontalno
* pygame.transform.rotate(surface, flip\_x, flip\_y), gdje su flip\_x i flip\_y bool varijable (za True se okreće, za False se ne okreće)

## zip()

* funkcija koja vrati zip objekt, gdje su spareni prvi s prvim, drugi s drugim itd., elementi više iterativnih objekata
* zip(iterator1, iterator2, iterator3…)

# PODZEMNE BORBE – FUNKCIJE/KLASE

## def LOADING\_SCREEN()

* crta loading screen kada se pali igrica i pušta zvuk

## Class SlikeGumbi()

* definira type objekta za gumb koji nije samo obojani surface, već ima svoju sliku
* **def provjeraSudara()**
  + provjerava je li miš na gumbu i ako je vraća True
* **def crtanjeGumba()**
  + crta hover izgled gumba ako je miš na njemu i crta običan izgled gumba ako miš nije na njemu

## class SpriteRectangle()

* radi objekt rectangle određene boje(za testing važno), na određenoj poziciji određenih dimenzija

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

## def flip\_image\_horizontally()

* okreće surface (sliku) horizontalno, mirrora je

Slika na kojoj se prikazuje tekst, Font, snimka zaslona, crta

Opis je automatski generiran

## class Andrej()

* **def \_\_init\_\_():**
  + gleda se koji igrač je odabrao Andreja (1 ili 2), inicijalizira se početni health, stamina, score, brojne varijable za movement i animacije za Andreja (boolean – koje prate stanje lika u fightu i je li trenutno u nekom od movementa, int – koje prate duration animacija i liste u koje se spremaju slike animacija )
* **def resetBeforeGame()**
  + vrati rectangle borca na početak, i varijable za movement stavi na početne vrijednosti
* **def idleRectangles()**; **def jumpRectangles()**; **def crouchRectangles()**; **def blockRectangles()**; **def stunnedRectangles()**; **def crouchPunchRectangles()**; **def jumpPunchRectangles()**; **def jumpKickRectangles()**; **def punchRectangles**; **def kickRectangles**
  + sve ove metode opisuju kako rectangleovi za borca izgledaju u određenom njegovom položaju i određuju pozicije tih rectangleova (**primjer** je na slici ispod)

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

* **def crtanjeRectenglova()**
  + crta na ekran sve rectangleove koji obilježuju jednog borca
* **def postavljanjeRectangleova()**
  + na borca aplicira one rectangleove u kojem se movementu/stanju borac nalazi
* **def animacije()**
  + crta one animacije borca ovisno u kojem se movementu/stanju borac nalazi
* **def flipanje\_slika()**
  + ovisno o movementu, flipa/mirrora surface borca ako zamjeni stranu sa drugim borcem (posljedično će se tako i rectanglovi borca crtati obrnuto)
* **def gravitacija()**
  + metoda koja borcu daje gravitaciju da pada prema podlozi na kojoj se zaustavi
* **def move()**
  + metoda koja borcu omogućava kretanje lijevo desno, i u stojećem i čućečem položaju

**def gravitacija()**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

**def move()**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

## class Broz(Andrej)

* **def \_\_init\_\_():**
  + gleda se koji igrač je odabrao Broza (1 ili 2), inicijalizira se početni health, stamina, score, brojne varijable za movement i animacije za Broza (boolean – koje prate stanje lika u fightu i je li trenutno u nekom od movementa, int – koje prate duration animacija i liste u koje se spremaju slike animacija )
* **def idleRectangles()**; **def jumpRectangles()**; **def crouchRectangles()**; **def blockRectangles()**; **def stunnedRectangles()**; **def crouchPunchRectangles()**; **def jumpPunchRectangles()**; **def jumpKickRectangles()**; **def punchRectangles**; **def kickRectangles**; **def moveRectangles**
  + sve ove metode opisuju kako rectangleovi za borca izgledaju u određenom njegovom položaju i određuju pozicije tih rectangleova (**primjer** je na slici ispod)
* **def postavljanjeRectangleova()**
  + na borca aplicira one rectangleove u kojem se movementu/stanju borac nalazi
* **def animacije()**
  + crta one animacije borca ovisno u kojem se movementu/stanju borac nalazi
* **def flipanje\_slika()**
  + ovisno o movementu, flipa/mirrora surface borca ako zamjeni stranu sa drugim borcem (posljedično će se tako i rectanglovi borca crtati obrnuto)

## def crtanjeHealthaIImena()

* crtaju se 2 healthbara na ekranu, smanjuje se health prilikom udarca i u pozadini i graficki, ovisno o kolicini healtha mijenja se boja health bara
* crtaju se 2 stamina bara, koji se smanjuju ovisno o udaranju i tako stamini borca

Slika na kojoj se prikazuje tekst, snimka zaslona

Opis je automatski generiran

## def crtanjeRunde()

* crta se broj runde povrh ekrana u fight sceni i crta se trenutačni score u borbi

## def provjeraPozicijeZaObrnuto()

* provjerava jesu li borci na ekranu jedan drugome okrenuli leđa. Ako jesu mijenja varijable obrnuto u True ili False.

## def provjeraTrajanjaUdaraca()

* funkcija provjerava duration određene vrste attacka, i ako je predviđeni duration za taj attack istekao, gasi ga postavljajući odgovarajuću varijablu za attack na False

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

## def provjeraCrouchanja()

* postavlja boolean varijablu croucha na False i provjerava ako se pritisnula tipka za crouch, ako je onda se ta varijabla stavlja na True

Slika na kojoj se prikazuje tekst, snimka zaslona

Opis je automatski generiran

## def provjeraBlokiranja()

* postavlja boolean varijablu blocka na False i provjerava ako se pritisnula tipka za blocking, ako je onda se ta varijabla stavlja na True, umanjuje se stamina i započinje timer za povratak stamine

*isječak*

Slika na kojoj se prikazuje tekst, Font, snimka zaslona, crta

Opis je automatski generiran

## def provjeraKretanjaIKretanje

* gasi animacije hodanja/crouch\_hodanja i onda gleda je li stisnuta tipka za kretanje u lijevo ili u desno i ovisno o inputu započinje metodu .move()

## def provjeraSkokaISkakanje

* ako je stisnuta tipka za jump, a borac već nije u skoku sve movement varijable stavlja na False i započinje jump (i kao animaciju i kao sve promjene za borca koje se pokreću kada jump započne)

## def provjeraUdarcaIUdaranje

* ovisi u kojem se položaju borac nalazi (crouch, idle ili jump) pripadne varijable za određeni punch se settaju True/False (npr. animacija puncha, varijabla pocinjen\_damage koja prati je li se dmg pocinio i dr.) mijenja se stamina, provjerava se ako je udareno u blok i stavlja ga se na False

## def provjeraNogeINogatanje

* ovisi u kojem se položaju borac nalazi (idle ili jump) pripadne varijable za određeni kick se settaju True/False (npr. animacija kicka, varijabla pocinjen\_damage koja prati je li se dmg pocinio i dr.) mijenja se stamina, provjerava se ako je udareno u blok i stavlja ga se na False

*isječak*

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

## def provjeraDamagea()

* funkcija koja gleda je li određeni borac izveo neki attack, ako je gleda se je li udario u drugog borca i oduzima mu health, stunna ga; gleda se je li udario u block, ako je onda se drugom borcu skida stamina; ako je drugi borac mrtav ne registrira vise attack

*isječak*

Slika na kojoj se prikazuje tekst, snimka zaslona, softver

Opis je automatski generiran

## def provjeraJeLiStunned()

* gleda je li borac stunnan, ako je i prošlo je više od stun durationa, stun se kao i ostali attack movementi, koji su se možda pritisnuli za vrijeme stuna, gase postavljajući boolean varijable na False

## def provjeraZaRegeneriranjeStamine()

* postavlja se max vrijednost stamine, smanjuje se duration prije negoli počne regeneracija stamine i definira se diferencija za koju stamina raste s vremenom

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

## class Player()

* objekt ove klase bilježi ime igrača, broj winova/loseova, achievemente i broj profila
* **def postotak()**
  + metoda računa win rate
* **def ispisi()**
  + metoda ispisuje achievemente lika

## def read\_data()

* čita imena profila, scoreove i achievemente iz .txt datoteka za profile koji su odabrani za nadolazeću borbu i onda ih aplicira kao argumente u objektu iz Player klase

## def read\_data\_sve()

* čita imena profila, scoreove i achievemente iz .txt datoteka i onda ih aplicira kao argumente u objektu iz Player klase

## def poredak()

* funkcija poreda igrače u kodu onim redom kojim su zapravo odabrani u toku same igre

## def update\_achievementa()

* bilježi u .txt datoteku je li neki achievement ostvaren za određenog playera / upedatea

## def zadnji\_achievement()

* funkcija provjerava je li igrač postignuo prvih pet postignuća i ako je dodjeljuje mu se i šesto (finalno)

## def update\_score()

* bilježi u .txt datoteku update scorea za određenog playera

## class Button()

* definira se type objekta gumb koji ima sve funkcije gumba (boja, tekst, rectangle)
* **def dodaj()**
  + gumbu pridjeljuje broj jednak broju profila igrača
* **def update()**
  + crta gumb na screen
* **def checkForCollision()**
  + gleda je li miš na gumbu
* **def changeButtonColor()**
  + na hover mijenja boju gumba

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, operacijski sustav

Opis je automatski generiran

## def escape\_screen()

* radi pop-up screen kad se stisne Escape gumb, s mogućnošću izlaska ili ostanka u trenutnom dijelu igrice

Slika na kojoj se prikazuje tekst, snimka zaslona

Opis je automatski generiran

## def main()

* hijerarhijski najviša funkcija u kodu
* funkcija koja se prva starta kad se kod upali
* zajedno sa svojom grafičkom reprezentacijom u igrici (main menu), ali i ovako u kodu služi prelaženje u druge funkcije i druge dijelove igrice

## def crtajAchievemente()

* crta postignuća u screenu Postignuća ovisno o onom koje je otključano ili nije

## def imali()

* vraća True ukoliko postoje dva kreirana profila, odnosno False u suprotnom i True za to da nema kreiranih profila

## def postignuca\_nema\_profila()

* crta screen Postignuća dok nema niti jednog kreiranog profila

## def draw\_buttons()

* crta gumbe s imenom i brojem pobjeda/gubitaka u screenu Postignuća

## def prije\_postignuca()

* odnosi se na screen Postignuća, poziva crtanje gumbi s imenima i nakon pritiska na određeni gumb/profil prikazuju se postignuća tog lika

## def postignuca()

* funkcija je zapravo screen sa postignućima i dakle svim pripadnim stvarima (vrstama postignuća i dr.)

## def imenovanje\_profila()

* odnosi se na screen gdje se upisuju imena igrača (ne može se nastaviti dok se ne kreiraju barem dva profila)
* pamti upisano (to sprema i u .txt dokument) i prikazuje koji igrač je na kojem gumbu ili „Napravi profil“ ukoliko na gumbu već nema nekog profila

## def biranje\_profila()

* odnosi se na screen gdje se biraju 2 od ponuđenih kreiranih profila, odnosno dva igrača koja će igrati u nastavku (ne može se nastaviti dok se ne odaberu dva profila)
* pamti se koji igrač igra na kojem profilu

## def odabir\_borca1()

* odnosi se na screen kada prvi igrač bira svojeg borca s kojim će igrati
* pamti se kojeg je borca odabrao prvi igrač

## def odabir\_borca2()

* odnosi se na screen kada drugi igrač bira svojeg borca s kojim će igrati
* pamti se kojeg je borca odabrao drugi igrač

## def keybind\_screen1()

* odnosi se na screen kada prvi igrač treba odabrati keybind preset s kojim želi igrati (ima 3 mogućnosti, 3 preseta)
* ovisno o onom koji odabere na pritisak jednog od 3 gumba, iz rječnika mu se dodjeljuju tipke koje će koristiti za movement svojega borca

## def keybind\_screen2()

* odnosi se na screen kada drugi igrač treba odabrati keybind preset s kojim želi igrati (ima zapravo 2 mogućnosti jer je jedan preset zauzet od prvog igrača)
* ovisno o onom koji odabere na pritisak jednog od 2 preostala gumba, iz rječnika mu se dodjeljuju tipke koje će koristiti za movement svojega borca

## def odabir\_rundi()

* odnosi se na screen kada igrači pomoću 3 gumba odabiru na koliko rundi će se boriti (1,3,7)
* odabir se pamti i uvjetovat će duljinu trajanja borbe

## def findDamageRectangle()

* funkcija vraća True, ako je neki attack upotrjebljen na nekom od boraca, što je potrebno dalje za gledanje damagea i „def provjeraDamagea“

## def provjeraJeLiTkoDefeated()

* gleda je li nekom borcu health = 0, ako je onda se pojedine varijable setaju na True kako bi se u ostatku koda znalo da je neki borac izgubio
* nadodaje se score bod suprotnom borcu
* ispisuje se tko je pobijedio rundu i nakon određenog vremena (5sec) poziva funkciju za početak nove runde (reset\_igre())

Slika na kojoj se prikazuje tekst, snimka zaslona

Opis je automatski generiran

## def reset\_igre()

* funkcija ovisno o prethodno odabranom broju rundi, gleda je li cijeli fight/igra završila i ako je pamti pobjednika te varijablu „kraj\_igre“ stavlja na True čija će vrijednost onda uvjetovati završavanje igre dalje u kodu
* ako još nije ostvaren dovoljan score (više od pola pobijeđenih rundi), uvjetuje da se započne nova igra

## def igranje()

* funkcija u kojoj se zapravo odvija gameplay i sama srž igrice
* sadrži svu grafičku prezentaciju gameplayja kao i health i stamina barove
* pozivaju se sve funkcije i metode koje definiraju borce i njihove mehanike te provjere za kraj fighta/runde

## def winscreen()

* odnosi se na screen koji se pojavi kada određeni igrač pobjedi
* prikazuje se tko je pobijedio i nudi se opcija za povratak u main menu
* prikazuju se zanimljive statistike iz prethodne borbe