**POTAPANJE BRODOVA** – *dokumentacija*

Sadržaj

[PYGAME i PYTHON OPĆENITO 3](#_Toc118843550)

[class() 3](#_Toc118843551)

[convert\_alpha() 3](#_Toc118843552)

[def \_\_init\_\_(self) 3](#_Toc118843553)

[display.set\_caption() 3](#_Toc118843554)

[display.set\_mode() 3](#_Toc118843555)

[display.update() 4](#_Toc118843556)

[draw.rect() 4](#_Toc118843557)

[font.Font() 4](#_Toc118843558)

[font.Font.redner() 4](#_Toc118843559)

[global() 4](#_Toc118843560)

[image.load() 4](#_Toc118843561)

[os. path. join 4](#_Toc118843562)

[ime\_rect.colliderect() 4](#_Toc118843563)

[mask 4](#_Toc118843564)

[mask.clear() 4](#_Toc118843565)

[mask.from\_surface() 4](#_Toc118843566)

[mask.Mask.get\_at() 4](#_Toc118843567)

[Mijenjanje veličine fonta 5](#_Toc118843568)

[mixer.init() 5](#_Toc118843569)

[mixer.Sound() 5](#_Toc118843570)

[mouse.get\_pos() 5](#_Toc118843571)

[mouse.get\_pressed() 5](#_Toc118843572)

[pygame.init() 5](#_Toc118843573)

[Rect() 5](#_Toc118843574)

[Rect.collidepoint() 5](#_Toc118843575)

[screen.blit() 5](#_Toc118843576)

[sprite 5](#_Toc118843577)

[sprite.Group() 5](#_Toc118843578)

[sprite.Sprite() 5](#_Toc118843579)

[Super() 5](#_Toc118843580)

[Surface() 5](#_Toc118843581)

[Surface.fill() 5](#_Toc118843582)

[Surface.get\_rect() 6](#_Toc118843583)

[Surface.set\_alpha() 6](#_Toc118843584)

[time.Clock 6](#_Toc118843585)

[time.Clock.tick() 6](#_Toc118843586)

[time.sleep() 6](#_Toc118843587)

[transform.rotate() 6](#_Toc118843588)

[POTAPANJE BRODOVA- *funkcije* 6](#_Toc118843589)

[def LOADING\_SCREEN 6](#_Toc118843590)

[class Button 6](#_Toc118843591)

[def \_init\_ 6](#_Toc118843592)

[def update 6](#_Toc118843593)

[def checkForInput 6](#_Toc118843594)

[def changeColor 6](#_Toc118843595)

[def checkforClick 7](#_Toc118843596)

[class CrtanjeBrod 7](#_Toc118843597)

[def \_init\_ 7](#_Toc118843598)

[def mask\_collide 7](#_Toc118843599)

[def update 7](#_Toc118843600)

[def provjera\_ klika\_ sum \_broda 7](#_Toc118843601)

[class Brod 7](#_Toc118843602)

[def \_init\_ 7](#_Toc118843603)

[def rotacija\_poz\_90 & def rotacija neg\_90 7](#_Toc118843604)

[def collide 7](#_Toc118843605)

[def vrati\_nazad 7](#_Toc118843606)

[class Veliki\_ Xevi 7](#_Toc118843607)

[def\_init\_ 7](#_Toc118843608)

[def esc\_screen 7](#_Toc118843609)

[def gridA 7](#_Toc118843610)

[def gridB 7](#_Toc118843611)

[def crtanje\_ obruba\_hover 8](#_Toc118843612)

[def crtanje\_ pozadine\_postavljanje 8](#_Toc118843613)

[def provjera\_ hovera 8](#_Toc118843614)

[def rotate \_key\_animacija 8](#_Toc118843615)

[def čekanje\_za\_odabir 8](#_Toc118843616)

[def collide\_kvadrat 8](#_Toc118843617)

[def provjera 8](#_Toc118843618)

[def zapis 8](#_Toc118843619)

[def postavljanje\_ igracaA 8](#_Toc118843620)

[def postavljanje\_ igracaB 8](#_Toc118843621)

[def crtanje\_ fulanih\_ podrucja 8](#_Toc118843622)

[def zbrajanje\_pogodenih\_dijelova\_brodova 8](#_Toc118843623)

[def crtanje\_xeva 8](#_Toc118843624)

[def hoveranje\_animacija 9](#_Toc118843625)

[def promjena\_poz\_odabranog\_kvadrata 9](#_Toc118843626)

[def animacija\_ podogenog 9](#_Toc118843627)

[def animacija\_ fulanog 9](#_Toc118843628)

[def gadanje 9](#_Toc118843629)

[def crtanje\_ekrana\_ igranje 9](#_Toc118843630)

[def igranje\_ A \_ekran 9](#_Toc118843631)

[def igranje\_ B\_ ekran 9](#_Toc118843632)

[def end\_screen 9](#_Toc118843633)

[def linija\_playscore\_animacija 9](#_Toc118843634)

[def imenovanje\_ profila 9](#_Toc118843635)

[def biranje\_ profila 9](#_Toc118843636)

[def score \_screen 9](#_Toc118843637)

[def resetiranje\_prije\_igre 9](#_Toc118843638)

[def pauza\_prije\_promjene\_igraca 10](#_Toc118843639)

[def play 10](#_Toc118843640)

[def animacija\_bombi 10](#_Toc118843641)

[def animacija\_broda 10](#_Toc118843642)

[def animacija\_ more 10](#_Toc118843643)

[def main 10](#_Toc118843644)

# PYGAME i PYTHON OPĆENITO

## class()

* korisno pri organiziranju neke cjeline
* npr. **class** Adresa():

**def** \_\_init\_\_(self):

self.ime = ""

self.prezime = ""

self.ulica= ""

self.grad= ""

## convert\_alpha()

* mijenja format pixela
* convert\_alpha(Surface)
  + ako nije naveden nikakav surface/površina, novI će surface biti optimiziran za kopiranje na zaslon

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

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

## display.set\_caption()

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

## 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.update()

* ažurira površinu zaslona

## draw.rect()

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

## 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.redner()

* ispisuje tekst na novoj površini, odosno 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)
  + ako se želi postići pixelizirani izgled na mjesto anti-aliasinga piše se False

## 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)

## ime\_rect.colliderect()

* ime1\_rect.colliderect(ime2\_rect)
* provjerava je li se rect1 i rect2 preklapaju/sudaraju
* vraća True ako se preklapaju, a False ako ne

## mask

* modul za slikovne “maske“ => mask iz Surfacea **ne** uzima transparentne pixele (0) te ostavlja samo one pune(1)

## mask.clear()

* postavlja da svi bitovi u masku budu 0

## mask.from\_surface()

* radi mask object od zadanoga surfacea tako da uzme samo pixele koji nisu transparentni (slično micanju pozadine u photoshopu)
* tip koji vraća je također mask

## mask.Mask.get\_at()

* dobiva bit na zadanoj poziciji
* argument je tuple ili lista
* vraća 1 ako je bit postavljen, 0 ako nije, a Error je pozicija van maska
* get\_at((pozicija/koordinate))

## Mijenjanje veličine fonta

* textsurface = font. render("tekst", False, boja)

## 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 displayau obliku (x ,y)

## mouse.get\_pressed()

* vraća boolove
* ako se vrati True miš je trenutno pritisnut
* ako se vrati False miš nije trenutno pritsnut

## 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)

## Rect.collidepoint()

* pygame.Rect.collidepoint( položaj miša )
* vratit će True ako je miš izvan površine zaslona, a False ako je unutar

## 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.Sprite()

* osnovan class za vidljive objekte igre

## 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 (kao kod heksadecimalnih boja)

## 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.sleep()

* prekida funkciju za upisan broj sekundi
* time.sleep(vrijeme u sekundama)

## transform.rotate()

* rotira sliku
* pygame.transform.rotate(objekt (odnosno njegova pozicija), vrijednost kuta za koji se rotira)

# POTAPANJE BRODOVA- *funkcije*

## def LOADING­\_SCREEN

* crta loading screen na ekranu i pušta pozadinsku glazbu

## class Button

### def \_init\_

* definira se klasa gumba koja kao argumente prima tekst koji ce pisati na gumbu, njegovu veličinu i boju, dimenzije gumba, poziciju gumba na ekranu i njegovu boju te boju hovera

### def update

* updatea gumb na ekranu

### def checkForInput

* provjerava je li se miš sudara sa tim gumbom

### def changeColor

* farba gumb u drugu boju prilikom hovera

### def checkforClick

* povjerava je li se miš sudara s tim gumbom i mijenja neka bool varijable potrebne za rad def postavljanja koji ćemo objasniti kasnije

## class CrtanjeBrod

### def \_init\_

* crtaju se brodovi na Sumsung monitoru

### def mask\_collide

* provjerava sudara li se miš sa maskom tih brodova

### def update

* updatea masku broda na ekranu

### def provjera­\_ klika\_ sum \_broda

* provjera kojom se utvrđuje maska kojeg je broda pritisnuta

## class Brod

### def \_init\_

* definira se klasa broda koji se stavlja na grid

### def rotacija\_poz\_90 & def rotacija neg\_90

* rotira brod za pozitivno ili negativno s obzirom na taj koji se pozove

### def collide

* provjera je li se pritiskom miša on sudario baš s nekim od brodova

### def vrati\_nazad

* brod briše s grida ukoliko je on nepropisno postavljen

## class Veliki\_ Xevi

### def\_init\_

* definira klasu za crtanje velikih Xeva koji će se od početka postavljati na desni grid, a postat će vidljivi tek u kasnijem dijelu koda kada neki brod bude uništen

## def esc\_screen

* na pritisak tipke ESC poziva se ova funkcija i otvara se prozorcic koji nas pita hoćemo li možda izaći iz trenutnog dijela igre u kojem se nalazimo ili nećemo

#### zmaj

* pomoću njega radi esc screen
* zmaj je na pocetku funkcije false
* ako se klikne da onda je true i vraća se u glavni izbornik

## def gridA

* crta jedan od dvije mreže kvadratića te pripadne brojeve i slova uz rubove tog grida
* radi listu rectangleova kvadratića

## def gridB

* crta jedan od dvije mreže kvadratića te pripadne brojeve i slova uz rubove tog grida
* crta drugi grid

## def crtanje\_ obruba\_hover

* crta bijeli obrub oko sum broda prilikom hoveranja istog

## def crtanje\_ pozadine\_postavljanje

* crta pozadinu untar postavljanja

## def provjera\_ hovera

* crta zelene i crvene obrube oko kvadratića u ovisno o tome smije li se brod postaviti na te kvadratiće, u gridu prilikom postavljanja brodova na mrežu

## def rotate \_key\_animacija

* animacije sličice tipke R pored broda

## def čekanje\_za\_odabir

* u nju se dolazi nakon klika na neki sum\_brod dok se čeka da igrač postavi brod na polje
* u tom periodu ima mogućnost rotiranja broda tipkom R

## def collide\_kvadrat

* provjerava je li pritisnut kvadrat prilikom postavljanja broda na grid i ako se brod smije postaviti na kvadrat tamo se i postavlja

## def provjera

* provjerava sudara li se brod sa nekim drugim brodom prilikom postavljanja ili je možda stavljen tako da dio njega ispada iz grida

## def zapis

* kreiraju se liste\_imena\_kvadratića koja sadrži 100 članova, tj.100 listi u koje se bilježe imena kvadratića, a kao drugi član se još može nadodati i početno slovo broda ukoliko se on nalazi na nekom od tih kvadratića

## def postavljanje\_ igracaA

* vrti se dio igre u kojem igrač A postavlja svoje brodove

## def postavljanje\_ igracaB

* vrti se dio igre u kojem igrač B postavlja svoje brodove

## def crtanje\_ fulanih\_ podrucja

* crta područja na gridu gdje su oba igrača fulala brod

## def zbrajanje\_pogodenih\_dijelova\_brodova

* funkcija sprema u zasebne varijable zbroj koliko je puta neki brod pogođen
* informacije potrebne za funkciju crtanje\_xeva()

## def crtanje\_xeva

* funkcija provjerava jesu li pogođeni suparnički brodovi i crta x-eve ako jesu, crta jedan veliki x ako je cijeli brod uništen

## def hoveranje\_animacija

* funkcija provjerava nalazi li se miš iznad kvadratića, ako da onda nacrta poseban kvadratić da to označi

## def promjena\_poz\_odabranog\_kvadrata

* funkcija koja mijenja poziciju kvadrata koji označuje odabrani kvadratić na gridu

## def animacija\_ podogenog

* vrti animaciju ako je pogođen brod

## def animacija\_ fulanog

* vrti animaciju ako brod nije pogođen

## def gadanje

* funkcija u listi imena kvadrata upisuje x i ažurira rezultat

## def crtanje\_ekrana\_ igranje

* crta ekran u igranju

## def igranje\_ A \_ekran

* vrti se dio igre u kojem prvi igrač odabire poziciju na gridu koju želi gađati i vidi animaciju

## def igranje\_ B\_ ekran

* vrti se dio igre u kojem drugi igrač odabire poziciju na gridu koju želi gađati i vidi animaciju

## def end\_screen

* crta se end screen nakon što pobjedi neki od igrača, a na njemu se ispisuje tko je pobijedio i prikazuje se ukupan broj pobjeda za oba igrača

## def linija\_playscore\_animacija

* crta liniju/text cursor pri pisanju

## def imenovanje\_ profila

* upisivanje imena igrača

## def biranje\_ profila

* odabiru se dva igrača koji će igrati tu partiju

## def score \_screen

* u njemu se nalaze imena igrača te broj njihovih dosadašnjih pobjeda

## def resetiranje\_prije\_igre

* resetira listu rectangleova prije svakog igranja

## def pauza\_prije\_promjene\_igraca

* napravi pauzu od 3s između igrača

## def play

* spaja sve funkcije koje su potrebne za igru
* loop u kojem se izvršavaju sve funkcije za rad same igre, tj. mogućnost igranja

## def animacija\_bombi

## def animacija\_broda

## def animacija\_ more

* animacije na main/početnom ekranu

## def main

* prikazuje main/početni meni
* u njemu se pozivaju i preostale najvažnije funkcije poput playa, scorea...