

Igrica 6

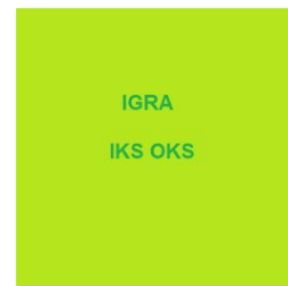
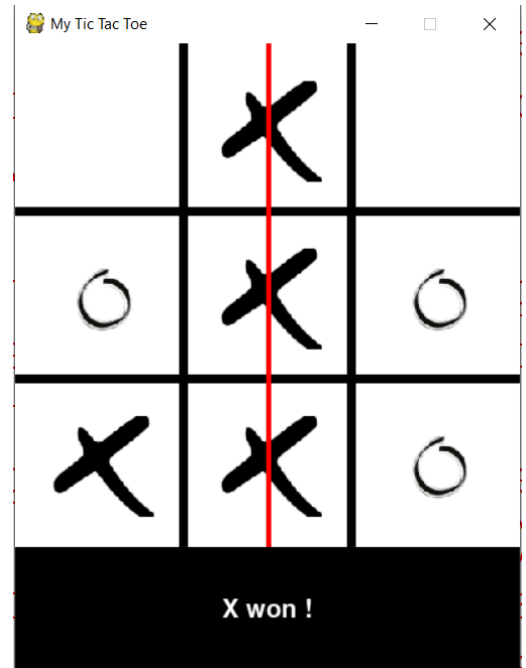
OPIS IGRE:

U ovoj igri igrač igra protiv drugog igrača na istoj igrici. Svaki klik predstavlja novi potez suprotnog igrača, i trenutno stanje se ispisuje u status baru na dnu igrice.

Na kraju igre ispisuje se da li je pobedio X ili O igrač, i pobjednička kombinacija se poveže crvenom linijom.

IDEJE ZA IMPLEMENTACIJU:

- Postojace mreza 3x3 gde na osnovu klika detektujemo koje polje je kliknuto
- Mrezu crtamo pomocu linija
- Ne crtamo x i o, vec stavljamo sliku na to polje
- Slike pripremiti npr u Paintu (velicina slika se transformise u kodu)
 - Jednu kao cover
 - Jednu za X
 - Jednu za O
- Potrebno da postoji board u kome cemo pamtiti sve pozicije koje su kliknute, kako bi se na kraju znalo da se iscrta linija crvene boje preko pobjednickog niza znakova (tj slika)
- Status bar samo ispisuje tekst



IMPLEMENTACIJA:

- Deo 1 - setap igre
 - **Osnovni setap prozora 400x500**
 - **Deo za igru, 3x3 board**
 - **Deo za ispis statusa**
 - **Priprema slika koje ce da se koriste za X i O znak na boardu**
- Deo 2
 - Detekcija klika musom
 - Dodavanje adekvatne slike
 - Blokiranje klika na istu poziciju
- Deo 3
 - Detekcija na boardu da li je bila pobjeda
 - Ispis statusa za sve slucajeve
- Deo 4

- Implementacija ostalih funkcionalnosti

DEO 1 IMPLEMENTACIJA:

```
# korak 1 - osnovni setup prozora
```

```
import pygame as pg
import pygamebg as pgbg
```

```
sirina = 400
visina = 400
```

```
BELA = (255, 255, 255)
CRNA = (0, 0, 0)
CRVENA = (255, 0, 0)
```

```
prozor= pgbg.open_window(sirina, visina + 100, "Iks oks")
```

```
# loadovanje slika
cover = pg.image.load("cover.png")
x_img = pg.image.load("x.png")
y_img = pg.image.load("o.png")
```

```
# podesavanje velicine slika
cover = pg.transform.scale(cover, (sirina, visina + 100))
x_img = pg.transform.scale(x_img, (80, 80))
o_img = pg.transform.scale(y_img, (80, 80))
```

```
#crtanje prozora
```

```
# hvatanje dogadjaja klika
```

```
pgbg.wait_loop()
```

```
# korak 2 - Crtanje mreze za iks oks
```

```
import pygame as pg
import pygamebg as pgbg
import time
```

```
def zapocniIgru():
    #dodamo cover sliku
    screen.blit(cover, (0, 0))

    #cover je prikazan 3 sekunde
    pg.display.update()
    time.sleep(3)
```

```
#prikaz mreze za iks oks
prozor.fill(BELA)
pg.draw.line(prozor,CRNA,(sirina // 3, 0),(sirina//3,visina), 7)
pg.draw.line(prozor,CRNA,(sirina//3*2,0),(sirina//3*2,visina), 7)
pg.draw.line(prozor,CRNA,(0,visina // 3),(sirina, visina // 3), 7)
pg.draw.line(prozor,CRNA,(0,visina//3*2),(sirina,visina//3*2), 7)
```

```
sirina = 400
```

```
visina = 400
```

```
# korak 3 - dodavanje promenljivih potrebnih za igru i za status bar
```

```
CRVENA = (255, 0, 0)
```

```
nereseno = False      # fleg da znamo da li je nereseno ili ne
pobednik = ""         # ova promenljiva menja vrednost u to ko je win
igrac = "X" # "O"     # ova promenljiva cuva info ko je na potezu
```

```
prozor = pgbg.open_window(sirina, visina + 100, "Iks oks")
```

```
# korak 4 - crtanje status bara i poruke
```

```
pg.draw.line(prozor,CRNA,(0,visina // 3),(sirina, visina // 3), 7)
pg.draw.line(prozor,CRNA,(0, visina//3*2),(sirina,visina//3*2), 7)
dodajStatus() # poziv fje na dnu iscrtavanja mreze iks oksa
```

```
def dodajStatus():
```

```
#definisanje poruke
if pobednik == "":
    poruka = igrac + " je na potezu!"
else:
    poruka = pobednik + " je pobedio!"
if nereseno:
    poruka = "Nereseno!"
```

```
#pravimo font
font = pg.font.SysFont("Arial", 30)
text = font.render(poruka, True, BELA)
```

```
# boji u crno, samo deo koji predstavlja drugi argument (ovo do
sada nije radjeno, da fill ima dva argumenta)
prozor.fill((0, 0, 0), (0, 400, 500, 100))
```

```
#blitujemo tekst po sirini po sredini
prozor.blit(text, ((sirina-text.get_width())// 2, 450))
```

```
sirina = 400
```

```
visina = 400
```

```
# korak 5 - posto ce se status stalno menjati, moramo da imamo satic i  
azuriranje displaya
```

```
nereseno = False  
pobednik = ""  
igrac = "X" # "O"
```

```
fps = 30  
sat = pg.time.Clock()
```

```
prozor = pgbg.open_window(sirina, visina + 100, "Iks oks")
```

```
# korak 6 - hvatanje svih dogadjaja, sada samo klika na X (update  
displaya pokrece video sistem, koji se gasi kroz sys.exit())
```

```
#crtanje prozora  
zapocniIgru()
```

```
# hvatanje dogadjaja klika
```

```
while True:  
    for event in pg.event.get():  
        if event.type == pg.QUIT:  
            pg.quit()  
            sys.exit()  
    pg.display.update()  
    sat.tick(fps)
```

```
pgbg.wait_loop()
```

```
# korak 7 - azuriranje update prozora za status, efekat ce se vidjeti  
kada se on zaista bude menjao
```

```
font = pg.font.SysFont("Arial", 30)  
text = font.render(poruka, True, BELA)  
prozor.fill((0, 0, 0), (0, 400, 500, 100))  
prozor.blit(text, ((sirina-text.get_width())// 2, 450))  
pg.display.update()
```

DEO 2 IMPLEMENTACIJA:

Korak 1 - dodavanje detekcije klika

```
if event.type == pg.QUIT:  
    pg.quit()  
    sys.exit()  
elif event.type == pg.MOUSEBUTTONDOWN:  
    klikIgraca() # upis X ili O
```

```
        # ovde kasnije dodati detekciju pobede nakon svakog klika
pg.display.update()
sat.tick(fps)
```

Korak 2 - dodati funkciju koristenu u prethodnom koraku - ucenik da proba sam da napravi ovu funkciju, samo da se objasni sta treba da bude ulaz u if na kraju

```
def klikIgraca():
    # mis pozicija
    x, y = pg.mouse.get_pos()

    # klik kolone
    if(x < sirina // 3):
        col = 1
    elif (x < sirina // 3 * 2):
        col = 2
    elif(x < sirina):
        col = 3
    else:
        col = -1

    # klik reda
    if(y < visina / 3):
        row = 1
    elif (y < visina / 3 * 2):
        row = 2
    elif(y < visina):
        row = 3
    else:
        row = -1

    if(row > 0 and col > 0):
        nacrtajXO(row, col)
```

```
def nacrtajXO(row, col):
    print(row)
    print(col)
```

Korak 3: testirati da li se na svakom kliku lepo očitaju kolona i red

Korak 4: crtanje X i O slike - isto ucenik moze sam da preračuna pozicije i da pozove blit, a pri tome stalno da menja ko je na potezu

```
def nacrtajXO(row, col):
    #print(row)
    #print(col)
```

```

global igrac

if row == 1:
    posy = 30
if row == 2:
    posy = sirina // 3 + 30
if row == 3:
    posy = sirina // 3 * 2 + 30

if col == 1:
    posx = 30
if col == 2:
    posx = visina // 3 + 30
if col == 3:
    posx = visina // 3 * 2 + 30
#print(igrac)
if igrac == "X":
    prozor.blit(x_img, (posx, posy))
    igrac = "O"
else:
    prozor.blit(o_img, (posx, posy))
    igrac = "X"
pg.display.update()

```

Korak 5: testirati

Korak 6: Azurirati status bar sa porukom ko je na potezu

```

if(row > 0 and col > 0):
    nacrtajXO(row, col)
    dodajStatus()

```

KORAK 7: Blokiranje klika na istu poziciju

- Diskusija kako ovo uraditi
- Treba doci do toga da je potrebno da se pamti 2dlista polja i da se upisuje sta je gde kliknuto, kroz tu listu da pratimo sta je popunjeno da bismo znali da blokiramo potez kada treba, i da znamo kada je bila pobjeda

KORAK 8: Dodati neophodne promenljive za tu listu

```

fps = 30
sat = pg.time.Clock()
XO = [["", "", ""], [ "", "", ""], [ "", "", ""]]

```

KORAK 9: Upisati vrednost kada se desi klik

```

def nacrtajXO(row, col):
    #print(row)
    #print(col)

    global igrac
    global XO

    ...
    #print(igrac)

    XO[row-1][col-1] = igrac
    print(XO)
    if igrac == "X":
        prozor.blit(x_img, (posx, posy))
    igrac = "O"
    ...

```

KORAK 10: Validacija klika

```

if(row > 0 and col > 0 and XO[row-1][col-1]== ""):
    nacrtajXO(row, col)
    dodajStatus()

```

DEO 3 MPLEMENTACIJA:

Korak 1 - Dodavanje koda kojim se poziva funkcija za proveru pobede nakon upisa X ili O

```

if(row > 0 and col > 0 and XO[row-1][col-1]== ""):
    #global XO
    nacrtajXO(row, col)
    dodajStatus()
    proveraPobede()

```

Korak 2 - definisanje te funkcije

```

def proveraPobede():

    global XO, pobednik, nereseno

    # redovi
    for red in range(0, 3):
        if((XO[red][0] == XO[red][1] == XO[red][2]) and
(XO[red][0] is not "")):
            pobednik = XO[red][0]
            tackaStart = (0 , (red + 1)*visina//3 -
visina//6)

```

```

        tackaKraj = (sirina, (red + 1)*visina//3 -
visina//6)
        pg.draw.line(prozor, CRVENA, tackaStart,
tackaKraj, 4)
        break

    # kolone
    for kolona in range(0, 3):
        if((XO[0][kolona] == XO[1][kolona] == XO[2][kolona]) and
(XO[0][kolona] is not "")):
            pobednik = XO[0][kolona]
            tackaStart = ((kolona + 1) * sirina // 3 -
sirina // 6, 0)
            tackaKraj = ((kolona + 1) * sirina // 3 - sirina
// 6, visina)
            pg.draw.line(prozor, CRVENA, tackaStart,
tackaKraj, 4)
            break

    # dijagonale
    if (XO[0][0] == XO[1][1] == XO[2][2]) and (XO[0][0] is not ""):

        # sa leva na desno
        pobednik = XO[0][0]
        pg.draw.line(prozor, CRVENA, (50, 50), (350, 350), 4)

    if (XO[0][2] == XO[1][1] == XO[2][0]) and (XO[0][2] is not ""):

        # sa desna na levo
        pobednik = XO[0][2]
        pg.draw.line(prozor, CRVENA, (350, 50), (50, 350), 4)

    if(all([all(red) for red in XO]) and pobednik == ""):
        nereseno = True
    dodajStatus()

```

OBJASNJENJE:

- Dodajemo promenljive koje su vec definisane u glavnom delu programa, da se i ovde mogu da menjaju preko global kljucne rece (XO, pobeda, nereseno)
- Proveravamo u boardu da li su ispisani isti znakovi po redovima, kolonama ili po dijagonalama
- U svakom slucaju definisemo ko je pobednik i tacke za iscrtavanje linije koja predstavlja smer pogodaka u nizu
- Zbog toga smo prosli put definisali board promenljivu u kojoj se cuva stanje boarda, to je ona linija koda: XO[row-1][col-1] = igrac
- Pri definisanju startne i krajnje tacke ostaviti da ucenik to izracuna

- Na kraju je i provera za nereseno, ako vise nema slobodnih polja i ako je pobednik ostao kao neresetovan, onda je fleg za nereseno potrebno podesiti
- Objasniti funkciju all(), to je novo

Korak 3 - Zabrana klika i igranja nakon sto se desila pobeda - dodajemo uslov kod funkcije klik, a sve ostalo tabujemo da bude deo tog ifa i da se klik desava samo u tom slucaju

```
def klikIgraca():
    if nereseno != True and pobednik == "":
        # mis pozicija
        x, y = pg.mouse.get_pos()
        # klik kolone
        if(x < sirina // 3):
            col = 1
        elif (x < sirina // 3 * 2):
            col = 2
        elif(x < sirina):
            col = 3
        else:
            col = -1
        # klik reda
        if(y < visina / 3):
            row = 1
        elif (y < visina / 3 * 2):
            row = 2
        elif(y < visina):
            row = 3
        else:
            row = -1
        if(row > 0 and col > 0 and XO[row-1][col-1]== ""):
            #global XO
            nacrtajXO(row, col)
            dodajStatus()
            proveraPobede()
```

Korak 4 - dodavanje provere da ako je kraj da se sve resetuje

```
elif event.type == pg.MOUSEBUTTONDOWN:
    klikIgraca()
    if nereseno == True or pobednik != "":
        setujPromenljive()
        zapocniIgru()
    pg.display.update()
    sat.tick(fps)
```

OBJASNJENJE:

- Objasniti sta je sve mogucnost da je kraj igre i koje promenljive nam to garantuju i sa kakvim vrednostima (nereseno da je true i da je pobednik ili X ili O)

Korak 5 - definisanje te funkcije za setovanje promenljive (fja zapocni igru vec postoji)

```
def setujPromenljive():
    global XO, pobednik, nereseno, igrac
    time.sleep(3)
    igrac = 'X'
    nereseno = False
    pobednik = ""
    XO = [["", "", ""], [ "", "", ""], [ "", "", ""]]
```

OBJASNJENJE:

- Sve promenljive se u ovoj funkciji resetuju, menja se njihova vrednost, tako da moraju da budu global

Korak 6 - posto se to poklapa sa pocetnim stanjem promenljivih, moze se malo srediti i refaktorisati kod

```
BELA = (255, 255, 255)
CRNA = (0, 0, 0)
CRVENA = (255, 0, 0)
```

```
global XO, pobednik, nereseno, igrac
setujPromenljive()
nereseno = False
pobednik = ""
igrac = "X" # "O"

fps = 30
sat = pg.time.Clock()
XO = [["", "", ""], [ "", "", ""], [ "", "", ""]]
```

```
prozor = pgbg.open_window(sirina, visina + 100, "Iks oks")
```

Korak 7 - random izbor da li partiju krece X ili O

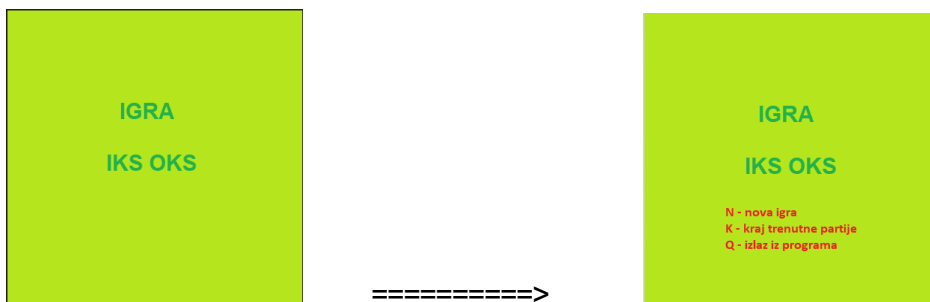
```
def setujPromenljive():
    global XO, pobednik, nereseno, igrac
    time.sleep(3)
    igrac = 'X' if (random.randint(0, 1) == 1) else 'O'
    nereseno = False
    pobednik = ""
    XO = [["", "", ""], [ "", "", ""], [ "", "", ""]]
```

DEO 4

- Implementacija ostalih funkcionalnosti
 - Meni na pocetku igre
 - **N** - nova igra
 - **K** - izlaz iz igre u meni / kraj trenutne igre
 - **Q** - zatvaranje skroz igrice

IMPLEMENTACIJA:

Korak 1 - Azuriranje slike cover u paintu



Korak 2 - Implementacija slova Q

```
while True:
    for event in pg.event.get():
        if event.type == pg.QUIT:
            pg.quit()
            sys.exit()
        elif event.type == pg.KEYDOWN:
            if event.key == pg.K_q:
                pg.quit()
                sys.exit()
        elif event.type == pg.MOUSEBUTTONDOWN:
            klikIgraca()
            if nereseno == True or pobednik != "":
                setujPromenljive()
                zapocniIgru()
```

OBJASNJENJE:

- Dodajemo proveru za KEYDOWN tip dogadjaja da bismo proverili koji je od mogucih tastera kliknut
- Na istom mestu cemo obraditi i ostale dogadjaje

Korak 2 - Implementacija slova k - kraj trenutne partije - u ovom slucaju treba da se prekine tekuca partija i da se prikaze meni

```

elif event.type == pg.KEYDOWN:
    if event.key == pg.K_q:
        pg.quit()
        sys.exit()
    if event.key == pg.K_n:
        novaIgra()
elif event.type == pg.MOUSEBUTTONDOWN:
    klikIgraca()

```

Korak 3 - Implementacija slova k - kraj trenutne partije - implementacija funkcije novalgra() - rasparcati postojece funkcije i srediti koja koju poziva

```

def zapocniIgru():

    prozor.blit(cover, (0, 0))
    pg.display.update()
time.sleep(3)

prozor.fill(bela)
pg.draw.line(prozor, crna, (sirina // 3, 0), (sirina // 3, visina), 7)
pg.draw.line(prozor, crna, (sirina // 3 * 2, 0), (sirina // 3 * 2, visina), 7)
pg.draw.line(prozor, crna, (0, visina // 3), (sirina, visina // 3), 7)
pg.draw.line(prozor, crna, (0, visina // 3 * 2), (sirina, visina // 3 * 2), 7)
statusBar()

def novaIgra():
    setujPromenljive()
    iscrtajPrazanIksOks()
    dodajStatus()

def iscrtajPrazanIksOks():
    prozor.fill(BELA)
    pg.draw.line(prozor, CRNA, (sirina // 3, 0), (sirina // 3, visina), 7)
    pg.draw.line(prozor, CRNA, (sirina // 3 * 2, 0), (sirina // 3 * 2, visina), 7)

    pg.draw.line(prozor, CRNA, (0, visina // 3), (sirina, visina // 3), 7)
    pg.draw.line(prozor, CRNA, (0, visina // 3 * 2), (sirina, visina // 3 * 2), 7)
    pg.display.update()

```

Korak 4 - igrati i naci bagove

Korak 5 - ispravka бага da je moguće krenuti klikati iks oks dok je još cover upaljen

```

BELA = (255, 255, 255)
CRNA = (0, 0, 0)
CRVENA = (255, 0, 0)

```

```

global XO, pobednik, nereseno, igrac, igraUtoku

```

```
igraUtoku = False
#setujPromenljive()
```

OBJASNJENJE

- Dodajemo fleg koji cemo koristiti da bi znali kada je klik dozvoljen
- Inicijalno setovanje promenljivih komenarisemo ili brisemo skroz, jer ce se to desiti tek kada se klikne N

Korak 5 - Nakon ovoga dodati setovanje flega na true

```
def setujPromenljive():
    global XO, pobednik, nereseno, igrac, igraUtoku
    igraUtoku = True
    time.sleep(1)
    igrac = 'X' if (random.randint(0, 1) == 1) else 'O'
    nereseno = False
    pobednik = ""
    XO = [["", "", ""], [ "", "", ""], [ "", "", ""]]
```

OBJASNJENJE

- Dodajemo true u delu za setovanje promenljivih
- Sleep moze da bude kraci umesto 3 da bude 1
- Sada ce da bude ogranicen klik na pocetku partije dok se ne klikne N, medjutim kada se posle 1 partije iscrta cover, problem je isti pa i to treba resiti

Korak 6 - dodavanje flega na kraju partije

```
elif event.type == pg.MOUSEBUTTONDOWN:
    if igraUtoku == True:
        klikIgraca()
        if nereseno == True or pobednik != "":
            igraUtoku = False
            #setujPromenljive()
            zapocniIgru()
```

OBJASNJENJE

- Dodajemo da se fleg prebaci
- Ubijamo setovanje promenljivih koje treba da se dese nakon sto se klikne N
- Ovim gubimo da se iscrta linija, tj da se vidi iscartavanje linije

Korak 7 - dodavanje u funkciji proveraPobede() sleep da bi se videlo iscartavanje

```
if(all([all(red) for red in XO]) and pobednik == ""):
    nereseno = True
    dodajStatus()
    time.sleep(1)
```

pg.display.update()

Korak 8 - testirati

Korak 9 - ne radi slovo k

```
if event.key == pg.K_q:  
    pg.quit()  
    sys.exit()  
if event.key == pg.K_n:  
    novaIgra()  
if event.key == pg.K_k:  
    zapocniIgru()
```