

Croatle

DOKUMENTACIJA

Robert Zenzerović
GIMNAZIJA ANDRIJE MOHOROVIČIĆA, 2.5 | RIJEKA

Sadržaj

Uvod	3
Opis rada	3
Početak igre	3
Pogađanje županije.....	3
Tehničke informacije.....	4
Preuvjeti, preporuke i podržanost	4
Objašnjenje rješenja.....	5
Screencast.....	7

Tablica slika

Slika 1: Korisničko sučelje nakon pokretanja	3
Slika 2: netočan unos županije	4
Slika 3: točan unos županije	4
Slika 4: pet uzastopnih krivih pokušaja	4
Slika 5: primjer potrebne postavke na Windows 11 (sličan postupak i za W10)	4
Slika 6: brojevnne varijable.....	5
Slika 7: ovisnost liste "values" o n.....	5
Slika 8: kod za grafičko sučelje.....	6
Slika 9: početak funkcije checking().....	6
Slika 10: primjer "testa" za provjeru rješenja	6

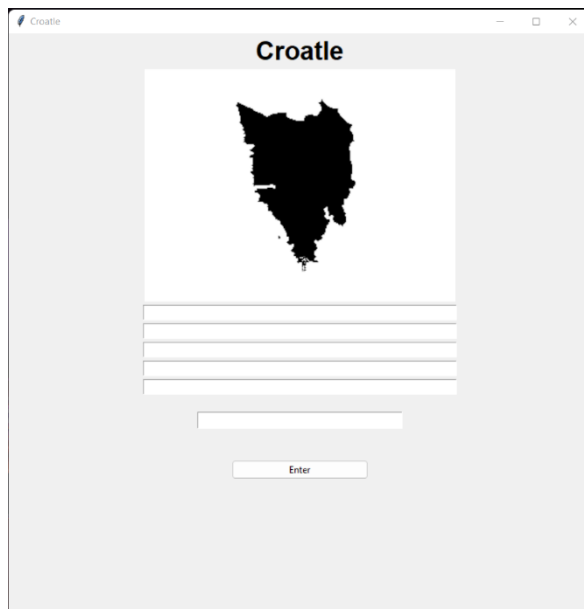
Uvod

Igru pogađanja Croatle sam osmislio uz inspiraciju iz online igre pogađanja [Worldle](#). Preuzeo sam koncept koju sam preinačio i zatim napisao pomoću Pythona. Odlučio sam koncept preinačiti na način da se u igrici pogađaju županije Republike Hrvatske umjesto država u svijetu. Ova igrica ima edukativnu svrhu jer može pomoći u učenju teritorijalnog ustroja RH.

Opis rada

Početak igre

Na početku igre program nasumično bira županiju koju će korisnik pogađati. Prikazat će se korisničko sučelje koje sadrži sliku županije koja se pogađa, pet polja za prikaz pokušaja koje je korisnik unio, prostora za unos pokušaja i gumb za potvrdu pokušaja.

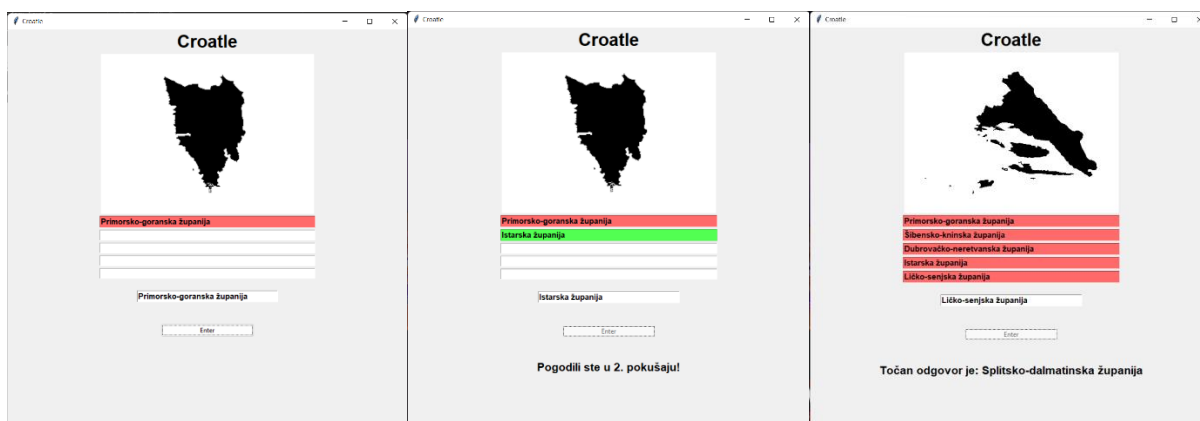


Slika 1: Korisničko sučelje nakon pokretanja

Pokretanje igre je potrebno izvršiti putem programa IDLE za Python.

Pogađanje županije

Unos županije se vrši pomoću *entry* polja koji se nalazi iznad tipke „Enter“. Nakon što se unese željena županija (pazeći na poredak slova i znakova), popunit će se jedno od polja koja služe za prikaz prošlih pokušaja. Ukoliko je pokušaj točan, polje će biti obojano zelenom bojom te će se ispod tipke „Enter“ (koja je onemogućena) prikazati podatak da je pogođena županija i broj pokušaja koje je igrač trebao. U slučaju da županija nije pogođena, polje će biti označeno crvenom bojom i igrač će imati mogućnost birati idući pokušaj. **Igrač može pogađati najviše pet puta**, a ukoliko ne pogodi nakon toga, igra će prikazati odgovor na sličan način kao za pogođenu županiju.



Slika 2: netočan unos županije

Slika 3: točan unos županije

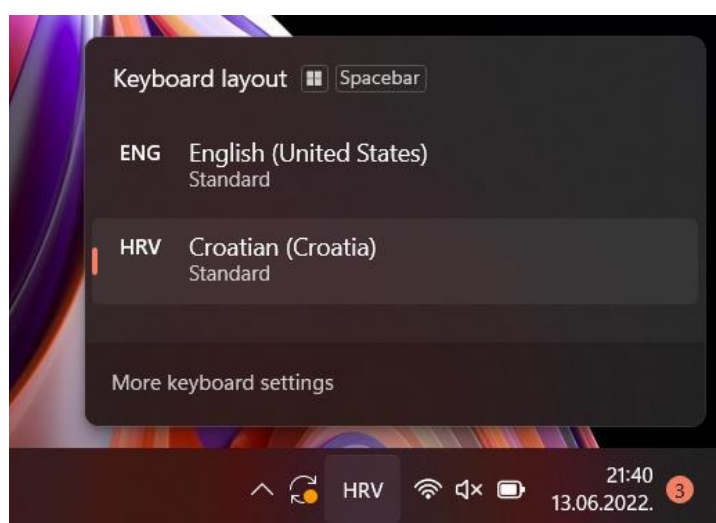
Slika 4: pet uzastopnih krivih pokušaja

Tehničke informacije

Preuvjeti, preporuke i podržanost

- Python 3 (preporučljivo verzija ≥ 3.8)
 - Python Image Library (paket *pillow*) – instalacija putem Naredbenog retka ili Terminala (ovisno o operacijskom sustavu):
 - `pip3 install pillow`
 - Paketi uključeni u Python: `tkinter`, `random`

Ukoliko koristite operacijski sustav Windows, pobrinite se da je jezik tipkovnice postavljen na hrvatski jezik, tj. Croatian (Croatia) zbog problema s prepoznavanjem teksta koji bi se mogli dogoditi u suprotnome.



Slika 5: primjer potrebne postavke na Windows 11 (sličan postupak i za W10)

Program je testiran za funkcionalnost u kombinacijama:

- Ubuntu 20.04.4 LTS i Python 3.8
- Windows 10 21H2 i Python 3.9
- Windows 11 21H2 i Python 3.10.

Ovaj program bi trebao raditi i na ostalim kombinacijama verzija Pythona i operacijskih sustava.

Objašnjenje rješenja

```
# guesses left (if the value goes up to 5, the game should show the answer and not allow further input
x = 0
# random selection
n = randint(0,20)
```

Slika 6: brojevne varijable

Za početak programa se uvode dvije brojevne varijable, x i n . Varijabla x se mijenja pomoću funkcije koja će biti objašnjena, a n je nasumično birana varijable pomoću naredbe `randint(0,20)`.

```
if n == 0:
    values = ['images/pgz.png', 'Primorsko-goranska županija']
if n == 1:
    values = ['images/iz.png', 'Istarska županija']
if n == 2:
    values = ['images/gz.png', 'Grad Zagreb']
if n == 3:
    values = ['images/zz.png', 'Zagrebačka županija']
if n == 4:
    values = ['images/zdz.png', 'Zadarska županija']
if n == 5:
    values = ['images/sz.png', 'Šibensko-kninska županija']
if n == 6:
    values = ['images/sdz.png', 'Splitško-dalmatinska županija']
if n == 7:
    values = ['images/dnz.png', 'Dubrovačko-neretvanska županija']
if n == 8:
    values = ['images/lasz.png', 'Ličko-senjska županija']
if n == 9:
    values = ['images/kz.png', 'Karlovačka županija']
if n == 10:
    values = ['images/smz.png', 'Sisačko-moslavačka županija']
if n == 11:
    values = ['images/kzz.png', 'Krapinsko-zagorska županija']
if n == 12:
    values = ['images/vz.png', 'Varaždinska županija']
if n == 13:
    values = ['images/mz.png', 'Međimurska županija']
if n == 14:
    values = ['images/kkz.png', 'Koprivničko-križevačka županija']
if n == 15:
    values = ['images/bbz.png', 'Bjelovarsko-bilogorska županija']
if n == 16:
    values = ['images/vpz.png', 'Virovitičko-podravska županija']
if n == 17:
    values = ['images/psz.png', 'Požeško-slavonska županija']
if n == 18:
    values = ['images/bpz.png', 'Brodsko-posavska županija']
if n == 19:
    values = ['images/obz.png', 'Osječko-baranjska županija']
if n == 20:
    values = ['images/vsz.png', 'Vukovarsko-srijemska županija']
```

Slika 7: ovisnost liste "values" o n

Pomoću varijable n se bira prikaz slike županije i ime županije koja se pogađa. Lista je izrađena na način da za `values[0]` se vrati vrijednost slike, a za `values[1]` vrati željeno rješenje županije.

```

gui = tk.Tk()
gui.geometry("750x750")
gui.grid_columnconfigure(0, weight=1)
gui.wm_title('Create')
title = tk.Label(gui, text='Create', font=('Arial', 25, 'bold'))
title.grid(row=1, column=0, columnspan=2)

img= Image.open(values[0])
resize_img = img.resize((400,300))
image_zup = ImageTk.PhotoImage(resize_img)

img_zup = tk.Label(gui, image = image_zup)
img_zup.grid(row=2, column=0, columnspan=2)

guessval = ['', '', '', '', '']

text_box1 = tk.Text(gui, height=1, width=50)
text_box1.insert('end', guessval[0])
text_box1.config(state='disabled')
text_box1.grid(row=3, column=0, columnspan=2, pady=2)

text_box2 = tk.Text(gui, height=1, width=50)
text_box2.insert('end', guessval[1])
text_box2.config(state='disabled')
text_box2.grid(row=4, column=0, columnspan=2, pady=2)

text_box3 = tk.Text(gui, height=1, width=50)
text_box3.insert('end', guessval[2])
text_box3.config(state='disabled')
text_box3.grid(row=5, column=0, columnspan=2, pady=2)

text_box4 = tk.Text(gui, height=1, width=50)
text_box4.insert('end', guessval[3])
text_box4.config(state='disabled')
text_box4.grid(row=6, column=0, columnspan=2, pady=2)

text_box5 = tk.Text(gui, height=1, width=50)
text_box5.insert('end', guessval[4])
text_box5.config(state='disabled')
text_box5.grid(row=7, column=0, columnspan=2, pady=2)

entry = tk.Entry(gui, font=('Arial', 11, 'bold'))
entry.grid(row=8, column=0, columnspan=2, pady=20, ipadx=50)
button = ttk.Button(gui, text="Enter", command=checking)
button.grid(row=9, column=0, columnspan=2, pady=20, ipadx=50)

gui.mainloop()

```

Slika 8: kod za grafičko sučelje

Grafičko sučelje je riješeno pomoću Tkintera korištenjem elemenata *Label*, *Text* (kutija za text), *Entry* (polje za unos teksta) i *Button*. Stvorena je i lista `guessval` koja služi za bilježenje unosa. Tipka `button` poziva funkciju `checking()` koja će biti pokrivena.

```

def checking():
    text=entry.get()
    global x
    global guessval
    global values
    for i in range(0, 21):
        accept = False
        if x < 5:
            if zupanje[i].lower() == text.lower():
                guessval[x] = zupanje[i]
                accept = True
                break

```

Slika 9: početak funkcije `checking()`

Varijabla `text` je definirana za dohvrat podatka sa ulaza na grafičkom sučelju. Nakon toga se definiraju globalne varijable `x`, `guessval`, `values`. Lista `zupanje` služi kao popis valjanih županija za unos u igru. Poslije toga je napravljena petlja koja provjerava valjanost unesene županije i mijenja varijablu `accept` ukoliko je valjan unos.

```

    if x == 0 and accept == True:
        if text.lower() == values[1].lower():
            text_box1 = tk.Text(gui, height=1, width=50, font=('Arial', 11, 'bold'), bg='#52FF52', fg='#000000')
            text_box1.insert('end', guessval[x])
            text_box1.config(state='disabled')
            text_box1.grid(row=3, column=0, columnspan=2, pady=2)
            button.config(state='disabled')
            notice = tk.Label(gui, text='Pogodili ste u 1. pokušaju!', font=('Arial', 16, 'bold'))
            notice.grid(row=10, column=0, columnspan=2, pady=20, ipadx=50)
        else:
            text_box1 = tk.Text(gui, height=1, width=50, font=('Arial', 11, 'bold'), bg='#FF6A6A', fg='#000000')
            text_box1.insert('end', guessval[x])
            text_box1.config(state='disabled')
            text_box1.grid(row=3, column=0, columnspan=2, pady=2)

```

Slika 10: primjer "testa" za provjeru rješenja

Prikazana je provjera za slučaj prvog pokušaja na slici 10 (no na sličan način rade i ostale provjere). Ukoliko je vrijednost unesenog teksta pretvorenog u mala slova jednaka vrijednosti rješenja u malim slovima, rješenje se prihvaća i smješta na određeno polje ovisno o pokušaju u zelenoj boji te se prikazuje poruka da je točno rješenje. Ukoliko to nije slučaj, rješenje se smješta na polje u crvenoj boji i moguće je dalje pokušati ukoliko nije zadnji pokušaj. Nakon svakog pokušaja varijabli x se pribraja broj 1.

Screencast

Primjere rada ove igrice možete pogledati na idućim poveznicama:

- [Pogađanje županije s točnim rješenjem](#)
- [Pogađanje županije bez točnog rješenja](#)