



Zavod sv. Stanislava

Škofiska klasična gimanzija

PROGRAMIRANJE V PYTHONU

PROGRAMIRANJE IGRE KRIŽCI IN KROŽCI

Maturitetna seminarska naloga iz informatike

Kandidat: Jan Zgonc

Mentor: prof. Valentin Sojar

Ljubljana, april 2026

POVZETEK

Za seminarsko nalogo sem izdelal preprosti program v programski jeziku Python. Izdelal sem program, s katerim lahko posameznik proti računalniku igra igro križcev in krožcev. V programu lahko posameznik izbira med tremi težavnostnimi stopnjami, od preprostega naključnega igranja do naprednega algoritma Minimax. Igra vključuje funkcionalnosti kot so obdelava uporabniškega vnosa, preverjanje zmagovalca ali neodločenega izida ipd...Pri delu sem se naučil uporabljati razne algoritme, osnovne znanje zank in uporabe knjižnic.

ABSTRACT

For my seminar paper, I developed a simple program in the Python programming language. I created a program that allows a user to play a game of Tic-Tac-Toe against the computer. The program offers three difficulty levels, ranging from simple random moves to the advanced Minimax algorithm. The game includes functionalities such as handling user input, checking for a winner, and detecting a draw. During this project, I learned how to use various algorithms, basic loop structures, and external libraries.

KLJUČNE BESEDE

Python, križci in krožci, algoritem Minimax

KAZALO VSEBINE

1	UVOD
1.1	PREDSTAVITEV PROBLEMA
1.2	UPORABLJENA STROJNA OPREMA.....
1.3	UPORABLJENA PROGRAMSKA OPREMA
2	TEORETIČNA REŠITEV.....
2.1	OPIS IGRE
2.2	PRAVILA IGRE.....
2.3	ZAKLJUČEK IGRE.....
3	PRAKTIČNA REŠITEV
3.1	RAZLAGA PROGRAMA S KOMENTARJI
3.2	OMEJITVE IZVEDENE REŠITVE
4	SKLEP.....
4.1	NAVODILA ZA UPORABO
4.2	OCENA REŠITVE.....
4.3	MOŽNE IZBOLJŠAVE
4.4	PORABA ČASA.....
5	VIRI.....

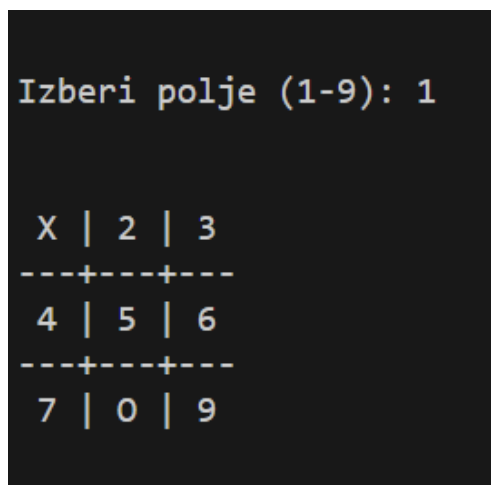
KAZALO SLIK

Slika 1: Digitalna različica igre križci in krožci	
Slika 2: Igra križci in krožci nasplošno	
Slika 3: Uvoženi knjižnici, začetek kode	
Slika 4: Prva in druga stopnja programa	
Slika 5: Preverjanje zmagovalca v programu	
Slika 6: Definiranje igralnega polja v programu.....	
Slika 7: Izbira težavnosti v programu	
Slika 8: Izbira polja v programu	

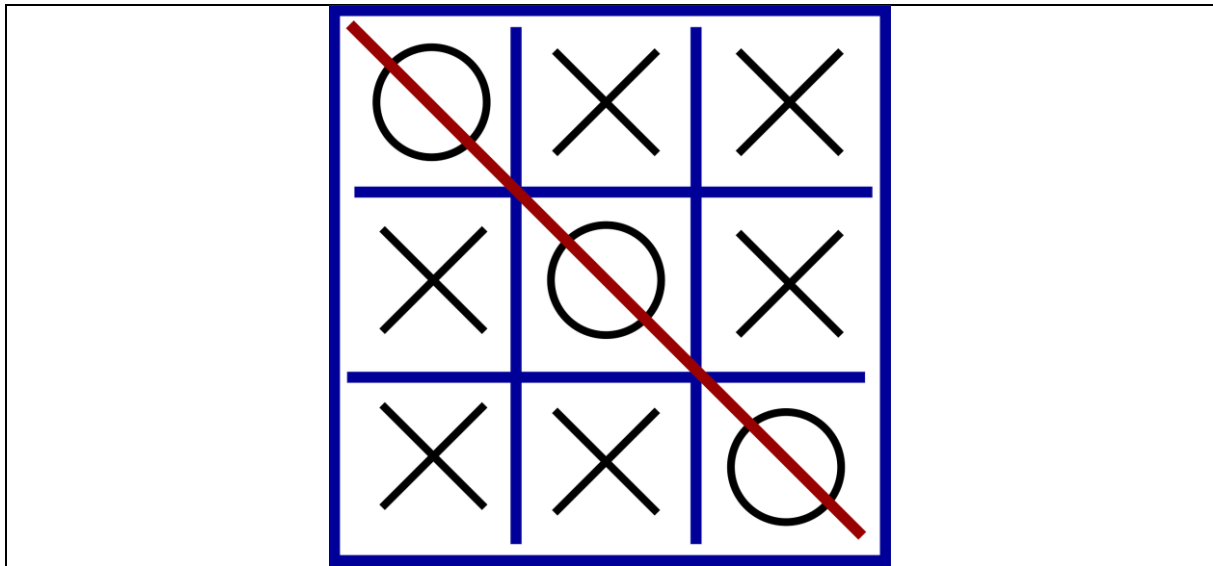
1.UVOD

1.1 PREDSTAVITEV PROBLEMA

Naredil bom program s katerim lahko posameznik igra igro križci in krožci proti računalniku, tako da lahko nadgradi svoje veščine v igri. Izdelal bom program, ki ne omogoča le osnovnega igranja, ampak tudi različne stopnje težavnosti, pri katerih se računalnik različno odziva na poteze igralca. V program bom tudi implementiral algoritem, ki omogoča pametno odločanje računalnika, tako da igra optimalno in predstavlja izziv za uporabnika. Naredil bom polje v terminalu, kjer bo lahko uporabnik izbiral polje v katerega bo vnašal svoje poteze, program pa bo hkrati vnašal svoje poteze.



Slika 1: Digitalna različica igre križci in krožci



Slika 2: Igra križci in krožci nasplošno

1.2 UPORABLJENA STROJNA OPREMA

Model	ASUS ROG Strix G513RM
Pomnilnik	16 GB DDR5
Procesor	AMD Ryzen 7 6800H
Grafična kartica	NVIDIA GeForce RTX 3060
Kapaciteta diska	1 TB SSD

1.3 UPORABLJENA PROGRAMSKA OPREMA

Operacijski sistem	Windows 11
Tip sistema	64-bit
Jedro	Linux kernel
Grafično okolje	Gnome

2. TEORETIČNA REŠITEV

2.1 OPIS IGRE

Igra Križci in krožci je preprosta igra za dva igralca, ki se igrata na mreži velikosti 3×3. V tej izvedbi igralec igra proti računalniku. Igralec uporablja znak X, računalnik pa znak O. Program omogoča izbiro med tremi težavnostnimi. Na najlažji stopnji računalnik izbira poteze naključno, na srednji stopnji že upošteva osnovno strategijo, na najtežji stopnji pa uporablja Minimax algoritem. Zmaga tisti, ki pri vodoravno, navpično ali horizontalno dobi tri znake v zaporedju.

2.2 PRAVILA IGRE

Igra se igra v mreži 3x3, računalnik in uporabnik tekmujeta kdo prvi vodoravno, navpično ali horizontalno dobi tri znake v zaporedju. Če so vsa polja zapolnjena in ni zmagovalca, se igra konča z neodločenim izidom.

2.3 ZAKLJUČEK IGRE

Igra se zaključi z zmago programa, z zmago uporabnika ali pa z izenačenjem, kar pa program tudi sporoči. Ko se igra zaključi lahko uporabnik še enkrat zažene program in igra se ponovi

3. PRAKTIČNA REŠITEV

3.1 RAZLAGA PROGRAMA S KOMENTARJI

V programu naložim dve knjižnici *random* in *math*. Ustvarim mrežo, katera predstavlja polje križcev in krožcev. V program vstavim tri stopnje, v prvi stopnji program naključno izbere svoje poteze, druga stopnja ima že malo več strategija, tretja stopnja pa uporablja *Minimax* umetno inteligenco. V kodi tudi preverjam zmagovalca igre, oziroma izenačenje.

```
import random
import math
# brez zunanjih knjižnic

def izpisi_plosco(plosca):
    print("\n")

    for i in range(0, 9, 3):
        print(" " + plosca[i] + " | " + plosca[i + 1] + " | " + plosca[i + 2])

        if i < 6:
            # okrasni deli za terminal
            print("----+----+----")

    print("\n")
```

Slika 3: Uvoženi knjižnici, začetek kode

```

#Easy AI
def lahek_ai(plosca):
    return random.choice(pridobi_mozne_poteze(plosca))

#MEDIUM AI
def srednji_ai(plosca):
    poteze = pridobi_mozne_poteze(plosca)

    # 1) Najprej preveri, ce lahko AI takoj zmaga.
    for poteza in poteze:
        plosca[poteza] = 'O'
        if preveri_zmagovalca(plosca, 'O'):
            plosca[poteza] = str(poteza + 1)
            return poteza
        plosca[poteza] = str(poteza + 1)

    # 2) Ce ne gre, blokira igralca.
    for poteza in poteze:
        plosca[poteza] = 'X'
        if preveri_zmagovalca(plosca, 'X'):
            plosca[poteza] = str(poteza + 1)
            return poteza
        plosca[poteza] = str(poteza + 1)

    # 3) Drugace random.
    return random.choice(poteze)

```

Slika 4: Prva in druga stopnja programa

```

def igranj():
    plosca = []

    for i in range(9):
        plosca.append(str(i + 1))

    ai = izberi_tezavnost()

    while True:
        izpisi_plosco(plosca)

        # Poteza igralca
        poteza = pridobi_potezo_igralca(plosca)
        plosca[poteza] = 'X'

        if preveri_zmagovalca(plosca, 'X'):
            izpisi_plosco(plosca)
            print("Zmagal si!")
            break

        if je_neodloceno(plosca):
            izpisi_plosco(plosca)
            print("Noben ni zmagal!")
            break

```

Slika 5: Preverjanje zmagovalca v programu

```

def izpisi_plosco(plosca):
    print("\n")

    for i in range(0, 9, 3):
        print(" " + plosca[i] + " | " + plosca[i + 1] + " | " + plosca[i + 2])

        if i < 6:
            # okrasni deli za terminal
            print("---+---+---")

    print("\n")

def preveri_zmagovalca(plosca, igravec):
    # vse kombinacije, kjer lahko nekdo zmaga
    zmagovalne_pozicije = [
        [0, 1, 2], [3, 4, 5], [6, 7, 8],
        [0, 3, 6], [1, 4, 7], [2, 5, 8],
        [0, 4, 8], [2, 4, 6]
    ]

```

Slika 6: Definiranje igralnega polja v programu

3.2 OMEJITVE IZVEDENE REŠITVE

Program kar obsežno upodobi igro, vendar je omejen na to da lahko uporabnik igra samo proti programu in ne z drugimi uporabniki. Poleg tega program ne more šteti rezultat več iger hkrati, ampak samo igre, ki je bila ravnokar odigrana.

4. SKLEP

4.1 NAVODILA ZA UPORABO

Ko zaženete program se vam bo na terminalu pokazala možnost izbiranja težavnosti. Ko s črko izberete težavnost se bo pričela igra in pokazala se bodo polja. Na ta polja s številkami potem vpisujete svoje poteze in program bo hkrati vpisoval tudi svoje. Ko se igra konča vam bo program to tudi sporočil in povedal rezultat.

```
Izberi težavnost:  
1 - Lahko  
2 - Srednje  
3 - Tezko  
> 2
```

Slika 7: Izbira težavnosti v programu

```
1 | 2 | 3  
-+--+  
4 | 5 | 6  
-+--+  
7 | 8 | 9
```

Slika 8: Izbira polja v programu

4.2 OCENA REŠITVE

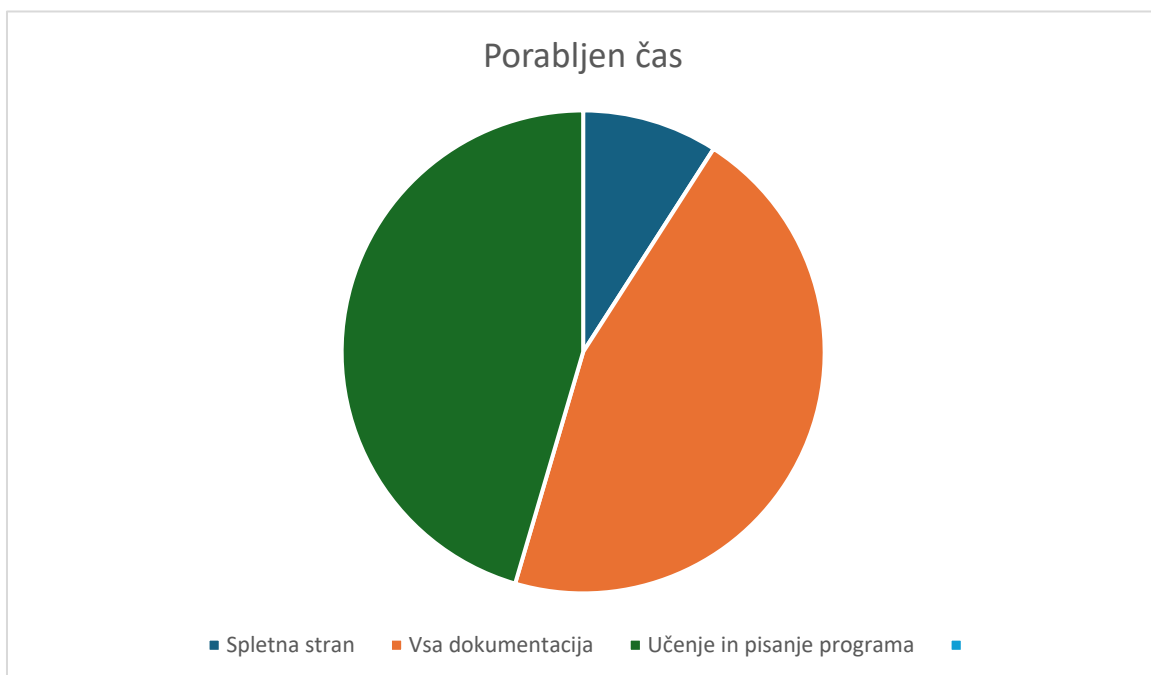
Rešitev je dokaj dobro strukturirana, zelo mi je všeč koncept nivojev v programu. Kljub temu da je program dokaj preprost ponuja zelo interaktivno izkušnjo in krepi znanje o igranju igrice križci in krožci. Poleg tega pa sem se ob pisanju programa tudi veliko naučil

4.3 MOŽNE IZBOLJŠAVE

Lahko bi izboljšal to, da bi igro naredil tako, da bi lahko igrala sva igralca med sabo in pa tudi to da bi program štel rezultat prejšnjih iger, ne samo ene. Lahko bi si tudi izbral igro, ki je bolj kompleksna od igre križcev in krožcev. Glede na moje znanje programa pa bi bila tu manjša težava.

4.4 PORABA ČASA

Za izdelovanje spletne strani sem porabil kakšno uro. Za pripravljane in učenje programa kakšne 3 ure, za samo pisanje kode pa še kakšni dve uri. Za pisanje dokumentacije in osnutkov sem skupaj porabil kakšnih 5 ur. Skupaj sem porabil kakšnih 11 zbranih ur.



Graf 1: Porabljen čas

5. VIRI

- OpenAI: The ChatGPT (online). [Uporabljeno 12. 4. 2026]. Dostopno na <https://chatgpt.com>.
- Python tutorial (online). [Uporabljeno 12. 4. 2026]. Dostopno na <https://www.w3schools.com/python/default.asp>.
- Šolska spletna učilnica (online). [Uporabljeno 12. 4. 2026]. Dostopno na <https://ucilnica.stanislav.si>