

TEHNIČKA DOKUMENTACIJA

SUDOKU

Autor: Marieta Jovančević – Zekić, 4.5

Mentor: prof. Goran Boneta

SADRŽAJ

1. UVOD.....	1
1.1. Riječi o autoru.....	1
1.2. Ideja projekta.....	1
1.3. O igri.....	1
2. DETALJAN OPIS RADA.....	2
2.1. Programski jezik.....	2
2.2. Početak igrice.....	2
2.3. Unos brojeva.....	3
2.4. Završetak igrice.....	4
2.5. Ploče.....	5
3. BUDUĆNOST PROJEKTA.....	6
3.1. Pogreške.....	6
3.2. Više razina igrice.....	6
3.3. Ploče za rješavanje.....	6

1. UVOD

1.1. Riječi o autoru

Moje ime je Marieta Jovančević – Zekić, učenica sam 4.5 razreda Gimnazije Andrije Mohorovičića u Rijeci, prirodoslovno – matematički smjer.

Priliku naučiti programirati imala sam u školi, te je ovaj projekt primjena svega dosad naučenog, a ujedno i dodatno upoznavanje i razumijevanje programskog jezika Python.

1.2. Ideja projekta

Sudoku je igra koju u zadnje vrijeme, s obzirom na situaciju u kojoj se nalazimo, igram dosta često. Instalirala sam aplikaciju na mobitel, te ponekad svo svoje slobodno vrijeme znam provesti rješavajući Sudoku.

Tako sam, razmišljajući o mogućnostima za projekt, došla na ideju da pokušam sama napraviti Sudoku.

1.3. O igri

Sudoku je vrsta matematičke zagonetke čije je rješavanje temeljeno na logici. Sastoji se od jednog velikog kvadratnog polja, podijeljenog na 81 manjih kvadratića. Unutar kvadratnog polja označeno je i 9 odjeljaka velikih 3x3 polja.

Cilj igre je ispuniti sva polja brojevima od 1 do 9, s time da se svaki broj smije pojaviti točno 9 puta. Dakle, jedan broj se smije pojaviti samo jednom u svakom retku, svakom stupcu i svakom odjeljku od 3x3 polja. Na početku igre, otkriveni su određeni brojevi, a rješavač mora otkriti gdje se nalaze svi ostali brojevi i kako su raspoređeni.

Igra potječe iz Japana gdje se igra već 1000 godina, dok u modernom svijetu postaje popularna zahvaljujući Howardu Garnsu koji ju je obnovio u američkom magazinu *Dell Magazines*.

U Sudoku se primjenjuju različite strategije, često više njih istovremeno.

- Metoda eliminacije polja – Ovom metodom se pokušava pronaći polje u nekom retku, stupcu ili 3x3 kvadratu gdje može biti neki broj.
- Metoda eliminacije brojeva – Ovom metodom pokušava se otkriti koji se broj nalazi u nekom polju. Promatra se redak, stupac i 3x3 kvadrat za koje pokušavamo otkriti broj.

2. DETALJAN OPIS RADA

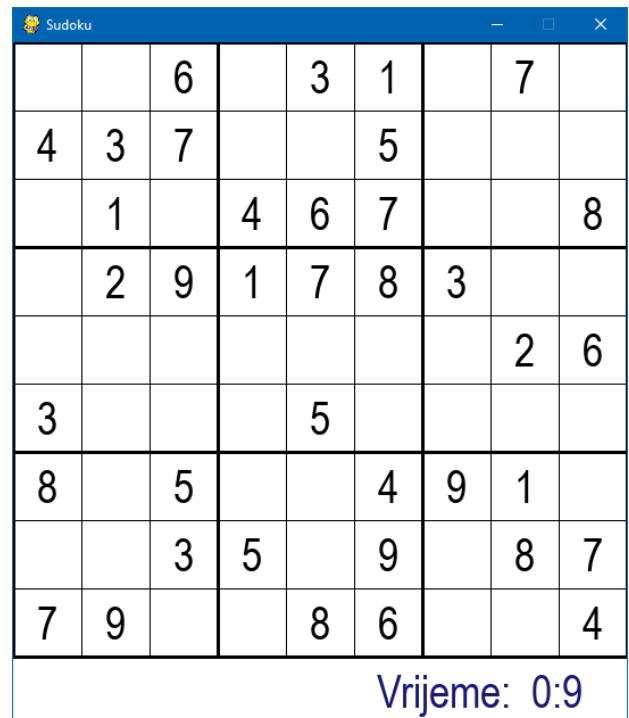
2.1. Programski jezik

Igrica je napravljena u Pythonu, a korišten je Pygame – cross-platformski set Python modula dizajniran za pisanje video igrica. Korišteni su i moduli `sys` (system-specific parameters and functions), `time` za prikaz vremena igranja i `random` kojim se nasumično odabiru ploče koje će igrač rješavati.

```
import pygame
from checker import check, valid
import time
from random import *
pygame.font.init()
```

2.2. Početak igrice

Kada pokrenemo program, pojavi se prozor s rešetkom, odnosno kvadratićima od kojih su neki već popunjeni određenim vrijednostima. U donjem desnom kutu počinje se odbrojavati vrijeme.



		6		3	1		7	
4	3	7			5			
	1		4	6	7			8
	2	9	1	7	8	3		
							2	6
3				5				
8		5			4	9	1	
		3	5		9		8	7
7	9			8	6			4
Vrijeme: 0:9								

2.3. Unos brojeva

Kada kliknemo na neki od kvadratića, on će se obojati svijetloplavom bojom, te tada možemo unesti broj. Sve dok nakon unosa ne pritisnemo tipku ENTER, broj će ostati u kvadratiću, ali neće biti zabilježen kao rješenje tog kvadratića.

U slučaju da se predomislimo i želimo u kvadratić upisati drugi broj, možemo jednostavno samo stisnuti drugi broj ili ako ga pak želimo obrisati možemo stisnuti tipku DELETE.

		6		3	1		7	
4	3	7			5			
	1		4	6	7			8
	2	9	1	7	8	3		
							2	6
3				5				
8		5	7		4	9	1	
		3	5		9		8	7
7	9			8	6			4
Vrijeme: 2:19								

Ako želimo potvrditi taj broj, stisnut ćemo tipku ENTER, te će on, ako je točan, ostati u kvadratiću. Tada će se u Shellu ispisati rečenica „Uspješno unesen broj!“.

S obzirom da se broj koji je upisan (i potvrđen) smije pojaviti samo jednom u svakom retku, svakom stupcu i svakom odjeljku od 3x3 polja, provjerava se ako se uneseni broj ponavlja.

		6		3	1		7	
4	3	7			5			
	1		4	6	7			8
	2	9	1	7	8	3		
							2	6
3				5				
8		5	7		4	9	1	
		3	5		9		8	7
7	9			8	6			4
Vrijeme: 2:51								

Ako je broj koji upišemo netočan, tada će se on izbrisati iz kvadratića, a u donjem lijevom kutu pojaviti će se **X**, a u Shellu će se ispisati rečenica „Krivo unesen broj!“. Zasada nema ograničenog broja mogućih pogrešaka.

		6		3	1		7	
4	3	7			5			
	1		4	6	7			8
	2	9	1	7	8	3		
				5			2	6
3				5				
8		5	7		4	9	1	
		3	5		9		8	7
7	9			8	6			4
Vrijeme: 3:38								

		6		3	1		7	
4	3	7			5			
	1		4	6	7			8
	2	9	1	7	8	3		
							2	6
3				5				
8		5	7		4	9	1	
		3	5		9		8	7
7	9			8	6			4
X Vrijeme: 3:58								

2.4. Završetak igrice

Kada nam je ostao samo jedan kvadratić prazan, te kada u njega unesemo broj i on bude točan, prozor se zatvara sam, a u Shellu se ispiše rečenica „Igra je završena!“

5	8	6	2	3	1	4	7	9
4	3	7	8	9	5	1	6	2
9	1	2	4	6	7	5	3	8
6	2	9	1	7	8	3	4	5
1	5	8	9	4	3	7	2	6
3	7	4	6	5	2	8	9	1
8	6	5	7	2	4	9	1	3
2	4	3	5	1	9	6	8	7
7	9	1	3	8	6	2	5	4
X Vrijeme: 4:16								

```
Uspješno unesen broj!
Uspješno unesen broj!
Krivo unesen broj!
Uspješno unesen broj!
```

```
Igra je završena!
```

```
>>> |
```

2.5. Ploče

Ploče koje se trenutno mogu rješavati, napisane su u listi `boards`, te se nasumično biraju svaki put kada se program pokrene.

```
# nasumicno se biraju ploce iz liste boards
board = boards[randint(0, 8)]
```

```
boards = [[
    [7, 8, 0, 4, 0, 0, 1, 2, 0],
    [6, 0, 0, 0, 7, 5, 0, 0, 9],
    [0, 0, 0, 6, 0, 1, 0, 7, 8],
    [0, 0, 7, 0, 4, 0, 2, 6, 0],
    [0, 0, 1, 0, 5, 0, 9, 3, 0],
    [9, 0, 4, 0, 6, 0, 0, 0, 5],
    [0, 7, 0, 3, 0, 0, 0, 1, 2],
    [1, 2, 0, 0, 0, 7, 4, 0, 0],
    [0, 4, 9, 2, 0, 6, 0, 0, 7]
```

],

```
    [
    [0, 0, 0, 0, 0, 0, 0, 8, 0],
    [6, 8, 0, 4, 7, 0, 0, 2, 0],
    [0, 1, 9, 5, 0, 8, 6, 4, 7],
    [0, 6, 0, 9, 0, 0, 0, 0, 4],
    [3, 4, 2, 6, 8, 0, 0, 0, 0],
    [1, 9, 0, 0, 5, 0, 8, 3, 0],
    [0, 0, 0, 7, 2, 0, 4, 0, 3],
    [0, 0, 6, 0, 0, 5, 0, 1, 0],
    [0, 0, 3, 8, 9, 1, 5, 0, 0]
```

],

```
    [
    [6, 0, 0, 1, 0, 0, 0, 0, 2],
    [8, 0, 1, 0, 9, 0, 0, 0, 0],
    [0, 7, 5, 0, 8, 4, 0, 0, 0],
    [4, 3, 0, 0, 2, 0, 5, 6, 1],
    [5, 1, 8, 7, 0, 0, 4, 0, 9],
    [0, 9, 6, 4, 1, 0, 3, 0, 0],
    [0, 0, 0, 0, 7, 0, 0, 0, 0],
    [0, 6, 0, 0, 3, 1, 0, 5, 0],
    [7, 0, 2, 5, 4, 0, 6, 0, 3]
```

],

```
    [
    [0, 0, 6, 0, 3, 1, 0, 7, 0],
    [4, 3, 7, 0, 0, 5, 0, 0, 0],
    [0, 1, 0, 4, 6, 7, 0, 0, 8],
    [0, 2, 9, 1, 7, 8, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 2, 6],
    [3, 0, 0, 0, 5, 0, 0, 0, 0],
    [8, 0, 5, 0, 0, 4, 9, 1, 0],
    [0, 0, 3, 5, 0, 9, 0, 8, 7],
    [7, 9, 0, 0, 8, 6, 0, 0, 4]
```

],

```
    [
    [0, 0, 0, 0, 0, 5, 4, 0, 9],
    [4, 5, 1, 0, 0, 2, 3, 0, 0],
    [9, 8, 2, 0, 0, 0, 5, 6, 1],
    [6, 0, 7, 0, 0, 0, 9, 8, 0],
    [0, 0, 3, 4, 6, 0, 0, 0, 0],
    [5, 0, 0, 2, 8, 7, 0, 1, 0],
    [0, 4, 0, 0, 7, 0, 0, 9, 6],
    [3, 0, 0, 0, 0, 0, 7, 0, 0],
    [0, 0, 5, 9, 4, 6, 8, 0, 2]
```

],

```
    [
    [0, 0, 1, 9, 8, 4, 7, 6, 0],
    [6, 0, 0, 0, 5, 7, 0, 0, 0],
    [8, 0, 7, 0, 1, 0, 0, 0, 0],
    [9, 6, 0, 3, 0, 8, 1, 0, 5],
    [1, 8, 5, 0, 2, 0, 0, 7, 3],
    [3, 0, 0, 0, 0, 0, 2, 0, 8],
    [2, 1, 0, 0, 0, 0, 0, 3, 6],
    [0, 0, 0, 1, 0, 0, 0, 0, 4],
    [0, 9, 6, 0, 0, 2, 5, 1, 0]
```

],

```
    [
    [8, 0, 0, 0, 5, 7, 0, 0, 0],
    [0, 0, 6, 0, 8, 0, 0, 0, 0],
    [0, 4, 5, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 4, 0, 5, 2, 6, 0],
    [4, 0, 0, 0, 6, 3, 5, 0, 0],
    [0, 0, 1, 9, 0, 0, 0, 0, 0],
    [9, 0, 4, 6, 0, 0, 1, 3, 0],
    [0, 5, 7, 8, 0, 1, 0, 4, 0],
    [6, 0, 0, 0, 0, 0, 7, 2, 0]
```

],

```
    [
    [4, 7, 0, 1, 3, 0, 0, 0, 0],
    [0, 0, 2, 0, 9, 0, 0, 0, 0],
    [5, 0, 0, 0, 6, 4, 0, 1, 0],
    [2, 0, 4, 0, 8, 0, 0, 9, 0],
    [0, 8, 0, 0, 0, 5, 0, 4, 0],
    [0, 1, 3, 0, 0, 0, 0, 0, 8],
    [0, 9, 0, 0, 0, 0, 4, 7, 3],
    [7, 0, 0, 0, 4, 0, 8, 5, 0],
    [0, 0, 0, 0, 0, 0, 9, 0, 6]
```

],

```
    [
    [5, 0, 0, 6, 7, 0, 9, 0, 0],
    [0, 4, 0, 8, 0, 0, 0, 0, 0],
    [8, 0, 0, 5, 0, 0, 6, 1, 3],
    [0, 6, 2, 4, 0, 0, 0, 7, 0],
    [1, 0, 0, 0, 0, 3, 0, 2, 0],
    [3, 7, 4, 9, 0, 8, 0, 0, 0],
    [0, 9, 6, 1, 0, 7, 8, 0, 2],
    [2, 1, 8, 0, 0, 6, 0, 4, 5],
    [0, 5, 0, 0, 8, 0, 0, 9, 0]
```

]]

3. BUDUĆNOST PROJEKTA

3.1. Pogreške

Zasada je broj pogreški koje napravimo tijekom rješavanja neograničen. U budućnosti bi trebala napraviti da broj mogućih pogrešaka bude najviše tri, te da se nakon treće pogreške, prozor automatski zatvori.

Također, kako bi pogreški bilo što manje, a rješavanje sudoku lakše, mogla bi omogućiti opciju „hint“ koja bi ukazala na moguće rješenje.

3.2. Više razina igrice

Kako bi se igrice unaprijedila, trebalo bi omogućiti više razina, pa bi tako na primjer bilo tri razine – lagana, srednja i teška.

- Lagana razina – U samoj ploči bilo bi ponuđeno više brojeva, te bi samim time bilo lako riješiti sudoku.
- Srednja razina – U ploči bi bilo ponuđeno manje brojeva nego na laganoj razini, ali više nego na teškoj razini, pa bi rješavanje sudoku bilo malo teže.
- Teška razina – U ploči bi bilo ponuđeno malo brojeva, te bi rješavanje sudoku bilo teško.

3.3. Ploče za rješavanje

Trenutno su ploče koje se mogu rješavati napisane u samom kodu i zasada ih je devet. U planu mi je nadodati još ploča, ali bi još bolje rješenje bila napraviti algoritam koji će sam generirati ploče. Tako će igračima biti ponuđeno više ploča za rješavanje.