

Python Sudoku Game

Erika Hyseni, Vlera Qaja, Korab Mustafa

Kolegji Riinvest, Kosovë, Prishtinë, 2020.

Erika.Hyseni@riinvest.net, Vlera.Qaja@riinvest.net,
Korab.Mustafa@riinvest.net

Abstrakt

Loja shumë e famshme botërisht Sudoku e krijuar me 1979 është temë shumë e përhapur edhe në botën e kompjuterave, në këtë research paper ne kemi vendosur ta abstrahojmë konceptin e Sudoku-t dhe t'a bëjmë sa më user friendly për përdoruesit, implementimi i lojës është bërë i mundur nga përdorimi i librarive të Python-it.

I. Hyrje

Sudoku është një lojë logjike - fjalëkryq matematikor.

Sudoku klasik - Rregullat janë të thjeshta, një fushë me madhësi 9 nga 9 qeliza. Fusha e madhe është e ndarë në nëntë fusha të vogla me madhësi 3 nga 3 qeliza. Secila prej qelizave përmban një nga numrat nga 1 deri në 9. Disa nga këta numra janë të hapur për lojtarin në fillim të lojës. Pjesa tjetër e numrave duhet të përcaktohet nga lojtari. Qëllimi është të fusni në boshllëqet me shifra nga 1 në 9 në mënyrë që secila rresht, secila kolonë dhe secila prej blloqeve të nëntëdhjetë me tre që përbëjnë rrjetin të përmbajë vetëm njërin nga secila nga nëntë shifrat. Numrat në qeliza duhet të jenë të rregulluar në mënyrë që në një rresht horizontale, një rresht vertikal dhe

një fushë të vogël 3 nga 3 kjo shifër të ndodhë vetëm një herë.

Problemi : Krijimi i lojës së Sudokut duke përdorur gjuhën programuese **Python**

Loja e sudokut ka ekzaktsisht 5.96×10^{26} grida të lojës, prandaj duke përdorur libraritë e Python do të dizajnojmë GUI-n e lojës dhe gjithashtu logjikën se si do të funksionoj sistemi.

II. State of the art

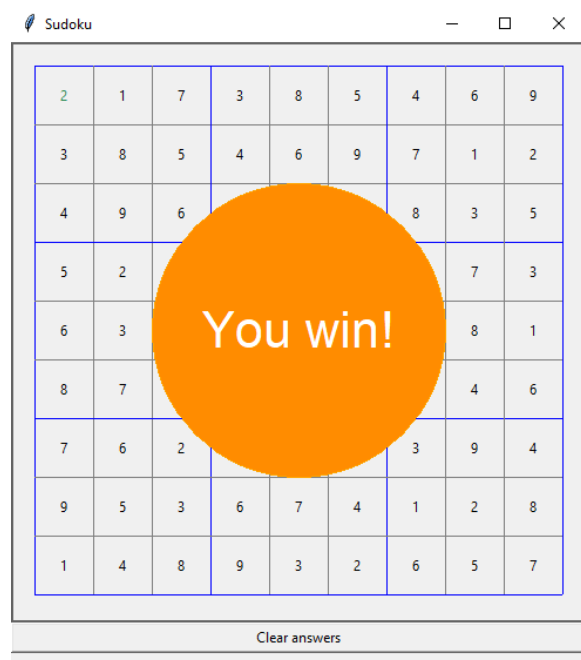
Gjatë viteve të fundit në botën e informatikës janë përdorur disa mënyra të ndryshme se si të arrihet te thjeshtëzimi i lojës së Sudoku-t dhe lidhjes së sajë me user input.

Në këtë research paper **PuzzleJAR**^[1] është përdorur një metodë që automatikisht do të gjeneroj probleme (puzzles) të reja duke u mbështetur mbi konceptin e machine learning. Mbi 200,000 probleme Sudoku të masave (9x9, 16x16) të ndryshme janë gjeneruar.

Në këtë research paper është përdorur një model tjetër që e gjeneron lojën në nivele të ndryshme të vështirsisë (Easy to Evil)^[2]. Sistemi i lojës përdorë algoritmin “dig-hole” për krijimin e gridave valid, pastaj për të caktuar vështirsinë e problemit përdoret algoritmi i krijuar nga vetë krijuesit.

Në këtë research paper është atakuar problemi i krijimit të lojës së Sudoku-t duke përdorur Python dhe libraritë (sudoku.py) e saj.^[3]

III. Rezultatet



```
def __draw_puzzle(self):
    self.canvas.delete("numbers")
    for i in xrange(9):
        for j in xrange(9):
            answer = self.game.puzzle[i][j]
            if answer != 0:
                x = MARGIN + j *
                SIDE + SIDE // 2
                y = MARGIN + i *
                SIDE + SIDE // 2
                original = self.game.start_puzzle[i][j]
                color = "black" if answer == original else "sea green"
                self.canvas.create_text(
                    x, y, text=answer, tags="numbers", fill=color
                )
```

```
def check_win(self):
    for row in xrange(9):
        if not self.__check_row(row):
            return False
    for column in xrange(9):
        if not self.__check_column(column):
            return False
    for row in xrange(3):
        for column in xrange(3):
            if not self.__check_square(row, column):
                return False
    self.game_over = True
    return True
```

IV. Përfundimi

Gjatë krijimit të kësaj scripture ne kemi përdorur shumë ‘built in libraries’ të pythonit, duke përfshirë edhe ‘tkinter’ që na mundëson vizatimin e tabelës së lojës.

Sa i përket logjikës së lojës, funksionet që kemi aplikuar e bëjnë të lehtë funksionalizimin e scripture-s, duke treguar menjëherë pas plotësimit të të gjithave numrave në tabelë se a kemi fituar apo kombinimi i dhënë është inkorrekt.

Ne besojmë që produkti final që ne kemi arritur ta sjellim është shumë lehtë i përdorshëm, sepse ka një user friendly GUI dhe mundëson user input efikas.

References

- [1] Chou Amy, Kaashoek Justin. : *Automated Constraint-based Generation of Puzzles of Varying Complexity*, MIT PRIMES, September 30, 2014. MIT,
<https://math.mit.edu/research/highschool/primmes/materials/2014/Chou-Kaashoek.pdf>
- [2] Sudoku Puzzles Generating: from Easy to Evil, Team 3485. Zhangroup.aporc,
http://zhangroup.aporc.org/images/files/Paper_3485.pdf
- [3] Davis Sean, Henderson Mattheë, Smith Andreë. *Modeling Sudoku Puzzles ëith Python*, Sean Davis, 2010. Spicy,
<https://conference.scipy.org/proceedings/scipy2010/pdfs/henderson.pdf>