

# Igra 8

## PRAVILA

- Korisnik upisom koordinata može da potrefi minu - kraj igre
- Korisnik može loše da oznaci mesto na kome je mina - igra mu to neće reći
- Uspesni kraj je samo ako su sve koordinate mina tačne kao inicijalne koje su se generisale

```
*****
**      M I N E S W E E P E R      **
*****

U minskom polju postoje 3 mine
Tvoj zadatak je da otkrijes gde su!!
Ovo je minsko polje:
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Unesi polje koje pogadjas :
X (1-5) ili 1-5B za bombu:1
Y (1-5) ili 1-5B za bombu :2
```

## ANALIZA

### Implementacija:

- > unutrašnji while za unos koordinata u kome se sve proverava
- > vanjski while u kome se igra vrti i kada se izgubi
- > detekcija kraja - pobede -> fja koja proverava koordinate 'X' u obe matrice i upoređuje njihove koordinate
- > detekcija kraja - gubitka -> ako korisnik pogodi koordinate mine
- > dodavanje mogućnosti za biranje težine igrice

- Prikaz minskog polja a zatim i način pogadjanja
- Pravi se funkcija kojom se pravi minsko polje dimenzija n. K je broj bombi koje definisemo
- Prave se dve matrice, jedna koja čuva pravo stanje i u kojoj dinamički generisemo random mine pa zatim setujemo brojeve koji označavaju u okolini nekog polja ima mina. Druga matrica je matrica zvezdica - ona predstavlja matricu koju korisnik pogadja
- Funkcija za prikaz matrice
- Game funkcija za testiranje do sada isprogramiranog koda
- Način za pogadjanje se sastoji od formatiranog inputa - korisnik mora da bude pažljiv sa unosom koordinata
- Random 2 bombe mogu da se potrefe na istom mestu

## PRVI DEO IMPLEMENTACIJE:

```
import random
def NapraviMinskoPolje(n, k):

    arr = [[0 for row in range(n)] for column in range(n)]
```

```
bombe = 0
```

```
while bombe != k: # mozda je bolji uslov ovaj nego sa forom,  
prodiskutovati
```

```
x = random.randint(0,n-1)
```

```
y = random.randint(0,n-1)
```

```
if (arr[y][x] != 'X'):
```

```
    arr[y][x] = 'X'
```

```
    print("x:y", x, y)
```

```
    bombe +=1
```

```
if (x >=0 and x <= n-2) and (y >= 0 and y <= n-1):
```

```
    if arr[y][x+1] != 'X':
```

```
        arr[y][x+1] += 1 # center right
```

```
if (x >=1 and x <= n-1) and (y >= 0 and y <= n-1):
```

```
    if arr[y][x-1] != 'X':
```

```
        arr[y][x-1] += 1 # center left
```

```
if (x >= 1 and x <= n-1) and (y >= 1 and y <= n-1):
```

```
    if arr[y-1][x-1] != 'X':
```

```
        arr[y-1][x-1] += 1 # top left
```

```
if (x >= 0 and x <= n-2) and (y >= 1 and y <= n-1):
```

```
    if arr[y-1][x+1] != 'X':
```

```
        arr[y-1][x+1] += 1 # top right
```

```
if (x >= 0 and x <= n-1) and (y >= 1 and y <= n-1):
```

```
    if arr[y-1][x] != 'X':
```

```
        arr[y-1][x] += 1 # top center
```

```
if (x >=0 and x <= n-2) and (y >= 0 and y <= n-2):
```

```
    if arr[y+1][x+1] != 'X':
```

```
        arr[y+1][x+1] += 1 # bottom right
```

```
if (x >= 1 and x <= n-1) and (y >= 0 and y <= n-2):
```

```
    if arr[y+1][x-1] != 'X':
```

```
        arr[y+1][x-1] += 1 # bottom left
```

```
if (x >= 0 and x <= n-1) and (y >= 0 and y <= n-2):
```

```
    if arr[y+1][x] != 'X':
```

```
        arr[y+1][x] += 1 # bottom center
```

```
return (arr, bombe)
```

```
def NapraviPraznoMinskoPolje(n):
```

```
    arr = [['*' for row in range(n)] for column in range(n)]
```

```
    return arr
```

```
def PrikaziMinskoPolje(map):
```

```
    for row in map:
```

```

        print(" ".join(str(cell) for cell in row))

def ispisiNaslovIgrice():
    print("*****")
    print("**      M I N E S W E E P E R      **")
    print("*****")

def Game():

    n = 5
    k = 3

    (minesweeper_map, bombe) = NapraviMinskoPolje(n, k)
    player_map = NapraviPraznoMinskoPolje(n)

    ispisiNaslovIgrice()
    print("U minskom polju postoje {} mine".format(bombe))
    PrikaziMinskoPolje(player_map)
    print("=====")
    #PrikaziMinskoPolje(minesweeper_map)

    print("Unesi polje koje pogadjas :")
    x = input("X (1-5) ili 1-5B za bombu:")
    y = input("Y (1-5) ili 1-5B za bombu :")
    print(x, y)
# Start of Program
Game()

```

## OBJASNJENJE:

Napomena 1: Zameniti sa whileom prvi for, jer moze da se desi da se generise isti random, pa to objasniti zasto se menja

Napomena 2: 8 ifova za azuriranje brojeva svuda oko mine

Napomena 3: printovi su tu radi testiranja

## DRUGI DEO IMPLEMENTACIJE:

- U drugom delu implementacije pravi se osnovna logika za pogadjanje i updateovanje matrice koja cuva podatke o pogadjanim poljima
- Posto je kraj igre (pozitiva) ako je igrac postavio sve 'X' na pravom mjestu, onda nam treba i funkcija koja ce to proveriti
- Imacemo i vanjsku petlju koja ce stalno da proverava da li korisnik zeli novu partiju da zapocne

```

def ProveriPobedu(player_map, game_map):
    #proveri da je igrac postavio sve 'X' na pravom mestu
    dimenzija = len(player_map[0])
    for i in range (dimenzija):
        for j in range (dimenzija):
            if game_map[i][j] == 'X':
                if player_map[i][j] != 'X':

```

```

        return False
    return True

def ProveriNastavakIgre():
    isContinue = input("Igraj ponovo? (d/n) :")
    if isContinue == 'n':
        return False
    return True

def Game():    /// neki deo je isti kao od proslog casa
    GameStatus = True
    while GameStatus:
        difficulty = input("Tezina: (1, 2, 3):")
        if difficulty.lower() == '1':
            n = 5
            k = 3
        elif difficulty.lower() == '2':
            n = 6
            k = 8
        else:
            n = 8
            k = 20

        (minesweeper_map, bombe) = NapraviMinskoPolje(n, k)
        player_map = NapraviPraznoMinskoPolje(n)

        ispisiNaslovIgrice()
        print("U minskom polju postoje {} mine".format(bombe))
        print("Tvoj zadatak je da otkrijes gde su!!!")
        print("Ovo je minsko polje: ")
        PrikaziMinskoPolje(player_map)

    while True:
        if ProveriPobedu(player_map, minesweeper_map) == False:

            print("Unesi polje koje pogadjas :")
            x = input("X (1-5) ili 1-5B za bombu:")
            y = input("Y (1-5) ili 1-5B za bombu :")
            if len(x) == 1 or len(y) == 1:
                x = int(x) - 1 # 0 based indexing
                y = int(y) - 1 # 0 based indexing
                if (minesweeper_map[y][x] == 'X'):
                    print("Izgubio siiii!!!!")
                    PrikaziMinskoPolje(minesweeper_map)
                    GameStatus = ProveriNastavakIgre()
                    break
                else:
                    player_map[y][x] = minesweeper_map[y][x]
                    PrikaziMinskoPolje(player_map)
            else:
                # markiraj minu
                x = x[0]
                y = y[0]

```

```
x = int(x) - 1 # 0 based indexing
y = int(y) - 1 # 0 based indexing
player_map[y][x] = 'X'
PrikaziMinskoPolje(player_map)
```

```
else:
```

```
    PrikaziMinskoPolje(player_map)
    print("Pobedio si!!!")
    GameStatus = ProveriNastavakIgre()
    break
```

```
# Start of Program
```

```
Game()
```