

Universidad de San Carlos de Guatemala

Introducción a la Computación y Programación

Ing. Hernán Véliz

Práctica 3

## **Manual Técnico**

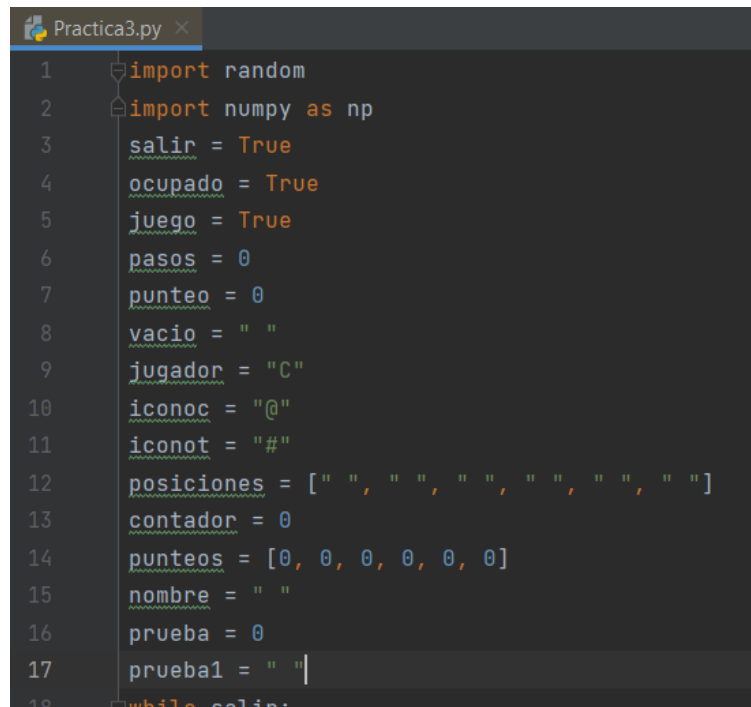
Ivan de Jesús Pisquiy Escobar

201901815

28/03/2022

En este manual podrá encontrar los métodos utilizados para que el usuario pueda entender el funcionamiento técnico de este programa.

## 1. Variables



```
Practica3.py x
1 import random
2 import numpy as np
3 salir = True
4 ocupado = True
5 juego = True
6 pasos = 0
7 punteo = 0
8 vacio = " "
9 jugador = "C"
10 iconoc = "@"
11 iconot = "#"
12 posiciones = [" ", " ", " ", " ", " ", " ", " ", " "]
13 contador = 0
14 punteos = [0, 0, 0, 0, 0, 0]
15 nombre = " "
16 prueba = 0
17 prueba1 = " "
```

Al inicio de nuestro código, declaramos ciertas variables que nos ayudarán para el funcionamiento de este, por ejemplo, utilizamos las variables booleanas para nuestros métodos while, utilizamos las variables Int para uso de contadores y en el caso de “Posiciones”, se utiliza para crear nuestra matriz e imprimir nuestro tablero con sus elementos.



### 3. Creación de tablero con paredes y comida:

```
34     po1 = random.randint(0, 9)
35     po2 = random.randint(0, 9)
36     tablero[po1][po2] = jugador
37     comida = random.randint(8, 40)
38     for i in range(0, comida):
39         ocupado = True
40         while ocupado:
41             pos1 = random.randint(0, 9)
42             pos2 = random.randint(0, 9)
43             if tablero[pos1][pos2] == " ":
44                 tablero[pos1][pos2] = iconoc
45                 ocupado = False
46     ocupado = True
47     for i in range(0, 30):
48         pos1 = random.randint(0, 9)
49         pos2 = random.randint(0, 9)
50         if tablero[pos1][pos2] == " ":
51             tablero[pos1][pos2] = iconot
52             ocupado = False
53     for line in tablero:
54         print("|", end="")
55         print(' '.join(map(str, line)), end="")
56         print("|")
```

Usando la función Random, se crea inicialmente 2 valores para poder asignar una posición aleatoria al jugar al comienzo de la partida.

Luego, dentro de una función For para comida y paredes que este se repite en función al tamaño de nuestro tablero, dentro de este se crea un método While donde el programa buscará posiciones aleatorias vacías donde colocar los íconos y cuando encuentra uno, este se asigna el booleano "Ocupado" como falso para cerrar y repetir el proceso hasta que encuentre un valor para cada signo.

## 4. Inicio del Juego

```
53 while juego and punteo < 40:
54     print("Punteo: ", punteo, " Pasos:", pasos)
55     print("-----")
56     for line in tablero:
57         print("|", end="")
58         print(' '.join(map(str, line)), end="")
59         print("|")
60     print("-----")
61     print("Realice un movimiento:")
62     mov = input()
63     if (mov == "w" or mov == "8") and (po1 - 1) < 0 and tablero[9][po2] == vacio:
64         tablero[po1][po2] = vacio
65         po1 = 9
66         tablero[po1][po2] = jugador
67         pasos = pasos + 1
68     elif (mov == "w" or mov == "8") and (po1 - 1) < 0 and tablero[9][po2] == iconoc:
69         tablero[po1][po2] = vacio
70         po1 = 9
71         tablero[po1][po2] = jugador
72         pasos = pasos + 1
73         punteo += 5
74     elif (mov == "w" or mov == "8") and (po1 - 1) < 0 and tablero[9][po2] == iconot:
75         pasos = pasos + 1
76     elif (mov == "w" or mov == "8") and tablero[po1-1][po2] == vacio:
77         tablero[po1][po2] = vacio
78         po1 = po1-1
79         tablero[po1][po2] = jugador
80         pasos = pasos + 1
81     elif (mov == "w" or mov == "8") and tablero[po1-1][po2] == iconoc:
82         tablero[po1][po2] = vacio
83         po1 = po1-1
```

Igual que métodos anteriores, se utiliza un método While, pero a diferencia de los anteriores, además de mantener un booleano, se agrega que el punteo no sea mayor a 40, esto siendo para indicar cuando el juego acaba y si el jugador decide, poder acabar el juego antes, pueda hacerlo.

A continuación, se ven los métodos para mover al jugador, cada letra o cada "Movimiento", tiene distintos escenarios donde se utilizaron distintos métodos ifs para cada uno de estos escenarios, permitiendo al usuario moverse con libertad en el tablero.

## 5. Tabla de Posiciones

```
172     if opcion == 2:
173         print("Todavía no está hecho, lo siento.")
174         print("La tabla de posiciones es:")
175         posiciones[contador] = nombre
176         punteos[contador] = pasos
177         for i in range(6):
178             if punteos[5] > punteos[4]:
179                 prueba = punteos[4]
180                 punteos[4] = punteos[5]
181                 punteos[5] = prueba
182                 prueba1 = posiciones[4]
183                 posiciones[4] = posiciones[5]
184                 posiciones[5] = prueba1
185             if punteos[4] > punteos[3]:
186                 prueba = punteos[3]
187                 punteos[3] = punteos[4]
188                 punteos[4] = prueba
189                 prueba1 = posiciones[3]
190                 posiciones[3] = posiciones[4]
191                 posiciones[4] = prueba1
192             if punteos[3] > punteos[2]:
193                 prueba = punteos[2]
194                 punteos[2] = punteos[3]
195                 punteos[3] = prueba
196                 prueba1 = posiciones[2]
197                 posiciones[2] = posiciones[3]
198                 posiciones[3] = prueba1
199             if punteos[2] > punteos[1]:
200                 prueba = punteos[1]
201                 punteos[1] = punteos[2]
202                 punteos[2] = prueba
203                 prueba1 = posiciones[1]
```

En este método, usando un For e ifs, se reordena cada vez que la opción de ver el tablero se elija, esto con el fin de si el usuario vuelve a jugar, los datos se reordenen para mostrar siempre al usuario que hizo más pasos durante el juego.

Busca en las posiciones comparándolas para saber si estas deben subir de puesto o mantener el que poseen. Este método abre paso a 5 posiciones en el tablero para distintos usuarios.