

Trabajo Práctico N°1 – Pensamiento Computacional

Universidad de San Andrés

“Simulador de Tenis en Python”

Nombre: Pedro Santiago Gentil

E-mail: pgentil@udesa.edu.ar

Objetivo:

El objetivo de este trabajo es realizar un programa interactivo en Python donde el usuario pueda simular un *game* de tenis, de modo manual y simulado. El programa debe poder mostrar los puntajes de cada jugador en el *game*, cuyos nombres son elegidos por el usuario, mientras se juega en cualquiera de los dos modos. En el modo manual, por cada jugada, el usuario debe poder elegir quien ganará el punto y quien lo pierde. Al final, el usuario elije quien va a ganar el *game*. En el modo simulado, el usuario solo debe elegir el nombre de los jugadores, pero por cada jugada que pase, el marcador debe aparecer junto con el ganador del punto. En la creación del programa se deben utilizar funciones, variables y tipos, secuencias, condicionales y ciclos, y diagramas de flujo

Asimismo, el trabajo mostrará el desarrollo del programa, junto con explicaciones sobre el camino tomado para realizarlo. Además, los errores que se cometen y los problemas encontrados junto con las soluciones van a ser expuestos.

Desarrollo:

El programa hecho está compuesto por dos archivos *Python*. El principal, llamado *main2.py*, y el secundario llamado *v2.py*. Dentro de *main2* se definió una función sin parámetros con el mismo nombre, y se importo el archivo secundario junto con la función dentro. La función *main()* contenía diferentes variables, tal como los nombres de los jugadores como entradas hechas por el usuario, los puntos de cada jugador como enteros, una tupla con los diferentes puntajes reglamentarios del tenis, una variable de tipo *input* que decide si se juega en modo automático o en modo manual y otro *input* que, en caso de jugar al modo automático, decide si mostrar las impresiones de cada jugada o directamente el resultado final.

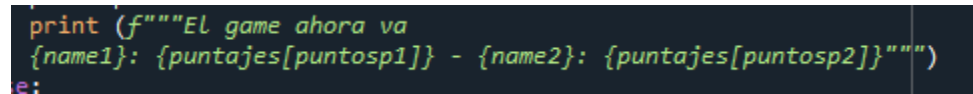
Dentro de *v2.py* hay una función con el nombre “score” que ejecuta la simulación del partido en sí, usando a las variables de la función *main2.py* como argumentos. Dentro del código se hayan condicionales y ciclos de tipo *while*.

El programa deja elegir el nombre, el modo y en caso de elegir el modo automático, si mostrar el *game* entero o solo el resultado, al usuario.

Alternativas consideradas y estrategias tomadas:

Durante el proceso de creación del código tuve dos cambios en mi idea original. El primero fue la definición de variables y su posición en el código. Al principio, quería hacer un programa dividido en tres archivos: uno para la función principal, otro para el modo manual y otro para el automático (simulado). Después, a recomendación del profesor y para no repetir código, lo cambie para que use solo dos archivos, uno para la función principal y otro para la función que ejecute el partido, y usar condicionales para hacer que el modo automático y manual se ejecuten en esa misma función.

El segundo cambio fue el de sistema de puntajes. No tenía claro como representar los puntajes en el programa, hasta que termine usando una tupla a la que llame “puntajes” con los puntajes reglamentarios del tenis, y otras dos variables, a las que llame “puntosp1” y “puntosp2”, que marcaban la posición del elemento llamado dentro de la tupla para representar los puntos de cada jugador.



```
print(f"""El game ahora va
{name1}: {puntajes[puntosp1]} - {name2}: {puntajes[puntosp2]}""")
```

Ilustración 1 Demostración de un f-string dentro de una función print() que imprime el marcador del game

Esto me ayudo, porque me vi más propicio a usar f-strings, lo que me ayudo a acortar bastante el código. En la Ilustración 2 se muestra que variables usaba para imprimir el marcador. Los puntos de los dos jugadores empezaban en 0, y cuando uno de los ganaba, se le sumaba un punto. Esto hacía que el puntaje de ese mismo jugador pase de 0 a 15, ya que se desplaza al siguiente elemento de la tupla que se puede visualizar en la Ilustración 1.

Resultados de ejecuciones:

Los problemas que siguen son ejecuciones de la versión del programa que estaba dividido en tres archivos.

```
TeniSim

Elija en que modo quiere jugar:
- Ingrese '1' si quiere jugar el modo manual
- Ingrese '2' si quiere jugar el modo simulado
> 1
No ha seleccionado ningun modo de juego.

Elija en que modo quiere jugar:
- Ingrese '1' si quiere jugar el modo manual
- Ingrese '2' si quiere jugar el modo simulado
> |
```

Ilustración 2 Primer problema en consola, al poner el numero 1 como entrada no se activaba el modo manual

El problema de la Ilustración 2 sucedió en la primera prueba. El problema residía en el código dentro de `main2()`, que estaba mal pensado.

```
while True:
    mode = int(input("""Elija en que modo quiere jugar:
- Ingrese '1' si quiere jugar el modo manual
- Ingrese '2' si quiere jugar el modo simulado
> """))
    if mode != 1 or mode != 2:
        print("No ha seleccionado ningun modo de juego.")
    name1 = input("Nombre del jugador 1: ")
```

Ilustración 3

```
while True:
    mode = int(input("""Elija en que modo quiere jugar:
- Ingrese '1' si quiere jugar el modo manual
- Ingrese '2' si quiere jugar el modo simulado
> """))
    if mode == 1 or mode == 2:
        break
    else:
        print("No ha seleccionado ningun modo de juego.")
    name1 = input("Nombre del jugador 1: ")
```

Ilustración 4

En ningún momento, con el código de la Ilustración 3, se iba a poder salir del *loop*. Entonces modifique el código al de la Ilustración 4. El funcionamiento de ese *loop* era que no se termine el programa en caso de apretar mal un número.

Después, cuando cambié la estructura del programa a dos archivos en vez de tres tuve algunos problemas también. Por ejemplo, al usar la función *choice* del módulo *random*.

```
File "C:\Users\Pedro\Desktop\UNIVERSIDAD\UEESA\Pensamiento Computacional\PC
\TPs\TP1\v2.py", line 39, in score
    score_sum = random.choice('1', '2')

TypeError: choice() takes 2 positional arguments but 3 were given

In [2]: |
```

Ilustración 5

No entendía por qué sucedía el error, pero al decirme que necesitaba solo dos argumentos posicionales pensé que capaz era la coma y así era. Entonces saque la coma entre el 1 y el 2, y al ejecutarlo me funcionó bien esa parte del programa.

Otro error que hubo fue en un *while loop*. En la consola me di cuenta de que el programa cuando se ejecutaba en automático se repetía después de que uno de los jugadores ganase como se muestra en la Ilustración 6.

```
p ha marcado un punto

El game ahora va:
p: 40 - c: 30

p ha marcado un punto
p ha ganado el game

c ha marcado un punto

El game ahora va:
p: 0 - c: 15

c ha marcado un punto

El game ahora va:
p: 0 - c: 30

c ha marcado un punto

El game ahora va:
p: 0 - c: 40
```

Ilustración 6

El problema estaba en el *while loop* en *main2()* que contenía a la función *score* en la parte automática del programa.

```
print ( "no ha seleccionado nada. vuelve a intentar" )
while True:
    v2.score(mode, puntosp1, puntosp2, name1, name2, puntajes, auto_ch)
    break
```

Ilustración 7

Lo solucione poniendo un *break* debajo de la función *score*.

Después no hubo muchos mas errores, ya que mucha parte del código ya la había probado cuando estaba dividido en tres partes, que es muy parecido al actual.

Problemas encontrados y soluciones:

Al momento de armar el código, me encontré con muchos problemas en los condicionales y con los *while loops* en Python. Las funciones específicas de los *ifs*, *elifs* y *else* fueron difíciles de incorporar al código. Igualmente pude solucionar al final los problemas generados por los condicionales probando repetidas veces y practicando diferentes formas de usarlos. Después lo mismo sucedió con los ciclos, pero generalmente los problemas ocurrían debido a errores en el código, como no poner *breaks* donde se debía. El sistema de puntos también fue un poco desafiante, ya que, en el tenis, los puntos en un *game* pasan del 15 al 30, pero después del 30 al 40, entonces con un algoritmo con sumas para modificar una variable directamente iba a ser complicado de realizar. Por eso utilice una tupla con los valores reglamentarios del tenis y una variable que llamase a los elementos de esta, que serían los puntajes de cada jugador, que variaban entre el 0 y el 5 como máximo. De esta manera pude solucionar el problema con el sistema de puntos.

Instrucciones de uso del programa:

- Se debe tener el módulo random.py y time.py descargado en la librería estándar de Python antes de ejecutar el programa.
- Se deben de tener los tres archivos que forman parte del programa: *main.py*, *simulado.py* y *manual.py*.

Bibliografía:

W3schools (2022). Refsnes Data. Recuperado de: <https://www.w3schools.com/default.asp>

Accedido en: marzo de 2022