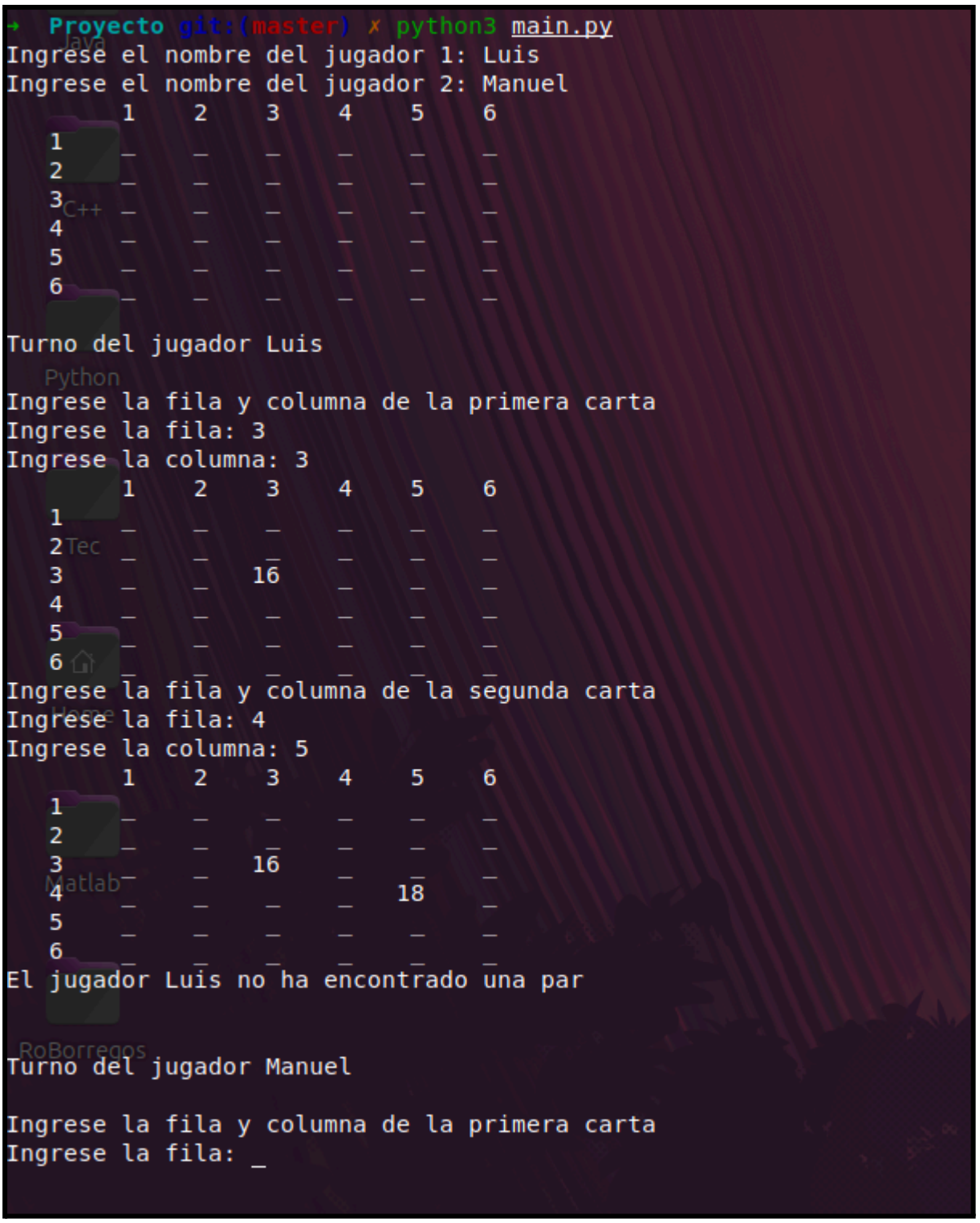
Tecnológico de Monterrey

Escuela De Ingeniería Y Ciencias

Pensamiento Computacional para la Ingeniería TC1028

Casos de Prueba: Proyecto- Juego Cognitivo

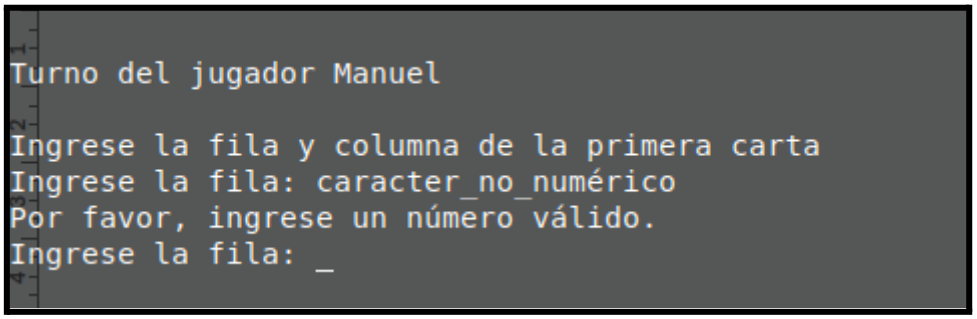
Luis Espinoza Navarrete A01668021



El juego inicia solicitando ambos nombres de jugadores que serán mostrados según sus turnos y a continuación el jugador en este caso “Luis” podrá indicar el número y fila deseado.

Caso de Prueba 1:

¿Qué pasa si ingresa un carácter no numérico?



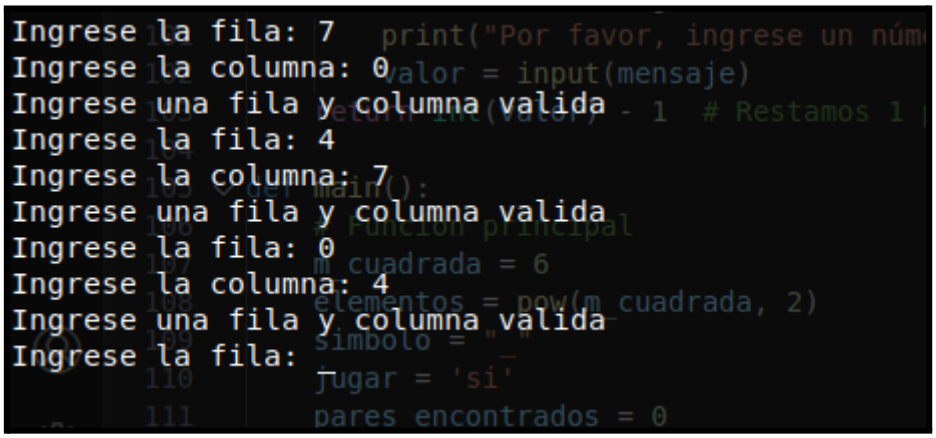
Para estos casos tenemos la función solicitar_posicion_valida(mensaje) que hace uso del método valor.isdigit() para validar que se reciban dígitos numéricos

```
def solicitar_posicion_valida(mensaje):
    valor = input(mensaje)
    while not valor.isdigit():  # Verificar que la entrada sea un número
        print("Por favor, ingrese un número válido.")
        valor = input(mensaje)
    return int(valor) - 1  # Restamos 1 para ajustar a índices de la matriz
```

Y en consecuencia se solicita una entrada válida

Caso de Prueba 2:

¿Qué pasa si se ingresa un número de fila o columna mayor o menor al rango permitido?

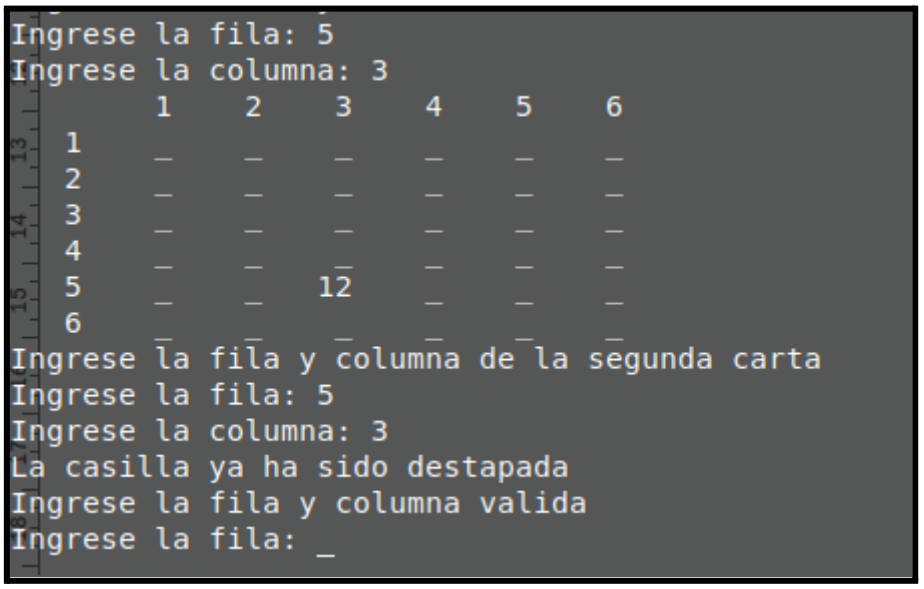


Para estos casos se ha implementado la función validar_posicion(fila,columna,m_cuadrada) que verifica que si las filas o columnas ingresadas estén fuera del rango deseado se solicite nuevamente según:

```python
def validar_posicion(fila,columna,m_cuadrada):
    #validar que la fila y columna esten dentro de la matriz
    while fila<0 or fila>=m_cuadrada or columna<0 or columna>=m_cuadrada:
        print("Ingrese una fila y columna valida")
        fila=int(input("Ingrese la fila: "))-1
        columna=int(input("Ingrese la columna: "))-1
    return [fila,columna]
```

Caso de Prueba 3 y 4:

¿Qué pasa si en su segunda carta ingresa una fila y columna válida pero es la misma que acaba de destapar?



¿Y si en general ya hay una casilla destapada pero se vuelve a solicitar esa posición para destapar?

```
      1    2    3    4    5    6
  1
  2
  3
  4
  5
  6   18
El jugador Manuel ha encontrado una par

Desea seguir jugando? si/no: si

Turno del jugador Luis

Ingrese la fila y columna de la primera carta
Ingrese la fila: 6
Ingrese la columna: 1
La casilla ya ha sido destapada
Ingrese la fila y columna valida
Ingrese la fila: 4
Ingrese la columna: 5
La casilla ya ha sido destapada
Ingrese la fila y columna valida
Ingrese la fila:
```

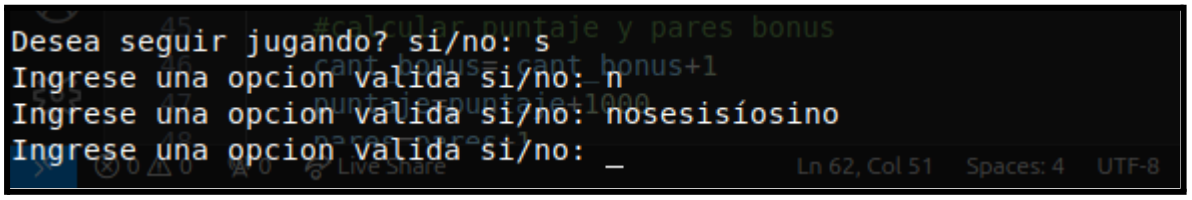
Para ambos casos tenemos la misma función que se aplica en cada turno y es validar_casilla_sin_voltear(simbolo,matriz,fila,columna)

```python
def validar_casilla_sin_voltear(simbolo,matriz,fila,columna):
    #validar que la casilla no haya sido destapada
    while matriz[fila][columna]!=simbolo:
        print("La casilla ya ha sido destapada")
        print("Ingrese la fila y columna valida")
        fila=int(input("Ingrese la fila: "))-1
        columna=int(input("Ingrese la columna: "))-1
    return [fila,columna]
```

Esta función valida que si la posición de la fila y columna en la matriz es distinta al símbolo, en este caso “*”, entonces debe solicitar nuevamente los inputs pues esta casilla ya ha sido volteada anteriormente.

Caso de Prueba 5:

¿Si responden con algo distinto a “si” o “no” cuando se les pregunta si quieren continuar ?

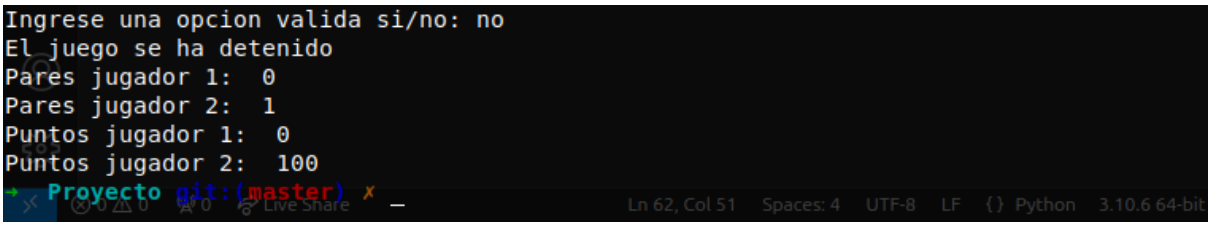


Un pequeño bucle en la parte final del main se encarga de evaluar esos casos según:

```
if num_jugador == 2 and jugar != "no":
    jugar = input("Desea seguir jugando? si/no: ")
    while jugar != "si" and jugar != "no":
        jugar = input("Ingrese una opcion valida si/no: ")
    if jugar == "no":
        validar_final_forzado(pares_1, pares_2, puntos_1, puntos_2, cant_bonus_1, cant_b
```

Caso de Prueba 6:

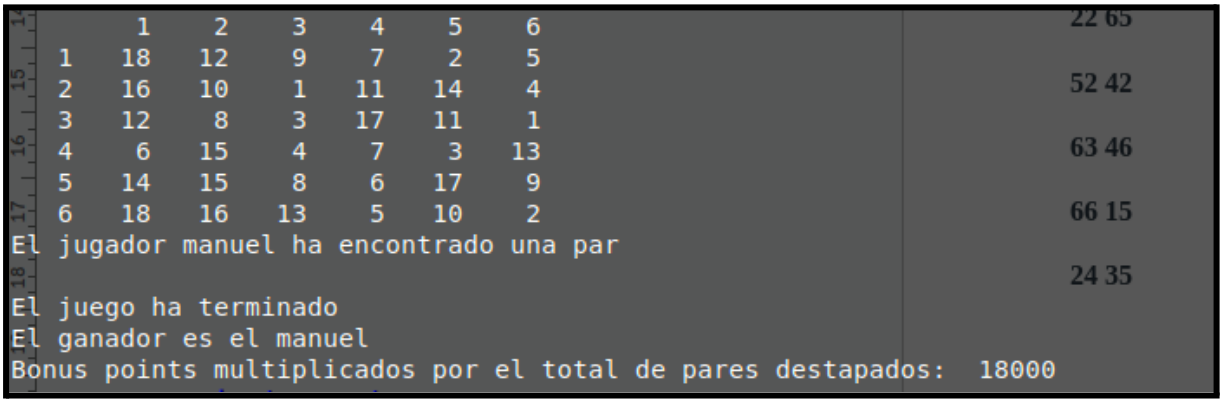
Si se responde negativamente el juego se termina y siguiendo las instrucciones, se muestra la cantidad de pares y los puntos obtenidos de cada jugador.



La función que se encarga de validar este final forzado es validar_final_forzado(pares_1,pares_2,puntos_1,puntos_2,cant_bonus_1,cant_bonus_2)

```
def validar_final_forzado(pares_1,pares_2,puntos_1,puntos_2,cant_bonus_1,cant_bonus_2):
    #validar si el juego ha terminado forzadamente y mostrar ganador
    print("El juego se ha detenido")
    print("Pares jugador 1: ",pares_1)
    print("Pares jugador 2: ",pares_2)
    print("Puntos jugador 1: ",puntos_1)
    print("Puntos jugador 2: ",puntos_2)
    if cant_bonus_1:
        print("Bonus points jugador 1: ",cant_bonus_1*1000)
    if cant_bonus_2:
        print("Bonus points jugador 2: ",cant_bonus_2*1000)
```

Caso de Prueba 7:



Cuando se termina el juego se muestra al ganador con su nombre, en este caso “manuel” y la cantidad de bonus points multiplicados por el total de pares destapados como indican las instrucciones del proyecto.