

# MANUAL TECNICO

PAC-USAC-PYTHON

Cristian Daniel Gomez Escobar

202107190

IPC 1

Sección D

## Menú inicial

Este será la codificación inicial, es donde se ha declarado el texto inicial

```
8
9     print("\n PACMAN - IPC 1 - 2022")
10    print("-----")
11    print("1.      iniciar juego")
12    print("2. tabla de posiciones")
13    print("3.      salir")
14    print("-----")
15    print("")
```

## Entrada de datos

A través de un try catch evitaremos que las opciones ingresadas sean de tipo carácter, si se escoge una opción diferente a las únicas 3 que hay, se enviara un mensaje de error hasta que se escoja una opción valida, la variable entrada enviara el valor a un if que definirá lo que se hará.

```
16    try:
17        entrada = int(input("ingrese la opcion: "))
18    except (ValueError, TypeError, IndexError):
19        print("tiene que ingresar un valor numerico")
```

## Inicial juego

Una vez evaluado que entrada sea igual a 1, ejecutara las acciones que generaran el juego pacman

### Tablero:

Aquí creara primero el tablero, recorriendo las filas y columnas asignando una lista en cada posición.

```
25
26    def creatablero(fil,col,val):
27        tablero=[]
28        comida=[]
29        numero=0
30        for i in range(fil):
31            tablero.append([])
32            for j in range(col):
33                tablero[i].append(val)
34        return tablero
35
36    def muestratablero(tablero):
37        comida=[]
38        numero=0
39        print(". . . . .")
40        for fila in tablero:
41            print(".*",end=" ")
42            for elem in fila:
43                print(elem, end=" ")
44            print(".*")
45        print(". . . . .")
```

## Distribución de la comida (@)

Se utilizó la librería random que será la encargada de definir la cantidad y posición de cada una de la comida, por cada comida colocada se ira aumentando un contador, una vez este alcance la cantidad definida por la librería random, dejara de posicionar la comida en el tablero, a través de un if se encargara de que la comida no se sobre ponga en una pared o el pacman.

```
80
81
82     while numero < cantidadcomida:
83         p= random.randint(0,columnas-1)
84         q=random.randint(0,filas-1)
85         if visible[p][q] != "c" and visible[p][q] != "#" and visible[p][q] != "@":
86             visible[p][q]="@"
87             numero +=1
88             comida.append((p,q))
89
```

## Distribución de paredes (#)

Se utilizó la librería random que será la encargada de definir la cantidad y posición de cada una de la comida, por cada comida colocada se ira aumentando un contador, una vez este alcance la cantidad definida por la librería random, dejara de posicionar la comida en el tablero, a través de un if se encargara de que la comida no se sobre ponga en una comida o el pacman.

```
91
92
93     while numero2 < cantidadparedes:
94         m= random.randint(0,columnas-1)
95         n=random.randint(0,filas-1)
96         if visible[m][n] != "c" and visible[m][n] != "@" and visible[m][n] != "#":
97             visible[m][n]="#"
98             numero2 +=1
99             paredes.append((m,n))
100
```

## Pedir datos

Se ha definido un método donde pediré los datos del movimiento, guardara valores de tipo carácter en la variable opción en donde luego evaluara la acción del movimiento.

```
48
49
50     def menu():
51
52         opcion = input("ingese el movimiento")
53         return opcion
54
```

## Opciones de movimiento

Cada movimiento esta definido por un especifico carácter, este a través de un if verificara que no se encuentre en un borde de la matriz (el tablero) de ser así la posición volverá a ser la misma, si no es así se moverá un espacio en especifico, y esta posición será guardada en la variable real

```
122
123
124     if mov == "w" or mov == "8":
125
126         if x==0 :
127             x=0
128             real=visible[x][y]
129             visible[x][y]="c"
130
131             movimientos+=0
132
133
134         else:
135             visible[x][y]=real
136             x-=1
137             real=visible[x][y]
138             visible[x][y]="c"
139             visible[x+1][y]=" "
140             movimientos+=1
141
142
143     elif mov == "s" or mov == "5":
144         if x==filas-1:
145             x=filas-1
146             real=visible[x][y]
147             visible[x][y]="c"
148             movimientos+=0
149         else:
150             visible[x][y]=real
151             x+=1
152             real=visible[x][y]
153             visible[x][y]="c"
154             visible[x-1][y]=" "
155             movimientos+=1
156
```

## Condicional de ganador

A través de un if se evaluará las únicas dos opciones que existen para ganar, estas serán que el contador de comida se reduzca a 0, o que la variable conteo llegue a 40.

```
271
272         if contador==40 or comidatotal==0:
273             print("ganaste :)")
274             vectormov.append(movimientos)
275             break
276
277     muestratablero(visible)
```

## Tabla de posiciones

Si la variable es igual a 2, este ordenará de forma descendente una lista que ha almacenado los nombres y punteos guardados por cada partida, posteriormente procederá a convertirla y guardarla en una lista, la cual mostrará los primeros tres lugares que se posicionan en la lista, luego con un for que recorre dicha lista, se irán mostrando cada uno de los datos que van ingresando y ordenándose en las primeras posiciones.

```
277
278     elif(entrada==2):
279
280         print("_____")
281         print("\n Tabla de posiciones\n")
282         vectormov.sort(reverse=True)
283         #convierto el diccionario a una lista ya descendente
284         ddd=sorted(diccionario.items(), key=itemgetter(1),reverse = True)
285         de=ddd[0:3]
286         #convierto el diccionario a otro diccionario pero ya ordenado (no fue utilizado)
287         descent=dict(sorted(diccionario.items(), key=itemgetter(1),reverse = True))
288         t=descent.items()
289         for clave,valor in de:
290             print("{0} = {1}".format(clave,valor))
291         print(" ")
292         print("_____")
293
```

## Salir

Si la variable entrada es igual a 3, esta conducirá a un elif en donde retornará un break que será el encargado de terminar el programa.

```
292
293     elif(entrada==3):
294         print("\nfin del programa")
295         break
296
```

