

Manual Técnico

1. Método Crear matriz

Este método es utilizado para la creación de un arreglo de tamaño $m \times n$ que será utilizado para la creación del tablero. Este método consiste en ir creando un arreglo bidimensional con números que posteriormente podrán ser modificados.

```
def crear_matriz(m, n):  
    return[[i*j for j in range(n)]for i in range(m)]
```

2. Método Rellenar matriz

Este método es utilizado para crear el tablero del juego, utilizando ciclos y condicionales para indicar donde agregar espacios en blanco y caracteres que indiquen el borde del tablero.

```
def rellenar_matriz(m, n, list):  
    for i in range(0, m):  
        for j in range(0, n):  
            if i == 0 or i == m-1:  
                list[i][j] = ('-')  
            elif j == 0 or j == n-1:  
                list[i][j] = ('|')  
            else:  
                list[i][j] = (' ')  
            # tableros[i][j]=5  
    return list
```

3. Método imprimir arreglo

Este método se utiliza para mostrar en pantalla el tablero creado, esto se logra gracias a ciclos que recorren e imprimen cada posición del arreglo.

```
def imprimir_matriz(m, n, list):  
    for i in range(0, 7):  
        for j in range(0, 8):  
            print(list[i][j], end=' ')  
        print('\n')
```

4. Definiendo número de ítems que aparecerán en el tablero

Para esto se utilizó el módulo random en el cual definíamos un rango del número de ítems que queríamos tener en el tablero.

```
# Implementando premios y demas  
noParedes = random.randint(5, 10)  
nofantasmas = random.randint(1, 5)  
noPremios = random.randint(3, 6)  
puntajeMaximo = noPremios*10  
noPac = 1
```

5. Colocar ítems en el tablero

Para esto se utilizó un ciclo while que dejaría de repetir cuando el numero de item a agregar llegara a cero, para esto se generó un numero aleatorio para la fila y para la columna que luego se comprobaría si esa posición se encontraba vacía se podría colocar el ítem ahí y el numero de ítem disponible des incrementaría hasta llegar a cero.

```
while noParedes > 0:
    fil = random.randint(1, 5)
    col = random.randint(1, 6)
    if matriz[fil][col].isspace():
        matriz[fil][col] = PARED
        noParedes = noParedes-1
```

6. Movimiento del PacMan

Para esto se utilizaron condicionales que dependiendo de la letra ingresada se movería el PacMan en una dirección específica en el tablero, ya sea restando o sumando una posición en el eje X o en el eje Y dependiendo del movimiento elegido por el usuario.

```
opcion = input()
opcion = opcion.lower()
if opcion == 'w':
    posActualX -= 1
elif opcion == 'a':
    posActualY -= 1
elif opcion == 's':
    posActualX += 1
elif opcion == 'd':
    posActualY += 1
elif opcion == 'f':
    exit = True
else:
    print('Ingresa una opcion valida')
```