Agentes reactivos simples basado en objetivos.

En esta práctica se hace uso del concepto de agentes reactivos simples a partir del planteamiento del caso de la aspiradora, el cual se resolvió por medio de Python. A continuación, se explicará brevemente acerca de qué son este tipo de agentes:

Los cuatro tipos básicos de programas para agentes que encarnan los principios que subyacen en casi todos los sistemas inteligentes:

- Agentes reactivos simples
- Agentes reactivos basados en modelos
- Agentes basados en objetivos
- Agentes basados en utilidad

El tipo de agente más sencillo es el agente reactivo simple. Estos agentes seleccionan las acciones sobre la base de las percepciones actuales, ignorando el resto de las percepciones históricas. Los agentes reactivos simples tienen la admirable propiedad de ser simples, pero poseen una inteligencia muy limitada. Estos agentes funcionaran solo si se puede tomar la decisión correcta sobre la base de la percepción actual, la cual es posible sólo si el entorno es totalmente observable. Incluso el que haya una pequeña parte que no se pueda observar puede causar serios problemas.

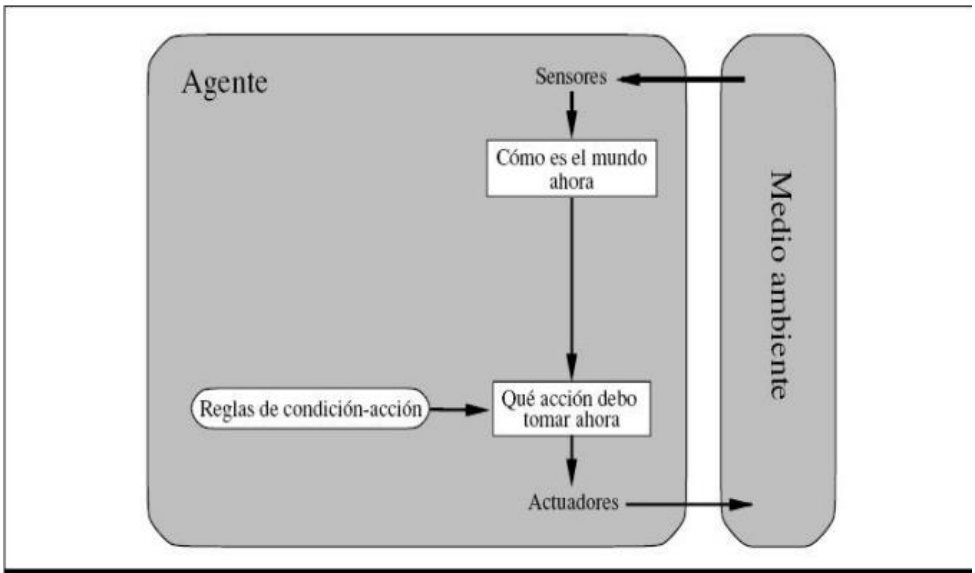


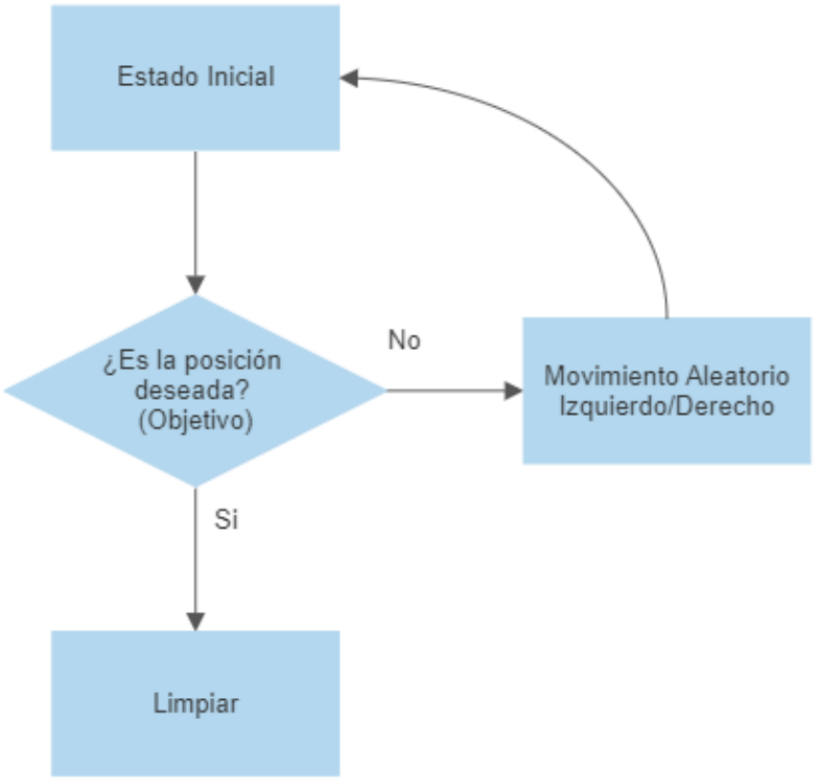
Diagrama esquemático de un agente reactivo simple.

Considerando lo anterior se expondrá el problema a resolver en esta práctica a continuación:

- La aspiradora puede ubicarse en 10 posiciones diferentes.
- La aspiradora se mueve a la izquierda o derecha de forma aleatoria.
- La aspiradora debe encontrar una posición en específico que debe limpiar.

1	2	3	4	5	6	7	8	9	10

Diagrama de flujo:



Código de la practica:

```
from random import randint
from matplotlib import pyplot as plt

"""La función llenar recibe como parámetros las posiciones de la aspiradora y basura, así como la lista con las 10 posiciones llenas de 0's, con ello se colocan los valores de
-1 y 1 en la lista, que hacen referencia a la basura y aspiradora respectivamente para poder ser visualizados en sus posiciones correspondientes"""
def llenar(basuraPos, a, listaPos):
    listaPos[basuraPos - 1] = -1
    listaPos[a - 1] = 1
    print("Posiciones iniciales:\n", listaPos)
    return listaPos

"""La función mov() permite hacer y registrar los movimientos correspondientes (izquierda y derecha) de la aspiradora hasta llegar a la posición de la basura y limpiarla,
finalmente retorna el valor del total de movimientos efectuados"""
def mov(lista, basuraPos, aspiradoraPos): #Recibe como parámetros la lista de 0's con los valores correspondientes a la basura y aspiradora ya asignados
    movimientos = 0 #Esta variable permitirá hacer el registro de todos los movimientos que le tome a la aspiradora hasta limpiar la basura
    posActual=0 #Esta variable permite registrar los cambios de posición de la aspiradora
    while (basuraPos != aspiradoraPos): #Mediante este ciclo se llevarán a cabo los movimientos de la aspiradora hasta que se encuentre con la basura
        print("-----------------------------------------------")
        direccion = randint(0, 1) #Con esta variable que toma valores entre 0 y 1 se define la dirección de movimiento de  la aspiradora, en donde 0 corresponde a izquierda y 1 a derecha
        if direccion == 1: #A partir de este condicional se ejecutan las instrucciones para el movimiento a la derecha
            posActual = lista.index(1) #Se identifica el valor de la posición actual de la aspiradora
            lista[posActual]=0 #Se elimina el 1 que identifica a la aspiradora de su posición actual para poder moverlo a la derecha
            lista[posActual + 1] = 1 #Se agrega el 1 en la siguiente posición para registrar el movimiento a la derecha
            aspiradoraPos += 1 #Se incrementa el valor de la posición de la aspiradora una unidad
            print("Derecha")
            print(lista)
            print(f'Posición de la aspiradora: {aspiradoraPos}')
            print(f'Posición de la basura: {basuraPos}')
            movimientos += 1 #Se incrementa un movimiento de la aspiradora

        else: #A partir de este condicional se ejecutan las instrucciones para el movimiento a la izquierda
            posActual = lista.index(1)
```

```
        if posActual >= 1: #Con este condicional se identifica que la aspiradora aún se encuentra en una posición a la que se le pueden restar valores, para evitar llegar a posiciones negativas
            lista[posActual] = 0 #Se elimina el 1 que identifica a la aspiradora de su posición actual para poder moverlo a la izquierda
            lista[posActual - 1] = 1 #Se agrega el 1 en la posición anterior para registrar el movimiento a la derecha
            if aspiradoraPos > 1: #Se identifica que la posición de la aspiradora sea mayor que 1 para poder restarle el valor del movimiento
              aspiradoraPos -=1 #Se resta el valor de la posición de la aspiradora una unidad
            elif aspiradoraPos == 1: #Si la posición de la aspiradora es igual a 1, entonces ya no se le resta valores para evitar posiciones negativas
               aspiradoraPos = 1
            print("Izquierda")
            print(lista)
            print(f'Posición de la aspiradora: {aspiradoraPos}')
            print(f'Posición de la basura: {basuraPos}')
            movimientos += 1 #Se incrementa un movimiento de la aspiradora
        elif posActual==0: #En el caso de que la aspiradora este en la primera posición, se ejecutan las siguientes instrucciones para no obtener posiciones negativas
            print("Izquierda")
            print(lista)
            print(f'Posición de la aspiradora: {aspiradoraPos}')
            print(f'Posición de la basura: {basuraPos}')
            movimientos += 1 #Se incrementa un movimiento de la aspiradora
    return movimientos #La función retorna el valor de los movimientos totales de la aspiradora para limpiar la basura

"""La función promedio permite obtener el promedio de una lista de valores, en este caso nos ayuda a obtener el promedio del número de movimientos
que le toma a la aspiradora limpiar la basura ubicada en una posición x"""
def promedio(listaMovimientos): #Se le pasa como argumento una lista con los valores de los diferentes movimientos de la aspiradora
    contador=0
    suma=0
    for x in listaMovimientos: #Mediante este ciclo se van sumando los valores de la lista
        suma=suma+x
        contador+=1
        return suma/contador #Finalmente, retorna el promedio al dividir la suma de los valores entre el total de elementos de la lista
if __name__ == "__main__": #A partir de aquí se encuentra
    puntosX = list() #Se crea una lista para almacenar los valores correspondientes a la distancia entre la aspiradora y la basura
    puntosY = list() #Se crea una lista para almacenar los valores correspondientes al promedio de movimientos que le toma a la aspiradora limpiar la basura
```

```
    for i in range(3): #Este ciclo se ejecuta 3 veces para ingresar 3 distancias diferentes entre la aspiradora y la basura
        listaPos = [0,0,0,0,0,0,0,0,0,0] #Esta lista permite identificar las 10 posiciones en las que se pueden encontrar la basura y aspiradora
        basuraPos = int(input("Posicion de la basura: ")) #Se registra y guarda la posición de la basura
        aspiradoraPos = int(input("Posicion de la aspiradora: ")) #Se registra y guarda la posición inicial de la aspiradora
        intentos = int(input("¿Cuántos intentos desea ejecutar? ")) #Se registra el número de intentos a realizar
        movimientos=list() #En esta lista se van a ir almacenando el total de movimientos que le toma a la aspiradora limpiar la basura en diferentes intentos

        for i in range(intentos): #Con este ciclo se van a realizar los diferentes intentos para que la aspiradora limpie la basura, invocando las diferentes funciones descritas
            totalMovimientos = mov(llenar(basuraPos, aspiradoraPos, listaPos), basuraPos, aspiradoraPos) #Se llena la lista de posiciones y también se realizan los movimientos de la aspiradora
            print("Total de movimientos de este intento:", totalMovimientos) #Se imprime el total de movimientos de la lista
            movimientos.append(totalMovimientos) #Se va llenando la lista de movimientos con los valores de cada intento
            print("Registro de movimientos:",movimientos)
            if i==intentos-1: #Al llegar al último intento, se realizan las siguientes instrucciones para evitar registrar todos los promedios obtenidos y contemplar sólo los valores totales
                y=promedio(movimientos) #Se guarda el promedio del total de movimientos que le tomó a la aspiradora limpiar la basura en estos intentos
                print("--------------------------------------------------")
                print("Promedio del total de movimientos:", y)
                x=basuraPos-aspiradoraPos #Se obtiene la distancia entre la aspiradora y la basura establecida para estos intentos
                print("Distancia entre la basura y la aspiradora:", x)
                puntosX.append(x) #Se agrega el promedio total de intentos a su lista correspondiente
                puntosY.append(y) #Se agrega la distancia entre la aspiradora y la basura a su lista correspondiente

print("**************************************************************")
#Separador para cada distancia


print("Distancias:", puntosX)
print("Promedios de cada distancia:", puntosY)
"""Se crea la gráfica para cada intento, pasando como parámetros las listas de promedios de movimientos (Eje Y) y valor de distancias (Eje X), que en este caso dibujan
3 puntos correspondientes a 3 distancias. Posteriormente se agregan algunas caracteristicas a la gráfica como el color y unión entre puntos"""
```

```
plt.plot(puntosX, puntosY, "o", color="green", linestyle="dashed",)

plt.show() #Se despliega la gráfica en pantalla
```

La ejecución del código se hará considerando 10 intentos para cada una de las distancias a evaluar, por lo que al final se obtendrán una gráfica con los 3 puntos correspondientes a cada caso, los cuales servirán para observar el comportamiento de la aspiradora conforme se va aumentando la distancia.

Preguntas:

- **Si empieza en la posición 2 (estado inicial) y debe aspirar en la posición 5(objetivo), en promedio ¿Cuántos movimientos (acciones) tiene que hacer el agente para llegar a la posición deseada?**

```
Posicion de la basura: 5
Posicion de la aspiradora: 2
¿Cuántos intentos desea ejecutar? 10
```

```
Registro de movimientos: [15, 3, 17, 3, 16, 47, 3, 13, 13, 45]
--------------------------------------------------
Promedio del total de movimientos: 15.0
Distancia entre la basura y la aspiradora: 3
```

  El promedio de movimientos es de 15 con 3 saltos de distancia entre la posición inicial y la final.

- **Si empieza en la posición 2 (estado inicial) y debe aspirar en la posición 7(objetivo), en promedio ¿Cuántos movimientos (acciones) tiene que hacer el agente para llegar a la posición deseada?**

```
Posicion de la basura: 7
Posicion de la aspiradora: 2
¿Cuántos intentos desea ejecutar? 10
```

```
Registro de movimientos: [17, 38, 22, 38, 16, 36, 41, 9, 86, 11]
--------------------------------------------------
Promedio del total de movimientos: 17.0
Distancia entre la basura y la aspiradora: 5
```

  El promedio de movimientos es de 17 con 5 saltos de distancia entre la posición inicial y la final.

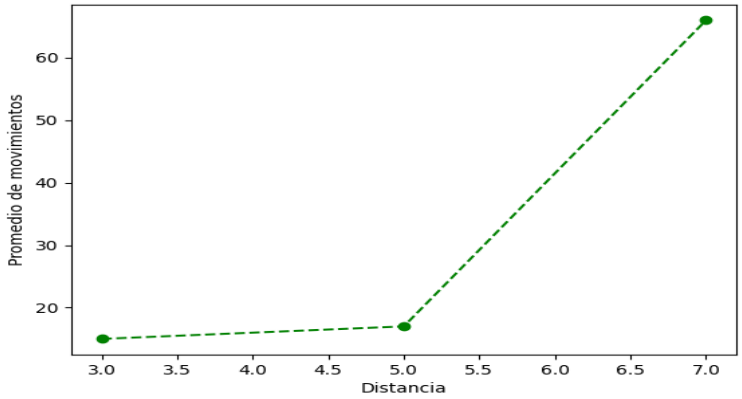
- **Si empieza en la posición 2 (estado inicial) y debe aspirar en la posición 9(objetivo), en promedio ¿Cuántos movimientos (acciones) tiene que hacer el agente para llegar a la posición deseada?**

```
Posicion de la basura: 9
Posicion de la aspiradora: 2
¿Cuántos intentos desea ejecutar? 10
```

```
Registro de movimientos: [66, 23, 28, 108, 63, 120, 33, 78, 28, 41]
--------------------------------------------------
Promedio del total de movimientos: 66.0
Distancia entre la basura y la aspiradora: 7
```

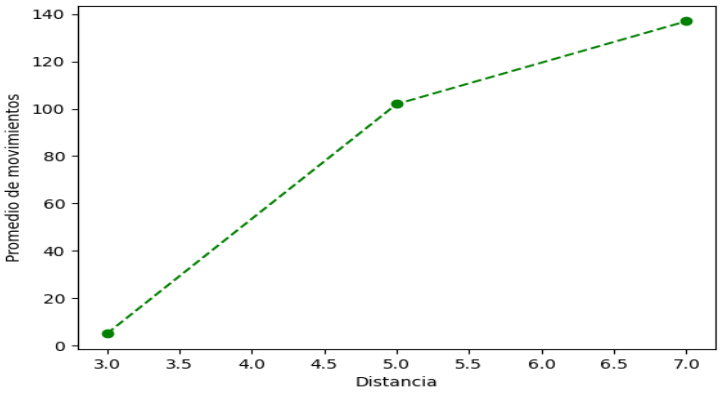
El promedio de movimientos es de 66 con 7 saltos de distancia entre la posición inicial y la final.

- **Graficar distancia vs promedio de movimientos realizado.**



Gráfica 1. Primera ejecución del código con 10 intentos

Posteriormente, con el objetivo de comprobar de mejor manera el comportamiento del código, se realizó una segunda prueba, pero aumentando el número de intentos a 100, obteniendo la siguiente gráfica.



Gráfica 1. Segunda ejecución del código con 100 intentos

- **¿La relación distancia vs promedio de movimientos es lineal?**

Para una mejor comparación se realizaron dos pruebas en las que se usaron diferente número de intentos para que la aspiradora limpie la basura a partir de 3 distintas distancias. Como se puede observar en las gráficas, la línea que se forma no es lineal, aunque tiende a ser lineal conforme se aumenten los intentos ya que eso genera que el promedio de movimientos sea un poco más proporcional a la distancia, pero nunca se observa un comportamiento totalmente lineal. En conclusión, la relación distancia vs promedio de movimientos no es lineal, pero se mantiene creciente conforme los movimientos aumentan.

Conclusión:

Por medio de esta práctica se implementaron los conceptos sobre Agentes Reactivos Simples Basados en Objetivos usando el lenguaje de programación Python, para así poder resolver este problema, logrando así el objetivo de comprender los conceptos vistos en clase sobre el tema y tomando en cuenta de que entre más grande sea la distancia de movimientos más grande será el promedio de movimientos que este necesita para llegar a su objetivo.