

Trabajo Práctico 2: Agentes racionales

1. Para cada una de las siguientes actividades, describa el PEAS para cada tarea y caracterizarlo en término de las propiedades vistas.
 - a) Jugar al CS (o cualquier otro 3d Shooter).
 - b) Explorar los océanos.
 - c) Comprar y vender tokens crypto (alguno).
 - d) Practicar tenis contra una pared.
 - e) Realizar un salto de altura.
 - f) Pujar por un artículo en una subasta.
2. Implementar un simulador que determine la medida de rendimiento para el entorno del mundo de la aspiradora según las siguientes especificaciones:
 - La medida de rendimiento premia con un punto al agente por cada recuadro que limpia (aspira) en un período de tiempo concreto, a lo largo de una «vida» de 1000 acciones.
 - La «dimensión» de la grilla se conoce a priori pero la distribución de la suciedad y la localización inicial del agente no se conocen (aleatorio).
 - Las cuadrículas se mantienen limpias y al aspirar se limpia la cuadrícula en la que se encuentra el agente.
 - Las acciones permitidas son:
 - Arriba
 - Abajo
 - Izquierda
 - Derecha
 - Limpiar (aspirar)
 - NoHacerNada
 - Las acciones Izquierda, Derecha, Arriba, Abajo mueven al agente en dichas direcciones, excepto en el caso en que lo puedan llevar fuera de la grilla.
 - El agente percibe su locación y si esta contiene suciedad.

A continuación se describe una posible interfaz a utilizar:

```
class Environment:
    def __init__(self, sizeX, sizeY, init_posX, init_posY, dirt_rate)
    def accept_action(self, action):
    def is_dirty(self):
    def get_performance(self):
    def print_environment(self):
```

3. Implementar un **agente reflexivo simple** para el entorno de la aspiradora del ejercicio anterior. A continuación se describe una posible interfaz a utilizar para el agente:

```

class Agent:
    def __init__(self,env): # recibe como parámetro un objeto
                            # de la clase Environment

    def up(self):
    def down(self):
    def left(self):
    def right(self):
    def suck(self): # Limpia
    def idle(self): # no hace nada
    def perspective(self,env): # sensa el entorno
    def think(self): # implementa las acciones a seguir por el agente

```

4. Evaluar el desempeño del agente reflexivo, esto es medida de desempeño y unidades de tiempo consumidas, para:

- Entornos de: 2×2 , 4×4 , 8×8 , 16×16 , 32×32 , 64×64 , 128×128 .
- Porcentaje de suciedad en el ambiente: 0.1, 0.2, 0.4, 0.8

Repetir 10 veces cada combinación.

Nota: Se recomienda elaborar una tabla en google sheets (o similar) en donde se presenten los resultados en términos de la medida de rendimiento para cada uno de los casos. Esto luego se podrá utilizar para realizar alguna visualización de los resultados.

5. Repetir el procedimiento descrito en el ejercicio anterior, para el caso de un agente con comportamiento totalmente aleatorio. Esto es, en cada periodo de tiempo, el agente toma una acción al azar.

6. Responder las preguntas 2.10 y 2.11 de AIMA 3^{ra} Edición.

7. Forma de entrega:

- a) Dentro del repositorio **ia-uncuyo-2024**, crear una carpeta con el nombre **tp2-agentes-racionales**.
- b) Dentro de la carpeta **tp2-agentes-racionales**, colocar un archivo con el nombre **tp2-peas.md** con la respuesta del ejercicio (1), y un archivo con el nombre **tp2-aima-questions.md** con las respuestas del ejercicio 6.
- c) Dentro de la carpeta **tp2-agentes-racionales**, crear una nueva carpeta **code**, que incluya la implementación de los ejercicios (2)-(5), la cual se debe realizar en **python3**. Pueden usarse todas las bibliotecas y/o frameworks que considere necesarios. En este último caso se deberá demostrar que se conoce lo que realiza el framework utilizado.
- d) Dentro de la carpeta **tp2-agentes-racionales**, colocar un archivo con el nombre **tp2-reporte.md** con la evaluación de desempeño de los ejercicios (4) y (5). Incluya en este reporte, gráficos que den cuenta de la comparación. Organice el reporte siguiendo la estructura:
 - Introducción: descripción general del problema.
 - Marco teórico: descripción teórica y general del funcionamiento del entorno y de los agentes, y sus principales elementos.
 - Diseño experimental: descripción de los experimentos, es decir, descripción de la medida de rendimiento, de las diferentes configuraciones del entorno (tamaño y porcentaje de suciedad), cantidad de repeticiones.
 - Análisis y discusión de resultados: presentar los resultados obtenidos en los experimentos, realizando un breve análisis de dichos resultados, e incluyendo gráficos o tablas que los resuman.
 - Conclusiones: conclusiones finales de los resultados obtenidos.
- e) Dentro de la carpeta **tp2-agentes-racionales**, crear una nueva carpeta **images**, que incluya todos los gráficos e imágenes utilizados en el reporte final.

Ejemplo:

```
ia-uncuyo-2024/  
  tp2-agentes-rationales/  
    code/*.py  
    images/  
    tp2-peas.md  
    tp2-aima-questions.md  
    tp2-reporte.md  
    Readme.md
```