

# Pràctica Bloc I

14-X-2024

## Objectiu

L'objectiu d'aquest exercici avaluable és aplicar els conceptes i tècniques apreses durant el curs d'Intel·ligència Artificial per a dissenyar i desenvolupar agents.

## Materials

La pràctica es realitzarà amb Python emprant l'entorn creat per la realització d'aquesta i disponible al repositori de l'assignatura <sup>1</sup>.

## Descripció del problema

El problema consisteix en un joc definit en un tauler de  $N \times N$ , on l'agent ha de trobar el camí des de la seva posició inicial fins a una casella objectiu. Les parets del laberint impedeixen el moviment a través d'algunes caselles, i l'agent només pot moure's en quatre direccions: nord, sud, est o oest.

Tant la posició dels agents, la posició de les parets i de la diana són completament aleatòries.

L'agent a desenvolupar pot realitzar tres accions:

- **MOURE.** L'agent es pot moure a una casella adjunta (Nord, Sud, Est o Oest) si no hi ha una paret.
- **BOTAR.** L'agent es pot moure a una casella a dues distàncies. El bot es realitza seguint les mateixes direccions (Nord, Sud, Est o Oest).
- **POSAR PARET.** L'agent pot afegir a una casella adjunta (Nord, Sud, Est o Oest) una nova paret.

Aquestes accions estan representades per un enumerat (**Accions**) i tenen com a paràmetre la direcció ("N", "S", "E", "O"). Al següent codi teniu un exemple de com s'ha de retornar l'acció i la direcció.

---

<sup>1</sup>[https://github.com/miquelmn/ia\\_2024](https://github.com/miquelmn/ia_2024)

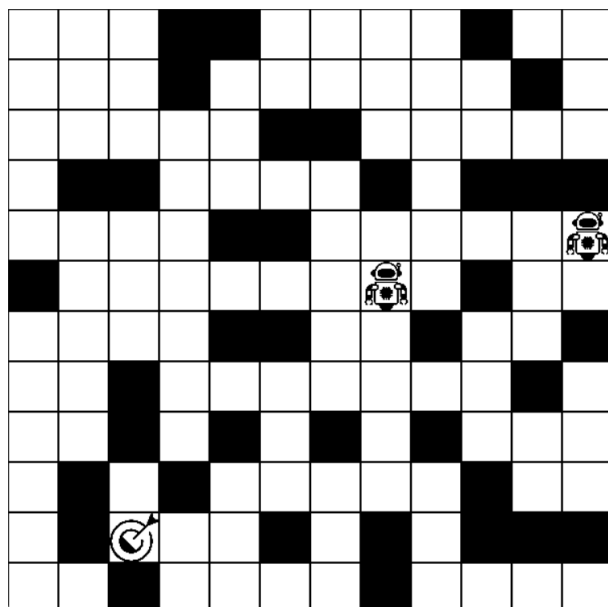


Figura 1: Exemple de joc a resoldre amb la pràctica. L'objectiu ve definit per la icona de diana i la posició dels agents per les icones de robots.

Acció	Pes
MOURE	1
BOTAR	2
POSAR PARET	4

Taula 1: Diferents accions i pesos associats

```

1 def actua(self, percepcio: dict) -> Accions | tuple[Accions, str]:
2     ...
3     return Accio.MOURE, "N"
4 
```

Cada una de les accions té un pes assignat, aquest es pot veure a la Taula 1. Aquest pes s'ha d'emprar per realitzar cerques informades.

Per altra part, l'agent rebrà un diccionari (percepció) amb la següent informació:

- **PARETS.** Retorna una llista de tuples indicant les coordenades a on hi ha parets.
- **TAULELL.** Retorna una matriu  $N \times M$  de caràcters. "0" indica que la casella està ocupada i "." que aquesta es troba buida.
- **DESTI.** Tupla indicant les coordenades de l'objectiu.
- **AGENTS.** Diccionari amb la següent forma  $\{ \text{"NOM\_AGENT"}: (pos_x, pos_y) \}$ .



Mida tauler	Nota màxima
$6 \times 6$	6
$8 \times 8$	8
$10 \times 10$	10

Taula 2: Mides de tauler i notes associades

## Treball a realitzar

El treball que s'ha de dur a terme és implementar un conjunt d'algoritmes de cerca i realitzar un informe pdf comparant els seus rendiments des de diferents punts de vista.

Els algoritmes a implementar per assolir una nota de 10 són els següents:

- Una cerca primer en profunditat.
- Algoritme d'A\*.
- Algoritme MiniMax amb poda alfa-beta (s'han d'emprar dos agents).

La mida del taulell a resoldre definirà la nota màxima que es pot obtenir. Aquesta relació es pot observar a la Taula 2.

La mida del taulell ve definida pels paràmetres `mida_taulell` dels objectes de les classes `Viatger` i `Laberint`.

## Condicions

És condició *sine qua non* perquè la pràctica es corregeixi que s'executi a l'ordinador del professor i la realització de la corresponent entrevista. A més, **es prohibeix** l'ús de qualsevol mètode del fitxer `joc.py` exceptuant l'enumerat `Accions`.

Únicament es valoraran les pràctiques dels grups que hagin realitzat l'entrevista personal.

## Format de l'equip

La pràctica es realitzarà en grups de dues persones.

## Data d'entrega

El darrer dia per entregar la pràctica serà el dia 28 d'octubre de 2024.