

# Homework 1: pathfinding con A\*

L'environment consiste in un grid world delimitato a una regione rettangolare, in cui un certo numero di celle sono "piene" e quindi inaccessibili. L'agente può muoversi nelle 4 direzioni, una cella per mossa. Si vuole programmare un agente che dia l'impressione di vagare senza una meta precisa. Più precisamente, l'agente deve ripetutamente:

1. estrarre una cella (non piena) da raggiungere all'interno della regione;
2. cercare un percorso ottimale che la raggiunga, se esiste;
3. seguire il percorso scelto.

A tal fine si sceglie di usare l'algoritmo A\* per implementare il passo 2.

## Esercizio 1

Completare il codice Python nel file "studente.py" (vedi la sezione "Struttura del codice"), in maniera che il comportamento dell'agente sia **quello descritto sopra**. In particolare, vanno implementate le seguenti funzioni:

- **l'algoritmo A\*** (si può usare la classe Node già presente nel codice per rappresentare i nodi);
- la funzione che genera i **successori** di uno stato e i **costi** associati a ogni transizione;
- **l'euristica** da utilizzare in A\*.

Dato che la funzione di costo è a scelta dello studente, il significato di *percorso ottimale* è da intendersi come quello di **percorso con numero di mosse minimo**.

## Esercizio 2

Modificare la funzione di costo e/o l'euristica in maniera che, anche se non è possibile arrivare al goal, **venga comunque ritornato** un percorso ottimale che contenga il **minor numero possibile di celle piene**. Si noti che l'implementazione del passo 3 fornita ignorerà (senza considerarlo un errore) la parte del percorso successiva alla prima cella piena.

## Struttura del codice

Il codice è diviso in due file: "studente.py", che contiene l'implementazione del passo 2 dello studente, e il file "main.py", che contiene il resto dell'implementazione e richiama il file "studente.py". Per testare la propria soluzione si può eseguire come script il file "main.py". Per eseguire il codice è necessario aver installato la libreria PyGame. Per ulteriori dettagli si vedano i commenti nel template del file "studente.py" fornito.

## Regole di consegna

- Va consegnato solo il file "studente.py", **rinominato in "<matricola>.py"** (ad es. **"123456.py"**), con scadenza **5 giorni prima della data dell'esame** che si intende sostenere.
- Gli homework sono validi **solo per gli appelli di esame di Giugno e Luglio**.

## Valutazione

Ogni homework è valutato da 0 a 30 punti. Il punteggio finale per questa parte del corso verrà calcolato come punteggio  $(\text{voto\_hw1} + \text{voto\_hw2} + 2 * \text{voto\_scritto}) / 4$