

Esercizio 0.

Ho aggiunto in entrambi i codici degli agenti le righe di codice:

```
global MyImage  
MyImage = PhotoImage(file = "Agentvacuum.png")
```

File relativi:

```
0-VacuumPlanner.ipynb  
0-Vaaccum_WiderPerception.ipynb  
+ AgentVacuum.png
```

Esercizio 1.

a) Ho optato per la seguente strategia:

Se l'agente si trova nella stanza più a sinistra, allora dopo aver controllato lo stato di pulizia della stanza (ed eventualmente pulito) fa due passi verso destra, continuando chiaramente a valutare ad ogni passo se si trova in una stanza sporca ed in tal caso pulendola.

Viceversa, se si trova nella stanza più a destra, fa il contrario: ovvero due passi verso sinistra.

Se si trova al centro vuol dire che:

1. Al passo precedente si trovava nella cella alla sua sinistra, in tal caso controlla se si trova in una stanza sporca, eventualmente la pulisce e fa un passo verso la stanza a destra;
2. Al passo precedente si trovava nella cella alla sua destra, anche in questo caso controlla se c'è sporco, eventualmente lo pulisce e si muove verso la stanza a sinistra;
3. L'agente è appena "nato" al centro, in tal caso sceglie una stanza casuale in cui muoversi e vi si dirige.

Per realizzare questa strategia ho dovuto dichiarare una variabile globale, "Mode", inizializzata di default a "Nato_al_centro".

La variabile Mode è modificata ogni volta che l'agente sceglie se muoversi a destra o a sinistra (rispettivamente viene posta dunque a "Right" o "Left" per scegliere il movimento).

Se l'agente si trova nella stanza centrale ed essa è pulita fa un controllo sulla variabile Mode: se è "Left" vuol dire che l'ultimo passo è stato a sinistra, quindi deve fare un altro passo a sinistra.

Al contrario, se legge "Right" vuol dire che deve fare un altro passo a destra.

Se legge "Nato_al_centro" sceglierà di fare una mossa casuale e modificherà la variabile Mode di conseguenza.

Se si trova nella cella più a sinistra o in quella più a destra, l'agente non legge la variabile globale ma si muove rispettivamente a destra o a sinistra, andando però a modificare la variabile Mode.

b) Per fare ciò che è stato richiesto ho modificato la funzione program() nel seguente modo:

Se il flag Cleaned è posto ad 1 (ossia se non c'è sporco in nessuna stanza) l'agente ripete il ciclo "for Room in range(3) [...]" atto a cercare tra le stanze dello sporco rimanendo però fermo. Per garantire quest'ultima condizione ho dovuto modificare l'azione NextAction che di default era "Right" portandola a "NoOp".

File relativi:

```
1-VacuumPlanner_ver_2.0.ipynb  
1-Vaaccum_WiderPerception.ipynb
```

Esercizio 2.

Per quanto riguarda la misurazione delle performance dei due agenti ho cambiato le misure:

Un agente guadagna otto punti per ogni cella pulita e ne perde due per ogni spostamento, gli agenti partono entrambi da 0 punti e non più da 20 come succedeva nel codice del wider-perception.

Ho aggiunto nella funzione `execute_action()` una stampa delle performance attuali dopo l'esecuzione dell'azione scelta.

a) Chiaramente Wider Perception è più performante in quanto, avendo una percezione di tutto l'ambiente sa sempre dove andare a cercare lo sporco e non fa movimenti a vuoto, mentre il VacuumPlanner se non trova lo sporco nella sua cella si muove a caso nell'ambiente finché non trova dello sporco da pulire.

Ho modificato anche le varianti dei due agenti ottenute dopo l'esercizio uno aggiungendo anche ad esse metodi per misurare e visualizzare le performance.

File Relativi:

2_Vacuum_Wider_perception

2_Vacuum_Planner.ipynb

Esercizio 3.

Per confrontare i 4 agenti ho utilizzato una sequenza di 23 “iterazioni”. Ogni qual volta un agente si trova in una situazione di pulizia totale sporco le celle e clicco next finché non pulisce completamente l’ambiente, poi passo all’iterazione successiva.

Presumiamo che il punto di partenza di ogni agente sia la stanza centrale.

Accanto ad ogni iterazione si tiene traccia delle performance dell’agente al termine della pulizia di quel passo. Le performance degli agenti sono misurate come descritto nell’esercizio 2. (+8 pulizia, -2 mossa)

VacuumPlanner base = **ROSSO**

VacuumPlanner con aggiunta punto 1 = **BLU**

Vacuum_WiderPerception base = **VERDE**

Vacuum_WiderPerception con aggiunta punto 1 = **NERO**

1. (0,0) = sporca, (0,1) = pulita, (0,2) = pulita | -6|6|6|6
2. (0,0) = sporca, (0,1) = pulita, (0,2) = sporca | -2|18|18|18
3. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | 6|26|26|26
4. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 12|32|32|32
5. (0,0) = sporca, (0,1) = pulita, (0,2) = pulita | 18|38|38|38
6. (0,0) = sporca, (0,1) = sporca, (0,2) = sporca | 32|58|58|58
7. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | 40|66|66|66
8. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 46|72|72|72
9. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | 52|74|78|78
10. (0,0) = pulita, (0,1) = sporca, (0,2) = sporca | 64|88|92|92
11. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 72|96|100|100
12. (0,0) = sporca, (0,1) = sporca, (0,2) = sporca | 90|114|118|118
13. (0,0) = pulita, (0,1) = sporca, (0,2) = sporca | 78|128|130|130
14. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 80|136|136|136
15. (0,0) = sporca, (0,1) = pulita, (0,2) = pulita | 80|142|142|142
16. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | 76|146|146|146
17. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 82|152|152|152
18. (0,0) = sporca, (0,1) = pulita, (0,2) = sporca | 86|162|162|162
19. (0,0) = sporca, (0,1) = pulita, (0,2) = sporca | 96|174|174|174
20. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 102|180|180|180
21. (0,0) = sporca, (0,1) = sporca, (0,2) = sporca | 120|198|198|198
22. (0,0) = pulita, (0,1) = sporca, (0,2) = sporca | 124|210|210|210
23. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | 130|216|216|216

Com’era intuibile, gli agenti con vasta percezione si discostano dal VacuumPlanner base: entrambi raggiungono lo stesso punteggio di 216 punti, più alto di 86 punti rispetto al primo.

Il motivo per cui i due agenti con wider perception performano allo stesso modo nella mia misurazione, è che la grande differenza tra i due agenti a vasta percezione la si ha quando l’ambiente è completamente pulito. Quando si preme “next” il primo agente si muove a vuoto perdendo punti, mentre il secondo rimane fermo a percepire, mantenendo il suo punteggio.

In linea teorica dovrebbe essere dunque, più performante il secondo poiché non fa movimenti a vuoto che gli fanno perdere punteggio.

Il Vacuum planner con percezione limitata ma con movimento “meno randomizzato” (ossia quello modificato all’esercizio 1) riesce a tenere testa ai due wider perception, questo perché per la maggior parte dei casi non fa movimenti a vuoto, ma scandendo sempre le stanze avanti e indietro riesce sempre (perdendo pochi punti) a prendere sempre lo sporco (aiutato probabilmente anche dal pattern di sporcizia che ho utilizzato).

Quindi anche avendo meno percezioni ma un pattern “furbo” riesce a tenere testa ad agenti più “dotati”.

Esercizio 4.

Sono riuscito ad aggiungere più stanze ed i relativi pulsanti sulla GUI, ho aggiunto innanzitutto le location dopo gli import (Loc_D e Loc_E) di conseguenza ho modificato tutto il codice andando ad aggiungere queste nuove locazioni, i relativi bottoni, e modificando le funzioni in modo che guardino a 5 e non più a 3 stanze.

Sono riuscito ad implementare la strategia di shortest path andando a modificare anche qui la funzione `program()`, in particolare la clausola *if Cleaned == 0*:

Anche se l'agente si sta già dirigendo verso una stanza contenente dello sporco, prima di decidere se proseguire, muovendosi opera una percezione, ovvero analizza stanza per stanza partendo dalla prima fino ad arrivare alla quinta e valuta se contiene dello sporco. Se non lo contiene passa alla successiva, se lo contiene allora valuta una condizione: nella linea di codice *"if (abs(Room-AgLocation[0])) <= (abs(Dirty-AgLocation[0]))"*

Se la condizione è verificata allora la stanza con indice Room diverrà la nuova stanza obiettivo e l'agente si muoverà verso essa, mettendo per il momento da parte l'obiettivo precedente.

Operando questa percezione prima di ogni passo e scegliendo la sua azione di conseguenza, l'agente sceglierà sempre di dirigersi verso la stanza più vicina.

File relativo:

4_Vacuum_WiderPerceptions.ipynb