

####

Per eseguire i notebook ho dovuto scaricare la cartella MyTest dal deposito di file del corso dello scorso anno accademico, questo perché senza alcuni moduli che erano presenti lì chiaramente il codice python non funzionava

####

### Esercizio 0.

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

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

Ho anche provato a fare un resize dell'immagine ma senza successo: purtroppo è la prima volta che mi approccio alle GUI.

File relativi:

```
VacuumPlanner_esercizio0.ipynb
Vaacum_WiderPerception_Esercizio0.ipynb
+ AgentVacuum.png
```

### Esercizio 1.

a) Avrei voluto inserire più pattern possibili (Ex-> se l'agente si trova a sx fa due movimenti a destra oppure uno solo a destra) ma sarebbe stato complicato gestire le performance. Ho optato invece per la seguente strategia: se l'agente si trova nella cella (0,0) o nella (2,0) si sposta rispettivamente a sinistra o a destra, se invece si trova nella cella (1,0) sceglie casualmente se muoversi a destra o a sinistra.

b) Per fare ciò che è stato richiesto ho modificato program(): se il flag Cleaned è posto a 1 (ossia se non ci sono stanze da pulire) la NextAction diventa 'Percepisci' (definita nella funzione execute\_action()) "Percepisci" sostanzialmente ripete il ciclo "for Room in range(3)" che l'agente compie nella funzione program(); quindi, sostanzialmente percepisce nuovamente l'ambiente rimanendo fermo nella stessa cella senza muoversi. L'agente ripete quest'azione fin quando non identifica dello sporco.

File relativi:

```
VacuumPlanner_esercizio1_performance.ipynb
Vaacum_WiderPerception_esercizio1_performance.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 in precedenza.

Ho tentato anche di creare un bottone header ed una funzione display\_env nel VacuumPlanner in modo da mostrare nella GUI le performance attuali (esattamente come avviene nel WiderPerception) purtroppo creando questi elementi si crea un conflitto nell'agente: ogni qual volta si trova nella cella (0,2) percepisce dello sporco anche se la cella è di fatto pulita (andando dunque a pulire e rimanendo bloccato in quella cella per sempre). Mi sono dunque limitato a stampare nella console le performance dopo ogni mossa.

- 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.

File Relativi:

```
VacuumPlanner_performance.ipynb
Vaacum_WiderPerception_performance.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.

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 | **2** | **8** | **4** | 6
2. (0,0) = sporca, (0,1) = pulita, (0,2) = sporca | **10** | **16** | **16** | 18
3. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | **18** | **24** | **24** | 26
4. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **22** | **30** | **30** | 32
5. (0,0) = sporca, (0,1) = pulita, (0,2) = pulita | **24** | **32** | **36** | 38
6. (0,0) = sporca, (0,1) = sporca, (0,2) = sporca | **30** | **48** | **56** | 58
7. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | **38** | **56** | **64** | 66
8. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **44** | **62** | **70** | 72
9. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | **50** | **68** | **76** | 78
10. (0,0) = pulita, (0,1) = sporca, (0,2) = sporca | **62** | **82** | **90** | 92
11. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **70** | **90** | **98** | 100
12. (0,0) = sporca, (0,1) = sporca, (0,2) = sporca | **70** | **84** | **116** | 118
13. (0,0) = pulita, (0,1) = sporca, (0,2) = sporca | **82** | **98** | **128** | 130
14. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **90** | **106** | **134** | 136
15. (0,0) = sporca, (0,1) = pulita, (0,2) = pulita | **96** | **88** | **140** | 142
16. (0,0) = pulita, (0,1) = pulita, (0,2) = sporca | **92** | **92** | **144** | 146
17. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **96** | **98** | **150** | 152
18. (0,0) = sporca, (0,1) = pulita, (0,2) = sporca | **94** | **100** | **160** | 162
19. (0,0) = sporca, (0,1) = pulita, (0,2) = sporca | **104** | **112** | **172** | 174
20. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **106** | **118** | **178** | 180
21. (0,0) = sporca, (0,1) = sporca, (0,2) = sporca | **124** | **136** | **196** | 198
22. (0,0) = pulita, (0,1) = sporca, (0,2) = sporca | **136** | **150** | **208** | 210
23. (0,0) = pulita, (0,1) = sporca, (0,2) = pulita | **144** | **158** | **214** | 216

Com’era intuibile gli agenti con vasta percezione si discostano dai due VacuumPlanner, Entrambi superano il punteggio di 200. Il distacco di due punti è semplicemente dovuto alla posizione iniziale dell’agente nell’ambiente, le loro performance sono dunque uguali in questo caso.

Chiaramente questo avviene poiché la grande differenza tra i due agenti a vasta percezione la si ha quando l’ambiente è completamente pulito e si preme “next”, cosa che non è avvenuta nella nostra simulazione.

Per quanto riguarda i due vacuum planner i loro punteggi sono simili, anche se il secondo raggiunge un punteggio più alto in quanto la casualità dei movimenti è ridotta rispetto al primo.

### Esercizio 4. (WORK IN PROGRESS)

Sono riuscito ad aggiungere più stanze ed i relativi pulsanti sulla GUI. Manca ancora la parte di implementazione di una nuova strategia.

File relativo:

Vacuum\_WiderPerceptions\_esercizio4.ipynb