

- **Domanda - Spiderman a Manhattan** Spiderman deve raggiungere il Ponte di Brooklyn a New York, e non ha tempo da perdere: Mary Jane è in pericolo. La strada che lo divide dalla sua amata però è l'intera Manhattan visto che lui si trova nell'East Harlem, nella zona Nord della città. A complicare le cose c'è la presenza a New York di tre pericolosi criminali: Octopus, Venom e Goblin. Poiché Octopus si trova nel quartiere di Upper East Side, Venom a Chelsea e Goblin proprio nel Lower East Side, vicino al Ponte di Brooklyn, lui deve liberarsi velocemente di tali ostacoli lungo la strada da Harlem al ponte. **Spiderman deve raggiungere nel modo più breve possibile i tre pericoli criminali, liberarsi di loro poiché non gli facciano perdere tempo, e poi raggiungere May Jane.**
- Spiderman può muoversi tra i quartieri ma SOLO attraversando i loro confini, quindi saltando tra quartieri confinanti. Ogni quartiere ha una sua dimensione, quindi il passaggio da uno all'altro fa perdere a Spiderman tempi diversi, proporzionali alla dimensione dei quartieri usati come partenza di un attraversamento. Nei molti modi di raggiungere il Ponte di Brooklyn, poiché consideriamo fermi i suoi antagonisti durante la sua traversata, Spiderman deve entrare nel quartiere dove uno dei criminali risiede uno dopo l'altro, trovando la strada più breve possibile che gli consenta di battere tutti e tre i suoi avversari. Si consideri nullo il tempo con cui Spiderman batte il suo avversario.
Si richiede di:
 1. Programmare un agente che agisca come Spiderman, descrivendo (in pseudocodice, FOL, o Prolog o in Python) l'ambiente dove si muove Spiderman, la nozione di stato del gioco, il goal ed le informazioni che durante il ciclo del comportamento di SpiderMan egli scambia con l'ambiente
 2. Discutere una o più proprietà dell'ambiente che sono necessarie per calcolare il piano migliore, cioè il cammino più breve. Ad esempio, come si verifica il raggiungimento del goal, o come stabilire quale quartiere di volta in volta è necessario raggiungere perché occupato da un supercriminale o perché lungo la strada che ad esso conduce. Discutere anche le azioni possibili e le loro conseguenze nell'ambiente.
 3. Descrivere il programma che l'agente Spiderman usa per pianificare le sue azioni a Manhattan, e discutere quindi l'algoritmo di ricerca da lui applicato.
 4. (Facoltativo) Discutere come il programma dovrebbe essere modificato nel caso in cui uno (o più) tra i supercriminali possa muoversi tra quartieri diversi, durante lo spostamento di Spiderman nella città.



PEAS

- **P:** Minimizzare il percorso intrapreso da SpiderMan per raggiungere i 3 cattivi e arrivare infine da MJ
- **E:** Quartieri di Manhattan, Venom, Goblin, Octopus, Ponte, SpiderMan
 - **COMPLETAMENTE OSSERVABILE:** SpiderMan conosce esattamente il quartiere in cui si trova in ogni stato dell'ambiente.
 - **DETERMINISTICO:** Ogni azione di spiderman lo porta in al massimo un quartiere e quindi ad ogni azione corrisponde un unico stato.
 - **SINGOLO AGENTE.**
 - **EPISODICO:** Ogni azione eseguita da spiderman in uno stato non influisce sui successivi.
 - **DISCRETO.**
 - **STATICO:** L'ambiente non cambia nel tempo.
 - **NOTO:** Spiderman conosce ogni regola dell'ambiente e tutto ciò che comporta ogni sua singola azione.
- **A:** Le gambe e le braccia di spiderman
- **S:** Gli occhi di spiderman

FORMULAZIONE DEL PROBLEMA

- **STATI:** Ogni stato contiene informazioni sulla posizione dei nemici, della posizione attuale di spiderman, e della posizione del ponte.
 $(NA, (V, G, O, P), CT)$
 NA è la posizione dell'agente, (V, G, O, P) sono i nodi goal, ovvero la posizione dei nemici e infine del ponte, CT è il costo totale s
- **STATO-INIZIALE:** E' lo stato in cui SpiderMan si trova East Harlem.
 $(NA = \text{nodo East Harlem}, (V, G, O, P), CT = 0)$
- **STATI-OBIETTIVO:** Abbiamo diversi stati obiettivo, il primo è quello in cui Spiderman raggiunge Octopus, il secondo Goblin e il terzo Venom e infine in cui Spiderman raggiunge il Ponte
- **AZIONI-POSSIBILI:** $\{ \text{Muoversi in un quartiere vicino} \}$
- **MODELLO DI TRANSIZIONE:** L'azione eseguita da spiderman lo porta a spostarsi un quartiere vicino (ES: $RISULTATO(s, a) = s'$)
- **COSTO-AZIONE:** Dimensione quartiere di attraversamento.

Per rappresentare l'ambiente usiamo una rappresentazione grafo

Lo score dell'agente è rappresentato dal costo totale del cammino seguito da spiderman per raggiungere un certo quartiere: CT

CICLO DI VITA DELL'AMBIENTE

function RUN EVAL ENV (stato, $UPDATE-FN$, spiderman, $PERFORMANCE-FN$) returns CT

local-variables: $CT=0$

repeat

$(NA, (V, G, O, P), CT) \leftarrow GET-PERCEPT(spiderman, stato)$ // siccome l'ambiente è comp-oss , GET-PERCEPT restituirà direttamente lo stato

Action $\leftarrow SPIDERMAN-AGENT((NA, (V, G, O, P), CT))$

state $\leftarrow UPDATE-FN(Action, (NA, (V, G, O, P), CT))$

CT $\leftarrow PERFORMANCE-FN(CT, state)$

until *TERMINATION*(state)

return CT

CICLO DI VITA DELL'AGENTE

function *SPIDERMAN-AGENT*((*NA*, (*V*, *G*, *O*, *P*), *CT*)) **returns** action:

persistent

seq; sequenza di azioni

state;

goal=[G,V,O];

final_goal=[P];

problem;

if seq is empty() and goal is not_empty():

$NG^* \leftarrow FORMULATEGOAL((NA, (V, G, O, P), CT), goal)$ // sceglie il nodo goal più vicino a spiderman in un certo stato, e lo toglie dalla lista goal.

problem $\leftarrow FORMULATEPROBLEM(state, NG^*)$

seq $\leftarrow A^*(problem)$ con *h* distanza di linea d'aria.

else

$NG^* \leftarrow FORMULATEGOAL((NA, (V, G, O, P), CT))$

problem $\leftarrow FORMULATEPROBLEM(state, NG^*)$

seq $\leftarrow A^*(problem)$ con *h* distanza di linea d'aria.

if $NA = NG^*$ then

action=STOP

else:

action=seq[1]

seq=REST(seq)

return action