

Progetto di Intelligenza Artificiale e Laboratorio
Modulo: Planning e Sistemi a Regole
a.a. 2024-2025

Colpito e affondato...?

L'obiettivo del progetto è quello di sviluppare un sistema esperto che giochi ad una versione semplificata della famigerata *Battaglia Navale*.

Il gioco è nella versione "in-solitario", per cui c'è un solo giocatore (il vostro sistema esperto) che deve indovinare la posizione di una flotta di navi distribuite su una griglia 10x10.

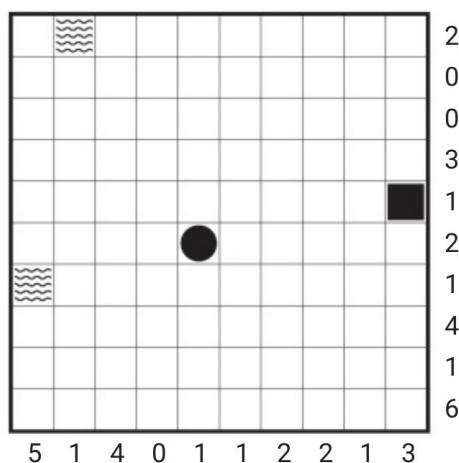
Come di consueto le navi da individuare sono le seguenti:

- 1 corazzata da 4 caselle
- 2 incrociatori da 3 caselle ciascuno
- 3 cacciatorpedinieri da 2 caselle ciascuno
- 4 sottomarini da 1 casella ciascuno

Le navi saranno, ovviamente, posizionate in verticale o in orizzontale e deve esserci almeno una cella libera (cioè con dell'acqua) tra due navi. *Anche nelle diagonali*.

Per rendere più semplice il problema, il contenuto di alcune celle sarà noto fin dall'inizio. Inoltre, in corrispondenza di ciascuna riga e colonna sarà indicato il numero di celle che contengono navi.

Ad esempio, la seguente situazione rappresenta un possibile stato iniziale del problema.



Sapete quindi che in posizione [7, 1] c'è dell'acqua, in posizione [6, 5] c'è un sottomarino, in posizione [5, 10] c'è un pezzo di nave che probabilmente continuerà nelle celle subito sopra e subito sotto. Sapete inoltre, che nella prima riga due celle sono occupate da una nave, mentre nella prima colonna sono 5, ecc.

Il vostro sistema esperto avrà quattro possibili azioni:

- *fire* x y
- *guess* x y
- *unguess* x y
- *solve*

L'azione *fire* è l'equivalente di un'azione percettiva e vi permette di vedere il contenuto della cella [x, y].

L'azione *guess* serve per indicare che il vostro sistema esperto ritiene ci sia una nave in posizione [x, y]. Questa azione è da considerarsi un'ipotesi, per cui è ritrattabile in un momento successivo, cioè il vostro sistema esperto potrebbe tornare sui suoi passi con il comando *unguess*.

Il comando *solve* è da usarsi solo quando il vostro sistema esperto ritiene di aver risolto il gioco, l'azione termina il gioco attivando il calcolo del punteggio secondo la seguente formula:

$$(15 * gok + 20 * sink) - (10 * gko + 10 * safe + 20 * nf + 20 * ng)$$

dove:

- gok è il numero di celle guessed corrette
- sink è il numero di navi totalmente affondate
- gko è il numero di celle guessed errate
- safe è il numero di celle che contengono una porzione di nave e che sono rimaste inviolate (né guessed né fired)
- nf è il numero di fire non usate
- ng è il numero di guess non usate

Per rendere le cose un po' più interessanti, avete solamente 5 *fire* a disposizione. Inoltre, in un dato momento non possono esserci più di venti caselle marcate "guessed".

Lo scopo del gioco è quindi marcare tutte le caselle che contengono una nave come guessed, o eventualmente averle colpite con *fire*.

Riassumendo

L'obiettivo del gioco è di annotare come "guessed" le venti caselle sotto cui si trovano le navi. Notate che dove marcare come guessed anche una cella che avete osservato con una fire e in cui avete trovato una porzione di nave. Solo in questo modo il punteggio sarà calcolato correttamente.

Dovete implementare almeno due varianti di giocatore che seguano due strategie diverse da mettere poi a confronto.

Presentazione

La presentazione deve evidenziare (in 10 minuti):

- come avete modellato la conoscenza, in particolare se avete dovuto creare fatti (ordinati o non-ordinati) per modellare ipotesi ecc.
- come avete modellato le regole di expertise. Sarebbe bene che provaste ad implementare diverse strategie di soluzione. Cioè diversi sistemi esperti più o meno capaci e li metteste a confronto.
- dovete fare prove con scenari alternativi, variando, oltre alla posizione delle navi, anche l'osservabilità iniziale (più o meno caselle note all'inizio).
- dovrete poter individuare i limiti della vostra soluzione: come si comporta se all'inizio non conosce nulla?

Materiale a disposizione

Le regole del gioco (il funzionamento delle diverse azioni e il calcolo del punteggio) sono già state implementate e scaricabili dalla pagina moodle del corso. Vi rimando al video per maggiori dettagli. Non siete autorizzati a modificare l'ambiente, a meno che non vi siano bachi, vi chiedo però di avvisarmi nel caso.

È importante che ricordiate che l'ambiente può eseguire una sola azione per volta. Quindi, anche se avete pianificato una sequenza di azioni, dovete darle all'ambiente una per volta, vedere il video.

NB Dopo 100 azioni il gioco termina automaticamente (viene in automatico eseguita l'azione *solve*).

Per la realizzazione di scenari alternativi vi metto a disposizione un editor di mappe che produce la codifica CLIPS necessaria per modellare il mondo dal punto di vista dell'ambiente.

NB L'editor di mappe è molto artigianale e non fa controlli sul numero di navi, sulla loro dimensione e sull'esistenza di spazi tra una nave e l'altra. Questi controlli dovrete farli voi nel momento in cui definite una mappa.