

1 Traccia - Robot 1

Si vuole sviluppare un algoritmo per il cammino di un robot in un labirinto. La stanza è pavimentata a tasselli quadrati (caselle) ed è dotata di pareti esterne e interne (vedi Figura 1 per un esempio). Il robot si può muovere nelle 8 direzioni adiacenti. Devono essere previsti almeno tre diversi scenari (livelli di difficoltà). Nella stanza sono presenti degli oggetti di diverso colore (red, green, yellow, cyan) che compaiono e scompaiono casualmente sul percorso. Il robot può assumere quattro stati *pursuit*, *evade*, *flee*, *seek*. Se il robot si trova in prossimità di un oggetto cambia stato secondo lo schema della Macchina a Stati Finiti di Figura 2.

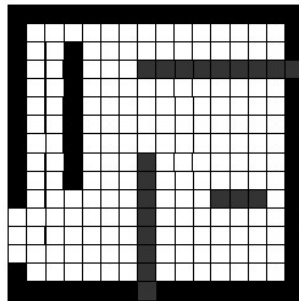


Figure 1: Esempio di percorso

Le strategie disponibili per ogni stato sono:

- **Pursuit e Seek** - il robot si muove nella direzione dell'uscita di una singola cella alla volta.
 - Utilizzare l'algoritmo di Dijkstra (vedi descrizione sotto).
- **Flee** - il robot si muove nella direzione dell'uscita di due celle alla volta.
 - Utilizzare l'algoritmo di Dijkstra (vedi descrizione sotto).
- **Evade** - il robot avanza in maniera casuale di una singola cella.

Scrivere un programma per la gestione del labirinto. Il programma deve permettere di registrare il robot con il suo nome e cognome e visualizzare, ad ogni inizio e fine partita, la classifica dei risultati migliori ottenuti, da tutti i robot, in tutte le partite (minore numero di passi per raggiungere l'uscita). Visualizzare il percorso del robot dopo ogni passo, mostrando la stanza, la sua posizione e quella degli oggetti.

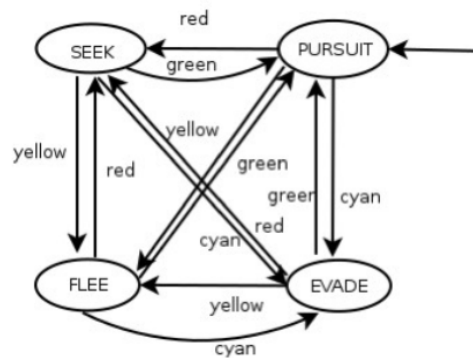


Figure 2: Macchina a stati finiti

Algoritmo di Dijkstra

L'algoritmo di Dijkstra è un algoritmo utilizzato per cercare i **cammini minimi** (o *Shortest Paths*) in un grafo con o senza ordinamento, ciclico e con pesi non negativi sugli archi.

Per ulteriori dettagli sull'implementazione, vedere:

https://it.wikipedia.org/wiki/Algoritmo_di_Dijkstra