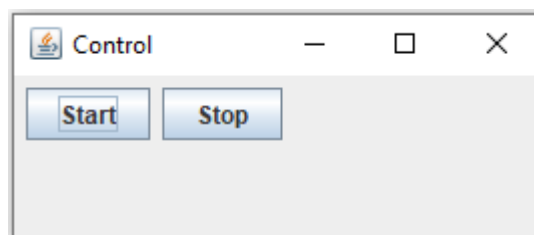
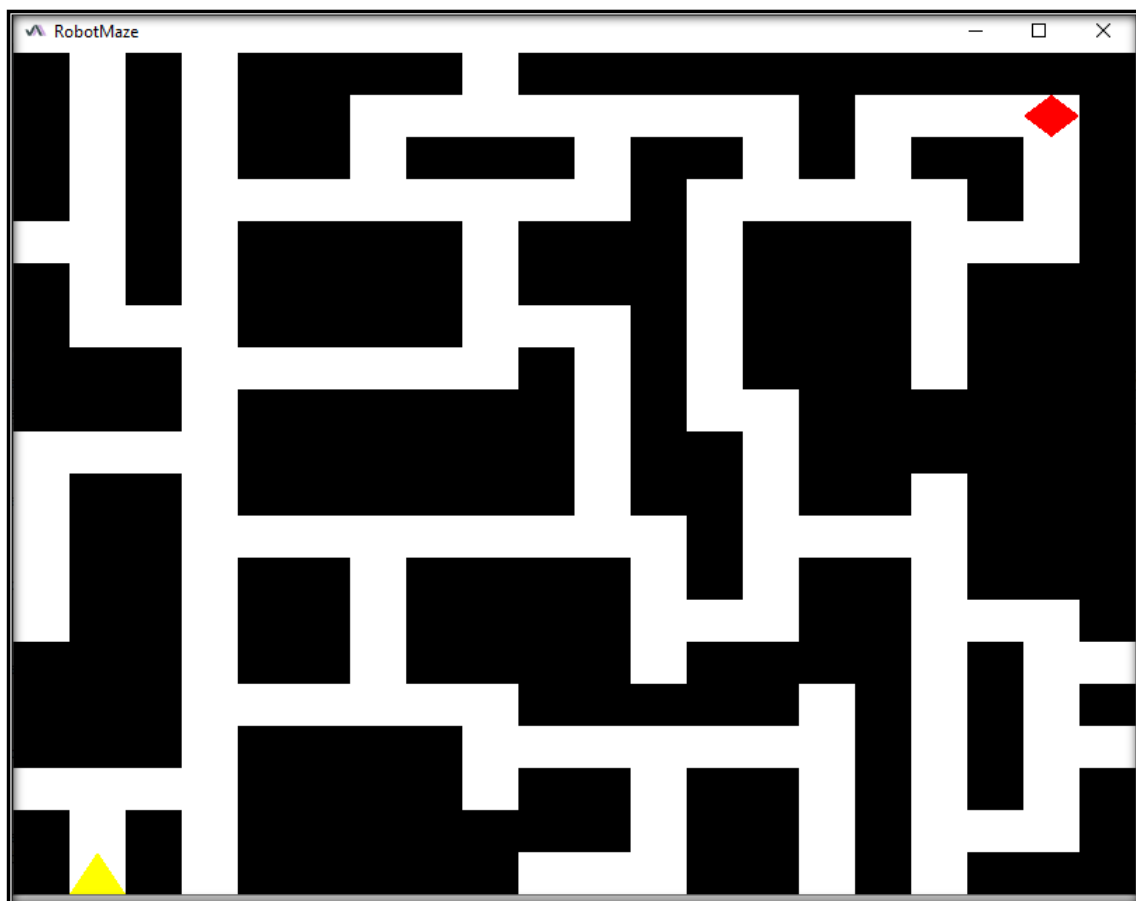


RobotMaze2.0

Autore: Giuseppe Mario Malandrone

Mail: giusemaland@gmail.com

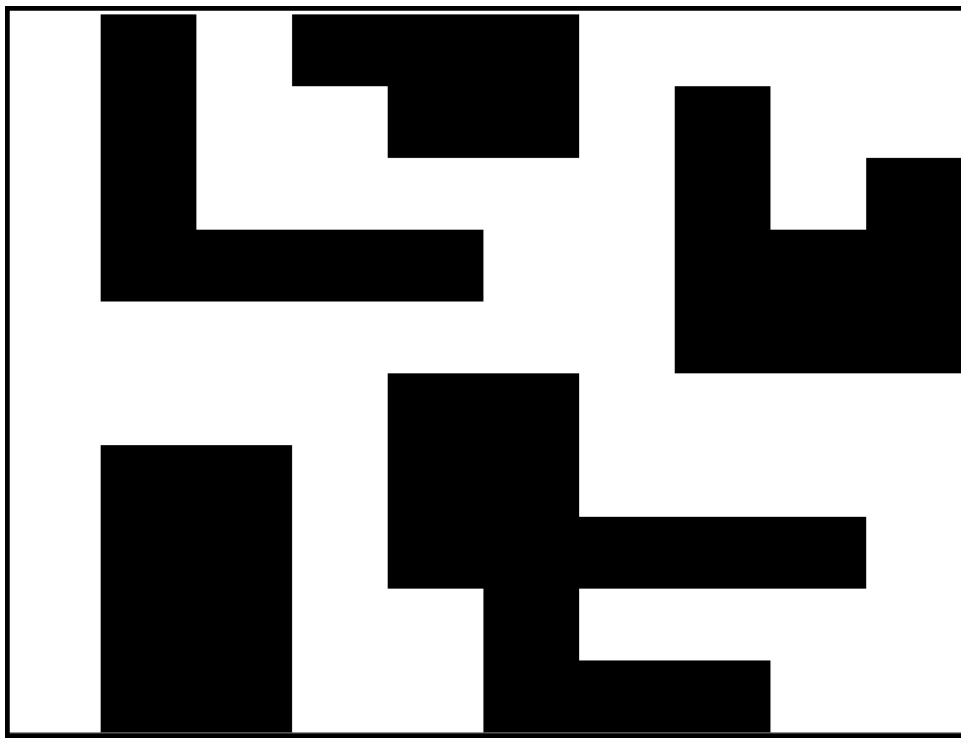


1. Introduzione

RobotMaze è un'applicazione Java che implementa un sistema di intelligenza artificiale. L'utente può generare un labirinto e posizionare al suo interno un obiettivo ed un automa. L'obiettivo irradia un campo che si propaga nel labirinto lungo tutte le direzioni, attenuandosi all'aumentare della distanza. L'attenuazione del campo sarà tanto maggiore quanti più sono gli ostacoli attraversati nel percorso da sorgente a punto campo. Avviando la simulazione, l'automa si muoverà lungo il labirinto alla ricerca dell'obiettivo basandosi esclusivamente sull'ampiezza del segnale ricevuto e sulla propria conoscenza delle zone del labirinto. Sulla base di queste informazioni, il motore inferenziale implementato consentirà all'automa di raggiungere l'obiettivo in un tempo finito.

2. Il Labirinto

Il labirinto è rappresentato come una matrice NxN in cui ogni elemento è una cella. Ogni cella è descritta da una coppia di numeri interi (x,y) , ovvero la posizione in coordinate cartesiane e da un valore logico che indica se essa è un muro o se è vuota. Nella grafica le zone nere costituiscono i muri mentre le zone bianche i vuoti.



3. L'Obiettivo



L'obiettivo è un'entità fittizia descritta dalla sua posizione nel labirinto in coordinate cartesiane (x,y) e dalla potenza massima del segnale che esso irradia.

3.1 Calcolo della potenza irradiata

Prima che l'automa inizi la ricerca, la potenza irradiata dall'obiettivo viene preliminarmente calcolata in tutte le celle del labirinto secondo la seguente formula:

$$P = 10\log\left(\frac{P_{max}}{d^2}\right) - \log(L)$$

Dove:

- P_{max} è la potenza massima del segnale irradiato dall'obiettivo. Essa è impostabile dall'utente come parametro d'ingresso al programma;
- d è il numero di celle nel percorso in linea retta dalla posizione dell'obiettivo (x_0, y_0) , ad un generico punto campo (x_p, y_p) , calcolate analiticamente secondo la seguente formula:

$$d = \text{ceil}\left(\sqrt{(x_p - x_0)^2 + (y_p - y_0)^2}\right)$$

- L è il coefficiente di attenuazione del segnale nel medesimo percorso da obiettivo a punto campo, calcolato come la somma dei singoli contributi di attenuazione di ogni cella, secondo la seguente formula:

$$L = \sum_i l_i$$

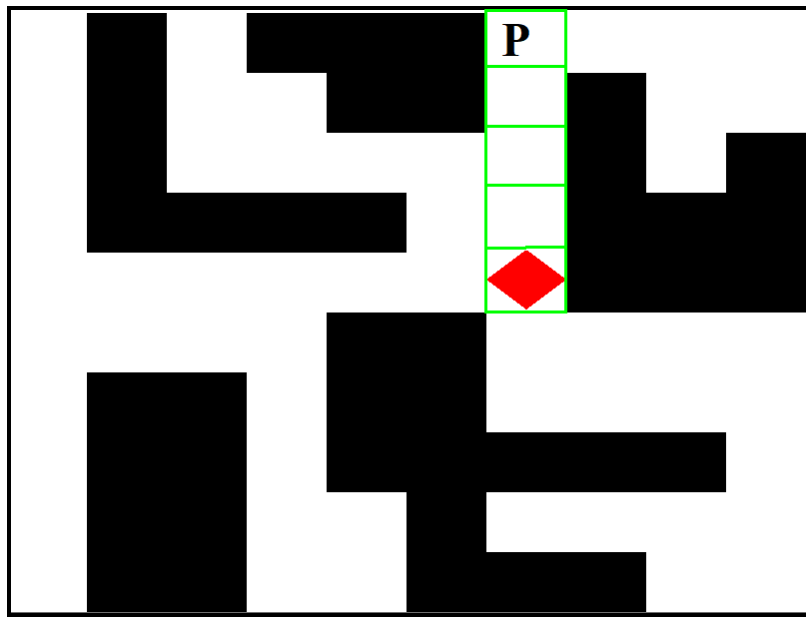
dove il contributo di attenuazione relativo alla singola cella vale:

$$l_i = \begin{cases} 1 & \text{vuoto} \\ 3 & \text{muro} \end{cases}$$

Le coordinate delle celle appartenenti ad un determinato percorso da obiettivo a punto campo, vengono calcolate nei modi seguenti:

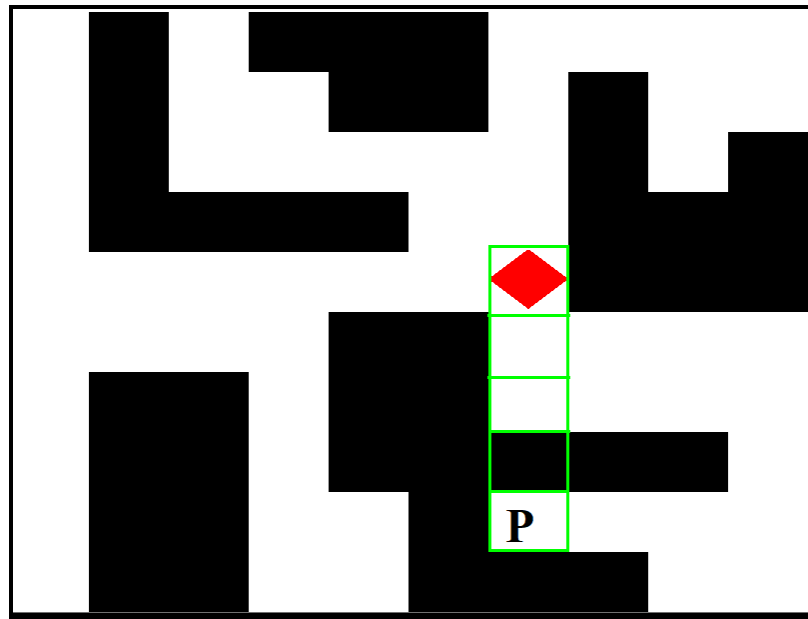
➤ **Caso 1:** punto campo ed obiettivo sulla stessa ascissa ($x_p = x_0$)

▪ Se $y_0 < y_p$



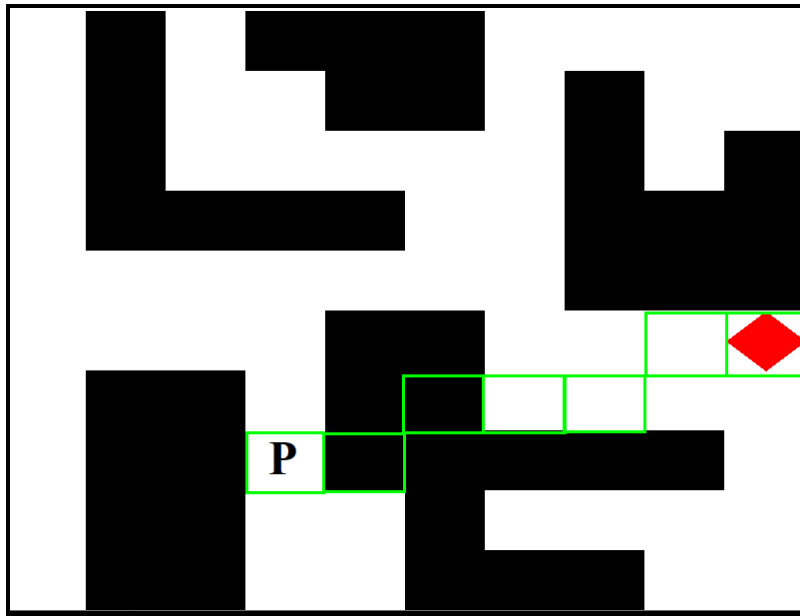
$$\begin{cases} x = x_0 \\ y \in [y_0, y_p] \end{cases}$$

- Se $y_0 > y_p$



$$\left\{ \begin{array}{l} x = x_0 \\ y \in [y_p, y_o] \end{array} \right.$$

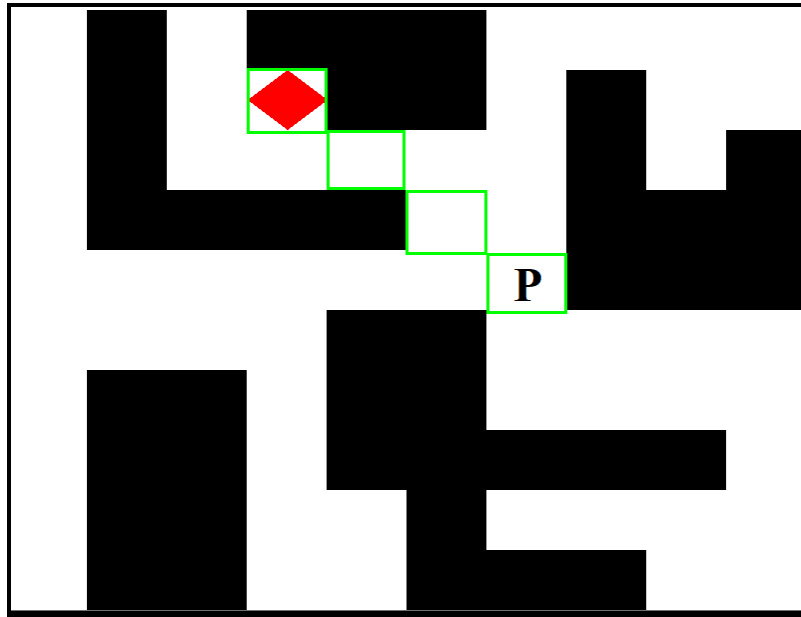
- **Caso 2:** punto campo a sinistra dell'obiettivo ($x_p < x_0$)



$$\begin{cases} x \in [x_p, x_o] \\ y = \text{round}(m(x - x_p) + y_p) \end{cases}$$

Con $m = \frac{y_p - y_o}{x_p - x_o}$

➤ **Caso 3:** punto campo a destra dell'obiettivo ($x_p > x_0$)



$$\begin{cases} x \in [x_0, x_p] \\ y = \text{round}(m(x - x_0) + y_0) \end{cases}$$

$$\text{Con } m = \frac{y_p - y_0}{x_p - x_0}$$

Le informazioni rappresentate fino a questo punto, costituiscono la base di conoscenza globale dell'ambiente.

4. L'Automa



L'automa è l'entità intelligente del programma. Esso è in grado di memorizzare informazioni e sulla base di queste ultime prendere delle decisioni su come muoversi nel labirinto. La sua base di conoscenza è un sottoinsieme della base di conoscenza globale dell'ambiente e viene aggiornata continuamente con l'avanzare della ricerca.

4.1 Memoria

L'automa dispone di una memoria nella quale memorizza i dati necessari per effettuare la ricerca. Con l'avanzare della simulazione, l'automa memorizza sia le coordinate delle celle esplorate che il numero di volte che una determinata cella è stata esplorata. Inizialmente, la memoria dell'automa contiene le seguenti informazioni:

- La sua posizione nel labirinto in termini di coordinate cartesiane (x_a, y_a) , preimpostate dall'utente;
- Il parametro α (guadagno di potenza ricevuta). Amplifica la potenza del segnale ricevuto. Più questo valore è elevato, più l'automa tenderà a muoversi verso le zone in cui il segnale ricevuto è maggiore. Tale parametro è preimpostato dall'utente;
- Il parametro β (guadagno di memoria). Amplifica il conteggio delle esplorazioni delle celle. Più questo valore è elevato, più l'automa tenderà a muoversi verso le zone del labirinto ancora inesplorate. Tale parametro è preimpostato dall'utente;
- Dati di inizializzazione di sensori e attuatori.

4.2 Sensori e Attuatori

Il movimento dell'automa nel labirinto, oltre che dai dati in memoria, dipende anche dai dati acquisiti da tre sensori:

- **Sensore obiettivo:** rivela la presenza dell'obiettivo nella stessa cella dell'automa;
- **Sensore di profondità:** Opera con l'ausilio di una bussola per rivelare la presenza di muri o vuoti nelle celle adiacenti a quella dell'automa, nelle quattro direzioni NORD, SUD, OVEST, EST;
- **Sensore di campo:** un'antenna in grado di ricevere il segnale irradiato dall'obiettivo.

Sulla base dei dati provenienti dai sensori e da quelli già presenti in memoria, l'automa decide verso quale cella muoversi, calcolando ad ogni passo il seguente vettore di decisione:

$$\underline{D} = \alpha \underline{P} - \beta \underline{E}$$

Dove:

- \underline{D} è un vettore di dimensione pari a 4, ovvero pari al numero delle possibili direzioni in cui effettuare uno spostamento (NORD, SUD, OVEST, EST). I suoi componenti sono dei numeri reali che rappresentano il peso della decisione relativa allo spostamento nella rispettiva direzione, secondo il criterio:
 - D_0 : peso della decisione di spostamento a NORD;
 - D_1 : peso della decisione di spostamento a SUD;
 - D_2 : peso della decisione di spostamento a OVEST;
 - D_3 : peso della decisione di spostamento a EST.

Se in una determinata direzione i è presente un muro, il relativo componente D_i del vettore di decisione assumerà un valore arbitrariamente piccolo;

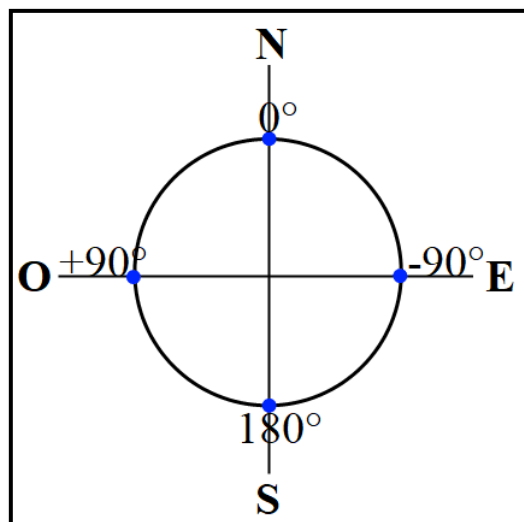
- \underline{P} è il vettore delle potenze misurate nelle celle adiacenti a quella dell'automa, con lo stesso criterio sulle direzioni mostrato al punto precedente;
- \underline{E} è il vettore del numero di esplorazioni delle celle adiacenti a quella dell'automa, con lo stesso criterio sulle direzioni mostrato al punto precedente;

L'elemento maggiore del vettore di decisione, in un dato istante, rappresenta la direzione verso cui l'automa dovrà muoversi. A tale scopo, esso dispone di due attuatori:

- **Attuatore di rotazione:** permette all'automa di ruotare sul proprio asse verticale in modo da orientarsi nella direzione in cui ha deciso di muoversi. Per effettuare ciò, l'automa memorizza il suo orientamento corrente ed in base a quest'ultimo determina l'angolo di rotazione mediante la seguente matrice delle rotazioni:

$$\bar{R} = \begin{bmatrix} 0^\circ & 180^\circ & 90^\circ & -90^\circ \\ 180^\circ & 0^\circ & -90^\circ & 90^\circ \\ -90^\circ & 90^\circ & 0 & 180^\circ \\ 90^\circ & -90^\circ & 180^\circ & 0^\circ \end{bmatrix}$$

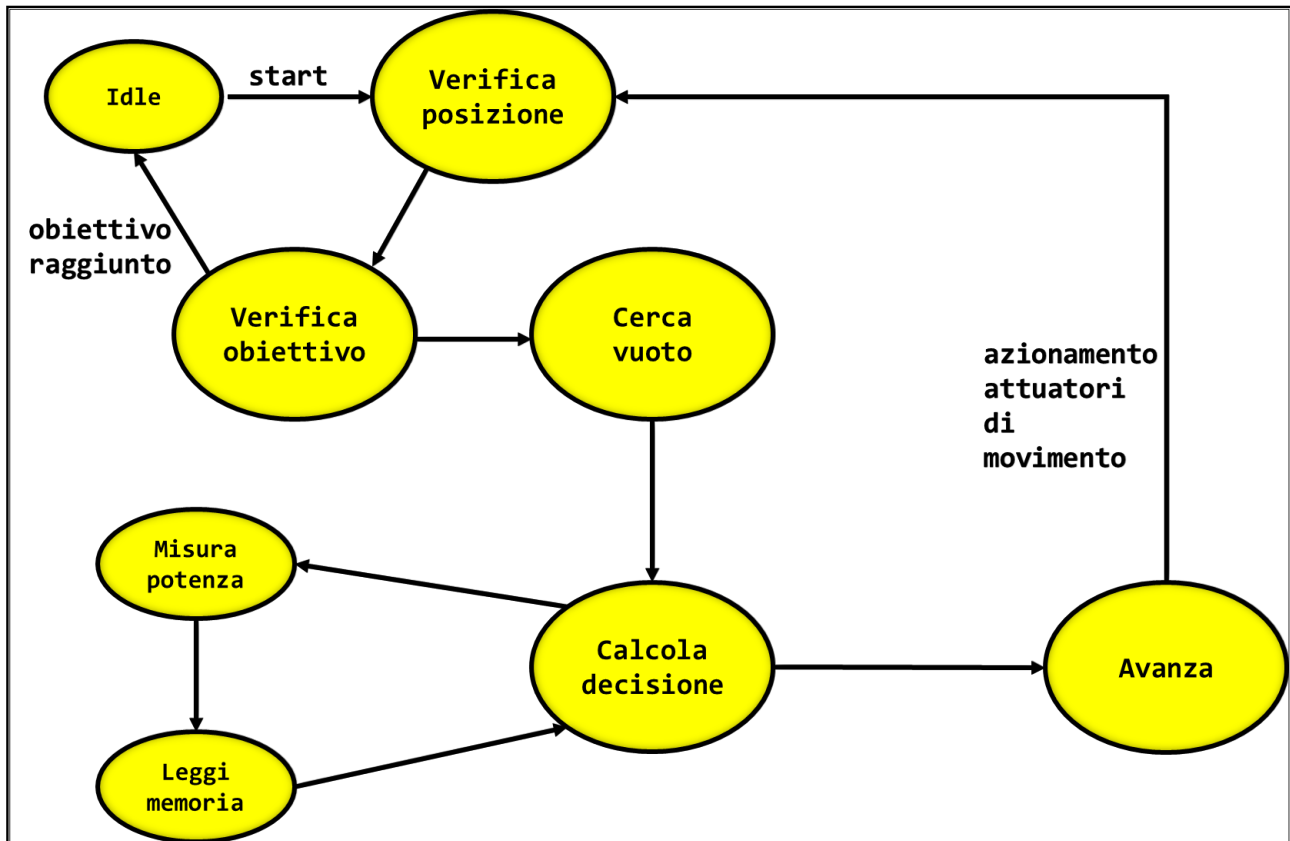
Il generico elemento r_{ij} della matrice indica l'angolo della rotazione che l'automa dovrà compiere per orientarsi nella direzione j quando esso è orientato nella direzione i . La convenzione sulle direzioni è la stessa descritta in precedenza (0=NORD, 1=SUD, 2=OVEST, 3=EST). Inoltre, alla direzione NORD è associata la posizione angolare 0° e come verso positivo viene considerato quello antiorario;



- **Attuatore di avanzamento:** consente all'automa lo spostamento in avanti di una lunghezza pari al lato di una cella.

4.3 Motore Inferenziale

Il motore inferenziale dell'automa è implementato mediante una macchina a stati, rappresentata nel seguente diagramma:



Di seguito si riporta la descrizione di ciascuno stato:

- **Idle**: è lo stato iniziale dell'automa. In questa fase esso è pronto per iniziare la ricerca;
- **Verifica posizione**: l'automa entra in questo stato non appena viene avviata la simulazione. In questa fase viene memorizzata la sua posizione corrente e viene aggiornato il numero di volte che tale zona è stata esplorata;
- **Verifica obiettivo**: in questa fase l'automa, legge i dati relativi al sensore obiettivo. In caso di rilevamento, la ricerca termina con successo;
- **Cerca vuoto**: l'automa entra in questo stato nel caso in cui la cella esplorata non contiene l'obiettivo. In questa fase l'automa legge e memorizza i dati acquisiti dal sensore di profondità;

- **Calcola decisione:** in questo stato l'automa effettua due distinte operazioni:
 - **Misura della potenza:** l'automa legge e memorizza i dati acquisiti dal sensore di campo;
 - **Lettura della memoria:** l'automa preleva dalla propria memoria le informazioni relative alle celle adiacenti alla sua posizione.

Ottenute queste informazioni, sarà possibile calcolare il vettore di decisione;

- **Avanza:** in questo stato l'automa conosce la direzione del prossimo spostamento e sulla base di tali informazioni vengono pilotati gli attuatori di movimento.

5. Tutorial

L'applicazione RobotMaze2.0 è un file .jar, pertanto, per avviare la simulazione è necessario disporre della Java Virtual Machine installata sul proprio dispositivo. Inoltre è necessario che nella stessa directory del file .jar siano presenti i seguenti elementi:

- Cartella natives;
- File "dati_labirinto.txt";
- File "dati_automa.txt";
- File "datioggetto.txt".

5.1 Generare il Labirinto

Il programma consente all'utente di generare arbitrariamente un qualsiasi labirinto di NxN celle. Con N intero, variabile fra 2 e 98. Per una grafica ottimale si consiglia di impostare un valore di N pari. Tutti i dati relativi al labirinto sono memorizzati nel file "dati_labirinto.txt". Per modificare la dimensione e i muri del labirinto, è necessario modificare tale file, seguendo la convenzione mostrata di seguito:

N
V
x_1
y_1
...
x_V
y_V

Dove:

- N indica la dimensione del lato del labirinto;
- V indica il numero di celle vuote;
- I valori nelle righe successive rappresentano le coordinate delle V celle vuote.

Per un corretto caricamento dei dati nel programma, assicurarsi che nel file "labirinto.txt" non siano presenti caratteri spazio.

Esempio:

Scrivendo nel file "dati_labirinto.txt" il seguente testo:

4
5
1
1
1
2
2
2
3
2
3
3

Sarà generato il seguente labirinto:



5.2 Posizionare l'Automa e l'Obiettivo

I dati sull'automa sono memorizzati nel file "dati_automa.txt" secondo la seguente convenzione:

x_a
y_a
α
β

Dove:

- x_a, y_a sono le coordinate della posizione iniziale dell'automa;
- α, β sono i parametri di decisione descritti in precedenza.

I dati sull'obiettivo sono memorizzati nel file "dati_obiettivo.txt" secondo la seguente convenzione:

x_o
y_o
P_{max}

Dove:

- x_o, y_o sono le coordinate della posizione dell'obiettivo;
- P_{max} è la potenza massima del segnale irradiato dall'obiettivo.

Tutti i dati vengono caricati automaticamente nel programma non appena il file .jar viene eseguito. Per avviare la simulazione è sufficiente premere il pulsante "Start" della finestra "Control".

Bibliografia

[1] S.Russel , P.Norvig - Artificial Intelligence: A Modern Approach

[2] C. De Sio Cesari - Manuale di Java 7

[3] Grafica 2D e 3D in Java, <https://liceocuneo.it/grafica/animazione/>

[4] JogAmp, <https://jogamp.org/>