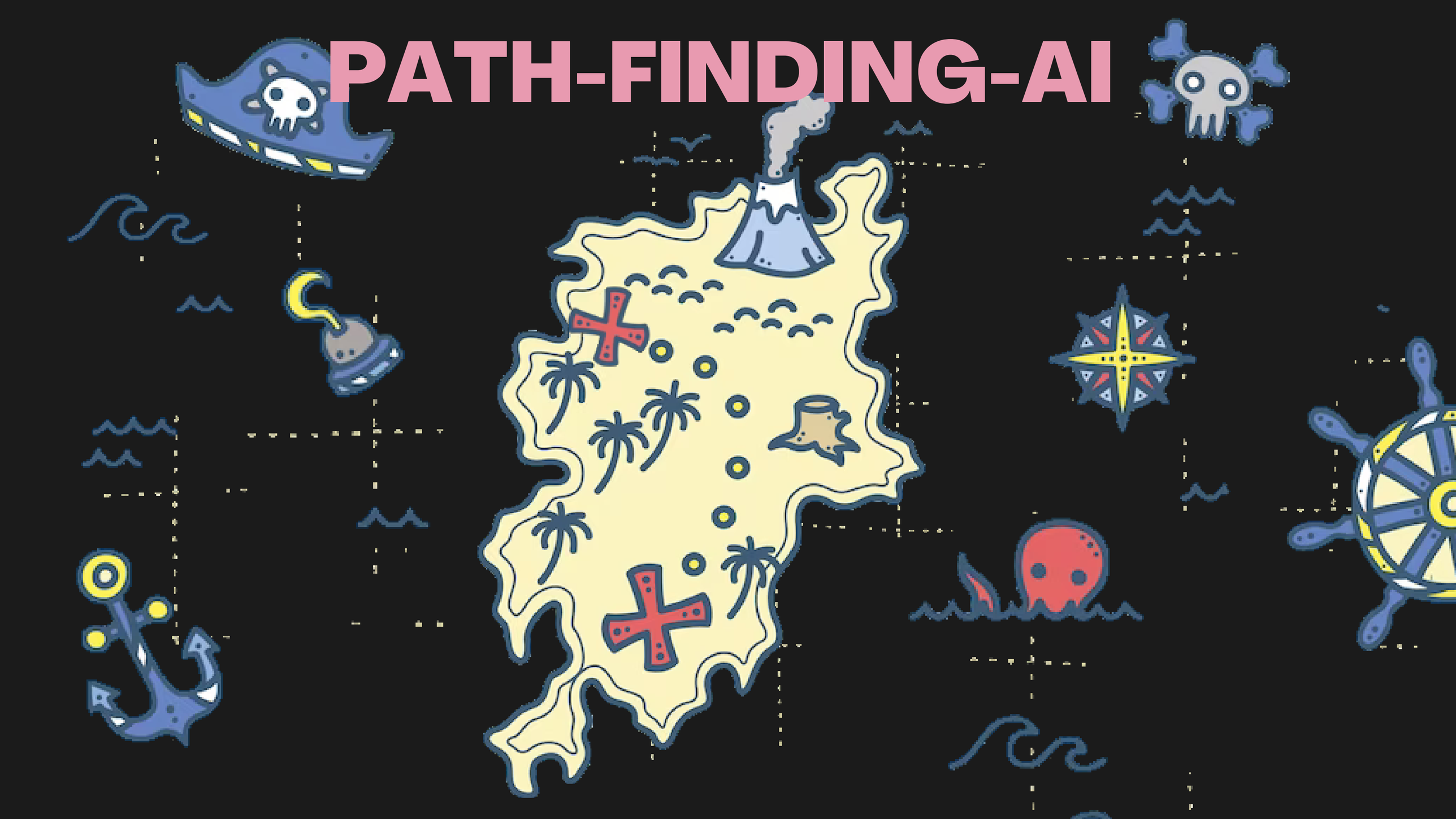
PATH-FINDING-AI



COSA È PATH-FINDING-AI?

Path-finding-AI nasce con l'idea di realizzare un avversario capace di percepire il player e inseguirlo tramite l'impiego di algoritmi per individuare il percorso ottimale.

Il classico nemico che ci segue per la mappa?

Non proprio in realtà.

All'interno del seguente progetto non ci siamo solo concentrati sul definire un algoritmo capace di raggiungere il player attraverso il percorso ottimale, anche...ma non solo.

OBBIETTIVI DEL PROGETTO

01 Efficace

In molti casi, queste AI non sono altro che un algoritmo di ricerca informata che trackano il player e sono troppo facili da fregare.

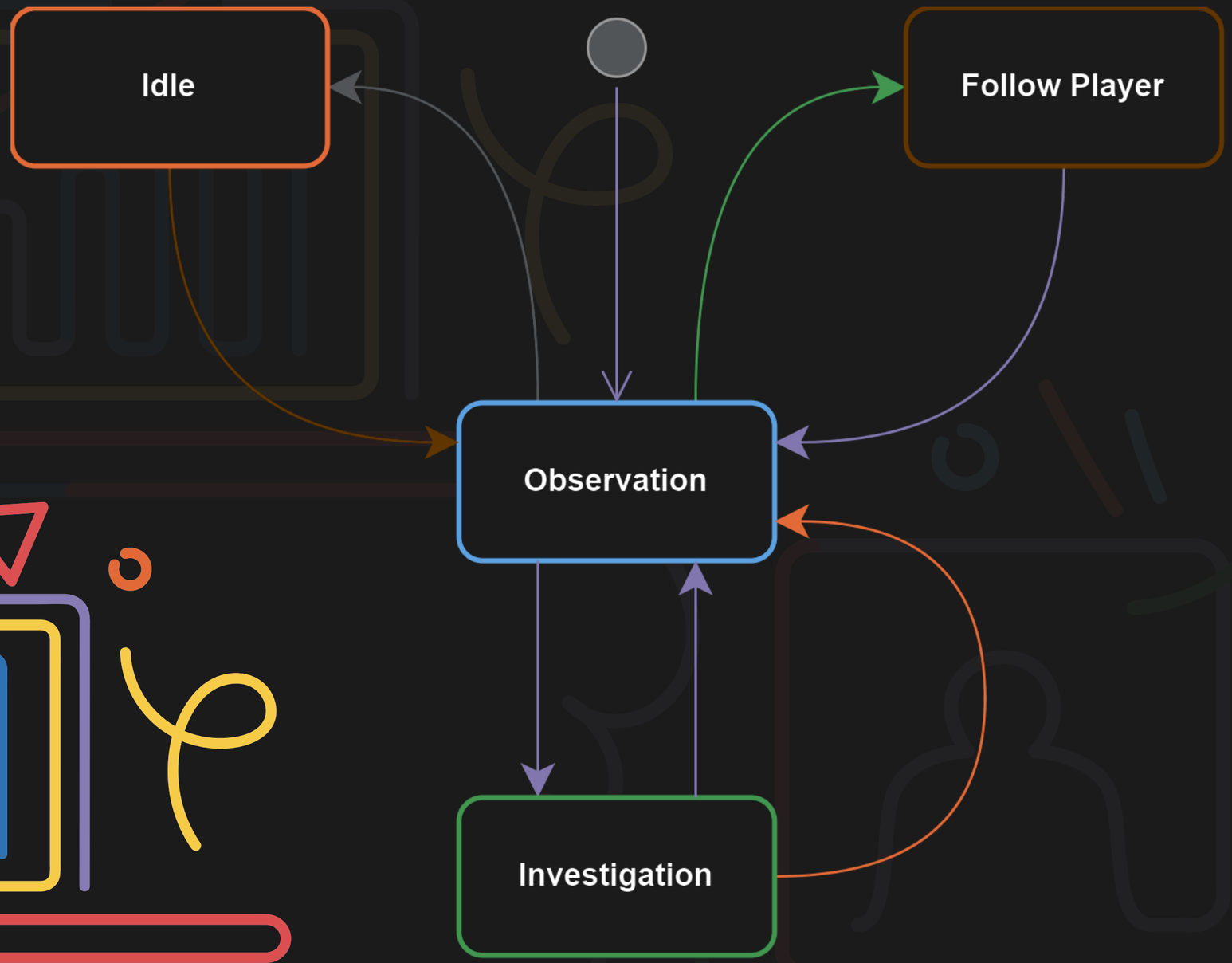
02 Efficace ma efficiente

Moltissime delle risorse online e molti esempi di progetti simili non potevano essere implicati in un gioco. Realizzare parte di un gioco, significa dover effettuare le computazioni nei tempi imposti da un gioco.

03 Efficace si, ma plausibile

In alcuni progetti e anche in molti giochi ho visto comportamenti che per quanto potessero essere efficaci erano veramente poco intuitivi. Ci siamo occupati anche di questo.

EFFICACIA



EFFICIENZA

RICERCA INFORMATA

RICERCA NON INFORMATA

EFFICIENZA

RICERCA INFORMATA

Esplorare il meno possibile

Guardando all'efficienza ci siamo resi conto che questa sarebbe stata totalmente dominata dal numero di nodi che avremmo dovuto esplorare per arrivare a una soluzione.

Il costo da pagare per riuscire a esplorare i nodi in maniera organizzata può essere esiguo rispetto ai vantaggi ottenuti.

EFFICIENZA

Quale algoritmo?

E’ necessario ora selezionare un algoritmo tra quelli di ricerca informata.

A*

Tra gli algoritmi di ricerca informata abbiamo infine optato per l’A*.

Tra tutti quanti gli altri algoritmi è da subito sembrato essere quello che si adattava meglio alle nostre esigenze.

Vediamo però per quale motivo abbiamo scartato gli altri algoritmi.

EFFICIENZA

Quale algoritmo?

Best-First-Greedy

Non ci garantiva nè ottimalità nè complettezza e noi vogliamo individuare il percorso ottimale. Anche se poteva risultare più efficiente, non sposa tutti i nostri design goals, scartato!

IDA*

Il suo principale vantaggio è quello risparmiare spazio di memoria, ma ha bisogno di riesplorare nodi già esplorati.
Noi non abbiamo grandi problemi di spazio di memoria, scartato!

Best-First-Recursive

Vogliamo efficienza, e la seguente tecnica utilizza un approccio ricorsivo. Troppe modifiche allo stack della memoria, scartato!

EFFICIENZA

EURISTICHE

- Distanza Euclidea
- Distanza Manhattan
- Distanza Chebyshev
- Distanza Octile
- Distanza Hamming

EFFICIENZA

EURISTICHE

- **Distanza Euclidea**
- **Distanza Manhattan**
- **Distanza Chebyshev**
- ~~**Distanza Octile**~~
- ~~**Distanza Hamming**~~

Octile è stata scartata perchè è molto utile quando il movimento a griglia supporta il movimento in diagonale, ma non è il caso del nostro gioco.

Hamming invece risultava essere difficilmente applicabile per il problema che stavamo tentando di risolvere. Non avevamo un percorso con il quale misurare la distanza tra i due

EFFICIENZA

LA MIGLIORE: MANHATTAN

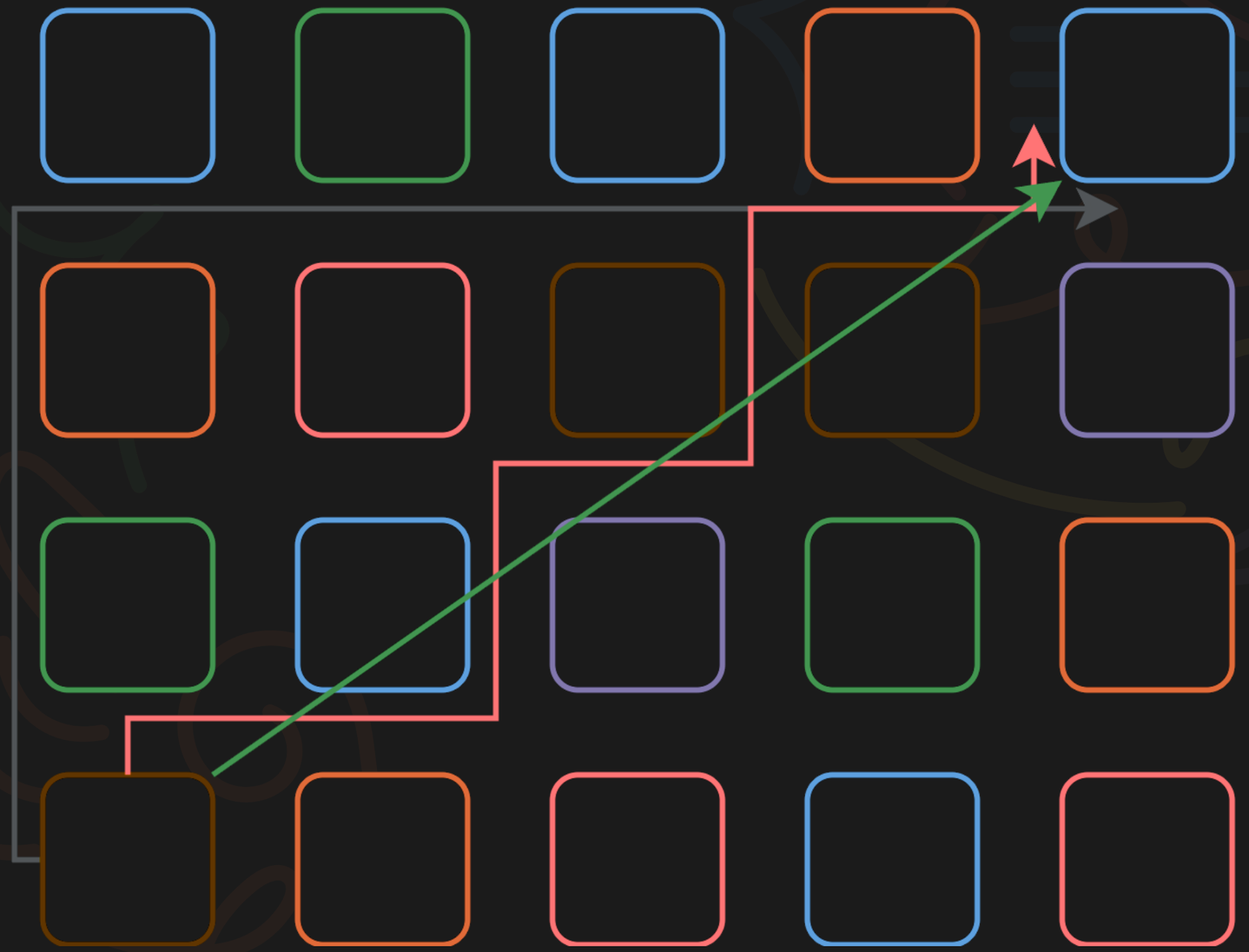
- **Distanza Euclidea**
- **Distanza Manhattan**
- **Distanza Chebyshev**

Euristica	Elapsed Time	Explored Nodes
Manhattan	317µs	741
Euclidean	1138µs	1546
Chebyshev	1088µs	1706

A partire dai test che abbiamo effettuato, possiamo concludere che la migliore è manhattan.

Ma perchè?

GEOMETRIA DEL TAXI



UN AI PLAUSIBILE

Meglio evitare comportamenti ridicoli...

In moltissimi tentativi che ho visto, non si ragiona minimamente sull'esperienza finale dell'utente.
L'enemy percepisce il player a qualunque distanza.

Addirittura l'enemy percepisce l'utente anche attraverso i muri!

E' meglio evitare questo genere di sciochezze. Un giocatore deve riuscire ad intuire quali sono le possibili reazioni di un nemico.

Ed essere percepiti attraverso i muri,
NON E' INTUIBILE!

UN AI PLAUSIBILE

SI MA COME?

Line of sight!

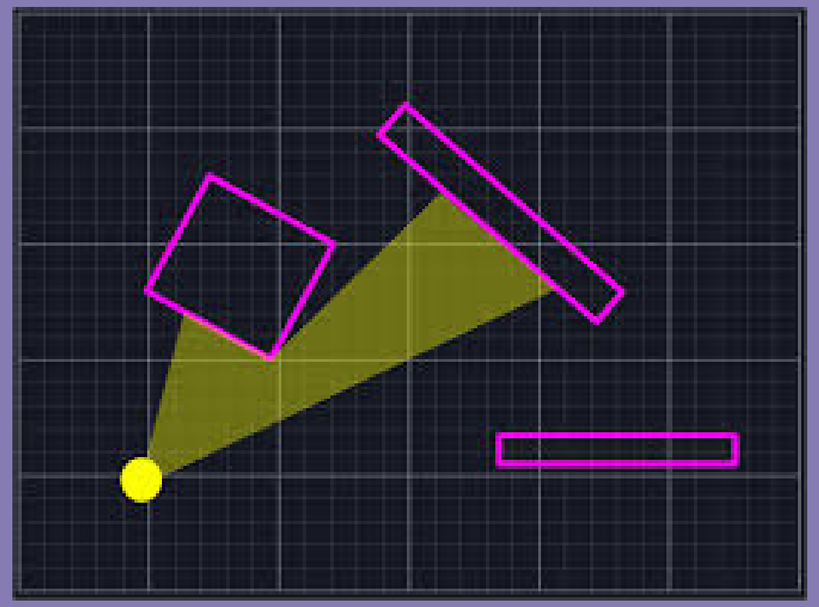
Line of sight è una tecnica che viene implicata all'interno di videogiochi per implementare il concetto di field of view.

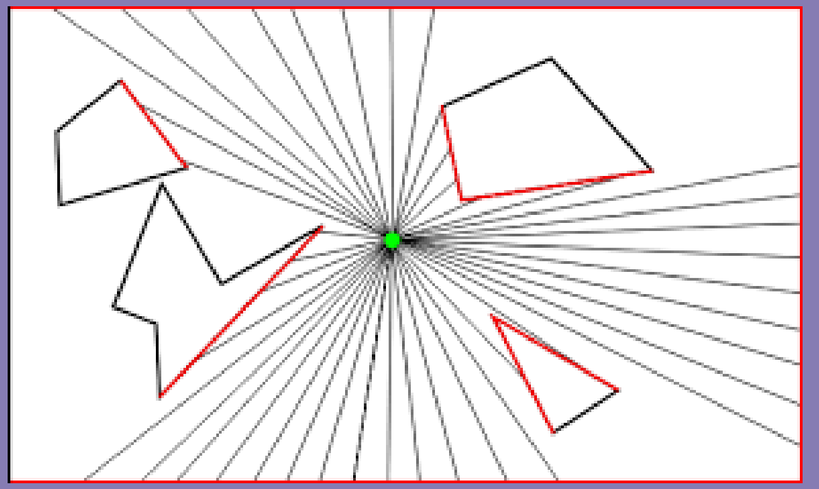
Quindi il nemico potrà rilevarci solo se ci troviamo in una certa area a lui adiacente e sopratutto se non vi è nessun ostacolo nel mezzo che ci rende invisibili.

Un solo problema...il Line of Sight non è pensato per i tilegame

Parte del nostro lavoro, è stato anche quello di adattare questa tecnica a un tilegame con movimento a griglia.

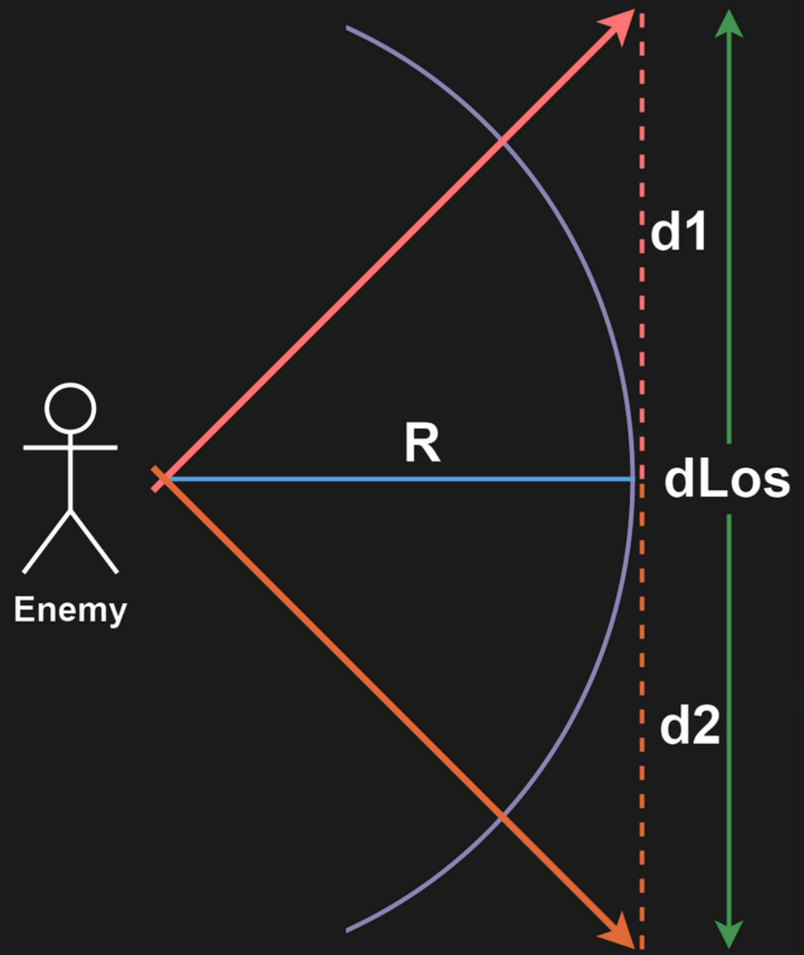
What:



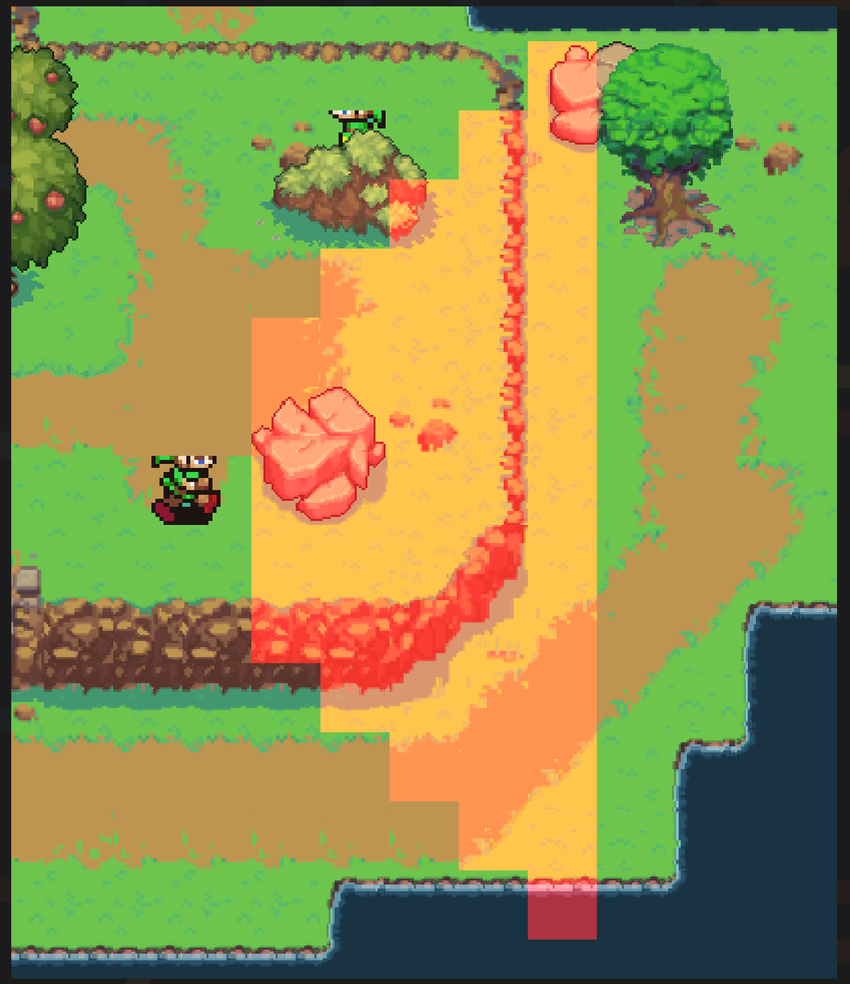


LINE OF SIGHT

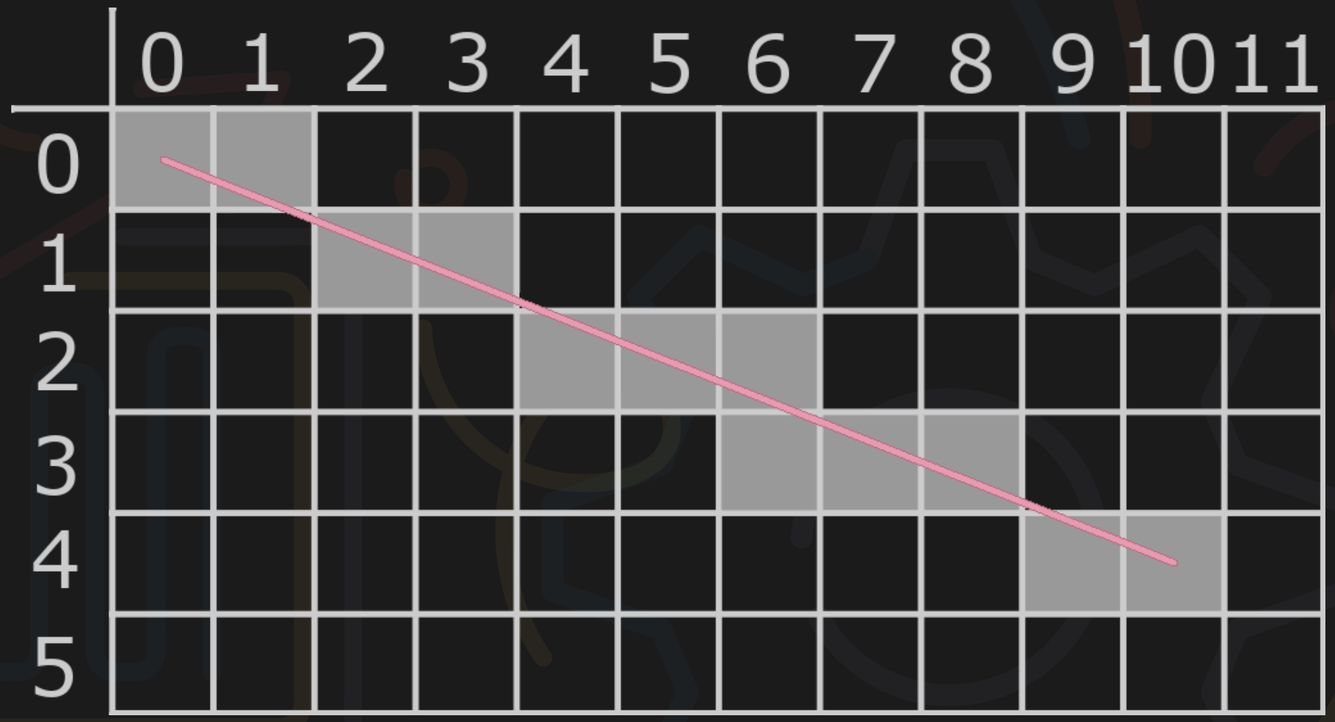
CONCETTO



IMPLEMENTAZIONE

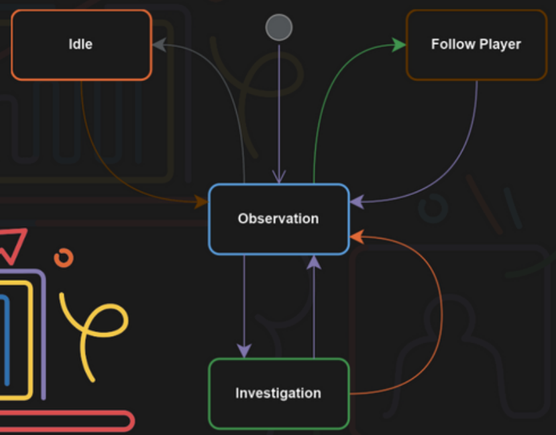


SOLUZIONE? DDA!



Q&A

EFFICACIA



EFFICIENZA

RICERCA INFORMATA

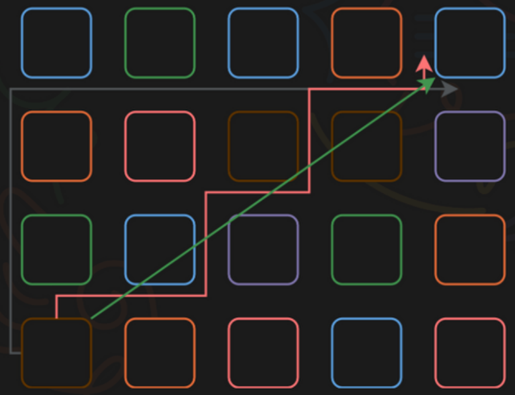
~~RICERCA NON INFORMATA~~

Esplorare il meno possibile

Guardando all'efficienza ci siamo resi conto che questa sarebbe stata totalmente dominata dal numero di nodi che avremmo dovuto esplorare per arrivare a una soluzione.

Il costo da pagare per riuscire a esplorare i nodi in maniera organizzata può essere esiguo rispetto ai vantaggi ottenuti.

GEOMETRIA DEL TAXI



LINE OF SIGHT