



Royale AI

Documentazione progetto per l'esame di Intelligenza Artificiale

Creata da:

Simone Stella, matr. 209392

Gabriele Orlando, matr. 213532

Introduzione

Royale AI è un progetto universitario che prevede l'implementazione di una intelligenza artificiale che simula le strategie di attacco e di difesa sul campo di **Clash Royale**, attraverso, delle regole e dei vincoli per definire la migliore strategia da applicare.

Sviluppo

Il primo passo per lo sviluppo è stato quello dell'implementazione di un algoritmo che fosse in grado di **riconoscere** le carte in campo e la situazione in un momento specifico della partita.

Questo attraverso delle immagini che sono state estrapolate direttamente dal campo di gioco, in modo da istruire l'intelligenza artificiale affinché sia in grado di attribuire dei parametri alle carte riconosciute.

Successivamente c'è stata l'implementazione delle **regole** che stabiliscono quale sia la **carta migliore** da giocare in difesa e quale sia l'attacco migliore da effettuare in un determinato momento.

Infine, una volta ottenute tutte le informazioni quali, l'elisir disponibile, la carta nemica in campo, le carte in mano e la posizione di tutti questi elementi effettua dei click su schermo in modo da giocare in **modo autonomo** le carte.

Implementazione Datalog

In Python, le regole Datalog prevedono l'utilizzo di una libreria chiamata **pyDatalog**, che permette di effettuare la conversione.

Ad ogni modo, prima di implementarle sono state definite tutte le regole e i vincoli che l'intelligenza artificiale deve rispettare:

Fatti

carta(Nome, Costo In Mana, Tipologia (1 = unità, 2 = magia), Livello di Forza, Tipo Target(1 = edifici, 2 = unità di terra, 3 = unità d'aria, 4 = tutte le tipologie))

```
carta('Knight', 3, 1, 5, 2).
carta('Minion', 3, 1, 2, 4).
carta('Archers', 3, 1, 3, 4).
carta('Arrows', 3, 2, 4, 4).
carta('Fireball', 4, 2, 6, 4).
carta('Giant', 5, 1, 7, 1).
carta('Pekka', 4, 1, 7, 2).
carta('Musketeer', 4, 1, 6, 4).
```

```
manaDisponibile(8).
```

```
cartaInMano(Gigante).
cartaInMano(Arceri).
cartaInMano(Palla Di Fuoco).
cartaInMano(Sgherri).
```

```
cartaInCampo(Pekka).
```

Regole

La regola stabilisce che la carta da giocare deve avere il livello di forza maggiore o uguale del livello di forza della carta in campo, ci dev'essere mana per giocarla, il target della carta da giocare NON devono essere gli edifici, ed il divario di forza non dev'essere troppo alto

```
cartaDaGiocareInDifesa(X) :- cartaInMano(X), cartaInCampo(X1), carta(X,Z,K,Y,T),
carta(X1,Z1,K1,Y1,T1), (Y >= Y1), manaDisponibile(Z2), (Z2 >= Z), (T > 1), ((Y-Y1) <
3).
```

La regola viene applicata se la precedente non ha effetto, quindi gioca una qualsiasi carta, purché sia un'unità ed il target NON siano gli edifici.

cartaDaGiocareInDifesaAlternativa(X) :- cartainMano(X), carta(X,Z,K,Y,T),
manaDisponibile(Z2), (Z2 >= Z), (K == 1), (T > 1).

In attacco deve schierare una qualsiasi unità e preferire le unità con target edifici.

attacco(X) :- cartainMano(X), cartainMano(X1), carta(X,Z,K,Y,T),
carta(X1,Z1,K1,Y1,T1), manaDisponibile(Z2), (Z2 >= Z), (K == 1)

:~ attacco(X), carta(X,Z,K,Y,T) [T@1, T]

OpenCV

OpenCV è una libreria Open Source che permette di riconoscere delle immagini in una scena.

In particolare all'interno del codice sono state inserite le immagini di riferimento, da cercare all'interno dell'acquisizione dello schermo:

```
roi = ref_image[position[1]:position[1]+size[1], position[0]:position[0]+size[0]]  
res = cv2.matchTemplate(roi, template, cv2.TM_CCOEFF_NORMED)  
_, max_val, _, max_loc = cv2.minMaxLoc(res)  
  
if max_val > 0.8:
```

Questo pezzo di codice verifica la presenza di una carta in una determinata posizione. Viene verificata la presenza, solo se il risultato di match, max_val, ha un valore superiore a 0.8, un indicatore che specifica quanto le due immagini si somigliano.

Istruzioni

Per eseguire l'intelligenza artificiale è sufficiente collegare il dispositivo mobile al computer in modo da poter effettuare lo screen mirroring.

Sul dispositivo mobile bisogna avviare il gioco Clash Royale e successivamente bisogna avviare una partita.

Una volta che la partita è iniziata, bisogna eseguire il codice python che si occuperà di avviare il programma per lo screen mirroring.