

# Progetto Clips

---

**Giulio Taralli & Ismaila Toure**

# Descrizione iniziale del gioco

Il progetto richiedeva di implementare degli agenti in grado di risolvere la versione solitaria del gioco battaglia navale, per poi confrontarli.

Si è deciso di implementare due agenti:

- Primo agente: bravo ma non si applica
- Secondo agente: miglioramento dei difetti del primo

In seguito descriviamo i due agenti, i loro moduli principali, i loro comportamenti e la caratteristiche.

# Agente 1: “Intelligente ma non si applica”

Il primo agente è la versione iniziale più semplice sviluppata.

È caratterizzato dai seguenti moduli:

- **Guess Module:** gestisce i casi dove applicare le guess, sia sulle celle note, sia su quelle dedotte
- **Fire Validation Module:** richiama il modulo Fire o esegue una fire sulla cella con il più alto valore k-per-row e k-per-col
- **Fire Module:** contiene tutti i casi di fire se sono presenti navi maggiori di 3 o 4 celle

# Guess Module

Primo modulo che verrà richiamato dall'agente quando non saprà più cosa fare.

In questo modulo vengono eseguite le seguenti azioni:

- Si eseguono delle guess sulle celle note, asserendo acqua attorno alla cella in base alla tipologia di cella nota appena guessata
- Si eseguono delle guess nelle celle vicino alle celle note, asserendo acqua nelle diagonali dalla guessed

# Fire Validation Module

Questo modulo verrà richiamato dall'agent quando non saprà più fare delle guess.

A seconda del contesto eseguirà le seguenti azioni:

- Se la corazzata (4 caselle) o gli incrociatori (3 caselle) non sono ancora state trovate tutte, allora verrà richiamato il Fire Module
- Altrimenti si cade nella regola di fallback e si esegue una fire sulla cella dove i valori k-per-row e k-per-col sono maggiori assunta come la cella con maggiore probabilità di contenere un pezzo di nave

# Fire Module

Richiamato solo dal modulo Fire Val, quindi se siamo qui è perché non sono già state trovate tutte le navi da 3 e da 4 celle.

In questo caso, a seconda della tipologia di cella nota, si esegue una fire due celle lontano da quella nota.

Questo per identificare immediatamente la corazzata e gli incrociatori.

Tale agente sfrutta in maniera limitata la conoscenza iniziale del gioco:

- Tiene conto delle navi trovate e da trovare ancora
- Usa i valori  $k$ -per-row e  $k$ -per-col ma non li modifica
- Riconoscimento iniziale delle righe/colonne dove  $k$ -per-row/ $k$ -per-col è zero: in questo caso asserisce acqua su tutta la riga/colonna
- Implementazione di una regola di fallback per le fire, ma non per le guess

L'agente implementato quindi si comporta in modo **conservativo**, usa tutte le fire a disposizione ma esegue delle guess **solo quando è sicuro** che lì è presente una possibile nave.

- Atteggiamento più **cauto** che intraprendente

## Agente 2: versione migliorata

Come già accennato, la seconda versione dell'agente è un miglioramento della prima versione, esso infatti sfrutta appieno la conoscenza iniziale della configurazione di gioco e la sfrutta anche durante la partita andando a modificare determinati dati.

La suddivisione è uguale a quella del primo agente, ma le regole sono state modificate ed espanse:

- **Guess Module**
- **Fire Validation Module**
- **Fire Module**

Verranno quindi descritti i cambiamenti o le aggiunte dei rispettivi moduli.



# Guess Module

Come prima questo modulo esegue le guess sulle celle, ma rispetto alla prima versione estende diverse regole aggiuntive:

- Se in una riga/colonna rimane solo una cella segnata come nave, ed è rimasta solo una cella non ancora segnata come water, k-cell, guess (quindi è sconosciuta), allora fai guess su quella cella
- Se si sono esaurite le fire, allora si esegue una guess sulle celle più probabili basate sul numero maggiore di k-per-row e k-per-col

# Fire Validation Module

In aggiunta alle regole precedenti si sono implementate altre regole più fini che potrebbero aiutare nella risoluzione del gioco:

- Se in una riga/colonna rimane solo una nave, e rimangono solo due celle sconosciute allora si fa una fire su una delle due celle
  - In caso la cella fosse water, il Guess Module farà una guess sulla cella rimanente
  - In caso la cella fosse una nave, il Guess Module farà la guess sulla cella nota per poi fare delle guess successive

Con questo comportamento aggiuntivo si gestiscono casi che l'agente 1 non era capace di tenere in considerazione, e con alta probabilità di avanzare nella risoluzione del gioco.

# Fire Module

Come prima, questo modulo verrà richiamato se sono ancora presenti delle navi da 4 celle e da 3 celle.

In questo caso verrà eseguita una fire due celle distanti dalla cella nota, a seconda della tipologia.

Non è cambiato rispetto all'agente 1.

Il secondo agente sfrutta appieno la conoscenza iniziale del gioco:

- Tiene conto delle navi trovate e da trovare ancora
- Usa i valori  $k$ -per-row e  $k$ -per-col e li modifica quando applica delle guess
- Riconoscimento delle righe/colonne dove  $k$ -per-row/ $k$ -per-col è zero per tutta la durata del gioco: in questo caso asserisce acqua sulle celle sconosciute
- Regole di fire e di guess più specifiche
- Implementazione di una regola di fallback sia per le fire, sia per le guess

Rimane l'approccio **conservativo** inizialmente, applica delle regole più interessanti successivamente che potrebbero aiutare nella risoluzione del gioco. Ma quando non sa più che cosa fare **osa di più** applicando le guess in zone sconosciute ma più probabili.

- Atteggiamento **cauto all'inizio, più intelligente a metà e intraprendente a fine partita**

# Configurazioni di test

Osserviamo come si comportano i due agenti su 5 diverse configurazioni di mappe.

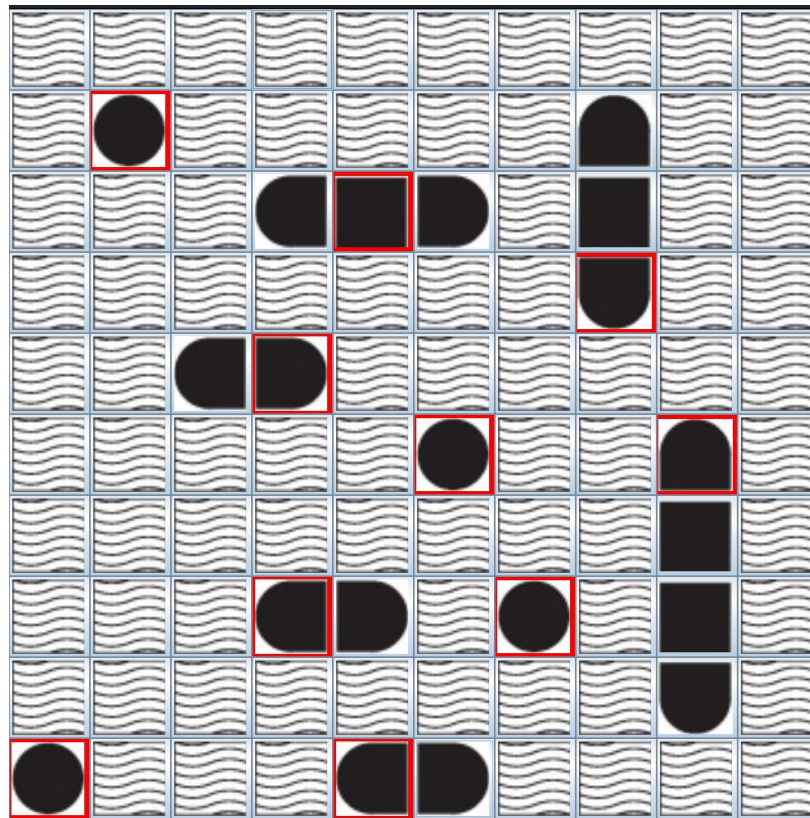
Una volta descritte le mappe confrontiamo i due agenti e cerchiamo di capire le differenze tra i due.

Inizieremo con le configurazioni più semplici, man man aumenteremo la difficoltà riducendo le celle note disponibili a inizio partita.

# Map environment: 1\_info\_every\_boat

Come prima configurazione, la più semplice, abbiamo assegnato una cella nota per ogni nave. Questo significa che i sottomarini sono già noti a inizio partita.

Ci aspettiamo un buon comportamento da entrambi gli agenti. Siccome tutte e due dovrebbero gestire bene le scelte quando hanno conoscenza delle posizioni a grandi linee di tutte le navi.



Infatti risolvono la configurazione senza problemi, come si può vedere dai risultati.

Agente 1:

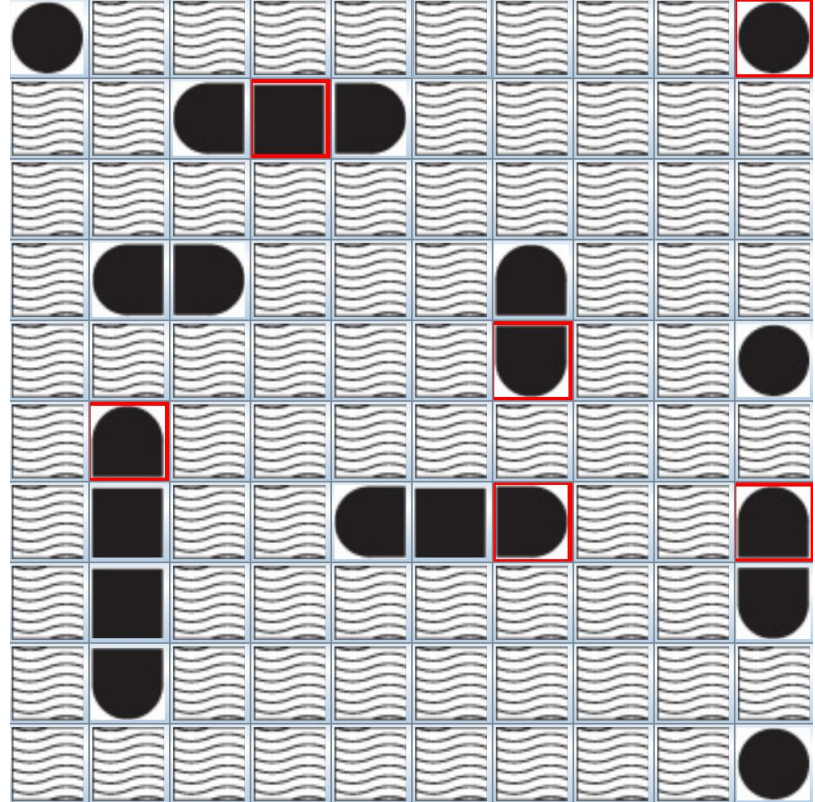
```
Fires left: 1    Guess left: 1
ENV --- [ 1 ] [ 7 ]
ENV --- [ 1 ] [ 1 ]
ENV --- [ 5 ] [ 8 ]
ENV --- [ 3 ] [ 7 ]
ENV --- [ 4 ] [ 3 ]
ENV --- [ 7 ] [ 3 ]
ENV --- [ 9 ] [ 4 ]
ENV --- [ 5 ] [ 5 ]
ENV --- [ 7 ] [ 6 ]
ENV --- [ 9 ] [ 0 ]
ENV --- [ 2 ] [ 4 ]
ENV --- [ 2 ] [ 3 ]
ENV --- [ 2 ] [ 5 ]
ENV --- [ 6 ] [ 8 ]
ENV --- [ 2 ] [ 7 ]
ENV --- [ 4 ] [ 2 ]
ENV --- [ 7 ] [ 4 ]
ENV --- [ 9 ] [ 5 ]
ENV --- [ 8 ] [ 8 ]
Your score is 445
```

Agente 2:

```
Fires left: 1    Guess left: 1
ENV --- [ 1 ] [ 7 ]
ENV --- [ 1 ] [ 1 ]
ENV --- [ 5 ] [ 8 ]
ENV --- [ 3 ] [ 7 ]
ENV --- [ 7 ] [ 3 ]
ENV --- [ 9 ] [ 4 ]
ENV --- [ 5 ] [ 5 ]
ENV --- [ 7 ] [ 6 ]
ENV --- [ 9 ] [ 0 ]
ENV --- [ 4 ] [ 3 ]
ENV --- [ 2 ] [ 4 ]
ENV --- [ 2 ] [ 3 ]
ENV --- [ 2 ] [ 5 ]
ENV --- [ 6 ] [ 8 ]
ENV --- [ 2 ] [ 7 ]
ENV --- [ 7 ] [ 4 ]
ENV --- [ 9 ] [ 5 ]
ENV --- [ 4 ] [ 2 ]
ENV --- [ 8 ] [ 8 ]
Your score is 445
```

## Map environment: 6\_info\_boat

In questa configurazione riduciamo il numero di celle conosciute inizialmente, restano però solo informazioni contenenti navi.





Già con questa configurazione è possibile notare un distacco nel punteggio tra i due agenti:

Agente 1:

```
Fires left: 2    Guess left: 6
ENV --- [ 6 ] [ 4 ]
ENV --- [ 0 ] [ 9 ]
ENV --- [ 5 ] [ 1 ]
ENV --- [ 6 ] [ 9 ]
ENV --- [ 4 ] [ 6 ]
ENV --- [ 6 ] [ 6 ]
ENV --- [ 1 ] [ 3 ]
ENV --- [ 1 ] [ 2 ]
ENV --- [ 1 ] [ 4 ]
ENV --- [ 6 ] [ 1 ]
ENV --- [ 7 ] [ 9 ]
ENV --- [ 3 ] [ 6 ]
ENV --- [ 6 ] [ 5 ]
ENV --- [ 8 ] [ 1 ]
Your score is 120
```

Agente 2:

```
Fires left: 0    Guess left: 0
ENV --- [ 0 ] [ 9 ]
ENV --- [ 5 ] [ 1 ]
ENV --- [ 6 ] [ 9 ]
ENV --- [ 4 ] [ 6 ]
ENV --- [ 6 ] [ 6 ]
ENV --- [ 1 ] [ 3 ]
ENV --- [ 6 ] [ 1 ]
ENV --- [ 7 ] [ 9 ]
ENV --- [ 3 ] [ 6 ]
ENV --- [ 6 ] [ 5 ]
ENV --- [ 8 ] [ 1 ]
ENV --- [ 6 ] [ 4 ]
ENV --- [ 1 ] [ 4 ]
ENV --- [ 1 ] [ 2 ]
ENV --- [ 3 ] [ 1 ]
Your score is 265
```



In questa configurazione notiamo un calo per entrambi gli agenti nel punteggio, leggero per l'agente 2 e molto pesante per l'agente 1.

Agente 1:

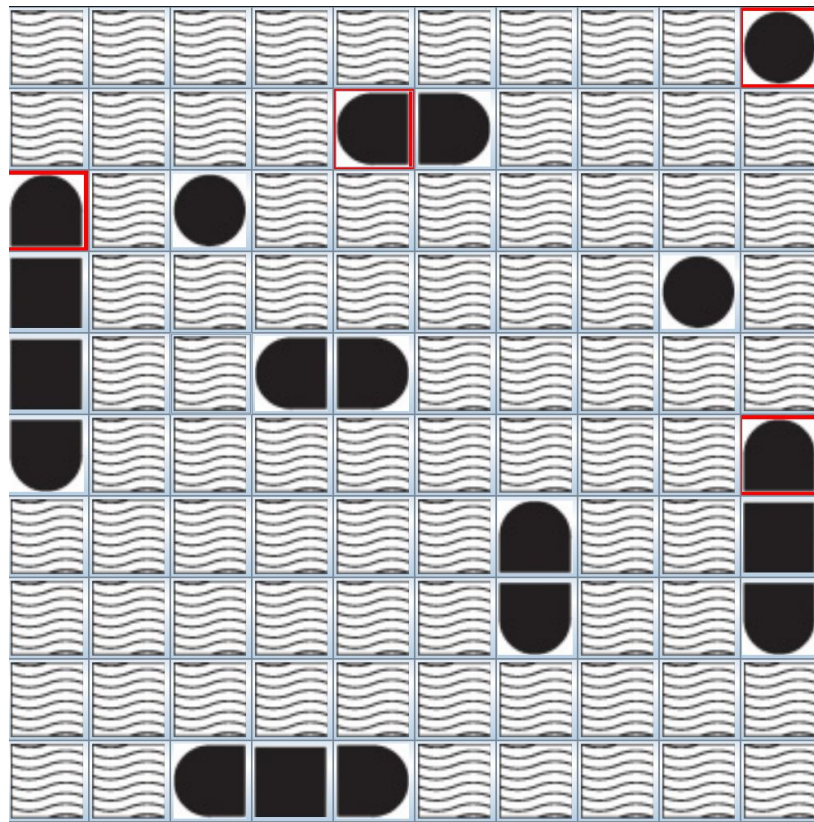
```
Fires left: 2    Guess left: 11
ENV --- [ 0 ] [ 9 ]
ENV --- [ 4 ] [ 0 ]
ENV --- [ 0 ] [ 0 ]
ENV --- [ 8 ] [ 3 ]
ENV --- [ 0 ] [ 5 ]
ENV --- [ 0 ] [ 7 ]
ENV --- [ 0 ] [ 1 ]
ENV --- [ 8 ] [ 4 ]
ENV --- [ 0 ] [ 3 ]
Your score is -115
```

Agente 2:

```
Fires left: 0    Guess left: 0
ENV --- [ 4 ] [ 0 ]
ENV --- [ 0 ] [ 0 ]
ENV --- [ 8 ] [ 3 ]
ENV --- [ 0 ] [ 1 ]
ENV --- [ 8 ] [ 4 ]
ENV --- [ 0 ] [ 3 ]
ENV --- [ 0 ] [ 9 ]
ENV --- [ 9 ] [ 9 ]
ENV --- [ 8 ] [ 9 ]
ENV --- [ 0 ] [ 2 ]
ENV --- [ 0 ] [ 5 ]
ENV --- [ 0 ] [ 6 ]
ENV --- [ 0 ] [ 7 ]
ENV --- [ 4 ] [ 6 ]
Your score is 220
```

## Map environment: 4\_info\_boat

Come penultima configurazione ne abbiamo creato una con solo 4 celle note, tutte e 4 però sono celle di navi.



La seguente configurazione produce i seguenti risultati per i due agenti:

Agente 1:

```
Fires left: 0    Guess left: 10
ENV --- [ 7 ] [ 9 ]
ENV --- [ 0 ] [ 9 ]
ENV --- [ 2 ] [ 0 ]
ENV --- [ 5 ] [ 9 ]
ENV --- [ 1 ] [ 4 ]
ENV --- [ 3 ] [ 0 ]
ENV --- [ 6 ] [ 9 ]
ENV --- [ 1 ] [ 5 ]
ENV --- [ 4 ] [ 0 ]
ENV --- [ 5 ] [ 0 ]
Your score is -70
```

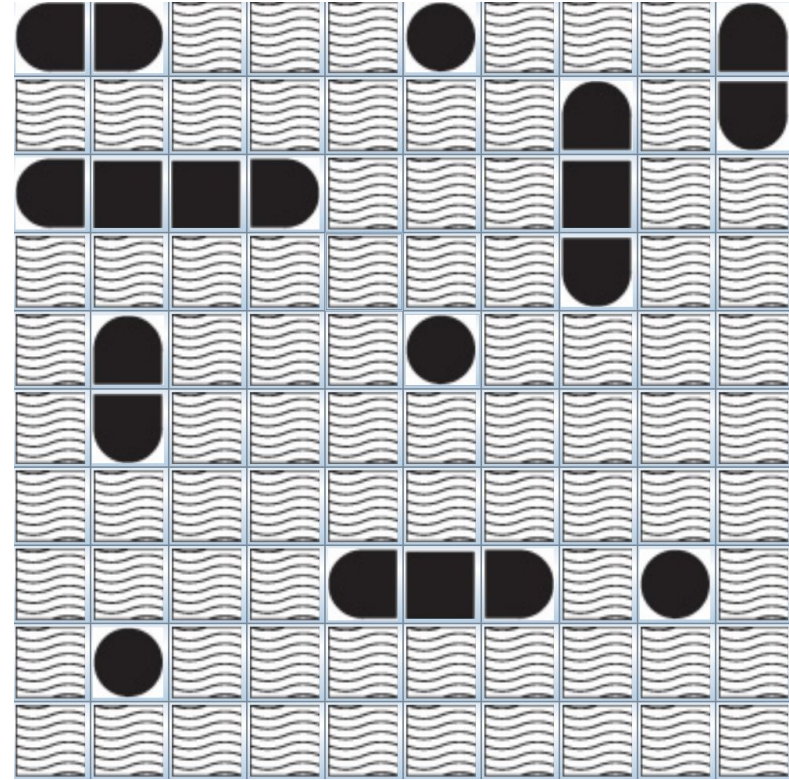
Agente 2:

```
Fires left: 0    Guess left: 0
ENV --- [ 7 ] [ 1 ]
ENV --- [ 0 ] [ 9 ]
ENV --- [ 2 ] [ 0 ]
ENV --- [ 5 ] [ 9 ]
ENV --- [ 1 ] [ 4 ]
ENV --- [ 3 ] [ 0 ]
ENV --- [ 6 ] [ 9 ]
ENV --- [ 1 ] [ 5 ]
ENV --- [ 4 ] [ 0 ]
ENV --- [ 5 ] [ 0 ]
ENV --- [ 7 ] [ 9 ]
ENV --- [ 9 ] [ 4 ]
ENV --- [ 9 ] [ 3 ]
ENV --- [ 9 ] [ 2 ]
Your score is 220
```

## Map environment: no\_obs

Abbiamo provato ad eseguire i due agenti con una delle configurazioni più difficili, ovvero quella senza nessuna informazione iniziale sulle celle.

Osserviamo che punteggi danno e come si comportano i due agenti nel caso estremo.





Come ci aspettavamo, l'agente 1 si comporta molto male, mentre il secondo riesce a ottenere un punteggio almeno positivo.

Agente 1:

```
Fires left: 4    Guess left: 19  
ENV --- [ 2 ] [ 1 ]  
Your score is -635
```

Agente 2:

```
Fires left: 0    Guess left: 0  
ENV --- [ 2 ] [ 1 ]  
ENV --- [ 0 ] [ 1 ]  
ENV --- [ 2 ] [ 0 ]  
ENV --- [ 2 ] [ 2 ]  
ENV --- [ 0 ] [ 0 ]  
ENV --- [ 7 ] [ 5 ]  
ENV --- [ 0 ] [ 5 ]  
ENV --- [ 0 ] [ 9 ]  
ENV --- [ 2 ] [ 3 ]  
ENV --- [ 2 ] [ 7 ]  
Your score is 10
```

# Risultati

Come ci aspettavamo, l'agente 2 si comporta meglio rispetto al primo, sia spiegando le regole aggiunte per gestire casi più delicati ma impattanti per la risoluzione del gioco, sia se osserviamo i punteggi calcolati attraverso la formula data in consegna.

L'agente 1, nonostante la buona base iniziale, è cauto nel fare guess, e non gestisce i casi particolari che si presentano a metà gioco, producendo risultati peggiori.



# Room for improvement

Possiamo sicuramente affermare che l'agente 2, nonostante sia migliore del primo, è ben lontano dall'essere perfetto. Si potrebbero fare ulteriori miglioramenti.

Dei possibili miglioramenti potrebbero essere i seguenti:

- Se esiste un contorno di celle water, al cui interno sono presenti  $n$  celle, e se la nave composta da  $n$  celle non è stata ancora trovata, allora marca le celle sconosciute contornate da water come guess
- Se abbiamo trovato tutti i 4 sottomarini, e ci sono dei “buchi” di celle sconosciute contornate da water, allora assumi water su quelle celle (questo può valere anche per “buchi” più grandi)

**Grazie per l'attenzione**

---