



Progetto SOAR

AA 2022/2023

Luca Frigato

Environment:

Stanza in cui risiede l'agente, gli oggetti e la finestra

Agente:

- Altezza: 1.5 metri

Oggetti:

- Elastico (1 solo)
- Ramo (1 solo)
- Tronchi (Quantità indefinita, ciascuno alto 1 metro)
- Pietre (Quantità indefinita, utilizzabile durante i lanci con fionda)

Finestra :

- La Finestra rappresenta il GOAL finale
- Si trova a 3.5 metri da terra
- Hp: L'agente può raggiungere la finestra sovrapponendo i tronchi

Ipotesi per la modellazione (1)

- **Finestra**

- L'agente deve rompere la finestra per liberarsi e scappare
- **Assunzione 1** :La finestra può essere colpita al centro o alle estremità
- **Assunzione 2**: È stata modellata una certa resistenza della finestra in base al punto in cui viene colpita
 - La finestra si rompe dopo aver ricevuto 50 danni
 - Colpire al centro fa solo 2 danni (parametrizzato) -> è più resistente
 - Colpire alle estremità fa 15 danni (parametrizzato)
- **Assunzione 3**: l'agente può raggiungere la finestra, spostando i tronchi in sua prossimità
 - Non è stata modellato il concetto di direzione o prossimità
 - **Implicitamente** verranno usate le azioni «**raccogli tronco**»(dall ambiente) o «**sposta tronco**» (verso la finestra)
- **Assunzione 4**: l'altezza del tronco è importante in quanto l'agente deve ammassare tanti tronchi, quanti necessari per raggiungere l'altezza della finestra

Ipotesi per la modellazione (2)

- **Azioni dell'agente**

- L'agente può raccogliere e spostare tronchi, combinare oggetti, attaccare la finestra con una fionda, scappare
- **Assunzione 1:** i tronchi vengono manipolati (raccolti e spostati)
- **Assunzione 2:** elastico, ramo e pietre sono possedute dall'agente. L'agente ha un numero indefinito di pietre
 - Di conseguenza, non c'è necessità di raccogliere pietre dopo averle usate (magari sparandole verso la finestra tramite la fionda)
- **Assunzione 3:** a seguito di ogni init-soar, l'agente parte sempre dallo stato di START e sfrutta le RL imparate nelle iterazioni precedenti.
- **Assunzione 4:** tra le condizioni di terminazione, sono presenti anche quelle dovute alle errare combinazioni di oggetti che possono essere create dall'agente. Questo perché non è possibile scomporre un oggetto combinato. Combo previste e relative «reward»
- (Molla + Rametto di legno) = **+1**
- (Pietre + Rametto di legno) = -1
- (Pietre + Molla) = -1

KB dell'agente

Inizializzazione
della KB

```
sp {apply*inizializza*gioco
  (state <s> ^operator.name inizializza-gioco)
-->
  (<s> ^name escape
    ^previous <p>
    # per gestire lo stato della finestra
    ^danno-estremite-finestra 15
    ^danno-centro-finestra 2
    ^vita-finestra 50
    ^finestra-rotta false

    # utilizzo della fionda
    ^fionda-creata false
    # per la costruzione della scala.
    # Ipotesi -> Servono 2 tronchi siccome ciascuno è di 1 metro, mentre l'agente è di 1.5 metri
    ^altezza-raggiungibile 1.5 # altezza uomo all inizio
    ^altezza-finestra 10.5
    ^altezza-tronco 1
    ^tronco-in-mano false
    ^scalinata-salita false

    #Verifica condizioni di exit
    ^fuggito false
    ^combinato-oggetti false
    ^lanciato-elastico-alla-finestra false
    ^lanciato-rametto-alla-finestra false
  )
  (<p> ^name none)
  (write (crlf) |Inizia il gioco!|)
}
```

Regole di produzione

Possibili Item Craftabili

- **combina-ramoscello-elastico:**
 - Propone di combinare un rametto con una elastico per creare una fionda.
 - Applica la combinazione del rametto e della elastico per **creare una fionda**.
- **combina-ramoscello-roccia:**
 - Propone di combinare un rametto con una roccia (senza ottenere nulla).
 - Applica la combinazione del rametto e della roccia senza ottenere nulla.
 - -> memorizzo però il fatto che sia stata fatta una combinazione, in quanto questa azione prevede l'HALT dell'agente
- **combina-roccia-elastico:** come prima

```
#####  
### Possibili item craftabili ###  
#####  
sp {propose*combina-ramoscello-elastico  
  (state <s> ^name escape ^previous <previous-action> ^fionda-creata false ^finestra-rotta false ^combinato-oggetti false ^tronco-in-mano false)  
  (<previous-action> ^name <> combina-ramoscello-elastico)  
-->  
  (<s> ^operator <o> +)  
  (<o> ^name combina-ramoscello-elastico)  
}  
  
sp {apply*combina-ramoscello-elastico  
  (state <s> ^operator.name combina-ramoscello-elastico ^previous <p>)  
-->  
  (<s> ^previous <p> - <new-previous-action> +)  
  (<new-previous-action> ^name combina-ramoscello-elastico ^reward 1)  
  (<s> ^fionda-creata false - true +)  
  (<s> ^combinato-oggetti false - true +)  
  (write (crlf) |Ha combinato il rametto con l'elastico e creato una fionda!|)  
}
```

```
# halt se si è combinato degli oggetti ma non si è ottenuta una fionda -> non si può scappare  
# ( è un caso più generico dei precedenti )  
sp {halt*wrong-combination  
  (state <s> ^name escape ^fuggito false ^fionda-creata false ^combinato-oggetti true)  
-->  
  (write (crlf) |Ha provato a combinare degli oggetti, ma non ha trovato la combinazione giusta, non può scappare!|)  
  (halt)  
}
```

Regole di produzione

Lanci di oggetti

- **lancia-elastico**
 - Propone di lanciare un elastico alla finestra.
 - -> porta ad una condizione di HALT
- **lancia-ramoscello:**
 - Propone di lanciare un rametto alla finestra.
 - -> porta ad una condizione di HALT

```
#####  
### Lanci di oggetti a caso, senza usare la fionda ###  
#####  
sp {propose*lancia-elastico  
  (state <s> ^name escape ^previous <previous-action> ^fionda-creata false ^finestra-rotta false ^lanciato-pietra-alla-finestra false ^tronco-in-mano false)  
  (<previous-action> ^name <> lancia-elastico)  
-->  
  (<s> ^operator <o> +)  
  (<o> ^name lancia-elastico)  
}  
  
sp {apply*lancia-elastico  
  (state <s> ^operator.name lancia-elastico ^previous <p>)  
-->  
  (<s> ^previous <p> - <new-previous-action> +)  
  (<new-previous-action> ^name lancia-elastico ^reward -1)  
  (<s> ^lanciato-pietra-alla-finestra false - true +)  
  (write (crlf) |Ha lanciato una pietra alla finestra senza romperla.|)  
}
```

Regole di produzione

Spari con la fionda

- **spara-centro-finestra:** L'agente propone questa azione quando la fionda è stata creata, la finestra non è ancora stata rotta e la vita della finestra è maggiore di zero. Tuttavia, l'effetto di questa azione è che la finestra subisce **danni minimi**. **L'azione non viene premiata.**
- **spara-estremità-finestra:** Questa azione viene proposta quando la fionda è **pronta**, la finestra **non è ancora rotta**, la vita della finestra è maggiore di zero **e l'agente non tiene un tronco in mano**.
 - Sparare alle estremità della finestra causa danni significativi e viene premiato con un **punteggio positivo**.

```
sp {apply*spara-estremità-finestra
  (state <s> ^operator.name spara-estremità-finestra ^previous <p> ^vita-finestra <vita-finestra> ^danno-estremità-finestra <danno-estremità-finestra>)
-->
  (<s> ^previous <p> - <new-previous-action> +)
  (<new-previous-action> ^name spara-estremità-finestra ^reward 1)
  (<s> ^vita-finestra <vita-finestra> - ( - <vita-finestra> <danno-estremità-finestra>) +)
  (write (crlf) |Ha sparato sulle estremità della finestra e la finestra si è danneggiata parecchio (- | <danno-estremità-finestra> | )! Vita finestra | ( -
  <vita-finestra> <danno-estremità-finestra>)|)
}
```

^vita-finestra, ^danno-estremità-finestra

Regole di produzione

Gestione dei tronchi

- **raccogli-tronco::** L'agente propone questa azione quando non ha ancora un tronco in mano, non ha ancora raggiunto la scalinata (altezza-raggiungibile è minore dell'altezza necessaria per raggiungere la finestra), e l'azione precedente non era raccogli-tronco. L'effetto di questa azione è che l'agente raccoglie un tronco **e guadagna un punteggio di 0**.
- **sposta-tronco::** Questa azione viene proposta quando l'agente ha un tronco in mano, non ha ancora raggiunto la scalinata (altezza-raggiungibile è minore dell'altezza necessaria per raggiungere la finestra) e l'azione precedente non era sposta-tronco. L'agente sposta il tronco sotto la finestra, aumentando l'altezza raggiungibile e **guadagna un punteggio positivo**.

```
#####
### Raccogli tronchi (nota: potrei mettere un vincolo in modo che l'agente non raccolga piu di 2 tronchi)
#####
sp {propose*raccogli-tronco
  (state <s> ^name escape ^previous <previous-action> ^tronco-in-mano false )
  ( <s> ^altezza-raggiungibile < <altezza> ^scalinata-salita false)
  (<s> ^altezza-finestra <altezza>)
  (<previous-action> ^name <> raccogli-tronco)
-->
  (<s> ^operator <o> +)
  (<o> ^name raccogli-tronco)
}

sp {apply*raccogli-tronco
  [state <s> ^operator.name raccogli-tronco ^previous <p> ]
-->
  (<s> ^previous <p> - <new-previous-action> +)
  (<new-previous-action> ^name raccogli-tronco ^reward 0)
  (<s> ^tronco-in-mano false - true + )
  (write (crlf) |Ha raccolto un tronco.|)
```

^Tronco-in-mano

```
#####
1  ### Spostamento dei tronchi per costruire una scala ###
2  #####
3  sp {propose*sposta-tronco
4    (state <s> ^name escape ^previous <previous-action> ^tronco-in-mano true )
5    (<previous-action> ^name <> sposta-tronco)
6    ( <s> ^altezza-raggiungibile < <altezza> ^scalinata-salita false)
7    (<s> ^altezza-finestra <altezza>)
8  -->
9    (<s> ^operator <o> +)
10   (<o> ^name sposta-tronco)
11 }
12
13 sp {apply*sposta-tronco
14   (state <s> ^operator.name sposta-tronco ^previous <p> ^altezza-raggiungibile <altezza-raggiungibile>
15     ^altezza-tronco <altezza-tronco>)
16 -->
17   (<s> ^previous <p> - <new-previous-action> +)
18   (<new-previous-action> ^name sposta-tronco ^reward 1)
19   (<s> ^altezza-raggiungibile <altezza-raggiungibile> - (+ <altezza-raggiungibile> <altezza-tronco>)+)
20   (<s> ^tronco-in-mano true - false +)
21   (write (crlf) |Ha spostato un tronco sotto la finestra!|)
```

Terminazione

Avviene se l'agente utilizza l'azione **sali-scalinata** e **osserva-rottura-finestra** che implicano la fuga dell'agente.

```
382
383
384 #####
385 ### Escape: termine del gioco ###
386 #####
387 sp {halt*escape
388 | (state <s> ^name escape ^fuggito true)
389 -->
390 | (write (crlf) |Fine del gioco!|)
391 | (halt)
392 }
393
394 # halt se ha lanciato un rametto, allora questo si è rotto e non si può scappare
395 sp {halt*lanciato-rametto-alla-finestra
396 | (state <s> ^name escape ^fuggito false ^lanciato-rametto-alla-finestra true)
397 -->
398 | (write (crlf) |Avendo rotto il rametto, l'agente non può scappare. FINE!|)
399 | (halt)
400 }
401
402 # halt se ha lanciato un rametto, allora questo si è rotto e non si può scappare
403 sp {halt*lanciato-elastico-alla-finestra
404 | (state <s> ^name escape ^fuggito false ^fionda-creata false ^lanciato-elastico-alla-finestra true)
405 -->
406 | (write (crlf) |Avendo perso l'elastico, l'agente non può scappare. FINE!|)
407 | (halt)
408 }
409
410
411
412 # halt se si è combinato degli oggetti ma non si è ottenuta una fionda -> non si può scappare
413 # ( è un caso più generico dei precedenti )
414 sp {halt*wrong-combination
415 | (state <s> ^name escape ^fuggito false ^fionda-creata false ^combinato-oggetti true)
416 -->
```

```
(S1 ^altezza-finestra 10.500000 ^altezza-raggiungibile 10.500000
    ^altezza-tronco 1 ^combinato-oggetti true ^danno-centro-finestra 2
    ^danno-estremite-finestra 15 ^epmem E1 ^finestra-rota true
    ^fionda-creata true ^fuggito false ^io I1
    ^lanciato-elastico-alla-finestra false
    ^lanciato-rametto-alla-finestra false ^name escape ^previous N26
    ^reward-link R1 ^scalinata-salita true ^smem L1 ^superstate nil
    ^tronco-in-mano false ^type state ^vita-finestra -12)
```

^fuggito

^finestra-rota

^scalinata-salita

Rewards

L'inizializzazione dell'agente prevede che questo non abbia alcuna conoscenza relativa ai rinforzi, ma li debba apprendere man mano che applica le SP nelle varie iterazioni.

```
7
8 #####
9 ### Reinforcement learning ###
0 ### Inizializzazione delle ricompense ###
1 #####
2 sp {rl*combina-ramoscello-elastico
3   (state <s> ^name escape ^operator <o> +)
4   (<o> ^name combina-ramoscello-elastico)
5   -->
6   (<s> ^operator <o> = 0)
7 }
8
9 sp {rl*combina-ramoscello-roccia
0   (state <s> ^name escape ^operator <o> +)
1   (<o> ^name combina-ramoscello-roccia)
2   -->
3   (<s> ^operator <o> = 0)
4 }
5
6 sp {rl*combina-roccia-elastico
7   (state <s> ^name escape ^operator <o> +)
8   (<o> ^name combina-roccia-elastico)
9   -->
0   (<s> ^operator <o> = 0)
1 }
2
3 sp {rl*spara-centro-finestra
4   (state <s> ^name escape ^operator <o> +)
```

```
7
8 #####
9 ### Reinforcement learning apply rewards ###
0 ### oppure la notazione di Left-Right ^reward.value ###
1 #####
2
3 sp {rl*apply-rewards
4   (state <s> ^name escape ^reward-link <r> ^previous <previous-action>)
5   (<previous-action> ^reward <reward>)
6   -->
7   (<r> ^reward <rr>)
8   (<rr> ^value <reward>)
9   (write (crlf) |Reward: | <reward>)
0 }
1
2
```

I reward linkati durante l'esecuzione sono invece:

+1

- combina-ramoscello-elastico
- spara-estremite-finestra
- osserva-rottura-finestra
- sposta-tronco
- sali-scalinata

-1

- ombina-ramoscello-roccia
- lancia-elastico
- combina-roccia-elastico
- sposta-tronco
- lancia-ramoscello

0

- spara-centro-finestra
- raccogli-tronco

Rewards

L'inizializzazione dell'agente prevede che questo non abbia alcuna conoscenza relativa ai rinforzi, ma li debba apprendere man mano che applica le SP nelle varie iterazioni.

```
7
8 #####
9 ### Reinforcement learning ###
0 ### Inizializzazione delle ricompense ###
1 #####
2 sp {rl*combina-ramoscello-elastico
3   (state <s> ^name escape ^operator <o> +)
4   (<o> ^name combina-ramoscello-elastico)
5   -->
6   (<s> ^operator <o> = 0)
7 }
8
9 sp {rl*combina-ramoscello-roccia
0   (state <s> ^name escape ^operator <o> +)
1   (<o> ^name combina-ramoscello-roccia)
2   -->
3   (<s> ^operator <o> = 0)
4 }
5
6 sp {rl*combina-roccia-elastico
7   (state <s> ^name escape ^operator <o> +)
8   (<o> ^name combina-roccia-elastico)
9   -->
0   (<s> ^operator <o> = 0)
1 }
2
3 sp {rl*spara-centro-finestra
4   (state <s> ^name escape ^operator <o> +)
```

```
7
8 #####
9 ### Reinforcement learning apply rewards ###
0 ### oppure la notazione di Left-Right ^reward.value ###
1 #####
2
3 sp {rl*apply-rewards
4   (state <s> ^name escape ^reward-link <r> ^previous <previous-action>)
5   (<previous-action> ^reward <reward>)
6   -->
7   (<r> ^reward <rr>)
8   (<rr> ^value <reward>)
9   (write (crlf) |Reward: | <reward>)
0 }
1
2
```

I reward linkati durante l'esecuzione sono invece:

+1

- combina-ramoscello-elastico
- spara-estremite-finestra
- osserva-rottura-finestra
- sposta-tronco
- sali-scalinata

-1

- ombina-ramoscello-roccia
- lancia-elastico
- combina-roccia-elastico
- sposta-tronco
- lancia-ramoscello

0

- spara-centro-finestra
- raccogli-tronco

Ricompense RL associate alle azioni dopo circa 10 iterazioni...

```
rl*sali-scalinata 4.000000 1.073370
rl*osserva-rottura-finestra 4.000000 0.759900
rl*sposta-tronco 54.000000 2.636181
rl*raccogli-tronco 54.000000 2.979489
rl*lancia-ramoscello 1.000000 -0.300000
rl*lancia-elastico 0.000000 0
rl*spara-estremita-finestra 16.000000 2.467944
rl*spara-centro-finestra 1.000000 0.081000
rl*combina-roccia-elastico 2.000000 -0.510000
rl*combina-ramoscello-roccia 1.000000 -0.300000
rl*combina-ramoscello-elastico 4.000000 1.738764
```

Pare che l'agente abbia imparato a **costruire la scala** prima di rompere la finestra.

Questo è anche dovuto al fatto che ho variato le variabili in gioco (portando l'altezza della finestra ad un valore più alto..e ciò richiede quindi più spostamenti di tronchi)

Questo avviene perché l'agente adatta le proprie azioni rispetto all'environment.

```
⊕ --> 1 decision cycle executed. No rules fired.
⊕ run 1 --phase
  Inizia il gioco!
```

```
selection probabilities:
O6 (raccogli-tronco) + = 3.104987588174417 :I (92.00%)
  From propose*raccogli-tronco

O5 (lancia-ramoscello) + = -0.3 :I (2.00%)
  From propose*lancia-ramoscello

O4 (combina-roccia-elastico) + = -0.7599 :I (2.00%)
  From propose*combina-roccia-elastico

O3 (combina-ramoscello-roccia) + = -0.51 :I (2.00%)
  From propose*combina-ramoscello-roccia

O2 (combina-ramoscello-elastico) + = 2.412019975480777 :I (2.00%)
  From propose*combina-ramoscello-elastico
```