



# Progetto d'esame per la parte programmazione architetture cognitive

- ❖ Aldo Rambaudo
- ❖ Michele Colombino
- ❖ Vito Ruggiero



# Vincoli Progetto “ESCAPE”

- ❖ L'agente è prigioniero in un ambiente dove l'unica via d'uscita è una finestra posta a 3,5 metri di altezza
- ❖ L'agente è un robot di altezza 150 cm
- ❖ La finestra è blindata ma ha un punto debole alle estremità.
- ❖ L'agente ha a disposizione diversi oggetti che possono essere combinati:
  - una molla,
  - una rametto in legno,
  - ciottoli e pietre,
  - due tronchi d'albero dello stesso diametro da 1 metro di altezza ciascuno.
- ❖ L'agente può decidere di creare nuovi oggetti a partire da quelli che ha a disposizione

# Struttura dell'agente

## Ciclo deliberativo

- **OBSERVE**

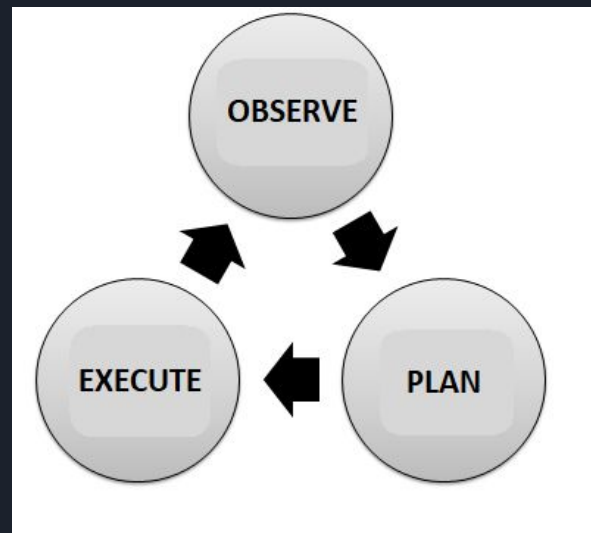
- L'agente osserva l'ambiente
- Elenca gli oggetti rilevanti e le loro posizioni

- **PLAN**

- L'agente formula un piano
- Ogni piano definisce quali oggetti usare, da quale posizione della stanza lanciare e il punto della finestra da colpire.

- **EXECUTE**

- L'agente esegue le azioni pianificate





# Scelte implementative

- ❖ Per semplicità gli elementi di gioco sono individuati da posizioni “vaghe” tramite coordinate Nord, sud, est, ovest, centro.
- ❖ L'agente e gli oggetti di gioco hanno una posizione iniziale
- ❖ Un attributo “azione” associato allo stato attuale segna l'azione in corso e modella il passaggio da uno stato all'altro
- ❖ I movimenti dell'agente sono limitati alle singole azioni da compiere, come “prendere un oggetto”, “salire sui tronchi”, “posizionarsi a Nord”...
- ❖ Una volta combinati e recuperati, l'uso degli oggetti è immediato

# Scelte implementative

- ❖ Le coordinate, per semplicità sono associate a dei numeri.
- ❖ La finestra nasce con dei punti vita. Ogni colpo (qualsiasi) da parte dell'agente infligge dei danni alla finestra.
- ❖ Ogni qualvolta l'agente colpisce la finestra perde gli oggetti usati, i quali cadono in posizioni randomiche
- ❖ Il danno inflitto alla finestra è una combinazione lineare dei valori di reward

```
# 1 SUD, 2 EST, 3 OVEST, 4 CENTER, 5 NORD
(<env>      ^has_object <window> <obj1> <obj2> <obj3> <obj4>)
(<window>   ^has_life_points 60 ^is_in_position 5 ^name window)

(<obj1> ^name spring    ^is_in_position 2)
(<obj2> ^name rocks     ^is_in_position 2)
(<obj3> ^name stick     ^is_in_position 3)
(<obj4> ^name trunks    ^is_in_position 4)
```

```
(<agent> ^has_object <ob> - nil +)
(<env_object> ^is_in_position <pos_obj> - (+ (rand-int 5) 1) +)

(write (crlf) |INFO                      Agent loose an object: | <ob>)
(write (crlf))
```

```
# damage score:      (5 * combine + 5 * throwing + 10 * position)
- (- <points> (max 1 (+ (+ (* 5 <rc>) (* 5 <rt>)) (* 10 <rp>)))) ) )
----- FIRE ON WINDOW|)
```

# Sistema di Reward

## Tre sistemi di reward

- ❖ Reward sulla combinazione di oggetti (**combining strategy**)
- ❖ Reward sulla posizione della stanza da cui lanciare gli oggetti combinati, (**Throwing tool strategy**) Nord, Sud, est, ovest, centro, tronchi
- ❖ Reward sulla posizione della finestra a cui effettuare il lancio (**Position window strategy**) : bordo o centro

```
# COMBINING TOOL STRATEGY
(<s1> ^name tool1 ^print_command spring-stick ^reward 1)
(<s2> ^name tool2 ^print_command rocks-stick ^reward -1)
(<s3> ^name tool3 ^print_command rocks-spring ^reward -1)
(<s4> ^name tool4 ^print_command stick ^reward -2)
(<s5> ^name tool5 ^print_command rocks ^reward -2)
(<s6> ^name tool6 ^print_command spring ^reward -2)

# THROWING TOOL STRATEGY
(<t1> ^name on_trunks ^print_command trunks ^reward 1)
(<t2> ^name 4 ^print_command center ^reward -0.5)
(<t3> ^name 5 ^print_command nord ^reward -0.5)
(<t4> ^name 1 ^print_command sud ^reward -2)
(<t5> ^name 2 ^print_command est ^reward -1)
(<t6> ^name 3 ^print_command ovest ^reward -1)

# POSITION WINDOW STRATEGY
(<p1> ^name border ^reward 1) # BORDER
(<p2> ^name center ^reward -1) # CENTER
```

# Sistema di Reward

I sistemi di reward scelti guidano l'apprendimento dell'agente di modo che le sue scelte, a seguito di diversi tentativi, cadano su alcune **“combinazioni vincenti”**:

- ❖ Best Reward *combining strategy* **tool1** +1  
Fionda (rametto + molla)
- ❖ Best Reward *throwing* **on\_trunks** +1  
Salire sui tronchi
- ❖ Best Reward *position strategy*: **border** +1  
Mirare ai bordi della finestra

```
agent*rl*strategy*fire*on*CENTER 3.000000 -3.211628
agent*rl*strategy*fire*on*BORDER 4.000000 7.503695
agent*rl*strategy-on-NORD 0.000000 0
agent*rl*strategy-on-EST 0.000000 0
agent*rl*strategy-on-OVEST 0.000000 0
agent*rl*strategy-on-SUD 0.000000 0
agent*rl*strategy-on-CENTER 0.000000 0
agent*rl*strategy-on_trunks 7.000000 4.071147
agent*rl*strategy-tool6*SPRING 0.000000 0
agent*rl*strategy-tool5*ROCKS 0.000000 0
agent*rl*strategy-tool4*STICK 0.000000 0
agent*rl*strategy-tool3*ROCKS-SPRING 2.000000 0.379940
agent*rl*strategy-tool2*ROCKS-STICK 0.000000 0
agent*rl*strategy-tool1*SPRING-STICK 5.000000 3.860033
```

```
agent*rl*strategy*fire*on*CENTER 3.000000 -3.211628
agent*rl*strategy*fire*on*BORDER 7.000000 10.178313
agent*rl*strategy-on-NORD 0.000000 0
agent*rl*strategy-on-EST 0.000000 0
agent*rl*strategy-on-OVEST 0.000000 0
agent*rl*strategy-on-SUD 0.000000 0
agent*rl*strategy-on-CENTER 0.000000 0
agent*rl*strategy-on_trunks 10.000000 8.136373
agent*rl*strategy-tool6*SPRING 0.000000 0
agent*rl*strategy-tool5*ROCKS 0.000000 0
agent*rl*strategy-tool4*STICK 0.000000 0
agent*rl*strategy-tool3*ROCKS-SPRING 2.000000 0.379940
agent*rl*strategy-tool2*ROCKS-STICK 0.000000 0
agent*rl*strategy-tool1*SPRING-STICK 8.000000 6.328694
```



# Esecuzione e risultati

Per valutare il sistema abbiamo condotto diversi test, considerando diverse inizializzazioni per i valori di reward. Dopo diversi tentativi abbiamo tracciato due test principali:

- ❖ TEST 1 - `indifferent-selection -e 0.1` (10% deviation from greedy)
- ❖ TEST 2 - `indifferent-selection -e 0.2` (20% deviation from greedy)



# Test e Risultati

## TEST 1 (da 253 a 128 cicli di elaborazione)

- ❖ In media, dopo 10 tentativi, con il 10% di deviation, l'agente impiega in media 11 round per completare il task
- ❖ Avendo molte scelte, se nei primi round l'agente esegue quelle meno premianti è difficile che provi altre strade

## Esecuzione 1

```
Values from single timers:
Kernel CPU Time:      0.019 sec.
Total CPU Time:       9.086 sec.

101 decisions (0.183 msec/decision)
253 elaboration cycles (2.505 ec's per dc, 0.073 msec/ec)
253 inner elaboration cycles
101 p-elaboration cycles (1.000 pe's per dc, 0.183 msec/pe)
502 production firings (1.984 pf's per ec, 0.037 msec/pf)
2113 wme changes (1115 additions, 998 removals)
WM size: 117 current, 121.004 mean, 141 maximum

<
state operator stack matches op_pref stats input output

print --rl

agent*rl*strategy*fire*on*CENTER  6.000000 0.583980
agent*rl*strategy*fire*on*BORDER  4.000000 7.231950
agent*rl*strategy-on-NORD  0.000000 0
agent*rl*strategy-on-EST  0.000000 0
agent*rl*strategy-on-OVEST  2.000000 0.286383
agent*rl*strategy-on-SUD  1.000000 0.300000
agent*rl*strategy-on-CENTER  0.000000 0
agent*rl*strategy-on-Trunks  7.000000 3.974070
agent*rl*strategy-tool6*SPRING  1.000000 -0.330000
agent*rl*strategy-tool5*ROCKS  0.000000 0
agent*rl*strategy-tool4*STICK  0.000000 0
agent*rl*strategy-tool3*ROCKS-SPRING  0.000000 0
agent*rl*strategy-tool2*ROCKS-STICK  0.000000 0
agent*rl*strategy-tool1*SPRING-STICK  9.000000 3.112277

----- Agent Lee IS ESCAPED !!!
----- Agent Lee TOOK 10 ROUND TO ESCAPE!
```

## Esecuzione 2

```
Values from single timers:
Kernel CPU Time:      0.009 sec.
Total CPU Time:       4.041 sec.

51 decisions (0.170 msec/decision)
128 elaboration cycles (2.510 ec's per dc, 0.068 msec/ec)
128 inner elaboration cycles
51 p-elaboration cycles (1.000 pe's per dc, 0.170 msec/pe)
248 production firings (1.938 pf's per ec, 0.035 msec/pf)
1103 wme changes (610 additions, 493 removals)
WM size: 117 current, 120.083 mean, 141 maximum

<
state operator stack matches op_pref stats input output

print --rl

agent*rl*strategy*fire*on*CENTER  6.000000 0.583980
agent*rl*strategy*fire*on*BORDER  9.000000 8.849973
agent*rl*strategy-on-NORD  0.000000 0
agent*rl*strategy-on-EST  0.000000 0
agent*rl*strategy-on-OVEST  3.000000 1.823095
agent*rl*strategy-on-SUD  1.000000 0.300000
agent*rl*strategy-on-CENTER  0.000000 0
agent*rl*strategy-on-Trunks  11.000000 8.755040
agent*rl*strategy-tool6*SPRING  1.000000 -0.330000
agent*rl*strategy-tool5*ROCKS  1.000000 -0.252677
agent*rl*strategy-tool4*STICK  1.000000 0.742959
agent*rl*strategy-tool3*ROCKS-SPRING  0.000000 0
agent*rl*strategy-tool2*ROCKS-STICK  0.000000 0
agent*rl*strategy-tool1*SPRING-STICK  12.000000 5.891718

----- Agent Lee IS ESCAPED !!!
----- Agent Lee TOOK 5 ROUND TO ESCAPE!
```

# Test e Risultati

## TEST 2 (da 178 a 103 cicli di elaborazione)

- ❖ In media, dopo 10 tentativi, con il 20% di deviation, l'agente impiega in media 6 round per completare il task
- ❖ L'agente ha più possibilità di variare le scelte. Una volta appreso i best reward è difficile che cambi strategia

## Esecuzione 1

```
Values from single timers:
Kernel CPU Time:      0.015 sec.
Total CPU Time:       6.077 sec.

71 decisions (0.214 msec/decision)
178 elaboration cycles (2.507 ec's per dc, 0.085 msec/ec)
178 inner elaboration cycles
71 p-elaboration cycles (1.000 pe's per dc, 0.214 msec/pe)
351 production firings (1.972 pf's per ec, 0.043 msec/pf)
1509 wme changes (813 additions, 696 removals)
WM size: 117 current, 120.638 mean, 141 maximum

<
state operator stack matches op_pref stats input output

print --ri

agent*rl'strategy*fire*on*CENTER 3.000000 0.724593
agent*rl'strategy*fire*on*BORDER 4.000000 6.713235
agent*rl'strategy-on-NORD 1.000000 0.160950
agent*rl'strategy-on-EST 0.000000 0
agent*rl'strategy-on-OVEST 0.000000 0
agent*rl'strategy-on-SUD 0.000000 0
agent*rl'strategy-on-CENTER 0.000000 0
agent*rl'strategy-on-trunks 6.000000 4.095615
agent*rl'strategy-tool6*SPRING 0.000000 0
agent*rl'strategy-tool5*ROCKS 0.000000 0
agent*rl'strategy-tool4*STICK 1.000000 0.788227
agent*rl'strategy-tool3*ROCKS-SPRING 0.000000 0
agent*rl'strategy-tool2*ROCKS-STICK 0.000000 0
agent*rl'strategy-tool1*SPRING-STICK 6.000000 3.825940

----- Agent Lee IS ESCAPED !!!
----- Agent Lee TOOK 7 ROUND TO ESCAPE!
```

## Esecuzione 2

```
Values from single timers:
Kernel CPU Time:      0.007 sec.
Total CPU Time:       3.028 sec.

41 decisions (0.163 msec/decision)
103 elaboration cycles (2.512 ec's per dc, 0.065 msec/ec)
103 inner elaboration cycles
41 p-elaboration cycles (1.000 pe's per dc, 0.163 msec/pe)
203 production firings (1.971 pf's per ec, 0.033 msec/pf)
909 wme changes (513 additions, 396 removals)
WM size: 117 current, 119.833 mean, 141 maximum

<
state operator stack matches op_pref stats input output

print --ri

agent*rl'strategy*fire*on*CENTER 4.000000 1.034620
agent*rl'strategy*fire*on*BORDER 7.000000 9.878174
agent*rl'strategy-on-NORD 1.000000 0.160950
agent*rl'strategy-on-EST 0.000000 0
agent*rl'strategy-on-OVEST 0.000000 0
agent*rl'strategy-on-SUD 0.000000 0
agent*rl'strategy-on-CENTER 0.000000 0
agent*rl'strategy-on-trunks 10.000000 7.420565
agent*rl'strategy-tool6*SPRING 0.000000 0
agent*rl'strategy-tool5*ROCKS 0.000000 0
agent*rl'strategy-tool4*STICK 1.000000 0.788227
agent*rl'strategy-tool3*ROCKS-SPRING 0.000000 0
agent*rl'strategy-tool2*ROCKS-STICK 0.000000 0
agent*rl'strategy-tool1*SPRING-STICK 10.000000 5.889535

----- Agent Lee IS ESCAPED !!!
----- Agent Lee TOOK 4 ROUND TO ESCAPE!
```



Grazie per l'attenzione