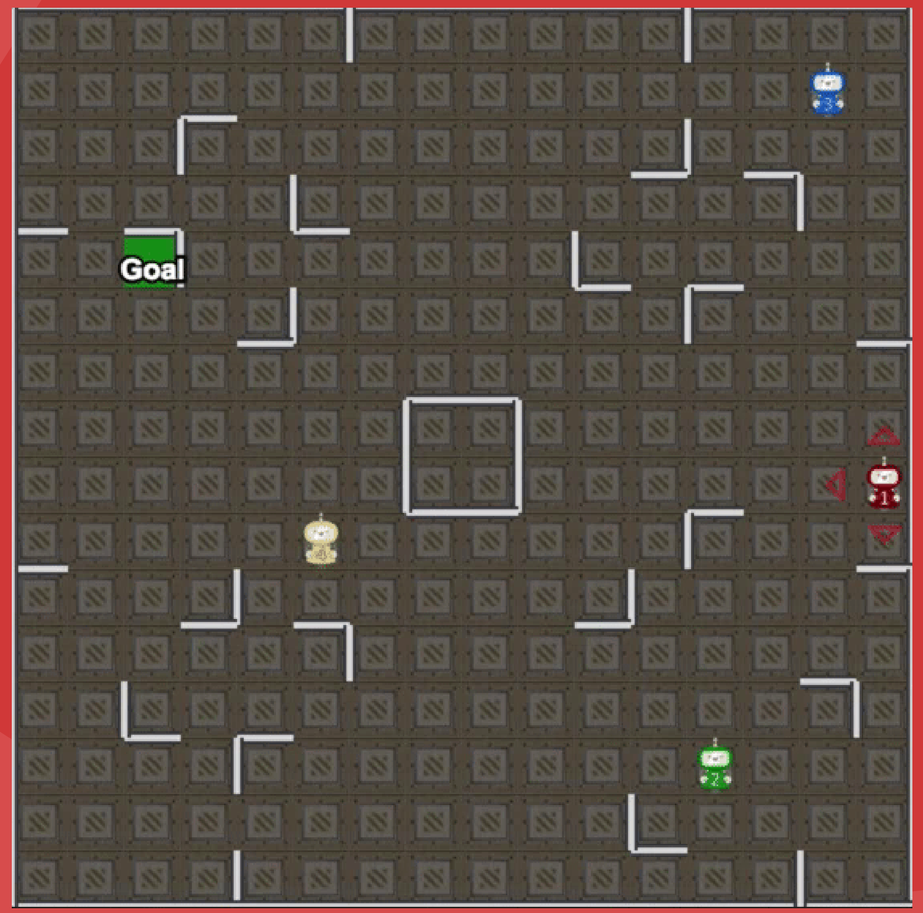
Grupo 1

Bruno Ribeiro - Danilo Carvalho - Igor Penha - Lucas Gobbi

O JOGO



Objetivo: O objetivo é levar um dos robôs até a linha de chegada. Dependendo da fase do jogo, é necessário que o robô que alcance a linha de chegada seja de uma cor específica.

Movimentos: o usuário move um robô para qualquer direção entre Norte, Sul, Leste e Oeste, parando ao encontrar um obstáculo, seja parede ou outro robo.

Sobre o Domínio

O domínio foi desenvolvido para a IPC de 2023, onde foi utilizado para avaliar os planejadores enviados para a competição.

Este domínio é uma reformulação do domínio de mesmo nome, porém desenvolvido para a “ASP Competition 2015”, em que a Rebecca Eifler (Saarland University) fez contribuições acentuadas no desenvolvimento do domínio atualizado.

Domínio

```
(define (domain ricochet-robots)
(:requirements :typing :adl :action-costs)

(:types
    robot - object
    cell - object
    direction - object
)
```

Domínio

```
(:predicates
     (NEXT ?c - cell ?cnext - cell ?dir -
direction)
     (BLOCKED ?c - cell ?dir - direction)
     (at ?r - robot ?c - cell)
     (free ?c - cell)
     (nothing-is-moving)
     (is-moving ?r - robot ?dir -
direction)
)
```

Domínio

```
(:functions
     (total-cost) - number
     (go-cost) - number
     (step-cost) - number
     (stop-cost) - number
)
```

Domínio

```
(:action go
    :parameters (?r - robot ?dir - direction)
    :precondition
        (and
            (nothing-is-moving)
        )
     :effect
        (and
            (not (nothing-is-moving))
            (is-moving ?r ?dir)
            (increase (total-cost) (go-cost))
        )
)
```

Domínio

```
(:action step
    :parameters (?r - robot ?cfrom - cell ?cto - cell ?
dir - direction)
    :precondition
        (and
            (is-moving ?r ?dir)
            (at ?r ?cfrom)
            (NEXT ?cfrom ?cto ?dir)
            (free ?cto)
            (not (BLOCKED ?cfrom ?dir))
        )
    :effect
        (and
            (not (at ?r ?cfrom))
            (free ?cfrom)
            (not (free ?cto))
            (at ?r ?cto)
            (increase (total-cost) (step-cost))
        )
```

Domínio

```
(:action stop-at-barrier
    :parameters (?r - robot ?cat - cell ?dir -
direction)
    :precondition
        (and
            (is-moving ?r ?dir)
            (at ?r ?cat)
            (BLOCKED ?cat ?dir)
        )
    :effect
        (and
            (not (is-moving ?r ?dir))
            (nothing-is-moving)
            (increase (total-cost) (stop-cost))
        )
)
```

Domínio

```
(:action stop-at-robot
    :parameters (?r - robot ?cat - cell ?cnext - cell ?
dir - direction)
    :precondition
        (and
            (is-moving ?r ?dir)
            (at ?r ?cat)
            (NEXT ?cat ?cnext ?dir)
            (not (free ?cnext))
        )
    :effect
        (and
            (not (is-moving ?r ?dir))
            (nothing-is-moving)
            (increase (total-cost) (stop-cost))
        )
)
)
```

Tabela de Execuções

Peak memory: 10432 KB
INFO Planner time: 0.17s

```
a211039288@gpu1:~$ cat sas_plan
(go robot-1 north)
(step robot-1 cell-3-3 cell-3-2 north)
(stop-at-robot robot-1 cell-3-2 cell-3-1 north)
; cost = 1 (general cost)
a211039288@gpu1:~$
```

Tabela de Execuções

Peak memory: 10980 KB
INFO Planner time: 0.25s

Tabela de Execuções

```
a210392888@gpu1:~$ cat sas_plan
(go robot-3 east)
(step robot-3 cell-1-1 cell-2-1 east)
(step robot-3 cell-2-1 cell-3-1 east)
(step robot-3 cell-3-1 cell-4-1 east)
(stop-at-robot robot-3 cell-4-1 cell-5-1 east)
(go robot-1 west)
(step robot-1 cell-4-5 cell-3-5 west)
(stop-at-barrier robot-1 cell-3-5 west)
(go robot-1 north)
(step robot-1 cell-3-5 cell-3-4 north)
(stop-at-barrier robot-1 cell-3-4 north)
(go robot-1 east)
(step robot-1 cell-3-4 cell-4-4 east)
(stop-at-barrier robot-1 cell-4-4 east)
(go robot-3 south)
(step robot-3 cell-4-1 cell-4-2 south)
(step robot-3 cell-4-2 cell-4-3 south)
(stop-at-robot robot-3 cell-4-3 cell-4-4 south)
(go robot-3 east)
(step robot-3 cell-4-3 cell-5-3 east)
(stop-at-barrier robot-3 cell-5-3 east)
(go robot-3 west)
(step robot-3 cell-5-3 cell-4-3 west)
(step robot-3 cell-4-3 cell-3-3 west)
(step robot-3 cell-3-3 cell-2-3 west)
(step robot-3 cell-2-3 cell-1-3 west)
(stop-at-barrier robot-3 cell-1-3 west)
```

```
(go robot-3 north)
(step robot-3 cell-1-3 cell-1-2 north)
(stop-at-barrier robot-3 cell-1-2 north)
(go robot-3 south)
(step robot-3 cell-1-2 cell-1-3 south)
(step robot-3 cell-1-3 cell-1-4 south)
(stop-at-barrier robot-3 cell-1-4 south)
(go robot-3 east)
(step robot-3 cell-1-4 cell-2-4 east)
(step robot-3 cell-2-4 cell-3-4 east)
(stop-at-robot robot-3 cell-3-4 cell-4-4 east)
(go robot-1 north)
(step robot-1 cell-4-4 cell-4-3 north)
(step robot-1 cell-4-3 cell-4-2 north)
(step robot-1 cell-4-2 cell-4-1 north)
(stop-at-barrier robot-1 cell-4-1 north)
(go robot-3 east)
(step robot-3 cell-3-4 cell-4-4 east)
(stop-at-barrier robot-3 cell-4-4 east)
(go robot-3 north)
(step robot-3 cell-4-4 cell-4-3 north)
(step robot-3 cell-4-3 cell-4-2 north)
(stop-at-robot robot-3 cell-4-2 cell-4-1 north)
; cost = 13 (general cost)
```


ALEX RANDOLPH
RICOCHET ROBOTS
DEVIR
ESP CAT POR
+10 años
30 min

Obrigado!