

Competição

Inteligência Artificial

Universidade Federal do ABC

22 de novembro de 2017

INTRODUÇÃO

- ▶ A equipe desenvolvedora de cada time da competição pode ser formada por 2 pessoas
- ▶ Na competição cada partida é disputada por 2 times, cada um com 2 agentes
 - ▶ O mapa é dividido em dois
 - ▶ Time vermelho deve comer todas as comidas do lado do time azul e vice-versa
 - ▶ Cada agente atua como um fantasma quando está no seu lado do mapa
 - ▶ Ao atravessar para o outro lado, o agente se torna um Pacman

INTRODUÇÃO

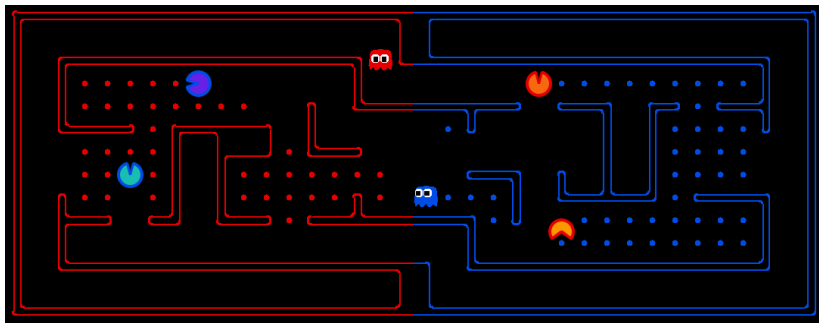


Figura 1: Competição

INTRODUÇÃO

- ▶ Ganha quem comer mais comida
 - ▶ A contabilização da comida só é feita *após* o retorno ao seu lado do mapa!
 - ▶ Capturar o fantasma adversário *não* gera pontos
- ▶ Uma partida termina quando um time capturar todas *menos* 2 comidas do adversário
 - ▶ Ou um máximo de 300 movimentos por agente ser realizado

INTRODUÇÃO

- ▶ Seu agente deverá selecionar uma ação em no máximo *1 segundo*
 - ▶ Se um total de 3 segundos de penalização for alcançado, você perde a partida!
 - ▶ No início da partida seu agente pode gastar 15 segundos para pré-computar valores
- ▶ Caso seu agente (ou o outro agente do time) estiver a uma distância ≤ 5 do adversário
 - ▶ A direção e a posição do adversário pode ser acessada (`agentState.getPosition`, `getDirection`)
- ▶ Adicionalmente, você tem acesso a uma leitura de distância ruidosa para cada agente

ARQUIVOS

- ▶ Baixe o arquivo `competicao.zip` e descompacte
- ▶ Você modificará apenas o seguinte arquivo:
 - ▶ `myTeam.py`
- ▶ Os seguintes arquivos contém informações relevantes:
 - ▶ `capture.py`: Definição do `GameState`
 - ▶ `captureAgents.py`: Classe pai para agentes
 - ▶ `baselineTeam.py`: Exemplo de agentes
 - ▶ `distanceCalculator.py`: Computa distância do labirinto

DESENVOLVENDO SUA EQUIPE

- ▶ Entenda os agentes desenvolvidos em `baselineTeam.py`
 - ▶ Você não precisa usar a mesma abordagem
 - ▶ Perceba o tipo de retorno de cada método!
 - ▶ `registerInitialState`
 - ▶ `chooseAction`
- ▶ Ao submeter seu código não esqueça de modificar os parâmetros `first` e `second` da função `createTeam`
 - ▶ Devem ser o nome das classes dos seus agentes!

DESENVOLVENDO SUA EQUIPE

- ▶ A classe do seu agente deve estender a classe **CaptureAgent**
- ▶ Os seguintes métodos estarão disponíveis:
 - ▶ `getFood(self, gameState)`: retorna o grid de comida que você deve *comer*, `m[x][y]=True` se na posição `x,y` tem uma comida que você pode comer
 - ▶ `getFoodYouAreDefending(self, gameState)`: retorna o grid de comida que você deve *defender*, `m[x][y]=True` se na posição `x,y` tem uma comida que seu adversário pode comer

DESENVOLVENDO SUA EQUIPE

- ▶ A classe do seu agente deve estender a classe **CaptureAgent**
- ▶ Os seguintes métodos estarão disponíveis:
 - ▶ `getOpponents(self, gameState)`: índices dos agentes do time adversário
 - ▶ `getTeam(self, gameState)`: índices dos agentes do seu time
 - ▶ `getScore(self, gameState)`: diferença entre o número de comidas que seu time pegou e do adversário

DESENVOLVENDO SUA EQUIPE

- ▶ A classe do seu agente deve estender a classe **CaptureAgent**
- ▶ Os seguintes métodos estarão disponíveis:
 - ▶ `getMazeDistance(self, pos1, pos2)`: retorna a distância do labirinto entre dois pontos
 - ▶ Pré-computadas pelo **registerInitialState** de **CaptureAgent** (não esqueça de chamá-lo!)
 - ▶ `getPreviousObservation(self)`: retorna o **GameState** do estado correspondente ao ultimo movimento do agente
 - ▶ `getCurrentObservation(self)`: retorna o **GameState** atual

PRATICANDO

- ▶ Para iniciar uma partida usando um time use o parâmetro -r
- ▶ `python2.7 capture.py -r baselineTeam -b baselineTeam`
 - ▶ Inicia uma partida com o time vermelho e azul criados a partir de `baselineTeam.py`
- ▶ Para testar outros labirintos use o parâmetro -l
 - ▶ Em conjunto com o nome de um arquivo em `layouts/`
 - ▶ Ou RANDOMN onde N é um número de semente para gerar um mapa aleatório
 - ▶ `-l RANDOM13`

PRATICANDO

- ▶ É possível gravar e rever partidas
 - ▶ `--record` para gravar
 - ▶ `--replay` para assistir
- ▶ Submissão de times ocorrerão em duas datas:
 - ▶ 05/12 e 12/12
 - ▶ No topo do arquivo **myTeam.py** deve constar o nome dos membros da dupla
 - ▶ Todos os times jogarão contra todos os times
 - ▶ Ranking final baseado no número de vitórias de cada time
- ▶ Boa sorte!