

# Dig Dug AI

Inteligência Artificial

Pedro Matos – 102993

Simão Antunes - 104092

# Ficheiro

- ◊ Todo o código produzido para a realização deste projeto encontra-se no ficheiro *student.py*.
- ◊ O *student.py*, contém:
  - ◊ A classe *SearchProblem*;
  - ◊ A classe *SearchNode*;
  - ◊ A classe *SearchTree*;
  - ◊ A classe *DigDugSearchDomain*;
  - ◊ A função *agente\_loop*;
  - ◊ E por último um conjunto de funções auxiliares.

# Algoritmo de pesquisa

- ◈ O algoritmo de pesquisa implementado foi o de pesquisa gulosa, usando apenas uma heurística, (distância do Dig Dug ao inimigo).
- ◈ Achamos também relevante realçar que no `DigDugSearchDomain()` a função, *actions(self, coords)*, que determina as ações possíveis tendo em conta as coordenadas dadas, tem em conta nessa determinação as posições das pedras existentes no mapa, (ex: se a pedra está em cima do dig dug, as únicas ações possíveis serão: ir para a direita ou para a esquerda), e também tem em conta o caso do inimigo do tipo fygar, ou seja, as coordenadas que estão á frente do fygar até quatro unidades de distância são intransitáveis, sendo que, as ações, que no momento façam com que o dig dug fique nessas coordenadas, não sejam consideradas como ações possíveis a tomar.

# Agente

- ◆ O agente pode ser dividido em duas partes:
  - ◆ 1º parte: descrição do mundo, ou seja, é aqui que o agente recebe a informação sobre o estado de todos os elementos do jogo, e onde o guarda e analisa,(calcula o inimigo mais próximo, o armazenamento dos dados do inimigo mais próximo, definição do domínio do problema entre outros).
  - ◆ 2º parte: reação ao mundo, é aqui que o agente toma ações para influenciar o mundo do jogo, nesta parte o agente tem dois tipos de comportamento distintos. O agente ou calcula o caminho mais rápido até ao objetivo definido na 1ª parte e toma as ações necessárias para o realizar ou quando está a 3 unidades de distância do inimigo toma as ações necessárias para o eliminar.



# Agente - estratégias

- ◆ Para garantir o bom funcionamento do agente no contexto do problema implementamos várias estratégias para o conseguir:
  - ◆ O objetivo final do problema para calcular o caminho na serachTree é sempre as coordenadas a 3 unidades de distância atrás do inimigo, tendo em conta a sua direção de movimento.
  - ◆ Se o inimigo que o dig dug está a ir atrás é um fygar então as 4 coordenadas que se encontram à frente desse fygar, o dig dug não pode estar.
  - ◆ É guardada a informação do inimigo mais perto de forma a se poder calcular a direção do movimento do mesmo e saber se é do tipo fygar ou não.
  - ◆ Quando o dig dug se encontra a 3 unidades de distancia do inimigo, é usada a função *ver\_coords* que determina a melhor ação a tomar.
  - ◆ Na função *ver\_coords* é tido em conta se o inimigo está em campo aberto ou não,(matar/não matar) o que implicou o armazenamento do estado do mapa assim como a sua atualização.

NOTA: estas não são as únicas estratégias mas apenas as mais relevantes que usamos neste projeto.