

Projeto IIA - Bomberman

Diogo Fernandes 89221

Pedro Alves 88861

04 de Dezembro de 2019



O nosso agente tem como base uma fila de ações (actions_in_queue) onde qualquer conjunto de ações (ou ação individual) são colocadas. No final de iteração do jogo, uma ação (key) é retirada e enviada para o servidor. Adota diferentes estratégias para diferentes níveis e situações recorrendo às funções implementadas e à árvore de decisões.

Funções Principais:

mover(player_pos,dst_pos) -> Função que devolve o caminho para a posição destino(dst_pos) a partir da posição atual(player_pos). Implementada com base no algoritmo de pesquisa A*. Quando não encontra o caminho (excedeu o n de iterações) devolve um para a parede mais próxima

dodge_ext(bomb_pos,bomb) / dodge_int(bomb_pos,bomb)->Função chamada ao fim de plantar uma bomba para encontrar um caminho para uma posição fora de perigo, em direção contrária ao inimigo. Se não encontrar é chamada a função **dodge_int()**, que procura uma posição livre na mesma mas pode ser em direção ao inimigo. Função **dodge_ext** e **dodge_int** implementada com base no algoritmo de pesquisa **breadth_first**. Poderíamos ter implementado melhor estas duas funções com recurso a uma flag.

Kill(pos,w)-> Função que utiliza a **in_range()** para descobrir a posição onde se deve plantar a bomba de forma a matar o inimigo. Depois de obter a (kill pos) esta é enviada para a **mover()** que por sua vez vai retornar o caminho desejado.

In_danger(player_pos,key)-> Função para que vê as proximidade do jogador e dá return de true se tiver em perigo.

Funções secundárias:

coord2dir(lista)-> Dado um caminho(lista de posições) calcula as keys necessárias para fazer esse caminho e coloca-as na fila de ações.

entity_Finder()->Encontra o objeto mais próximo na lista.

side_step(pos)/can_side_step(pos)-> função que retorna uma posição adjacente de pos que está desocupada

enemy_dir()/calc_danger()-> funções para indicar a próxima posição dos inimigos com base na direção deles

In_range(enemy)->Função que itera todas as posições à volta do inimigo vendo as que matariam (in range) devolvendo uma delas

dir_enemy(enemy_pos)->Em todas iterações tendo em conta todas posições prévias e atuais dos inimigos infere a direção dos mesmos

calc_danger(enemy_pos,list_directions)->Tendo em conta as direções devolvidas pela dir_enemy(enemy_pos) devolve uma lista com todas as futuras posições dos inimigos

Funções auxiliares

get_enemyPos()

Plant_bomb()

go2Wall()

get_power()

near_wall()

distancia_calculation()

plant_bomb()

to_exit()

Get_enemyName()

Get_index()

Is_oneal()

Get_out()