

IIA-BOMBERMAN

Francisco Petronilho – 89241

Rui Pedro – 89293

Gonçalo Perna - 88823



Ficheiros

- Student.py
- Movement.py
- Ai.py (Classe Ai e Node)

Como se movimenta?

- A classe Ai tem uma variável “moves” que é uma lista com todos os movimentos calculados anteriormente.
- Cada letra na lista é enviada para o servidor no student.py caso cumpra todos os requisitos.

Student.py

- A primeira iteração cria 2 mapas que são usados na classe AI.
- A cada iteração envia o “state” para a classe AI pela função “update” e “update_map”, a primeira função faz update a algumas variaveis necessárias na classe (por exemplo, posição do bomberman, inimigos, powerups,etc.), a segunda faz update a um mapa da classe que tem informações importantes (por exemplo o raio de explosão na bomba), estas estão diferenciadas por diferentes números.
- Após os updates, decide as próximas ações do bomberman, entre estas ações está, por exemplo, partir paredes, apanhar o powerup, ir para a saída, perseguir inimigos, etc. Após a decisão é guardada na variável “action” da classe Ai e chama a função “think”. Caso esteja a meio de um movimento vai ignorar esta parte do código.
- De seguida, vai efetuar o próximo movimento na lista (pode ser um movimento ou colocar/detonar uma bomba), aqui há várias coisas que podem evitar isto, por exemplo, caso um inimigo esteja a 2 blocos do bomberman este coloca a bomba e mete os movimentos para fugir na lista.

Movement.py

- Tem várias funções que auxiliam a calcular os movimentos.
- A função “destroy_wall” chamada na “think” do Ai, chama outras funções para calcular o caminho para destruir a próxima parede. Uma delas a “calc_bomb_deploy” que calcula a posição onde deve meter a bomba.
- Também contém a função que traduz a lista com posições que sai da função “findpath”.
- As restantes funções auxiliam as faladas anteriormente.

Ai.py: Componentes Importantes

- think
- findpath
- update *
- update_map *

*já explicado

Ai.py: think

- Esta função usa a variável “action” e dependendo do seu valor vai chamar outras funções para calcular os movimentos.
- Tem algumas redundâncias para evitar loops infinitos. Isto usa uma variável incrementável para cada “action” que, por exemplo, a cada step a perseguir um inimigo incrementa a variável correspondente e quando atingir o limite definido muda a decisão feita no student e parte a parede mais próxima. Cada action faz reset aos contadores das outras.

Ai.py: findpath

- Esta função usa uma versão do A*, com adaptações para esta aplicação.
- Tem um limite máximo de nós abertos dependendo da ação.
- Abre todos os nós correspondente às posições adjacentes(exceto as posições diagonais) e também a posição do bomberman.
- As paredes são ignoradas visto que o bomberman não pode passar sem o powerup wallpass*.
- Caso o powerup esteja em campo, para evitar passar o nível sem o powerup, a saída é ignorada.
- A escolha do melhor nó é influenciada pela distância ao nó anterior, e caso haja inimigos perto ou uma bomba perto(estas informações estão nos mapas já referidos).

* desativado porque piorava a pontuação