



PROJETO FINAL


DIG DUG

107348 | Pedro Ramos
107603 | Daniel Madureira
107876 | Gabriel Teixeira



ALGORITMO UTILIZADO

PESQUISA A*



Para a navegação e aproximação aos inimigos, foi utilizada uma pesquisa do tipo A* em árvore que prioriza o inimigo mais próximo (distância de Manhattan).

Esta pesquisa está otimizada para diminuir o número de túneis cavados e evitar túneis com formatos diagonais ou excessivamente cavados.

Túneis “diagonais” são evitados pois inimigos em modo “ghost” são muito mais perigosos neste tipo de caminhos, podendo rapidamente enganar o nosso agente ao atravessar um bloco não esperado.



Decisões de proximidade



Quando um inimigo é detectado num raio de 3 blocos da nossa posição atual, a árvore de decisões passa do modo A* para uma série de condições que levam em consideração a nossa posição e direção, a posição e direção dos inimigos detectados e o estado atual do mapa.

Estas decisões podem ser divididas em 3 categorias principais, dependendo da posição dos inimigos relativas ao agente.

Por cada inimigo dentro deste raio, o algoritmo deduz uma ação a realizar, todas estas ações são colocadas num Queue com prioridade, e a ação mais importante é realizada.



ALGORITMO A*

Funcionamento

Para a aproximação ao inimigo, o nosso agente utiliza um algoritmo A*, sendo este calculado a cada frame para apenas um inimigo.

O inimigo escolhido é o mais próximo dos inimigos “sólidos”. Caso não exista nenhum no mapa, será o inimigo em modo “ghost” mais próximo.

O algoritmo calcula o caminho ótimo, utilizando uma função que determina se um bloco é seguro ou não a cada movimento.

HEURÍSTICA

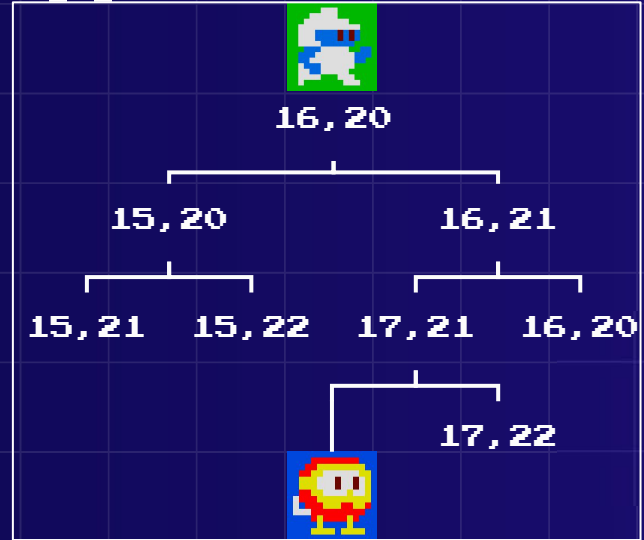
A heurística selecionada tenta sempre colocar o agente numa posição vertical ou horizontal em relação ao inimigo a seguir, evitando os perigosos caminhos diagonais e a repetição de padrões “esparguete”, em que o agente desperdiça frames a viajar de cima para baixo do mapa numa espécie de “S”. A heurística final é:

$$\text{abs}(x - xE) + \text{abs}(y - yE) + \max(\text{abs}(x - xE), \text{abs}(y - yE))$$

CUSTO

O cálculo do custo envolve promover o agente a viajar por caminhos já cavados, para não perfurar demasiado o mapa, pois isto providencia aos inimigos mais opções de fuga.

Este cálculo também tem em conta que posições mais baixas no mapa atribuem mais pontos às eliminações de inimigos, fazendo o agente preferir viajar por blocos nos “pisos” mais baixos possível.



DECISÕES DE PROXIMIDADE

Funcionamento

Após a aproximação com a pesquisa A*, o nosso algoritmo recorre a uma árvore de decisões que se pode dividir em três categorias, baseado na posição e proximidade do inimigo. Estas categorias estão representadas pelas diferentes cores no diagrama em baixo.

Para cada inimigo detectado neste raio, será calculada uma reação.

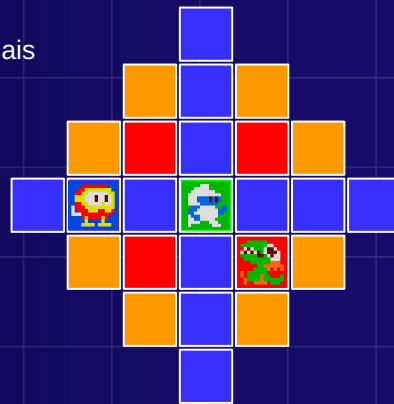
Todas as ações são colocadas numa queue com prioridade, sendo executada a ação vista como a mais importante.

As possíveis ações são:

- Esperar;
- Atacar;
- Movimentar para o angulo 0, 1, 2 ou 3.

As prioridades são (0 sendo a mais alta):

- 0: Fugir de um inimigo;
- 1: Atacar;
- 2: Evitar inimigo (fugir com menos prioridade);
- 3: Aproximar;
- 4: Mudar de direção;
- 5: Esperar.



Posições horizontais e verticais (azul)

Caso seja possível disparar e em segurança, ou seja, temos pelo menos um bloco entre nós e o inimigo ou o inimigo está "stunned", disparamos.

Caso não podemos disparar, tentamos mover para uma direção que faça isso possível e seguro.

Posições diagonais (vermelho)

Caso a nossa direção seja para onde achamos que o inimigo vai avançar e o bloco atual seja seguro, esperamos um turno.

Caso contrário, "preparamos" o mapa para fazer uma emboscada nos próximos turnos.

Posições diagonais em "L" (laranja)

Estas posições envolvem mais a predição do movimento do inimigo, para decidir se este vai entrar numa posição vermelha ou azul, ou se vai sair do nosso raio de 3 blocos.

Com esta previsão, podemos tentar seguir o inimigo ou preparar uma emboscada já com uma frame de avanço.

OUTROS BLOCOS PERIGOSOS

Todos os movimentos quer sejam pelo algoritmo A* ou pela árvore de seleção, primeiro garantem que o bloco onde estaremos na próxima frame é seguro.

Caso não seja, cabe à árvore de decisão decidir a próxima ação.

Foram então implementadas várias funções de segurança para tentar prever a próxima frame.

Um bloco é considerado perigoso/inacessível caso:

- Seja “out-of-bounds” (fora do mapa);
- Contenha uma pedra ou inimigo;
- Esteja dois blocos à frente de um inimigo ou um bloco atrás, pois estes podem mudar de direção e matar o agente;
- Um inimigo esteja previsto para avançar para o mesmo bloco na próxima frame (colisão);
- O bloco esteja a menos de quatro blocos de distância da frente de um inimigo do tipo “Fygar” ou três atrás, pois este pode matar o agente com o fogo.

TÓPICOS MANIAC MODE

O grande problema deste algoritmo é que em níveis mais altos, os inimigos tem a capacidade de evitar e esperar fora do alcance das emboscadas do nosso agente, levando a “deadlocks” e loops infinitos em que eventualmente o jogo acaba devido ao limite de frames máximo.

Para tal, foi criado um modo sequencial de “desespero”, ou como chamado no código, “Maniac Mode”.

Este modo tem como base dois fatores, o tipo de inimigo e o número de frames já passadas.

O modo em si é integrado de forma simples, após um certo número de frames, ações de segurança começam a ser ignoradas, como por exemplo o relaxamento das considerações de “blocos perigosos” explicados anteriormente, levando a que o agente fique progressivamente mais agressivo e a redução do raio de 3 blocos, causando ao agente a utilizar cada vez mais a pesquisa A*, ignorando tentativas de emboscadas.

Isto tem os seus óbvios problemas de instabilidade, mas é uma boa forma de obter mais pontuação em muitos casos. Com isto podemos focar-nos em fazer o agente em si mais estável (pelo menos 45 mil pontos em todas as runs), mas deixando este tentar obter muitos mais caso consiga.

TESTES

ENTREGA 1



Pontuação média em 30 jogos

— 90431 Pontos

Média final na avaliação

— 73380 Pontos

Performance esperada vs obtida

— 81%

ENTREGA 2

Pontuação média em 30 jogos

— 68524 Pontos

Performance face à entrega anterior

— 75%

Média na avaliação esperada (assumindo as mesmas seeds)

— 55603 Pontos

CONCLUSÕES

Concluindo, a discrepância entre o valor médio calculado na primeira entrega e o valor real provavelmente deve-se ao baixo tamanho de “sample-size” (apenas 10 runs) e ao potencial das “seeds” selecionadas serem mais difíceis de jogar para o nosso agente.

MÉTODOS DE TESTAGEM

- Todos os testes foram executados com recurso à base de dados providência na pasta /prof e com o game definido a 10.
- Os testes aqui apresentados foram executados sempre na mesma máquina;
- Resultados de outras máquinas foram obtidos, mas devido a serem extremamente parecidos, não achamos relevante apresentar a comparação;
- De modo parecido, a máquina utilizada foi colocada num estado de baixa potência (1.4Ghz de clock de CPU máximo, ao invés dos 4.3Ghz dos outros testes) e a diferença de pontuação média foi negligível, logo concluímos que o agente não deverá ter problemas de performance.



THANKS
FOR
PLAYING!

