

Relatório TPG – Inteligência Artificial

Jean Brito
82783

Estados do Pacman

O pacman assume 3 estados nesta respectiva ordem de prioridade:

– Runner:

- Caso algum fantasma não *zombie* se encontre a distancia *ghost_visibility* do pacman;
- Neste modo, o programa ira definir entre as posições disponiveis para mover-se, qual o deixará mais perto de um segundo objetivo e mais longe dos fantasmas usando a função *high_coast()*:
- Os segundos objetivos podem ser 3 casos nesta respectiva ordem de prioridade:
 - **Zombie ghost:** caso exista algum a uma distancia igual ou menor que o *time_zombie*;
 - **Boost:** caso algum se encontre a uma distância igual ou menor de 10 *moves*;
 - **Energy:** caso
(Sempre é escolhido o mais próximo heurísticamente)
- Neste modo, a variável *scape* é alterada para *True*.

– Hunter:

- Caso exista algum *zombie ghost* a uma distancia igual ou menor que o *time_zombie*;
- Neste modo o pacman irá traçar o seu objetivo, a posição do ghost mais próximo.

– Walker:

- Irá traçar como objetivo a *energy* mais próxima;
- O objetivo só é alterado, a posição anterior não se encontra no array *energy*

Funções

O pacman assume 3 estados nesta respectiva ordem de prioridade:

- **trace_router:**

- Função recursiva similar que retorna a melhor posição a tomar para um determinado objetivo, ou a key necessaria para tal.
- Foi implementado um sistema para prever o valor de retorno, e evitar pesquisas muito exaustivas, limitado a 100 ciclos, com um grau de acerto de 78%. Com este algoritmo foi testado alguns jogos, e verificado que mesmo este algoritmo sendo executado diversas vezes por ciclo e teve uma média de 0.000651 seg com máxima de 0.004085 seg.

- **high_coast:**

- Função que determina a melhor posição para mover-se caso tenha algum fantasma:
- Ela recebe a posição que os fantasmas a 8 *moves* poderiam chegar ao pacman, e a posição que deixaria o pacmna mais próximo do *second_objective*, e verifica se esta posição encontra-se nas previsões do fantasma, caso não se encontre, ela retorna como *goal*, a posição do *second_goal*;
- Se não se encontrar, ela apenas calcula qual posição deixará o pacman mais afastado dos fantasmas.

Definição da chave

Existem três situações em que a chave é definida, ela pode ser definido para um objetivo imediato ao pacman, pode ser definido a uma distancia maior, ou pode manter-se a mesma

- Primeiro é verificado se o pacman se encontra em modo *scape*, neste caso, o objetivo encontra-se imediatamente, e o movimento que levará o pacman a este objetivo é traçado;
- Caso o pacman não se encontre em modo *scape*, é verificado se existe um novo caminho possível em relação a posição anterior, e caso exista, é traçado uma nova *key*, se não for verificado, o pacman encontra-se em uma reta, portanto, mantém o curso;

Escolha dos Valores

Como o desempenho na partida pode depender de imensos fatores, e o código elaborado por mim tinha um alto número de variáveis específicas. Implementei um *ArgumentParser* para testar valores específicos em diferentes partes do projeto, passando o argumento por bash, verificando sempre qual trazia o melhor desempenho através de extensivos testes.

Também foi verificado para o pacman entrar em modo “***runner***” com diferentes distancias de acordo com o *level* dos *ghosts*, e não foi identificado melhora significativa para tal.