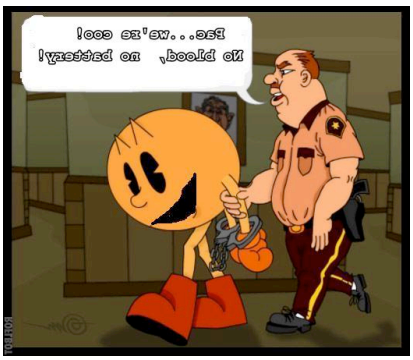
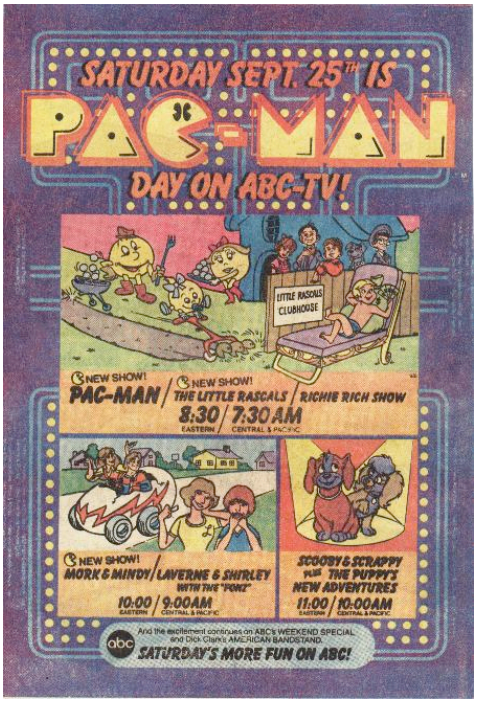


Sistemas Baseados em Conhecimento: 2013/14

O Pac-Man MinMax — 2º Projecto

Prazo: 27 de Maio de 2014 às 24h



O Pac-Man:

Pac-Man, que foi criado nos anos 80 por Tohru Iwatani, é um dos jogos de vídeo com mais sucesso. A mecânica do jogo é simples: o jogador é representado por uma cabeça redonda com uma boca que se abre e fecha, colocado num labirinto repleto de pastilhas mas guardado por 4 fantasmas que o perseguem. De um modo geral, o objetivo é comer todas as pastilhas sem ser alcançado pelos fantasmas, num ritmo progressivo de dificuldade.

Há certas pastilhas que dão super-poderes temporários ao PacMan e este pode comer os fantasmas, aumentado a sua pontuação. De vez em vez podem aparecer em lugares escolhidos aleatoriamente vários bónus com valores variáveis e que ao fim de um tempo, também este variável, desaparecem. Estes bónus poderão ser papados pelo Pac-Man contribuindo para a sua pontuação.

O Pac-Man está limitado a 3 vidas e perde uma vida sempre que é apanhado por um dos fantasmas e não esteja sob efeito das super-pastilhas (que duram 50 unidades de tempo). Em cada tentativa, o Pac-Man não pode ultrapassar as 800 unidades de tempo, perdendo uma vida caso isso aconteça.

Os labirintos, bidimensionais, são quadrados em que tantos as coordenadas nos eixos dos x e dos y variam de -10 a 10, em que as paredes são azuis e as células navegáveis estão a preto. As pastilhas são representadas por bolas brancas e são pequenas excepto as super-pastilhas que são bolas grandes. Os bónus são estrelas coloridas e os fantasmas terão várias formas dependendo do estado em que se encontrarem.

Vamos ter 5 labirintos diferentes e em cada labirinto teremos dois níveis de dificuldade: o Pac-Man move-se ao dobro da velocidade dos fantasmas no 1º nível e no 2º andam todos à mesma velocidade..

O jogador que controla o Pac-Man possui uma percepção total do jogo: da configuração do labirinto, da posição e direcção do Pac-Man, das posições das pastilhas por consumir, tanto as simples com as super-poderosas, as posições dos fantasmas; sabe também se os fantasmas estão com medo devido ao efeito das super-pastilhas e em caso afirmativo quanto tempo mais dura esse poder; sabe também onde estão os bónus, por quanto tempo mais e qual o valor de cada um deles. Para além disso sabe em que instante vai o jogo, o limite de tempo, o número de vidas do Pac-Man, o nível de dificuldade, o score.

Os fantasmas quando são comidos regressam à base onde está o antídoto. Quando estão no estado normal perseguem o fantasma se o virem numa direcção que não seja a oposta à que estão a tomar, senão escolhem ao acaso uma das direcções possíveis excepto a oposta. Quando estão com medo porque o Pac-Man está com super-poderes, tentam fugir do Pac-Man quando o veêm, para qualquer uma, escolhida ao acaso, das direcções diferentes da que se encontra o Pac-Man, Se não virem o fantasma andam ao acaso mas nunca para trás.

Em cada instante o Pac-Man dá um passo, excepto se está orientado para alguma parede e é preciso decidir apenas em que direcção será esse passo dado: norte (0), leste (90), sul (180) ou oeste (270).

O objectivo do jogador que controla o Pac-Man é ter o maior score possível, cumprindo todos os labirintos. O facto de manter todas as vidas ou sacrificar alguma delas não é relevante, apenas ter o maior score possível.

Pontuação:

- Pastilha simples: 100
- Super-pastilha: 500
- Fantasma: 500
- Bónus: entre 100 e 1000 pontos
- Cada tick dá 2 pontos (durar é bom)

O que é pedido:

É necessário desenvolver um programa em PROLOG que decida o que o Pac-Man tem que fazer a cada momento (1 tic do relógio), sendo o interface controlado pelo modelo Pac-Man do Netlogo, usando para isso a extensão NETPROLOGO. Na verdade, é pedido um predicado *pacman/16* (16 argumentos) que é invocado a partir do Netlogo sempre que o Pac-Man tem de decidir orientar-se. Esse predicado recebe informação sobre o relógio, o tempo limite, a pontuação corrente, o

número do labirinto, o nível de dificuldade, o número de vidas extra, a posição da rã, a sua orientação, as células livres do labirinto, os fantasmas, a zona onde os fantasmas ressuscitam, o nível de medo dos fantasmas, as posições das pastilhas simples e das super, e finalmente informação sobre os bónus, nos 15 primeiros argumentos e devolvem o código numérico da acção no décimo sexto e último argumento. Vão usar o algoritmo minmax com um limite de 2 níveis, tendo que representar o jogo em termos de estados e das operações sobre os estados (jogadas) de modo a poder gerar a árvore usada pelo minimax. É necessário também implementar uma funçãoo de avaliação para as folhas da árvore.

Queremos preparar o jogo do PacMan para ser utilizado por uma versão simplificada do algoritmo Minimax, limitando a pesquisa a 2 níveis: a jogada do *Pacman* e a seguir a dos *Fantasmas*. Para isso é preciso modelizar o jogo em termos de estados e de jogadas que modificam os estados.

compoeEstado/16 que recebe toda a info do Netlogo (15 argumentos) e que constrói um estado.

jogadaPacman/3 que recebe um estado e devolve um novo estado resultante da jogada do Pacman, e a etiqueta da jogada: por exemplo pacman-(2,4), indicando que o Pacman se moveu para a célula (2,4) do labirinto do jogo.

jogadaFantasmas/3 que recebe um estado e devolve um novo estado resultante da jogada dos Fantasmas, e a etiqueta da jogada: poderia ser algo como [(2,5)->(2,6),(6,7)->(7,7),(9,3)->(9,2),(5,5)->(4,5)], indicando o movimento executado por cada um dos fantasmas.

minmax/3 que recebe um estado e que executa o minmax até nível 2 devolvendo a melhor jogada, o estado que resulta e o valor associado.

pacman/16 que recebe 15 argumentos do Netlogo e devolve a melhor jogada do Pacman: 0, 90, 180 ou 270.

avaliação/3: quando o minmax chega a uma folha é preciso avaliar o estado do jogo de acordo com um jogador: pacman ou fantasmas. Neste caso, só será preciso avaliar o jogo do ponto de vista do pacman, mas podem fazer um predicado geral. O primeiro argumento é o jogador, o segundo o estado e o terceiro o valor.

Notem que o Pacman no nível de menor dificuldade faz dois movimentos e portanto a decisão corresponde a uma antecipação de 2 tics. No nível de maior dificuldade, dois níveis de minimax corresponde a apenas um tic porque nesse tic jogará o pacman e os quatro fantasmas. Um fantasma morto passa a vivo sempre que tiver como vizinha a norte uma das células da zona.

Representação da informação vinda do Netlogo

O NetPrologo passa em cada instante ao Prolog as informações relevantes do jogo:

Se quiserem ver os predicados *pacman* quando são passados ao Prolog basta, no interface, colocarem a variável booleana Netlogo *show-prolog* a true.

Os argumentos são os seguintes, por esta ordem:

1. Relógio: o número de tics decorrido.

2. Limite de tempo: o limite de tempo para cada vida
3. Pontuação: a pontuação corrente.
4. Nível: qual o labirinto: 1, 2, 3, 4 ou 5.
5. Dificuldade: o tipo de dificuldade (1 ou 2). 1 significa que os fantasmas se movem de 2 em 2 tics e 2 que eles se movem em cada tic como o Pac-Man.
6. Vidas extra: o número de vidas que faltam.
7. Posição do Pac-Man: Um par com a posição do Pac-Man (xcor,ycor).
8. Orientação do Pac-Man: a orientação corrente do Pac-Man (0, 90, 180 or 270)..
9. Labirinto: uma lista com todas as posições das casas livres do labirinto corrente: uma lista de pares (x,y).
10. Fantasmas: uma lista de info dos fantasmas: cada fantasma é dado por ((x,y),comido?). o elemento comido? Está a true quando o fantasma foi comido e ainda não regressou à base para sacar o antídoto.
11. Medo: quando mais tics faltam ainda para que os fantasmas deixem de ter medo. No fundo, este parâmetro refere-se ao tempo que resta do efeito das super-pastilhas.
12. Pastilhas: uma lista com as posições de todas as pastilhas ainda não comidas, uma lista de pares (x,y).
13. Super-Pastilhas: uma lista com as posições de todas as super-pastilhas ainda não comidas, uma lista de pares (x,y).
14. Bónus: uma lista de triplos com a info de cada bónus. Cada triplo ((x,y), valor, duracao) que representa a posição do bónus, o seu valor e o tempo de vida.
15. Células da zona: lista de células (x,y), que quando são atravessadas, os fantasmas passam de mortos a vivos. (Se um fantasma tiver uma delas como vizinha a norte…)
16. Output – a acção escolhida pelo Pac-Man: {0, 90, 180, 270}

O que cada grupo de trabalho deve entregar:

Entregar: Um único ficheiro contendo o programa e um relatório sucinto (em PDF), que identifique os elementos do grupo e:

- Que descreva a forma como o Pac-Man seleciona as suas direcções;
- Inclua, como anexo, todo o código do programa.

O ficheiro deve ser submetido através da página do mocho da cadeira, por um só dos elementos do grupo, e ter como nome pacman-XX.pl, onde XX deve ser substituído pelo número do grupo.

Extensões do Netlogo a usar:

- Sites do NetProLogo: http://www.cs.us.es/~fsancho/NetProLogo/
- Sites do pathdir: http://sophia.smith.edu/~cstaelin/NetLogo/pathdir.html

Têm de colocar as pastas das extensões no interior da pasta extensions da pasta da aplicação Netlogo. O ficheiro .jar do Netprologo deve chamar-se **netprologo.jar** e o ficheiro .jar do pathdir deve chamar-se **pathdir.jar**. Têm de ser renomeados para os nomes indicados.