como utilizar

* Para executar o jogo:

Windows: Abrir app\win32\pacman.exe

Linux: Abrir terminal na pasta app/linux e executar o comando ./pacman

* Para ver o código fonte JAVA:   
  Abrir pacman\build\source\pacman.java
* Para ver o código fonte na IDE do Processing:

Abrir pacman\pacman.pde

Para executar o ficheiro pacman.pde na IDE do Processing é necessário adicionar a biblioteca *Minim*.

DESCRição

**Pacman Multijogador**

* Maria João Lavoura  
  N. Mec. 84681
* Pedro Teixeira   
  N. Mec. 84715

Programação I | DETI-UA

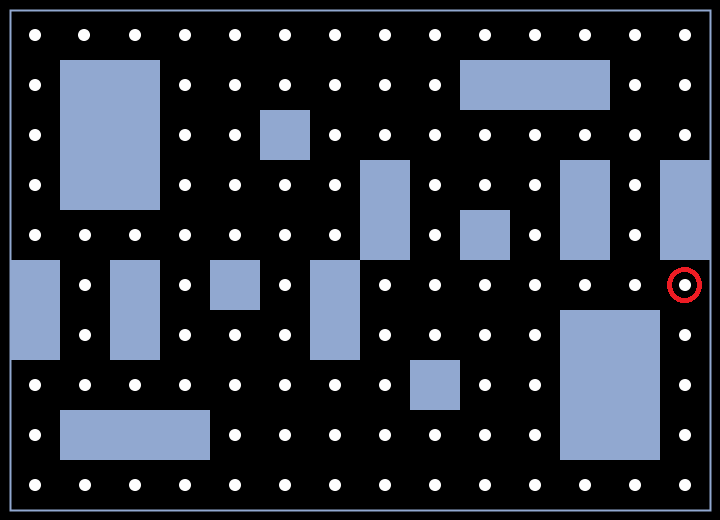
Trabalho Prático

13-12-2016

* Neste trabalho foi implementado um jogo Pacman com 2 modos de jogo – 1 Jogador e 2 Jogadores. Estes modos são escolhidos pelo utilizador no menu inicial.
* No modo 1 Jogador (*Single Player,* tecla 1), o utilizador controla o Pacman num labirinto onde estão presentes 4 fantasmas que se movem sem *input* do utilizador.
* No modo 2 Jogadores (*Multiplayer,* tecla 2), são necessários 2 utilizadores, sendo que um deles controla o Pacman e o outro controla o único fantasma presente.
* O jogo termina quando:
  + Pacman colide com um fantasma;
  + Pacman elimina todos os pontos;
  + Pacman elimina um ponto especial.

O ponto especial é um ponto que aparece apenas quando são eliminados mais de ½ e menos de ¾ dos pontos iniciais (ou seja, aparece por pouco tempo) e que permite ao Pacman eliminar os fantasmas e ganhar o jogo.

A localização é a mesma para os 2 modos de jogo:



* Sendo que nas 2 últimas situações, o utilizador que controla o Pacman ganha, enquanto que na 1ª perde.
* Quando o jogo termina, é apresentada a pontuação obtida, que corresponde ao número de pontos eliminados pelo Pacman até àquele instante. Isto significa que não será possível obter pontuação máxima ao ganhar o jogo através do ponto especial.
* Após terminar o jogo, é possível consultar as 8 pontuações máximas (no menu, tecla 3) para cada modo de jogo. Este valor 8 pode ser facilmente alterado.

As pontuações também podem ser consultadas nos ficheiros   
pacman\data\ scores\_singleplayer.txt e   
pacman\data\ scores\_multiplayer.txt.

* Em qualquer momento é possível consultar o menu de ajuda (através da tecla H), onde estão descritos quer os controlos do Pacman e fantasmas quer funcionalidades extras:
  + Ativar/Desativar Som (tecla M);
  + Definir cor do labirinto aleatoriamente (tecla C). O labirinto pode tomar qualquer cor excepto preto e cores escuras para garantir a sua visibilidade;
  + Pausar o jogo (tecla P);
  + Reiniciar o jogo (tecla ESC);

Principais OBjectivos e como foram alcançados

1. Implementar um Pacman com 2 modos de jogo (*Single Player*/*Multiplayer*) e
2. Interagir com o utilizador, através de menus.

* Criação de uma variável gamestate (uma *flag*) que, consoante o seu valor, executa uma função correspondente ao estado de jogo:
  + Mostrar o menu inicial (gamestate=0);
  + Começar o jogo *Single Player*/*Multiplayer* (gamestate=1/gamestate=2);
  + Mostrar pontuações (gamestate=3);
  + Mostrar menu de ajuda (gamestate=4);
  + Terminar o jogo (gamestate=5).

1. Detectar colisões (do Pacman com fantasmas e com pontos)

* Através da comparação de posições: se as posições forem iguais, existe colisão.
* Através do cálculo de distâncias entre 2 pontos correspondentes às posições do Pacman e dos diferentes fantasmas.
  + A colisão do Pacman com fantasmas dá-se quando a distância entre estes é muito pequena (neste jogo quando distância<10).

1. Impedir o Pacman de transpor obstáculos

* Após pressionar a tecla W/A/S/D, o jogo obtém a cor da célula correspondente à posição para a qual o Pacman vai deslocar-se.
* Se a cor dessa célula for diferente da cor dos obstáculos, o Pacman desloca-se para a nova posição. Caso contrário, fica na sua posição original.

1. Rodar o Pacman consoante a direcção e sentido do seu movimento

* Após pressionar a tecla ↑/↓/←/→, o Pacman é redesenhado de forma a ter a boca apontada na direcção e sentido do seu movimento.
* Após pressionar uma das teclas são devolvidos 2 valores de ângulos, através de 2 funções (rotatePacmanStart e rotatePacmanStop), valores esses que vão definir, na função que desenha o Pacman, onde começa e onde termina o desenho deste.

1. Reproduzir sons aquando da ocorrência de eventos – início de jogo, comer ponto, fim de jogo

* Importação da biblioteca *Minim*.
* Criação uma *array* de objectos *AudioPlayer* com os 3 sons utilizados.
* Na função correspondente ao evento, utilização de .play() e .stop()do som a reproduzir e dos sons a não reproduzir, respectivamente (o que evita uma sobreposição indesejada de sons).

1. Utilizar *arrays* bidimensionais para criar um mapa do tabuleiro do jogo

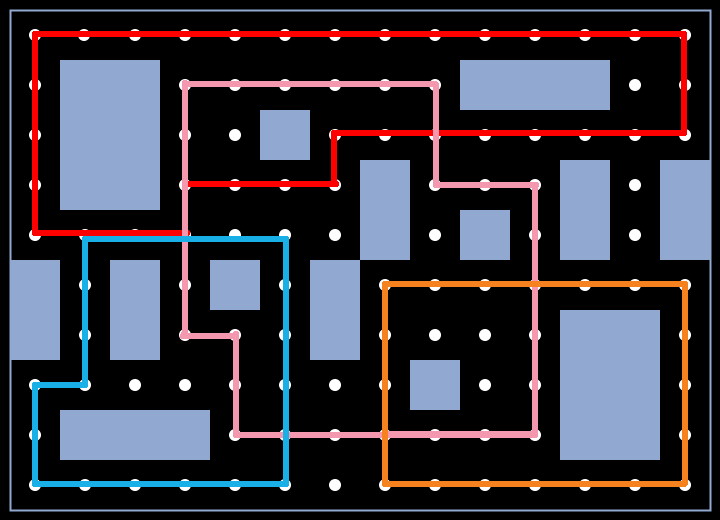
* Criação de uma *array*, comida[i][j], sendo i e j as dimensões do labirinto.
* A *array* tem valores de “1” se a célula do mapa tem comida/bola (ie ponto) e valores de “2” se a célula do mapa não tem ponto. Estes valores vão permitir que não seja desenhado um ponto nos obstáculos e em sítios por onde o Pacman tenha passado.

1. Implementar fantasmas como uma classe

* Criação da classe Ghost com as posições, velocidades, módulo da velocidade e imagens de cada fantasma.
* A criação da classe Ghost evita a criação de 6 variáveis *float* e de 1 *array* de imagens por fantasma, bastando a criação de 4 objectos do tipo Ghost.

1. Criar um movimento aleatório (ie. sem *input* do utilizador) para os fantasmas

* Criação de um percurso predefinido para cada fantasma:



* Para cada fantasma, quando o fantasma se encontra numa posição onde muda de direcção e/ou sentido, são alteradas as velocidades segundo *x* e segundo *y.*
* Por exemplo

else if ((red.px==centroX(7)) && (red.py==centroY(3))) {

red.vx=0;

red.vy=red.set\_vy;

}

* significa que o fantasma vermelho, quando está na célula (7,3), desce para baixo.

10. Pontuações: contar pontos

* Criação de 2 funções, para ler e escrever em ficheiros. Foram adaptadas funções genéricas para leitura e escrita de ficheiros que criamos ao longo do guião 09.
* Estas funções recebem um argumento que decide qual o ficheiro a utilizar: o ficheiro das pontuações do modo *Single Player* ou o do modo *Multiplayer*.
* No caso da leitura das pontuações, como se quer apenas algumas das maiores pontuações, os valores do ficheiro são lidos para uma *array* temporária, que é ordenada (ordenação por flutuação). Depois é criada uma nova *array*, para onde são copiados apenas o número de pontuações pretendidas.
* No caso da escrita de pontuações, é utilizado o FileWriter, para evitar que as pontuações antigas que estejam nos ficheiros sejam apagadas.

Agradecimentos

* Agradecemos aos nossos colegas Beatriz Borges e Rui Lopes pela ajuda dada.

Referências

* Sons utilizados provenientes de  
  <http://www.classicgaming.cc/classics/pac-man/sounds>
* Imagens utilizadas provenientes de   
  <http://pacman.com/en/pac-man-news/pac-man-free-downloads/pac-man-wallpaper-classic-ghosts>   
  <http://molotov-arts.deviantart.com/art/Pac-Man-113138504>   
  <http://www.deviantart.com/art/CBS-PAC-MAN-369019724>  
  http://pacman.wikia.com/wiki/File:Pacman\_logo.gif [[1]](#footnote-1)

1. Ao abrigo de <https://www.copyright.gov/fair-use/> [↑](#footnote-ref-1)