

Projeto e Implementação do Jogo Pac-Man

Este trabalho consiste na implementação de uma versão simplificada do famoso jogo Pac-Man. O trabalho pode ser realizado em grupo de até 3 alunos e deve ser feito com a biblioteca SFML. Um tutorial dessa biblioteca, apresentado na aula prática extra, está disponível no PVANet, juntamente com alguns exemplos de código, incluindo um arquivo `pacman.cpp` contendo parte do jogo.

PACMAN

A Figura 1 mostra imagens do jogo original do PacMan. A figura da esquerda é a tela inicial do jogo, a do meio logo após iniciar o jogo e a da direita pouco tempo depois. O PacMan é o personagem amarelo que aparece na parte de baixo do labirinto, inicialmente um pouco abaixo do texto “Ready!” na imagem da esquerda. O objetivo do PacMan é “comer” todas as pílulas (pontos laranja claro) encontradas no labirinto. Ele come uma pílula ao encostar nela, à medida que se move pelo labirinto, conforme ilustrado nas imagens do meio e da direita. Durante seu percurso, o PacMan deve evitar contato com os fantasmas (personagens coloridos posicionados inicialmente no meio da tela), pois ele morre ao tocar em um fantasma.



Figura 1: Telas do PacMan original

As figuras mostram ainda várias outras características do jogo. Neste trabalho deve ser implementada uma versão simplificada, com as seguintes características:

1. o jogo termina quando não há mais pílulas no labirinto (não existem várias fases) ou quando o PacMan é atacado por um fantasma (não há novas chances, nem várias vidas);
2. não existem as “frutas”, que aparecem de tempos em tempos no centro labirinto e valem ponto extra quando comidas pelo PacMan;
3. não existem as pílulas de força que, quando comidas pelo PacMan, tornam os fantasmas vulneráveis;
4. não existem os “túneis” que permitem sair de um lado da tela e aparecer no outro;
5. existe ainda um placar que mostra a pontuação no jogo, considerando os objetos devorados pelo PacMan (pílula, frutas e fantasmas).

Entretanto, 15% da nota do trabalho será pela implementação de uma dessas funcionalidades mencionadas não existentes. O grupo pode escolher qual(is) delas implementar, mas deve entrar em contato caso queira implementar uma funcionalidade diferente no lugar delas. Pode também acrescentar outras características no jogo para torná-lo mais interessante e mudar o tema do jogo, definindo outros personagens para o papel do PacMan e dos fantasmas.

Para facilitar a implementação, o trabalho é separado em três partes, descritas a seguir, mas deve ser entregue apenas no final, quando as três partes estiverem prontas. Entretanto, recomenda-se fortemente terminar por completo uma das partes e marcar um horário com o professor para apresentá-la antes de iniciar a próxima.

PARTE 1: mapa e personagens

Esta parte consiste em definir o mapa que representará o labirinto do jogo. O programa deverá conter uma definição para o mapa em forma de uma matriz de 2 dimensões. Códigos numéricos devem ser associados aos diferentes tipos de elementos que podem aparecer no mapa. Cada célula da matriz armazenará um desses códigos. Por exemplo, podem ser definidos os seguintes códigos:

- 0 - células vazias
- 1 - paredes
- 2 - pílulas

Ao ser executado, o programa deverá exibir uma representação gráfica para a matriz definida, que deverá formar um desenho com o mapa do jogo. A Figura 2 mostra um exemplo, resultado da compilação e execução do programa `pacman.cpp` disponível no PVANet. Ele define uma matriz para o mapa no próprio código e desenha quadrados azuis (paredes) onde há valor 1 nessa matriz.

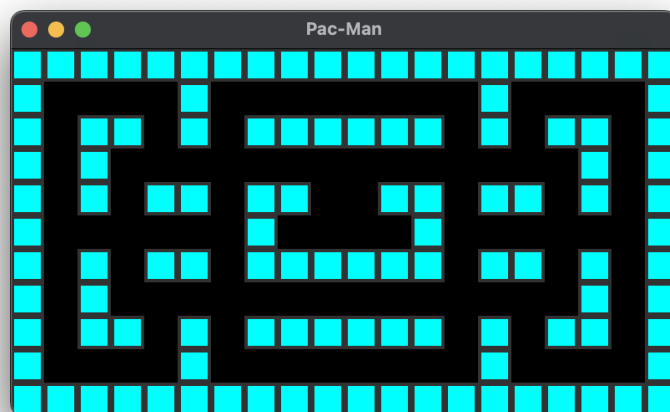


Figura 2: Mapa usado no programa `pacman.cpp`

As figuras utilizadas em seu jogo podem ser diferentes, e seria legal que fossem. O labirinto pode ter um formato diferente, desde que tenha um bom número de pílulas e ofereça alguma dificuldade. O labirinto original, por exemplo, é melhor que o mostrado acima. Em vez de desenhar o labirinto usando a matriz que representa suas paredes, seu código pode também carregar uma imagem de fundo com as paredes pré-desenhadas nas posições correspondentes.

Nesta parte, as células vazias do mapa devem ser preenchidas com pílulas. Também devem ser mostrados os personagens do jogo: o PacMan deve aparecer em algum local próximo ao centro e os 4 fantasmas devem ser adicionados, um em cada canto do mapa, ou na “casa” deles, se houver um local para isso (no mapa acima, a casa dos fantasmas está no centro do mapa).

PARTE 2: movimento do Pacman e pontuação

Esta parte consiste na definição do comportamento do PacMan e na contagem de pontos. Parte do código para o movimento do PacMan é fornecida nos arquivos `pacman.cpp` e `pacman2.cpp`. No primeiro, ele se move a cada vez que uma tecla direcional é pressionada. No segundo, ele continua se movendo mesmo quando se solta a tecla, gerando um resultado mais interessante para o jogo. Nesse último caso, o movimento não ocorre em toda iteração do loop principal, mas apenas após um intervalo de tempo. Sem isso, os movimentos seriam rápidos demais para serem controlados.

Ao ser executado, o programa deverá exibir o mapa com as pílulas e os personagens construído na Parte 1. Deve exibir também um placar indicando o número de pontos obtidos, que equivale ao número de pílulas coletadas pelo PacMan. As regras de movimentação e contagem de pontos são:

1. quando o PacMan passar sobre uma pílula, ela deve ser apagada e o placar deve aumentar;
2. se não houver mais nenhuma pílula no labirinto, o jogo é encerrado (vitória);
3. a movimentação do PacMan deve ser controlada pelas teclas direcionais (“setas” do teclado).
4. a direção de seu movimento assume umas das possibilidades: PARA CIMA, PARA BAIXO, PARA ESQUERDA, PARA DIREITA.

A direção de movimento indica também o estado do PacMan, definindo a figura a ser exibida. Isto é, se o estado for PARA ESQUERDA, então o desenho a ser exibido na tela deverá ser do PacMan “olhando” para a esquerda. Cada estado de movimento terá um desenho distinto. A Figura 3 ilustra o PacMan no estado PARA ESQUERDA e PARA BAIXO.

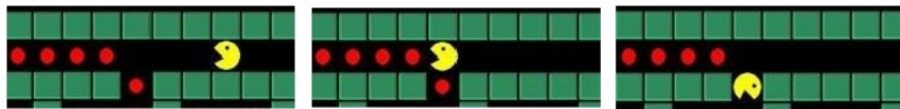


Figura 3: Exemplo de movimento e estado do PacMan

Note que o movimento do PacMan feito pelo programa `pacman.cpp` não é muito parecido com o do jogo original. Ao soltar uma tecla direcional, ele simplesmente para de se movimentar. No jogo original, mesmo soltando a tecla, ele continuaria se movimentando na mesma direção até chegar em uma parede, ou mudaria de direção, quando possível, se o jogador pressionasse outra tecla. Este movimento, que é bem mais interessante e torna o jogo mais divertido de se jogar, é parcialmente usado no programa `pacman2.cpp`, em que ele continua se movendo. A ideia completa envolve:

1. o Pacman deve possuir um **estado** de movimento (em que direção está se movimentando) e uma **intenção** de movimento (para qual direção o jogador quer que ele se movimente);
2. ao pressionar uma tecla direcional, o jogador muda a **intenção** de movimento;
3. se as condições atuais permitem alterar o estado de movimento para a intenção definida pelo jogador, ou seja, se não há parede adjacente na direção escolhida, ele altera seu estado;
4. senão, ele continua se movimentando na direção em que estava, ou seja, de acordo com seu estado; mas o sistema armazena a última intenção do jogador até que ela possa ser concretizada (ou que o jogador altere essa intenção, pressionando outra tecla).

Suponha, por exemplo, que na situação mostrada na Figura 3a, o jogador pressione a tecla PARA BAIXO. O sistema registra esta intenção, podendo o jogador até mesmo soltar a tecla. Mas, como na condição atual o PacMan não pode se movimentar PARA BAIXO, ele continua se movimentando PARA ESQUERDA. Suponha que o jogador não pressione nenhuma tecla até que

o PacMan chegue na posição mostrada na Figura 3b. O sistema continua com a intenção de movimento PARA BAIXO e nesse momento é possível realizar o movimento desejado. Então, o estado é alterado e ele passa a se movimentar PARA BAIXO, como na Figura 3c, mesmo que o jogador não esteja pressionando a tecla correspondente neste momento. Assim, o jogador não precisa pressionar na hora exata do movimento. A intenção continua sendo PARA BAIXO até que o jogador pressione outra tecla direcional. Esta parte de controle de movimento por intenção vale 15% da nota.

PARTE 3: movimento dos fantasmas (jogo completo)

Esta parte consiste da definição do comportamento dos fantasmas, completando assim a implementação do jogo.

O jogo inicia com 4 fantasmas devidamente posicionados nos cantos do mapa (ou no centro, se houver um local apropriado para isso). Cada fantasma deve ter uma cor diferente (ou personagens diferentes, dependendo do tema de seu jogo). Os fantasmas se movem no labirinto e são os inimigos do PacMan. O PacMan morre quando um fantasma ocupar a mesma célula que ele.

O movimento dos fantasmas pode ser de 2 tipos: aleatório e de perseguição. 15% da nota consiste em haver pelo menos um fantasma aleatório, e outros 15% por pelo menos um de perseguição.

O movimento de perseguição é caracterizado por deslocamentos no sentido de matar o PacMan. Diversas estratégias podem ser seguidas, e cada grupo poderá implementar a estratégia que desejar. A avaliação levará em conta a qualidade da estratégia e sua documentação (texto explicando os mecanismos utilizados).

O movimento aleatório é caracterizado pelas seguintes regras:

1. A direção de movimento é independente para cada fantasma e poderá ser: PARA CIMA, PARA BAIXO, PARA ESQUERDA, PARA DIREITA.
2. O fantasma está sempre em movimento e sua direção pode mudar quando se deparar com uma **parede** (assim como o PacMan, eles não atravessam paredes) ou uma **encruzilhada** (ponto com mais de duas possibilidades de movimento)
3. Ao encontrar uma parede ou encruzilhada, é selecionada uma nova direção de movimento, escolhida aleatoriamente entre as direções válidas no momento. Uma direção é válida se a próxima célula a ser ocupada pelo fantasma não for uma parede.

A Figura 4 ilustra o comportamento numa encruzilhada. Suponha que o fantasma esteja com a direção de movimento PARA BAIXO. Ao se movimentar três células para baixo, ele encontra uma encruzilhada (marcada com um círculo). Nesta célula há três movimentos possíveis: PARA CIMA, PARA BAIXO e PARA ESQUERDA. Ao chegar à encruzilhada o fantasma deve sortear uma destas três direções como sua nova direção de movimento (melhor ainda se ele não escolher PARA CIMA, ou seja, não inverter o sentido do movimento atual). De forma semelhante, o fantasma deve selecionar aleatoriamente uma direção a ser tomada ao se encontrar com uma parede.

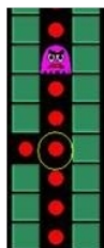


Figura 4: Exemplo de movimento do fantasma

O QUE E COMO ENTREGAR

1. Apresente o trabalho nas datas estipuladas (formulário disponível no PVANet)
(note que isto é antes do envio final do trabalho, apresentará o estágio atual de desenvolvimento)
2. Prepare um arquivo ZIP com os arquivos necessários para a compilação e execução do jogo:
 - código fonte (arquivos .cpp desenvolvidos devidamente comentados e indentados)
 - imagens usadas no jogo
 - demais arquivos usados não disponíveis no PVANet (como arquivos de áudio, por exemplo)
 - **NÃO** enviar o executável!
 - **NÃO** enviar DLLs, lib's, e outros arquivos da biblioteca SFML
3. Prepare ainda um relatório (texto ou slides) em formato PDF contendo:
 - manual de utilização do jogo, caso não seja simplesmente utilizar as setas direcionais
 - descrição sucinta dos métodos utilizados para movimentação dos fantasmas
 - descrição de características adicionais acrescentadas no jogo
 - identificação de quem foi responsável por cada parte/código do trabalho
4. Prepare ainda um ZIP com figuras para divulgação do trabalho:
 - Prints e/ou GIFs animados das partes que considera mais legais do seu jogo
5. Envie os arquivos ZIP e o PDF no PVANet Moodle.

*Obs.: Devido ao atraso no dia da aula extra, pode entregar **sem uma** das partes marcadas com “15% da nota”; a entrega de todas funcionando corretamente será considerada como ponto extra.*