

Projeto Final Pacman utilizando a Biblioteca de Threads Desenvolvida

1. Descrição

O projeto consiste em utilizar a biblioteca de Threads no espaço de usuário desenvolvida durante o semestre para implementar o jogo do Pacman.

Um exemplo do jogo do Pacman pode ser encontrado no seguinte vídeo:
<https://www.youtube.com/watch?v=AuoH0vz3Mqk>

O jogo consiste no Pacman fugindo de 4 fantasmas. Se algum dos fantasmas encostar no Pacman, ele morre (tem 3 vidas antes de acabar o jogo). Ao capturar uma das 4 “frutas” disponíveis na tela, o Pacman cria super poderes para comer os fantasmas. Quando um fantasma é comido, ele deve permanecer na “cadeia” por um tempo entre 1 e 7 segundos (aleatório). Ao capturar um fantasma o Pacman ganha 200 pontos. O tabuleiro do Pacman consiste também de círculos amarelos. Toda vez que o Pacman passar por um dos círculos ele ganha 10 pontos. De vez em quando aparece uma cereja no centro do tabuleiro (abaixo da cadeia). Ao comer a cereja o Pacman ganha 100 pontos. O tabuleiro também possui portais nas laterais para permitir a passagem do pacman e fantasmas de um lado ao outro.

O seguinte site descreve o comportamento dos fantasmas no jogo:
<https://gameinternals.com/understanding-pac-man-ghost-behavior>. Se o grupo decidir implementar o comportamento único para cada fantasma terá um bônus de 1 ponto na nota.

O projeto deve implementar o Pacman utilizando a biblioteca de Threads no espaço de usuário desenvolvida durante o semestre. Neste sentido, **cada fantasma deve ser uma Thread, o Pacman deve ser outra Thread, assim como uma Thread responsável pelo desenho da tela e outra para o tratamento da entrada do teclado (mínimo de 7 Threads)**. Os alunos têm a liberdade para criar outras Threads como acharem necessário.

As teclas de entrada são as seguintes:

- Teclas para cima, para baixo, lado esquerdo e lado direito para controlar o Pacman.
- P para pausar e despausar o jogo.
- Q para sair do jogo.
- Pontuação Bônus de 0.5 ponto: R para reiniciar o jogo quando acabar.

A biblioteca gráfica utilizada será a SFML (<https://www.sfm1-dev.org/>). Para instalar a biblioteca no Ubuntu digite: **sudo apt-get install libsfm1-dev**. A documentação da biblioteca encontra-se em <https://www.sfm1-dev.org/learn.php>.

Cada grupo terá a liberdade para desenvolver o software seguindo o paradigma de orientação a objetos do C++ obrigatoriamente. Isso significa que o uso de programação estruturada e elementos da linguagem C terão descontos na nota (veja avaliação abaixo).

Dica: vocês podem manter os elementos da tela (como posição do pacman, frutas, pontos, fantasmas, etc) em uma matriz que será compartilhada pelas Threads e atualizada a cada “tick” do jogo. Lembrem-se de proteger a matriz e qualquer outra estrutura de dados compartilhada com Semáforos. Não utilizem variáveis globais, utilizem classes e atributos estáticos.

Por fim, o grupo que testar o comportamento final do jogo (ou seja funcionando) com o escalonamento preemptivo por tempo e o escalonamento cooperador (usando yield+dispatcher) terá um bônus de 1 ponto na nota.

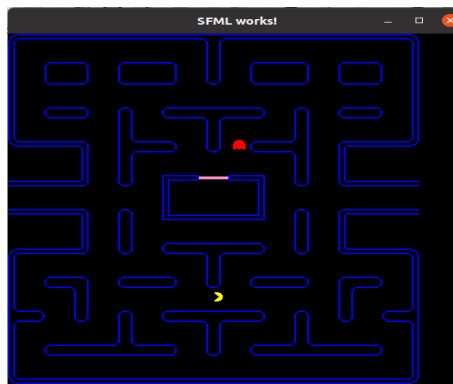
2. Arquivos Disponibilizados

Foram disponibilizados no moodle arquivos para criação de uma janela e desenho das estruturas do jogo na tela utilizando a biblioteca SFML. Dentro do diretório “sprites” são disponibilizados todos os arquivos para desenhar na tela. Por exemplo, dentro de sprites/pacman, existem 3 arquivos “0.png”, “1.png”, e “2.png” que demonstram a abertura e fechamento da boca do pacman.

Os desenhos foram retirados de <https://github.com/tkilminster/pacman>. Esse repositório pode servir para retirar ideias (**não copiar código**) de como implementar, embora utilize linguagem C e outra biblioteca gráfica.

O exemplo disponibilizado também já mostra o tratamento das teclas pela biblioteca:

```
./main  
Keyboard para baixo!  
Keyboard para cima!  
Keyboard direita!  
Keyboard esquerda!  
Keyboard esquerda!
```



3. Formato de Entrega

Todos os arquivos utilizados na implementação do trabalho devem ser entregues em um único arquivo .zip ou .tar.gz na atividade do moodle. Deve ser anexado um arquivo Makefile para compilar o código e um relatório descrevendo o projeto do software (diagramas UML).

4. Data de Entrega

Data e horário da entrega estipulados na tarefa do moodle.

5. Avaliação

A avaliação se dará em 3 fases

1. Avaliação de compilação: compilar o código enviado. Caso haja erros de compilação, a nota do trabalho será automaticamente zerada.
2. Avaliação de execução: para validar que a solução executa corretamente sem falhas de segmentação. Caso haja falhas de segmentação, a nota é zerada. Será também avaliado o uso de variáveis globais (-5 pontos) e vazamentos de memória (-20%). **Variáveis estáticas de classes não são consideradas globais.**
3. Avaliação da organização do código: busca-se nesta fase avaliar a organização do código orientado a objetos. Deve-se usar classes e objetos e não estilo de programação baseado em procedimentos (como na linguagem C). Uso de programação estruturada (-30%). Uso da linguagem C ao invés de linguagem C++ (-30%)

Este trabalho precisará ser apresentado ao professor em horário a ser agendado. Plágio não será tolerado em nenhuma hipótese ao longo dos trabalhos, acarretando em nota 0 a todos os envolvidos.

6. Tempo Estimado

Estima-se um tempo total necessário para a conclusão desta atividade de 16 horas.