

# Projeto Pac-Man – parte 1

Programação Orientada a Objetos

Prof. Robson L. F. Cordeiro

19 de outubro de 2021

## 1 Descrição

O projeto a ser desenvolvido nessa disciplina será a implementação do jogo *Pac-Man*<sup>1</sup>. A figura a seguir apresenta um exemplo da tela deste jogo.

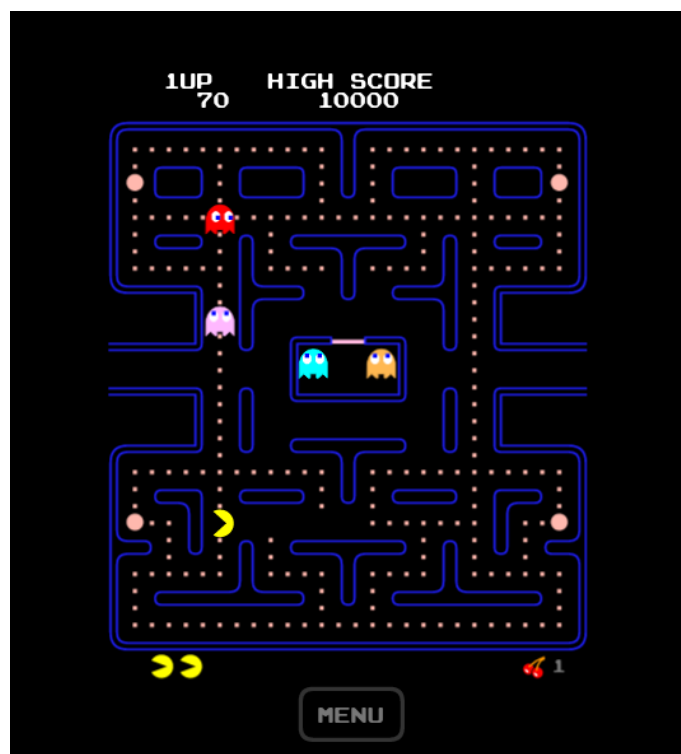


Figura 1: Tela do jogo *Pac-Man*.

---

<sup>1</sup><https://en.wikipedia.org/wiki/Pac-Man>

Regras detalhadas de funcionamento do jogo podem ser facilmente encontradas na Internet. Em específico, ao menos as regras listadas abaixo devem ser seguidas:

**Controle:** O jogo requer apenas um controle direcional ou *joystick*, pois envolve apenas mover o Pac-Man em qualquer uma das quatro direções principais para **cima**, para **baixo**, para a **esquerda** e para a **direita** em uma única sala semelhante a um labirinto.

**Objetivo:** Comer todos os **240** Pac-dots e **quatro** pílulas de energia dentro do labirinto. Os Pac-dots valem **10** pontos cada e as pílulas de energia valem **50**. Quando todos são consumidos, o jogo avança para o próximo nível, o qual ocorre na mesma sala/labirinto.

**Frutas:** Uma **fruta bônus** aparece após você comer **70** Pac-dots e novamente após mais **100** Pac-dots. O nível UM tem uma **cereja** que vale **100** pontos, o nível DOIS tem um **morango** que vale **300** e o nível TRÊS tem uma **laranja** que vale **500**. Os valores dos pontos continuam a aumentar a cada nível.

**Fantasmas:** **Blinky** (vermelho) e **Pinky** (rosa) irão persegui-lo, enquanto **Inky** (verde) e **Clyde** (laranja) se movem aleatoriamente. Blinky aumenta de velocidade à medida que você come mais pontos.

**Pílulas de Poder:** As pílulas de energia deixam os fantasmas temporariamente **azuis**, e você pode comê-los neste momento. O primeiro fantasma comido lhe dá **200** pontos, e isso dobra a cada vez até que o último fantasma lhe dê **1.600**. A quantidade de tempo que os fantasmas permanecem azuis fica menor a cada nível.

**Vidas:** O jogo dá a você **três** vidas, e você perde uma sempre que toca um fantasma. Você ganha uma **vida extra** durante o jogo quando atinge **10.000** pontos. Se você perder todas as vidas, o jogo acabou.

## 2 Conteúdo de Entrega

Nessa primeira etapa, o que precisa ser entregue envolve:

- A estrutura de classes que represente os elementos do jogo: tabuleiro, fantasmas, pacman, pontos, pílulas, etc. Lembre-se de criar estruturas hierárquicas de classes abstratas (ou interfaces) e classes concretas;

- O sistema deve ser **modularizado** em três **pacotes**: (1) elementos do sistema (pacman, fantasmas, etc.); (2) interface gráfica, e; (3) *engine* de funcionamento (movimentação dos elementos, contagem de pontos, etc.). Nessa etapa, o que precisa ser entregue são os itens (1) e (3). A *engine* ainda não precisa estar completa; o que precisa ser implementado é o método que faz os fantasmas buscarem o pacman;
- Além do código, um **diagrama de classes UML**, indicando quais as classes desenvolvidas, seus métodos, atributos e relacionamentos (herança, agregação, composição, etc.) precisa ser entregue. A ferramenta de código aberto utilizada para desenhar o diagrama é:
  - StarUML: <http://staruml.io/download>
- Como teste dessa primeira etapa, deve-se, ao executar o projeto, desenhar o tabuleiro na tela (usando *System.out.println(...)*) e em seguida deve-se colocar o pacman e um fantasma em posições válidas do tabuleiro (aleatoriamente, então cada execução é um posicionamento diferente), mostrando qual o caminho que o fantasma irá seguir para encontrar o pacman;

Observações importantes:

- Apesar da *engine* não precisar estar completa nessa fase do projeto, ela já deve ser pensada e projetada considerando que outras funcionalidades serão acrescentadas;
- Dica para implementação da *engine*: pode-se interpretar o labirinto como um grafo, em que cada cruzamento de caminho é um nó, e cada trecho de um caminho é uma aresta. Assim, para cada fantasma, é possível fazer uma busca em largura no labirinto/grafos visando encontrar o pacman, e então seguir o menor caminho para se mover. Essa é uma forma válida de busca, mas estratégias mais elegantes também podem ser usadas;
- Faça um programa bem feito, pois você ficará com ele até o final do semestre! Vou pedir que você estenda este programa;
- Um programa que faça tudo que pedi acima vale 8,0. Se você me surpreender positivamente, seu programa vale 10.

### 3 Formato e Data de Entrega

O trabalho é **individual**, e a data de entrega é **19/11/2021** (até as 8:00hs do dia posterior). Trabalhos atrasados não serão aceitos, recebendo nota **zero**. Quaisquer programas similares terão nota zero independente de qual for o original e qual for a cópia. Será utilizada **ferramenta automatizada** para a detecção de **plágio**, com conferência manual de casos suspeitos. Os projetos devem ser entregues via Atividades do Tidia-ae (<https://ae4.tidia-ae.usp.br/portal>).

O formato da entrega deve ser um arquivo \*.ZIP contendo:

- O nome da pasta a ser zipada deve ser o número USP do aluno. **Trabalhos sem esse padrão de nome de arquivo não serão corrigidos e valerão ZERO**;
- Uma imagem (\*.JPG ou \*.PNG) contendo o diagrama de classes;
- Um projeto NetBeans contendo todo o código desenvolvido.

### 4 Critério de Avaliação

- Funcionamento correto e organização de classes e pacotes (separação de funcionalidades, relação de hierarquia, classe abstrata/interface, etc.): 50%
- Eficiência e elegância da abordagem sugerida da *engine* do sistema: 40%;
- JavaDoc e documentação interna das classes e métodos: 10%. Pesquise como criar e padronizar a documentação de software escrito em Java.