

Carlos Silvestre Moreira Júnior
Brayan Deyvid dos Santos Freitas
Lucas Santiago Vieira Melgaço

Pacman
Algoritmo e Estrutura de Dados

Itabira

2018

Carlos Silvestre Moreira Júnior
Brayan Deyvid dos Santos Freitas
Lucas Santiago Vieira Melgaço

Pacman

Algoritmo e Estrutura de Dados

Relatório escrito do trabalho da disciplina de ECOI04 - Algoritmo e Estrutura de Dados, ministrada pelo professor, Sandro Carvalho Izidoro, com o objetivo de justificar a apresentar a modelagem do jogo Pacman utilizando Java.

Universidade Federal de Itajubá

Campus Itabira

Algoritmo e Estrutura de Dados I

Itabira

2018

Resumo

Neste projeto, será abordado como foi a implementação do game Pacman, o jogo original apesar de ser "antigo" conta com técnicas de busca pelo caminho mais otimizado envolvendo grafos. Com base nisso implementamos as básicas funções do jogo, devido as dificuldades que outras implementações nos gerou.

Durante o projeto foi notório o protagonismo da biblioteca gráfica JLabel para execução de processos envolvendo imagens, sons, sendo uma ferramenta muito explorada. A decisão de dividir em quatro classes o código, auxiliou na identificação de métodos e atributos.

Entretanto para chegarmos a essa divisão obtivemos uma série de problemas como deixar o mapa matricial proporcional as condições da imagem utilizada para este atributo no código. Além disso obtivemos alguns imprevistos como o tempo levado para executar a manobra de colisão entre elementos no game, muito disso devido a quantidade de métodos para retornar posição dos elementos.

Palavras-chave: Pacman, Fantasma, mapa, jogo.

Lista de ilustrações

Figura 1 – Distinção entre estruturas e POO	6
Figura 2 – Flexibilidade em distintos Sistemas operacionais	7
Figura 3 – Roteiro para criação do game	7
Figura 4 – Jogo Pacman	8
Figura 5 – Modelagem de classes	9
Figura 6 – Função Movimento	10
Figura 7 – Movimento Fantasma	10
Figura 8 – Método que retorna dimensões de fantasma	10
Figura 9 – Limites da imagem da tela	11
Figura 10 – Condicionais do método mover	11
Figura 11 – Colisão	12
Figura 12 – Abertura do portão	12

Sumário

1	INTRODUÇÃO	5
2	REFERENCIAL TEÓRICO	6
2.1	Programação Orientada a Objetos	6
2.2	Java	7
2.3	Algoritmo Pacman	7
3	METODOLOGIA	9
3.1	Modelagem	9
3.2	Classe Pacman	9
3.3	Classe Fantasma	10
3.4	Classe Jogo	11
3.5	Classe Iniciar	12
4	RESULTADOS E DISCUSSÕES	13
5	CONSIDERAÇÕES FINAIS	14
	REFERÊNCIAS BIBLIOGRÁFICAS	12

1 Introdução

A implementação do jogo pacman é uma oportunidade de aprimorar os conceitos lógicos vistos em Lógica de Programação, e utilizar as ferramentas de orientação a objeto vistas em Algoritmo e Estrutura de Dados. E quando se utiliza a linguagem Java o conceito do POO torna-se mais trivial, pois java utiliza objetos, com estes temos métodos e atributos, utilizando o conceito de encapsulamento limitando o acesso as informações, mas tornando a programação mais intuitiva e organizada.

2 Referencial Teórico

A disciplina de AEDS (Algoritmos e Estrutura de Dados) I é extremamente necessária para o curso de Engenharia da Computação, sendo de grande importância para o formando nessa área. Trabalhos feitos e aplicados nessa disciplina mostram-se importantes para o aprendizado da matéria, fazendo uso na prática das estruturas de dados e da orientação a objeto que diversas linguagens de programação proporcionam.

Neste trabalho foi realizado a implementação do jogo Pacman em Java fazendo uso das diversas ferramentas da linguagem, principalmente suas bibliotecas gráficas onde destaca-se a JLabel. Usada neste projeto. Serão mostradas algumas partes importantes que se destacam nesse projeto principalmente para o funcionamento do mesmo.

2.1 Programação Orientada a Objetos

A orientação a objetos, é uma ferramenta utilizada em desenvolvimento de softwares, que seguem diferentes modelos. Para descrever esses modelos, a orientação a objetos é o paradigma mais empregado entre todos.

A Programação Orientada a Objetos (POO), é um padrão seguido por muitas linguagens como C++ e Java. Diferentemente da programação estruturada onde temos funções que são aplicadas globalmente em aplicações, em POO, temos métodos que são aplicados aos dados de cada objeto. Entretanto, as funções e métodos são iguais, sendo distintos apenas pelo seu escopo.



Figura 1 – Distinção entre estruturas e POO

2.2 Java

A linguagem Java é uma alternativa para gerenciamento de memória, organização de código e inclusão de bibliotecas, além de ser compatível com inúmeros dispositivos e sistemas operacionais. Sendo que esta compatibilidade só é possível pela existência da Java Virtual Machine (JVM). Essa camada, a máquina virtual, não entende código java, ela entende um código de máquina específico. Esse código de máquina é gerado por um compilador java, como o javac, e é conhecido por "bytecode", pois existem menos de 256 códigos de operação dessa linguagem, e cada "opcode" gasta um byte. O compilador Java gera esse bytecode que, diferente das linguagens sem máquina virtual, vai servir para diferentes sistemas operacionais, já que ele vai ser "traduzido" pela JVM.

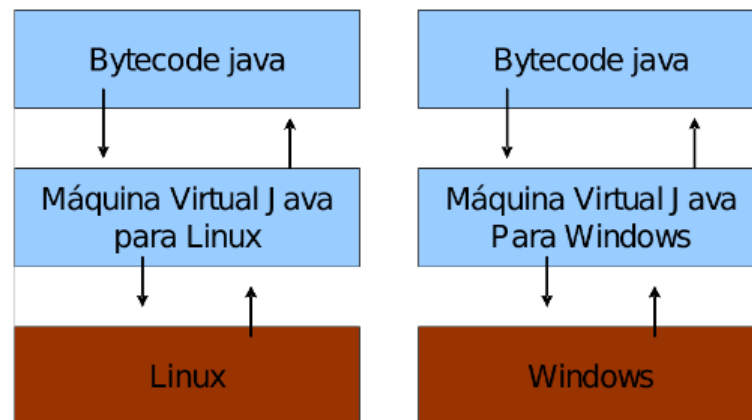


Figura 2 – Flexibilidade em distintos Sistemas operacionais

2.3 Algoritmo Pacman

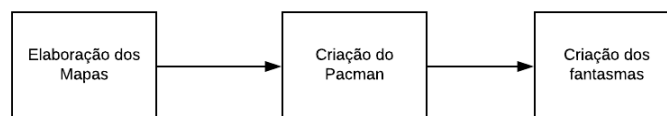


Figura 3 – Roteiro para criação do game

A primeira etapa do código consiste na elaboração do mapa, utilizando como base uma matriz, determinando os labirintos e os trajetos através de distintos caracteres.

Posteriormente é necessário dar vida ao personagem Pacman, este seguirá o plano de coordenadas tendo seu movimento limitado pelos obstáculos já construídos no game. Além disso, fazer com que as pílulas existentes no trajeto desapareçam caso haja interseção com Pacman.

A criação dos fantasmas, inclui detalhes como a parte da saída, inversões de condições do portão, colisão com Pacman, e sua rota no mapa depende da trajetória do personagem Pacman, seguindo sempre as coordenadas do personagem.

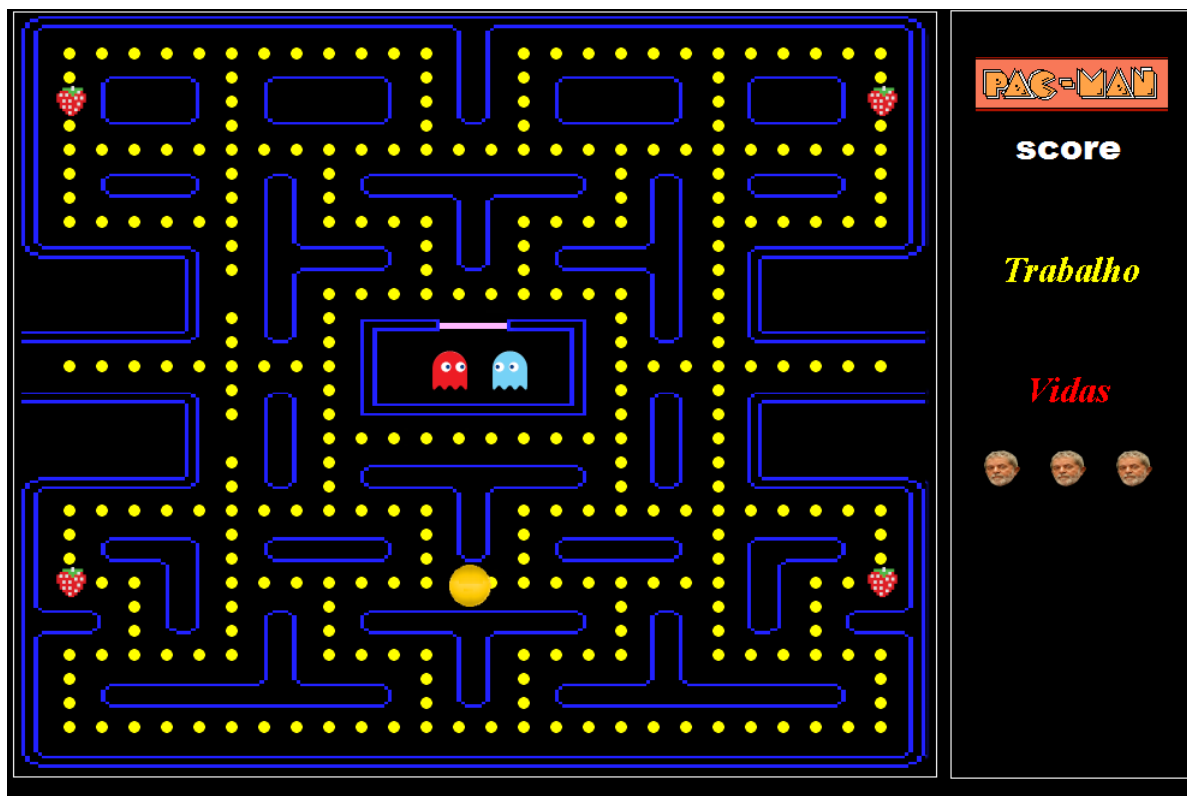


Figura 4 – Jogo Pacman

3 Metodologia

Para a elaboração do código foi utilizado ambiente de desenvolvimento eclipse, que compila códigos da linguagem Java. A escolha da linguagem foi um consenso dos integrantes visando a possibilidade de expandir conhecimento em uma linguagem extremamente usual no mundo da computação, seja em desenvolvimento de programas ou aplicativos.

Inicialmente foram estabelecidos as condições para criar as classes, sendo estas planejar uma herança de uma classe denominada "entidades móveis" da qual herdaria seus métodos de movimentação e suas coordenadas, tanto para fantasmas, quanto para o pacman. Infelizmente esse modelagem não pôde ser implementada, entretanto o programa foi modelado e dividido em quatro classes principais.

3.1 Modelagem

As classes implementadas foram a classe Iniciar, Jogo, Pacman e Fantasma. Sendo que nestas classes a maioria dos atributos são relacionados a parte gráfica e recursos sonoros, trazendo assim dinâmica semelhante a do jogo original.

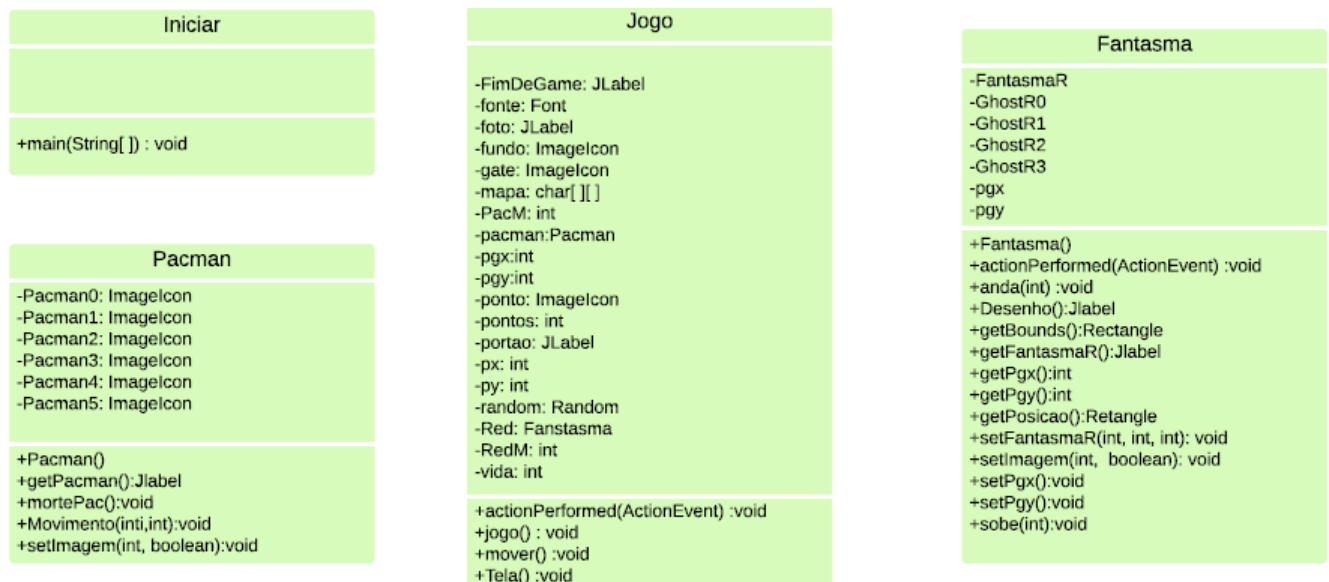


Figura 5 – Modelagem de classes

3.2 Classe Pacman

A Classe Pacman é altera os atributos do Pacman basicamente setando os ícones e a posição dele no mapa. O método mostrado na figura 6 exhibe como funciona a mudança

de coordenada do pacman quando o usuário muda sua direção.

Foram inseridas 5 imagens para o Pacman, uma pra indicá-lo na direção superior, inferior, esquerda, direita e ele totalmente fechado, situação necessária para relizar a sensação de mordida no personagem.

```
public void setImagemPac(int PacM,boolean iniciar) {  
    if (iniciar) {
```

Figura 6 – Função Movimento

3.3 Classe Fantasma

Nesta classe basicamente tem-se o controle das ações dos fantasmas gerando atualização em suas imagens conforme os comandos aleatórios de direções são gerados:

```
public void setImagemFR(int FanR,boolean anima){  
    if(anima) {  
        FantasmaR.setIcon(GhostR0);  
    }  
    else {  
        FantasmaR.setIcon(GhostR1);  
    }  
    switch(FanR) {  
        case 0:  
            FantasmaR.setIcon(GhostR0);  
            break;  
        case 1:  
            FantasmaR.setIcon(GhostR1);  
            break;  
        case 2:  
            FantasmaR.setIcon(GhostR3);  
            break;  
        case 3:  
            FantasmaR.setIcon(GhostR2);  
            break;  
        case 4:  
            FantasmaR.setIcon(GhostR1);  
            break;  
        default:  
            break;  
    }  
}
```

Figura 7 – Movimento Fantasma

Um método a se destacar é o que trata minha imagem como um retângulo e retorna a imagem do respectivo fantasma, auxiliando futuramente na parte de colisão:

```
public Rectangle getBoundsR() {  
    return FantasmaR.getBounds();  
}
```

Figura 8 – Método que retorna dimensões de fantasma

3.4 Classe Jogo

Nesta classe, se encontram todos os elementos do jogo, como a mapa, sejam estes as coordenadas do pacman, fantasma, portão. Se tratando especificamente do mapa todas as regiões para trajetos e que contém pílulas são representadas pelo caractere "o", e quando o pacman as intercepta são substituídas por "espaço em branco"(incrementando automaticamente a pontuação). Regiões onde tenho obstáculos são representadas por "x", sendo impossível tanto para pacman ou para fantasmas transitarem por estas.

```
foto.setBounds(0,0, 1001, 686);
```

Figura 9 – Limites da imagem da tela

Para a movimentação é necessário limitá-la sempre que encontrar um obstáculo representado pelo caractere "x". No método void mover() da classe jogo essas condicionais são impostas anulando movimentos nas regiões especificadas, dando a sensação de labirinto ao game.

```
if(e.getKeyCode()==KeyEvent.VK_RIGHT && mapa[py][px+1]!='x') {  
    PacM=1;  
}  
if(e.getKeyCode()==KeyEvent.VK_LEFT && mapa[py][px-1]!='x') {  
    PacM=2;  
}  
if(e.getKeyCode()==KeyEvent.VK_DOWN && mapa[py+1][px]!='x') {  
    PacM=3;  
}  
if(e.getKeyCode()==KeyEvent.VK_UP && mapa[py-1][px]!='x') {  
    PacM=4;  
}
```

Figura 10 – Condicionais do método mover

Uma etapa, com particularidades é a parte de colisão, que teoricamente, é "simples pensar", pois tenho uma matriz, e caso os elementos destas (pacman e fantasma), tenham as mesmas coordenadas basta reiniciar o game e decrementar a vida.

Entretanto, foram utilizados devido aos problemas da aplicação anterior, associação de retângulos as imagens do pacman e do fantasma. Caso ocorra interseção entre estes objetos a colisão é realizada, parando o game e decrementando a vida.

```

public void jogo() {
    //System.out.print(pacman.getBounds());
    //System.out.print(Red.getBounds());
    if (iniciar) {
        if (Red.getBounds().intersects(pacman.getBounds())) {//*****
            pacman.mortePac();
            JOptionPane.showMessageDialog(null, "Você perdeu 1 vida!!!");
            pacman.setBounds(px * 27 + 10, py * 20 - 3, 45, 43);
            vida--;
        }
    }
}

```

Figura 11 – Colisão

A observação a se destacar no código da figura anterior é que as posições x crescem da esquerda pra direita e em y a particularidade é as coordenadas são contadas da extremidade superior, fazendo com que cresça de cima para baixo.

Outro condicional que tem papel fundamental no game é a de gerenciamento do portão, no qual no game é aberto na condição inicial caso usuário tecla "space". O elemento portão na matriz recebeu o caractere "z", e através deste é possível saber quando os fantasmas atingiram o portam para os mesmos, terem a possibilidade de sair da "prisão":

```

else if (mapa[pgy][pgx] == 'z' || mapa[pgy][pgx + 1] == 'z' || mapa[pgy][pgx - 1] == 'z') {
    mapa[pgy][pgx] = 'x';
    mapa[pgy][pgx + 1] = 'x';
    mapa[pgy][pgx - 1] = 'x';
    Red.sobe(-1);
    portao.setBounds(360, 238, 60, 52);
    sair = true;
}

```

Figura 12 – Abertura do portão

E como é visto na figura passada assim que o fantasma passa pela fronteira do portão x é escrito, indicado que o portão é novamente fechado.

3.5 Classe Iniciar

É basicamente a classe main do jogo, ela inicia o jogo criando a janela em que será jogado o pacman.

4 Resultados e Discussões

A dificuldade mais aparente foi a maneira com que seria feita a busca do fantasma pelo caminho mais fácil até o pacman, pesquisando utilizaríamos conceitos de grafos, ferramenta computacional que só teríamos conhecimento em disciplinas futuras do curso, por isso houve preferência por deixar o fantasma circular de maneira randômica pelo labirinto.

O Pacman apesar de ser um jogo bem antigo, mostra-se suscetível ao uso de Programação Orientada a Objeto para sua construção, mostrando-se de extrema importância para o aprendizado da matéria. A utilização das várias ferramentas orientadas a objeto na linguagem Java, mostrou-se importante para o desenvolvimento do trabalho, mostrando as diversas formas de se trabalhar um código.

5 Considerações Finais

O uso da Orientação a Objeto com o Java, possibilitou a conhecer melhor outra linguagem, conhecer algumas de suas ferramentas. A aplicação dessas ferramentas no trabalho foi de essencial importancia para o aprendizado da matéria AEDS. Foi de extrema importância construir tal tipo de trabalho pois integra o conhecimento que se deve ter um Engenheiro da Computação. Dito isso mostra-se muito importante construir trabalhos práticos e não somente ficar na parte teórica, possibilitando assim o formando de ver e usar ferramentas que estão a sua disposição.

Referências Bibliográficas

DEITEL, PAUL. Java como programar. 4ª Edição, 2003.

Java2DPacmanGame. Disponível em <<https://github.com/leonardo-ono/Java2DPacmanGame>>. Acesso em 27/10/2018. Pacman Game Using Java. Disponível em <<https://github.com/MohamadSobri/Java2DPacmanGame-using-Java>>. Acesso em 02/11/2018.