

 <b>FACULDADE DE VIÇOSA</b> Ampliando Horizontes		Roteiro 10 - Pacman dentro do mapa	
<b>Disciplina:</b>	DET476 - Tópicos especiais 2		
<b>Data</b>	11/05/2016	<b>Valor da Avaliação:</b>	5 pontos
<b>Professor:</b>	Vinícius Duarte Lopes		

**Caso ainda não tenha um projeto pronto para o seu Pacman, proceda aos passos abaixo. Caso contrário, vá direto à especificação própria deste roteiro contida a partir da próxima página. Quem não entregou o roteiro 9 até o dia 11/05/2016 as 19 horas (até começar a aula), perdeu os pontos.**

1) Baixe o projeto da Engine do jogo em:

<https://github.com/vinduarte/EngineJogo>

2) Abra o projeto no seu NetBeans. (Abrir projeto e selecionar a pasta EngineJogo)

3) Crie um novo projeto para ser o projeto do jogo, chame-o de Pacman. Crie o projeto com classe principal.

4) Adicione a biblioteca EngineJogo. Clique com o segundo botão em Biblioteca (dentro da aba do projeto) e clique em “adicionar projeto...”. Adicione o projeto EngineJogo.

5) Faça a sua classe criada herdar da classe Jogo de EngineJogo. Implemente:

A. O construtor.

B. As variáveis privadas da classe (x, y, BufferedImage, direção etc).

C. O método desenha.

D. O método atualiza.

E. Sobrecarregue o método keyPressed.

Link publico da imagem do Pacman:

<https://dl.dropboxusercontent.com/u/46945366/pacman.png>

6) Faça seu pacman mover-se na tela, respondendo a comandos das setas do teclado e respeitando os limites da tela.

7) Armazene seu projeto no GitHub para futuras alterações.

## Roteiro 10: Carregando mapas e fazendo o *pacman* mover-se apenas dentro dos caminhos possíveis.

- Caso ainda não tenha adicionado seu projeto no *GitHub*, para futuros acessos e alterações, faça isso antes de proceder ao primeiro passo deste roteiro.
- Não é necessário entregar o código deste roteiro ao professor, como nos últimos roteiros.
- É necessário mandar por *e-mail*, ao professor, o *link* do seu projeto *Pacman* no seu *GitHub*.
- **É necessário explicar ao professor o código que você fez, respondendo às perguntas que poderão ser feitas ao longo da explicação. A nota do roteiro, assim como dos roteiros futuros, será baseada no quão boa será sua explicação. Este roteiro deve ser apresentado até o dia da prova 2, 25/05/2016, antes ou depois da prova das 18 as 21 horas.**
- É recomendável que a cada alteração que você considere significativa no código do seu Pacman, você faça um *commit* no seu repositório *Git*.

### Objetivo deste roteiro

- 1) Baixe o código que você deve editar: <https://github.com/vinduarte/PacmanVinicius>
- 2) Baixe novamente o código da engine do jogo, que foi atualizado: <https://github.com/vinduarte/EngineJogo>
- 3) Fazer seu *pacman* mover-se apenas dentro dos caminhos possíveis do mapa.
- 4) Fazer o *pacman* só mudar de a posição da boca caso seja possível movimentar-se na direção solicitada. Armazenar o “pedido” de movimento. Ou seja, quando o usuário clicar na seta para cima e não puder ir para cima, armazene o clique dele e mova para cima quando for possível.

### Códigos necessários

Engine do jogo:

<https://github.com/vinduarte/EngineJogo>

Projeto a continuar:

<https://github.com/vinduarte/PacmanVinicius>

### Explicações úteis

O mapa deve ser lido pelo seu jogo, de alguma maneira. O código dado usa a opção que segue. Usar um arquivo texto simples para descrever o mapa e ser lido numa entrada. O arquivo pode estar no formato abaixo exemplificado:

```
1 1 1 1 1 1
1 2 0 0 0 1
1 0 1 1 0 1
1 0 1 1 0 1
1 0 0 0 0 1
1 1 1 1 1 1
```

Neste arquivo, o número 1 representa a parede, que são limites por onde o *pacman* não pode entrar. O número 0 representa uma região onde o *pacman* pode estar. O número 2 representa a posição inicial do *pacman*. Lembre-se que este é apenas um exemplo, os números usados num caso real de implementação podem ser quaisquer que o programador queira utilizar. Anexado a este projeto estão alguns mapas possíveis, que **não** devem funcionar no seu jogo.

Para armazenar o mapa na memória, podemos utilizar um *ArrayList* de *String*, que armazena, em cada elemento da lista, uma linha do mapa. Assim, seguindo o exemplo de mapa dado anteriormente, poderíamos ter uma lista da seguinte maneira:

Elemento 0: `String {111111}`

Elemento 1: String {120001}  
Elemento 2: String {101101}  
Elemento 3: String {101101}  
Elemento 4: String {100001}  
Elemento 5: String {111111}

Assim, para acessarmos a terceira linha e quarta coluna, precisamos pegar o elemento 2 da lista e, deste, o caracter de índice 3 na *String*. É bom lembrar que o mapa deve ficar visível no código todo da classe principal do jogo.

Para declarar o *ArrayList*, podemos fazer:

```
private ArrayList<String> mapa;
```

Para ler o arquivo, precisamos que ele esteja dentro de uma pasta do projeto, podemos criar uma pasta chamada "mapas" e colocar os arquivos dos mapas lá dentro. Feito isso, podemos fazer a leitura através de um objeto *Scanner*, por exemplo:

```
Scanner s = new Scanner(new File("mapas/mapa.txt"));
```

Para carregar o *ArrayList* precisamos primeiro criar um objeto:

```
mapa = new ArrayList<>();
```

Usando o *Scanner* para preencher, percorremos o arquivo da entrada até que não haja mais linhas nele e adicionamos a linha lida em cada iteração do *while*:

```
while (s.hasNextLine()) {  
    mapa.add(s.next());  
}
```

Depois de feita a leitura, podemos armazenar o total de linhas que o arquivo contém e o total de colunas. Para isso, podemos apenas pegar o tamanho da lista para o total de linhas e o tamanho de uma linha qualquer para o tamanho da coluna. Quando não precisar mais usar o *Scanner*, é bom fechá-lo:

```
s.close();
```

Uma forma de desenhar o mapa é usando a função *fillRect*. Um exemplo é mostrado a seguir, o código deve ir dentro do método *desenha*:

```
for (int i = 0; i < totalLinhas; i++) {  
    for (int j = 0; j < totalColunas; j++) {  
        if (charAt(i, j) != '0') {  
            g.fillRect(j * MAPA_X_PIXEL , i * MAPA_X_PIXEL , TAM, TAM);  
        }  
    }  
}
```

O método *fillRect* preenche um retângulo na tela, a ser construído na posição passa nos dois primeiros parâmetros e com tamanho igual aos dois últimos parâmetros. Note que com o código acima acaba preenchendo posições maiores do que apenas uma "posição" do mapa. Pense nisso.

Se a boca do *pacman* estiver abrindo e fechando muito rápido, faça o ajuste para ela abrir e fechar mais devagar e não parecer frenético.