



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Trabalho Prático

Tiny-PAC

Programação Avançada



2022/2023

João Santos - 2020136093

1. Introdução

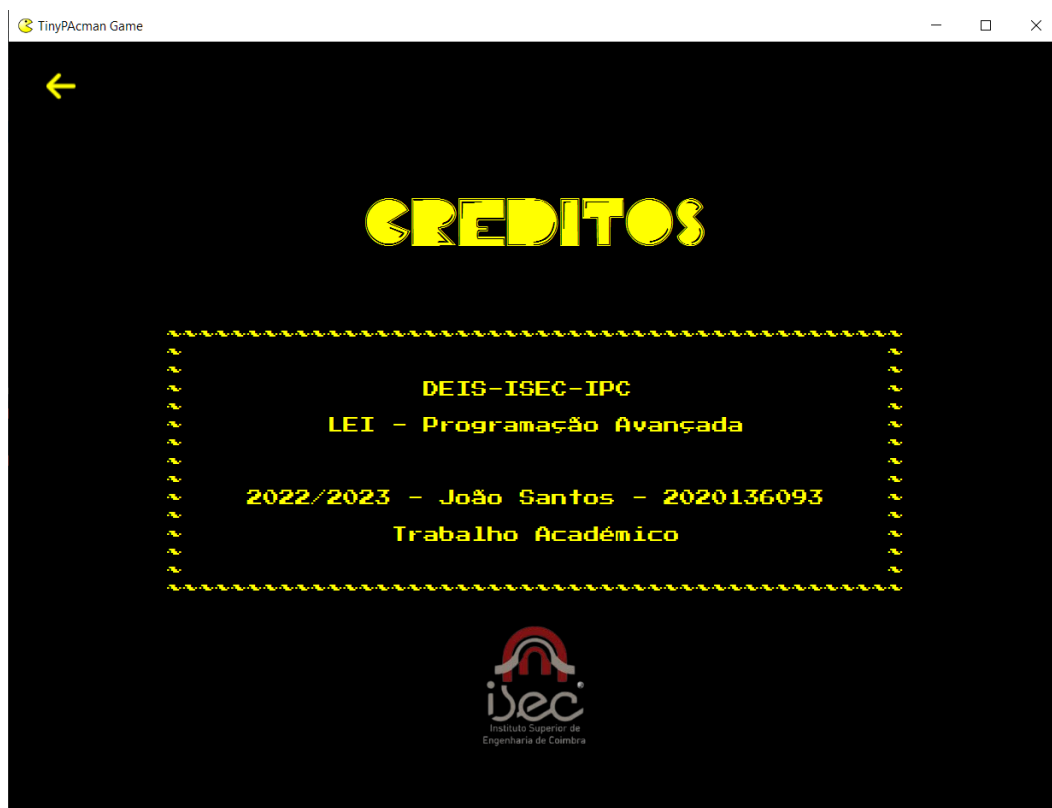
Na disciplina de Programação Avançada, foi criada uma versão do famoso jogo Pac-Man em linguagem Java, chamada Tiny-PAC. Nesse jogo, o objetivo é controlar o personagem Pac-Man enquanto ele se move por um labirinto, coletando itens e evitando os fantasmas controlados pelo computador. O jogador deve guiar o Pac-Man para comer o maior número possível de itens, ao mesmo tempo em que busca evitar os fantasmas, que tentam capturá-lo.

2. Decisões Tomadas

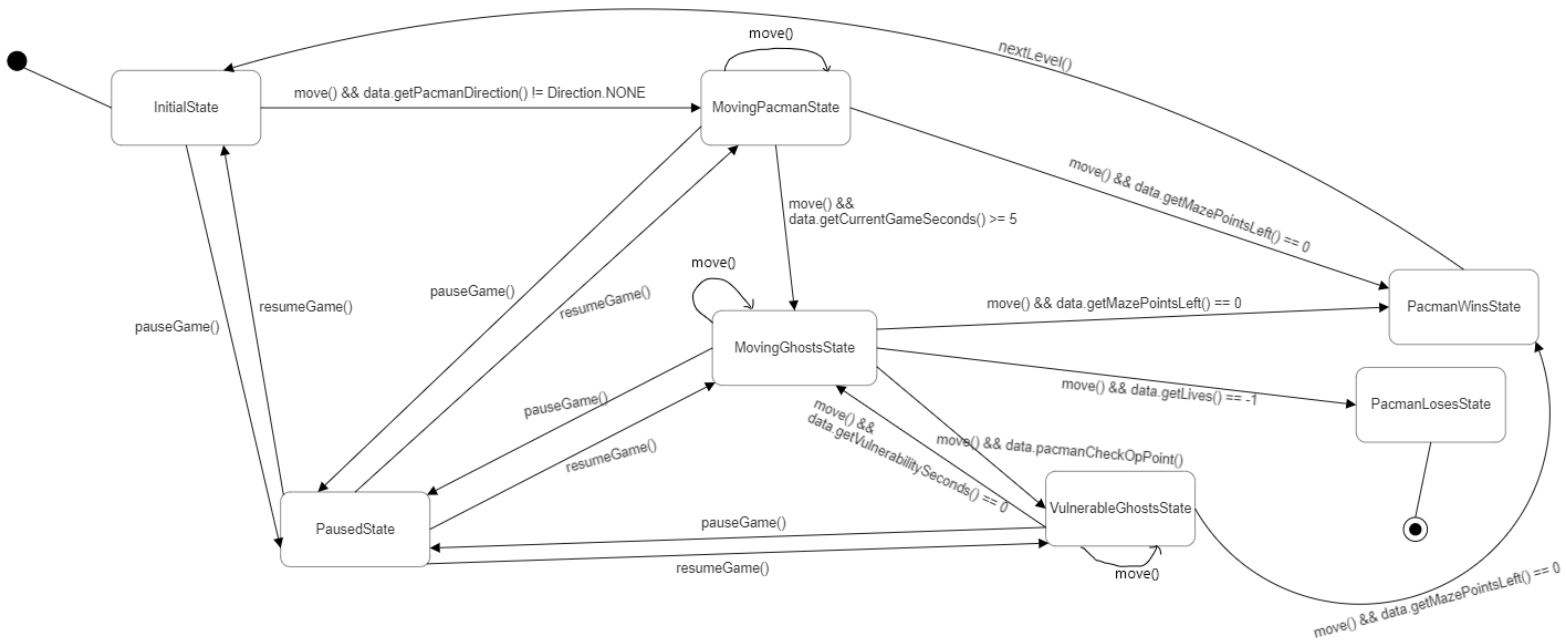
- Criar uma enumeração de direções (UP, DOWN, LEFT, RIGHT) para gerenciar os movimentos dos elementos do jogo.

```
public enum Direction {  
    14 usages  
    UP, DOWN, RIGHT, LEFT, NONE;  
}
```

- Criar uma classe Ranking serializável que contém variáveis para criação de uma lista para guardar num ficheiro, para que seja mais fácil guardar a pontuação e nome de cada utilizador.
- Colocar os créditos em uma das opções do menu inicial



3. Diagrama da máquina de estados



4. Descrição de classes

- O jogo começa no estado `PlinioState` e muda para o estado `MoverPacmanState` através da função `move()` e atendendo à condição `data.getDirection() != Direction.NONE`.
- Em qualquer momento do jogo é possível mudar de estado para o `JogoEmPausaState`.
- O estado muda do `MoverPacmanState` para o `MoverFantasmasState` através da função `move()` e atendendo à condição `data.getContadorSegundos()>=5`.
- O estado muda do `MoverFantasmasState` para o `PacmanGameOverState` através da função `move()` e atendendo à condição `data.getVidas()==0`.
- O estado muda do `MoverFantasmasState` para o `FantasmasVulneraveisState` através do `move()` e da condição `data.comeBolaComestivel()`. Volta ao estado `MoverFantasmasState` com a condição `data.getSegundosVulnerabilidade()==0`.

5. Descrição de classes

ghostHome – É uma implementação da interface `"IMazeElement"` e representa o local onde os fantasmas se encontram antes de iniciar o movimento (“casa”).

ghostSpawn - É uma implementação da interface `"IMazeElement"` e representa o local por onde os os fantasmas saem para fora da “casa”.

Blinky - Representa o comportamento e estado do personagem Blinky durante o jogo no labirinto, incluindo o seu movimento, posição, histórico e vulnerabilidade.

Point - é uma implementação da interface `"IMazeElement"` e representa a bola comestível pelo pacman (“ponto”).

opPoint - É uma implementação da interface `"IMazeElement"` e representa a bola comestível com poderes.

Clyde - Representa o comportamento e estado do personagem Clyde durante um jogo no labirinto, incluindo o seu movimento, posição, histórico e vulnerabilidade.

freeSpace- É uma implementação da interface "IMazeElement" e representa um espaço vazio no labirinto.

Food- É uma implementação da interface "IMazeElement" e representa a fruta comestível pelo pacman.

PACMAN- A classe PACMAN representa o personagem principal do jogo. Esta implementa a interface IMazeElement, e define o comportamento dos pacman no labirinto. Isto inclui o seu movimento e posição.

Wall - É uma implementação da interface "IMazeElement" e representa as paredes do labirinto. São intransponíveis por qualquer elemento.

Warp - É uma implementação da interface "IMazeElement" e representa os portais por onde o Pac Man atravessa. Se entrar num, aparece no outro.

Ranking- a classe Ranking representa uma pontuação associada a um nome e é responsável por armazenar a pontuação, tempo de jogo e nome do jogador para o ranking Top 5.

TinyPacData- Classe serializável que representa todos os dados do jogo.

A classe possui métodos para obter e definir os valores dos atributos, como as coordenadas do PacMan e dos Fantasmas, a posição da fruta, o nível atual, a pontuação atual, entre outros.

Também há métodos para criar o labirinto a partir de um arquivo de texto e preenchê-lo com os elementos do jogo, como paredes, bolas, frutas, etc.

VulnerableGhostsState- É responsável por controlar o estado dos fantasmas no jogo TinyPac quando estes estão vulneráveis.

A classe herda do TinyPacStateAdapter e implementa métodos para movimentar os fantasmas e o Pac-Man, controlar a duração da vulnerabilidade dos fantasmas e retornar informações sobre o jogo.

PausedGameState- Estado responsável por controlar o estado do jogo TinyPac quando está em pausa.

A classe herda do TinyPacStateAdapter e possui um atributo lastState que armazena o último estado do jogo antes da pausa.

MovingGhostsState - É responsável por controlar o estado do jogo TinyPac quando os fantasmas e o pacman estão em movimento.

MovingPacmanState- É responsável por controlar o estado do jogo TinyPac quando apenas o pacman se está a mexer.

PacmanWinsState- É responsável pelo estado do jogo quando o jogo termina e o desfecho é a vitória do pacman.

PacmanLosesState- É responsável pelo estado do jogo quando o jogo termina e o desfecho é a derrota do pacman. Será o último estado da aplicação.

InitialState- Este estado é usado no início do jogo onde espera até que o jogador inicie o movimento do pacman. Quando isso acontece muda o estado para MovingPacmanState.

ITinyPacmanState- Interface onde se declaram todos os métodos de todos os estados.

TinyPacmanContext- É responsável por gerir toda a máquina de estados.

TinyPacmanStateAdapter - Classe abstrata que serve de base para os diferentes estados do jogo. Implementa a interface do ITinyPacState e apresenta as implementações default dos métodos.

TinyPacmanManager- É responsável pela ligação UI -> Context. Toda a informação / ação requerida pela UI é chamada pelo Manager ao Context.

CSSManager- A classe CSSManager é uma classe utilitária responsável por aplicar um arquivo CSS a um elemento gráfico em JavaFX.

FontManager- A classe FontManager é uma classe responsável por carregar uma fonte personalizada em JavaFX.

ImageManager- A classe ImageManager é uma classe utilitária responsável por carregar imagens em JavaFX.

CreditosUI – UI responsável por mostrar a interface com os créditos.

JogoUI- UI responsável por mostrar o labirinto de jogo e outras informações tais como pontuação e tempo.

MainJFX - É uma classe que estende a Application do JavaFX e é responsável por iniciar a aplicação gráfica do jogo Tiny-Pac.

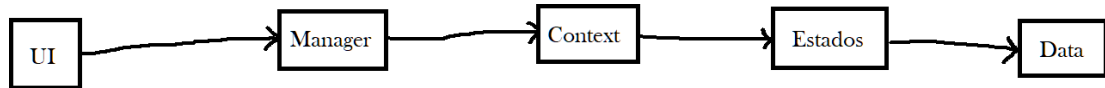
MenuUI- UI responsável por mostrar a interface do menu de jogo.

TOP5UI- UI responsável por mostrar a interface do TOP 5.

RootPane- Configura o layout principal para a interface gráfica do jogo Tiny-Pac, cria e organiza os elementos da interface do utilizador.

6. Relacionamento entre classes

Todo o acesso das UI's aos dados é feito através do seguinte diagrama:



(UI chama métodos do Manager que por sua vez chama métodos do Context, que por sua vez chama métodos dos estados, que chamam os métodos de obtenção de dados presentes no data).

7. Funcionalidades Implementadas

Descrição funcionalidade/ requisito	Implementado	Não Implementado	Implementada Parcialmente
Créditos	X		
Pausa	X		
Maze e GameEngine	X		
Portais	X		
Bola com Poderes	X		
Máquina de estados	X		
Carregar nível anterior em caso de não existir um labirinto definido para o atual	X		
Implementação do jogo em modo gráfico	X		
TOP 5	X		
Gravação do jogo	X		
Restauração do jogo	X		
Blinky	X		
Clyde	X		
Pinky		X	
Inky		X	
Pontuação	X		
Movimento inverso dos fantasmas	X		
Pacman	X		

Mecanismo da fruta	X		
Mudança de nível	X		
Velocidade dos fantasmas aumenta		X	
Verificar se o utilizador deseja carregar jogo	X		
Confirmação se o utilizador deseja sair	X		
Testes unitários		X	
Pacman comer fantasmas	X		
Vidas	X		
JavaDoc	X		

