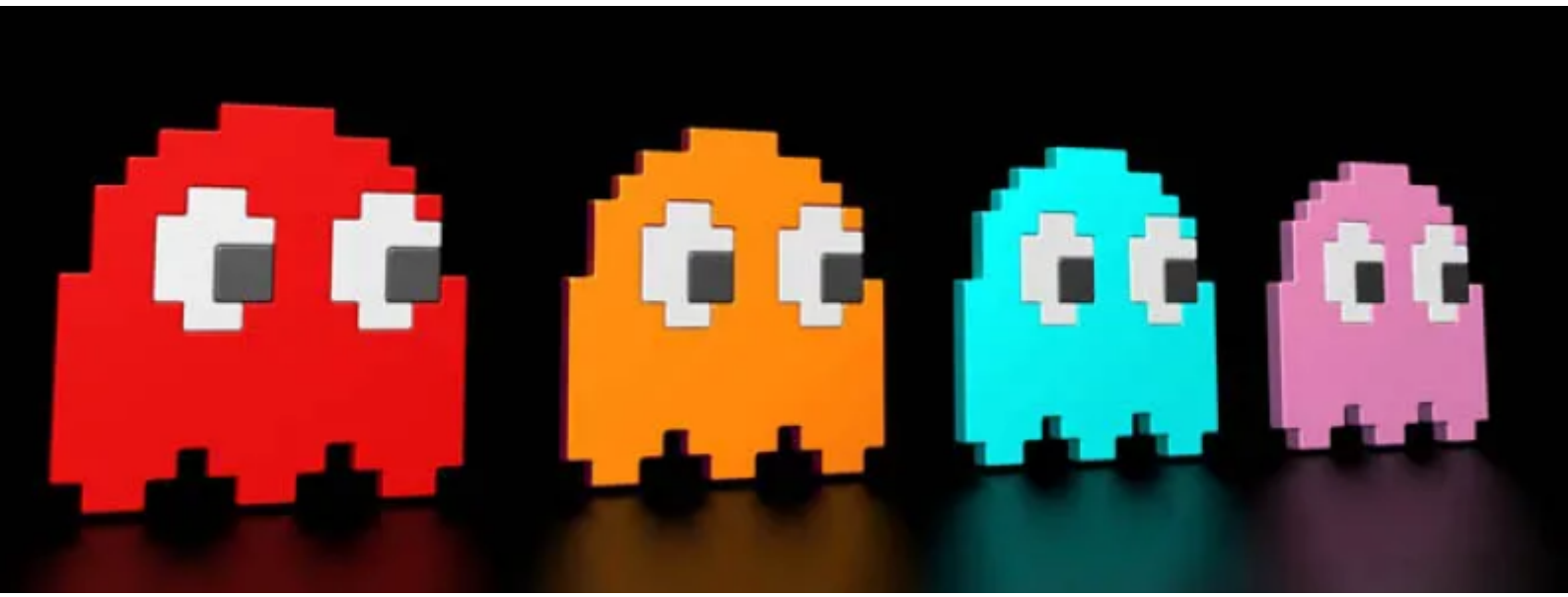
isec Engenharia

2022/2023

Programação Avançada

Meta 1

Licenciatura em Engenharia Informática



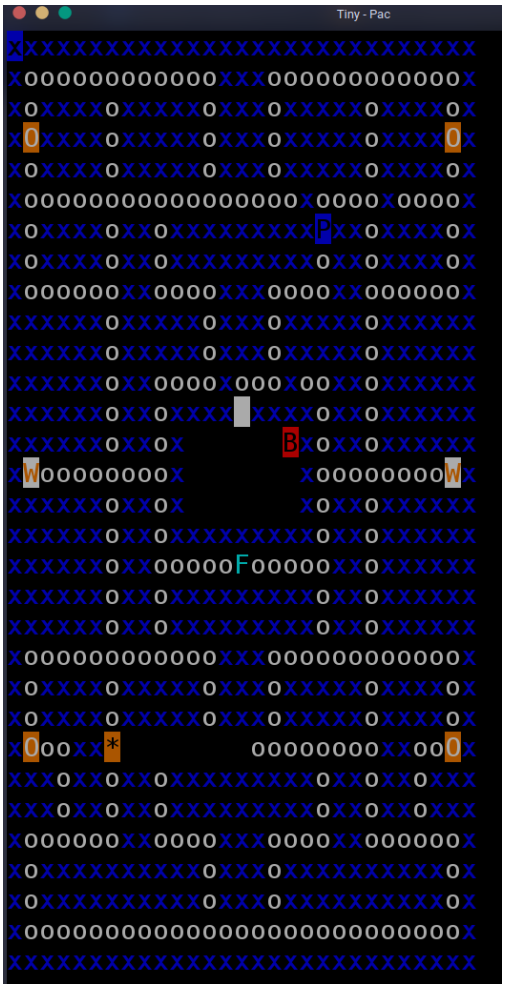
Meta 1 realizada por:

Pedro Nogueira - a2020136533

Índice

1 – Em que Consiste o Trabalho Prático

O trabalho prático em questão consiste em implementar uma versão do jogo Pac-Man, o “Tiny - PAC”. No qual o jogador controla um personagem em um labirinto, com o objetivo de comer frutas e evitando os fantasmas que o perseguem. O jogo em questão deve ser composto também por um menu inicial com a possibilidade de visualizar os melhores jogadores consoantes as pontuações e um jogo parecido com o pacman onde o jogador poderá jogar diversos mapas diferentes consoante o seu desempenho no jogo.



2 - Desenvolvimento do Tiny - PAC

2.1 - Packages do Projeto

2.1.1 - GameEngine

O package do gameEngine é constituído pelos ficheiros do motor de jogo fornecidos pelo Professor Álvaro Nuno Santos, que possibilitam o desenvolvimento do jogo com o decorrer do tempo.

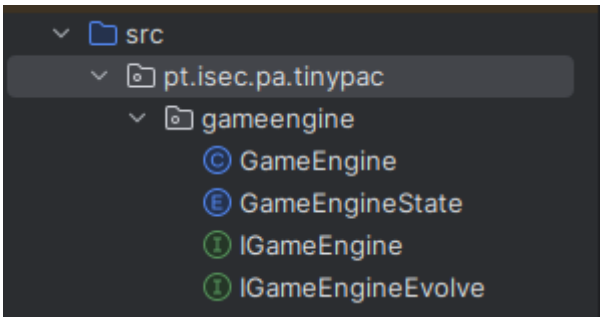


Figura 1 - Packages do gameEngine

2.1.2 - Níveis

O package demonstrado na figura 2 é destinado para a inserção dos ficheiros para os diferentes níveis.

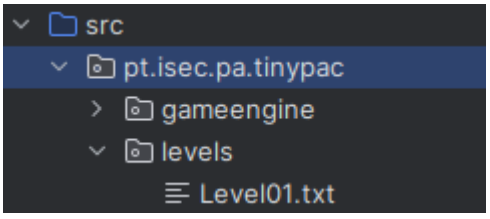


Figura 2 - Package dos Níveis

2.1.3 - Model

2.1.3.1 - Data

Este package Data é composto pelos diversos componentes do jogo, como o labirinto, os fantasmas e os objetos do jogo. Contém toda a informação necessária para o correto funcionamento do jogo.

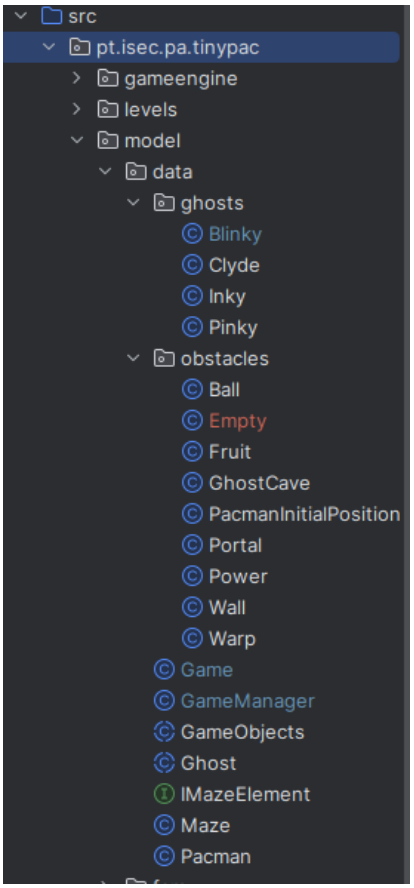


Figura 3 - Package Data

Na classe gameManager funciona como mais uma camada entre a classe que game, é nela que são redirecionados os pedidos vindos da fsm e onde acontecem as leituras de ficheiros. Implementa a interface IGameEngineEvolve para ser possível implementar o evolve, que é capaz de chamar os vários objetos a evoluir no jogo. Na leitura do ficheiro é verificado quais os níveis que existem na pasta especificada, sabendo os níveis existentes o gameManager irá encontrar o melhor nível para um determinado nível, lendo esse ficheiro e criando o maze conforme o necessário.

Cada movimento dos fantasmas e do pacman é implementado individualmente em cada classe, sendo esse movimento chamado no método evolve.

2.1.3.1 - FSM

Uma “Finite State Machine” é composta por um conjunto de transições onde é necessário alterar o comportamento do programa nos diferentes estados, desencadeando várias ações nas diferentes transições.

Para este projeto é utilizada uma máquina de estados com 4 estados diferentes, sendo eles:

- WaitForDirection State - Este estado é responsável por aguardar por uma tecla de direção do utilizador, podendo também dar pausa do jogo
- LockedGhost State - Este estado é responsável por permitir o controle do movimento do pacman e também a gestão da libertação dos fantasmas, podendo também dar pausa do jogo.
- Game State - Este estado é responsável pela gestão normal do jogo, fazendo a gestão da transição de estados de quando existe um termino do jogo.
- Pause State - Este estado permite ao utilizador guardar o jogo para poder ser retomado mais tarde, permite também sair sem guardar e voltar para o ponto de onde estava.
- Game Over - Este estado será usado para apresentar certas informações ao terminar um jogo.

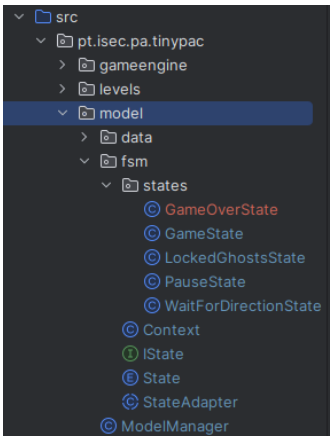


Figura 4 - Package fsm

O model Manager é uma camada do Context que permite uma interligação do utilizador com a fsm, tendo diversos métodos que são necessários por parte do utilizador.

O Context pretende gerir o conteúdo que se pretende realizar na fsm, permitindo ao utilizador realizar uma certa ação de um estado corrente.

2.1.4 - UI

Este package é constituído pela interface gráfica com o apoio da biblioteca Curses Lanterna, usada para a criação de GUI baseadas em texto.

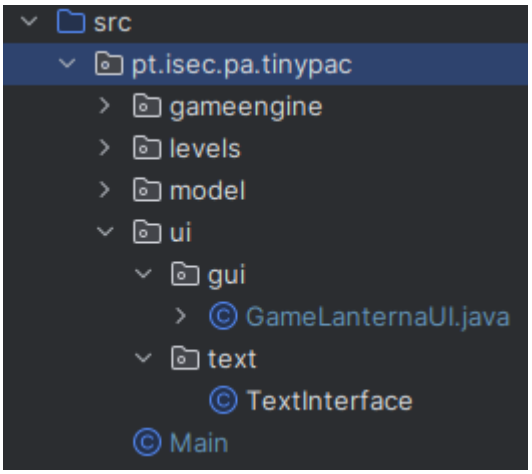


Figura 5 - Package UI

2.1.5 - Utils

Contém algumas classes e enums que dão apoio ao projeto.

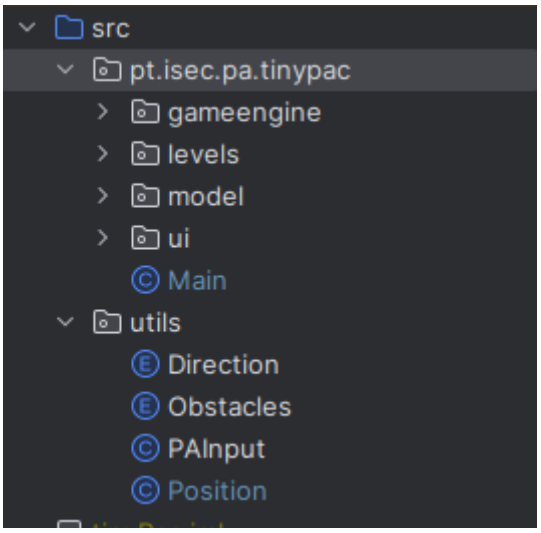


Figura 6 - Package Utils

3 - Diagrama de Estados

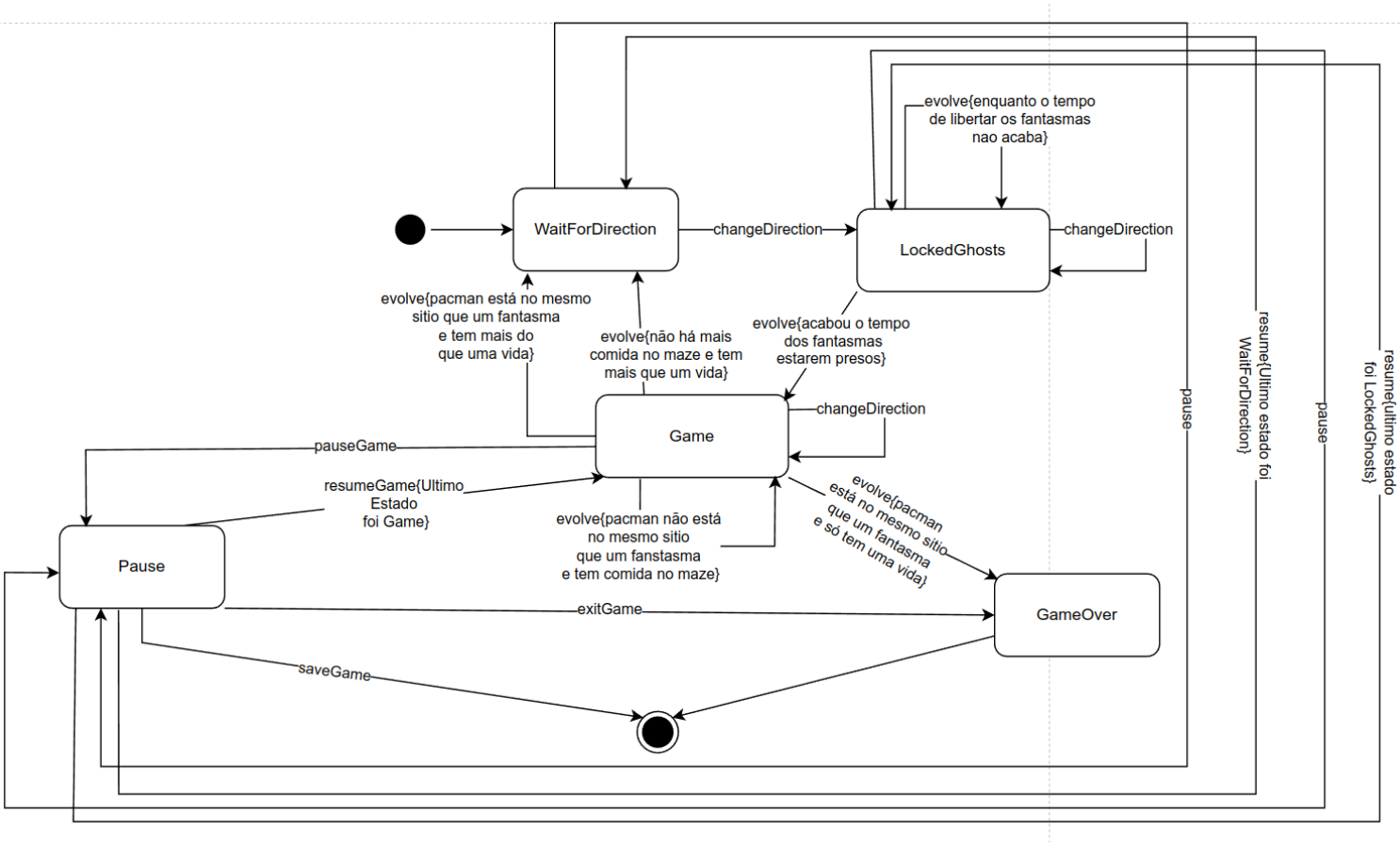


Figura 7 - Diagrama de Estados

- WaitForDirection State - Este estado é responsável por aguardar por uma tecla de direção do utilizador, o jogo volta para este estado quando não existir mais comida no labirinto ou quando o pacman for comido, podendo também dar pausa do jogo.

- LockedGhost State - Este estado é responsável por permitir o controle do movimento do pacman e também a gestão da libertação dos fantasmas, podendo também dar pausa do jogo.

- Game State - Este estado é responsável pela gestão normal do jogo, é permitido mudar a direção do pacman, movimentando também os fantasmas, estando disponibilizada um conjunto de verificações para controlar o funcionamento normal do jogo. Podendo também pausar o jogo.

- Pause State - Este estado permite ao utilizador guardar o jogo para poder ser retomado mais tarde, permite também sair sem guardar e voltar para o ponto de onde estava.

- Game Over - Este estado será usado para apresentar certas informações ao terminar um jogo, o jogo será reencaminhado para este estado quando um pacman é comido e tem menos que uma vida.

4 - Funcionalidades Implementadas

- Leitura dos ficheiros - Métodos situados no gameManager. Permite a leitura de um ficheiro de texto com o mapa e os seus componentes.
- Criação e Organização do projeto
- Diagramas
- FSM
- Alguma movimentação de fantasmas
- Controlo do pacman