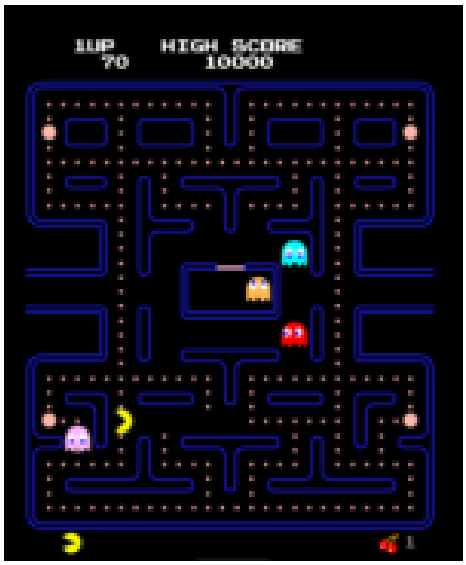
Instituto Superior
de Engenharia
Politécnico de Coimbra

Tiny-Pac Game



Programação Avançada

2022 / 2023

Meta 1

Trabalho realizado por:

- 2021142041 – Pedro Rodrigues Jorge

Instituto Superior
de Engenharia
Politécnico de Coimbra

Índice

1. Introdução

No âmbito da unidade curricular de Programação Avançada foi proposto o desenvolvimento do jogo Tiny-Pac conhecido como Pac-man. Durante a leitura deste relatório, será possível encontrar informações sobre as diferentes classes usadas no programa, bem como a máquina de estados implementada para o seu desenvolvimento.

Neste relatório, serão abordados os detalhes técnicos da arquitetura de desenvolvimento do jogo, com destaque para as funcionalidades mais importantes implementadas e os desafios enfrentados durante todo o processo de desenvolvimento. Adicionalmente, serão discutidos os aspetos futuros que irão ser desenvolvidos no jogo bem como sugestões para as melhorias e expansões futuras.

Figura 1. Interface atual do jogo

2. Classes Utilizadas

- A tabela seguinte pretende descrever as classes mais relevantes utilizadas no desenvolvimento do jogo que necessitam de uma explicação para a fácil interpretação no contexto do programa.

Classes	Package	Descrição
TinyPacUI	Pt.isec.pa.tinypac.ui.text	Esta classe é a responsável por gerir os elementos de visualização do projeto, ou seja, tudo o que envolve a interface de texto sugerida para a implementação.
GameManager	Pt.isec.pa.tinypac.model.data.game	Esta classe é responsável por interagir diretamente com as instancias do elementos necessários para o jogo, bem como algumas propriedades como a chamada dos métodos associados aos personagens do jogo. Posteriormente a máquina de estados irá apenas interagir com esta classe não sendo necessário criar instancias dos elementos do jogo na máquina de estados.

GameLevel	Pt.isec.pa.tinypac.model.data.game	Esta classe é responsável por carregar os ficheiros associados aos níveis de jogo. A existência desta classe permite organizar melhor o projeto e sempre que seja necessário passar de nível, o gameManager irá interagir com a instância do GameLevel e assim carregar qualquer ficheiro de texto, bem como realizar as devidas verificações de requerimentos necessários para carregar um ficheiro.
(Classes dos personagens e elementos)	Pt.isec.pa.tinypac.model.data	Dentro do package “Pt.isec.pa.tinypac.model.data” Existem diferentes tipos de classes como por exemplo o PacMan, Ghost, etc. Estas classes são os modelos para instanciar os personagens e elementos necessários para o decorrer do jogo, bem como garantir as corretas funcionalidades propostas no enunciado do trabalho prático.
Messages	Pt.isec.pa.tinypac.utils	Esta classe foi realizada seguindo o padrão “Singleton” pois apenas permite a criação de uma única instancia para todo o projeto. Esta classe tem a funcionalidade de criar listas de “logs” e mensagens que podem ser utilizadas por diversas razões durante o programa como para situações de “debug” ou até mesmo passagem de informação.

3. Máquina de estados

- Seguidamente está anexado um diagrama representativo da máquina de estados utilizada para gerir o fluxo do jogo.

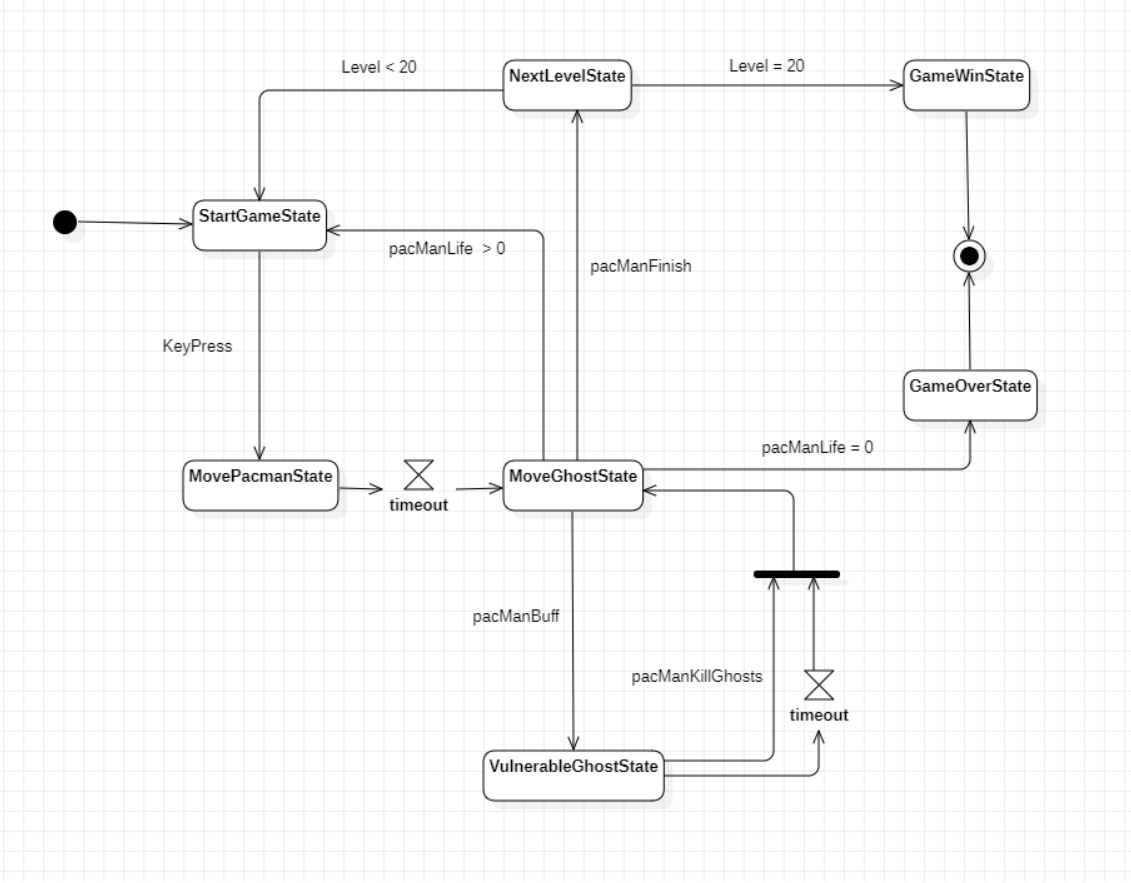


Figura 2. Diagrama de Atividades da máquina de estados

Nome do Estado	Descrição	Mudança de Estado
StartGameState	Corresponde ao estado inicial onde começa a maquina de estados.	Quando for pressionada alguma tecla direcional do teclado, o estado será alterado para o estado "MovePacmanState"
MovePacmanState	Este estado corresponde ao estado onde o movimento do pacman terá inicio.	Este estado é um estado de transição e irá mudar de estado assim que uma contagem definida no sistema seja completa. Quando este "temporizador" atingir o valor definido o estado será alterado para o "MoveGhostState"
MoveGhostState	Este estado corresponde ao estado onde os fantasmas saem da caixa inicial e começam o seu movimento pelo labirinto.	Existem diversas verificações a serem processadas neste estado sendo assim possível seguir o padrão do jogo. Caso o pacman ingira todos os alimentos presentes no labirinto o estado será alterado para o estado "nextLevelState", caso o pacman seja apanhado por um fantasma existe uma operação neste estado onde será verificada a vida do pacman sendo assim possível proceder a mudança de estado correta, ou seja, caso a sua vida seja 0 este irá mudar de estado para o "GameOverState", caso contrário o estado será novamente o "StartGameState". Outra das verificações necessárias neste estado é saber se o pacman ingeriu algum alimento especial, caso sim então o estado será alterado para o "VulnerableGhostState"

VulnerableGhostState	Estado onde os fantasmas ficam vulneráveis e é possível eliminá-los.	Durante este estado apenas existem duas mudanças de estado possíveis sendo estas para o "MoveGhostState" onde os fantasmas já não são vulneráveis ao pacman. A primeira maneira de mudar de estado é matando todos os fantasmas do labirinto e a segunda maneira de mudar de estado é acabando o tempo predefinido na classe deste estado.
GameLevelState	Durante este estado será verificado se o jogo já acabou ou ainda é possível jogar mais níveis.	Caso o nível de jogo seja inferior a 20 então o estado será alterado para o "StartGameState" onde o jogo será reiniciado com as características do novo nível selecionado. Caso o nível de jogo seja igual a 20 então o estado será alterado para o "GameWinState" onde o programa terá o fim.
GameWinState	Durante este estado o jogo será terminado como vitoria e assim será guardado no top 5 caso a pontuação o permita.	N/A
GameOverState	Este estado termina o jogo como derrota. O facto de existir um estado só para isto irá simplificar a futura interface gráfica a ser implementa pois será mais fácil realizar as validações por estado.	N/A