isec Engenharia

Meta 2

Programação Avançada

Daniel Bravo – 2021137795

Índice

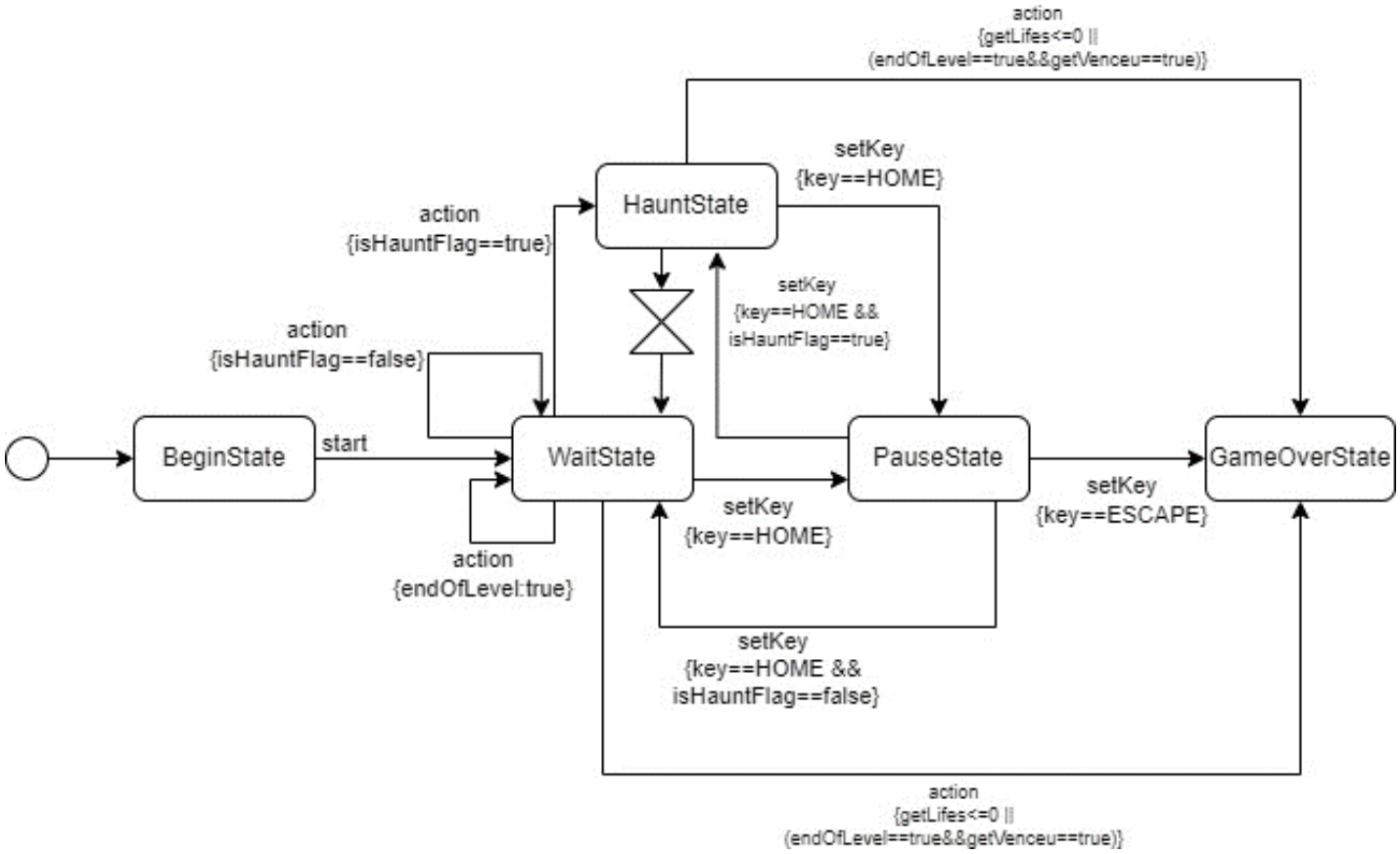
Descrição das opções e decisões tomadas

Este trabalho consiste na implementação do jogo Pacman em Java, sendo nesta meta usado uma interface gráfica usando JavaFX.

Na minha implementação os objetos para além de se encontrarem na Maze são também guardados numa arraylist o que permite diminuir o número de iterações dos for's e for each assim como contribuindo para uma melhor legibilidade do código.

O menu existe à parte da máquina de estados sendo que quando o usuário seleciona para começar o jogo é criado um RootPane(Pane) que cria um pane com a ui correspondente ao diferentes estados. Ao longo do jogo apenas uma pane com a ui vai estar visível sendo a ui correspondente ao estado naquele momento.

Diagrama de Estados



Descrição das Classes

Blinky, Clyde, Inky, Pinky

Representam respetivamente os diferentes tipos de fantasma presentes no jogo e guardam o seu caracter representativo, assim como as suas coordenadas

GameData

Classe que possui toda a lógica do jogo e que é dona de todos os objetos.

IGhost

Base das classes fantasma

MazeElement

Classe que implementa a classe IMazeElement de forma a permitir o armazenamento de um char

Pacman

Representa o pacman e guarda o seu caracter representativo, assim como as suas coordenadas

IGameState

Classe interface dos estados

GameStateAdapter

Classe que implementa *IGameBWState* que é depois extendida pelos outros estados

GameState

Enum que contem o nome dos diferentes estados e responsável por os criar

Direction

Enum que contem as diferentes direções possíveis

BeginState

Classe que representa o estado responsável por iniciar um jogo ou um novo nível

GameOverState

Classe que representa o estado responsável por terminar o jogo

HauntState

Classe que representa o estado no qual após o pacman comer a fruta os fantasmas ficam vulneráveis

MenuState

Classe que representa o estado responsável pelo menu inicial

PauseState

Classe que representa o estado no qual o jogo se encontra em pausa

WaitState

Classe que representa o estado no qual o jogo está em ação, mas os fantasmas não se encontram vulneráveis

Player

Classe que representa um jogador onde é guardado o seu nome e pontuação

Top5

Classe que guarda o Top5 e que é usada para ler e gravar o top5 num ficheiro

GameOverUI

Classe gráfica que representa no ecrã o estado de final de jogo

WaitUI

Classe gráfica que representa no ecrã o estado em que o jogo está a decorrer, mas os fantasmas não estão vulneráveis

HauntUI

Classe gráfica que representa no ecrã o estado em que o jogo está a decorrer, mas os fantasmas estão vulneráveis

PauseUI

Classe gráfica que representa no ecrã o estado de pausa

BeginUI

Classe gráfica que representa no ecrã o estado de inicio de jogo ou mudança de nível

PointsLifes

Classe gráfica que coloca na parte de baixo da aplicação informação relativa ao número de vidas e a quantidade de pontos obtida

RootPane

Classe gráfica que é a raiz da ui do jogo possuindo a ui correspondente aos diferentes estados

StartMenuUI

Classe gráfica que é a raiz da ui da aplicação onde pode apresentar o Top5 ou o jogo

CSSManager

Classe que aplica um ficheiro CSS a uma interface gráfica

ImageManager

Classe que permite aplicar imagens a uma interface gráfica

Direction

Enum que contem as diferentes direções possíveis

Descrição do relacionamento das classes

A interface *IGameState* é implementada pela classe *gameStateAdapter*, que é extendida pelas classes que representam estados da máquina de estados: *BeginState*, *GameOverState*, *HauntState*, *MenuState*, *PauseState*, *WaitState.*

Estes estados são criados pela classe *GameState* e geridos pela classe *GameStateAdapter*.

A classe *GameContext* tem acesso a estes estados através da interface *IGameState*. Para alem de ter acesso a estes estados também tem acesso á *GameData* que contém toda a logica do jogo assim como é dona das classes *Pacman*, *Blinky*, *Pinky*, *Clyde* e *Inky*, onde estas quatro últimas possuem todas a mesma interface.

A classe *GameManager* permite a ligação entre a interface Gráfica e a máquina de estados sendo dono da *GameContext*.

Na parte gráfica temos a classe *StartMenuUI* que é a raiz da UI do programa onde ao iniciar o jogo cria um *RootPane*.

O *RootPane* é dona das classes *GameOverUI, WaitUI, HauntUI, PauseUI* e *BeginUI.*

Funcionalidades Implementadas

- Ecrã inicial
- Controle do personagem Pac-Man com teclado
- 4 fantasmas com comportamentos diferentes
- Vários labirintos com níveis de dificuldade crescente obtidos através da leitura de ficheiros
- Zona Warp para teletransporte do Pac-Man
- Bolas para recolher e ganhar pontos
- Frutas que aparecem a cada 20 bolas recolhidas para ganhar pontos extras
- Bolas comestíveis que tornam os fantasmas vulneráveis e dão pontos
- Efeito de vulnerabilidade dos fantasmas e velocidade vai aumentando conforme o nível
- Opção de pausar jogo
- Aviso de Vitória
- Top5
- Implementação de interface gráfica
- Gravação e restauro do Jogo em ficheiro binário

Funcionalidades não implementadas

- N/A