Manual do Sistema

Maíra de Souza Canal - Número USP: 11819403

Janeiro 14, 2022

1 Introdução

Esse manual de sistema aborda, de forma sucinta, o funcionamento de cada uma das partes do *software*, buscando explicar a estratégia explicada e as dificuldades de implementação.

2 Engine

A Engine é responsável pela gerência da linha de execução do jogo e pela interação entre a abstração do jogo, um grafo, e a interface gráfica.

A Engine é composta pelas seguintes classes: Controller, Engine, GameRules, GameStatus e MenuController. A linha de execução do jogo e as regras do jogo encontram-se nas classes Engine, GameRules e GameStatus, enquanto as classes Controller e MenuController são responsáveis pelo gerenciamento da interface gráfica.

A classe MenuController possui, como única atribuição, lidar com o botão "Play" do menu principal. Esse botão leva o usuário ao jogo e, portanto, o *handle* desse botão realiza a mudança do *stage*.

A classe Controller é a classe que concentra todos os principais componentes do jogo, isto é, a Engine, o Render e o Map. Nesse sentido, essa classe, além de lidar com os botões de "Pause" e de "Reset", controla o loop principal do jogo, coordenando a Engine e o Render. Essa classe possui um *Timer*, que executa um loop, que encontra-se na Engine, e uma renderização, que encontra-se no Render, do jogo de acordo com o *framerate* determinado.

A classe Engine possui o mapa do jogo, as entidades e o conjunto de regras do jogo, a classe GameRules. A cada loop do jogo, a Engine movimenta todas as entidades do jogo e aplica as regras do jogo por meio dos métodos da classe GameRules.

A classe GameRules possui todas as regras do jogo e executá-as no mapa e nos atributos do jogo, a classe GameStatus. Essa classe é responsável por contabilizar pontos, as vidas do Pacman, o funcionamento das pílulas de energia e o nível do jogo.

A classe GameStatus é uma classe que possui todos os seus atributos e métodos *static*, sendo o local que as informações do jogo que são armazenadas e acessadas por diversas classes. Nessa classe, são armazenados a quantidade de pontos e de vidas, o nível, a direção do Pacman e outras *flags* relevantes, que indicam o fim do jogo ou a passagem de nível.

3 Render

O Render é responsável pela interface gráfica do jogo, sendo composta pelos arquivos FXML e pela classe Render.

A classe Render é um componente gráfico, composto por uma matriz dos componentes *ImageView*. Cada componente *ImageView* representa um nó do tabuleiro, ou seja, um nó do grafo.

O *ImageView* é capaz de renderizar uma imagem e, por isso, foi feito um HashMap que cria uma paleta de correspondência entre imagens e identificadores do jogo.

Sendo assim, a interface gráfica do jogo é, essencialmente, uma matriz de imagens, que são atualizadas a cada loop do jogo.

Essa abordagem possui uma limitação: a incapacidade de representar dois elementos simultaneamente em um único nó. Isso, por vezes, prejudica a experiência visual do usuário, já que um elemento pode se "esconder" atrás de outro, dando a impressão que o elemento sumiu do mapa.

Para resolver isso, seria necessário que a renderização das entidades fosse baseada na classe *javafx.scene.shape.Rectangle* e nas posições x e y. Essa abordagem não foi utilizada porque necessitaria de muitas modificações na Engine construída na parte 1 do projeto.

4 GameElements

Os elementos do jogo são as abstrações que constituem o tabuleiro e as entidades do jogo.

O tabuleiro do jogo é construído por meio de um arquivo-texto, que possui o identificador de cada nó. A partir de tal arquivo, é construído um grafo, que encontra-se na classe Map, e é constituído da classe Node. Cada nó pode possuir entidades e consumíveis dentro dele, além de possuir uma informação de distância, utilizada para movimentação dos fantasmas. As distâncias são calculadas na Engine por meio de um algoritmo de busca em largura.

Todas as entidades do jogo herdam a classe abstrata Entity, que possui, como atributos, um identificador, a velocidade e o nó em que se encontra a entidade, e, como métodos abstratos, um método para nascer, mover e morrer.

O Pacman, representado pela classe Pacman, nasce sempre na mesma posição e se move de acordo com a entrada de teclado do usuário. Além disso, o Pacman não possui nenhuma rotina específica para morrer.

Os fantasmas herdam a classe abstrata Ghost, que, por sua vez, herda a classe abstrata Entity. Os fantasmas possuem comportamentos diferentes em cada fase de sua vida e, por isso, o método de movimento de cada fantasma é distinto e é baseado nas *flags*: *isChasing*, *isLeavingHome*, *isGoingHome* e *isRunningAway*.

Quando a *flag isChasing* é *true*, o fantasma adota o seu comportamento de movimentação pelo tabuleiro. Os fantasmas Blinky e Pinky possuem o comportamento de perseguir ao Pacman. Para perseguir o Pacman, eles movem-se para o nó que possui a menor distância em relação ao Pacman. Por outro lado, os fantasmas Inky e Clyde movem-se aleatoriamente, utilizando a classe *java.util.Random*.

Quando a *flag isLeavingHome* é *true*, o fantasma adota o seu comportamento de saída de sua casa até chegar no labirinto. Para realizar isso, os fantasmas perseguem o primeiro nó que há fora de casa. Esse comportamento faz parte do processo de nascer do fantasma.

Quando a *flag isGoingHome* é *true*, o fantasma adota o seu comportamento de ir até sua casa, após a morte. Para realizar isso, os fantasmas perseguem um nó que existe dentro de casa.

Quando a *flag isRunningAway* é *true*, todos os fantasmas começam a vagar aleatoriamente pelo tabuleiro. Isso ocorre quando o Pacman come as pílulas de energia e os fantasmas ficam azuis.

Para controlar a velocidade das entidades, criou-se um contador, que é incrementado de acordo com a velocidade da entidade. Quando o contador chega a 1000, a entidade se move.

O fantasma Blinky aumenta de velocidade à medida que a quantidade de pontos aumenta. Para isso ser executado, a medida que a quantidade de pontos aumenta, o incremento ao contador do Blinky torna-se maior. Para que o Blinky não fique com uma velocidade maior que o Pacman rapidamente, a velocidade inicial do Blinky é menor que as dos demais personagens.