

Manual do Sistema - Space Invaders: Doomed Edition

Aluno: João Pedro Gonçalves Ferreira

Nº USP: 12731314

Professor: Robson Leonardo Ferreira Cordeiro

Introdução

Esse projeto faz parte da disciplina SCC604 - Programação Orientada a Objetos. O Sistema do jogo foi modularizado basicamente com 5 packages (pacotes), sendo eles: GUI, GameElements, GameEngine, Images e space.invaders, sendo que cada um desses pacotes será explicado a seguir.

1. GUI

O primeiro pacote a ser explicado será o GUI, que significa “Guided User Interface”, e nele teremos duas classes que implementam toda a interface gráfica do jogo e a interação do usuário com ela, a classe GUI e a classe Screen.

A classe GUI cria um Canvas, que cria a janela do jogo, além de desenhar todos os objetos do jogo. Ainda dentro da GUI, criamos um objeto jogo e damos início a ele, com os métodos implementados na classe Game. Além disso, para o loop principal do jogo, criamos uma InnerClass do tipo AnimationTimer que basicamente irá limpar a tela do jogo e receber os comandos/inputs do jogador, gerencia também a movimentação dos aliens e a informação que será mostrada após ganhar ou perder o jogo.

A classe Screen é responsável por gerenciar o papel de parede do jogo e tem métodos que mostram mensagens ao jogador ao ganhar ou perder o jogo.

2. *GameElements*

O pacote *GameElements* agrega todas as entidades do jogo, possuindo as classes *Alien*, *Barrier*, *Entity*, *Shot* e *Spaceship*.

A classe *Alien* é uma especialização da classe *Entity* e implementa os métodos de movimentação, atirar, ser morto, obter tipo, obter direção de cada alien, e outros. Além disso, no jogo possuímos três tipos de aliens, cada um, ao ser eliminado, dará uma pontuação diferente. os aliens mais distantes dão mais pontos, e os mais pertos da nave, menos pontos.

O método de movimentação dos aliens está em função de um valor booleano, que, se for *true*, entramos na condição de movimentação para esquerda e, se for *false*, os aliens vão para direita.

A classe *Barrier* é uma especialização da classe *Entity* e possui métodos próprios para destruir a barreira quando necessário, criá-la e, além disso, a barreira possui um contador de dano, que só permite que a barreira seja destruída caso ela tenha tomado 3 ou mais tiros, seja dos aliens ou do jogador.

A classe *Entity* é abstrata e possui métodos genéricos que todas as outras entidades do jogo precisam ter, como métodos de locomoção, ter posição, obter e colocar em uma posição, além de outros.

A classe *Shot* também é uma especialização da classe *Entity* e possui todos os métodos necessários para a implementação do tiro, tanto do jogador quanto dos aliens, seu construtor carrega sua imagem *.png* e de quem será o tiro.

A classe Spaceship é a última das especializações da classe Entity, e possui métodos de locomoção, que nesse caso, é só horizontal, além de um método hit, que decrementa a vida do jogador, caso a nave seja atingida por um tiro.

3. *GameEngine*

Esse é o maior e mais completo pacote. nele, temos todo o funcionamento e lógica do jogo, nela cria-se todas as entidades, define-se se o jogo acabou ou não, além de outras funções.

A classe Game é responsável por comandar a criação, renderização, disparo, colisão de entidades, etc. ela é o motor principal do jogo e possui também funções mais secundárias, como “getters” e “setters” para obter vida da nave, score, cria a quantidade necessária de aliens, possui a thread sleep, gerencia a pontuação do jogo, etc.

4. *Images*

Esse é o pacote que armazena todas as imagens utilizadas no jogo, bem como a imagem de fundo, a imagem da nave, as três imagens dos aliens, as imagens das barreiras e dos tiros. Todas as imagens possuem extensão .png.

5. *space.invaders*

Esse é o pacote que abriga a classe principal do jogo, ontem temos o main, que apenas chama a classe GUI para dar início ao jogo.