

MANUAL DO SISTEMA

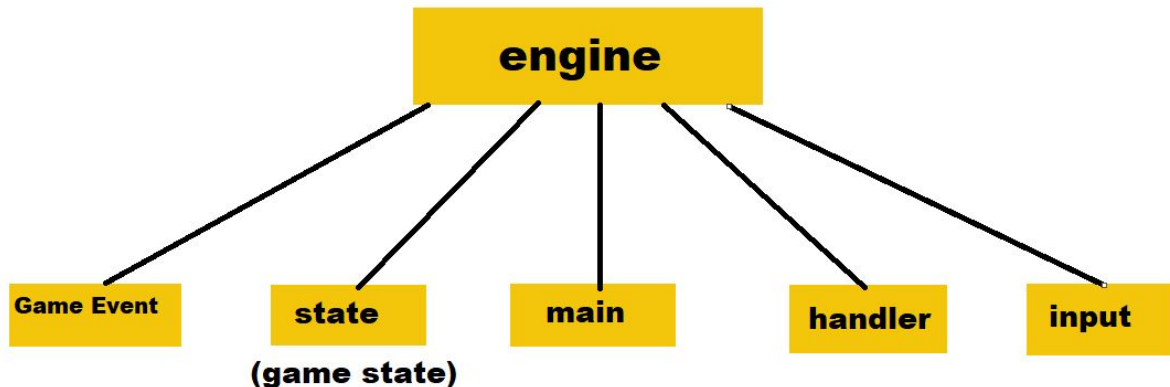
Igor Antunes Boson Paes nºUSP 11200571

Disciplina POO - SSC0604 - Prof. Robson L. F. Cordeiro

Este manual aborda como foi feita a implementação do projeto SpaceInvaders em java

* Observação: Os diagramas mostrados são apenas para ilustração, a intenção deles não é ser uma representação fiel, à risca, do sistema, e sim uma forma de facilitar a explicação.

1.Engine



A engine é responsável por fazer o jogo rodar com sucesso, e cuida da parte mais técnica do programa.

O main é quem produz a janela do jogo, e possui o loop principal do jogo, que pratica a função de `update()` e `render(GraphicsContext)`, que atualizam e renderizam os objetos do jogo.

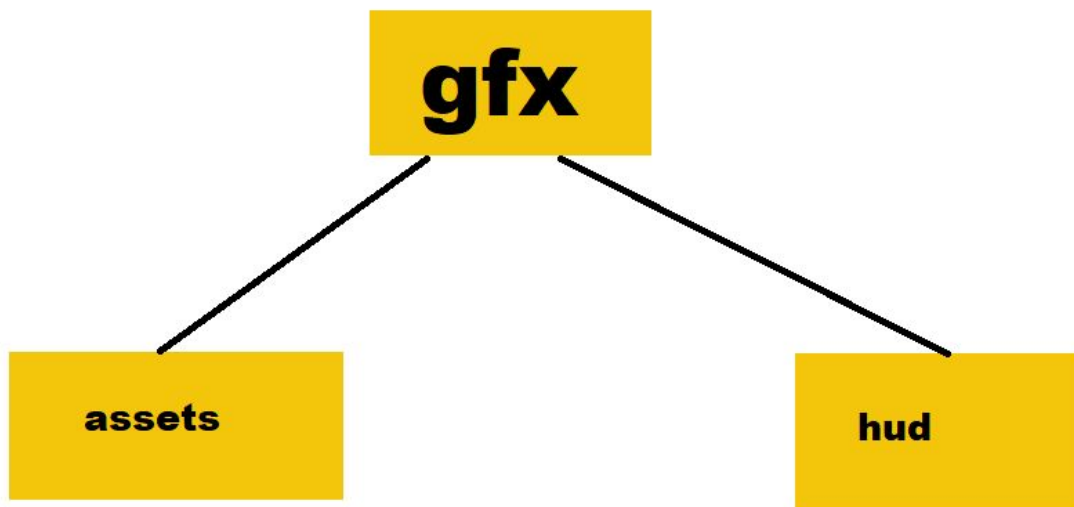
Esse loop é acessado pelo input, que recebe a entrada do teclado. O objeto do input é passado para o handler, para ser utilizado pelo resto do sistema.

O handler é uma classe simples, mas também muito importante, que serve para passar objetos entre outros objetos dentro do sistema, através de vários setters e getters.

O GameState é uma classe muito importante pro jogo, que serve para instanciar todas as entidades, e outras classes que fazem o jogo funcionar. Ele também é responsável por rodar as funções `update` e `render` desses objetos.

O game event possui as classes `GameOver` e `YouWin`, que são efetivadas quando o jogo deve acabar.

2. Parte gráfica



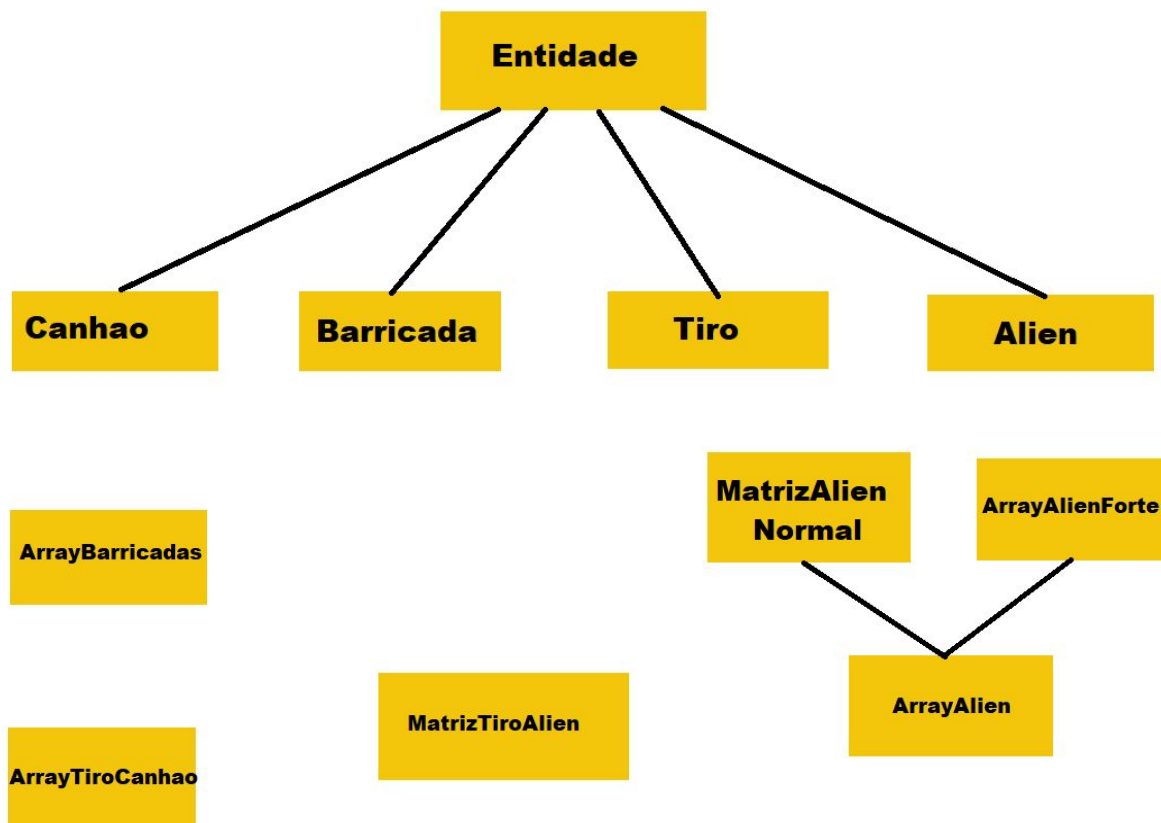
Observação: a renderização dos objetos do jogo, como os aliens, não é feita aqui. A renderização desse tipo de objeto é feita individualmente, na função `render(GraphicsContext)` de cada classe.

Na parte gráfica, o `assets` é responsável por acessar os arquivos de imagem e som dentro do projeto, na pasta `/res`, e disponibilizar eles ao resto do sistema através do handler.

Na `hud`, há coisas puramente gráficas, como o display de vidas e o background, que são classes cuja única função é renderizar algo útil na tela.

A música do jogo é Gerudo Valley, The Legend of Zelda: Ocarina of time
(<https://www.youtube.com/watch?v=0hEYvdMoF2g>)

3. Entidades do jogo



A parte das entidades do jogo é dividida principalmente em duas:

- As entidades em si, como tiros e os aliens
- Os arrays correspondentes, que são responsáveis por instanciar essas entidades

As entidades herdam a classe `javafx.scene.shape.Rectangle`, que possui a função `intersect`, que é utilizada para analisar a colisão de certas entidades com outras, por exemplo, do tiro do canhão com o alien.

O mais importante a se deixar claro, é como os objetos são criados. Todos os objetos são criados no momento que o jogo começa, contudo, aqueles que não são usados desde o começo, como os aliens forte, que surgem ao longo do tempo, e os tiros, que são usados ao longo do jogo, são criados mas não sofrem `update()`, que é responsável por checar colisão, realizar o movimento, por exemplo, e nem `render()`.

Ao serem criados, eles são criados como “mortos”, e são inutilizados até serem necessários. No caso dos tiros, até eles serem atirados, e no caso do alien forte, ele é spawnado na tela a partir da classe `ArrayAlienForte`. A partir desse momento, eles sofrem o `update()` e são renderizados.

E após morrerem e não serem mais úteis, eles voltam ao estado de “morto”, pronto para serem reutilizados de novo. Desse modo, o sistema “recicla” os objetos que ele cria ao ser inicializado.

Observações adicionais:

Ainda há bastante coisa que ainda quero adicionar no jogo, mas por questão de tempo, e demanda da faculdade, não consegui adicionar ainda, mas acredito que vale a pena ao menos citar algumas funcionalidades.

- Faltou adicionar animações gráficas quando os aliens morrem (eu queria que eles explodissem, e não só sumissem) e efeitos sonoros que acompanhassem elas.
- Eu queria fazer um sistema de menu, mesmo que básico, com uma tela inicial adequada, e que desse para mostrar também os créditos.
- Também, um sistema de dados para armazenar a pontuação do jogar, guardando o nome e pontuação dele, e mostrar um ranking das melhores pontuações no menu.
- Algum tipo de menu de seleção para a skin da espaçonave, e a música que o jogo toca

Obrigado Professor!

Igor