



UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Disciplina: Programação Orientada a Objetos

SPACE INVADERS - JavaFX

Manual do Sistema

Pedro Arthur do Prado Franoso - 12547301

Docente responsvel: Prof. Robson Leonardo Ferreira Cordeiro

So Carlos
2 semestre / 2022

SUMÁRIO

1	Main	1
2	Menus	1
3	Classes	1
3.1	Entidades	2
3.1.1	Nave	2
3.1.2	Aliens	2
3.2	Engine	2
4	Interface Gráfica	3

O jogo está separado em 3 pacotes, fora o pacote da main: interface gráfica, as classes e o pacote dos menus. Primeiramente, cabe introduzir a main:

1. MAIN

A main, além de chamar o menu principal que dá sequencia à corrente do jogo, carrega algumas variáveis globais de interesse de todos os pacotes, são elas:

- dimensão vertical da tela (Height)
- dimensão horizontal da tela (Width)
- tamanho base das imagens do jogo (64 pixels)
- vetor que representa as naves possuídas pelo usuário
- nível que o jogador está
- tipo da nave que o jogador está usando
- pontuação do jogador
- método de atualização do arquivo de progresso

O método de atualização do arquivo de progresso é chamado sempre que uma nova nave é comprada dentro da loja ou quando o jogo é fechado. Dessa forma, o arquivo de progresso está sempre atualizado.

Cabe ressaltar que, embora os menus sejam menores, o jogo está definido para sua janela ficar em tela cheia e com dimensões de 1920x1080. Esse fato pode gerar problemas para rodar o jogo em computadores com monitor de diferentes dimensões. Foi uma decisão tomada no início do desenvolvimento e que, na reta final, percebeu-se que talvez fosse melhor optar por dimensões menores. Entretanto, alterando essas variáveis de controle de tamanho base e dimensões, o jogo pode ser facilmente alterado para diferentes monitores.

2. MENUS

O pacote dos menus possuem os arquivos de FXML de cada um dos menus do jogo e os controladores deles. Os controladores alteram as imagens e os efeitos dos elementos do menu e algumas variáveis globais do jogo no caso da loja, que usa as moedas e define a nave que o jogador usará.

O primeiro menu que aparece quando o jogador abre o jogo é o de carregamento de progresso anterior. Esse menu lê um arquivo de salvamento de progresso ("LoadProgress.txt") que carrega o nível, score, moedas e naves adquiridas pelo usuário em uma outra jogatina. Se ele preferir, ele pode sobrescrever esse progresso com um novo jogo.

Muitas vezes, as telas de menu chamam outras telas, reabrindo o arquivo do FXML da tela a ser aberta.

É o pacote mais simples do jogo, tendo em vista que os arquivos de FXML foram construídos através do programa Scene Builder conforme visto em aula.

Dentro desse pacote, encontra-se também uma pasta com todas as imagens usadas na construção dos menus.

3. CLASSES

No pacote das classes, encontram-se todas aquelas que definem objetos relacionados ao funcionamento do jogo, ou seja, as entidades e a própria engine do jogo. Partindo pelas entidades, podemos definir elas da seguinte forma:

3.1. Entidades

Entidades são classes abstratas para elementos do jogo que possuem vida, posição e imagem. Elas não necessariamente se movem. Entre elas, estão:

- o alien especial, que passa periodicamente no topo da tela
- as barreiras que são os meteoros que separam a nave dos aliens
- os projéteis atirados pelos aliens e pela nave do jogador
- os atiradores, que, por sua vez, trata-se de outra classe que engloba mais de um elemento do jogo

A classe de atirador é uma classe abstrata de todas as entidades que atiram; engloba a nave do usuário e os aliens inimigos.

3.1.1 Nave

A nave do jogador se movimenta através dos comandos recebidos do teclado pela tela do jogo, usando a função "setOnKeyPressed" vinculada à "Scene" do jogo, recebida da interface gráfica do jogo.

Existem 4 tipos de nave no jogo, cada uma com características diferentes e especiais entre si. Essas diferenças podem ser encontradas na loja do jogo ou no manual do usuário.

A ideia por trás de inserir diversas naves veio pois havia a necessidade de premiar o jogador de alguma forma após ele acertar a nave especial bem como, devido à forma como a dificuldade do jogo aumenta ao passar dos níveis, sentiu-se a necessidade de que a nave melhorasse também.

Dessa forma, o desejo de testar todas as naves e a forma como cada uma é consideravelmente melhor que a anterior - visível na hora de jogar - gera mais interesse em evoluir no jogo e permite que o jogador alcance níveis mais distantes, ainda que eles fiquem muito difíceis.

A quantidade de tiros do jogador não é limitada, mas ele tem um tempo de recarga até poder atirar novamente, que é controlado pela engine do jogo.

3.1.2 Aliens

Os aliens por sua vez são os responsáveis por gerar a dificuldade do jogo. Dessa forma, sua vida e seu tempo de recarga depende do nível que o jogador está.

Seu tiro ocorre de forma aleatória no que tange ser guiado ou não. O método que faz com que o grupo de aliens atira varre esse grupo todo até que o tempo de recarga seja atingido. Se, nesse momento, o alien que estiver sendo analisado pelo método está alinhado à nave do jogador, ele atira. Caso contrário, um outro alien é selecionado de forma randômica para fazer o tiro. Isso gera uma grande dificuldade na movimentação do usuário, que não pode nem ficar parado e nem se movimentando de qualquer forma, pois precisa desviar dos tiros.

A lista dos 55 aliens é varrida periodicamente dentro de uma thread de forma que assim que algum deles atinge um dos limites da tela, todos os outros também alteram sua direção. Dessa forma, a movimentação vista no jogo original pode ser recriada perfeitamente.

Ao todo, são 3 tipos de aliens que compõem a matriz, sendo a primeira linha de maior vida, duas com uma vida a menos que a primeira e as últimas duas com menos vida.

3.2. Engine

A engine controla toda a movimentação das entidades, constrói o jogo posicionando todos os objetos e criando a interface gráfica, controla colisões removendo as entidades coerentes, limita a região do jogo de acordo com a resolução definida da main, controla o spawn do alien especial e cria e gerencia as threads que controlam e gerem o jogo.

Uma peculiaridade a respeito das threads do jogo é a escolha pela implementação direta dentro da engine, sem uma classe externa, para facilitar o acesso às entidades que foram definidas na engine e porque, conforme observado nos exercícios da lista 12 da disciplina, essa forma apresenta uma execução mais fluída. Como muitas colisões são verificadas em cada Thread e muitas entidades interagem de forma independente a cada ms, optou-se pela implementação nesse formato.

O que define a velocidade do jogo são os tempos de sleep de cada Thread, que são definidos baseado no nível atual do jogo. Além disso, outra coisa que influencia nos tempos de sleep da Thread dos aliens é a quantidade de aliens vivos, de modo que quanto menor a quantidade de aliens vivos, menor será o tempo de sleep do loop da Thread.

4. INTERFACE GRÁFICA

O pacote de interface gráfica contém todas as classes que constituem a parte visual do jogo, como as classes de elementos visuais, efeitos - que são elementos visuais que se alteram passado um tempo - e o objeto tela, que desenha todo o jogo.

A interface é baseada na classe Canvas do JavaFX que, em resumo, funciona como uma tela de pintura onde os objetos podem ser desenhados e sobrepostos sem que se acumulem em camadas, já que as imagens são pintadas na tela.

Por isso, precisa-se que a tela tenha o plano de fundo pintado novamente toda vez que passa pelo loop de execução. Foi usado uma Thread própria para animações (AnimationTimer) e todas as entidades são recebidas via uma função getter da engine, que gera uma lista de todas as entidades do jogo. Com essa lista, a função de interface só precisa desenhar as imagens nas posições varrendo uma única lista, sem precisar visitar todas as listas de entidade definidas na engine de funcionamento do jogo.

Como a tela (gameStage) do jogo fica armazenada na interface, é ela a responsável por finalizar um jogo. Por isso, o método encerraJogo fica na classe de interface ainda que quando ela é chamada o jogo já encontra-se sem funcionamento pela engine.

Em relação aos desenhos usados no jogo, todos foram pegos do GoogleImages e modificados. O jogo se prende a um padrão de cores alinhadas à uma paleta pré-definida.

Diferente do pacote dos menus, que tem uma pasta própria para as imagens, as imagens usadas no jogo em si ficam livres no pacote padrão do projeto para facilitar a implementação do código. Nos menus, isso não acontece pois é fácil acessá-las na configuração do arquivo FXML pelo Scene Builder.