

Manual do sistema

Fabício Samapio - nUSP: 12547423

December 26, 2022

1 Introdução

Esse manual vai abordar de forma bem resumida o funcionamento de algumas classes e funções do jogo, mostrando qual a estratégia que foi empregada.

2 Arquivos e classes

O jogo foi feito usando uma classe para cada objeto que aparece no jogo. Desse modo há uma classe para:

- Nave
- Inimigo
- Tiro
- Mapa
- Inimigo Especial
- Menu
- Barreira

Também há uma classe para o Game Update, que atualiza o jogo a cada ciclo.

2.1 Inimigo

A estratégia usada para lidar com os inimigos foi usar uma matriz de objetos Inimigo de 5x11. Desse modo cada inimigo da matriz possui uma posição, nível e imagem. No game updater há uma parte que move todos os inimigos que ainda estiverem vivos, verificando se algum inimigo vivo já bateu na borda, porque se esse for o caso, ao invés da matriz inteira se mover no eixo horizontal, ela deverá se mover no eixo vertical. É importante ressaltar que tudo que envolve a matriz dos inimigos como, movimentação, tiro, colisão, etc, antes de manipular algum valor nela, é verificado se o inimigo ainda está "ingame".

A movimentação dos inimigos, explicando mais detalhadamente, acontece na função `updateEnemys`, nela, primeiro é colocado um `for` que percorre todos os inimigos vivos da matriz, verificando se algum deles está encostado na borda e, se for mover na horizontal novamente, vai ultrapassar os limites do jogo,. Nesse caso, é retornado uma `flag true`, a qual indica que a movimentação dos inimigos dessa vez vai ser na vertical e mudando o sentido que eles vão se mover, direita ou esquerda. Se não for esse o caso, os inimigos continuam se movendo na horizontal no mesmo sentido que antes.

Os inimigos são divididos em 3 níveis, sendo a primeira fileira nível **1**, as fileiras 2 e 3 nível **2** e as fileiras 4 e 5 nível **3**, sendo que cada nível tem uma imagem para o inimigo diferente e o ganho de pontos ao matar diferente também.

2.2 Tiro

No jogo há apenas dois objetos da classe tiro, um para o tiro da nave e um para os tiros dos inimigos. Cada um desses objetos possui posição e um boolean ingame, que indica se os inimigos ou a nave atirou e o tiro ainda está no mapa. Sendo assim, é apenas possível atira quando o ingame está em false, uma vez que é possível ter apenas 1 tiro no mapa por vez.

2.3 GameUpdate

Tudo que envolve o jogo acontece nessa classe. Então é aqui que os objetos são instanciados, aonde a função que atualiza todos os objetos do jogo está e aonde chama a função de desenhar na tela de cada classe. A função de update está bem dividida, com comentários antes de cada ação indicando o que acontecerá em seguida.

3 Atualização e loop do jogo

O loop do jogo está na main, o loop foi feito usando Timeline. Dentro do loop tem um if que indica se está no loop do menu ou no loop do jogo. No loop do jogo que é o que importa, temos uma função que verifica a entrada no teclado do jogador, a própria função de atualizar o jogo e a função de desenhar os objetos na tela.

4 Tempo no jogo

O jogo roda a 60 fps, então a função de update roda 60 vezes por segundo, sendo assim, a maioria das rotinas de ações na função update possui um if que faz com que aquela rotina não aconteça toda vez que a função update é chamada mas aconteça apenas às vezes, para que o jogo não fique muito rápido e seja possível entender o que está acontecendo no jogo.

5 Render

Para a parte gráfica, foi utilizado o Canvas e para desenhar na tela, foi usado o GraphicsContext. Cada classe tinha sua própria função de desenhar na tela, além de uma posição posX e posY e o size de cada objeto. A função utilizada para desenhar foi a drawImage para isso, cada objeto instanciado tinha a sua própria imagem baseado no seu nível, ou na sua vida.