

INF1350 - Jogo Lua
Livia Aloise - 1510952
Marcelo Costalonga - 1421229

Funcionamento do jogo:

O objetivo do jogo consiste em sobreviver o maior tempo possível, assim como matar o maior número possível de oponentes (o jogador possui apenas um ataque que é atirar em seus oponentes com a tecla 'a'). O jogador começa com 10 pontos de vida, cada tiro que o acerta é descontado 1 ponto, caso ele tome 10 tiros, o jogo acaba. O jogador pode se movimentar para frente, lados e para baixo utilizando as setas do teclado, caso ele saia da tela pela direita ele irá aparecer à esquerda e vice versa (assim como os inimigos).

O jogo começa com 5 inimigos, se o jogador conseguir matar todos eles, irão aparecer 10 inimigos, e caso ele consiga matá-los também, irão aparecer 15 e assim por diante (sempre 5 a mais do que da última vez). A cada onda de oponentes, os inimigos ficam mais fortes, isso quer dizer que são necessários mais tiros para matá-los.

Os inimigos sempre irão aparecer no topo da tela em determinadas fileiras, sua movimentação consiste apenas em andar da direita pra esquerda ou esquerda pra direita (de acordo com a posição vertical, fileira, em que eles aparecem).

Ao longo do jogo irão aparecer itens, aleatoriamente, se o jogador passar por cima deles, o valor de determinados atributos (velocidade de movimento, ou tempo de intervalo entre tiros) irão ser alterados para melhor ou pior. O efeito dos itens varia de acordo com seu formato (indicado na imagem abaixo), os itens irão aparecer depois de um determinado tempo, caso o jogador não os pegue, e enquanto existirem ficaram piscando a uma determinada frequência.



Implementação:

Utilizamos estruturas corrotinas para encapsular e manipular mais facilmente nossos componentes, visando também utilizar mais variáveis locais e menos globais para garantir que o jogo não ficasse travando como ocorreu com nossos primeiros testes.

Há 5 componentes principais no nosso jogo Player, Blips (inimigos), Bullets (tiros do jogador), Attack (tiros dos inimigos) e Items. Todos, com exceção de player, foram encapsulados em estruturas utilizando corrotinas, com uma estrutura similar a que implementamos no exercício 2, dessa forma conseguimos controlar os intervalos da função update() de cada componente, entre outras funcionalidades, como movimentação.

Além disso os componentes Attack e Items, foram divididos em 2 subestruturas de corrotinas cada (AttackList e ItemsGenerator), dessa forma além de manter uma lista com

várias instâncias de itens ou tiros (assim como os Blips que são armazenados em uma lista), também encapsulamos essa estrutura para ter mais controle. No caso, AttackList ficou responsável por fazer o tempo de espera entre os tiros dos Blips e guardava instâncias dos tiros numa lista, cada elemento dessa lista era uma instância da componente Attack, que faz o controle do movimento dos tiros, ou seja, a função update de AttackList faz os blips atirarem e a função update de Attack faz os tiros se moverem, em tempos diferentes. Assim como para o ItemGenerator, que faz mais itens surgirem cada vez que executa sua update() e guarda as instâncias de Items numa lista, enquanto que o componente Items em si, faz o controle de quanto tempo um item deve existir, se o jogador já passou pelo item, piscar e etc.

No caso de Bullets (ataque do jogador) não foi preciso utilizar uma estrutura mais complexa como essas, pois o ataque era feito apenas quando o botão 'a' é apertado, além disso só existe um único jogador a todo momento, o que torna seu controle mais fácil.

Para tratamento das colisões do jogador (tanto com itens quanto com tiros), apesar de sua imagem se der uma nave, consideramos sua área de colisão como um retângulo, para facilitar os cálculos.

Telas do Jogo:

Ao começar o jogo o jogador poderá escolher entre começar a partida ou ver as instruções. Durante a partida se o jogador escolher pausar o jogo irá ver a tela de pause e se perder o jogo verá a de game over.



