

UNIVERSIDADE FEDERAL DE MINAS GERAIS

VINÍCIUS GABRIEL SILVA FERREIRA

Trabalho Prático : R-Type

Belo Horizonte

2022

1. Introdução

O trabalho prático da disciplina de Programação e Desenvolvimento de sistemas 1 consiste em criar uma versão do jogo R-Type. Na presente versão, o jogador controla uma nave estelar, a Swordfish, ele deve navegar pelo terreno e abater os inimigos com os tiros da nave enquanto toma o cuidado para não colidir com os inimigos nem com os obstáculos do cenário. A nave possui dois tipos de disparo. O tiro básico: que é destruído assim que colide com qualquer objeto. E o tiro carregado: que não é destruído ao colidir com os inimigos, podendo matar vários de uma vez. Além disso, o jogador recebe pontos por cada inimigo morto, quanto maior o inimigo, mais pontos são recebidos.

2. Controles

Movimentação da nave:

W: Movimenta a nave para cima

A : Movimenta a nave para a direita

S : Movimenta a nave para baixo

D : Movimenta a nave para a esquerda

Tiro:

básico: apertar e soltar a tecla ESPACO

carregado: segurar a tecla ESPACO até que o raio do tiro atinja o seu valor máximo.

3. Implementação

3.1 Estruturas de Dados

- **Circulo e Retângulo:** formas geométricas criadas principalmente para a facilitação da validação de hitbox e colisões do jogo.
- **Tiro, Nave, Bloco e Inimigo:** São as entidades de fato usadas no jogo, que aparecem e interagem com o jogador e entre elas.

3.2 Funções e Procedimentos

- ***void initTiro(Nave * nave)*** : Inicia o tiro numa posição pouco a frente da nave.
- ***void initNave(Nave * nave)*** : Inicia a nave do jogador, com sua hitbox, posição, velocidade etc.
- ***void initBloco(Bloco * bloco)*** : Inicia o bloco distante da tela, setando suas características com valores aleatórios.
- ***void initInimigo(Inimigo * inimigo, int id)*** : Inicia um dos inimigos distante da tela, setando seus atributos com valores aleatórios.
- ***void initInimigos(Inimigo inimigos[QUANTIDADE_INIMIGOS])*** : Inicia o vetor de inimigos, definindo o id de cada um.
- ***void destroiTiro(Nave * nave)*** : Destrói o tiro (somente chama a função de inicialização, mas desta maneira é mais semântico)
- ***void destroiInimigo(Inimigo * inimigo)*** : Destrói um inimigo (somente chama a função de inicialização, mas desta maneira é mais semântico)
- ***void desenhaCenario()*** : desenha o cenário a partir da cor definida globalmente..
- ***void desenhaTiro(Tiro tiro)*** : desenha o tiro, um círculo vermelho na posição da nave.
- ***void desenhaNave(Nave nave)*** : desenha a nave a partir da imagem obtida, setando o tamanho e posições da figura.
- ***void desenhaBloco(Bloco bloco)*** : desenha o bloco, um retângulo branco na posição definida.
- ***void desenhaInimigo(Inimigo inimigo)*** : desenha o inimigo, um círculo com raio, posição e cor definidas pelo parâmetro.
- ***void desenhaInimigos(Inimigo inimigos[QUANTIDADE_INIMIGOS])*** : percorre o array de inimigos e desenha cada um deles.
- ***void atualizaTiro(Nave * nave)*** : atualiza o tiro de acordo com seus atributos, aumentando o raio enquanto o tiro está sendo carregado e atualizando a posição quando o tiro for disparado.

- ***void atualizaHitboxNave(Nave * nave)*** : atualiza os dados da hitbox da nave de acordo com sua posição.
- ***void atualizaNave(Nave * nave)*** : atualiza a nave de acordo com sua posição e movimento.
- ***void atualizaBloco(Bloco * bloco)*** : atualiza o bloco de a partir de sua posição e velocidade.
- ***void atualizaInimigo(Inimigo * inimigo)*** : atualiza o inimigo de acordo com sua posição e velocidade.
- ***void atualizaInimigos(Inimigo inimigos[QUANTIDADE_INIMIGOS])*** : percorre o array de inimigos, atualizando cada um deles.
- ***Retangulo criaRetangulo(int x, int y, int largura, int altura)*** : Função utilitária para a criação de um retângulo a partir de seus atributos
- ***Circulo criaCirculo(int xCentro, int yCentro, float raio)*** : Função utilitária para a criação de um círculo a partir de seus atributos.
- ***float calculaDistanciaPontos(int xPrimeiro, int yPrimeiro, int xSegundo, int ySegundo)*** : Função utilitária que retorna a distância entre dois pontos a partir de suas coordenadas.
- ***int validaColisaoCirculoRetangulo(Circulo circulo, Retangulo retangulo)*** : Função utilitária que valida a colisão entre um círculo e um retângulo, para se generalizar e reutilizar a validação.
- ***int validaColisaoCirculos(Circulo primeiro, Circulo segundo)*** : Função utilitária que valida a colisão entre dois círculos, para se generalizar e reutilizar a validação.
- ***float validaColisaoCirculoInimigos(Circulo circulo, Inimigo inimigos[])*** : Função que valida a colisão entre um círculo qualquer e cada um dos inimigos, caso a colisão aconteça: o inimigo é destruído e seu raio é retornado, caso não haja colisão, retorna falso.
- ***void validaColisaoRetanguloInimigos(Retangulo retangulo, Inimigo inimigos[])*** : Valida a colisão entre um retângulo qualquer e cada um dos inimigos, destroi o inimigo caso haja colisão.
- ***int validaColisaoNave(Nave nave, Bloco bloco, Inimigo inimigos[])*** : Valida as colisões entre a nave e os outros objetos usando as funções utilitárias.

- ***void validaColisaoTiro(Nave * nave, Bloco bloco, Inimigo inimigos[]) :***
Valida a colisão entre o tiro e os outros objetos do jogo, tomando as ações caso haja colisão, como destruir o tiro e aumentar a pontuação caso algum inimigo seja destruído por ele.
- ***int validaColisaoInimigo(Inimigo primeiro , Inimigo segundo) :*** Valida a colisão entre dois inimigos diferentes.
- ***void validaColisaoEntreInimigos(Inimigo inimigos[]) :*** Percorre o array de inimigos e valida a colisão entre eles, destruindo ambos caso a colisão aconteça.
- ***void validaColisaoInimigos(Inimigo inimigos[], Bloco bloco) :*** Valida a colisão entre cada um dos inimigos e os outros objetos do cenário.
- ***void manipulaEventoMovimentacaoNave(int tipoEvento, int teclaEvento , Nave * nave) :*** Altera o valor dos atributos relacionados a direção de movimento da nave a partir dos eventos de teclado.
- ***void manipulaEventoTiro(int tipoEvento, int teclaEvento, Nave * nave) :***
Altera os atributos relacionados a carga e disparo do tiro de acordo com os eventos de teclado.
- ***void manipulaEventoTeclado(int tipoEvento, int teclaEvento, Nave * nave) :*** Centraliza as manipulações de evento do teclado.
- ***void exibePontuacaoParcial(ALLEGRO_FONT *fonte, int pontuacao) :***
Exibe um texto no canto superior esquerdo da tela, informando a pontuação atual do jogador.
- ***int obterRecorde() :*** Lê o recorde atual no arquivo **recorde.txt** e o retorna.
- ***void escreverRecorde(int recorde) :*** Escreve o novo recorde do jogador no arquivo **recorde.txt**.
- ***void exibePontuacaoFinal(ALLEGRO_FONT *fonte, int pontuacao) :***
Exibe a pontuação final e o recorde do jogador, caso o jogador bata o recorde, o novo valor é guardado no arquivo.
- ***int main(int argc, char **argv) :*** função principal onde o jogo de fato acontece, inicializa o allegro e os objetos do cenário, centraliza a chamada e integra a cada uma das outras funções do jogo, ao final, destroi e desaloca os recursos utilizados.