

Relatório do jogo:

Como usar o jogo: Iniciar o executável do jogo. Após isso, o jogo deve inicializar no menu principal com as seguintes opções: Start (iniciar), Continue (continuar), Load (carregar), Highscores (pontuações), Settings (opções), Exit Game (sair).

Start: Inicia um jogo normal, tentar procurar para um arquivo com o nome level e o número do nível, caso não ache, cria um nível aleatório. O jogo continua até o jogador morrer. Caso a pontuação final seja maior que a menor pontuação salva, o jogador poderá inserir seu nome.

Controles:

A tecla ESC pausa o jogo e é possível selecionar as opções mostradas.

1 Jogador: O tanque pode ser controlado por WASD ou as Setas, e dispara com J ou Espaço.

2+ Jogadores: O jogador 1 fica com WASD e J e o Jogador 2 fica com as Setas e Espaço, os demais jogadores são mapeados para os controles conectados ao computador.

Continue: Caso o último jogo tenha sido terminado pela opção "Save and Exit", essa opção se torna disponível, continuando o jogo exatamente de onde o jogador pausou-o.

Load: Carrega uma tela para inserir o nome do arquivo da fase sem extensão, caso esse arquivo exista e seja válido, carrega essa fase para ser jogada infinitamente. Esse modo não impacta a pontuação.

Highscores: Carrega uma tela que mostra as maiores pontuação já feitas no jogo.

Settings: Carrega uma tela com diversas configurações, incluindo a resolução, quantidade de jogadores, tema do jogo (claro ou escuro) e quantidade escolhida de offset para os controles.

Exit Game: Fecha o jogo.

Estruturas:

O jogo contém somente 4 grandes estruturas que são utilizadas durante todo seu funcionamento, sendo elas, *Obj*, *Setti*, *Textus* e *SFX*.

Essas serão descritas abaixo em ordem crescente de seu tamanho.

SFX: Estrutura utilizada para armazenar os sons utilizados no jogo, foi criada para ser passada como parâmetro de outras funções mais facilmente, evitando passar cada som individualmente. Ela contém os seguintes tipos: 5 "Sound". Sound é uma estrutura da própria Raylib que armazena um arquivo de som que pode ser tocado a qualquer momento.

Textus: Estrutura utilizada para armazenar as texturas utilizadas no jogo, foi criada para ser passada como parâmetro de outras funções mais facilmente, evitando passar cada textura individualmente. Ela contém os seguintes tipos: 9 "Texture2D". Texture2D é uma estrutura da própria Raylib que armazena a informação de textura carregada a partir de uma imagem.

Setti: Estrutura utilizada para armazenar as configurações e informações entre "telas" do jogo, utilizada para não ter que passar cada um de seus 21 parâmetros individualmente. Ela contém os seguintes tipos: 9 Int, 6 Bool, 2 "Color", 2 Strings, 1 Unsigned Long e 1 Double. A função de cada um destes será descrita logo abaixo na ordem que eles aparecem na estrutura:

int players: Armazena a quantidade de jogadores escolhida.

int level: Armazena o nível do jogo.

int score: Armazena a pontuação total entre "telas" do jogo.

int select: Armazena qual opção foi selecionada em algum menu.

bool won: Diz para o jogo se o nível foi finalizado pela vitória ou derrota do jogador.

bool quit: Booleana para fechar o jogo, assim podendo vir de qualquer tela.

Color theme: Armazena a cor do fundo do jogo, podendo ser preto ou branco.

Color lettercolor: Armazena a cor das letras, varia conforme a cor do fundo.

int extended: Armazena o offset dos controles para multijogador maior.

bool pause: Booleana para saber se o jogo está pausado ou não.

bool exitgame: Booleana para saber se o jogador quer ir para o menu principal.

int pauseselect: Armazena a opção do menu de pausa, assim não resetando-o.

bool loadgame: Booleana para saber se o jogo está carregando ou criando fase.

int foundplayerposition: Armazena quantos jogadores foram encontrados no mapa.

int enemiesremaining: Armazena quantos inimigos já foram mortos e estão no mapa.

bool filenamefound: Booleana para não carregar mapa caso o nome não seja encontrado.

int lowscore: Armazena a pontuação mais baixa encontrada, para comparar com o do jogador.

unsigned long errortime: Armazena o tempo que a mensagem de erro deve ficar na tela.

double realTime: Armazena os segundos desde que uma tela foi iniciada.

char error[100]: String para armazenar a mensagem de erro.

char loadgamenname[100]: String para armazenar o nome do arquivo de fase que deve ser aberto

Obj: Estrutura utilizada para a maioria dos objetos do jogo, como os jogadores, os inimigos, as balas e a energia. Ela contém os seguintes tipos: 5 Int, 4 "Vector2", 3 "Rectangle", 2 Bool, 2 Float, 2 "Sound", 2 Unsigned Long, 1 "Vector4" e 1 "Color". A função de cada um destes será descrita logo abaixo na ordem que eles aparecem na estrutura:

int id: Número do objeto.

Vector2 pos: Posição do objeto no espaço do jogo, tem X e Y.

float ratio: Escala de diferença entre o tamanho vertical e horizontal da imagem.

Vector2 cen: Posição do centro do Objeto, tem X e Y.

Vector2 draw: Posição de desenho do objeto no espaço do jogo, tem X e Y.

int health: Armazena a vida do objeto.

int rot: Armazena a rotação do objeto.

int score: Armazena a pontuação do objeto, utilizada de diferentes formas.

unsigned long time: Armazena o tempo individual do objeto.

unsigned long death: Armazena o tempo individual do objeto, normalmente para o tempo que está morto.

float speed: Armazena a velocidade do objeto.

bool ammo: Diz se o objeto pode atirar ou não.

Rectangle sourceRec: Retângulo para criação da textura do objeto.

Rectangle colRec: Retângulo para a colisão com objetos.

Rectangle drawRec: Retângulo para o desenho da textura na tela.

Vector4 colSide: Vector4 que indica de qual lado existe uma colisão com o objeto.

X = cima, Y = direita, Z = baixo, W = esquerda.

Color color: Cor para a dificuldade dos inimigos e multijogador.

bool dying: Diz se o objeto está morrendo, para animação de morte.

Vector2 deathpos: Armazena a última posição antes da morte, para animação.

int deathtimer: Armazena o tempo da animação de morte.

Sound soundEffect: Armazena o efeito sonoro individual do objeto.
Sound soundEffect2: Armazena o efeito sonoro individual do objeto.
double realTime: Armazena o tempo real, para delay de tiro.