

# COMBAT - Documentação

**Nome:** Júnio Veras de Jesus Lima

**Matrícula:** 2020054617

**Curso:** Sistemas de Informação

## Introdução:

O trabalho consiste na criação de um jogo de mecânica parecida com a do jogo Combat, através do uso da linguagem de programação C e da biblioteca Allegro 5. No jogo Combat existem 2 tanques e obstáculos, o objetivo dos tanques é eliminar o tanque inimigo através de tiros.

## Funcionamento do Jogo

Primeiramente temos a tela de menu, onde existem os botões de play, easy, medium, hard e options.

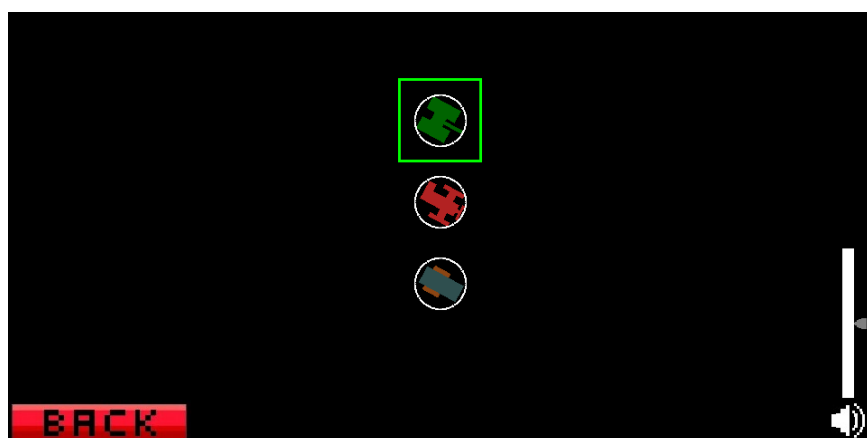


Play: O botão play inicia o jogo com as configurações que o usuário colocou.

Dificuldades: Os botões referentes às dificuldades alteram o mapa do jogo para um mais complexo de acordo com a dificuldade escolhida. Um detalhe é que na dificuldade hard o mapa possui um efeito de vento.

Options: O botão options leva a uma nova página para seleção de skins, sendo que existem as skins de tanque, aeronave e canhão, cada uma com uma

mecânica de tiro diferente. A página possui um botão back para voltar ao menu principal e um ícone de volume para alterar o volume de sons e da música.



Com o jogo iniciado, os tanques podem se mover para frente, para trás e girar para esquerda e direita através dos respectivos botões W, S, A e D, para o jogador 1, e através das setas, para o jogador 2. Também é possível que o tanque atire na direção em que estiver mirando com os botões F e L, respectivos aos jogadores 1 e 2.

O objetivo do jogo é acertar 5 vezes o tanque inimigo com os tiros, sendo que existem obstáculos no mapa que impedem movimento e passagem de tiros. Quando um dos tanques coleta os 5 pontos, ele recebe uma vitória e a pontuação da partida é armazenada no histórico, no qual terá o número de vitórias dos jogadores e a pontuação da última partida. Com a vitória de um dos tanques, a tela irá se alterar para essa:

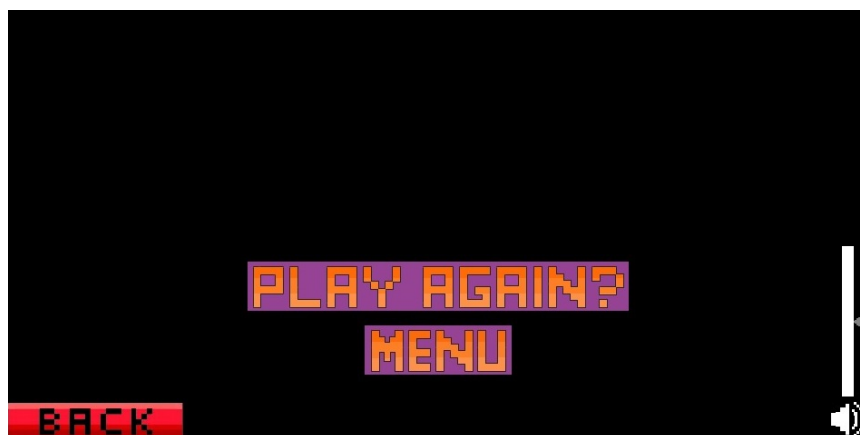


A tela nova contém o histórico de partidas mencionado anteriormente e dois botões e o ícone de volume que já foi mencionado anteriormente.

Play Again: O botão play again reinicia o jogo com as mesmas especificações do anterior.

Menu: O botão menu retorna à tela inicial do menu principal.

OBS.: Durante um jogo é possível apertar a tecla P para pausar a partida e ir para seguinte tela:



Os botões Play Again e Menu estão presentes novamente, assim como o ícone de volume, e todos fazem as mesmas funções já mencionadas. O botão back retorna ao jogo.

## Descrição do Código

### Variáveis Globais:

**const float FPS** - FPS do jogo

**const int SCREEN\_W** - Largura da tela do jogo

**const int SCREEN\_H** - Altura da tela do jogo

**const int SPACE\_ABOVE** - Espaço para a pontuação durante o jogo

**const int WIDTH\_REC** - Largura de obstáculos retangulares

**const float VELOCIDADE\_TANK** - Velocidade do tanque

**const float VELOCIDADE\_TIRO** - Velocidade do tiro

**const float VENTO** - Velocidade do vento

**const float GRAVIDADE** - Aceleração da gravidade

**const float TANK\_W** - Largura do bitmap do tanque

**const float TANK\_H** - Altura do bitmap do tanque

**const float RAIO\_TANK** - Raio do Escudo do tanque

**const int CD** - Tempo de recarga do tiro

**const int CD\_EXPLOSAO** - Tempo até o tiro explodir no modo canhão

**const int RAI0\_EXPLOSAO** - Raio de explosão do tiro no modo canhão

**const float BORDAS[4][4], const float MEIO[2][4], const float CANTOS[4][4], const float REC[4][4]** - Valores dos pontos iniciais e finais dos obstáculos retangulares do mapa.

**const float CIRCULO[1][3], const float BOLAS[4][3]** - Valores dos pontos centrais e raios dos obstáculo circulares do mapa.

## Structs:

**Bot** - Estrutura criada para armazenar as configurações de cada botão ou imagem fixa do jogo. Possui as variáveis x, y, width e height, que representam respectivamente as localizações x e y na tela, a largura e a altura do botão ou imagem.

**Skin** - Estrutura criada para armazenar as configurações de cada skin mostrada na tela. Possui as variáveis x, y e raio, que representam respectivamente as localizações x e y na tela, e o raio do escudo da skin.

**Tank** - Estrutura criada para armazenar as configurações de cada um dos dois tanques do jogo. Possui variáveis que representam a localização x e y, seu giro em graus e para onde está girando, o seno e cosseno de seu giro, tentativas de movimento para frente e para trás, permissão de mover-se para frente ou para trás, os pontos do tanque, e a cor do tanque.

**Shot** - Estrutura criada para armazenar as configurações de cada um dos tiros realizados no jogo. Possui variáveis que representam a localização x e y, seu tamanho, seu giro em graus, o seno e cosseno de seu giro, tentativas de tiro, existência de um tiro, próximo movimento, força que o desloca (no caso do tiro de uma aeronave), evento de explosão (no caso do tiro de um canhão), o tipo do tiro e sua cor.

## Funções:

- **int random()**  
Função que retorna um número inteiro entre 0 e 4, incluindo os extremos.
- **void draw\_scenario0(ALLEGRO\_DISPLAY \*display)**  
Função que desenha o cenário 0 na tela, correspondente a dificuldade fácil.
- **void draw\_scenario1(ALLEGRO\_DISPLAY \*display)**  
Função que desenha o cenário 1 na tela, correspondente a dificuldade

média.

- **void draw\_scenario2(ALLEGRO\_DISPLAY \*display)**  
Função que desenha o cenário 2 na tela, correspondente a dificuldade difícil.
- **void initBot(Bot \*bot, ALLEGRO\_BITMAP \*bitmap)**  
Função que inicia as variáveis Bot com suas configurações de altura e largura.
- **void initSkins(Skin \*skinv1, Skin \*skinv2, Skin \*skinv3)**  
Função que inicia as skins com suas configurações de localização na tela.
- **void initBots(Bot \*playb, Bot \*diffb, Bot \*playagb, Bot \*menub, Bot \*backb, Bot \*gameoverb, Bot \*titleb, Bot \*optionsb, Bot \*volumeb, Bot \*setab)**  
Função que inicia as variáveis Bot com suas configurações de localização na tela.
- **void initTanks(Tank \*tank1, Tank \*tank2)**  
Função que inicializa todas as variáveis presentes no tipo Tank com suas configurações iniciais, que correspondem a posição inicial dentro do jogo.
- **void initTiros(Shot \*shot1, Shot \*shot2)**  
Função que inicializa as variáveis principais presentes no tipo Shot com suas configurações iniciais.
- **void initTiros(Shot \*shot1, Shot \*shot2, Tank \*tank1, Tank \*tank2, float tempo, int \*som1, int \*som2)**  
Função que inicia as configurações iniciais do tiro quando um tanque atira.
- **void tipoTiro(Shot \*shot, int cenario)**  
Função que indica os próximos movimentos do tiro em suas respectivas variáveis de próximo movimento, de acordo com o tipo de tanque utilizado e o mapa.
- **void draw\_tanks(ALLEGRO\_BITMAP \*t1, ALLEGRO\_BITMAP \*t2, Tank \*tank1, Tank \*tank2, Shot \*shot1, Shot \*shot2, int cenario)**  
Função que desenha os bitmaps dos tanques de acordo com a skin escolhida pelo usuário.
- **void draw\_skins(ALLEGRO\_BITMAP \*s1, ALLEGRO\_BITMAP \*s2, ALLEGRO\_BITMAP \*s3)**  
Função que desenha os bitmaps das skins do tanque.

- **void draw\_shots(ALLEGRO\_BITMAP \*s1, ALLEGRO\_BITMAP \*s2, Shot \*shot1, Shot \*shot2)**  
Função que desenha os bitmaps dos tiros de acordo com o escolhido pelo usuário.
- **void draw\_shots1(Shot \*shot1, Shot \*shot2)**  
Função que indica as configurações de raio, altura e largura do tiro de acordo com o tiro escolhido pelo usuário.
- **void verificaTeclaDown(Tank \*tank1, Tank \*tank2, Shot \*shot1, Shot \*shot2, ALLEGRO\_EVENT ev, int \*state)**  
Função que verifica os botões pressionados pelo usuário e indica o que deve ser feito de acordo com o botão pressionado.
- **void verificaTeclaUp(Tank \*tank1, Tank \*tank2, Shot \*shot1, Shot \*shot2, ALLEGRO\_EVENT ev)**  
Função que verifica o botão soltado pelo usuário e indica o que deve ser feito de acordo com o botão soltado.
- **void calculaSenCos(Tank \*tank1, Tank \*tank2)**  
Função que calcula os senos e cossenos dos tanques de acordo com seus giros.
- **void verificaExplosao(Shot \*shot, int raio, Tank \*tank1, Tank \*tank2)**  
Função que é chamada para um tanque com skin de canhão. Ela verifica se houve colisão de tanques com a explosão do tiro e acrescenta pontos ao jogador(es) correspondente(s) caso ocorra.
- **int verificaUp(Tank tank, float x1, float y1, float x2, float y2)**  
Função que verifica a colisão do tanque movendo para frente com um obstáculo retangular.
- **int verificaDown(Tank tank, float x1, float y1, float x2, float y2)**  
Função que verifica a colisão do tanque movendo para trás com um obstáculo retangular.
- **int verificaTankUp(Tank tank, float x1, float y1, float raio)**  
Função que verifica colisão do tanque movendo para frente com um círculo ou outro tanque.
- **int verificaTankDown(Tank tank, float x1, float y1, float raio)**  
Função que verifica colisão do tanque movendo para trás com um círculo ou outro tanque.

- **int verificaTiro(Shot shot, float x1, float y1, float x2, float y2)**  
Função que verifica colisão de um tiro com um obstáculo retangular.
- **int verificaTiroTank(Shot shot, float x1, float y1, float raio)**  
Função que verifica colisão de um tiro com um obstáculo circular ou um tanque.
- **int verificaVento(Tank tank, float x1, float y1, float x2, float y2)**  
Função somente chamada com o cenário 2. Ela verifica colisões provocadas pelo vento com obstáculos retangulares.
- **int verificaVentoCirculos(Tank tank, float x, float y, float raio)**  
Função somente chamada com o cenário 2. Ela verifica colisões provocadas pelo vento com obstáculos circulares ou outro tanque.
- **void obstaculo(Tank \*tank1, Tank \*tank2, Shot \*shot1, Shot \*shot2, int cenario)**  
Função que chama as funções de verificação de colisão para todos os obstáculos de um cenário e altera as configurações de movimento do tanque baseado na existência ou não de uma colisão. Isso é feito através de mudanças nas variáveis de permissão ou não de movimento do tanque. A função faz todas as verificações com tiros também, alterando as configurações da mesma forma e alterando a pontuação caso um tiro acerte um tanque.
- **void obstaculoVento(Tank \*tank1, Tank \*tank2, int cenario)**  
Função que chama as funções de verificação de colisão por vento para todos os obstáculos do cenário 2 e altera o movimento do tanque de acordo com o vento se não existir problemas de colisão.
- **void moveUpDown(Tank \*tank1, Tank \*tank2)**  
Função que move o tanque para frente e para trás quando puder e for solicitado através das variáveis do tanque.
- **void moveLeftRight(Tank \*tank1, Tank \*tank2)**  
Função que gira o tanque para direita e esquerda alterando a variável de ângulo do giro.
- **void arquivaHistorico(FILE \*historico, char h[], int v1, int v2, int p1, int p2)**  
Função que arquiva o histórico de forma organizada como uma tabela.
- **void arquivaHistorico2(FILE \*historico, char h[], int v1, int v2, int p1, int p2)**  
Função que arquiva somente os números do histórico.

- **void imprimeHistorico(FILE \*historico, char h[], ALLEGRO\_FONT \*fonte3)**  
Função que mostra o histórico durante a tela de derrota a partir do arquivo de números.

## **Main:**

### **Primeira Parte:**

Durante a primeira parte são declaradas todas as variáveis de display, fila de eventos, timer, bitmaps, samples, fontes e audio streams. Também são declaradas as variáveis que representam o cenário escolhido, o estado dentro do jogo, os arquivos do histórico e a variável que armazena as vitórias de cada jogador.

Algumas variáveis auxiliares também são criadas, elas são utilizadas para realização dos sons de tiros, explosões e pontos, e para regulação do volume do jogo.

### **Segunda Parte:**

São feitas rotinas de inicialização para as bibliotecas incluídas, são criadas as variáveis Bot, Skin, Tank e Shot, sendo inicializadas com suas funções de inicialização respectivas. As variáveis tipo ALLEGRO também são inicializadas com suas configurações, algumas através de funções, além disso imagens, sons e fontes são carregadas às respectivas variáveis.

São feitos testes para verificar erros de criação dessas variáveis. Os bitmaps dos tiros e tanques ainda não são desenhados porque ainda não há escolha do usuário.

A fila de eventos é criada e o timer é iniciado.

### **Quarta Parte:**

Enquanto o looping do jogo estiver acontecendo existirão 5 estados possíveis.

#### **Estado 0: Menu do jogo**

Durante o menu é tocada a trilha sonora do menu, os bitmaps dos tanques e tiros são zerados e os botões de play, dificuldade, e options são desenhados na tela, assim como o título do jogo. Também é desenhado um círculo ao lado das dificuldades mostrando a atual seleção.

Caso o botão play seja clicado, o jogo é iniciado indo ao estado 1 com as configurações feitas. Também é escolhida aleatoriamente uma trilha sonora



entre 5 para ser tocada durante o jogo.

Caso alguma dificuldade seja selecionada o círculo que mostra a dificuldade atual se alterará para a escolhida.

Caso o botão options seja clicado, a tela se alterará para um menu de opções indo ao estado 4.

#### **Estado 4: Menu de opções**

Durante o menu de opções são desenhadas as skins na tela girando e um quadrado verde que indica qual skin é a atual, além de um botão back para retornar ao menu do jogo e um ícone de volume para alteração do mesmo.

Caso uma skin seja selecionada o quadrado verde se altera para a desejada. Caso clique na barra de volume, a seta indicando o volume se altera para o local clicado e o volume é alterado proporcionalmente à barra.

#### **Estado 1: Jogo**

A trilha sonora é alterada, o cenário é desenhado na tela de acordo com a dificuldade escolhida pelo jogador e os tanques são desenhados.

É verificado o movimento desejado através da verificação de teclas, caso o movimento seja possível e não ocorra colisão, o tanque correspondente irá mover de acordo com o desejado. Também é verificado se o jogador quer atirar, caso seja possível, ou seja, caso não haja outro tiro do jogador acontecendo e o tempo de recarga já tenha passado, ele atira e o tiro é desenhado e movimentado de acordo com seu tipo. Para tipos 0 e 1, se o tiro acertar um obstáculo ou um tanque ele some, sendo que a pontuação aumentará quando acertar o tanque inimigo. Para o tipo 2, se o tiro acertar um obstáculo ou tanque ele explode, tanques afetados pela explosão darão um ponto para o oponente, caso o tiro não acerte nada ele explode depois de 1 segundo.

Um detalhe é que para o cenário 2 é verificado se os tanques podem se mover para esquerda, se puderem o efeito de vento é aplicado e eles se movem nessa direção.

Os sons de tiros, explosões e da pontuação são ativados nos momentos correspondentes.

Caso a tecla P seja clicada o jogo irá para o estado 2. Caso algum tanque chegue a 5 pontos o estado muda para 3 e o histórico é atualizado.

#### **Estado 2: Pause**

A trilha sonora é alterada para a de pausa, são desenhados 3 botões na tela, um de play again, um de menu e um back, e a barra de volume. Caso o play again seja selecionado o jogo vai para o estado 1 inicializando os tanques e tiros para as configurações iniciais e utilizando as especificações de mapa e skin da partida anterior. Selecionando o menu, o jogo vai para o estado 0 inicializando os tanques e tiros para as configurações iniciais. Se o back for clicado, o jogo retorna ao estado 1 sem outras alterações.

Caso clique na barra de volume, a seta indicando o volume se altera para o local clicado e o volume é alterado proporcionalmente à barra.

### **Estado 3: Derrota**

A trilha sonora é alterada para a final, são desenhados 2 botões na tela (play again e menu), uma imagem de Game Over e a barra de volume. Além disso, o histórico de vitórias e a pontuação da última partida é mostrada na tela.

Caso o play again seja selecionado o jogo vai para o estado 1 inicializando os tanques e tiros para as configurações iniciais, e utilizando as especificações de mapa e skin da partida anterior. Selecionando o menu, o jogo vai para o estado 0 inicializando os tanques e tiros para as configurações iniciais.

Caso clique na barra de volume, a seta indicando o volume se altera para o local clicado e o volume é alterado proporcionalmente à barra.

Finalmente, quando o botão de fechar o display é pressionado o looping do jogo é terminado, todas as variáveis ALLEGRO são destruídas e o programa termina.