**1 INSTRUÇÕES DO JOGO**

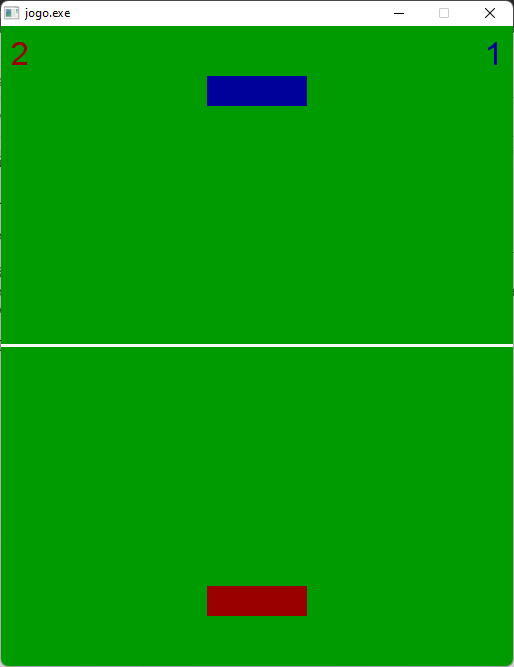
**1.1 DESCRIÇÃO**

O jogo, denominado de Multi-Tênis, consiste em dois jogadores jogando um contra o outro em uma partida parecida com o esporte Tênis.

No jogo são lançadas múltiplas bolas, cada uma contendo velocidades e tamanhos diferentes. As bolas devem ser lançadas dos quatro cantos da quadra e devem ricochetear sempre que bater nas laterais da tela.

Os jogadores podem se movimentar dentro das suas perspectivas áreas e devem rebater as bolas usando um botão específico para cada um. O jogador que atingir 10 pontos primeiro ganha o jogo e em seguida aparece o placar com o ganhador e o número de vitórias que cada um possui.

**1.2 TELAS**

*Tela de Jogo Tela Final*

**1.2.1 Tela de Jogo**

Os jogadores são representados pelo retângulo azul e pelo retângulo vermelho.

No canto superior esquerdo está contido a pontuação do jogador vermelho e no canto direito a pontuação do jogador azul.

No centro da tela passa uma linha branca horizontal para delimitar a área de movimentação de cada jogador

**1.2.2 Tela Final**

O jogador que venceu é colocado na tela, abaixo é colocado o placar da partida e em seguida é colocado o número de vitórias que cada jogador possui.

**1.3 CONTROLES**

**SETA CIMA, SETA ESQUERDA, SETA DIREITA, SETA BAIXO:** move o Jogador Azul.

**0 (zero):** o Jogador Azul rebate.

**W, A, D, S:** move o Jogador Vermelho.

**ESPAÇO:** o Jogador Azul Vermelho.

**BOTÃO DIREITO DO MOUSE:** fecha o jogo quando clica em fechar o display.

**2 DESCRIÇÃO DO CÓDIGO**

**2.1 PROCEDIMENTOS ESPECÍFICOS DO CÓDIGO ANTES DE RODÁ-LO**

Linha 1 à 8: Includes das bibliotecas utilizadas no jogo.

Linha 10 à 29: Declaração das constantes e variáveis globais usadas no jogo.

Linha 33 à 35: Função para gerar números aleatórios entre um número mínimo e um número máximo.

float randfloat(float max, float min);

Linha 38 à 42: Struct das bolas como Bola com os campos de posição, velocidade vertical, velocidade horizontal, cor, raio e se está em jogo ou não.

Linha 46 à 51: Função para desenhar na tela todas as bolas que estão dentro do jogo.

void desenhaBolas(Bola bolas[]);

Linha 54 à 58: Struct dos jogadores como Jogador com os campos de posição, velocidade, cor, tamanho, pontuação e se está rebatendo ou não.

Linha 61 à 74: Função para a criação do Jogador 1 (vermelho) com suas propriedades.

void initJogador1(Jogador \*p1);

Linha 77 à 90: Função para a criação do Jogador 2 (azul) com suas propriedades.

void initJogador2(Jogador \*p2)

Linha 92 à 114: Função para atualizar os jogadores.

void atualizaJogadores(Jogador \*p1, Jogador \*p2)

Linha 117 à 120: Função para desenhar na tela os jogadores de acordo com seus dados.

void desenhaJogadores(Jogador p1, Jogador p2);

Linha 123 à 126: Função para desenhar na tela a quadra.

void desenhaQuadra();

Linha 129 à 177: Função para atualizar as bolas em jogo.

void atualizaBolas(Bola bolas[], Jogador \*p1, Jogador \*p2, int bolasjogo);

Linha 180 à 223: Função para lançar as bolas em cada canto da tela.

void canhao(Bola bolas[], int bolasjogo, int posicao, int atirou);

**2.2 FUNÇÃO MAIN**

Linha 225 à 517: Função main.

int main(int argc, char \*\*argv);

**2.2.1 ROTINAS DE INICIALIZAÇÃO DO ALLEGRO**

Linha 234 à 303: Rotinas de inicialização do Allegro.

Linha 234 à 237: Inicializa o Allegro.

Linha 240 à 243: Inicializa o módulo de primitivas do Allegro.

Linha 246 à 249: Inicializa o módulo que permite carregar imagens no jogo.

Linha 252 à 256: Cria um temporizador que incrementa uma unidade a cada 1.0/FPS segundos.

Linha 259 à 264: Cria uma tela com dimensões de SCREEN\_W, SCREEN\_H pixels.

Linha 267 à 270: Instala o teclado.

Linha 273: Inicializa o módulo allegro que carrega as fontes.

Linha 276 à 279: Inicializa o módulo allegro que entende arquivos tff de fontes.

Linha 282 à 285: Carrega o arquivo arial.ttf da fonte Arial e define que será usado o tamanho 32.

Linha 288 à 293: Cria a Fila de Eventos

Linha 296: Registra na fila os eventos de tela.

Linha 298: Registra na fila os eventos de tempo: quando o tempo altera de t para t+1.

Linha 300: Registra na fila os eventos de teclado.

Linha 303: Inicializa o temporizador.

**2.2.2 PROCEDIMENTOS ESPECÍFICOS INICIAIS**

Linha 306 à 309: Criação e inicialização dos jogadores.

Linha 312: Criação de vetor para armazenar as bolas.

Linha 316 à 324: Criação de variáveis para o código.

Linha 327 à 339: Criação de variáveis, abertura e fechamento de arquivo e outros procedimentos relacionados ao registro.

**2.2.3 WHILE {**

Linha 341 à 501: While principal que faz o código do jogo rodar até ele ser parado.

Linha 342 à 344: Atualização da fila de eventos.

**2.2.3.1 SE O TIPO DE EVENTO FOR O TEMPO PASSAR**

Linha 347 à 423: Se o tipo de evento for um evento do temporizador, ou seja, se o tempo passou de t para t+1.

Linha 422: Atualiza a tela.

**2.2.3.1.A SE NENHUM JOGADOR GANHOU**

Linha 348 à 359: Chama as funções desenhaQuadra(), atualizaJogadores(&p1, &p2), atualizaBolas(bolas, &p1, &p2, bolasjogo), desenhaBolas(bolas). As funções servem para desenhar e atualizar a quadra, jogadores e bolas.

Em seguida é feito o processo de desenhar a pontuação de cada jogador nos cantos superiores da tela.

Linha 362 à 373: Se passar 5 segundos o canhão é ativado e sua posição muda.

**2.2.3.1.B SE UM JOGADOR GANHOU**

Linha 381 à 386: Faz com que o jogo juntamente com a tela final fiquem abertos por um intervalo de tempo antes de fechar.

Linha 388: Pinta a tela de branco.

Linha 391 à 419: Faz com que sejam escritos os textos finais da tela dependendo qual jogador ganhou.

**2.2.3.2 SE O TIPO DE EVENTO FOR O FECHAMENTO DE TELA**

Linha 425 à 428: Se o apertar no x da aba o jogo termina.

**2.2.3.2 SE O TIPO DE EVENTO FOR O PRESSIONAR DE UMA TECLA**

Linha 431 à 501: Dependendo da tecla que for abaixada ou levantada, o jogador começa ou deixa de fazer algo.

**2.2.4 SALVAR HISTÓRICO**

Linha 503 à 509: Abri o arquivo onde está salvo o histórico para sua atualização e logo em seguida ele é fechado

**2.2.5 PROCEDIMENTOS DE FIM DE JOGO**

Linha 512 à 514: Fecha o timer, o display e a fila de eventos

**2.2.6 RETURN 0**

Linha 516: “return 0;” e fim do código.