GUIA DE USO

Trabalho Prático - TENISMULTIPLAYER (TENISMP)
Universidade Federal de Minas Gerais
PDS1 01/2022
Thiago Henrique Santos da Silva

Introdução -

Este se trata do trabalho prático do primeiro semestre da disciplina de Programação e Desenvolvimento de Software 1
O jogo foi baseado especialmente em pong, com a diferença de que, desta vez, o jogo é jogado na horizontal ao invés da vertical, fora que os jogadores podem se mover em até 8 direções, enquanto simula uma partida de tênis, onde temos a rede no meio que divide as duas quadras e a habilidade de rebater a bola de volta para seu adversário, ganha aquele que fizer 10 pontos primeiro.
O trabalho foi feito utilizando da biblioteca Allegro na linguagem C, abrindo as portas para a criação de jogos com diversos elementos.

Gameplay -

No jogo temos dois jogadores, controlados pelo usuário, que tem como objetivo vencer seu adversário, o que fica cada vez mais difícil conforme mais e mais bolas vem a tela.
Os jogadores são movimentados via as teclas direcionais (setas) para o jogador 1, e as teclas WASD para o jogador 2, incluindo também o botão de ataque, sendo espaço para o jogador 1 e a tecla ALT para o jogador 2, a fim de tornar os controles mais próximos de suas teclas de direcionais.

A cada 5 segundos uma nova bola é colocada na tela, chegando até o limite de 12 bolas na tela, sendo criadas enquanto houverem menos de 12 bolas no ecrã do jogo, e para deixar o jogo mais técnico, aqueles que preferirem arriscar com sua precisão serão recompensados, caso a bola seja rebatida quando estiver bem em cima do centro do jogador, um bônus de velocidade é aplicado, dificultando a vida do seu adversário, mas não se preocupe, é possível freá-la se acertar nos primeiros momentos que ela entra em contato com seu jogador.

Código -

O código foi feito buscando que o máximo de tarefas fossem terceirizadas através de funções, isso pode ser notado através de procedimentos como *CriaBola* responsável pela criação das bolas na tela, *Respawn_Bola* responsável pela inserção das bolas na tela após elas serem destruídas, *ColisãoJogadorBola* responsável pela interação entre o jogador e a bola, permitindo que ela seja ricocheteada para o campo adversário, a coleção de modificadores como *add_tela, add_criadas, minus_telas, atualiza_kontador* responsáveis por alterar valores através de ponteiros no código, fora eles existem mais dois procedimentos muito importantes, sendo eles *On_off_screen* responsável por dizer se a bola está dentro ou fora da tela, útil para

que a função *Respawn_Bola* funcione corretamente, e por fim o *ColisãoOob* responsável pela contagem de pontos (mais explicações abaixo).

Foram utilizados 3 tipos diferentes de Struct no código, um para o jogador, um para a bola e um último para as hitboxes de Oob (Out of Bounds/Fora dos Limites), sendo utilizadas para a contagem dos pontos, uma solução alternativa para que não houvesse conflitos com o *On_off_screen,* estes structs são iniciados em suas respectivas funções *CriaBola, InitJogador* e *InitOob*

Fora as funções e procedimentos, temos o código principal, que cuida de boa parte da inicialização dos procedimentos do alegro, assim como das verificações, especialmente as por tempo, onde são realizadas grande maioria dos checks.

```
if(ev.type == ALLEGRO_EVENT_TIMER) {

    x = pontos_p1/6;
    y = pontos_p2/6;

    sprintf(placar1, "%d", x);
    sprintf(placar2, "%d", y);

    draw_scenario();

    for (int i = 0; i < MAX_BOLAS; i++){
        for (int j = 0; j< 2; j++){
            ColisaoOob(&bolas[i], Oobs, j, &pontos_p1, &pontos_p2);
        }
    }

    al_draw_text(font, al_map_rgb(255,255,255), 0, 0, ALLEGRO_ALIGN_LEFT, &placar1[0]);
    al_draw_text(font, al_map_rgb(255,255,255), SCREEN_W, 0, ALLEGRO_ALIGN_RIGHT, &placar2[0]);

    for (int i = 0; i<2; i++){
        update_jogador(jogadores, i);
    }

    for (int i = 0; i<2; i++){
        draw_jogador(&jogadores[i]);
    }

    for (int i = 0; i <MAX_BOLAS; i++){
        colisaobolaparede(&bolas[i], bolas_tela);
        on_off_screen(&bolas[i]);
    }


    if(al_get_timer_count(timer)%(int)FPS == 0){
        if (segundos == 4){
            segundos = 0;
            if (bolas_criadas < MAX_BOLAS){
                criaBola(&bolas[bolas_criadas], bolas_criadas);
                PosicaoBola(&bolas[bolas_criadas], next_bola);
                add_criada(&bolas_criadas);
                add_next(&next_bola);
            }else{
                respawn_bola(bolas,next_bola,kontador_prox_respawn);
                atualiza_kontador(&kontador_prox_respawn);
                add_next(&next_bola);
            }
        }else{
            segundos++;
        }
    }

    for (int i = 0; i <MAX_BOLAS; i++){
        update_bola(&bolas[i]);
        desenhaBola(&bolas[i]);
    }
```

Por fim, quando um dos dois jogadores pontua 10 pontos, o jogo se encerra e o placar é mostrado na tela, caso contrário (em fechamento prematuro por clique no X) o jogo apenas é encerrado.

```
if (x >= 10 || y >= 10){
    char final[20];

    al_clear_to_color(al_map_rgb(0,0,0));
    if (x >= 10){

        sprintf(final, "Jogador vencedor: 1    Total de pontos: %d", x);
        al_draw_text(font, al_map_rgb(15, 200, 30), SCREEN_W/4, SCREEN_H/2, 0, final);
    }

    if (y >= 10){
        sprintf(final, "Jogador vencedor: 2    Total de pontos: %d", y);
        al_draw_text(font, al_map_rgb(15, 200, 30), SCREEN_W/4, SCREEN_H/2+100, 0, final);
    }

    //reinicializa a tela
    al_flip_display();
    al_rest(3);
    break;
}
```