#include <allegro5/allegro5.h>

#include <allegro5/allegro\_font.h>

#include <allegro5/allegro\_image.h>

#include <stdbool.h>

#include <allegro5/allegro\_primitives.h>

#include <allegro5/allegro\_ttf.h>

#include <stdio.h>

#include <stdlib.h>

#include <allegro5/keyboard.h>

#include <time.h>

#define FRAME\_WIDTH 44

#define FRAME\_HEIGHT 48

#define NUM\_FRAMES 4

#define FRAME\_DURATION 0.2

#define JUMP\_HEIGHT 80

#define GRAVITY 4

//varaiveis globais

const float FPS = 60.0;

const int screen\_w = 640;

const int screen\_h = 480;

const int tamanho\_jogador = 32;

const int tamanho\_inimigo = 32;

const int linhas = 10;

const int colunas = 10;

int board[linhas][colunas];

bool sair = false;

bool sairBomber = true;

int inicialX = 1;

int inicialY = 1;

typedef struct {

float x, y;

float vel\_jogador;

} Player;

typedef struct {

float x, y;

bool active;

} Tiro;

typedef struct {

double x;

double y;

bool active;

} Inimigo;

int fase\_atual = 0;

bool menu\_aberto = false;

void comecaBoard() {

srand(time(NULL));

for (int i = 0; i < linhas; i++) {

for (int j = 0; j < colunas; j++) {

int n1 = rand() % 10;

if (n1 % 2 == 0) {

board[i][j] = 1; // Bloco sólido

}

else {

board[i][j] = 0; // Espaço vazio

}

}

}

for (int i = 0; i <= linhas; i += 2) {

for (int j = 0; j <= colunas; j += 2) {

if (board[i][j] == 0) {

board[i][j] = 2;

}

}

}

board[inicialY][inicialX] = 3; //Posição do jogador

board[linhas - 1][colunas - 1] = 4; //Objetivo

}

void drawBoard(ALLEGRO\_DISPLAY\* display) {

for (int i = 0; i < linhas; i++) {

for (int j = 0; j < colunas; j++) {

switch (board[i][j]) {

case 0: // Espaço em branco

al\_draw\_filled\_rectangle(j \* 40, i \* 40, (j + 1) \* 40, (i + 1) \* 40, al\_map\_rgb(255, 255, 255));

break;

case 1: // Bl solido

al\_draw\_filled\_rectangle(j \* 40, i \* 40, (j + 1) \* 40, (i + 1) \* 40, al\_map\_rgb(0, 0, 0));

break;

case 2: // Bl destruidor

al\_draw\_filled\_rectangle(j \* 40, i \* 40, (j + 1) \* 40, (i + 1) \* 40, al\_map\_rgb(200, 0, 0));

break;

case 3: // Jogador

al\_draw\_filled\_rectangle(j \* 40, i \* 40, (j + 1) \* 40, (i + 1) \* 40, al\_map\_rgb(0, 0, 255));

break;

case 4: // Objetivo

al\_draw\_filled\_rectangle(j \* 40, i \* 40, (j + 1) \* 40, (i + 1) \* 40, al\_map\_rgb(0, 255, 255));

break;

}

}

}

}

// Função para mover o jogador

int movJogador(int dx, int dy) {

int novoX = inicialX + dx;

int novoY = inicialY + dy;

if (novoX >= 0 && novoX < colunas && novoY >= 0 && novoY < linhas &&

(board[novoY][novoX] == 0 || board[novoY][novoX] == 2 || board[novoY][novoX] == 4)) {

board[inicialY][inicialX] = 0;

inicialX = novoX;

inicialY = novoY;

board[inicialY][inicialX] = 3;

}

if (board[novoX][novoY] == board[linhas - 1][colunas - 1]) {

printf("Voce conseguiu");

return 2;

}

}

int Bomberman(int x) {

srand(time(NULL));

ALLEGRO\_DISPLAY\* disp = al\_create\_display(800,600 );

ALLEGRO\_EVENT\_QUEUE\* fila\_de\_eventos = al\_create\_event\_queue();

al\_register\_event\_source(fila\_de\_eventos, al\_get\_display\_event\_source(disp));

al\_register\_event\_source(fila\_de\_eventos, al\_get\_keyboard\_event\_source());

int cont;

al\_clear\_to\_color(al\_map\_rgb(0, 0, 0));

drawBoard(disp);

al\_flip\_display();

//transformar tudo em Disp (displaY)

while (!sairBomber) {

//Agora a fase 3 que não está rodando no atual momento, no código principal!!!!

ALLEGRO\_EVENT event;

al\_wait\_for\_event(fila\_de\_eventos, &event);

if (event.type == ALLEGRO\_EVENT\_KEY\_DOWN) {

switch (event.keyboard.keycode) {

case ALLEGRO\_KEY\_RIGHT:

movJogador(1, 0);

break;

case ALLEGRO\_KEY\_LEFT:

movJogador(-1, 0);

break;

case ALLEGRO\_KEY\_UP:

movJogador(0, -1);

break;

case ALLEGRO\_KEY\_DOWN:

movJogador(0, 1);

break;

}

}

if (event.type == ALLEGRO\_EVENT\_KEY\_DOWN) {

switch (event.keyboard.keycode) {

case ALLEGRO\_KEY\_SPACE:

comecaBoard();

}

}

int numero = movJogador(0, 0);

if (numero == 2) {

return 2;

al\_destroy\_display(disp);

}

}

}

void desenhar\_menu(ALLEGRO\_FONT\* fonte) {

al\_clear\_to\_color(al\_map\_rgb(0, 0, 0));

al\_draw\_text(fonte, al\_map\_rgb(255, 255, 255), 400, 200, ALLEGRO\_ALIGN\_CENTER, "1. Fase 1");

al\_draw\_text(fonte, al\_map\_rgb(255, 255, 255), 400, 250, ALLEGRO\_ALIGN\_CENTER, "2. Fase 2");

al\_draw\_text(fonte, al\_map\_rgb(255, 255, 255), 400, 300, ALLEGRO\_ALIGN\_CENTER, "3. Fase 3");

al\_draw\_text(fonte, al\_map\_rgb(255, 255, 255), 400, 350, ALLEGRO\_ALIGN\_CENTER, "4. Voltar para casa");

al\_flip\_display();

}

void spaceInvaders() {

ALLEGRO\_DISPLAY\* display = al\_create\_display(screen\_w, screen\_h);

ALLEGRO\_TIMER\* timer = al\_create\_timer(1.0 / FPS);

ALLEGRO\_EVENT\_QUEUE\* event\_queue = al\_create\_event\_queue();

// Objetos

Player jogador = { screen\_w / 2 - tamanho\_jogador / 2, screen\_h - tamanho\_jogador, 5.0 };

Tiro bala = { jogador.x, jogador.y, false };

Inimigo inimigos[5];

bool andar\_esquerda = false;

bool andar\_direita = false;

// Inicializando os inimigos

for (int i = 0; i < 5; i++) {

inimigos[i].x = i \* 100 + 50.0;

inimigos[i].y = 50.0;

inimigos[i].active = true;

}

// Registrar eventos

al\_register\_event\_source(event\_queue, al\_get\_display\_event\_source(display));

al\_register\_event\_source(event\_queue, al\_get\_timer\_event\_source(timer));

al\_register\_event\_source(event\_queue, al\_get\_keyboard\_event\_source());

al\_start\_timer(timer);

bool redraw = true;

bool Sair\_jogo = false;

// Laço principal

while (!Sair\_jogo) {

ALLEGRO\_EVENT ev;

al\_wait\_for\_event(event\_queue, &ev);

//Condição para checar a tecla apertada

if (ev.type == ALLEGRO\_EVENT\_KEY\_DOWN) {

switch (ev.keyboard.keycode) {

case ALLEGRO\_KEY\_LEFT:

andar\_esquerda = true;

break;

case ALLEGRO\_KEY\_RIGHT:

andar\_direita = true;

break;

case ALLEGRO\_KEY\_A:

andar\_esquerda = true;

break;

case ALLEGRO\_KEY\_D:

andar\_direita = true;

break;

case ALLEGRO\_KEY\_SPACE:

if (!bala.active) {

bala.x = jogador.x + tamanho\_jogador / 2;

bala.y = jogador.y;

bala.active = true;

}

break;

}

}

//Checa se a tecla foi solta

else if (ev.type == ALLEGRO\_EVENT\_KEY\_UP) {

switch (ev.keyboard.keycode) {

case ALLEGRO\_KEY\_LEFT:

andar\_esquerda = false;

break;

case ALLEGRO\_KEY\_RIGHT:

andar\_direita = false;

break;

case ALLEGRO\_KEY\_A:

andar\_esquerda = false;

break;

case ALLEGRO\_KEY\_D:

andar\_direita = false;

break;

}

}

if (ev.type == ALLEGRO\_EVENT\_TIMER) {

// Atualizar posição do jogador com base em teclas pressionadas

if (andar\_esquerda == true) {

jogador.x -= jogador.vel\_jogador;

}

if (andar\_direita == true) {

jogador.x += jogador.vel\_jogador;

}

// Atualizar posição do tiro

if (bala.active) {

bala.y -= 5.0;

if (bala.y < 0) {

bala.active = false;

}

}

// Verificar se o tiro pegou em algum inimigo

for (int i = 0; i < 5; i++) {

if (inimigos[i].active && bala.active &&

bala.x < inimigos[i].x + tamanho\_inimigo &&

bala.x + 5 > inimigos[i].x &&

bala.y < inimigos[i].y + tamanho\_inimigo &&

bala.y + 10 > inimigos[i].y) {

inimigos[i].active = false;

bala.active = false;

}

}

redraw = true;

}

else if (ev.type == ALLEGRO\_EVENT\_DISPLAY\_CLOSE) {

Sair\_jogo = true;

}

else if (ev.type == ALLEGRO\_EVENT\_KEY\_DOWN) {

// Atirando

if (ev.keyboard.keycode == ALLEGRO\_KEY\_SPACE && !bala.active) {

bala.x = jogador.x + tamanho\_jogador / 2;

bala.y = jogador.y;

bala.active = true;

}

}

if (redraw && al\_is\_event\_queue\_empty(event\_queue)) {

redraw = false;

// Desenhar tudo

al\_clear\_to\_color(al\_map\_rgb(0, 0, 0));

// Desenhar player

al\_draw\_filled\_rectangle(jogador.x, jogador.y, jogador.x + tamanho\_jogador, jogador.y + tamanho\_jogador, al\_map\_rgb(0, 255, 0));

// Desenhar tiro

if (bala.active) {

al\_draw\_filled\_rectangle(bala.x, bala.y, bala.x + 5, bala.y + 10, al\_map\_rgb(255, 255, 255));

}

// Desenhar inimigos

for (int i = 0; i < 5; i++) {

if (inimigos[i].active) {

al\_draw\_filled\_rectangle(inimigos[i].x, inimigos[i].y, inimigos[i].x + tamanho\_inimigo, inimigos[i].y + tamanho\_inimigo, al\_map\_rgb(255, 0, 0));

}

}

al\_flip\_display();

}

}

// Finalizando o Allegro

al\_destroy\_display(display);

al\_destroy\_timer(timer);

al\_destroy\_event\_queue(event\_queue);

}

int main() {

al\_init();

al\_install\_keyboard();

al\_init\_image\_addon();

al\_init\_font\_addon();

al\_init\_primitives\_addon();

ALLEGRO\_FONT\* fonte = al\_create\_builtin\_font();

int cot = 0;

ALLEGRO\_DISPLAY\* disp = al\_create\_display(800, 600);

ALLEGRO\_TIMER\* timer = al\_create\_timer(1.0 / 30.0);

ALLEGRO\_EVENT\_QUEUE\* fila\_eventos = al\_create\_event\_queue();

bool sair = false;

bool on\_homepage = true;

float x = 30;

float y = 435;

float frame = 0;

bool pulando = false;

float jump\_speed = 0;

float jump\_start\_y = y;

bool andando\_esquerda = false;

bool andando\_direita = false;

bool parado = true;

ALLEGRO\_BITMAP\* casa = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/casa.png");

ALLEGRO\_BITMAP\* Personagem = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/Walking.png");

ALLEGRO\_BITMAP\* Pulando = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/Pulando.png");

ALLEGRO\_BITMAP\* Parado = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/PGParado.png");

ALLEGRO\_BITMAP\* AndandoESQ = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/WalkingLeft.png");

ALLEGRO\_BITMAP\* Homepage = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/bg2.jpg");

ALLEGRO\_BITMAP\* cenario1 = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/cenario1.png");

ALLEGRO\_BITMAP\* cenario2 = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/cenario2.png");

ALLEGRO\_BITMAP\* cenario3 = al\_load\_bitmap("C:/Users/guilherme.basilva1/source/repos/Sprites/cenario3.png");

al\_register\_event\_source(fila\_eventos, al\_get\_display\_event\_source(disp));

al\_register\_event\_source(fila\_eventos, al\_get\_keyboard\_event\_source());

al\_register\_event\_source(fila\_eventos, al\_get\_timer\_event\_source(timer));

al\_start\_timer(timer);

while (!sair) {

ALLEGRO\_EVENT evento;

al\_wait\_for\_event(fila\_eventos, &evento);

if (evento.type == ALLEGRO\_EVENT\_TIMER) {

if (on\_homepage) {

al\_draw\_scaled\_bitmap(Homepage, 0, 0, al\_get\_bitmap\_width(Homepage), al\_get\_bitmap\_height(Homepage), 0, 0, 800, 600, 0);

al\_flip\_display();

}

else if (menu\_aberto) {

desenhar\_menu(fonte);

}

else {

ALLEGRO\_BITMAP\* cenario\_atual;

if (fase\_atual == 1) {

cenario\_atual = cenario1;

}

else if (fase\_atual == 2) {

cenario\_atual = cenario2;

}

else if (fase\_atual == 3) {

cenario\_atual = cenario3;

}

else {

cenario\_atual = casa;

}

al\_draw\_scaled\_bitmap(cenario\_atual, 0, 0, al\_get\_bitmap\_width(cenario\_atual), al\_get\_bitmap\_height(cenario\_atual), 0, 0, 800, 600, 0);

if (andando\_esquerda) {

x -= 3;

parado = false;

}

if (andando\_direita) {

x += 3;

parado = false;

}

if (!andando\_esquerda && !andando\_direita && !pulando) {

parado = true;

}

if (pulando) {

y -= jump\_speed;

jump\_speed -= GRAVITY \* 0.1;

if (y >= jump\_start\_y) {

y = jump\_start\_y;

pulando = false;

jump\_speed = 0;

}

}

frame += 0.3;

if ((int)frame >= NUM\_FRAMES) {

frame = 0;

}

if (pulando) {

al\_draw\_scaled\_bitmap(Pulando, ((int)frame) \* FRAME\_WIDTH, 0, FRAME\_WIDTH, FRAME\_HEIGHT, x, y, 100, 100, 0);

}

else if (parado) {

al\_draw\_scaled\_bitmap(Parado, 0, 0, FRAME\_WIDTH, FRAME\_HEIGHT, x, y, 100, 100, 0);

}

else if (andando\_esquerda) {

al\_draw\_scaled\_bitmap(AndandoESQ, ((int)frame) \* FRAME\_WIDTH, 0, FRAME\_WIDTH, FRAME\_HEIGHT, x, y, 100, 100, 0);

}

else if (andando\_direita) {

al\_draw\_scaled\_bitmap(Personagem, ((int)frame) \* FRAME\_WIDTH, 0, FRAME\_WIDTH, FRAME\_HEIGHT, x, y, 100, 100, 0);

}

al\_flip\_display();

}

}

else if (evento.type == ALLEGRO\_EVENT\_KEY\_DOWN) {

if (on\_homepage && evento.keyboard.keycode == ALLEGRO\_KEY\_ENTER) {

on\_homepage = false;

}

else if (!on\_homepage) {

switch (evento.keyboard.keycode) {

case ALLEGRO\_KEY\_LEFT:

andando\_esquerda = true;

parado = false;

break;

case ALLEGRO\_KEY\_RIGHT:

andando\_direita = true;

parado = false;

break;

case ALLEGRO\_KEY\_UP:

if (!pulando) {

pulando = true;

jump\_speed = 10;

jump\_start\_y = y;

}

break;

case ALLEGRO\_KEY\_ESCAPE:

menu\_aberto = !menu\_aberto;

break;

case ALLEGRO\_KEY\_1:

if (menu\_aberto) {

fase\_atual = 1;

menu\_aberto = false;

}

break;

case ALLEGRO\_KEY\_2:

if (menu\_aberto) {

spaceInvaders();

}

break;

case ALLEGRO\_KEY\_3:

if (menu\_aberto) {

sairBomber = false;

Bomberman(1);

}

break;

case ALLEGRO\_KEY\_4:

if (menu\_aberto) {

fase\_atual = 0;

}

break;

}

}

}

else if (evento.type == ALLEGRO\_EVENT\_KEY\_UP) {

switch (evento.keyboard.keycode) {

case ALLEGRO\_KEY\_LEFT:

andando\_esquerda = false;

break;

case ALLEGRO\_KEY\_RIGHT:

andando\_direita = false;

break;

}

}

else if (evento.type == ALLEGRO\_EVENT\_DISPLAY\_CLOSE) {

sair = true;

}

}

al\_destroy\_bitmap(Personagem);

al\_destroy\_bitmap(casa);

al\_destroy\_bitmap(Homepage);

al\_destroy\_bitmap(cenario1);

al\_destroy\_bitmap(cenario2);

al\_destroy\_bitmap(cenario3);

al\_destroy\_display(disp);

al\_destroy\_timer(timer);

al\_destroy\_event\_queue(fila\_eventos);

al\_destroy\_font(fonte);

return 0;

}