#define SDL\_MAIN\_HANDLED

//#include"D:\C++\Visual\_studio\MAIN\_GAME\MAIN\_GAME\lib.h"

#include <iostream>

#include <SDL.h>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <SDL\_ttf.h>

#include <string>

#include<SDL\_image.h>

using namespace std;

const int SCREEN\_WIDTH = 1000;

const int SCREEN\_HEIGHT = 600;

const int PLAYER\_SPEED = 10;

const int BULLET\_SPEED = 5;

const int ENEMY\_SPEED = 4;

const int ENEMY\_SPAWN\_TIMER = 1000;

struct GameObject {

int x, y, w, h;

};

struct Bullet : public GameObject {

bool active;

};

struct Enemy : public GameObject {

bool alive;

};

GameObject player = { SCREEN\_WIDTH / 2, SCREEN\_HEIGHT - 50, 80, 80 };

vector<Bullet> bullets;

vector<Enemy> enemies;

int score = 0;

Uint32 lastEnemySpawn = 0;

SDL\_Texture\* heartTexture = nullptr;

void handleInput(SDL\_Renderer\* renderer, GameObject& player, bool& isRunning);

void movePlayer(GameObject& player);

void fireBullet();

void spawnEnemy();

void moveBullets();

void moveEnemies();

void checkCollisions(SDL\_Renderer\* renderer);

void drawPlayer(SDL\_Renderer\* renderer, GameObject player, SDL\_Texture\* playerTexture);

void drawBullets(SDL\_Renderer\* renderer);

void drawEnemies(SDL\_Renderer\* renderer);

void drawScore(SDL\_Renderer\* renderer, TTF\_Font\* font, int score);

void initializeSDL(SDL\_Window\*& window, SDL\_Renderer\*& renderer, TTF\_Font\*& font, SDL\_Texture\*& backgroundTexture, SDL\_Texture\*& playerTexture);

void cleanup(SDL\_Window\* window, SDL\_Renderer\* renderer, TTF\_Font\* font);

void runGameLoop(SDL\_Renderer\* renderer, SDL\_Window\* window, TTF\_Font\* font, SDL\_Texture\* backgroundTexture, SDL\_Texture\* playerTexture,

GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score, bool& isRunning);

void drawMenu(SDL\_Renderer\* renderer, TTF\_Font\* font, SDL\_Window\* window, const string& backgroundImagePath, SDL\_Texture\* playerTexture,

GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score);

void countdownTimer(SDL\_Renderer\* renderer, TTF\_Font\* font);

void drawHearts(SDL\_Renderer\* renderer, int numHeartsRemaining);

void moveEnemiesAndCheckHearts(SDL\_Renderer\* renderer, int& numHeartsRemaining, bool& isRunning, int& score);

void resetGame(GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score, int& numHeartsRemaining);

void drawEndGameMenu(SDL\_Renderer\* renderer, TTF\_Font\* font, bool& isRunning, GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score);

int main(int argc, char\* argv[]) {

SDL\_Window\* window;

SDL\_Renderer\* renderer;

TTF\_Font\* font;

SDL\_Texture\* playerTexture;

SDL\_Texture\* backgroundTexture;

const string backgroundImagePath = "D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\bkgd\_menu.png"; // Đường dẫn menu background

initializeSDL(window, renderer, font, backgroundTexture, playerTexture);

bool inMenu = true;

SDL\_Event event;

while (inMenu) {

drawMenu(renderer, font, window, backgroundImagePath, playerTexture, player, lastEnemySpawn, bullets, enemies, score);

while (SDL\_PollEvent(&event)) {

if (event.type == SDL\_QUIT) {

inMenu = false; // Thoát khỏi menu nếu đóng cửa sổ

}

else if (event.type == SDL\_MOUSEBUTTONDOWN) {

int mouseX, mouseY;

SDL\_GetMouseState(&mouseX, &mouseY);

if (mouseX >= 400 && mouseX <= 600 && mouseY >= 200 && mouseY <= 250) {

// chọn "Start"

bool isRunning = true;

// Hiển thị đếm ngược trước khi bắt đầu trò chơi

countdownTimer(renderer, font);

runGameLoop(renderer, window, font, backgroundTexture, playerTexture,

player, lastEnemySpawn, bullets, enemies, score, isRunning);

inMenu = false; // Thoát khỏi menu sau khi kết thúc trò chơi

}

else if (mouseX >= 400 && mouseX <= 600 && mouseY >= 300 && mouseY <= 350) {

// chọn "Exit"

inMenu = false; // Thoát khỏi menu

}

}

}

}

cleanup(window, renderer, font);

return 0;

}

void drawMenu(SDL\_Renderer\* renderer, TTF\_Font\* font, SDL\_Window\* window, const string& backgroundImagePath, SDL\_Texture\* playerTexture,

GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score) {

SDL\_Surface\* backgroundSurface = IMG\_Load(backgroundImagePath.c\_str());

if (backgroundSurface == nullptr) {

// Xử lý khi không thể tải hình ảnh

cerr << "Failed to load background image: " << IMG\_GetError() << endl;

}

else {

SDL\_Texture\* backgroundTexture = SDL\_CreateTextureFromSurface(renderer, backgroundSurface);

SDL\_FreeSurface(backgroundSurface);

SDL\_SetRenderDrawColor(renderer, 0, 0, 0, 255);

SDL\_RenderClear(renderer);

// Vẽ background

SDL\_Rect backgroundRect = { 0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT };

SDL\_RenderCopy(renderer, backgroundTexture, NULL, &backgroundRect);

// Vẽ menu

SDL\_Color textColor = { 255, 255, 255 }; // Màu trắng

// init "Start"

TTF\_SetFontStyle(font, TTF\_STYLE\_BOLD); // Đặt kiểu chữ in đậm

TTF\_SetFontSize(font, 50); // kích thước của chữ "start" với "exit"

SDL\_Surface\* startSurface = TTF\_RenderText\_Blended(font, "Start", textColor);

SDL\_Texture\* startTexture = SDL\_CreateTextureFromSurface(renderer, startSurface);

SDL\_Rect startRect = { (SCREEN\_WIDTH - startSurface->w) / 2, 200, startSurface->w, startSurface->h }; // Căn giữa theo chiều ngang

SDL\_RenderCopy(renderer, startTexture, NULL, &startRect);

SDL\_FreeSurface(startSurface);

SDL\_DestroyTexture(startTexture);

// init "Exit"

SDL\_Surface\* exitSurface = TTF\_RenderText\_Blended(font, "Exit", textColor);

SDL\_Texture\* exitTexture = SDL\_CreateTextureFromSurface(renderer, exitSurface);

SDL\_Rect exitRect = { (SCREEN\_WIDTH - exitSurface->w) / 2, 300, exitSurface->w, exitSurface->h }; // Căn giữa theo chiều ngang

SDL\_RenderCopy(renderer, exitTexture, NULL, &exitRect);

SDL\_FreeSurface(exitSurface);

SDL\_DestroyTexture(exitTexture);

SDL\_RenderPresent(renderer);

SDL\_DestroyTexture(backgroundTexture);

}

}

void initializeSDL(SDL\_Window\*& window, SDL\_Renderer\*& renderer, TTF\_Font\*& font, SDL\_Texture\*& backgroundTexture, SDL\_Texture\*& playerTexture) {

SDL\_Init(SDL\_INIT\_VIDEO);

window = SDL\_CreateWindow("GUN BATTLE", SDL\_WINDOWPOS\_CENTERED, SDL\_WINDOWPOS\_CENTERED, SCREEN\_WIDTH, SCREEN\_HEIGHT, 0);

renderer = SDL\_CreateRenderer(window, -1, SDL\_RENDERER\_ACCELERATED);

TTF\_Init();

font = TTF\_OpenFont("D:/C++/Visual\_studio/SDL\_GAME\_1/lazy.ttf", 25);

srand(static\_cast<unsigned int>(time(NULL)));

// Load hình ảnh

SDL\_Surface\* backgroundSurface = IMG\_Load("D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\new\_bkground.png");

backgroundTexture = SDL\_CreateTextureFromSurface(renderer, backgroundSurface);

SDL\_FreeSurface(backgroundSurface);

SDL\_Surface\* playerSurface = IMG\_Load("D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\newbkground.png");

playerTexture = SDL\_CreateTextureFromSurface(renderer, playerSurface);

SDL\_Surface\* heartSurface = IMG\_Load("D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\heart.png");

if (heartSurface == nullptr) {

cerr << "Failed to load heart image: " << IMG\_GetError() << endl;

// Xử lý khi không thể tải hình ảnh

}

else {

heartTexture = SDL\_CreateTextureFromSurface(renderer, heartSurface);

SDL\_FreeSurface(heartSurface);

}

}

void handleInput(SDL\_Renderer\* renderer, GameObject& player, bool& isRunning) {

SDL\_Event event;

while (SDL\_PollEvent(&event)) {

switch (event.type) {

case SDL\_QUIT:

isRunning = false;

break;

case SDL\_KEYDOWN:

switch (event.key.keysym.sym) {

case SDLK\_UP:

player.y -= PLAYER\_SPEED; // Di chuyển lên khi nhấn phím mũi tên lên

break;

case SDLK\_DOWN:

player.y += PLAYER\_SPEED; // Di chuyển xuống khi nhấn phím mũi tên xuống

break;

case SDLK\_LEFT:

player.x -= PLAYER\_SPEED; // Di chuyển sang trái khi nhấn phím mũi tên trái

break;

case SDLK\_RIGHT:

player.x += PLAYER\_SPEED; // Di chuyển sang phải khi nhấn phím mũi tên phải

break;

}

break;

}

}

}

void movePlayer(GameObject& player) {

const Uint8\* keystate = SDL\_GetKeyboardState(NULL);

if (keystate[SDL\_SCANCODE\_UP] && player.y > 0) {

player.y -= PLAYER\_SPEED; // Di chuyển lên khi nhấn phím mũi tên lên

}

if (keystate[SDL\_SCANCODE\_DOWN] && player.y < SCREEN\_HEIGHT - player.h) {

player.y += PLAYER\_SPEED; // Di chuyển xuống khi nhấn phím mũi tên xuống

}

if (keystate[SDL\_SCANCODE\_LEFT] && player.x > 0) {

player.x -= PLAYER\_SPEED; // Di chuyển sang trái khi nhấn phím mũi tên trái

}

if (keystate[SDL\_SCANCODE\_RIGHT] && player.x < SCREEN\_WIDTH - player.w) {

player.x += PLAYER\_SPEED; // Di chuyển sang phải khi nhấn phím mũi tên phải

}

}

void fireBullet() {

static int bulletCooldown = 0;

const int BULLET\_COOLDOWN\_TIME = 10; // Thời gian chờ giữa các lần bắn đạn

if (bulletCooldown > 0) {

bulletCooldown--;

}

if (bulletCooldown == 0) {

Bullet bullet;

bullet.x = player.x + player.w / 2 - 2;

bullet.y = player.y;

//kích thước của đạn

bullet.w = 25;

bullet.h = 25;

bullet.active = true; // đạn đang hoạt động

bullets.push\_back(bullet);

bulletCooldown = BULLET\_COOLDOWN\_TIME; // init thời gian chờ giữa các lần bắn đạn

}

}

void spawnEnemy() {

Enemy enemy;

enemy.x = rand() % (SCREEN\_WIDTH - 40); // Random dich xuat hien

enemy.y = 0;

enemy.w = 60; // width

enemy.h = 60; // height

enemies.push\_back(enemy); // Thêm địch vào vector enemies

}

void moveBullets() {

for (int i = 0; i < bullets.size(); i++) {

if (bullets[i].active) {

bullets[i].y -= BULLET\_SPEED; // Di chuyển lên trên theo hướng y

if (bullets[i].y < 0) {

bullets[i].active = false; // out màn hình => mất

}

}

}

}

void moveEnemies() {

for (int i = 0; i < enemies.size(); i++) {

enemies[i].y += ENEMY\_SPEED; // Di chuyển địch xuống dưới theo hướng y

if (enemies[i].y > SCREEN\_HEIGHT) {

enemies.erase(enemies.begin() + i); //out màn hình = > mất

i--; // Giảm chỉ số i đi 1 sau khi xóa phần tử

}

}

}

//check va chạm

bool isColliding(const GameObject& obj1, const GameObject& obj2) {

if (obj1.x < obj2.x + obj2.w &&

obj1.x + obj1.w > obj2.x &&

obj1.y < obj2.y + obj2.h &&

obj1.y + obj1.h > obj2.y) {

return true;

}

return false;

}

void checkCollisions(SDL\_Renderer\* renderer) {

for (int i = 0; i < bullets.size(); i++) {

if (bullets[i].active) {

for (int j = 0; j < enemies.size(); j++) {

if (enemies[j].alive) {

if (isColliding(bullets[i], enemies[j])) {

// process đạn

bullets[i].active = false;

enemies[j].alive = false;

score += 10; // Tăng điểm khi trúng địch

}

}

}

}

}

for (int i = 0; i < enemies.size(); i++) {

if (enemies[i].alive && isColliding(player, enemies[i])) {

}

}

}

void drawPlayer(SDL\_Renderer\* renderer, GameObject player, SDL\_Texture\* playerTexture) {

SDL\_Rect playerRect = { player.x, player.y, player.w, player.h };

SDL\_RenderCopy(renderer, playerTexture, NULL, &playerRect);

}

void drawBullets(SDL\_Renderer\* renderer) {

// Load hình ảnh mới cho đạn

SDL\_Surface\* bulletSurface = IMG\_Load("D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\bullet\_bkground.png");

SDL\_Texture\* bulletTexture = SDL\_CreateTextureFromSurface(renderer, bulletSurface);

SDL\_FreeSurface(bulletSurface);

for (int i = 0; i < bullets.size(); i++) {

if (bullets[i].active) {

// Vẽ đạn sử dụng texture mới

SDL\_Rect bulletRect = { bullets[i].x, bullets[i].y, bullets[i].w, bullets[i].h };

SDL\_RenderCopy(renderer, bulletTexture, NULL, &bulletRect);

}

}

SDL\_DestroyTexture(bulletTexture); // Giải phóng texture

}

void drawEnemies(SDL\_Renderer\* renderer) {

// Load hình ảnh mới cho kẻ địch

SDL\_Surface\* enemySurface = IMG\_Load("D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\Enemies bkgrond.png");

SDL\_Texture\* enemyTexture = SDL\_CreateTextureFromSurface(renderer, enemySurface);

SDL\_FreeSurface(enemySurface);

for (int i = 0; i < enemies.size(); i++) {

if (enemies[i].alive) {

// Vẽ kẻ địch sử dụng texture mới

SDL\_Rect enemyRect = { enemies[i].x, enemies[i].y, enemies[i].w, enemies[i].h };

SDL\_RenderCopy(renderer, enemyTexture, NULL, &enemyRect);

}

}

SDL\_DestroyTexture(enemyTexture); // Giải phóng texture sau khi sử dụng

}

void drawScore(SDL\_Renderer\* renderer, TTF\_Font\* font, int score) {

SDL\_Color textColor = { 255, 255, 0 }; //yellow

string score\_str = "Score: " + to\_string(score);

SDL\_Surface\* textSurface = TTF\_RenderText\_Blended(font, score\_str.c\_str(), textColor);

SDL\_Texture\* textTexture = SDL\_CreateTextureFromSurface(renderer, textSurface);

int textWidth, textHeight;

SDL\_QueryTexture(textTexture, NULL, NULL, &textWidth, &textHeight);

SDL\_Rect textRect = { 10, 10, textWidth, textHeight }; // vi tri hien diem so

SDL\_RenderCopy(renderer, textTexture, NULL, &textRect);

SDL\_FreeSurface(textSurface);

SDL\_DestroyTexture(textTexture);

}

void cleanup(SDL\_Window\* window, SDL\_Renderer\* renderer, TTF\_Font\* font) {

TTF\_CloseFont(font);

TTF\_Quit();

SDL\_DestroyRenderer(renderer);

SDL\_DestroyWindow(window);

SDL\_Quit();

}

void runGameLoop(SDL\_Renderer\* renderer, SDL\_Window\* window, TTF\_Font\* font, SDL\_Texture\* backgroundTexture,

SDL\_Texture\* playerTexture, GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies,

int& score, bool& isRunning) {

SDL\_Event event;

int numHeartsRemaining = 3; // Số lượng trái tim ban đầu

while (isRunning) {

while (SDL\_PollEvent(&event)) {

if (event.type == SDL\_QUIT) {

isRunning = false; // Đặt biến isRunning thành false để kết thúc vòng lặp chính

}

}

handleInput(renderer, player, isRunning);

movePlayer(player);

fireBullet();

Uint32 currentTime = SDL\_GetTicks();

if (currentTime - lastEnemySpawn >= ENEMY\_SPAWN\_TIMER) {

spawnEnemy();

lastEnemySpawn = currentTime;

}

moveBullets();

moveEnemiesAndCheckHearts(renderer, numHeartsRemaining, isRunning, score);// Di chuyển enemy và kiểm tra trái tim

checkCollisions(renderer);

// Vẽ background

SDL\_RenderCopy(renderer, backgroundTexture, NULL, NULL);

drawPlayer(renderer, player, playerTexture);

drawBullets(renderer);

drawEnemies(renderer);

drawScore(renderer, font, score);

drawHearts(renderer, numHeartsRemaining); // Vẽ số lượng trái tim còn lại

SDL\_RenderPresent(renderer);

if (numHeartsRemaining <= 0) {

isRunning = false; // Kết thúc trò chơi khi hết trái tim

// Hiển thị thông báo "Game Over" hoặc thực hiện các hành động khác khi trò chơi kết thúc

SDL\_Color textColor = { 0, 255, 0 };

// Màu green

TTF\_SetFontSize(font, 50); // Đặt kích thước chữ cho thông báo "Game Over"

SDL\_Surface\* gameOverSurface = TTF\_RenderText\_Blended(font, "Game Over!", textColor);

SDL\_Texture\* gameOverTexture = SDL\_CreateTextureFromSurface(renderer, gameOverSurface);

int textWidth, textHeight;

SDL\_QueryTexture(gameOverTexture, NULL, NULL, &textWidth, &textHeight);

SDL\_Rect textRect = { (SCREEN\_WIDTH - textWidth) / 2, (SCREEN\_HEIGHT - textHeight) / 2, textWidth, textHeight }; // Vị trí để hiển thị thông báo "Game Over"

SDL\_RenderCopy(renderer, gameOverTexture, NULL, &textRect);

SDL\_FreeSurface(gameOverSurface);

SDL\_DestroyTexture(gameOverTexture);

SDL\_RenderPresent(renderer);

SDL\_Delay(3000); // Dừng chương trình trong 3 giây trước khi thoát

drawEndGameMenu(renderer, font,isRunning, player, lastEnemySpawn, bullets, enemies, score);

numHeartsRemaining = 3;

}

if (score >= 200 ) {

isRunning = false; // Kết thúc trò chơi khi đạt điều kiện chiến thắng

// Hiển thị thông báo "You Win" hoặc thực hiện các hành động khác khi trò chơi kết thúc

SDL\_Color textColor = { 0, 255, 0 }; // Màu xanh lá cây

TTF\_SetFontSize(font, 50); // Đặt kích thước chữ cho thông báo "You Win"

SDL\_Surface\* youWinSurface = TTF\_RenderText\_Blended(font, "You Win", textColor);

SDL\_Texture\* youWinTexture = SDL\_CreateTextureFromSurface(renderer, youWinSurface);

int textWidth, textHeight;

SDL\_QueryTexture(youWinTexture, NULL, NULL, &textWidth, &textHeight);

SDL\_Rect textRect = { (SCREEN\_WIDTH - textWidth) / 2, (SCREEN\_HEIGHT - textHeight) / 2, textWidth, textHeight }; // Vị trí để hiển thị thông báo "You Win"

SDL\_RenderCopy(renderer, youWinTexture, NULL, &textRect);

SDL\_FreeSurface(youWinSurface);

SDL\_DestroyTexture(youWinTexture);

SDL\_RenderPresent(renderer);

SDL\_Delay(3000); // Dừng chương trình trong 3 giây trước khi thoát

drawEndGameMenu(renderer, font, isRunning, player, lastEnemySpawn, bullets, enemies, score);

numHeartsRemaining = 3;

}

}

// Dọn dẹp và thoát chương trình sau khi người dùng thoát

cleanup(window, renderer, font);

}

void countdownTimer(SDL\_Renderer\* renderer, TTF\_Font\* font) {

// Load hình ảnh background

SDL\_Surface\* backgroundSurface = IMG\_Load("D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\Screenshot 2024-04-06 121850.png");

if (backgroundSurface == nullptr) {

// Xử lý khi không thể tải hình ảnh

cerr << "Failed to load background image: " << IMG\_GetError() << endl;

return;

}

SDL\_Texture\* backgroundTexture = SDL\_CreateTextureFromSurface(renderer, backgroundSurface);

SDL\_FreeSurface(backgroundSurface);

int countdown = 10; // Thời gian đếm ngược ban đầu

Uint32 startTime = SDL\_GetTicks(); // Thời gian bắt đầu đếm ngược

bool welcomeDisplayed = false;

bool ruleDisplayed = false;

bool heartLossDisplayed = false;

while (countdown > 0) {

// Xóa màn hình

SDL\_SetRenderDrawColor(renderer, 0, 0, 0, 255);

SDL\_RenderClear(renderer);

// Vẽ background lên cửa sổ

SDL\_RenderCopy(renderer, backgroundTexture, NULL, NULL);

// Hiển thị thời gian đếm ngược lên màn hình

SDL\_Color textColor = { 255, 255, 255 }; // Màu trắng

string countdownText = "Countdown: " + to\_string(countdown);

SDL\_Surface\* textSurface = TTF\_RenderText\_Blended(font, countdownText.c\_str(), textColor);

SDL\_Texture\* textTexture = SDL\_CreateTextureFromSurface(renderer, textSurface);

SDL\_Rect textRect = { 10, 10, textSurface->w, textSurface->h }; // Vị trí để hiển thị thời gian đếm ngược

SDL\_RenderCopy(renderer, textTexture, NULL, &textRect);

// Hiển thị thông báo "Welcome to Gun Battle!"

TTF\_SetFontSize(font, 35);

SDL\_Surface\* welcomeSurface = TTF\_RenderText\_Blended(font, "Welcome to Gun Battle!", textColor);

SDL\_Texture\* welcomeTexture = SDL\_CreateTextureFromSurface(renderer, welcomeSurface);

int welcomeTextWidth, welcomeTextHeight;

SDL\_QueryTexture(welcomeTexture, NULL, NULL, &welcomeTextWidth, &welcomeTextHeight);

SDL\_Rect welcomeRect = { (SCREEN\_WIDTH - welcomeTextWidth) / 2, (SCREEN\_HEIGHT - welcomeTextHeight) / 2 - 100, welcomeTextWidth, welcomeTextHeight };

SDL\_RenderCopy(renderer, welcomeTexture, NULL, &welcomeRect);

SDL\_FreeSurface(welcomeSurface);

SDL\_DestroyTexture(welcomeTexture);

// Hiển thị luật chơi "Get 200 points and you win"

TTF\_SetFontSize(font, 35);

SDL\_Surface\* ruleSurface = TTF\_RenderText\_Blended(font, "Reach 200 points to emerge victorious!", textColor);

SDL\_Texture\* ruleTexture = SDL\_CreateTextureFromSurface(renderer, ruleSurface);

int ruleTextWidth, ruleTextHeight;

SDL\_QueryTexture(ruleTexture, NULL, NULL, &ruleTextWidth, &ruleTextHeight);

SDL\_Rect ruleRect = { (SCREEN\_WIDTH - ruleTextWidth) / 2, (SCREEN\_HEIGHT - ruleTextHeight) / 2 + 75 - 100 , ruleTextWidth, ruleTextHeight };

SDL\_RenderCopy(renderer, ruleTexture, NULL, &ruleRect);

SDL\_FreeSurface(ruleSurface);

SDL\_DestroyTexture(ruleTexture);

// Hiển thị thông báo "Nếu mất 3 trái tim thì game over!"

TTF\_SetFontSize(font, 35);

SDL\_Surface\* heartLossSurface = TTF\_RenderText\_Blended(font, "Lose 3 hearts and face defeat!", textColor);

SDL\_Texture\* heartLossTexture = SDL\_CreateTextureFromSurface(renderer, heartLossSurface);

int heartLossTextWidth, heartLossTextHeight;

SDL\_QueryTexture(heartLossTexture, NULL, NULL, &heartLossTextWidth, &heartLossTextHeight);

SDL\_Rect heartLossRect = { (SCREEN\_WIDTH - heartLossTextWidth) / 2, (SCREEN\_HEIGHT - heartLossTextHeight) / 2 + 150 - 100, heartLossTextWidth, heartLossTextHeight };

SDL\_RenderCopy(renderer, heartLossTexture, NULL, &heartLossRect);

SDL\_FreeSurface(heartLossSurface);

SDL\_DestroyTexture(heartLossTexture);

TTF\_SetFontSize(font, 35);

SDL\_Surface\* moveInstructionSurface = TTF\_RenderText\_Blended(font, "Use the arrow keys to navigate your path.", textColor);

SDL\_Texture\* moveInstructionTexture = SDL\_CreateTextureFromSurface(renderer, moveInstructionSurface);

int moveInstructionTextWidth, moveInstructionTextHeight;

SDL\_QueryTexture(moveInstructionTexture, NULL, NULL, &moveInstructionTextWidth, &moveInstructionTextHeight);

SDL\_Rect moveInstructionRect = { (SCREEN\_WIDTH - moveInstructionTextWidth) / 2, (SCREEN\_HEIGHT - moveInstructionTextHeight) / 2 + 225 - 100, moveInstructionTextWidth, moveInstructionTextHeight };

SDL\_RenderCopy(renderer, moveInstructionTexture, NULL, &moveInstructionRect);

SDL\_FreeSurface(moveInstructionSurface);

SDL\_DestroyTexture(moveInstructionTexture);

SDL\_RenderPresent(renderer);

// Tính thời gian đã trôi qua

Uint32 currentTime = SDL\_GetTicks();

Uint32 elapsedTime = currentTime - startTime;

// Cập nhật thời gian đếm ngược

if (elapsedTime >= 1000) {

countdown--;

startTime = currentTime;

}

}

// Giải phóng texture của background

SDL\_DestroyTexture(backgroundTexture);

}

void drawHearts(SDL\_Renderer\* renderer, int numHeartsRemaining) {

// heart's position and size

int heartWidth = 50;

int heartHeight = 50;

int heartMargin = 5; // hearts's length

// Vẽ số lượng trái tim còn lại

int startX = SCREEN\_WIDTH - (numHeartsRemaining \* (heartWidth + heartMargin)) - 10;

int startY = 10;

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

SDL\_Rect heartRect = { startX + i \* (heartWidth + heartMargin), startY, heartWidth, heartHeight };

SDL\_RenderCopy(renderer, heartTexture, NULL, &heartRect);

}

}

void moveEnemiesAndCheckHearts(SDL\_Renderer\* renderer, int& numHeartsRemaining, bool& isRunning, int& score) {

for (int i = 0; i < enemies.size(); i++) {

enemies[i].y += ENEMY\_SPEED; // Di chuyển địch xuống dưới theo hướng y

bool enemyHitByBullet = false; // Biến kiểm tra xem kẻ địch có bị đánh bởi đạn hay không

for (int j = 0; j < bullets.size(); j++) {

if (bullets[j].active && isColliding(bullets[j], enemies[i])) {

bullets[j].active = false; // Đánh dấu đạn đã trúng kẻ địch

enemies.erase(enemies.begin() + i); // Xóa kẻ địch khi bị trúng đạn

i--; // Giảm chỉ số i đi 1 sau khi xóa phần tử

enemyHitByBullet = true; // Đánh dấu kẻ địch bị trúng đạn

score += 10; // Tăng điểm khi trúng địch

break; // Thoát khỏi vòng lặp khi kẻ địch bị trúng đạn

}

}

if (!enemyHitByBullet && enemies[i].y > SCREEN\_HEIGHT) {

enemies.erase(enemies.begin() + i); // Xóa kẻ địch khi thoát khỏi màn hình

i--; // Giảm chỉ số i đi 1 sau khi xóa phần tử

numHeartsRemaining--; // Giảm số lượng trái tim nếu kẻ địch thoát ra khỏi màn hình

if (numHeartsRemaining <= 0) {

isRunning = false; // Kết thúc trò chơi khi hết trái tim

break; // Thoát khỏi vòng lặp khi trò chơi kết thúc

}

}

}

}

void drawEndGameMenu(SDL\_Renderer\* renderer, TTF\_Font\* font, bool& isRunning, GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score) {

bool inEndGameMenu = true;

SDL\_Event event;

int numHeartsRemaining;

const string backgroundImagePath = "D:\\C++\\Visual\_studio\\SDL\_GAME\_1\\x64\\Debug\\bkground.bmp";

SDL\_Surface\* backgroundSurface = IMG\_Load(backgroundImagePath.c\_str());

if (backgroundSurface == nullptr) {

cerr << "Failed to load background image: " << IMG\_GetError() << endl;

return;

}

SDL\_Texture\* backgroundTexture = SDL\_CreateTextureFromSurface(renderer, backgroundSurface);

SDL\_FreeSurface(backgroundSurface);

while (inEndGameMenu) {

while (SDL\_PollEvent(&event)) {

if (event.type == SDL\_QUIT) {

inEndGameMenu = false;

isRunning = false;

}

else if (event.type == SDL\_MOUSEBUTTONDOWN) {

int mouseX, mouseY;

SDL\_GetMouseState(&mouseX, &mouseY);

if (mouseX >= 400 && mouseX <= 600 && mouseY >= 200 && mouseY <= 250) {

// Chọn "Play again" - Reset trạng thái của trò chơi và bắt đầu một ván mới

inEndGameMenu = false;

resetGame(player, lastEnemySpawn, bullets, enemies, score, numHeartsRemaining);

isRunning = true;

}

else if (mouseX >= 400 && mouseX <= 600 && mouseY >= 300 && mouseY <= 350) {

// Chọn "Exit" - Dừng trò chơi

inEndGameMenu = false;

isRunning = false;

}

}

}

SDL\_SetRenderDrawColor(renderer, 0, 0, 0, 255);

SDL\_RenderClear(renderer);

SDL\_Rect backgroundRect = { 0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT };

SDL\_RenderCopy(renderer, backgroundTexture, NULL, &backgroundRect);

// Vẽ "Play again"

SDL\_Color textColor = { 255, 255, 255 }; // Màu trắng

TTF\_SetFontSize(font, 50);

SDL\_Surface\* playAgainSurface = TTF\_RenderText\_Blended(font, "Play again", textColor);

SDL\_Texture\* playAgainTexture = SDL\_CreateTextureFromSurface(renderer, playAgainSurface);

SDL\_Rect playAgainRect = { (SCREEN\_WIDTH - playAgainSurface->w) / 2, 200, playAgainSurface->w, playAgainSurface->h };

SDL\_RenderCopy(renderer, playAgainTexture, NULL, &playAgainRect);

SDL\_FreeSurface(playAgainSurface);

SDL\_DestroyTexture(playAgainTexture);

// Vẽ "Exit"

SDL\_Surface\* exitSurface = TTF\_RenderText\_Blended(font, "Exit", textColor);

SDL\_Texture\* exitTexture = SDL\_CreateTextureFromSurface(renderer, exitSurface);

SDL\_Rect exitRect = { (SCREEN\_WIDTH - exitSurface->w) / 2, 300, exitSurface->w, exitSurface->h };

SDL\_RenderCopy(renderer, exitTexture, NULL, &exitRect);

SDL\_FreeSurface(exitSurface);

SDL\_DestroyTexture(exitTexture);

SDL\_RenderPresent(renderer);

}

SDL\_DestroyTexture(backgroundTexture);

}

void resetGame(GameObject& player, Uint32& lastEnemySpawn, vector<Bullet>& bullets, vector<Enemy>& enemies, int& score, int& numHeartsRemaining) {

player.x = SCREEN\_WIDTH / 2;

player.y = SCREEN\_HEIGHT - 50;

lastEnemySpawn = 0;

bullets.clear();

enemies.clear();

score = 0;

numHeartsRemaining = 3; // Thiết lập số lượng trái tim ban đầu

}