10/2：

1. 完成代码编写。
2. 安装所需的SDL和SDL\_mixer库。
3. 库连接成功，代码编译通过

#include <SDL.h>

#include <SDL\_mixer.h>

#include <iostream>

#include <string>

#include <vector>

class Song {

public:

std::string title;

std::string filepath;

Song(const std::string& t, const std::string& p) : title(t), filepath(p) {}

};

class Playlist {

private:

std::vector<Song> songs;

public:

void addSong(const std::string& title, const std::string& filepath) {

songs.emplace\_back(title, filepath);

}

void playNext() {

if (!songs.empty()) {

if (Mix\_PlayingMusic()) {

Mix\_HaltMusic(); // 停止当前音乐

}

Mix\_Music\* music = Mix\_LoadMUS(songs.front().filepath.c\_str());

if (music) {

Mix\_PlayMusic(music, 1);

std::cout << "Playing: " << songs.front().title << std::endl;

Mix\_FreeMusic(music);

}

else {

std::cerr << "Error loading music: " << Mix\_GetError() << std::endl;

}

}

else {

std::cout << "No songs in playlist." << std::endl;

}

}

void clear() {

songs.clear();

}

};

void initAudio() {

if (SDL\_Init(SDL\_INIT\_AUDIO | SDL\_INIT\_VIDEO) < 0) { // 添加 SDL\_INIT\_VIDEO

std::cerr << "SDL could not initialize! SDL\_Error: " << SDL\_GetError() << std::endl;

return;

}

if (Mix\_OpenAudio(44100, MIX\_DEFAULT\_FORMAT, 2, 2048) == -1) {

std::cerr << "SDL\_mixer could not initialize! SDL\_mixer Error: " << Mix\_GetError() << std::endl;

}

}

void cleanupAudio() {

Mix\_CloseAudio();

SDL\_Quit();

}

void displayMenu() {

std::cout << "1. Play next song" << std::endl;

std::cout << "2. Add song" << std::endl;

std::cout << "0. Exit" << std::endl;

}

int getUserChoice() {

int choice;

std::cout << "Enter your choice: ";

std::cin >> choice;

return choice;

}

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

// 初始化音频

initAudio();

Playlist playlist;

// 示例：添加歌曲

playlist.addSong("First Song", "path\_to\_your\_audio/song1.mp3"); // 替换为实际音频路径

playlist.addSong("Second Song", "path\_to\_your\_audio/song2.mp3"); // 替换为实际音频路径

// 创建窗口

SDL\_Window\* window = SDL\_CreateWindow("Music Player",

SDL\_WINDOWPOS\_UNDEFINED,

SDL\_WINDOWPOS\_UNDEFINED,

640, 480,

SDL\_WINDOW\_SHOWN);

if (!window) {

std::cerr << "Window could not be created! SDL\_Error: " << SDL\_GetError() << std::endl;

cleanupAudio();

return 1;

}

// 事件处理

SDL\_Event event;

bool running = true;

while (running) {

displayMenu();

int choice = getUserChoice();

switch (choice) {

case 1: // 播放下一首

playlist.playNext();

break;

case 2: // 添加歌曲

{

std::string title, filepath;

std::cout << "Enter song title: ";

std::cin.ignore(); // 清除前一次输入的换行符

std::getline(std::cin, title);

std::cout << "Enter song filepath: ";

std::getline(std::cin, filepath);

playlist.addSong(title, filepath);

}

break;

case 0: // 退出

playlist.clear();

running = false;

break;

default:

std::cout << "Invalid choice." << std::endl;

break;

}

// 处理事件

while (SDL\_PollEvent(&event)) {

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

running = false; // 窗口关闭事件

}

}

}

// 清理和退出

SDL\_DestroyWindow(window);

cleanupAudio();

return 0;

}

10/3：

1. 添加用户操作界面，准备进行测试
2. 简单绘制三个按钮，分别代表，切歌，添加歌曲和退出

#include <SDL.h>

#include <SDL\_mixer.h>

#include <iostream>

#include <string>

#include <vector>

class Song {

public:

std::string title;

std::string filepath;

Song(const std::string& t, const std::string& p) : title(t), filepath(p) {}

};

class Playlist {

private:

std::vector<Song> songs;

public:

void addSong(const std::string& title, const std::string& filepath) {

songs.emplace\_back(title, filepath);

}

void playNext() {

if (!songs.empty()) {

if (Mix\_PlayingMusic()) {

Mix\_HaltMusic(); // 停止当前音乐

}

Mix\_Music\* music = Mix\_LoadMUS(songs.front().filepath.c\_str());

if (music) {

Mix\_PlayMusic(music, 1);

std::cout << "Playing: " << songs.front().title << std::endl;

Mix\_FreeMusic(music);

}

else {

std::cerr << "Error loading music: " << Mix\_GetError() << std::endl;

}

songs.erase(songs.begin()); // 移除已播放的歌曲

}

else {

std::cout << "No songs in playlist." << std::endl;

}

}

void clear() {

songs.clear();

}

};

void initAudio() {

if (SDL\_Init(SDL\_INIT\_AUDIO | SDL\_INIT\_VIDEO) < 0) {

std::cerr << "SDL could not initialize! SDL\_Error: " << SDL\_GetError() << std::endl;

return;

}

if (Mix\_OpenAudio(44100, MIX\_DEFAULT\_FORMAT, 2, 2048) == -1) {

std::cerr << "SDL\_mixer could not initialize! SDL\_mixer Error: " << Mix\_GetError() << std::endl;

}

}

void cleanupAudio() {

Mix\_CloseAudio();

SDL\_Quit();

}

void displayMenu(SDL\_Renderer\* renderer) {

// 设置背景颜色为白色

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

SDL\_RenderClear(renderer);

// 设置文本颜色为黑色

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

// 在窗口中绘制简单的文本（替代真实的文本绘制方式，因为未实现TTF支持）

// 绘制按钮文本

// 这里只是代表文字部分，实际上需要用文本渲染库，例如SDL\_ttf

SDL\_Rect rect1 = { 50, 50, 200, 50 }; // 播放下一首按钮

SDL\_RenderFillRect(renderer, &rect1);

SDL\_Rect rect2 = { 50, 120, 200, 50 }; // 添加歌曲按钮

SDL\_RenderFillRect(renderer, &rect2);

SDL\_Rect rect3 = { 50, 190, 200, 50 }; // 退出按钮

SDL\_RenderFillRect(renderer, &rect3);

SDL\_RenderPresent(renderer);

}

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

initAudio();

Playlist playlist;

playlist.addSong("First Song", "path\_to\_your\_audio/song1.mp3"); // 替换为实际音频路径

playlist.addSong("Second Song", "path\_to\_your\_audio/song2.mp3"); // 替换为实际音频路径

SDL\_Window\* window = SDL\_CreateWindow("Music Player",

SDL\_WINDOWPOS\_UNDEFINED,

SDL\_WINDOWPOS\_UNDEFINED,

640, 480,

SDL\_WINDOW\_SHOWN);

if (!window) {

std::cerr << "Window could not be created! SDL\_Error: " << SDL\_GetError() << std::endl;

cleanupAudio();

return 1;

}

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

SDL\_Event event;

bool running = true;

while (running) {

displayMenu(renderer);

while (SDL\_PollEvent(&event)) {

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

running = false; // 窗口关闭事件

}

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

switch (event.key.keysym.sym) {

case SDLK\_RETURN: // 按下回车键播放下一首

playlist.playNext();

break;

case SDLK\_a: // 按下 'A' 键添加歌曲

{

std::string title, filepath;

std::cout << "Enter song title: ";

std::cin.ignore(); // 清除前一次输入的换行符

std::getline(std::cin, title);

std::cout << "Enter song filepath: ";

std::getline(std::cin, filepath);

playlist.addSong(title, filepath);

break;

}

case SDLK\_ESCAPE: // 按下 ESC 键退出

running = false;

break;

}

}

}

}

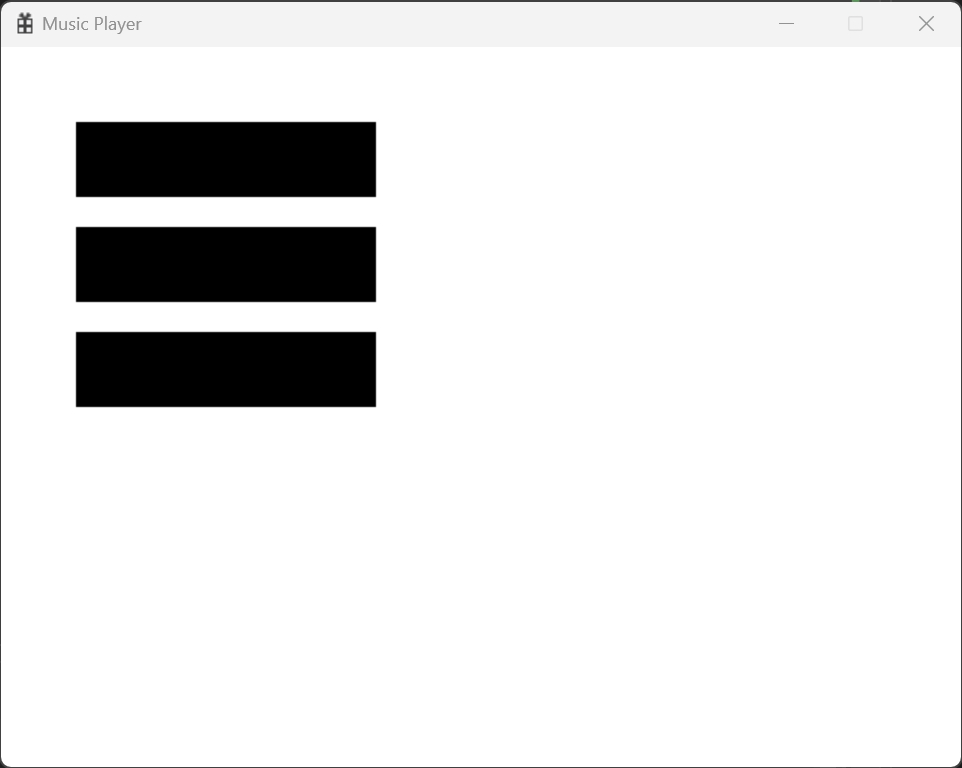
SDL\_DestroyRenderer(renderer);

SDL\_DestroyWindow(window);

cleanupAudio();

return 0;

}

1. 
2. 问题1：未实现TTF支持，操作界面只能用三个黑色举行替代按钮
3. 问题2：