

FILE README VỀ GAME FLAPPYBIRD

Ngày 18 tháng 5 năm 2024

1 Giới thiệu

- Ứng dụng các kiến thức đã học về phương pháp lập trình hướng đối tượng để lập trình game FlappyBird.
- Game được viết bằng ngôn ngữ lập trình C++ kết hợp với phương pháp lập trình hướng đối tượng và thư viện mã nguồn mở SDL.

2 Các công cụ hỗ trợ

1. Phần mềm soạn thảo Visual Studio 2022.
 2. Thư viện mã nguồn mở Simple DirectMedia Layer (SDL) phiên bản SDL2.
- Thư viện SDL2 là thư viện đa nền tảng, bao gồm các API để thao tác với âm thanh, bàn phím, chuột, joystick, graphics hardware thông qua OpenGL và Direct3D.
 - Thư viện SDL2 được viết bằng C chuẩn, nhưng hỗ trợ tốt với C++ và nhiều ngôn ngữ lập trình khác như: C#, Java, Pascal, PHP, Python,...
 - Thư viện SDL2 hỗ trợ trên các hệ điều hành như: Windows, Mac OS X, Linux, iOS và Android.
 - Thư viện SDL2 được dùng để phát triển trò chơi, phần mềm đa phương tiện, ứng dụng đồ họa, phần mềm giả lập.

3 Cài đặt

Để cài đặt game FlappyBird, bạn cần làm theo các bước sau:

Bước 1: Cài Đặt Phần mềm soạn thảo Visual Studio 2022.

Bước 2: Sau khi mở Visual Studio 2022, cài thêm hỗ trợ Desktop Development with C++ trong Visual Studio 2022.

Bước 3: Tải file (...) từ Github: ([link github](#)).

Bước 4: Chạy file.

4 Mô tả mã nguồn

- **BaseObject**: Lớp cơ bản (lớp cha) đại diện cho các đối tượng đồ họa trong game, các hàm thành viên bao gồm:

+ virtual bool LoadImageFile(std::string path, SDL_Renderer* screen): Tải hình ảnh từ đường dẫn path và hiển thị lên bề mặt screen.

+ void Render(SDL_Renderer* des, const SDL_Rect* clip = NULL): Vẽ đối tượng lên bề mặt renderer des với tùy chọn cắt (clip).

+ void Free(): Giải phóng tài nguyên liên quan đến đối tượng.

+ void SetRect(const int& x, const int& y): Thiết lập vị trí của đối tượng.

+ SDL_Rect GetRect() const: Trả về hình chữ nhật đại diện cho vị trí và kích thước của đối tượng.

+ SDL_Texture* GetObject(): Trả về con trỏ đến hình ảnh của đối tượng.

- **TextObject**: Lớp này xử lý việc hiển thị văn bản lên bề mặt renderer của SDL2, các hàm thành viên bao gồm:

- + bool loadFromRenderedText(TTF_Font* gFont, SDL_Renderer* screen): Tạo hình ảnh từ văn bản sử dụng font gFont.
- + void Free(): Giải phóng tài nguyên liên quan đến đối tượng văn bản.
- + void setColor(Uint8 red, Uint8 green, Uint8 blue): Thiết lập màu sắc của văn bản.
- + void setColor(int type): Thiết lập màu sắc dựa trên các loại màu đã được định nghĩa (ví dụ: đỏ, trắng, đen).
- + void RenderText(SDL_Renderer* screen, int x, int y, SDL_Rect* clip = NULL, double angle = 0.0, SDL_Point* center = NULL, SDL_RendererFlip flip = SDL_FLIP_NONE):
Hiển thị văn bản lên bề mặt renderer screen tại vị trí (x, y) với các tham số tùy chọn cho việc cắt, xoay và lật.
- + int getWidth() const: Trả về chiều rộng của văn bản đã được hiển thị.
- + int getHeight() const: Trả về chiều cao của văn bản đã được hiển thị.
- + void SetText(const std::string& text): Thiết lập nội dung văn bản để hiển thị.
- + std::string GetText() const: Trả về nội dung văn bản hiện tại.
- **BlockObject**: Đây là lớp đại diện cho một khối (block) trong ứng dụng, lớp này kế thừa từ lớp BaseObject, các hàm thành viên bao gồm:
 - + bool LoadImg(std::string path, SDL_Renderer* screen): Tải hình ảnh từ đường dẫn path và hiển thị lên bề mặt screen.
 - + void SetPos(const int& xp, const int& yp): Thiết lập vị trí của khối.
 - + void SetXPos(const int&xp): Thiết lập vị trí theo trục x của khối.
 - + SDL_Rect GetRectPos() const: Trả về hình chữ nhật đại diện cho vị trí và kích thước của khối.
 - + void ShowImg(SDL_Renderer* screen): Hiển thị hình ảnh của khối lên bề mặt

screen.

+ void DoRun(UINT& x_val): Xử lý chuyển động của khối.

+ bool GetStateBack() const: Trả về trạng thái của khối (đang ở phía sau hay không).

+ void SetBack(bool b): Thiết lập trạng thái của khối (đang ở phía sau hay không).

- **PlayerObject**: Đây là lớp đại diện cho một đối tượng người chơi trong ứng dụng, lớp này kế thừa từ lớp BaseObject, các hàm thành viên bao gồm:

+ bool LoadImg(std::string path, SDL_Renderer* screen): Tải hình ảnh từ đường dẫn path và hiển thị lên bề mặt screen.

+ void Show(SDL_Renderer* des): Hiển thị đối tượng người chơi lên bề mặt renderer des.

+ void HandleInputAction(SDL_Event events, SDL_Renderer* screen): Xử lý các hành động đầu vào (ví dụ: sự kiện bàn phím) liên quan đến người chơi.

+ void DoFalling(SDL_Renderer* des): Xử lý hành vi rơi tự do (ví dụ: tác động của trọng lực) cho người chơi.

+ bool GetFalling() const: Trả về trạng thái người chơi có đang rơi tự do hay không.

+ void SetFalling(bool falling): Thiết lập trạng thái rơi tự do của người chơi.

+ void DoGround(SDL_Renderer* screen): Xử lý tương tác với mặt đất (ví dụ: kiểm tra va chạm).

+ bool GetIsDie() const: Trả về trạng thái người chơi có trong trạng thái “chết” hay không.

- **ImpTimer**: Đây là lớp đại diện cho một đối tượng thời gian (timer) trong ứng dụng, các hàm thành viên bao gồm:

+ void start(): Bắt đầu đếm thời gian.

- + void stop(): Dừng đếm thời gian.
- + void pause(): Tạm dừng đếm thời gian.
- + void unpause(): Tiếp tục đếm thời gian sau khi tạm dừng.
- + int get_ticks(): Trả về số ticks (đơn vị thời + gian) đã trôi qua.
- + bool is_started(): Trả về trạng thái đang đếm + thời gian hay không.
- + bool is_paused(): Trả về trạng thái đang tạm dừng hay không.
- **Geometric**: Đây là lớp đại diện cho các chức năng liên quan đến hình học và vẽ hình trên bề mặt renderer, các hàm thành viên bao gồm:
 - + static void RenderRectangle(const GeometricFormat& geo_size, const ColorData& color, SDL_Renderer* screen): Vẽ một hình chữ nhật đặc (filled rectangle) trên bề mặt renderer screen sử dụng thông tin về kích thước hình (geo_size) và màu sắc (color).
 - + static void RenderOutline(const GeometricFormat& geo_size, const ColorData& color, SDL_Renderer* screen): Vẽ viền (outline) của một hình chữ nhật trên bề mặt renderer screen sử dụng thông tin về kích thước hình và màu sắc.
- **CommonFunc**: Cung cấp các công cụ và hàm tiện ích cho việc làm việc với đồ họa và sự kiện trong game sử dụng thư viện SDL, các hàm thành viên bao gồm:
 - + bool CheckCollision(const SDL_Rect& object1, const SDL_Rect& object2): Kiểm tra va chạm giữa hai hình chữ nhật.
 - + int ShowMenu(SDL_Renderer* g_screen, TTF_Font* font, const std::string& menu1, const std::string& menu2, const std::string& img_name): Hiển thị menu trên bề mặt renderer g_screen sử dụng font font và các tùy chọn menu.
 - + int MyRandom(int min, int max): Tạo số ngẫu nhiên trong khoảng từ min đến max.

HẾT.