Cập nhật thêm các file: sprite, dxinput

Sửa đổi cấu trúc vòng lặp:

Bao gồm 3 thành phần:

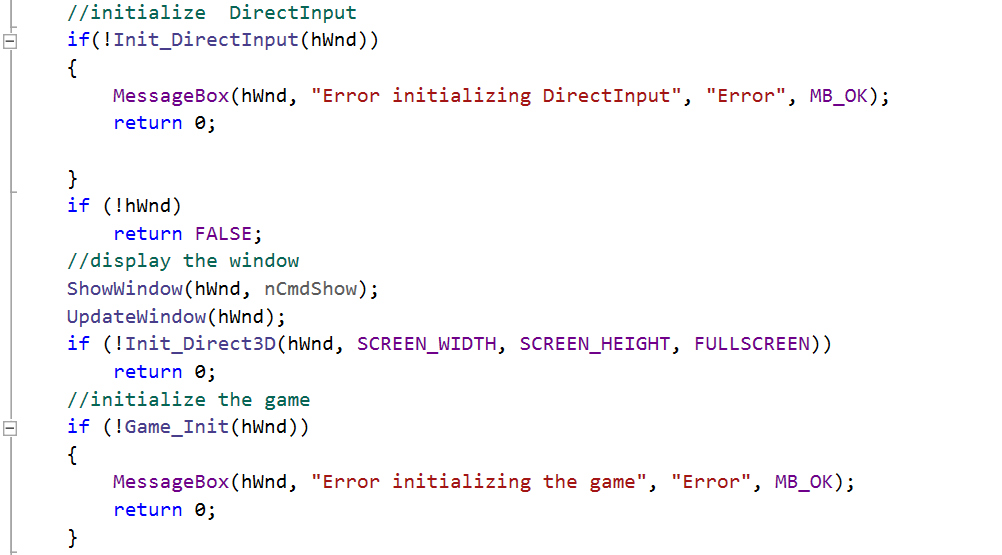
InputUpdate – cập nhật các thay đổi dựa trên tín hiệu nhận vào từ bàn phím

PhysicsUpdate – cập nhật thế giới vật lý trong game

GraphicUpdate – tiến hành vẽ các vật thể để thể hiện lên màn hình

Các file thủ tục đã được trừu tượng hóa, chỉ cần chạy hàm khởi tạo trước khi vào vòng lặp chính (Main Loop) giúp tách biệt phần logic game ra để dễ quản lí hơn. Các file đến nay bao gồm:

Winmain, dxgraphic, dxinput



Đây gọi các hàm ở đây (ở trong file winmain ngay trên main loop)

Cách dung file Sprite:

Constructor cho sprite

Sprite::Sprite(LPCSTR FilePath, int Width, int Height, int Count, int SpritesPerRow)

Các tham số lần lượt là: đường dẫn, độ dài và chiều cao của mỗi ảnh cắt, tổng số hoạt ảnh và số hoạt ảnh mỗi hang.

void Sprite::Next()

hàm đây để chọn cho sprite load hoạt ảnh tiếp theo một cách tuần hoàn (từ hoạt cuối quay về hoạt ảnh 0)

void Sprite::Next2()

hàm đây là hàm không tuần hoàn ( đến hoạt ảnh cuối thì ngừng lại luôn)

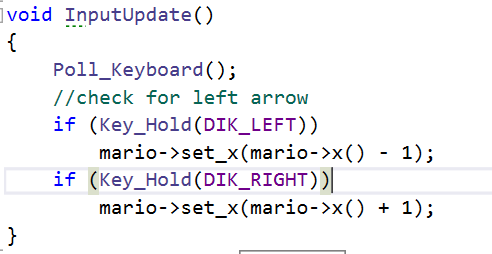
void Sprite::Render(int X, int Y, D3DXVECTOR3\* AnchorPoint)

hàm để vẽ sprite lên backbuffer (thực chất là vẽ thông qua device mà device lại có 1 con trỏ buffer do nó quản lí đã được get từ trước, you know it’s a mess)

Anchorpoint là điểm neo, gọi là trọng tâm của sprite cho dễ hiểu

Cách sử dụng Key\_Hold():

Ví dụ:



Poll\_Keyboard() là hàm thu hết những tín hiệu có thể được kích hoạt từ bàn phím (256 tín hiệu), vẫn chưa hiểu tại sao phải tái thu mỗi lần lặp vòng lặp

Đã fix thành công lỗi outfocus của chương trình!! (cần thêm acquire keyboard ở thông điệp wm\_activate – thông điệp được gửi lại mỗi khi chọn window phần mềm này làm foreground)

Chương trình ở sau khi escape sẽ báo lỗi null, đơn giản là vì trưởng nhóm đã giải phóng hết những thành phần được cấp phát trước đó

That’s it for today!!