**CHƯƠNG 2: PHƯƠNG PHÁP NGHIÊN CỨU**

## 2.1. Cài đặt DevC và thư viện graphics.h

Tải file cài đặt phần mềm DevC++ theo đường dẫn trong mục 1.4. Sau đó mở file

vừa tải, và tiến hành cài đặt. Thư viện graphics.h được tiến hành cài đặt theo các bước:

**Bước 1:** Copy 6-ConsoleAppGraphics và ConsoleApp\_cpp\_graph

Paste C:\...\Dev-Cpp\Templates

**Bước 2:** Copy graphics và winbgim

Paste C:\... Dev-Cpp \MinGW64\x86\_64-w64-mingw32\include

**Bước 3:** Copy libbgi.a

Paste C:\...\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\lib

**Bước 4:** Ở DevC++ => New Project => Console Graphics Application

**Bước 5:** Thay đổi Tools - Complier Option: TDM – GCC 4.9.2 32 bit Release

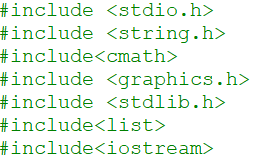
**Bước 6:** Sử dụng đoạn code mẫu bên dưới để test thư viện winbgim

|  |  |
| --- | --- |
| Capture | Capture2 |

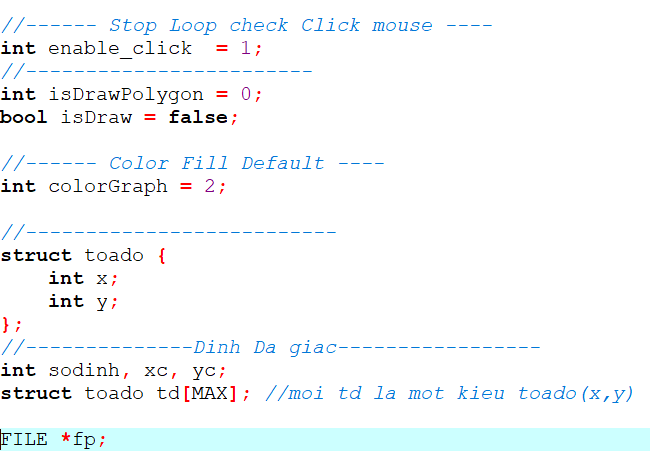
Hình 2.1. Ví dụ minh hoạ thư viện winbgim

## 2.2. Cài đặt thuật toán Scanline

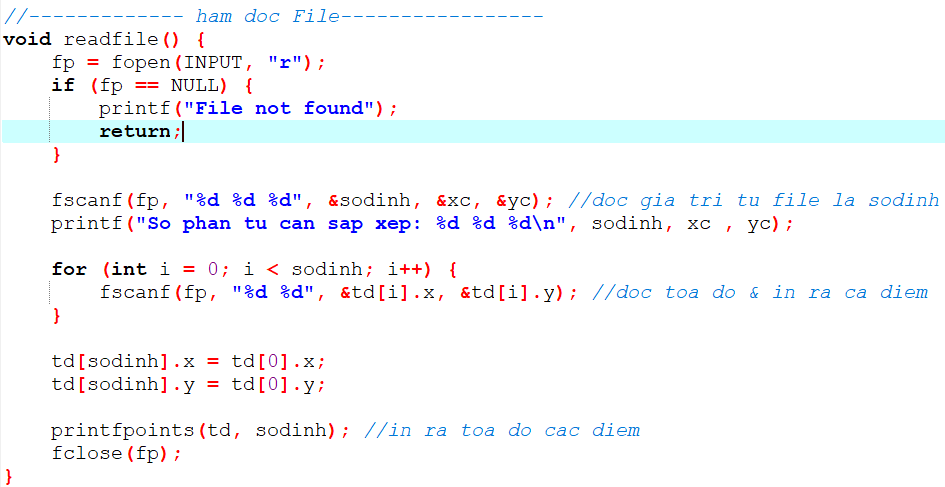
**Khai báo thư viện**



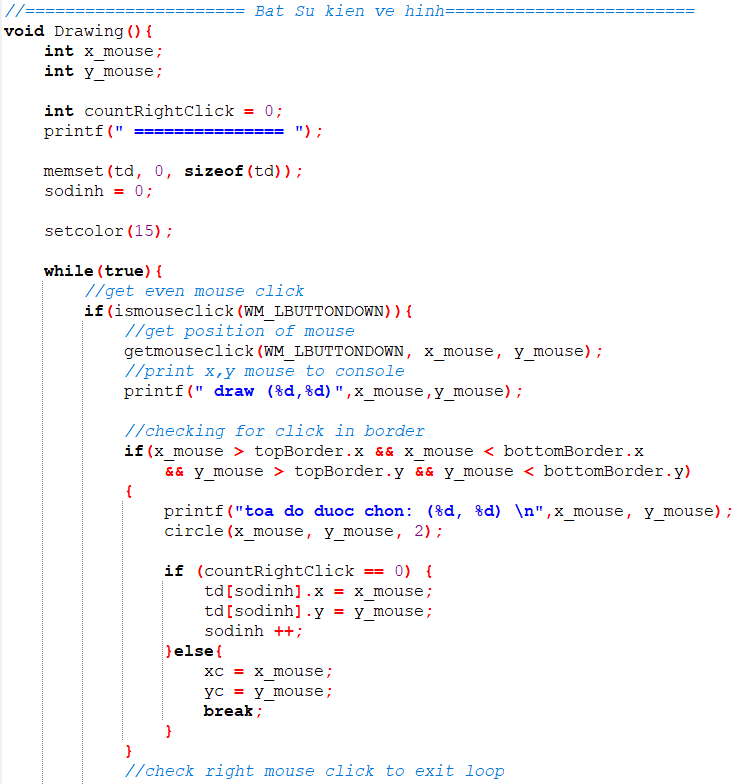
**Khai báo biến**



**Tạo đa giác:** Mỗi đỉnh của đa giác bao gồm hoành độ X[] và tung độ Y[] được lưu trữ trong mảng struct xd. Mỗi phần tử là một struct gồm tọa độ X và Y. Tọa độ các đỉnh của đa giác được lấy từ file thông qua hàm readfile() và thông qua thao tác chuột với hàm Drawing().

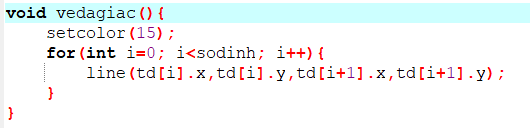


**Hàm đọc file**

****

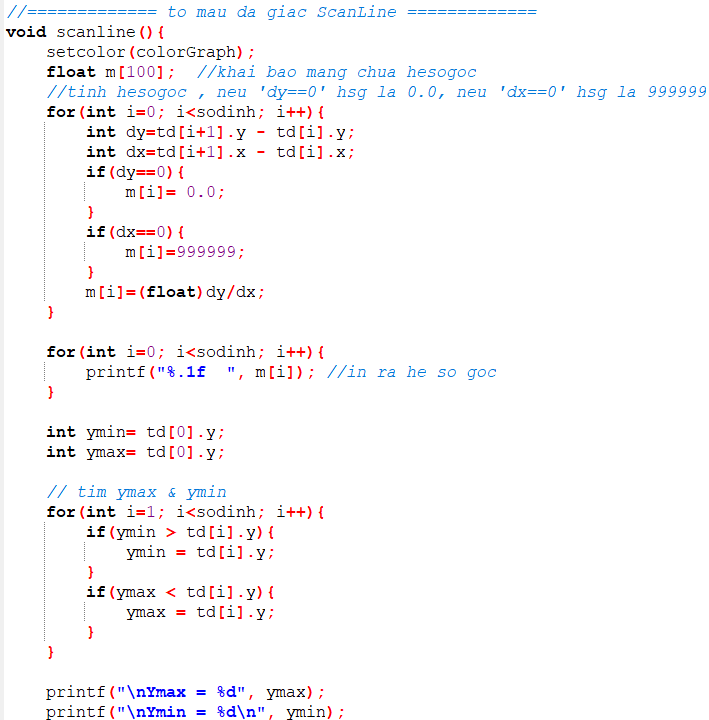
**Hàm vẽ đa giác với thao tác chuột**

**Vẽ đa giác:** Sau khi đã xác định được tọa độ các đỉnh, đa giác sẽ được hiển thị lên màn hình thông qua gọi hàm vedagiac().

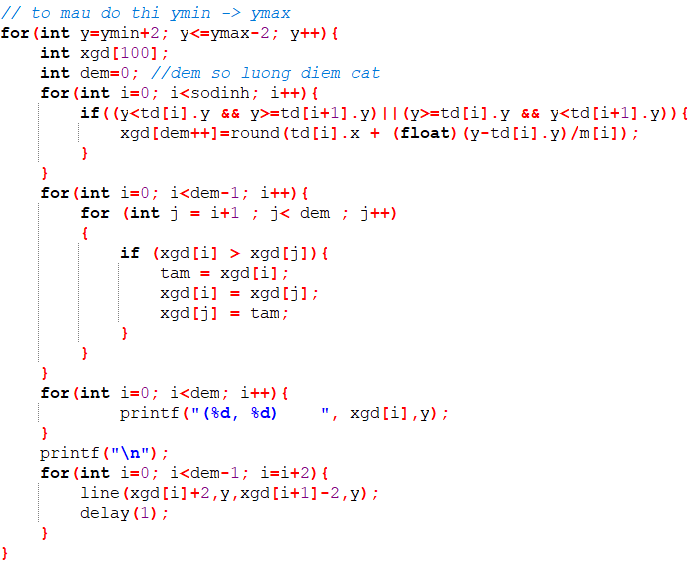


**Hàm vẽ đa giác**

Để tô màu đồ thị với thuật toán Scanline, chương trình cần tính toán hệ số góc trên mỗi cạnh của đa giác và đồng thời xác định tung độ của đỉnh lớn nhất (yMax) và bé nhất (yMin).

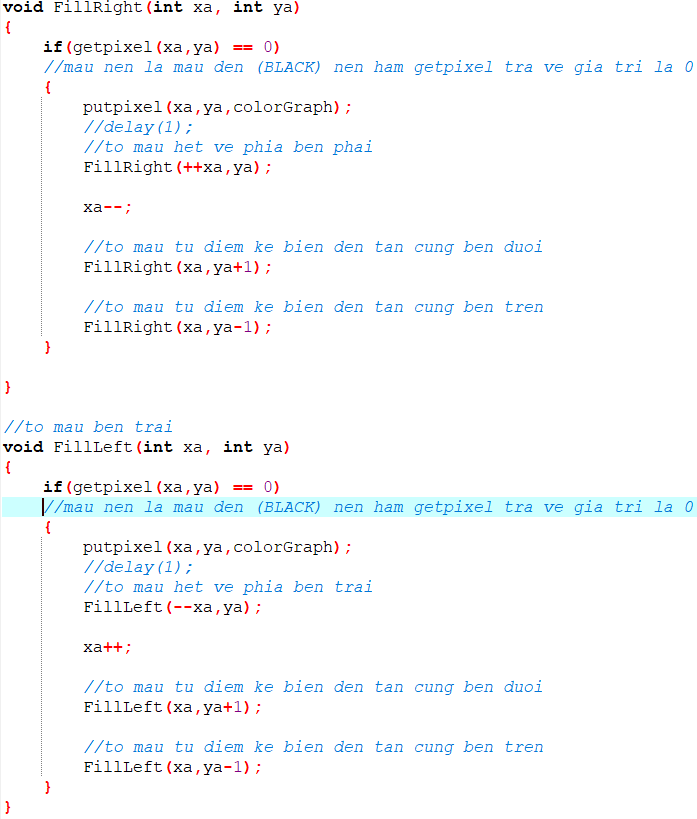


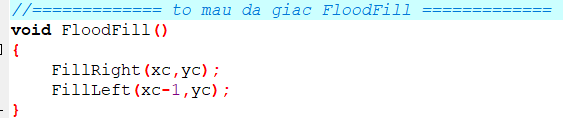
Sau đó chương trình sẽ thực hiện tô màu của đa giác với thuật toán Scanline



**Thuật toán tô màu đồ thị Scanline**

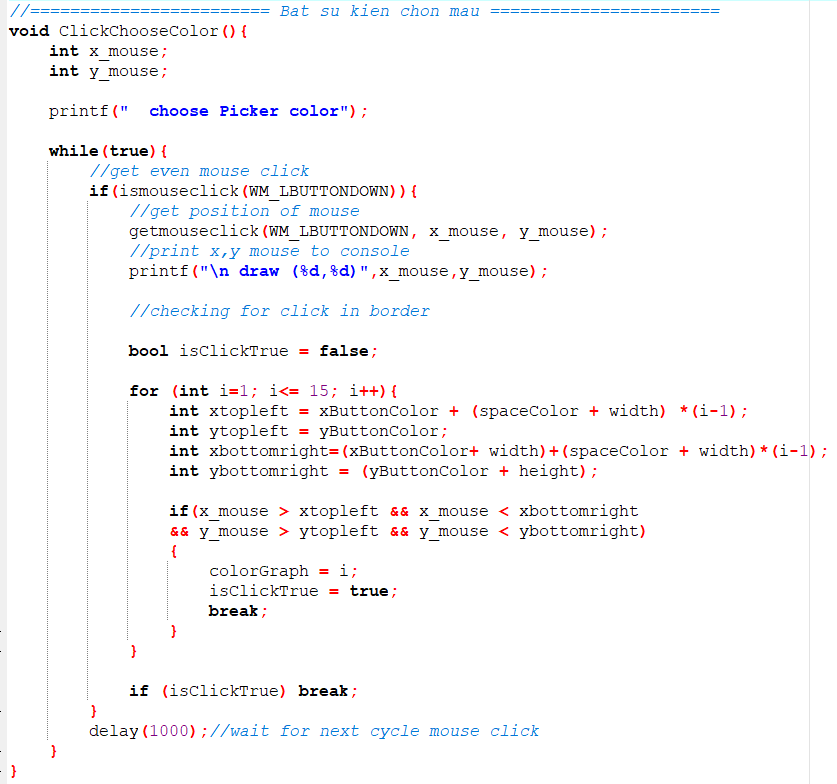
Với thuật toán FloodFill, chương trình định nghĩa hai hàm tô màu từ điểm khai báo sang bên trái và sang bên phải.





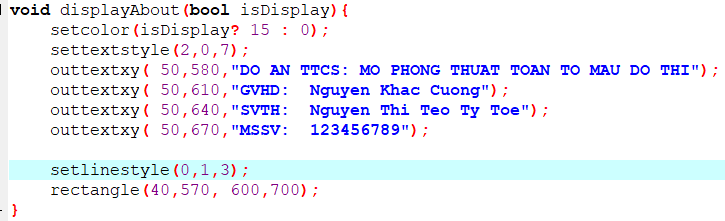
**Thuật toán tô màu đa giác FloodFill**

Người dùng có thể thay đổi màu của đa giác thông qua hàm ClickChooseColor()



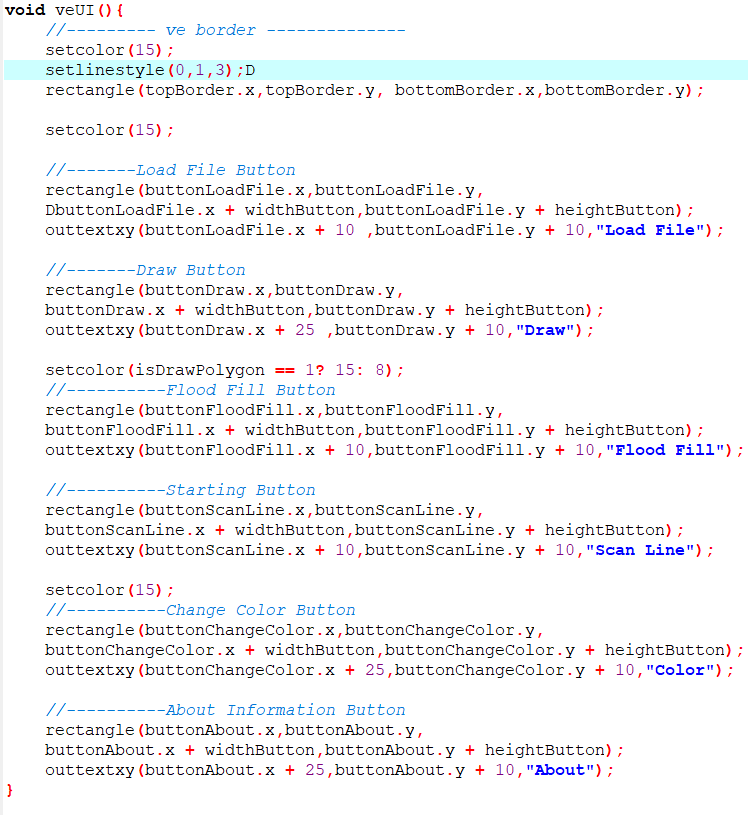
**Hàm chọn màu cho đa giác**

Người dùng có thể xem thông tin chi tiết của đề tài thông qua hàm displayAbout()



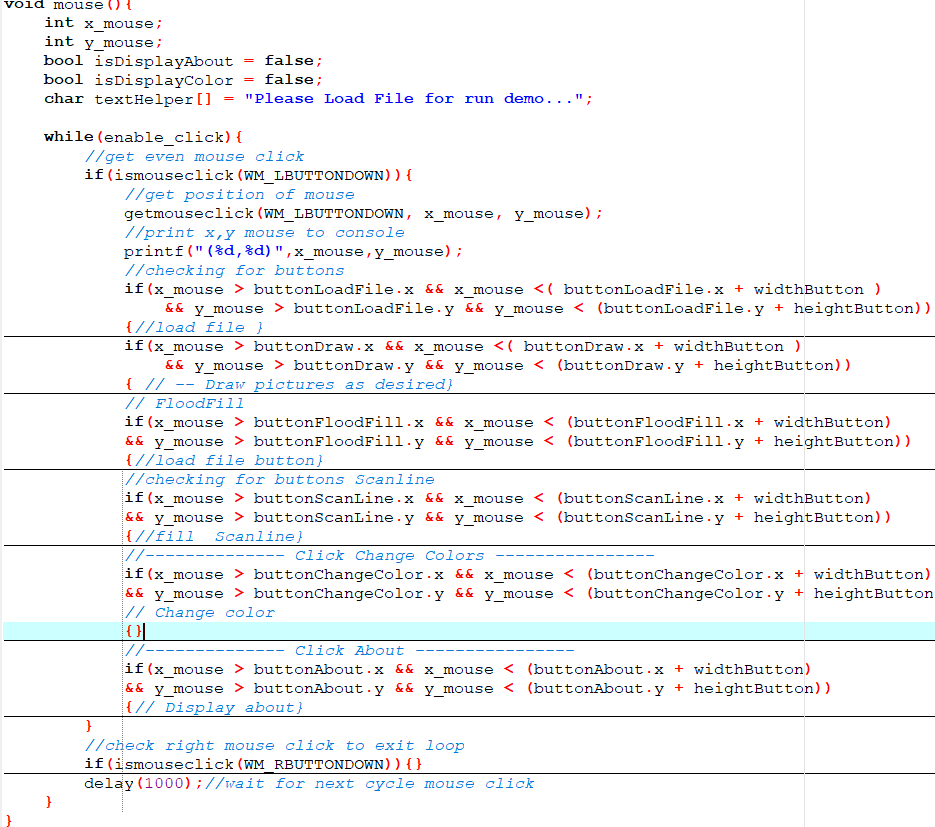
**Hàm hiển thị thông tin của đề tài**

Màn hình chính của chương trình bao gồm một khung hiển thị đa giác, các button với các sự kiện như lấy dữ liệu đa giác từ file, vẽ đa giác với thao tác chuột, tô màu đa giác với thuật toán FloodFill, tô màu đa giác với thuật toán Scanline, thay đổi màu sắc đa giác, xem thông tin chi tiết của đề tài.

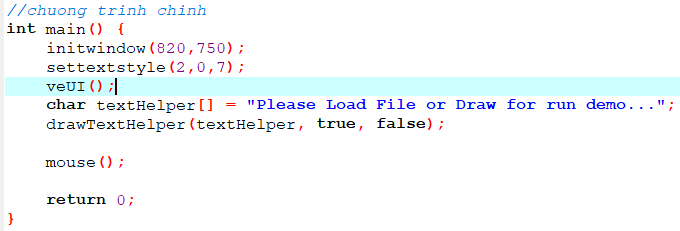


**Hàm vẽ giao diện người dùng**

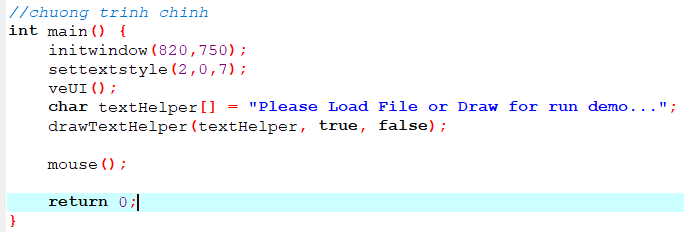
Khi người dùng chọn các button được hiển thị, chương trình sẽ kiểm tra button nào được chọn và gọi các hàm tương ứng để xử lý.

****

**Hàm xử lý các sự kiện click chuột**

****

**Chương trình chính**

****