BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**SỬ DỤNG THƯ VIỆN ĐỒ HỌA.H XÂY DỰNG GIAO DIỆN ĐỒ HỌA MINH HỌA CHỨC NĂNG ĐƠN GIẢN NHƯ MS PAINT**

**Giảng viên hướng dẫn : Đoàn Vũ Thịnh**

**Sinh viên thực hiện : Nguyễn Đình Nguyên**

**Mã số sinh viên : 64131538**

**Lớp : 64.CNTT-3**

Khánh Hòa - 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**SỬ DỤNG THƯ VIỆN ĐỒ HỌA.H XÂY DỰNG GIAO DIỆN ĐỒ HỌA MINH HỌA CHỨC NĂNG ĐƠN GIẢN NHƯ MS PAINT**

**Giảng viên hướng dẫn : Đoàn Vũ Thịnh**

**Sinh viên thực hiện : Nguyễn Đình Nguyên**

**Mã số sinh viên : 64131538**

Khánh Hòa - 2024

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa: Công nghệ Thông tin**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ**

**Tên đề tài:** Sử dụng thư viện đồ họa.h xây dựng giao diện đồ họa minh họa chức năng đơn giản như MS Paint

**Giảng viên hướng dẫn:** ThS. Đoàn Vũ Thịnh

**Sinh viên được hướng dẫn:** Nguyễn Đình Nguyên

**MSSV:** 64131538

**Khóa:** 64.CNTT-3 **Ngành:** Công nghệ Thông tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 | 28/11/2024 | Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. |  |
| 2 | 8/12/2024 | Trình bày việc mô phỏng thuật toán chính dựa trên kiến thức đã được học ở những học phần trước đó. |  |
| 3 | 15/12/2024 | Sinh viên tiến hành thực hiện thuật toán đã đề ra với việc tạo ra các cách nhập dữ liệu đầu vào đa dạng như bàn phím, sử dụng chuột, random. |  |
| 4 | 22/12/2024 | Hoàn thành phần xây dựng giao diện và sửa lại giao diện cho hoàn chỉnh |  |
| 5 | 28/12/2024 | Sinh viên thực hiện demo và sửa những lỗi phát sinh trong quá trình thực hiện |  |
| 6 | 5/1/2025 | Sinh viên nộp bản thảo lần cuối sau khi đã chỉnh sửa các yêu cầu như đã đề ra. |  |

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc185827420)

[TÓM TẮT 2](#_Toc185827421)

[Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI 3](#_Toc185827422)

[1.1. Giới thiệu MS Paint 3](#_Toc185827423)

[1.2. Một số chức năng cơ bản của MS Paint 3](#_Toc185827424)

[1.3. DevC/C++ 3](#_Toc185827425)

[1.4. Graphics.h 4](#_Toc185827426)

[1.5. Giải thuật vẽ đoạn thẳng thông thường 4](#_Toc185827427)

[1.6. Giải thuật sinh đa giác 5](#_Toc185827428)

[1.7. Giải thuật sinh đường tròn Midpoint 6](#_Toc185827429)

[Chương 2: PHƯƠNG PHÁP NGHIÊN CỨU 9](#_Toc185827430)

[2.1. Cài đặt Dev-C++ và Graphics.h 9](#_Toc185827431)

[2.2. Tổng quan về thiết kế giao diện 9](#_Toc185827432)

[2.2.1. Sơ lược giao diện 9](#_Toc185827433)

[2.2.2. Xây dựng giao diện 10](#_Toc185827434)

[2.3 Xử lý sự kiện chuột 11](#_Toc185827435)

[2.4. Vẽ hình 13](#_Toc185827436)

[2.5. Lưu hình 13](#_Toc185827437)

[2.6. Hiển thị trạng thái 14](#_Toc185827438)

[Chương 3: KẾT QUẢ 15](#_Toc185827439)

[Chương 4: KẾT LUẬN 20](#_Toc185827440)

[4.1 Ưu điểm 20](#_Toc185827441)

[4.2 Nhược điểm 20](#_Toc185827442)

**MỤC LỤC HÌNH ẢNH**

[Hình 1.1 Giao diện của Microsoft Paint 3](#_Toc186412742)

[Hình 1.2 Mô phỏng giải thuật sinh đa giác 6](#_Toc186412743)

[Hình 1.3 Giải thuật sinh đường tròn Midpoint 6](#_Toc186412744)

[Hình 1.4 Giải thuật sinh đường tròn Midpoint 7](#_Toc186412745)

[Hình 1.4 Giải thuật sinh đường tròn Midpoint 8](#_Toc186412746)

[Hình 2.1 Đoạn code demo cài đặt thư viện graphics.h vào Dev-C++ 9](#_Toc186412747)

[Hình 2.2.1 Giao diện chính sản phẩm 10](#_Toc186412748)

[Hình 2.2.2.1 Cửa sổ đồ họa BGI 10](#_Toc186412749)

[Hình 2.3 Xử lý sự kiện chuột 12](#_Toc186412750)

[Hình 2.3 Xử lý màu sắc 12](#_Toc186412751)

[Hình 2.3 Xử lý vẽ polyline và polygon 13](#_Toc186412752)

[Hình 2.4 Code vẽ hình 13](#_Toc186412753)

[Hình 2.5 Code lưu hình ảnh 14](#_Toc186412754)

[Hình 2.6 Code hiển thị trạng thái 15](#_Toc186412755)

[Hình 3.1 Giao diện chính của sản phẩm 16](#_Toc186412756)

[Hình 3.2 Giao diện vẽ đường thẳng 17](#_Toc186412757)

[Hình 3.3 Giao diện vẽ hình chữ nhật 17](#_Toc186412758)

[Hình 3.4 Giao diện vẽ hình tròn 18](#_Toc186412759)

[Hình 3.5 Giao diện vẽ đa đoạn thẳng 18](#_Toc186412760)

[Hình 3.6 Giao diện vẽ đa giác 19](#_Toc186412761)

[Hình 3.7 Giao diện vẽ màu 19](#_Toc186412762)

[Hình 3.8 Giao diện tăng giảm độ dày 20](#_Toc186412763)

[Hình 3.9 Giao diện lưu ảnh 20](#_Toc186412764)

# LỜI CẢM ƠN

Để có thể hoàn thành đợt thực tập lần này, em xin chân thành cảm ơn đến quý thầy cô khoa Công nghệ Thông tin đã tạo điều kiện hỗ trợ và giúp đỡ em trong quá trính học tập và thực hiện đề tài này.

Qua đây, em xin chân thành cảm ơn thầy Đoàn Vũ Thịnh, người đã trực tiếp quan tâm và hướng dẫn em hoàn thành tốt đợt thực tập trong thời gian qua.

Do kiến thức còn hạn chế và thời gian thực hiện còn ngắn nên bài báo cáo của em còn nhiều thiếu sót, kính mong sự góp ý của quý thầy cô.

Em xin chân thành cảm ơn!

# TÓM TẮT

Đề tài được chia làm bốn phần: Giới thiệu về đề tài, phương pháp nguyên cứu, kết quả và kết luận

Trong phần giới thiệu về đề tài sẽ giới thiệu về MS Paint và các chức năng cơ bản của MS Paint. Tiếp theo, tìm hiểu về phần mềm Dev C++ và thư viện Graphics.h và các giải thuật đồ họa

Trong phần phương pháp nguyên cứu, chúng ta biết được cách cài đặt phần mềm Dev C++ và thư viện Graphics.h. Sau đó, xây dựng giao diện và các chức năng của chúng

Trong phần kết quả sẽ hiển thị ra giao diện chính

# Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI

## **1.1. Giới thiệu MS Paint**

Microsoft Paint, hay còn gọi là Paint, là một phần mềm đồ họa chỉnh sửa hình ảnh được phát triển bởi Microsoft. Paint được cài đặt như một phần mềm mặc định trong tất cả các phiên bản của Windows. Paint có thể xử lý ảnh ở nhiều định dạng khác nhau, bao gồm từ JPG, JPEG, PNG, GIF, BMP, WebP…

Paint là một ứng dụng chỉnh sửa hình ảnh đơn giản và không hề cung cấp những tính năng chỉnh sửa ảnh nâng cao như Photoshop hoặc những phần mềm chỉnh sửa ảnh chuyên nghiệp khác. Một số tính năng cơ bản mà Paint cung cấp bao gồm: Viết chữ lên ảnh, thay đổi kích thước của ảnh, cắt ảnh, xóa một phần trong ảnh, vẽ các đường hoặc các hình khối…



Hình 1.1 Giao diện của Microsoft Paint

## **1.2. Một số chức năng cơ bản của MS Paint**

* Tạo và chỉnh sửa hình ảnh
* Màu sắc và tô màu
* Chèn văn bản
* Cắt, dán và thay đổi kích thước hình ảnh
* Làm việc với nhiều định dạng tệp
* Tẩy xóa và chỉnh sửa đơn giản
* Lưới và thu phóng

## **1.3. DevC/C++**

Dev-C++ 4.9.9.2 là một bộ công cụ phát triển tích hợp (IDE Integrated Development Environment) các ứng dụng C/C++ thuộc dạng mã nguồn mở. DevCpp dựa  trên trình biên dịch mã nguồn mở MinGW (Minimalist GNU\* for Windows). MinGW sử dụng GCC (the GNU g++ compiler collection) sử dụng cho cả hệ thống Windows và Linux. Hiện nay Dev-C++ là công cụ phát triển các ứng dụng C/C++ được úng dụng rộng rãi để dạy về lập trình cũng như để phát triển các ứng dụng mã nguồn mở. Mặc dù không có nhiều tính năng cao cấp như IDE thương mại khác (Visual Studio của Microsoft) nhưng DevCpp vẫn là một công cụ rất thích hợp trong môi trường giáo dục ở bậc đại học khi dạy và học các ngôn ngữ C/C++.

DevC/C++ phát triền trên nền Windows do vậy để lập trình giao diện đồ họa như BorlandC++ Builder 6.0 (https://taimienphi.vn/download-c-fundamentals-for-borland-c-builder-19020/60-phien-ban) gặp không ít trở ngại. Để giải quyết khó khăn này một thư viện mở rộng cho phép các lập trình viên thực.

## **1.4. Graphics.h**

Thư viện Graphicss.h hay tên chính xác và đầy đủ của nó là Borland Graphics Interface – còn được biết đến với tên gọi BGI – là một thư viện đồ họa rất phổ biến trên DOS và các máy tính chạy hệ điều hành Windows thời kì đầu như Windows 95, Windows 98. Thư viện này cung cấp cho người dùng hai file graphics.h và graphics.lib để có thể sử dụng đucợ với ngôn ngữ C/C++ cũng như module graph nếu người dùng sử dụng ngôn ngữ Pascal. Bộ thư viện này đi kèm với IDE Borland C++ 3.1 (1992). Nó chứa một số hàm vẽ ảnh và cách tạo ảnh chuyển động, in chữ. Một trong những điểm mạnh của thư viện này là việc khởi tạo cũng như sử dụng rất đơn giản, vì vậy mặc dù ra đời rất lâu nhưng hiện tại vẫn có rất nhiều trường đại học sử dụng cho mục đích giảng dạy.**(**<https://github.com/thinhdoanvu/ComputerGraphics/tree/master/Coding/library>).

## **1.5. Giải thuật vẽ đoạn thẳng thông thường**

Giả sử tọa độ các điểm nguyên sau khi xấp xỉ đối tượng thực lần lượt là (*xi*, *yi*), *i* = 0,1,2,.... Đây là các điểm nguyên sẽ được hiển thị trên màn hình.

Bài toán đặt ra là nếu biết được (*xi*, *yi*) là tọa độ nguyên xác định ở bước thứ i, điểm nguyên tiếp theo (*xi*+1, *yi*+1) sẽ được xác định như thế nào?

| 8 |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  | **?** |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |

Line: (3,2) -> (9,6)

Pixels nào sẽ được bật lên?  
 0 1 2 3 4 5 6 7 8 9 10 11

Nhận xét rằng để đối tượng hiển thị trên lưới nguyên được liền nét, các điểm mà (*xi*+1,*yi+1*) có thể chọn chỉ là một trong tám điểm được đánh số từ 1 đến 8 (điểm đen chính là (*xi*,*yi*). Hay nói cách khác: (*xi+1*,*yi +1*) = (*xi±1*,*yi±1*).

Dáng điệu của đường sẽ cho ta gợi ý khi chọn 1 trong 8 điểm trên. Cách chọn các điểm như thế nào sẽ tùy thuộc vào từng thuật toán trên cơ sở xem xét tới vấn đề tối ưu tốc độ.

Cho phương trình đường thẳng y = mx + b

Với (x0,y0) là tọa độ điểm đầu, (x1, y1) là tọa độ điểm cuối, xác định m và b?

Ta có:

b = y0 – m \* x0

(x1,y1)

dy

(x0,y0)

dx

Cho ví dụ sau để xác định hệ số góc m: (Ax, Ay) = (23, 41), (Bx, By) = (125, 96)

Ta có: :

Cho đường thẳng xuất phát từ điểm có tọa độ x0, y0

x và y tăng dần theo hệ số trong khoảng 0 – 1

Từ kết quả của công thức xác định m, ta nhận thấy.

+ Nếu x tăng nhanh hơn y: m < 1

x= x + 1, tính toán và làm tròn y theo x

+ Ngược lại: m > 1

y = y + 1, tính toán và làm tròn x theo y

Xét đoạn thẳng có hệ số góc 0 < *m* < 1 và *Dx* > 0

Với các đoạn thẳng dạng này, nếu (*xi*, *yi*) là điểm đã xác định được ở bước thứ i (điểm màu đen) thì điểm cần chọn (*xi*+1,*yi*+1) ở bước thứ (i+1) sẽ là một trong hai trường hợp như hình bên:

Như vậy:

## **1.6. Giải thuật sinh đa giác**

Việc biểu diễn đa giác thông qua:

* Tập các đoạn thẳng
* Tập các điểm thuộc đa giác

Các loại đa giác:

* Đa giác lồi: là đa giác có đường thẳng nối bất ký 2 điểm bên trong nào của đa giác đều nằm trọn trong đa giác.
* Đa giác không lồi là đa giác lõm.

Các đường thẳng bao đa giác - cạnh của đa giác. Các điểm giao của cạnh - đỉnh của đa giác.

Thông tin cần thiết để xác định đa giác:

* Số cạnh
* Toạ độ các đỉnh của đa giác

A blue arrow pointing to a blue arrow

Description automatically generated with medium confidence

Hình 1.2 Mô phỏng giải thuật sinh đa giác

**1.7.** **Giải thuật sinh đường tròn Midpoint**

Do tính đối xứng của đường tròn (C) nên ta chỉ cần vẽ cung (C1/8) là cung 1/8 đường tròn, sau đó lấy đối xứng. Cung (C1/8) được mô tả như sau (cung của phần tô màu trong hình vẽ)

Nếu có (x, y) ∈ (C1/8) thì các điểm:(y,x), (x,-y), (y,-x), (-x,-y), (-y,-x), (-y,x), (-x,y) ∈ (C).

A circle with a triangle and a triangle with a triangle in the center

Description automatically generated with medium confidence

Hình 1.3 Giải thuật sinh đường tròn Midpoint

Chọn điểm (0,R) bắt đầu để vẽ. Dựa vào hình trên, nếu (*xi*, *yi*) là điểm nguyên đã tìm được ở bước thứ i, thì điểm ( *xi*+1, *yi*+1) ở bước thứ (i+1) là sự lựa chọn giữa S và P.

Tương tự như thuật toán MidPoint vẽ đoạn thẳng, việc quyết định chọn một trong hai điểm S và P sẽ được thực hiện thông qua việc xét dấu của một hàm nào đó tại điểm MidPoint là điểm nằm giữa chúng.

A diagram of a math problem

Description automatically generated

Hình 1.4 Giải thuật sinh đường tròn Midpoint

Đặt *f*( *x*, *y* ) = *x*2 + *y*2 − *R*2, ta có

Xét

Nếu *pi* < 0 , điểm MidPoint nằm trong đường tròn. Lúc này điểm thực Q gần S hơn nên ta chọn S, tức là *yi*+1 = *yi*

A diagram of a graph

Description automatically generated

Hình 1.4 Giải thuật sinh đường tròn Midpoint

Ngược lại, nếu *pi* ≥ 0, điểm MidPoint nằm ngoài đường tròn. Lúc này điểm thực Q gần P hơn nên ta chọn P, tức là *yi*+1 = *yi* − 1. Mặt khác:

Do

Vậy:

, nếu *pi* < 0 do ta chọn *yi*+1 = *yi*.

, nếu *pi* ≥ 0 do ta chọn *yi*+1 = *yi*−1

Ta tính giá trị *p*0 ứng với điểm ban đầu (x0,*y*0 ) = (0, *R*).

# Chương 2: PHƯƠNG PHÁP NGHIÊN CỨU

## **2.1. Cài đặt Dev-C++ và Graphics.h**

**Cài đặc Dev-C++ (Download DevCpp-4.9.9.2\_Setup)**

**Cài đặt thư viện Graphics.h vào Dev-C++**

***Bước 0****:* Tải thư viện và giải nén **(https://github.com/thinhdoanvu/ComputerGraphics)**

***Bước 1***: copy 2 file 6-ConsoleAppGraphics và ConsoleApp\_cpp\_graph vào đường dẫn sau C:\Program Files\dev-Cpp\Templates.

***Bước 2***: copy 2 file graphics và winbgim vào đường dẫn sau C:\Program Files\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\include.

***Bước 3***: copy file libbgi.a vào đường dẫn sau C:\Program File\Dev-Cpp\MinGW6x86\_64-w64-mingw32\lib.

***Bước 4***: Mở Dev-C++ lên, chọn New Project – Console Graphics Application, đặt tên và lưu project.

***Bước 5***: vào Tools - Compiler Option, chọn TDM-GCC 4.9.2 32 bit Release.

***Bước 6***: tiến hành code thử nghiệm đoạn code (Hình 2.1) vào project đã lưu.

A screenshot of a computer code

Description automatically generated

Hình 2.1 Đoạn code demo cài đặt thư viện graphics.h vào Dev-C++

## **2.2.** **Tổng quan về thiết kế giao diện**

### **2.2.1.** **Sơ lược giao diện**

Giao diện sản phẩm gồm

* Khung để vẽ các đường thẳng, hình chữ nhật, hình tròn, hình đa giác và hiển thị màn hình
* Các nút chức năng gồm Line, Rectangle, Circle, Clear, Thickness+, Thickness-, Save, Polyline, Polygon, Finish
* Các ô màu thể hiện chọn màu để vẽ
* Có góc hiển thị trạng thái gồm Start và End, Mode, Thickness
* Có góc chứa thông tin mssv, tên sinh viên, giáo viên hướng dẫn, lớp

**A screenshot of a computer

Description automatically generated**

Hình 2.2.1 Giao diện chính sản phẩm

### **2.2.2. Xây dựng giao diện**

#### **2.2.2.1** **Khởi tạo cửa sổ đồ họa BGI**

Trong hàm main, sử dụng hàm initwindow(width,height) với width = 800 và height = 600 để khởi tạo cửa số đồ họa có độ rộng 800 và chiều cao 600 trên màn hình do thư viện đồ họa graphics.h cung cấp

A black rectangle with white text

Description automatically generated

Hình 2.2.2.1 Cửa sổ đồ họa BGI

#### **2.2.2.2 Khởi tạo cửa sổ đồ họa BGI**

Hàm initInterface dùng để hiển thị các thực thể (khung, nút, ô màu, góc thông tin) lên cửa sổ đồ họa BGI. Đây là hàm initInterface

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**Hình 2.2.2.2 Toàn bộ code các thực thể cửa sổ đồ họa BGI

## **2.3 Xử lý sự kiện chuột**

Trong hàm handleClick sẽ chia làm ba phần:

Phần thứ nhất, sẽ xử lý giao diện điều khiển gồm:

* Chọn chế độ vẽ
* Xóa hình vẽ và khởi tạo lại
* Thay đổi độ dày nét vẽ
* Thay đổi màu sắc
* Hoàn tất vẽ
* Lưu hình ảnh

A screenshot of a computer code

Description automatically generated

**A computer code with many numbers

Description automatically generated with medium confidence**

Hình 2.3 Xử lý sự kiện chuột

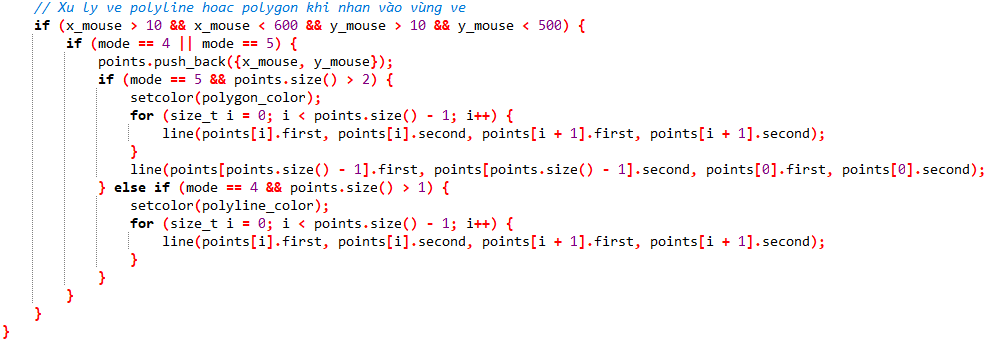
Phần thứ hai,sẽ xử lý màu sắc ngoài màu chính để vẽ ra (màu trắng) thì còn có ba màu khác để vẽ hình là đỏ, xanh dương và xanh lá

A screenshot of a computer code

Description automatically generated

Hình 2.3 Xử lý màu sắc

Phần thứ 3, sẽ xử lý phần vẽ polyline và polygon



Hình 2.3 Xử lý vẽ polyline và polygon

## **2.4. Vẽ hình**

Trong hàm drawShape, sẽ thể hiện 3 chế độ vẽ là đường thẳng, hình chữ nhật và hình tròn. Khi nhấp chuột và 2 điểm bất kỳ thì chương trình sẽ vẽ tạo thành một đường thẳng từ hai điểm đã chọn sau đó dừng lại và tiếp tục cho lần vẽ tiếp theo, cách vẽ hình chữ nhật và hình tròn cũng tương tự như vậy

A white screen with black and red text

Description automatically generated

Hình 2.4 Code vẽ hình

## **2.5. Lưu hình**

Trong hàm savePNG, sẽ lưu lại hình đã vẽ của cửa sổ đồ họa BGI dưới dạng hình ảnh trong file , khi đã lưu hoàn tất thì chương trình sẽ thông báo “Image saved as output\_high\_res.png” là hình đã được lưu

A screenshot of a computer code

Description automatically generated

A screenshot of a computer error

Description automatically generated

Hình 2.5 Code lưu hình ảnh

## **2.6. Hiển thị trạng thái**

Trong hàm displayStatus, sẽ hình hiển thị trạng thái khi vẽ gồm:

* Điểm bắt đầu (Start) và điểm kết thúc (End) chương trình sẽ hiển thị tọa độ (x,y) tương ứng với hai điểm đó
* Chức năng vẽ (Mode), khi nhấp chọn vào một nút chức năng vẽ bất kỳ thì chương trình sẽ hiển thị tên tương ứng với chức năng đó
* Màu sắc (Current Color), khi nhấp chọn vào một màu sắc bất kỳ thì chương trình sẽ hiển thị tên tương ứng với màu sắc đó
* Tăng giảm độ dày (Thickness), khi nhấp chuột vào hai nút chức năng Thickness+ và Thickness- thì chương trình sẽ tăng giảm độ dày nét vẽ tối đa là 1 10

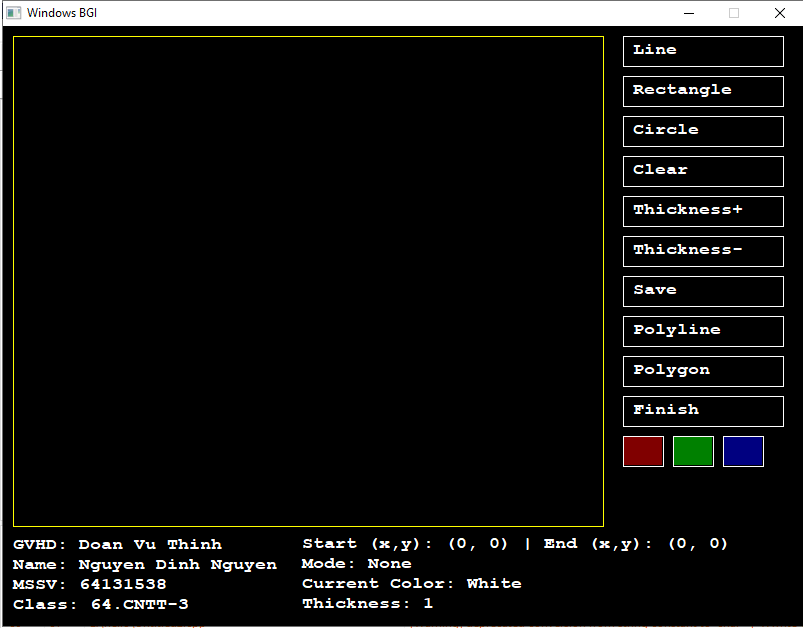
A close-up of a code

Description automatically generated

Hình 2.6 Code hiển thị trạng thái

# Chương 3: KẾT QUẢ

Link youtube demo của giao diện đơn giản MS Pain: <https://youtu.be/p37WfOHLy8I>

****

*Hình 3.1 Giao diện chính của sản phẩm*

**

Hình 3.2 Giao diện vẽ đường thẳng

*A screenshot of a computer

Description automatically generated*

Hình 3.3 Giao diện vẽ hình chữ nhật

A screen shot of a computer

Description automatically generated

Hình 3.4 Giao diện vẽ hình tròn

**A screenshot of a computer

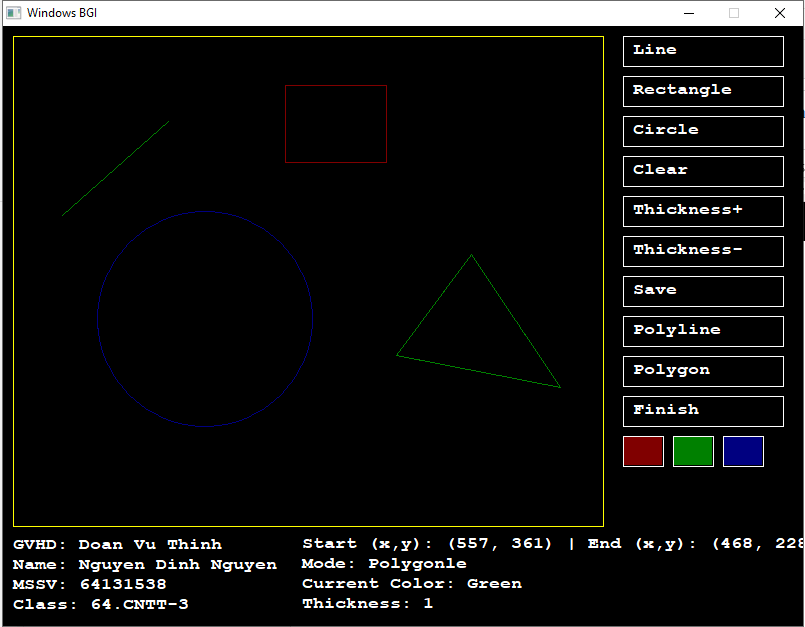
Description automatically generated**

Hình 3.5 Giao diện vẽ đa đoạn thẳng

**A screenshot of a computer

Description automatically generated**

Hình 3.6 Giao diện vẽ đa giác



Hình 3.7 Giao diện vẽ màu

A screenshot of a computer

Description automatically generated

Hình 3.8 Giao diện tăng giảm độ dày

A screenshot of a computer

Description automatically generated

Hình 3.9 Giao diện lưu ảnh

# Chương 4: KẾT LUẬN

## **4.1 Ưu điểm**

* Nắm vững được kiến thức Giải thuật vẽ đoạn thẳng thông thường, giải thuật sinh đa giác, giải thuật sinh đường tròn Midpoint
* Làm được giao diện phù hợp, dễ nhìn
* Nắm kỹ năng vẽ đồ họa của môn Kỹ thuật đồ họa

## **4.2 Nhược điểm**

* Chưa mô tả được chức năng lưu hình (hình lưu còn bị mờ)
* Xử lý code rườm rà chưa được hoàn hảo