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 GRAPHICS.H MÔ PHỎNG GIẢI THUẬT XÉN TỈA ĐOẠN THẲNG BẰNG GIẢI THUẬT COHEN-SUTHERLAND**

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

**Sinh viên thực hiện : Nguyễn Khánh Như**

**Mã số sinh viên : 64131703**

**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 GRAPHICS.H MÔ PHỎNG GIẢI THUẬT XÉN TỈA ĐOẠN THẲNG BẰNG GIẢI THUẬT COHEN-SUTHERLAND**

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

**Sinh viên thực hiện : Nguyễn Khánh Như**

**Mã số sinh viên : 64131703**

**Lớp : 64.CNTT-3**

Khánh Hòa – 2024

MỤC LỤC

[LỜI CẢM ƠN 4](#_Toc185876785)

[TÓM TẮT 5](#_Toc185876786)

[Chương 1: Giới thiệu 6](#_Toc185876787)

[1. Thuật toán xén tỉa đoạn thẳng: 6](#_Toc185876788)

[1.1. Giải thuật xén tỉa (clipping): 6](#_Toc185876789)

[1.2. Thuật toán Cohen-Sutherland: 7](#_Toc185876790)

[1.3. Thư viện graphics.h: 10](#_Toc185876791)

[1.4. Dev C++: 10](#_Toc185876792)

[Chương 2: Phương pháp nghiên cứu 11](#_Toc185876793)

[2.1. Cài đặt Dev-C++ và thư viện graphics.h: 11](#_Toc185876794)

[2.2. Xây dựng giao diện: 12](#_Toc185876795)

[2.2.1. Xây dựng giao diện chính: 12](#_Toc185876796)

[2.3. Cài đặt thuật toán Cohen-Sutherland: 14](#_Toc185876797)

[2.3.1. Khai báo thư viện: 14](#_Toc185876798)

[2.3.2. Khai báo toàn cục: 14](#_Toc185876799)

[2.3.3. Nhập dữ liệu đầu vào: 14](#_Toc185876800)

[2.3.4. Xử lí chuột: 19](#_Toc185876801)

[2.3.5. Thuật toán xén tỉa: 21](#_Toc185876802)

[Chương 3: Kết quả thực hiện 24](#_Toc185876803)

[3.1. Giao diện chính: 24](#_Toc185876804)

[3.2. Giao diện đọc dữ liệu từ file: 24](#_Toc185876805)

[3.3. Giao diện nhập từ bàn phím: 24](#_Toc185876806)

[3.4. Giao diện dùng chuột: 25](#_Toc185876807)

[3.3.1. Về các nút: 25](#_Toc185876808)

[3.3.2. Xén tỉa đoạn thẳng: 28](#_Toc185876809)

[Chương 4: Kết luận 30](#_Toc185876810)

[TÀI LIỆU THAM KHẢO 31](#_Toc185876811)

MỤC LỤC HÌNH ẢNH

Chương 1.

[Hình 1.1. Ví dụ về xén điểm 6](file:///C:\TTCS\TTCSChinhThuc\BaoCaoTTCS_NguyenKhanhNhu_64131703.docx#_Toc185871819)

[Hình 1.2. Ví dụ về xén đường thẳng. 7](file:///C:\TTCS\TTCSChinhThuc\BaoCaoTTCS_NguyenKhanhNhu_64131703.docx#_Toc185871820)

[Hình 1.4. Trước khi cắt xén. 7](file:///C:\TTCS\TTCSChinhThuc\BaoCaoTTCS_NguyenKhanhNhu_64131703.docx#_Toc185871821)

[Hình 1.3. Sau khi cắt xén. 7](file:///C:\TTCS\TTCSChinhThuc\BaoCaoTTCS_NguyenKhanhNhu_64131703.docx#_Toc185871822)

[Hình 1.5. Chia mặt phẳng hình chữ nhật thành phần và gắn mã vùng. 8](file:///C:\TTCS\TTCSChinhThuc\BaoCaoTTCS_NguyenKhanhNhu_64131703.docx#_Toc185871823)

[Hình 1.6. Vẽ đoạn thẳng vào mặt phẳng hình chữ nhật. 9](file:///C:\TTCS\TTCSChinhThuc\BaoCaoTTCS_NguyenKhanhNhu_64131703.docx#_Toc185871824)

Chương 3.

[Hình 3.1. Giao diện chính. 24](#_Toc185871825)

[Hình 3.2. Kết quả đọc dữ liệu từ file. 24](#_Toc185871826)

[Hình 3.3. Kết quả nhập dữ liệu từ bàn phím. 25](#_Toc185871827)

[Hình 3.4. Giới thiệu các nút. 25](#_Toc185871828)

[Hình 3.5. Cửa sổ console sau khi vẽ hình. 26](#_Toc185871829)

[Hình 3.6. Kết quả sau khi vẽ hình. 26](#_Toc185871830)

[Hình 3.7. Cửa sổ console sau khi vẽ đường thẳng. 27](#_Toc185871831)

[Hình 3.8. Kết quả sau khi vẽ đường thẳng. 27](#_Toc185871832)

[Hình 3.9. Trường hợp vẽ đường thẳng nằm ngoài hình chữ nhật. 28](#_Toc185871833)

[Hình 3.10. Trường hợp vẽ đường thẳng nằm trong hình chữ nhật. 29](#_Toc185871834)

[Hình 3.11. Trường hợp vẽ đường thẳng cắt hình chữ nhật. 29](#_Toc185871835)

# 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 trình học tập và nghiên cứu đề tài này.

Qua đây, em xin chân thành cảm ơn thầy Đoàn Vũ Thịnh, người đã trực tiếp quan tâm và hướng dẫn chúng 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

Trong thời đại số hóa, công nghệ thông tin đóng vai trò then chốt trong việc phát triển và ứng dụng các giải pháp kỹ thuật số, góp phần nâng cao hiệu suất làm việc và chất lượng cuộc sống. Một lĩnh vực quan trọng của công nghệ thông tin là đồ họa máy tính, đặc biệt là các thuật toán xử lý và biểu diễn hình ảnh.

Đồ họa máy tính không chỉ mang đến những bộ phim hoạt hình sống động hay các trò chơi video hấp dẫn, mà còn có ứng dụng rộng rãi trong nhiều lĩnh vực như y học, thiết kế kiến trúc, mô phỏng khoa học kỹ thuật, phân tích dữ liệu. Các thuật toán đồ họa cho phép chúng ta trực quan hóa thông tin phức tạp, biến chúng thành những hình ảnh dễ hiểu và dễ tương tác hơn.

Báo cáo này tập trung vào việc nghiên cứu và mô phỏng thuật toán xén tỉa đoạn thẳng bằng thuật toán Cohen-Sutherland, sử dụng thư viện đồ họa graphics.h nhằm mục tiêu giúp người học tiếp cận sâu hơn với thuật toán này.

Bài báo cáo gồm các nội dung chính như sau: giới thiệu về thuật toán xén tỉa, môi trường DevC++ và thư viện graphics.h; các bước triển khai mô phỏng thuật toán; thiết kế giao diện; hiển thị kết quả trên màn hình.

# Chương 1: Giới thiệu

Đồ họa máy tính (Computer Graphics) là ngành nghiên cứu và áp dụng các kỹ thuật sử dụng máy tính để tạo ra hình ảnh, mô phỏng hình ảnh, xử lý hình ảnh. Trong lĩnh vực này, việc xén tỉa đoạn thẳng là một khâu quan trọng, giải thuật Cohen-Sutherland được coi là một trong những phương pháp hiệu quả nhất để thực hiện điều đó trong khung hình chữ nhật. Giải thuật này giúp tối ưu hóa quá trình xử lý hình ảnh, làm tăng hiệu suất và chất lượng hình ảnh trong các ứng dụng đồ họa máy tính.

## **1. Thuật toán xén tỉa đoạn thẳng:**

### **1.1. Giải thuật xén tỉa (clipping):**

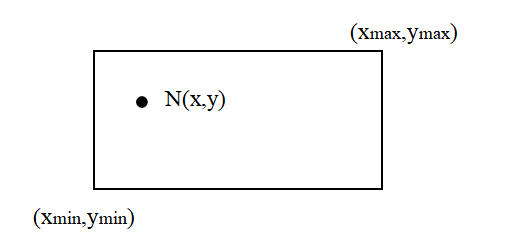
Xén tỉa là tiến trình tự động xác định các điểm của 1 đối tượng nằm trong hay ngoài cửa sổ hiển thị. Nằm trong thì hiển thị, nằm ngoài thì loại bỏ.

Vùng được dùng xén hình gọi là cửa sổ xén (clip window).

**Xén điểm:**

Ví dụ: Cho hình chữ nhật có tọa độ (xmin,ymin) và (xmax,ymax).

Điểm N có tọa độ (x,y) được hiển thị khi thỏa mãn:

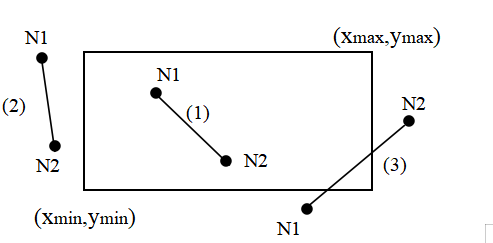
- xmin <= x <= xmax

- ymin <= y <= ymin

Hình 1.1. Ví dụ về xén điểm

**Xén đoạn thẳng:**

Đoạn thẳng được tạo bởi 2 điểm N1(x1,y1) và N2(x2,y2) trong hình chữ nhật:

1. Đoạn thẳng nằm trong hình chữ nhật.

Hình 1.2. Ví dụ về xén đường thẳng.

1. Đoạn thẳng nằm ngoài hình chữ nhật.

Giải thuật kiểm tra và chấp nhận các đoạn thẳng nằm trong hình chữ nhật và loại bỏ đoạn thẳng nằm ngoài dựa trên 2 điểm đầu cuối:

- Các đoạn thẳng có cả 2 điểm nằm hoàn toàn trong cửa sổ thì cả đoạn thẳng nằm trong cửa sổ.

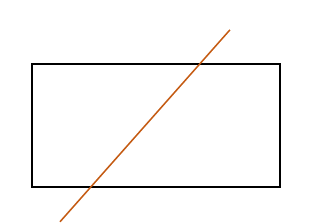
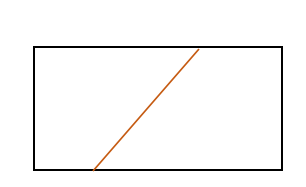
- Các đoạn thẳng có cả 2 điểm nằm cùng 1 phía cửa cửa sổ thì cả đoạn thẳng luôn nằm ngoài cửa sổ.

- Với các đoạn thẳng cắt cửa sổ thì ta cần tìm giao điểm của đoạn thẳng với biên của cửa sổ để tìm đoạn thẳng cần xén.

### **1.2. Thuật toán Cohen-Sutherland:**

Thuật toán Cohen-Sutherland là một trong những phương pháp xén tỉa đoạn thẳng hiệu quả, được phát triển bởi Danny Cohen và Ivan Sutherland vào năm 1967.

Thuật toán loại bỏ các đọạn không cần xén bằng cách kiểm tra tọa độ 2 đầu mút của đoạn thẳng sau khi được mã hóa để xét vị trí tương đối của đoạn thẳng với cửa sổ (hình chữ nhật).



Hình 1.4. Trước khi cắt xén.

Hình 1.3. Sau khi cắt xén.

**Ý tưởng:**

Có 3 bước:

- Chia mặt phẳng chứa cửa sổ hiển thị (hình chữ nhật) thành 9 phần và gắn mã vùng.

- Xét các trường hợp.

- Tính giao điểm của đoạn thẳng với HCN.

**Bước 1: Chia mặt phẳng chứa cửa sổ hiển thị (hình chữ nhật) thành 9 phần và gắn mã vùng.**

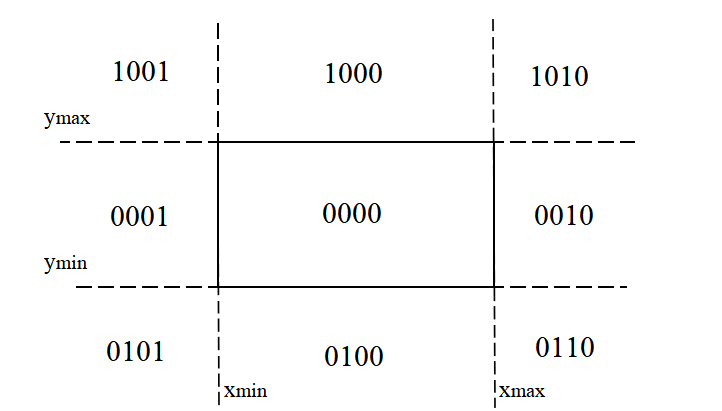
Kéo dài các biên của hình chữ nhật.

Giả sử ta mã hóa điểm N(x,y):

- Nếu x<xmin, bit1 là 1, những vị trí còn lại là 0.

- Nếu x>xmax, bit2 là 1, những vị trí còn lại là 0.

- Nếu y<ymin, bit3 là 1, những vị trí còn lại là 0.

- Nếu y>ymax, bit4 là 1, những vị trí còn lại là 0.

Hình 1.5. Chia mặt phẳng hình chữ nhật thành phần và gắn mã vùng.

**Bước 2: Xét các trường hợp.**

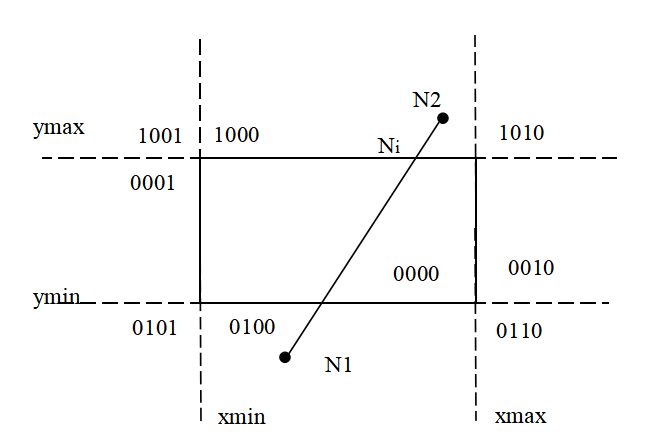
Cho đoạn thẳng N1(x1,y1) N2(x2,y2).

- Nếu N1 = N2 = 0000 → Đoạn thẳng nằm trong hình chữ nhật.

- Nếu N1 AND N2 != 0000 → Đoạn thẳng nằm ngoài hình chữ nhật.

- Nếu không thuộc 2 trường hợp trên → Đoạn thẳng cắt hình chữ nhật.

**Bước 3: Tính giao điểm của đoạn thẳng với hình chữ nhật.**



Hình 1.6. Vẽ đoạn thẳng vào mặt phẳng hình chữ nhật.

Gọi tọa độ điểm cắt là Ni (xi,yi).

Tọa độ cắt:

xi = xmin hoặc xmax

yi = y1 + m(xi - x1)

Hoặc:

xi = x1 + (yi - y1)/ m

yi = ymin hoặc ymax

81bf98ec-a2a7-475e-b5e7-8d18c0e4bb45Với:

### **1.3. Thư viện graphics.h:**

Graphics.h tên đầy đủ là Borland Graphics Interface (BGI) là một thư viện đồ họa chủ yếu sử dụng trong môi trường MS-DOS. Thư viện này hỗ trợ việc tạo ra các đối tượng đồ họa 2D như đường thẳng, hình tròn, hình chữ nhật, đa giác và các hiệu ứng trực quan cơ bản.

Thư viện này cung cấp hai file là graphics.h và graphics.lib, cho phép người dùng sử dụng trong C/C++ cũng như module graph khi lập trình bằng Pascal. Bộ thư viện này được đi kèm với IDE Borland C++ 3.1 (1992).

### **1.4. Dev C++:**

DevC++ là một IDE (Môi trường Phát triển Tích hợp) miễn phí dành cho lập trình C/C++, được phát triển bởi Bloodshed Software. DevC++ cung cấp một môi trường phát triển dễ sử dụng với nhiều tính năng hỗ trợ lập trình viên. DevC++ sử dụng MinGW (Minimalist GNU for Windows) làm trình biên dịch mặc định, giúp biên dịch các chương trình C/C++ trên hệ điều hành Windows.

DevC++ có giao diện người dùng dễ hiểu và trực quan, thích hợp cho cả người mới bắt đầu và các lập trình viên có kinh nghiệm. Là công cụ thích hợp cho môi trường giáo dục ở đại học.

DevC++ hỗ trợ sử dụng thư viện đồ họa như graphics.h trong các dự án, giúp vẽ đồ họa và mô phỏng các thuật toán đồ họa, rất hữu ích trong các bài học về đồ họa máy tính.

# Chương 2: Phương pháp nghiên cứu

## **2.1. Cài đặt Dev-C++ và thư viện graphics.h:**

Vào đường dẫn sau: <https://sourceforge.net/projects/orwelldevcpp/> để cài đặt phần mềm Dev-C++. Sau khi tải, mở file và tiến hành cài đặt.

Tiến hành cài đặt thư viện graphics.h bằng cách truy cập vào link download: https://drive.google.com/file/d/1qMnzjQdm2yVzmUvzxjTHV74BAj956HKd/view

**Bước 1:** Copy 2 file "graphics.h" và "winbgim.h" vào thư mục "(Path)\MinGW64\x86\_64-w64-mingw32\include".

**Bước 2:** Copy file "libbgi.a" vào thư mục "(Path)\MinGW64\x86\_64-w64-mingw32\lib".

**Bước 3:** Copy 2 file "6-ConsoleAppGraphics.template" và "ConsoleApp\_cpp\_graph.txt" vào thư mục "(Path)\Templates".

**Hướng dẫn sử dụng:** Sau khi cài đặt theo các bước như hướng dẫn ở trên, tiến hành mở Dev-C++.

**Bước 1:** Chọn trình biên dịch là TDM-GCC 32 bit ứng với phiên bản Dev-C++ đã cài đặt. Ví dụ: Nếu Dev C++ 5.6.3 64 bit thì chọn trình biên dịch là "TDM-GCC 4.8.1 32 bit Release".

**Bước 2:** Tạo một Project mới bằng cách chọn File -> New -> Project...

+ Chọn Console Graphics Application.

+ Tick vào ô C++ Project.

+ Gõ tên Project vào ô Name.

+ Click Ok và chọn vị trí lưu.

**Bước 3:** Tiến hành lập trình đồ họa và chạy thử.

## **2.2. Xây dựng giao diện:**

### **2.2.1. Xây dựng giao diện chính:**

Khởi tạo cửa sổ đồ họa BGI: Tại hàm main(), ta sử dụng initwindow(900, 500) để khởi tạo cửa sổ đồ họa với chiều rộng 900 và chiều dài 500.

Giao diện dùng để hiển thị thông tin: tên giảng viên hướng dẫn, tên sinh viên, mã số sinh viên, lớp. Có các nút:

* New: nút tạo mới.
* LoadFile: tải dữ liệu từ file văn bản.
* Keyboard: nhập dữ liệu từ bàn phím.
* DEMO: có 2 nút, nút có kí hiệu dấu / dùng để vẽ đường thẳng, nút có kí hiệu hình vuông dùng để vẽ hình. Đây là chức năng nhập dữ liệu bằng chuột.

**Code**:

void GiaoDienChinh(){

setcolor(WHITE);

// boder

setlinestyle(0, 1, 3);

rectangle(30, 10, 550, 400);

// nut tao moi

setcolor(GREEN);

settextstyle(2, 0, 7);

rectangle(580, 10, 680, 50);

outtextxy(610, 20, "New");

// nut loadfile

setcolor(WHITE);

settextstyle(2, 0, 7);

rectangle(580, 70, 680, 110);

outtextxy(590, 80, "Load File");

// nut Keyboard

setcolor(WHITE);

settextstyle(2, 0, 7);

rectangle(580, 130, 680, 170);

outtextxy(590, 140, "Keyboard");

// Mouse

setcolor(RED);

settextstyle(2, 0, 7);

rectangle(580, 190, 710, 290);

outtextxy(615, 200, "Mouse");

//nut ve duong thang

setcolor(WHITE);

settextstyle(2, 0, 7);

rectangle(590, 230, 640, 280);

setcolor(YELLOW);

line(600, 270, 630, 240);

//nut ve hinh

setcolor(WHITE);

settextstyle(2, 0, 7);

rectangle(650, 230, 700, 280);

setcolor(YELLOW);

rectangle(665, 245, 685, 265);

// Thông tin

setcolor(WHITE);

outtextxy(580, 300, "GVHD: Doan Vu Thinh");

outtextxy(580, 340, "Ten sinh vien: Nguyen Khanh Nhu");

outtextxy(580, 380, "MSSV: 64131703");

outtextxy(580, 420, "Lop: 64.CNTT-3");

}

## **2.3. Cài đặt thuật toán Cohen-Sutherland:**

### **2.3.1. Khai báo thư viện:**

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <stdlib.h>

#include <string.h>

### **2.3.2. Khai báo toàn cục:**

#define INSIDE 0 // 0000

#define LEFT 1 // 0001

#define RIGHT 2 // 0010

#define BOTTOM 4 // 0100

#define TOP 8 // 1000

// Kich thuoc khung xen

int xmin, ymin, xmax, ymax;

### **2.3.3. Nhập dữ liệu đầu vào:**

**Nhập dữ liệu đầu vào bằng bàn phím (nút Keyboard):**

***Theo dõi và xử lí nhấp chuột:***

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x\_mouse, y\_mouse);

* x\_mouse, y\_mouse: Lưu tọa độ nhấp chuột.

**Code:**

void keyboard() {

int x\_mouse, y\_mouse;

int x1, y1, x2, y2; // Toa do 2 diem

while (1) {

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x\_mouse, y\_mouse);

// Nhap du lieu tu ban phim

printf("NHAP PHAI THOA MAN DIEU KIEN \n 80<X<550 && 10<Y<370 \n");

printf("Nhap toa do hinh chu nhat 'xmin ymin xmax ymax': ");

scanf("%d %d %d %d", &xmin, &ymin, &xmax, &ymax);

setcolor(GREEN);

rectangle(xmin, ymin, xmax, ymax);

printf("Nhap toa do diem 1 (x1 y1): ");

scanf("%d %d", &x1, &y1);

printf("Nhap toa do diem 2 (x2 y2): ");

scanf("%d %d", &x2, &y2);

// Ve duong thang da nhap

setcolor(RED);

line(x1, y1, x2, y2);

outtextxy(x1, y1, "P1");

outtextxy(x2, y2, "P2");

// Thuc hien thuat toan Cohen-Sutherland

printf("Dang thuc hien thuat toan Cohen-Sutherland...\n");

int code1 = computeCode(x1, y1);

int code2 = computeCode(x2, y2);

if ((code1 == 0) && (code2 == 0)) {

// Duong thang nam trong hinh chu nhat

printf("Duong thang nam trong hinh chu nhat.\n");

setcolor(RED);

line(x1, y1, x2, y2);

} else if (code1 & code2) {

// Duong thang nam ngoai cung 1 phia

printf("Duong thang nam ngoai hinh chu nhat.\n");

setcolor(WHITE);

line(x1, y1, x2, y2);

} else {

// Duong thang cat hinh chu nhat

int cx1 = x1, cy1 = y1, cx2 = x2, cy2 = y2;

if (cohenSutherlandClip(&cx1, &cy1, &cx2, &cy2)) {

printf("Duong thang cat hinh chu nhat.\n");

printf("Toa do giao diem: N1(%d, %d), N2(%d, %d)\n", cx1, cy1, cx2, cy2);

setcolor(BLUE);

line(cx1, cy1, cx2, cy2);

}

}

break;

}

delay(100);

}

}

**Nhập dữ liệu đầu vào bằng đọc file văn bản (nút Load File):**

**Code:**

void DocFile() {

FILE \*file = fopen("ttcs12.txt", "r");

if (!file) {

printf("err\n");

return;

}

// Doc toa do xmin, ymin, xmax, ymax

fscanf(file, "%d %d %d %d", &xmin, &ymin, &xmax, &ymax);

setcolor(GREEN);

rectangle(xmin, ymin, xmax, ymax);

// Doc toa do cac diem

int x1, y1, x2, y2;

fscanf(file, "%d %d", &x1, &y1);

fscanf(file, "%d %d", &x2, &y2);

// Ve doan thang ban dau

setcolor(RED);

line(x1, y1, x2, y2);

outtextxy(x1, y1, "P1");

outtextxy(x2, y2, "P2");

// Thuc hien thuat toan Cohen-Sutherland

printf("Dang thuc hien thuat toan Cohen-Sutherland...\n");

if (cohenSutherlandClip(&x1, &y1, &x2, &y2)) {

setcolor(BLUE);

line(x1, y1, x2, y2);

printf("Toa do giao diem: N1(%d, %d) N2(%d, %d)\n", x1, y1, x2, y2);

} else {

printf("Doan thang nam ngoai hinh chu nhat\n");

}

fclose(file);

}

**Nhập dữ liệu đầu vào bằng chuột**

**Code:**

// Kiem tra nut ve duong thang

else if (x\_mouse > 590 && x\_mouse < 640 && y\_mouse > 230 && y\_mouse < 280) {

printf("Ve duong thang...\n");

int x1, y1, x2, y2;

// Nhan chuot lan 1 lay diem dau

while (!ismouseclick(WM\_LBUTTONDOWN)) delay(100);

getmouseclick(WM\_LBUTTONDOWN, x1, y1);

// Nhan chuot lan 2 lay diem cuoi

while (!ismouseclick(WM\_LBUTTONDOWN)) delay(100);

getmouseclick(WM\_LBUTTONDOWN, x2, y2);

printf("Toa do duong thang: P1(%d, %d) P2(%d, %d)\n", x1, y1, x2, y2);

// Kien tra vi tri duong thang so voi hinh chu nhat

int code1 = computeCode(x1, y1);

int code2 = computeCode(x2, y2);

setcolor(WHITE);

line(x1, y1, x2, y2);

if ((code1 == 0) && (code2 == 0)) {

// Truong hop nam trong hinh chu nhat

setcolor(RED);

line(x1, y1, x2, y2);

printf("Duong thang nam trong hinh chu nhat.\n");

} else if (code1 & code2) {

// Truong hop nam ngoai cung phia

printf("Duong thang nam ngoai hinh chu nhat.\n");

} else {

// Truong hop cat hinh chu nhat

int cx1 = x1, cy1 = y1, cx2 = x2, cy2 = y2;

if (cohenSutherlandClip(&cx1, &cy1, &cx2, &cy2)) {

setcolor(RED);

// Ve phan giao cat voi hinh chu nhat

line(cx1, cy1, cx2, cy2);

printf("Toa do duong thang cat hinh chu nhat: N1(%d, %d), N2(%d, %d)\n", cx1, cy1, cx2, cy2);

}

}

}

// Kiem tra nut ve hinh

else if (x\_mouse > 650 && x\_mouse < 700 && y\_mouse > 230 && y\_mouse < 280) {

printf("Ve hinh...\n");

int x1, y1, x2, y2;

// Nhan chuot lay goc dau

while (!ismouseclick(WM\_LBUTTONDOWN)) delay(100);

getmouseclick(WM\_LBUTTONDOWN, x1, y1);

// Nhan chuot lay goc doi dien

while (!ismouseclick(WM\_LBUTTONDOWN)) delay(100);

getmouseclick(WM\_LBUTTONDOWN, x2, y2);

setcolor(GREEN);

rectangle(x1, y1, x2, y2);

// Luu lai toa do

xmin = (x1 < x2) ? x1 : x2;

ymin = (y1 < y2) ? y1 : y2;

xmax = (x1 > x2) ? x1 : x2;

ymax = (y1 > y2) ? y1 : y2;

printf("Toa do hinh: xmin=%d, ymin=%d, xmax=%d, ymax=%d\n", xmin, ymin, xmax, ymax);

}

### **2.3.4. Xử lí chuột:**

Hàm clickmouse() được sử dụng để quản lý và xử lý các sự kiện chuột. Đóng vai trò trung tâm trong việc nhận diện và phản hồi khi người dùng nhấp chuột vào các nút trên giao diện. Dưới đây là đoạn code xử lý sự kiện chuột khi nhấp vào các nút New, Load File, Keyboard.

x\_mouse và y\_mouse: Lưu tọa độ của vị trí chuột khi người dùng nhấp chuột.

Kiểm tra xem có nhấn chuột trái không: ismouseclick(WM\_LBUTTONDOWN).

Nếu có, hàm getmouseclick(WM\_LBUTTONDOWN, x\_mouse, y\_mouse) sẽ lấy tọa độ chuột tại thời điểm nhấn và lưu vào x\_mouse và y\_mouse.

**Code:**

int x\_mouse;

int y\_mouse;

while (1) {

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x\_mouse, y\_mouse);

// Kiem tra nut New

if (x\_mouse > 580 && x\_mouse < 680 && y\_mouse > 10 && y\_mouse < 50) {

printf("Resetting....\n");

New();

}

// Kiem tra nut Load File

else if (x\_mouse > 580 && x\_mouse < 680 && y\_mouse > 70 && y\_mouse < 110) {

printf("Doc file...\n");

DocFile();

}

// Kiem tra nut Keyboard

else if (x\_mouse > 580 && x\_mouse < 680 && y\_mouse > 130 && y\_mouse < 170) {

setcolor(RED);

keyboard();

}

//……nhap du lieu dau vao bang chuot da trinh bay o muc 2.2.3.

}

delay(100);

}

}

### **2.3.5. Thuật toán xén tỉa:**

**Hàm computeCode:** Tính mã vùng cho một điểm (x, y) dựa trên vị trí của điểm so với hình chữ nhật.

int computeCode(int x, int y) {

int code = INSIDE;

if (x < xmin)

code |= LEFT;

else if (x > xmax)

code |= RIGHT;

if (y < ymin)

code |= BOTTOM;

else if (y > ymax)

code |= TOP;

return code;

}

Khởi tạo code với giá trị INSIDE (0000), biểu thị điểm nằm trong hình chữ nhật. Dựa trên vị trí của điểm:

* Nếu điểm nằm trái của hình chữ nhật (x < xmin): Bật bit LEFT (0001).
* Nếu điểm nằm phải của hình chữ nhật (x > xmax): Bật bit RIGHT (0010).
* Nếu điểm nằm dưới hình chữ nhật (y < ymin): Bật bit BOTTOM (0100).
* Nếu điểm nằm trên hình chữ nhật (y > ymax): Bật bit TOP (1000).

**Hàm cohenSutherland**: Hàm chính thực hiện thuật toán xén tỉa đoạn thẳng.

// Cohen-Sutherland, tim toa do giao diem

bool cohenSutherlandClip(int \*x1, int \*y1, int \*x2, int \*y2) {

int code1 = computeCode(\*x1, \*y1);

int code2 = computeCode(\*x2, \*y2);

bool accept = false;

while (true) {

if ((code1 == 0) && (code2 == 0)) {

// 2 diem deu nam trong hinh chu nhat

accept = true;

break;

} else if (code1 & code2) {

// 2 diem nam ngoai cung 1 phia

break;

} else {

int codeOut;

int x, y;

// Chon ma vung ngoai

codeOut = code1 ? code1 : code2;

// Tính giao điểm dựa trên mã vùng (codeOut), xác định cạnh bị cắt và tính giao điểm:

if (codeOut & TOP) {

x = \*x1 + (\*x2 - \*x1) \* (ymax - \*y1) / (\*y2 - \*y1);

y = ymax;

} else if (codeOut & BOTTOM) {

x = \*x1 + (\*x2 - \*x1) \* (ymin - \*y1) / (\*y2 - \*y1);

y = ymin;

} else if (codeOut & RIGHT) {

y = \*y1 + (\*y2 - \*y1) \* (xmax - \*x1) / (\*x2 - \*x1);

x = xmax;

} else if (codeOut & LEFT) {

y = \*y1 + (\*y2 - \*y1) \* (xmin - \*x1) / (\*x2 - \*x1);

x = xmin;

}

//Cap nhat diem ngoai voi giao diem

if (codeOut == code1) {

\*x1 = x;

\*y1 = y;

code1 = computeCode(\*x1, \*y1);

} else {

\*x2 = x;

\*y2 = y;

code2 = computeCode(\*x2, \*y2);

}

}

}

return accept;

}

***Cập nhật tọa độ điểm ngoài:***

if (codeOut == code1) {

\*x1 = x;

\*y1 = y;

code1 = computeCode(\*x1, \*y1);

} else {

\*x2 = x;

\*y2 = y;

code2 = computeCode(\*x2, \*y2);

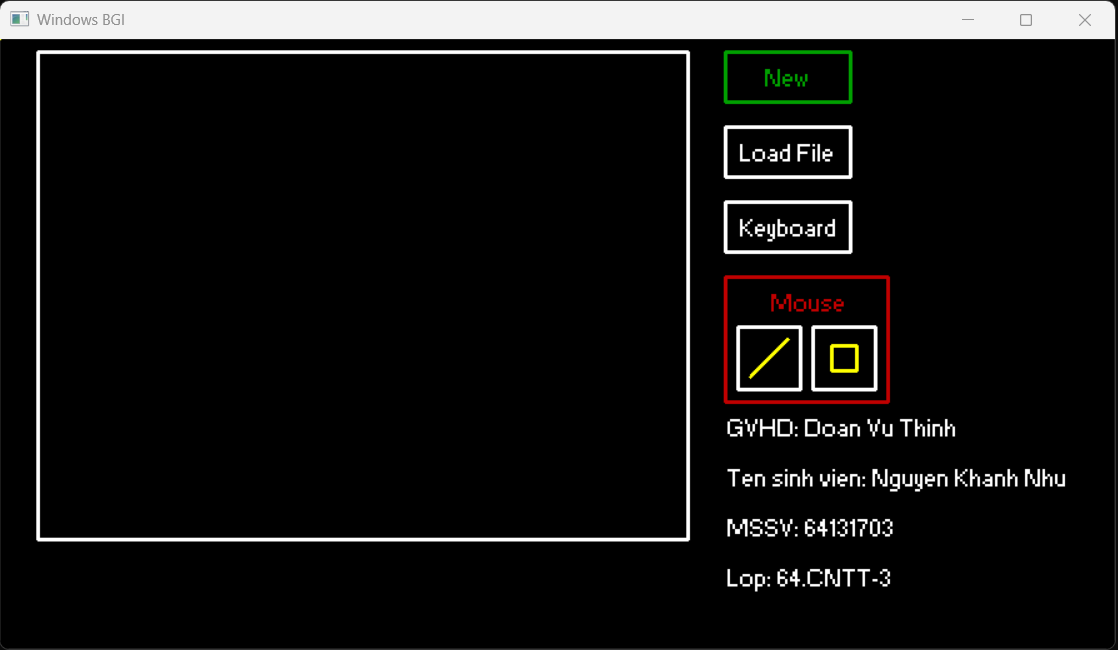
}

Nếu codeOut thuộc điểm code1:

Cập nhật tọa độ của điểm (\*x1, \*y1) thành tọa độ giao điểm, tính lại mã vùng cho điểm đó. Tương tự với code2.

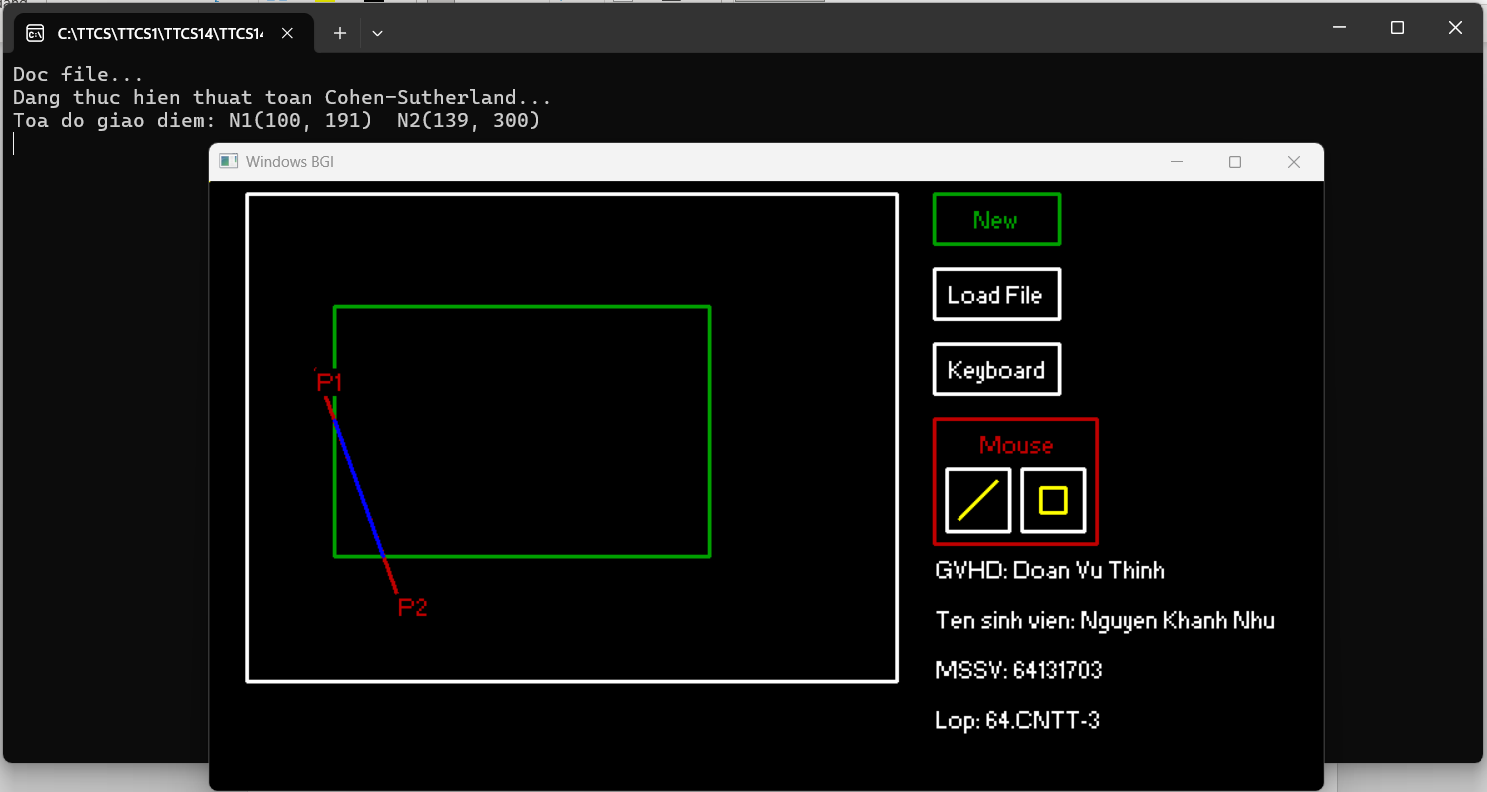
# Chương 3: Kết quả thực hiện

## **3.1. Giao diện chính:**

****

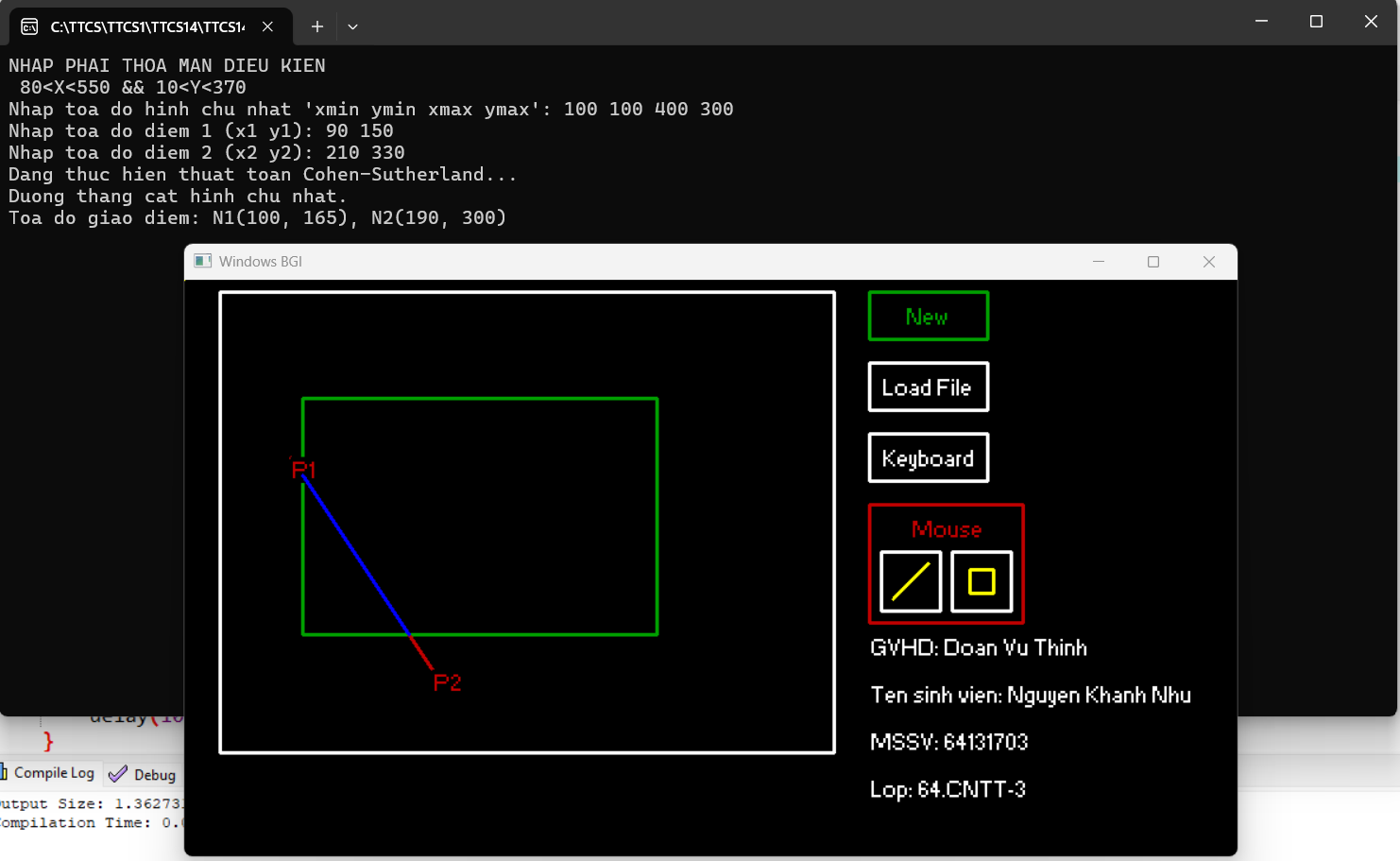
Hình 3.1. Giao diện chính.

## **3.2. Giao diện đọc dữ liệu từ file:**



Hình 3.2. Kết quả đọc dữ liệu từ file.

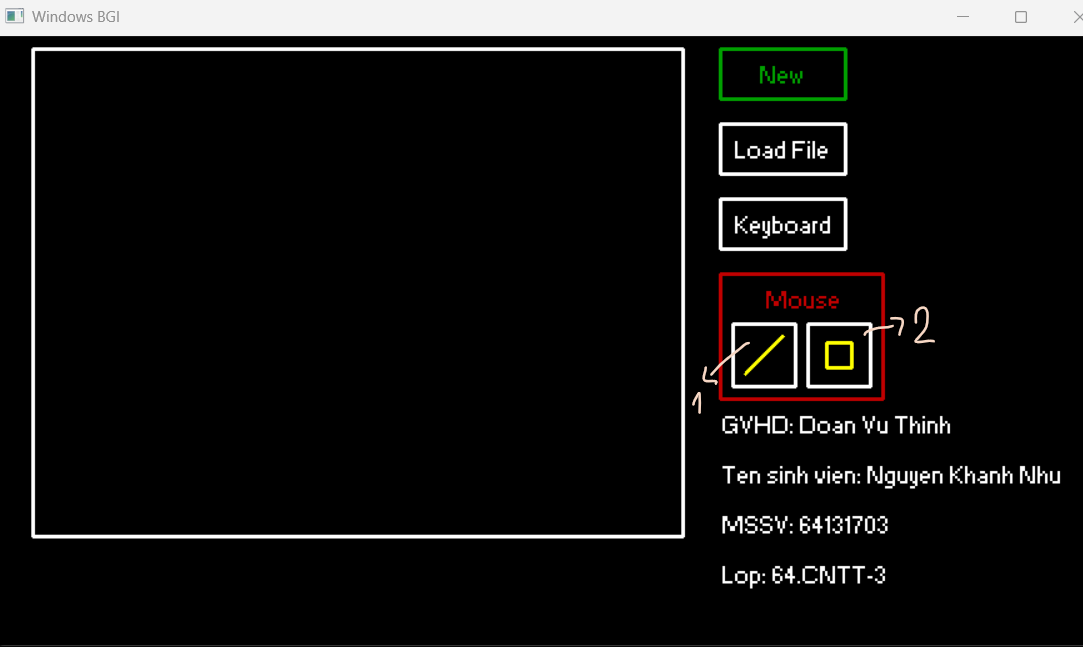
**3.3. Giao diện nhập từ bàn phím:**



Hình 3.3. Kết quả nhập dữ liệu từ bàn phím.

## **3.4. Giao diện dùng chuột:**

### **3.3.1. Về các nút:**

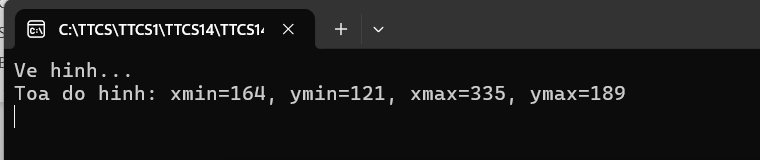


Hình 3.4. Giới thiệu các nút.

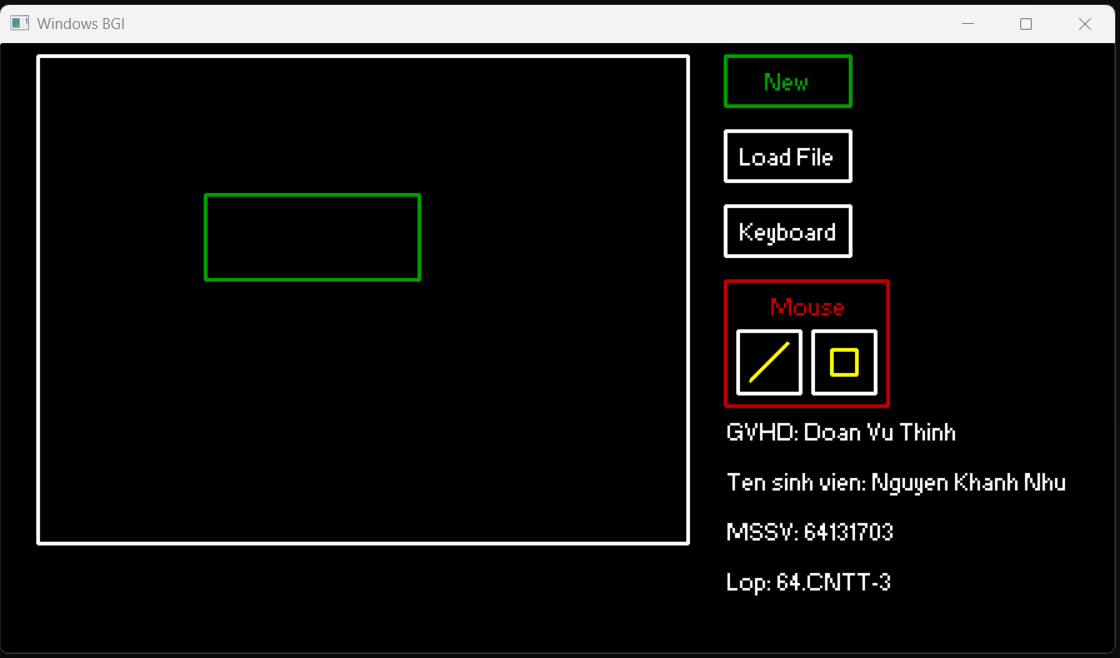
1. Nút dùng để vẽ đường thẳng.
2. Nút dùng để vẽ hình chữ nhật.

**Nút vẽ hình:**

Đầu tiên click vào nút sau đó đưa chuột vào khung, nhấn chuột lần 1 để chọn điểm đầu của hình sau đó nhấn chuột lần 2 để chọn điểm đối diện để vẽ hình. Sau khi vẽ thì tọa độ sẽ được hiển thị trên cửa sổ console.



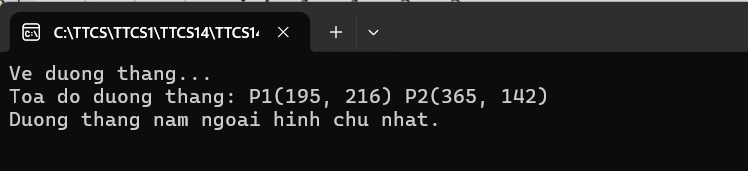
Hình 3.5. Cửa sổ console sau khi vẽ hình.



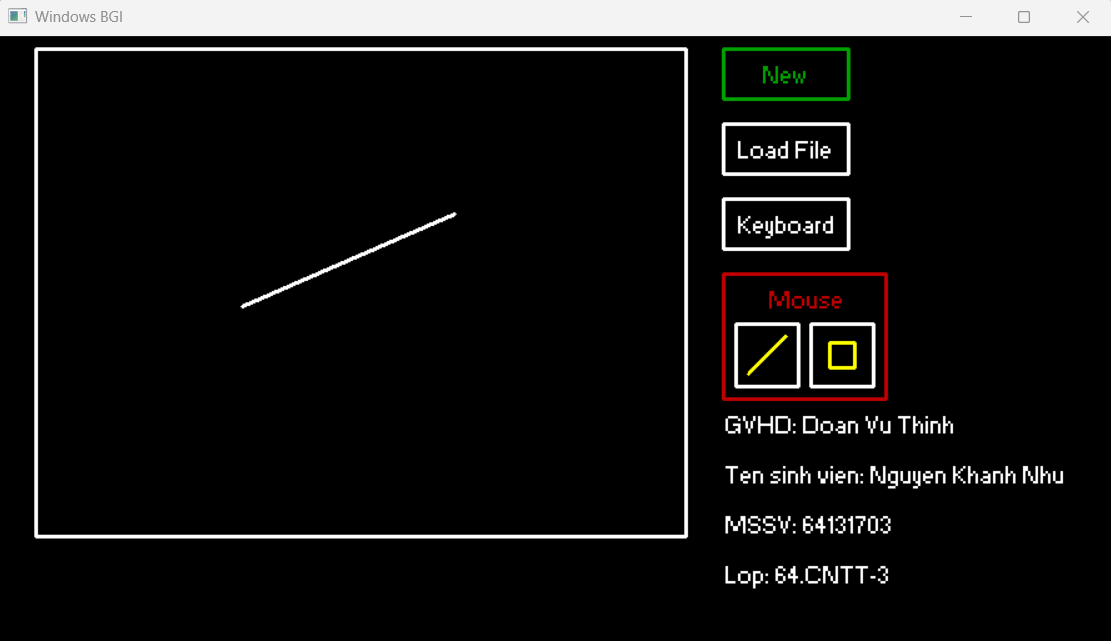
Hình 3.6. Kết quả sau khi vẽ hình.

**Nút vẽ đường thẳng:**

Đầu tiên click vào nút sau đó đưa chuột vào khung, nhấn chuột lần 1 để chọn điểm đầu của đoạn thẳng sau đó nhấn chuột lần 2 để chọn điểm cuối của đoạn thẳng. Sau khi vẽ thì tọa độ sẽ được hiển thị trên cửa sổ console.



Hình 3.7. Cửa sổ console sau khi vẽ đường thẳng.



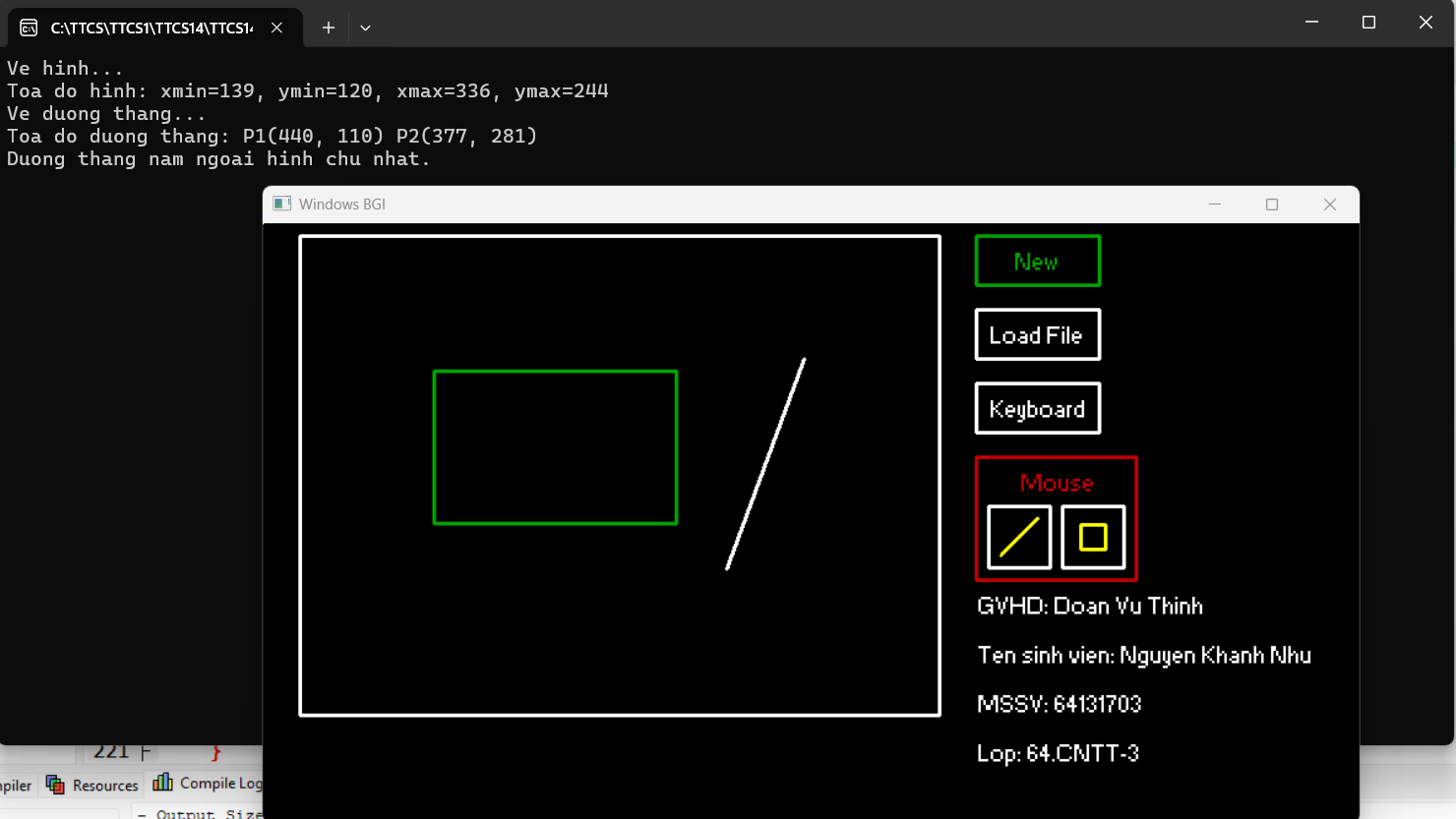
Hình 3.8. Kết quả sau khi vẽ đường thẳng.

### **3.3.2. Xén tỉa đoạn thẳng:**

Để thực hiện xén tỉa cần thực hiện vẽ hình và vẽ đường thẳng như hướng dẫn ở mục trên. Sau khi vẽ đường thẳng thì phần đường thẳng nằm trong cửa sổ xén sẽ hiển thị màu đỏ.

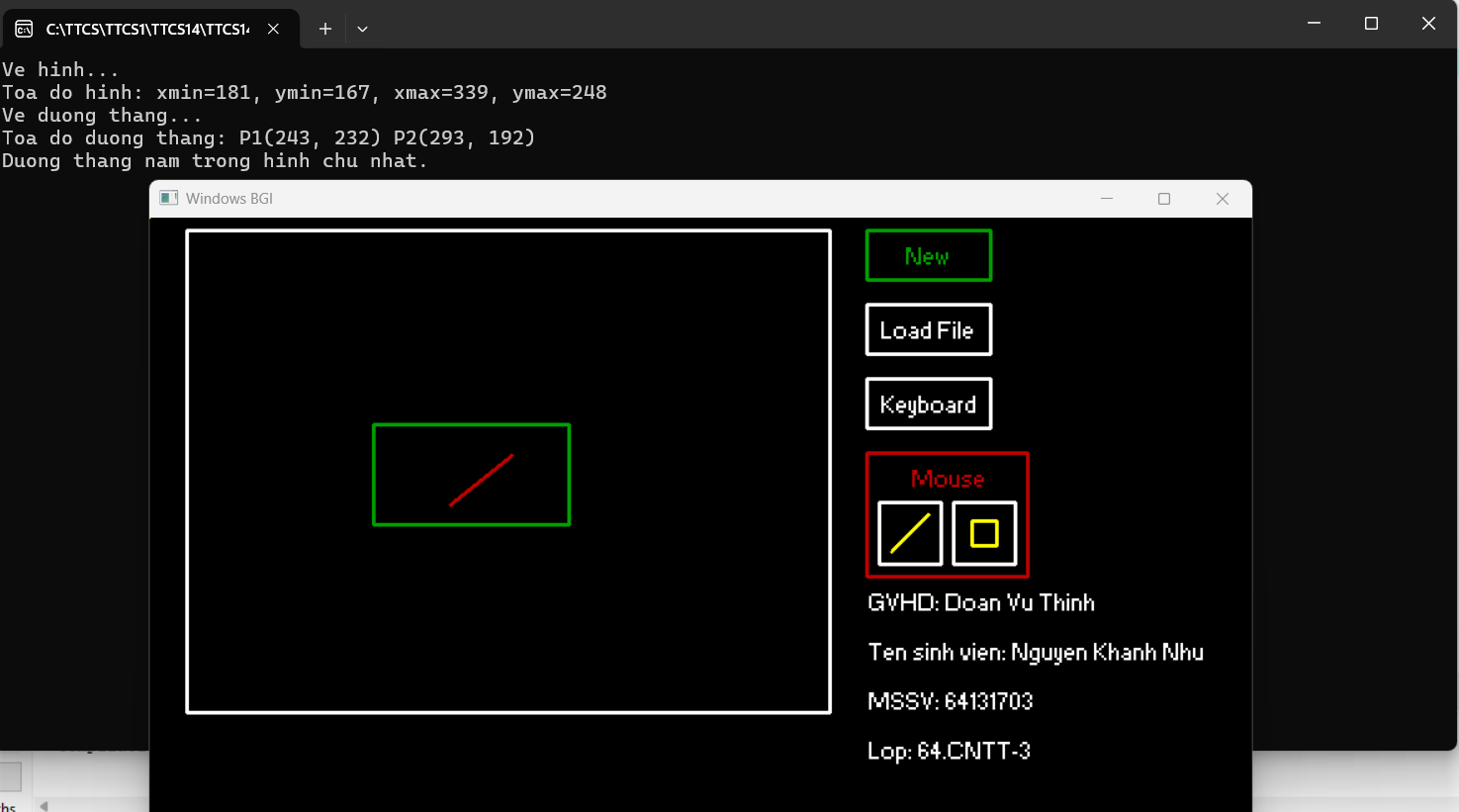
Cửa sổ console sẽ hiển thị tọa độ hình chữ nhật, tọa độ đường thẳng, các trường hợp của xén tỉa đường thẳng và hiển thị tọa độ giao điểm nếu cắt hình chữ nhật.

**Trường hợp 1:** Đường thẳng nằm ngoài hình chữ nhật.



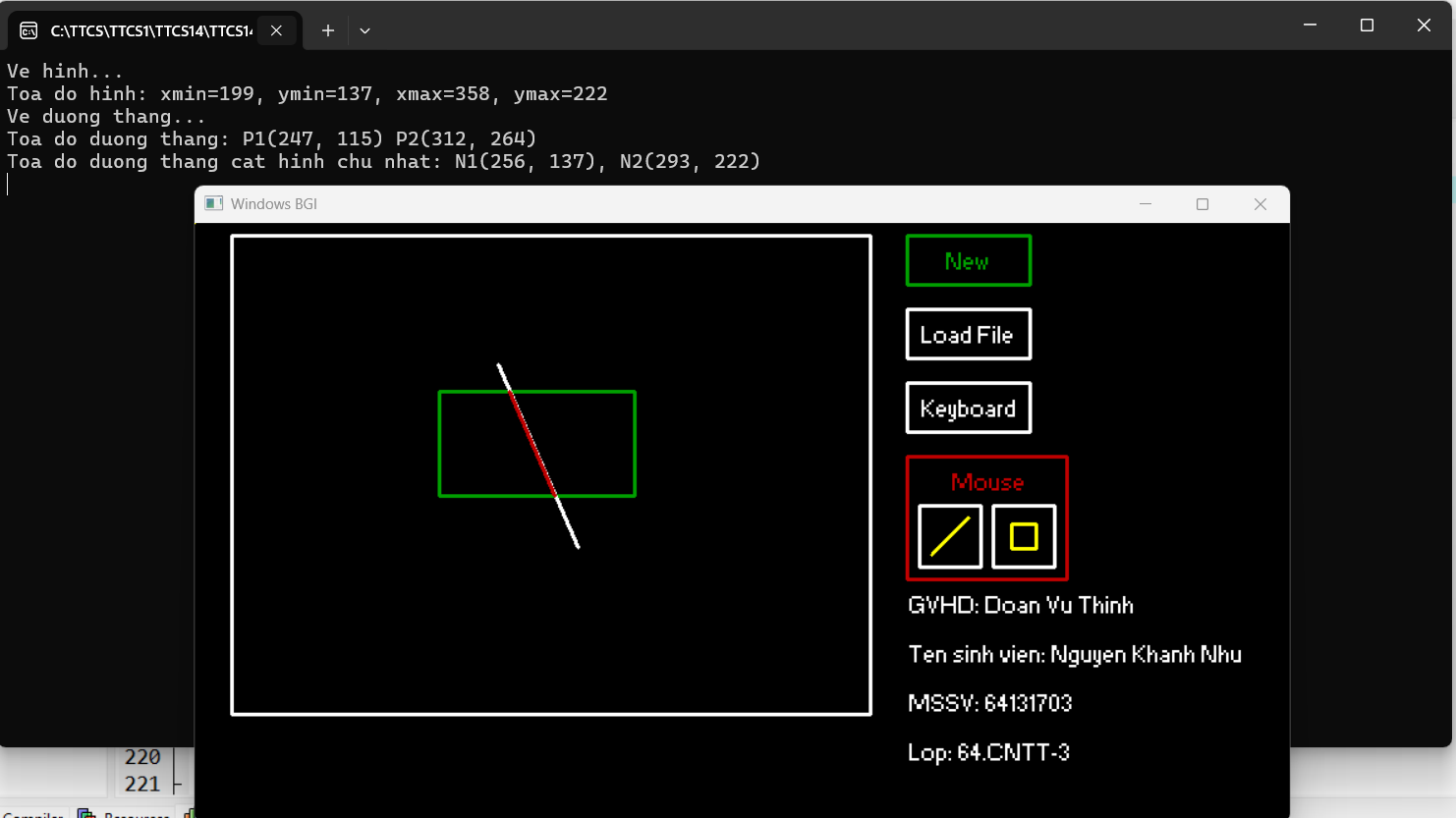
Hình 3.9. Trường hợp vẽ đường thẳng nằm ngoài hình chữ nhật.

**Trường hợp 2:** Đường thẳng nằm trong hình chữ nhật.



Hình 3.10. Trường hợp vẽ đường thẳng nằm trong hình chữ nhật.

**Trường hợp 3:** Đường thẳng cắt hình chữ nhật.



Hình 3.11. Trường hợp vẽ đường thẳng cắt hình chữ nhật.

# Chương 4: Kết luận

Bài báo cáo này đã trình bày và mô phỏng giải thuật xén tỉa đoạn thẳng bằng giải thuật Cohen-Sutherland sử dụng thư viện graphics.h trong môi trường lập trình DevC/C++. Qua quá trình tìm hiểu và thực hiện mô phỏng thì đã có thể áp dụng giải thuật Cohen-Sutherland để xén tỉa các đoạn thẳng. Việc sử dụng thư viện graphics.h giúp việc hiển thị đồ họa và các thao tác trên màn hình trở nên dễ dàng.

Tổng kết lại, bài báo cáo này đã trình bày về giải thuật Cohen-Sutherland. Việc nhập dữ liệu đầu vào thông qua 3 cách: tải file, nhập bằng phím và bằng chuột. Trong quá trình hoàn thành bài báo cáo với kinh nghiệm thực tế chưa nhiều nên báo cáo không tránh khỏi những sai sót, rất mong sự góp ý của thầy.

Em xin chân thành cảm ơn.

# TÀI LIỆU THAM KHẢO

[1] Bài giảng Kỹ thuật đồ họa, Đoàn Vũ Thịnh, 2019.

[2] Hướng dẫn giải chi tiết và lập trình Kỹ thuật đồ hoạ, Đoàn Vũ Thịnh, 2021.[3] ĐHMT Trần Văn Hưng (25.03.2019). Cohen sutherLand\_Nguyễn Tam Thành\_16CNTT1\_Nhóm 1. YouTube. [*https://www.youtube.com/watch?v=0Dc-d4r9Bp8&list=LL&index=3*](https://www.youtube.com/watch?v=0Dc-d4r9Bp8&list=LL&index=3).

[4] Phạm Thái Sơn (18.03.2023). Các thuật toán xén đoạn thẳng vào hình chữ nhật - Liang Barsky - Cohen Sutherland. YouTube. [*https://www.youtube.com/watch?v=Z54mIllFNws&t=24s*](https://www.youtube.com/watch?v=Z54mIllFNws&t=24s)*.*