BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG   
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**SỬ DỤNG CÔNG CỤ ĐỒ HỌA GRAPHICS.H ĐỂ MÔ PHỎNG PHÉP CHẾU ĐỐI TƯỢNG 3D TRONG KHÔG GIAN 2D**

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

**Sinh viên thực hiện : Nguyễn Huỳnh Gia Lục**

**Mã số sinh viên : 63132245**

**Lớp : 63.CNTT-4**

**Khánh Hoà – tháng 1/2024**

**MỤC LỤC**

[**GIỚI THIỆU CHUNG 3**](#_Toc155684880)

[**I. Nguyên lý về 3D 3**](#_Toc155684881)

[**II. Đặc điểm của kỹ thuật đồ họa 3D 3**](#_Toc155684882)

[**III. Các phương pháp hiển thị 3D 3**](#_Toc155684883)

[**IV. Phép chiếu 4**](#_Toc155684884)

[**1. Định nghĩa 4**](#_Toc155684885)

[**2. Các loại phép chiếu 4**](#_Toc155684886)

[**3. Các bước xây dựng hình chiếu 4**](#_Toc155684887)

[**PHÉP CHIẾU TRỰC GIAO 5**](#_Toc155684888)

[**I. Khái niệm và công thức 5**](#_Toc155684889)

[**1. Khái niệm 5**](#_Toc155684890)

[**2. Công thức toán học 6**](#_Toc155684891)

[**II. Tính chất chiếu vuông góc và đối xứng 6**](#_Toc155684892)

[**1. Tính chất vuông góc 6**](#_Toc155684893)

[**1. Tính chất đối xứng 6**](#_Toc155684894)

[**III. Phương pháp nghiên cứu 6**](#_Toc155684895)

[**1. Cài đặt Dev – C++ và thư viện Graphics.h 6**](#_Toc155684896)

[***2.*  Khai báo thư viện 7**](#_Toc155684897)

[***3.* Cấu trúc dữ liệu 7**](#_Toc155684898)

[**5. Các hàm vẽ đối tượng 9**](#_Toc155684899)

[**6. Có thể nhập dữ liệu từ bàn phím 13**](#_Toc155684900)

[**7. Bắt tọa độ chuột 14**](#_Toc155684901)

[**8. Giao diện người dung và giao diện demo 16**](#_Toc155684902)

[**9. Hàm Main 18**](#_Toc155684903)

[**KẾT LUẬN 20**](#_Toc155684904)

[**TÀI LIỆU THAM KHẢO 20**](#_Toc155684905)

# 

# GIỚI THIỆU CHUNG

## **Nguyên lý về 3D**

Đồ họa 3 chiều (3D computer graphics) bao gồm việc bổ xung kích thước về chiều sâu của đối tượng, cho phép ta biểu diễn chúng trong thế giới thực một cách chính xác và sinh động hơn. Tuy nhiên các thiết bị truy xuất hiện tại đều là 2 chiều, do vậy việc biểu diễn được thực thi thông qua phép tô trát (render) để gây ảo giác (illusion) về độ sâu.

Đồ hoạ 3D là việc chyển thế giới tự nhiên dưới dạng các mô hình biểu diễn trên các thiết bị hiển thị thông qua kỹ thuật tô trát (rendering).

## **Đặc điểm của kỹ thuật đồ họa 3D**

Có các đối tượng phức tạp hơn các đối tượng trong không gian 2D:

* Bao bởi các mặt phẳng hay các bề mặt
* Có các thành phần trong và ngoài
* Các phép biến đổi hình học phức tạp
* Các phép biến đổi hệ toạ độ phức tạp hơn
* Thường xuyên phải bổ xung thêm phép chiếu từ không gian 3D vào 2D
* Luôn phải xác định các bề mặt hiển thị

## **Các phương pháp hiển thị 3D**

Với các thiết bị hiển thị 2D thì chúng ta có các phương pháp sau để biểu diễn đối tượng 3D:

* Kỹ thuật chiếu (projection): Trực giao (orthographic)/phối cảnh (perspective)
* Kỹ thuật đánh dấu độ sâu (depth cueing)
* Nét khuất (visible line/surface identification)
* Tô trát bề mặt (surface rendering)
* Cắt lát (exploded/cutaway scenes, cross-sections)
* Các thiết bị hiển thị 3D:
* Kính stereo - Stereoscopic displays
* Màn hình 3D – Holograms

## **Phép chiếu**

### **Định nghĩa**

Định nghĩa về phép chiếu: Một cách tổng quát, phép chiếu là phép chuyển đổi những điểm của đối tượng trong hệ thống tọa độ n chiều thành những điểm trong hệ thống tọa độ có số chiều nhỏ hơn n.

Định nghĩa về hình chiếu: Ảnh của đối tượng trên mặt phẳng chiếu được hình thành từ phép chiếu bởi các đường thẳng gọi là tia chiếu (projector) xuất phát từ một điểm gọi là tâm chiếu (center of projection) điqua các điểm của đối tượng giao với mặt chiếu (projection plan).

### **Các loại phép chiếu**

* Phép chiếu song song (Parallel Projections) là phép chiếu mà ở đó các tia chiếu song song với nhau hay xuất phát từ điểm vô cùng.Phân loại phép chiếu song song dựa trên hướng của tia chiếu (Direction Of Projection) và mặt phẳng chiếu (projection plane).
* Phép chiếu phối cảnh (Perspective Projection) là phép chiếu mà các tia chiếu không song song với nhau mà xuất phát từ 1 điểm gọi là tâm chiếu. Phép chiếu phối cảnh tạo ra hiệu ứng về luật xa gần tạo cảm giác về độ sâu của đối tượng trong thế giới thật mà phép chiếu song song không lột tả được.Các đoạn thẳng song song của mô hình 3D sau phép chiếu hội tụ tại 1 điểm gọi là điểm triệt tiêu - vanishing point

### **Các bước xây dựng hình chiếu**

* Đối tượng trong không gian 3D với tọa độ thực được cắt theo một không gian xác định gọi là view volume.
* View volume được chiếu lên mặt phẳng chiếu. Diện tích chiếm bởi view volume trên mặt phẳng chiếu đó sẽ cho chúng ta khung nhìn.
* Là việc ánh xạ khung nhìn vào trong một cổng nhìn bất kỳ cho trước trên màn hình để hiển thị hình ảnh.

# PHÉP CHIẾU TRỰC GIAO

## **Khái niệm và công thức**

### **Khái niệm**

Phép chiếu trực giao (Orthographic projection) là phép chiếu song song và tia chiếu vuông góc với mặt phẳng chiếu thường dùng mặt phẳng z=0

Ứng với mỗi mặt phẳng chiếu ta có 1 ma trận chiếu tương ứng

Các phép chiếu trực giao hầu như được dùng để tạo ra quang cảnh nhìn từ phía trước, bên sườn, và trên đỉnh của đối tượng.

Quang cảnh phía trước, bên sườn, và phía sau của đối tượng được gọi là “mặt chiếu” (elevation).

Quang cảnh phía trên được gọi là “mặt phẳng” (plane).

Các bản vẽ trong kỹ thuật thường dùng các phép chiếu trực giao này, vì các chiều dài và góc miêu tả chính xác và có thể đo được từ bản vẽ.

Phép nhìn từ phía trước (front-view): Tia chiếu song song với trục x và mặt phẳng quan sát là yz. Phép chiếu này loại bỏ thành phần x của P.

Phép nhìn từ phía bên cạnh (side-view): Tia chiếu song song với trục y và mặt phẳng quan sát là xz. Phép chiếu này loại bỏ thành phần y của P.

Phép nhìn từ phía bên trước (front-view): Tia chiếu song song với trục z và mặt phẳng quan sát là xy. Phép chiếu này loại bỏ thành phần z của P.

Chúng ta cũng có thể xây dựng các phép chiếu trực giao để có thể quan sát nhiều hơn một mặt của một đối tượng. Các quang cảnh như thế được gọi là các phép chiếu trực giao trục lượng học (axonometric orthographic projection).

Hầu hết phép chiếu trục lượng học được dùng là phép chiếu cùng kích thước (isometric projection).

Một phép chiếu cùng kích thước được thực hiện bằng việc sắp xếp song song mặt phẳng chiếu mà nó cắt mỗi trục tọa độ ở nơi đối tượng được định nghĩa (được gọi là các trục chính) ở các khoảng cách như nhau từ ảnh gốc

### **Công thức toán học**

Trong toán học, trực giao là tổng quát hóa của khái niệm tính vuông góc trong lĩnh vực đại số tuyến tính về các dạng song tuyến tính. Hai phần tử u và v của một không gian vectơ với dạng song tuyến tính B là trực giao nếu B(u, v) = 0. Tùy vào dạng song tuyến tính, không gian vectơ có thể có vectơ khác không trực giao với chính nó.

## **Tính chất chiếu vuông góc và đối xứng**

### Tính chất vuông góc

Vector v−p vuông góc với đường chiếu L. Điều này dẫn đến tính chất quan trọng trong việc hiểu sâu về chiều góc và tương quan giữa các vector.

### **Tính chất đối xứng**

Phép chiếu trực giao là một phép ánh xạ đối xứng, có nghĩa là nếu p là điểm chiếu của v lên đường chiếu L, thì v−p cũng là một vector nằm trên đường chiếu L.

## III. **Phương pháp nghiên cứu**

## ***1. Cài đặt Dev – C++ và thư viện Graphics.h***

DevC/C++ phiên bản 5.9.2 được tải về và cài đặt theo đường dẫn sau: [*https://sourceforge.net/projects/orwelldevcpp/*](https://sourceforge.net/projects/orwelldevcpp/)*,* sau đó thư viện graphics.h được tải về và cài đặt theo các bước sau:

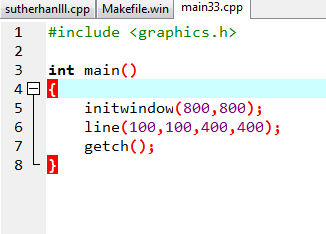
**Bước 1:** Copy tập tin 6-ConsoleAppGraphics.template và ConsoleApp\_cpp\_graph.txt vào thư mục C:\Program Files\Dev-Cpp\Templates

**Bước 2:** Copy graphics.h và winbgim.h vào thư mục C:\Program Files\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\include.

**Bước 3:** Copy libbgi.a vào thư mục C:\Program Files\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\lib.

**Bước 4:** Chọn trình biên dịch là**TDM-GCC 32 bit hay 64 bit** ứng với phiên bản Dev C++ đã cài đặt.

**Bước 5:** Tạo một Project mới bằng cách chọn File → New → Project → Console Graphics Application.



## ***2. Khai báo thư viện***

#include <stdio.h>

#include <graphics.h>

#include <stdlib.h>

#define maxdinh 10

#define MAX 100

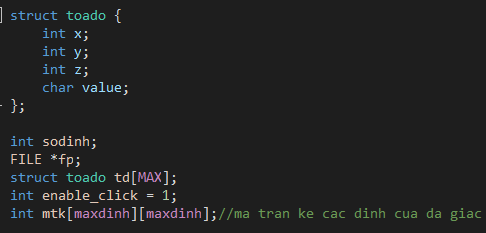
#define INPUT "points.inp"

#define MTK "mtk.inp"

## ***3. Cấu trúc dữ liệu***

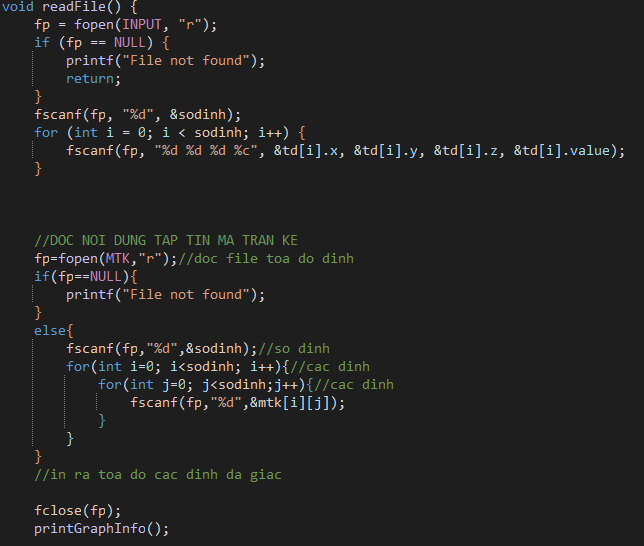
struct toado: Định nghĩa một cấu trúc để lưu trữ tọa độ (x, y, z) của mỗi điểm trên không gian và giá trị (value) của điểm đó.

int mtk[maxdinh][maxdinh]: Ma trận kề để lưu trữ thông tin về các cạnh nối giữa các điểm.



***4. Đọc dữ liệu từ 2 file***

* Mở 2 tệp tin INPUT và MTK để lấy dữ liệu lần lượt là tọa độ trục x,y,z và value. File MTK lấy dữ liệu ma trận kề nối các đỉnh lại với nhau.



### **5. Các hàm vẽ đối tượng**

1. **Vẽ các đỉnh hiển thị lên màn hình**

* Hàm drawpoint:

Nhận vào một cấu trúc toado (giả sử là biểu diễn cho một điểm) và một số nguyên color làm tham số.

Đặt màu vẽ bằng cách sử dụng hàm setcolor.

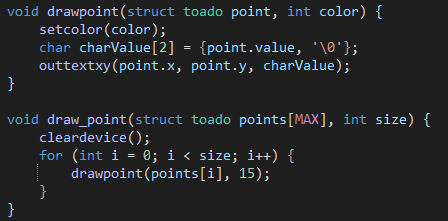
Chuyển đổi point.value thành một mảng ký tự charValue và sau đó sử dụng outtextxy để vẽ văn bản tại các tọa độ cụ thể (point.x, point.y).

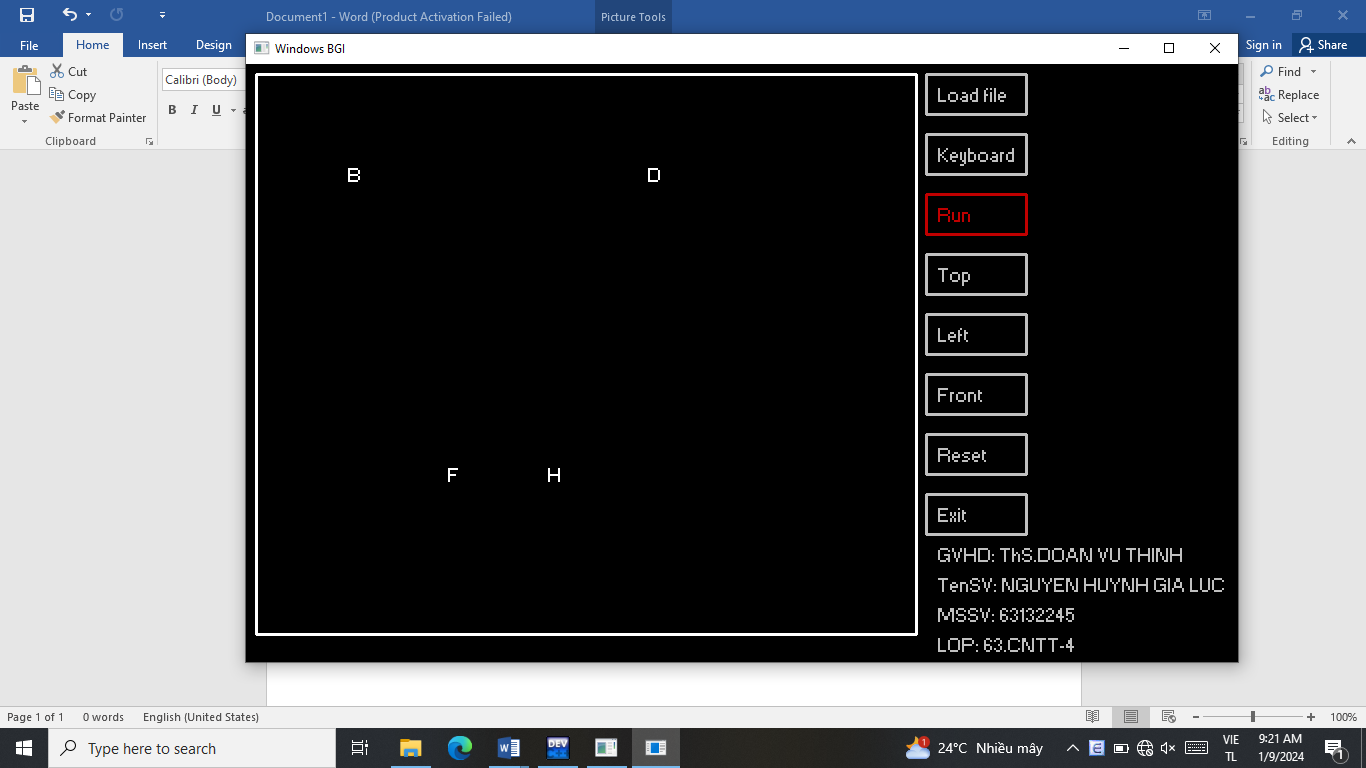
* Hàm draw\_point:

Nhận vào một mảng cấu trúc toado (points) và một số nguyên size làm tham số.

Xóa cửa sổ đồ họa bằng cách sử dụng cleardevice.

Lặp qua mảng các điểm và gọi hàm drawpoint cho mỗi điểm với giá trị màu là 15.

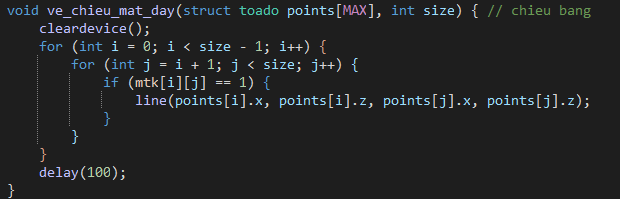


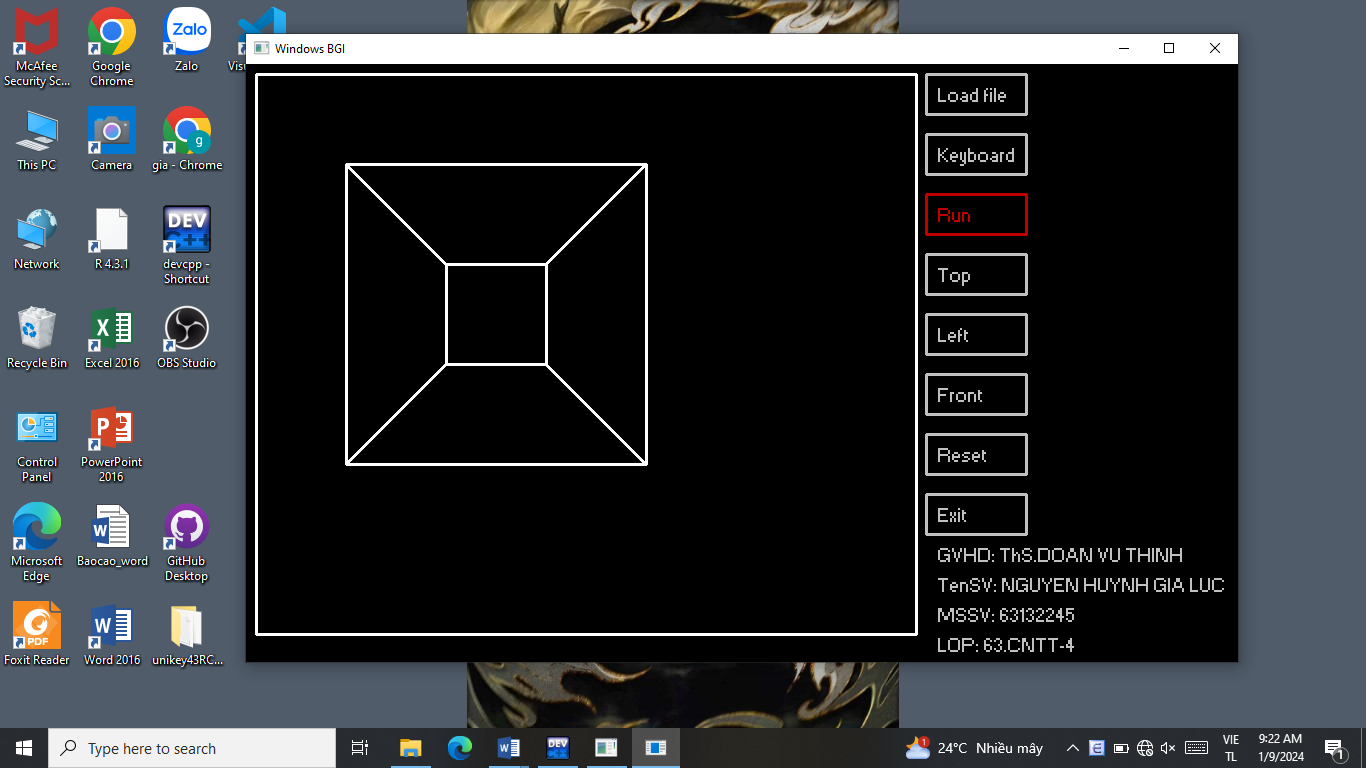


1. **Vẽ hình chiếu mặt đáy**

Hàm này sử dụng hàm line để vẽ các đoạn thẳng giữa các điểm của đa giác,

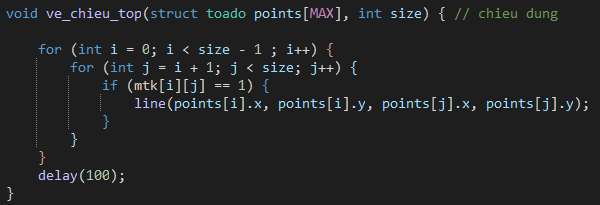
Nhưng sử dụng giá trị x và z để xác định tọa độ theo chiều bên (theo trục x và z). Hàm delay(100) có thể được sử dụng để tạo một sự trễ giữa các đoạn thẳng khi vẽ.

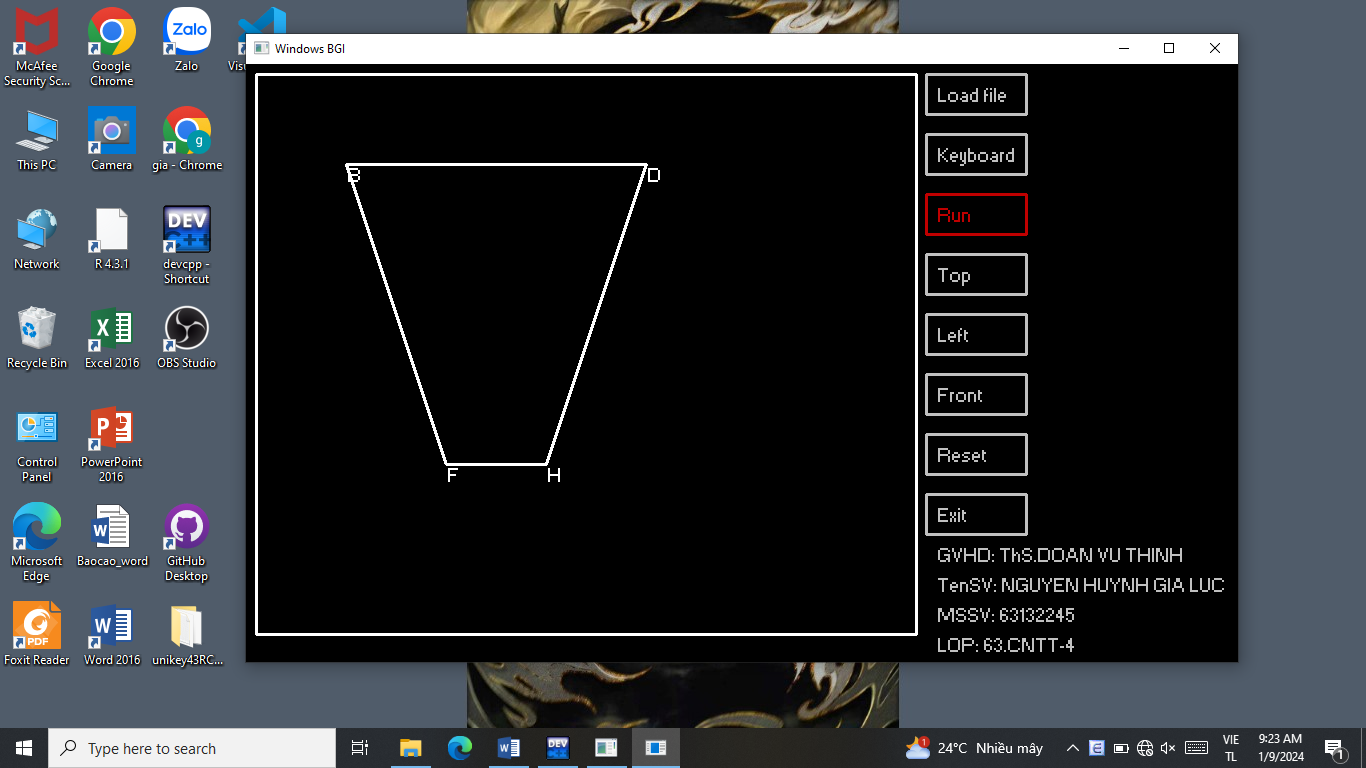




1. **Vẽ hình chiếu nhìn từ trên xuống**

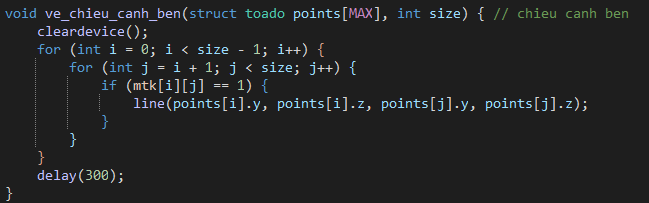
Hàm này sử dụng hàm line để vẽ các đoạn thẳng giữa các điểm của đa giác, nhưng sử dụng giá trị z thay vì y để xác định tọa độ theo chiều top (theo trục z). Tương tự như trước, hàm delay(100) có thể được sử dụng để tạo một sự trễ nhỏ giữa các đoạn thẳng khi vẽ.

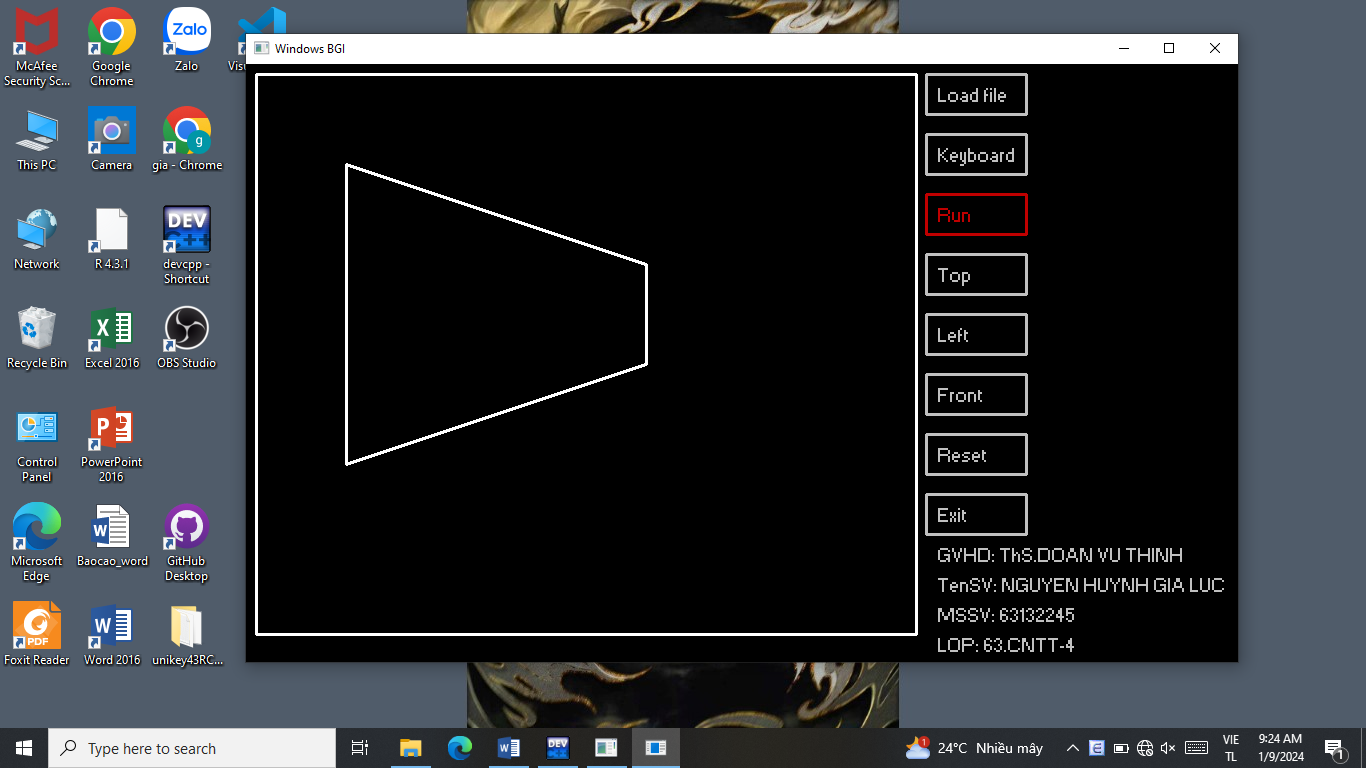




1. **Vẽ hình chiếu từ từ trái sang**

Hàm này sử dụng hàm line để vẽ các đoạn thẳng giữa các điểm của đa giác, nhưng sử dụng giá trị y và z để xác định tọa độ theo chiều bên (theo trục y và z). Hàm delay(300) có thể được sử dụng để tạo một sự trễ giữa các đoạn thẳng khi vẽ.





### **6. Có thể nhập dữ liệu từ bàn phím**

Mở tệp tin có tên là INPUT để đọc ("r" là chế độ đọc).

Kiểm tra xem tệp có tồn tại không. Nếu không tồn tại, hiển thị thông báo lỗi và thoát khỏi hàm.

Sử dụng fscanf để đọc số đỉnh (sodinh) từ tệp.

Hiển thị thông báo về số lượng đỉnh của đa giác.

Sử dụng fscanf để đọc thông tin về mỗi đỉnh từ tệp, bao gồm tọa độ x, y, z và giá trị (value).

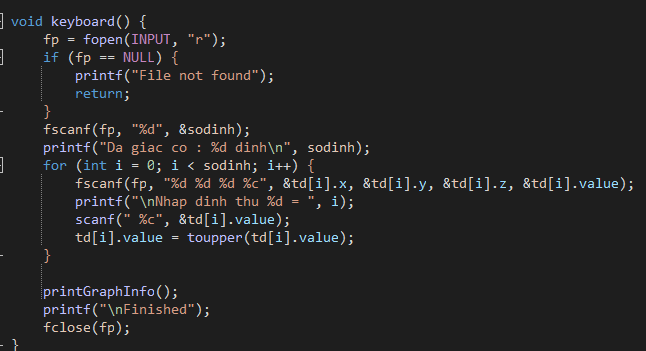
Hiển thị thông báo để nhập giá trị của value từ bàn phím.

Chuyển giá trị value thành chữ hoa bằng cách sử dụng hàm toupper.

Gọi hàm printGraphInfo để hiển thị thông tin về đa giác.

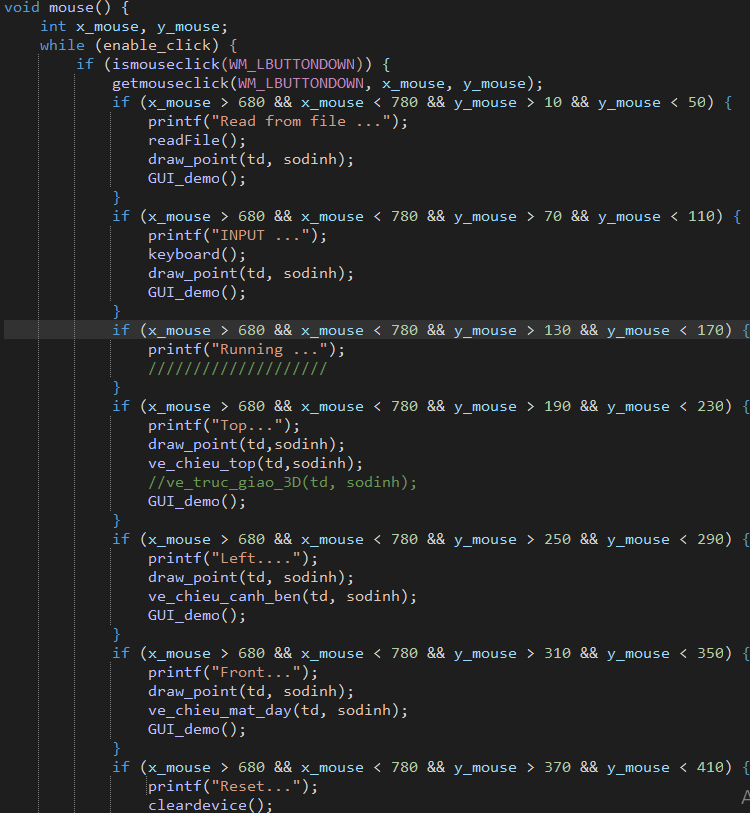
Hiển thị thông báo "Finished".

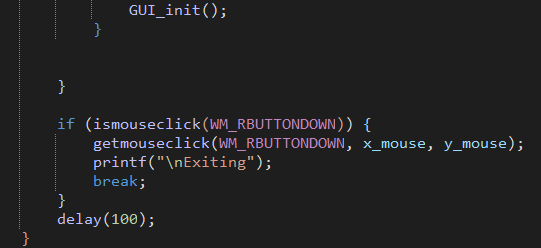
Đóng tệp tin bằng cách sử dụng fclose(fp).



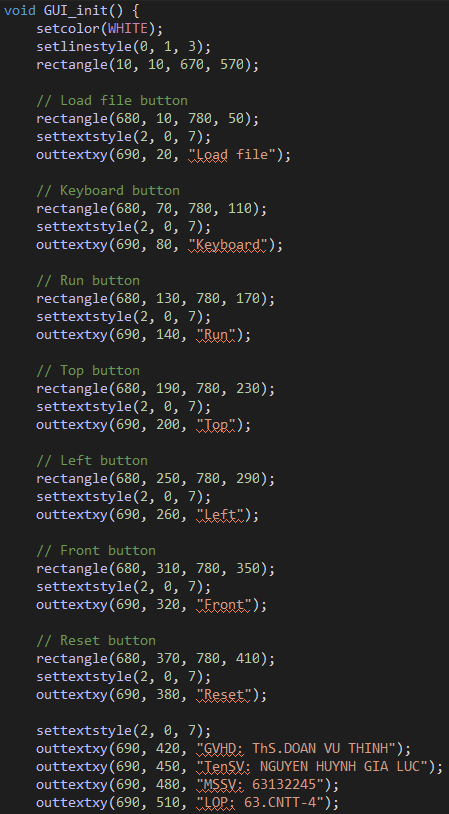
### **7. Bắt tọa độ chuột**

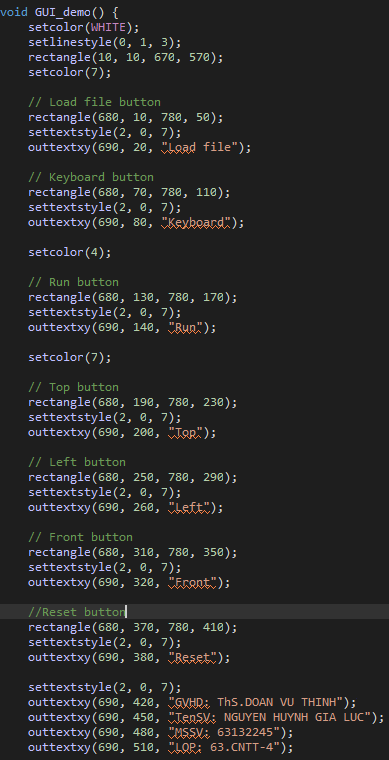
Hàm này giúp bắt sự kiện chuột khi nhấp vào tọa độ lập trình sẵn trên màn hình.





### **8. Giao diện người dung và giao diện demo**

Tạo giao diện người dùng và giao diện demo khi chạy chương trình



### **9. Hàm Main**

readFile(): Gọi hàm readFile().

* Đây là một hàm được sử dụng để đọc dữ liệu từ một tệp tin hoặc thực hiện một công việc khác liên quan đến đọc file. Đảm bảo rằng hàm này được định nghĩa và xử lý chính xác.

initwindow(800, 600): Gọi hàm initwindow từ thư viện đồ họa BGI (Borland Graphics Interface) để khởi tạo cửa sổ đồ họa với kích thước 800x600 pixel.

* Hàm này thường được sử dụng trong lập trình đồ họa để tạo ra một cửa sổ đồ họa để vẽ đồ họa và thao tác với các hình ảnh.

GUI\_init(): Gọi hàm GUI\_init().

* Tạo giao diện người dùng hiển thị lên màn hình

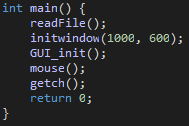
mouse(): Gọi hàm mouse().

* Hàm này bắt sự kiện khi bạn click chuột vào tọa độ đã lập trình trước sẽ chạy chức năng hàm đó

getch(): Gọi hàm getch() để chờ người dùng nhập một phím bất kỳ trước khi đóng cửa sổ.

* Hàm này thường được sử dụng để giữ cửa sổ mở sau khi chương trình kết thúc, giúp người dùng xem kết quả trước khi thoát.

return 0: Trả về giá trị 0 để báo hiệu rằng chương trình đã kết thúc mà không có lỗi.



# KẾT LUẬN

Phép chiếu trực giao không chỉ là một công cụ toán học mạnh mẽ mà còn là một lý thuyết quan trọng đằng sau nhiều ứng dụng thực tế. Sự hiểu biết vững về phép chiếu trực giao giúp ta nhìn nhận sâu sắc hơn về cách dữ liệu và không gian được xử lý, đồng thời mở rộng kiến thức trong nền tảng toán học và đại số tuyến tính.

# TÀI LIỆU THAM KHẢO

1. Giáo trình học phần Kỹ Thuật Đồ Họa của thầy Thạc sĩ Đoàn Vũ Thịnh, trường Đại học Nha Trang
2. Hướng dẫn giải chi tiết và lập trình Kỹ thuậ tđồ hoạ, Đoàn Vũ Thịnh, 2021
3. Computer Graphics, C Version (2nd Edition), Donald Hearn, 1996
4. Giải thuật và lập trình, Lê Minh Hoàng, 2006