**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SƠ**

**Sử dụng công cụ đồ hoạ graphics.h để mô phỏng sinh các đối tượng đồ hoạ cơ sở**

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

**Sinh viên thực hiện: Nguyễn Hữu Nghĩa**

**Mã số sinh viên: 63134775**

**Lớp: 63.CNTT-4**

**Khánh Hoà – tháng 12/2023**

**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 công cụ đồ hoạ graphics.h để mô phỏng sinh các đối tượng đồ hoạ cơ sở

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

**Sinh viên được hướng dẫn:** Nguyễn Hữu Nghĩa

**MSSV:** 63134775

**Khoá**: 63 **Ngành:** Công nghệ thông tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 | 27/11/2023 | Lên kế hoạch thực hiện  Lựa chọn giải pháp để cài đặt |  |
| 2 | 4/12/2023 | Làm giao diện chính  Làm thuật toán vẽ đa giác, hình tròn ,hình elip  Thuật toán nhập bàn phím, reset, exit |  |
| 3 | 18/12/2023 | Làm thuật toán flood fill  Thuật toán vẽ bằng chuột  Chỉnh lại giao diện chính |  |
| 4 | 20/12/2023 | Viết báo cáo |  |
| 5 | 25/12/2023 | Chỉnh sửa báo cáo, demo |  |
| 6 | 5/1/2024 | Sinh viên nộp bản thảo lần cuối sau khi đã chỉnh sửa các yêu cầu như đã đề ra. |  |

**Mục lục**

[Lời cảm ơn 7](#_Toc155514385)

[Giới thiệu 8](#_Toc155514386)

[CHƯƠNG 1. Tổng quan đề tài 9](#_Toc155514387)

[1.1. Thư viện đồ họa GRAPHICS.H 9](#_Toc155514388)

[1.2. Giải thuật sinh đa giác 9](#_Toc155514389)

[1.3. Giải thuật sinh đường tròn Midpoint 10](#_Toc155514390)

[1.4. Giải thuật sinh đường ellipse Midpoint 13](#_Toc155514391)

[1.5. Giải thuật vết dầu loang (Flood Fill - fill Algorithm) 15](#_Toc155514392)

[Chương 2. Phương pháp nghiên cứu 17](#_Toc155514393)

[2.1 Cài đặt thư viện graphic.h 17](#_Toc155514394)

[2.2 Tổng quan về thiết kế giao diện 17](#_Toc155514395)

[2.2.1 Sơ lược giao diện 17](#_Toc155514396)

[2.2.2 Xây dưng giao diện 18](#_Toc155514397)

[2.2.2.1 Khởi tạo cửa sổ đồ họa BGI 18](#_Toc155514398)

[2.2.2.2 Xây dựng các thực thể cửa sổ đồ họa BGI 19](#_Toc155514399)

[2.3 Xử lí sự kiện chuột 20](#_Toc155514400)

[2.3.1 Kiểm tra khi nhấn nút Mouse 22](#_Toc155514401)

[2.3.2 Kiểm tra khi nhấn nút Keyboard 23](#_Toc155514402)

[2.3.3 Kiểm tra khi nhấn nút Flood Fill 24](#_Toc155514403)

[2.3.4 Kiểm tra khi nhấn nút Reset 24](#_Toc155514404)

[2.3.5 Kiểm tra khi nhấn nút Exit 25](#_Toc155514405)

[2.3.6 Kiểm tra khi nhấn nút Polygon 25](#_Toc155514406)

[2.3.7 Kiểm tra khi nhấn nút Cricle 25](#_Toc155514407)

[2.3.8 Kiểm tra khi nhấn nút Ellipse 25](#_Toc155514408)

[2.3.9 Kiểm tra khi nhấn nút màu 25](#_Toc155514409)

[2.4 Phân tích thuật toán 26](#_Toc155514410)

[2.4.1 Dùng chuột 26](#_Toc155514411)

[2.4.2 Nhập bàn phím 29](#_Toc155514412)

[2.4.3 Giải thuật sinh đa giác 29](#_Toc155514413)

[2.4.4 Giải thuật sinh đường tròn Midpoint 30](#_Toc155514414)

[2.4.5 Giải thuật sinh đường ellipse Midpoint 31](#_Toc155514415)

[2.4.6 Giải thuật vết dầu loang (Flood Fill - fill Algorithm) 32](#_Toc155514416)

[Chương 3. Kết quả 35](#_Toc155514417)

[Chương 4 Kết luận 39](#_Toc155514418)

[4.1 Ưu điểm 39](#_Toc155514419)

[4.2 Nhược điểm 39](#_Toc155514420)

Danh mục hình

[Hình 1.1 Các đối tượng của thư viện graphic.h 9](#_Toc155514238)

[Hình 1.2 Mô phỏng giải thuật sinh đa giác 10](#_Toc155514239)

[Hình 1.3 Giải thuật sinh đường tròn Midpoint 11](#_Toc155514240)

[Hình 1.3 Giải thuật sinh đường tròn Midpoint 11](#_Toc155514241)

[Hình 1.3 Giải thuật sinh đường tròn Midpoint 12](#_Toc155514242)

[Hình 1.3 Giải thuật sinh đường tròn Midpoint 13](#_Toc155514243)

[Hình 1.4 Giải thuật sinh đường ellipse Midpoint 14](#_Toc155514244)

[Hình 1.4 Giải thuật sinh đường ellipse Midpoint 14](#_Toc155514245)

[Hình 1.4 Giải thuật sinh đường ellipse Midpoint 15](#_Toc155514246)

[Hình 1.5 Giải thuật vết dầu loang (Flood Fill - fill Algorithm) 16](#_Toc155514247)

[Hình 2.2.1 Giao diện chính sản phẩm 18](#_Toc155514248)

[Hình 2.2.2.1 Cửa sổ đồ họa BGI 19](#_Toc155514249)

[Hình 2.2.2 Toàn bộ code các thực thể cửa sổ đồ họa BGI 20](#_Toc155514250)

[Hình 2.3 Xử lí sự kiện chuột 22](#_Toc155514251)

[Hình 2.3.1 Kiểm tra khi nhấn nút Mouse 23](#_Toc155514252)

[Hình 2.3.2 Kiểm tra khi nhấn nút Keyboard 24](#_Toc155514253)

[Hình 2.3.3 Kiểm tra khi nhấn nút Flood Fill 24](#_Toc155514254)

[Hình 2.3.4 Kiểm tra khi nhấn nút Reset 25](#_Toc155514255)

[Hình 2.3.5 Kiểm tra khi nhấn nút Exit 25](#_Toc155514256)

[Hình 2.3.6 Kiểm tra khi nhấn nút Polygon 25](#_Toc155514257)

[Hình 2.3.7 Kiểm tra khi nhấn nút Cricle 25](#_Toc155514258)

[Hình 2.3.8 Kiểm tra khi nhấn nút Ellipse 25](#_Toc155514259)

[Hình 2.3.9 Kiểm tra khi nhấn nút màu 26](#_Toc155514260)

[Hình 2.4.1 Kiểm tra dùng chuột vẽ đa giác 27](#_Toc155514261)

[Hình 2.4.1 Kiểm tra dùng chuột vẽ hình tròn 28](#_Toc155514262)

[Hình 2.4.1 Kiểm tra dùng chuột vẽ hình ellipse 29](#_Toc155514263)

[Hình 2.4.4 Code giải thuật sinh đường tròn Midpoint 31](#_Toc155514264)

[Hình 2.4.5 Code giải thuật sinh đường ellipse Midpoint 32](#_Toc155514265)

[Hình 2.4.6 Code giải thuật flood fill đa giác 33](#_Toc155514266)

[Hình 2.4.6 Code giải thuật flood fill hình tròn 33](#_Toc155514267)

[Hình 2.4.6 Code giải thuật flood fill hình ellipse 34](#_Toc155514268)

[Hình 3 Giao diện chính sản phẩm 35](#_Toc155514269)

[Hình 3 Giao diện khi chọn nút Mouse 36](#_Toc155514270)

[Hình 3 Giao diện vẽ hình bằng chuột 37](#_Toc155514271)

[Hình 3 Giao diện vẽ hình bằng bàn phím 38](#_Toc155514272)

[Hình 3 Giao diện tô màu 38](#_Toc155514273)

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!

Giới thiệu

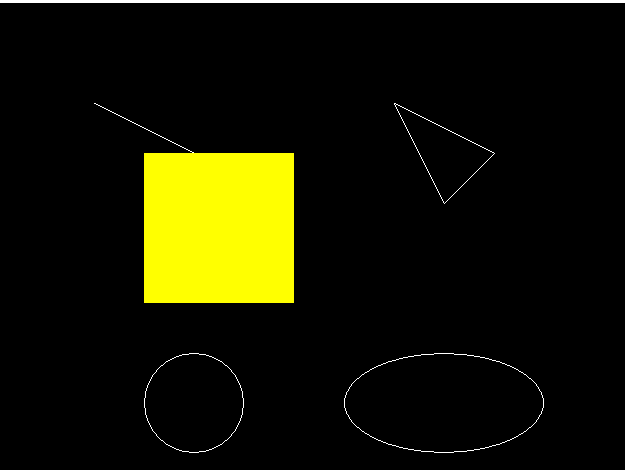
Kỹ thuật đồ họa máy tính là kỹ thuật và công nghệ chuyển đổi qua lại giữa dữ liệu và hình ảnh. Với kỹ thuật này các mô hình, hình ảnh của đối tượng được cấu tạo bởi các điểm ảnh (pixel). Các ảnh hiển thị bằng cách nạp các pixel vào vùng bộ nhớ màn hình. Với các ảnh được mô tả bằng các đối tượng hình học cơ sở, cần có một quá trình chuyển các đối tượng này về dạng ma trận các pixel trước. Quá trình này còn được gọi là quá trình chuyển đổi bằng dòng quét. Đối tượng đồ họa cơ sở đơn giản nhất là điểm và đoạn thẳng, ngoài ra còn có đường tròn, và các đường conics, mặt bậc hai, các mặt và đường splines, các vùng tô đa giác, chuỗi kí tự… Với các ảnh đồ họa phát sinh bằng máy tính, hình dạng và màu sắc của mỗi đối tượng.

Sản phẩm minh họa được quá trình mô phông thông qua việc thực hiện các giải thuật sinh đa giác, giải thuật sinh đường tròn Midpoint, giải thuật sinh ellipse Midpiont, giải thuật vết dầu loang. Sản phẩm thực hiện dựa trên dữ liệu đầu dùng chuột và bàn phím vẽ các hình và click chuột vào trong hình đang xét để xuất hiện màu

CHƯƠNG 1. Tổng quan đề tài

* 1. Thư viện đồ họa GRAPHICS.H

Là thư viện đồ họ phổ biến được sử dụng ngôn ngữ lập trình C và C++ để tạo và quản lý đồ họa trên máy tính. Thư viện này cung cấp các hàm và kiểu dữ dữ liệu cho việc vẽ các hình ảnh cơ bản như đường thẳng, hình đa giác, hình tròn, elip, tô màu, văn bản …



Hình 1.1 Các đối tượng của thư viện graphic.h

* 1. 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. 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 1.3 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 1.3 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

A diagram of a graph

Description automatically generated

Hình 1.3 Giải thuật sinh đường tròn Midpoint

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*).

* 1. Giải thuật sinh đường ellipse Midpoint

Ta có:

Nếu , điểm midpoint nằm bên trong Ellipse. Hay S là điểm kế tiếp.

Ngược lại, điểm midpoint nằm ngoài Ellipse. Hay P là điểm kế tiếp.

A diagram of a mathematical equation

Description automatically generatedA diagram of a mathematical equation

Description automatically generated

Hình 1.4 Giải thuật sinh đường ellipse Midpoint

Xét trên miền 1:

Từ pt:

Hay:

A diagram of a line graph

Description automatically generated

Hình 1.4 Giải thuật sinh đường ellipse Midpoint

Xét trên miền 2:

Từ pt:

Nếu , điểm midpoint nằm bên ngoài ellipse, gần đường biên hơn

Ngược lại, , điểm midpoint nằm bên trong ellipse, gần đường biên hơn

Từ pt:

Từ pt:

A diagram of a line graph

Description automatically generated

Hình 1.4 Giải thuật sinh đường ellipse Midpoint

* 1. Giải thuật vết dầu loang (Flood Fill - fill Algorithm)

Thuật toán tô màu loang (hay còn gọi là tô màu theo đường biên, tô lân cận). Khác với thuật toán tô màu dựa theo dòng quét, đường biên của vùng tô màu ở thuật toán tô loang được xác định bởi tập các đỉnh của 1 đa giác, đường biên trong thuật toán được mô tả bằng một giá trị duy nhất, đó là màu của tất cả các điểm thuộc về đường biên (nói ngắn gọn là chúng ta sẽ tô đường biên một màu riêng).

Bắt đầu từ 1 điểm nằm bên trong vùng tô, ta sẽ kiểm tra các điểm lân cận của nó đã được tô màu hay có phải điểm biên hay không. Nếu không phải là điểm đã tô và không phải là điểm biên ta sẽ tô màu nó. Lặp lại cho tới khi nào không còn tô được điểm nào nữa thì dừng.

A picture containing screen, drawing

Description automatically generated A picture containing screen, food, drawing

Description automatically generated

Hình 1.5 Giải thuật vết dầu loang (Flood Fill - fill Algorithm)

Thuật toán này có thể sẽ không hoạt động chính xác khi có 1 số điểm nằm trong vùng tô có màu là màu cần tô của vùng tô. Để khắc phục điều này, trước khi tô màu, cần phải đảm bảo rằng toàn bộ các điểm thuộc về vùng tô có màu khác màu tô.

Có hai quan điểm về cách tô này, đó là dùng 4 điểm lân cận hay 8 điểm lân cận đối với điểm đang xét

Nếu chọn sai điểm trung tâm (nằm trên hoặc nằm bên ngoài đa giác) kết quả là không thể tô màu

Chương 2. Phương pháp nghiên cứu

* 1. Cài đặt thư viện graphic.h

**Bước 1** Tải và cài đặt Dev\_C++ cho máy tính ([Dev-C++ download | SourceForge.net](https://sourceforge.net/projects/orwelldevcpp/))

**Bước 2** Tải thư viện đồ họa về máy tính ([ComputerGraphics/Coding/library at master · thinhdoanvu/ComputerGraphics · GitHub](https://github.com/thinhdoanvu/ComputerGraphics/tree/master/Coding/library))

**Bước 3**

Copy:

6-ConsoleAppGraphics.template

ConsoleApp\_cpp\_graph.txt

Paste**:**

C:\Program Files\Dev-Cpp\Templates

**Bước 4**

Copy:

graphics.h

winbgim.h

Paste:

C:\Program Files\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\include

**Bước 5**

Copy:

libbgi.a

Paste:

C:\Program Files\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\lib

**Bước 6**

Thiết lập trên DevC++

Open DevC - New Project - Console Graphics Application

C++ File

Project name: Hello

Save in some where

Tools - Compiler Option:

TDM-GCC 4.9.2 32bit Release

* 1. Tổng quan về thiết kế giao diện
     1. Sơ lược giao diện

Giao diện sản phẩm gồm

* Khung để vẽ các hình đa giác, hình tròn, ellipse và hiển thi màu hình
* Các nút chức năng gồm mouse, keyboard, flood fill, reset, exit, poloyon, cricle, ellipse.
* Các ô màu thẻ hiện màu để chọn tô
* Có góc chứa thông tin mssv, tên sinh viên, giáo viên hưng dẫn, lớp

A screenshot of a computer

Description automatically generated

Hình 2.2.1 Giao diện chính sản phẩm

* + 1. Xây dưng giao diện
       1. Khởi tạo cửa sổ đồ họa BGI

Trong hàm main, sử dụng hàm initwindow(width,height) với width = 800 và height = 500 để khởi tạo cửa số đồ họa có độ rộng 800 và chiều cao 500 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

* + - 1. Xây dựng các thực thể cửa sổ đồ họa BGI

Tại hàm main, gọi hàm CUI\_init để hiển thị các thực thể (khung, nút, ô màu, góc thông tin) lên cửa sổ đồ họa BGI

Đây là hàm CUI\_init

A screen shot of a computer program

Description automatically generatedA screenshot of a computer code

Description automatically generatedA screenshot of a computer code

Description automatically generated

Hình 2.2.2 Toàn bộ code các thực thể cửa sổ đồ họa BGI

* 1. Xử lí sự kiện chuột

Thư viện BGI cung cấp 1 số phương pháp thao tác với chuột và bàn phím

Các loại sự kiện chuột:

* WM\_MOUSEMOVE: phát hiện chuyển động của chuột
* WM\_LBUTTONDBCLK: phát hiện khi nút chuột trái được nhấp đúp
* WM\_LBUTTONDOWN: phát hiện khi nút chuột trái được nhấn xuống
* WM\_LBUTTONUP: phát hiện khi nút chuột trái được thả lên
* WM\_MBUTTONDBCLK: phát hiện khi nút chuột giữa được nhấp đúp
* WM\_MBUTTONDOWN: phát hiện khi nút chuột giữa được nhấp xuống
* WM\_MBUTTONUP: phát hiện khi nút chuột giữa được thả ra
* WM\_RBUTTONDBCLK: phát hiện khi nút chuột phải được nhấp đúp
* WM\_RBUTTONDOWN: phát hiện khi nút chuột phải được nhấn xuống
* WM\_RBUTTONUP: phát hiện khi nút chuột phải được thả lêN
* Khai báo x\_mouse và y\_mouse để lưu trữ tọa độ của click chuột.
* Khai báo isShapeSelected, isMouseClicked và isKeyboardClicked được sử dụng để theo dõi trạng thái của các hoạt động liên quan đến hình dạng và chuột.
* Vòng lặp vô hạn (while (1)) để liên tục kiểm tra sự kiện click chuột.
* Trong vòng lặp, mã kiểm tra xem có sự kiện click chuột trái (ismouseclick(WM\_LBUTTONDOWN)) hay không. Nếu có, nó sử dụng getmouseclick(WM\_LBUTTONDOWN, x\_mouse, y\_mouse) để lấy tọa độ x và y của click chuột.
* Tiếp tục kiểm tra các vị trí click chuột để xác định hành động tương ứng

A screenshot of a computer program

Description automatically generated

Hình 2.3 Xử lí sự kiện chuột

* + 1. Kiểm tra khi nhấn nút Mouse

Kiểm tra xem đã click chuột vào các nút polygon, criclr, ellipse chưa

Nếu đã click chuột vào các nút đó thì nút mouse sẽ xuất hiện màu đỏ

Click chuột nút mouse thì chuyển sanng màu trắng

Khi vẽ xong nút mouse sẽ trở lại ban đầu

A screenshot of a computer program

Description automatically generated

Hình 2.3.1 Kiểm tra khi nhấn nút Mouse

* + 1. Kiểm tra khi nhấn nút Keyboard

Kiểm tra xem đã click chuột vào các nút polygon, criclr, ellipse chưa

Nếu đã click chuột vào các nút đó thì nút keyboard sẽ xuất hiện màu đỏ

Click chuột nút keyboard thì chuyển sanng màu trắng

Khi vẽ xong nút keyboard sẽ trở lại ban đầu

A screenshot of a computer code

Description automatically generated

Hình 2.3.2 Kiểm tra khi nhấn nút Keyboard

* + 1. Kiểm tra khi nhấn nút Flood Fill

A screenshot of a computer code

Description automatically generated

Hình 2.3.3 Kiểm tra khi nhấn nút Flood Fill

* + 1. Kiểm tra khi nhấn nút Reset

Có chức năng xóa các hình đã vẽ và cho số đỉnh bằng 0

A white background with red and blue text

Description automatically generated

Hình 2.3.4 Kiểm tra khi nhấn nút Reset

* + 1. Kiểm tra khi nhấn nút Exit

Có chức năng thoát ta chương trình

A close up of a number

Description automatically generated

Hình 2.3.5 Kiểm tra khi nhấn nút Exit

* + 1. Kiểm tra khi nhấn nút Polygon

A white background with black and red text

Description automatically generated

Hình 2.3.6 Kiểm tra khi nhấn nút Polygon

* + 1. Kiểm tra khi nhấn nút Cricle

A close up of a white background

Description automatically generated

Hình 2.3.7 Kiểm tra khi nhấn nút Cricle

* + 1. Kiểm tra khi nhấn nút Ellipse

A white background with black and red text

Description automatically generated

Hình 2.3.8 Kiểm tra khi nhấn nút Ellipse

* + 1. Kiểm tra khi nhấn nút màu

Khi click chuột ô có sẵn màu thì sẽ tô màu đó cho đa giác, hình tròn. ellipse

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

Hình 2.3.9 Kiểm tra khi nhấn nút màu

* 1. Phân tích thuật toán
     1. Dùng chuột
* Polygon
* Kiểm tra xem nhấp chuột trái được hay chưa
* Hàm getmouseclick được sử dụng để lấy tọa độ td[i].x, td[i].y
* Kiểm tra xem có nằm trong ô to dùng để vẽ các hình đa giác, hình tròn, ellipse và hiển thi màu hình
  + Gán tọa độ x và y cho điểm trong một mảng td[i].x, td[i].y
  + Tăng số đỉnh lên 1 đơn vị
  + Vẽ 1 tọa độ (x,y) trên màn hình với màu trắng
  + Kiểm tra xem có ít nhất 3 điểm được chọn để vẽ đa giác

**A computer screen shot of a code

Description automatically generated**

Hình 2.4.1 Kiểm tra dùng chuột vẽ đa giác

* Cricle
* Kiểm tra xem nhấp chuột trái được hay chưa
* Hàm getmouseclick được sử dụng để lấy tọa độ td[i].x, td[i].y
* Kiểm tra xem có nằm trong ô to dùng để vẽ các hình đa giác, hình tròn, ellipse và hiển thi màu hình
  + Gán tọa độ x và y cho điểm trong một mảng td[i].x, td[i].y
  + Tăng số đỉnh lên 1 đơn vị
  + Vẽ 1 tọa độ (x,y) trên màn hình với màu trắng
  + Kiểm tra xem có 2 điểm được chọn để vẽ hình tròn

**A computer code with text

Description automatically generated with medium confidence**

Hình 2.4.1 Kiểm tra dùng chuột vẽ hình tròn

* Ellipse
* Kiểm tra xem nhấp chuột trái được hay chưa
* Hàm getmouseclick được sử dụng để lấy tọa độ td[i].x, td[i].y
* Kiểm tra xem có nằm trong ô to dùng để vẽ các hình đa giác, hình tròn, ellipse và hiển thi màu hình
  + Gán tọa độ x và y cho điểm trong một mảng td[i].x, td[i].y
  + Tăng số đỉnh lên 1 đơn vị
  + Vẽ 1 tọa độ (x,y) trên màn hình với màu trắng
  + Kiểm tra xem có 2 điểm được chọn để vẽ hình ellipse

**A screenshot of a computer program

Description automatically generated**

Hình 2.4.1 Kiểm tra dùng chuột vẽ hình ellipse

* + 1. Nhập bàn phím
* Nhập số đỉnh
* Nhập số tọa độ (td[i].x, td[i].y)

**A computer code with text and symbols

Description automatically generated**

Hình 2.4.2 Nhập bàn phím

* + 1. Giải thuật sinh đa giác
* Nếu số đỉnh nhỏ hơn 3 thì thông báo sô đỉnh không hợp lệ và kết thúc
* Nếu số đỉnh đủ điều kiện
* Thiết lập màu vẽ màu trắng
* Sử dụng thuật toán line để vẽ một đoạn thẳng nối các điểm.
* Sau khi nối các điểm tiếp theo sử dụng thuật toán line vẽ đoạn thẳng cuối cùng nối điểm cuối cùng với điểm đầu tiên hoàn thành vẽ đa giác

**A white background with text and numbers

Description automatically generated with medium confidence**

Hình 2.4.3 Code giải thuật sinh đa giác

* + 1. Giải thuật sinh đường tròn Midpoint
* Nếu số đỉnh khác 2 thì thông báo sô đỉnh không hợp lệ và kết thúc
* Nếu số đỉnh đủ điều kiện
  + Thiết lập màu trắng để vẽ
  + Tính bán kính của đường tròn r = sqrt(pow(td[1].x - td[0].x, 2) + pow(td[1].y - td[0].y, 2))
  + Dùng vòng lặp while và hàm putpixel để vẽ các điểm trên đường tròn
  + Sử dụng midpoint đề cập nhật giá trị p và chọn điểm tiếp theo trên đường tròn
  + Tăng x lên 1 đơn vị

**A screenshot of a computer code

Description automatically generated**

Hình 2.4.4 Code giải thuật sinh đường tròn Midpoint

* + 1. Giải thuật sinh đường ellipse Midpoint
* Nếu số đỉnh khác 2 thì thông báo sô đỉnh không hợp lệ và kết thúc
* Nếu số đỉnh đủ điều kiện
  + Thiết lập màu trắng để vẽ
  + Tính hai bán trục của hình ellipse dựa vào khoảng cách hai điểm trong mản td
  + Sử dụng thuật toán midpoint để tính toán các điểm trên ellipse
  + Sử dụng vòng lặp while để vẽ nửa trên hình ellipse
  + Sau khi vẽ xong nửa trên rồi tính lại giá trị để vẽ nữa dưới của hình

**A screenshot of a computer program

Description automatically generated**

Hình 2.4.5 Code giải thuật sinh đường ellipse Midpoint

* + 1. Giải thuật vết dầu loang (Flood Fill - fill Algorithm)
* Đa giác
* Kiểm tra xem click chuột có nằm bên trong đa giác đang xét
* Sử dụng thuật toán flood fill để tô màu hình đa giác

A screen shot of a computer code

Description automatically generated

Hình 2.4.6 Code giải thuật flood fill đa giác

* Hình tròn
* Kiểm tra click chuột có nằm bên trong hình tròn đang xét
* Tính bán kính hình tròn
* Sử dụng thuật toán flood fill để tô màu hình tròn

**A computer screen shot of a code

Description automatically generated**

Hình 2.4.6 Code giải thuật flood fill hình tròn

* Hình ellipse
* Khởi tạo các biến centerX, centerY, radiusX, và radiusY để lưu trữ thông tin về tâm và bán kính của hình elip
* Kiểm tra click chuột có nằm bên trong hình elip đang xét
* Tính toán khoảng cách giữa tâm hình elip và điểm được nhấp chuột.
* Sử dụng thuật toán flood fill để tô màu hình elip

A screen shot of a computer code

Description automatically generated

Hình 2.4.6 Code giải thuật flood fill hình ellipse

Chương 3. Kết quả

A screenshot of a computer

Description automatically generated

Hình 3 Giao diện chính sản phẩm

A screenshot of a computer

Description automatically generatedHình 3 Giao diện khi chọn các hình

A screenshot of a computer

Description automatically generated

Hình 3 Giao diện khi chọn nút Mouse

A screenshot of a computer

Description automatically generatedHình 3 Giao diện khi click chuột vào khung để vẽ hình

A screenshot of a computer

Description automatically generated

Hình 3 Giao diện vẽ hình bằng chuột

A screenshot of a computer

Description automatically generatedHình 3 Giao diện khi chọn nút Keyboard

A screenshot of a computer

Description automatically generated

Hình 3 Giao diện vẽ hình bằng bàn phím

A screenshot of a computer

Description automatically generated

Hình 3 Giao diện tô màu

Chương 4 Kết luận

4.1 Ưu điểm

* Nắm vững được kiến thức giải thuật sinh đa giác, giải thuật sinh đường tròn Midpoint, giải thuật sinh ellipse Midpiont, giải thuật vết dầu loang
* Làm được giao diện cho phù hợp dễ nhìn
* Nắm kỹ năng vẽ đồ họa của môn Kỹ thuật đồ họa

4.2 Nhược điểm

* Chưa mô tả được tô màu từ từ
* Xử lý code rườm rà và dài dòng

**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] Computer Graphics, C Version (2nd Edition), Donald Hearn, 1996

[4] Giải thuật và lập trình, Lê Minh Hoàng, 2006