**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



**PHÙNG TRI AN**

**NGUYỄN BÌNH**

ĐỒ HỌA MÁY TÍNH

**BÀI TẬP LỚN**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, THÁNG 01 NĂM 2022BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

XÂY DỰNG TRÒ CHƠI BẮN VỊT TRỜI VỚI THƯ VIỆN GRAPHICS.H

**BÁO CÁO BÀI TẬP LỚN ĐỒ HỌA MÁY TÍNH**

Ngành: Khoa học máy tính

Lớp: DH19CS01

Sinh viên thực hiện

Phùng Tri An – MSSV: 1851050002

Nguyễn Bình – MSSV: 1851050007

Giảng viên giảng dạy: ThS. Võ Thị Hồng Tuyết

**TP. HỒ CHÍ MINH, THÁNG 01 NĂM 2022**

**LỜI MỞ ĐẦU**

Ngày nay, với sự phát triển của công nghệ thông tin kỹ thuật số, mọi công việc và nhu cầu của chúng ta đều có thể đưa lên máy tính, ứng dụng phần mềm. Trong lĩnh vực giải trí nói riêng, sự tiến bộ của khoa học, kỹ thuật số đã làm cho các trò chơi, tựa game thêm phần đa đạng và phong phú hơn. Trong đó có thể kể đến trò chơi bắn vịt trời là một trong những tựa game khá phổ biến với các bạn học sinh, sinh viên, giúp chúng ta giải trí trong những lúc học tập hoặc công việc căng thẳng. Mô típ trò chơi đơn giản nhưng lại đầy thử thách làm cho người chơi có tinh thần và giải trí hơn. Nhờ vào sự tiến bộ của tin học, trò chơi bắn vịt trời đã áp ụng những công nghệ việc thiết kế đối tượng đồ họa thêm sinh động, những họa tiết cuốn hút người chơi đã giúp cho tựa game này khá phổ biến không chỉ ở khu vực của mình mà ở toàn thế giới. Hiện nay, chúng ta đã có thể chơi trên nhiều nền tảng khác nhau như trên điện thoại, máy tính bảng, laptop, máy vi tính,…

Đến với đồ họa máy tính, chúng em được phổ cập những kiến thức cơ bản về các đường vẽ, những thuật toán trong đồ họa. Những điều này làm cho chúng em có cơ hội và khả năng xây dựng lại trò chơi bắn vịt trời. Đồng thời chúng em cũng có thể cũng cố và ôn tập lại kiến thức môn học đồ họa máy tính, sử dụng thành thạo các hàm, thuật toán trong thư viện Graphics.h.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ 6](#_Toc92372462)

[Chương 1. GIỚI THIỆU ĐỀ TÀI 7](#_Toc92372463)

[1.1. Giới thiệu về đề tài 7](#_Toc92372464)

[1.2. Mục tiêu đề tài 7](#_Toc92372465)

[1.3. Giới hạn đề tài 7](#_Toc92372466)

[1.4. Bố cục của đề tài 8](#_Toc92372467)

[Chương 2. TRÒ CHƠI BẮN VỊT TRỜI 9](#_Toc92372468)

[2.1. Cơ sở lý thuyết 9](#_Toc92372469)

[2.1.1. Thư viện Graphics.h 9](#_Toc92372470)

[2.1.2. Các thuật toán vẽ đường 10](#_Toc92372471)

[2.1.3. Thuật toán vẽ đường thẳng DDA (Digital Differential Analyzer) 10](#_Toc92372472)

[2.1.4. Thuật toán vẽ đường tròn Bresenham 13](#_Toc92372473)

[2.1.5. Sự kiện bắt chuột 15](#_Toc92372474)

[2.1.6. Phép biến đổi 2D cơ bản 16](#_Toc92372475)

[2.2. Vẽ đối tượng đồ họa 17](#_Toc92372476)

[2.2.1. Vẽ vịt trời 17](#_Toc92372477)

[2.2.2. Vẽ virus 19](#_Toc92372478)

[2.2.3. Vẽ cung bắn 21](#_Toc92372479)

[2.3. Thuật giải trò chơi 22](#_Toc92372480)

[2.3.1. Ý tưởng xử lý giao diện menu 22](#_Toc92372481)

[2.3.2. Ý tưởng giao diện chơi 23](#_Toc92372482)

[2.3.3. Ý tưởng giao diện kết thúc trò chơi 23](#_Toc92372483)

[2.4. Hiện thực trò chơi 24](#_Toc92372484)

[2.4.1. Giao diện menu 24](#_Toc92372485)

[2.4.2. Giao diện chơi 25](#_Toc92372486)

[2.4.3. Giao diện kết thúc trò chơi 27](#_Toc92372487)

[Chương 3. Kết luận 28](#_Toc92372488)

[3.1. Những kết quả đạt được 28](#_Toc92372489)

[3.2. Ưu điểm của đề tài 28](#_Toc92372490)

[3.3. Nhược điểm của đề tài 28](#_Toc92372491)

[3.4. Hướng phát triển trong tương lai 29](#_Toc92372492)

[Tài liệu tham khảo 30](#_Toc92372493)

**DANH MỤC HÌNH VẼ**

[Hình 2‑1: Màn hình đồ họa theo hệ trục tọa độ bàn tay trái 9](#_Toc92651713)

[Hình 2‑2: Màn hình đồ họa BGI 10](#_Toc92651714)

[Hình 2‑3: Thuật toán DDA 11](#_Toc92651715)

[Hình 2‑4: Lưu đồ thuật toán DDA 12](#_Toc92651716)

[Hình 2‑5: Tám điểm đối xứng trên đường tròn 13](#_Toc92651717)

[Hình 2‑6: Thuật toán Bresenham 13](#_Toc92651718)

[Hình 2‑7: Lưu đồ thuật toán Bresenham 14](#_Toc92651719)

[Hình 2‑8: Phép tịnh tiến điểm P thành điểm Q 16](#_Toc92651720)

[Hình 2‑9: Quay quanh một góc theta 16](#_Toc92651721)

[Hình 2‑10: Quay quanh một góc bất kỳ 17](#_Toc92651722)

[Hình 2‑11: Vẽ thân và mắt vịt 17](#_Toc92651723)

[Hình 2‑12: Vẽ phần bụng vịt 18](#_Toc92651724)

[Hình 2‑13: Vẽ miệng vịt 18](#_Toc92651725)

[Hình 2‑14: Vẽ cánh vịt 19](#_Toc92651726)

[Hình 2‑15: Vẽ chân vịt 19](#_Toc92651727)

[Hình 2‑16: Đối tượng vịt trời sau khi tô màu 20](#_Toc92651728)

[Hình 2‑17: Định hình các đỉnh gai 20](#_Toc92651729)

[Hình 2‑18: Vẽ các gai protein 21](#_Toc92651730)

[Hình 2‑19: Đối tượng virus khi tô màu 21](#_Toc92651731)

[Hình 2‑20: Vẽ nữa vòng tròn cung 22](#_Toc92651732)

[Hình 2‑21: Đối tượng cung tên 22](#_Toc92651733)

[Hình 2‑22: Giao diện menu 24](#_Toc92651734)

[Hình 2‑23: Giao diện chơi ban đầu 25](#_Toc92651735)

[Hình 2‑24: Giao diện chơi khi đạt trên 50 điểm 26](#_Toc92651736)

[Hình 2‑25: Giao diện chơi khi đạt trên 200 điểm 26](#_Toc92651737)

[Hình 2‑26: Giao diện kết thúc trò chơi 27](#_Toc92651738)

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu về đề tài

Trò chơi bắn vịt trời là trò được được nhiều người biết đến và yêu thích nhờ vào những quy luật đơn giản. Là một trò chơi nhắm bắn đòi hỏi người chơi cần tính tập trung để nhắm bắn chính xác đối tượng, người chơi ban đầu có 3 lượt bắn, khi bắn trúng vịt người chơi được cộng 10 điểm, bắn hụt người chơi mất 1 lượt bắn. Khi hết lượt bắn trò chơi sẽ két thúc. Trò chơi mang tính giải trí cao nhưng cũng không kém phần thử thách để tăng sự kích thích, yêu thích với tựa game này.

Trò chơi bắn vịt trời đã được phát triển và xây dựng khá lâu nhưng đến hiện nay vẫn còn khá nhiều các bạn trẻ cũng như người trưởng thành chơi. Đến với môn học đồ họa máy tính, chúng em được học những kiến thức cơ bản về cách vẽ và tạo hình đối tượng, nhân vật. Phần nào đó giúp chúng em có cơ hội tái hiện lại trò chơi bắn vịt này với đường nét riêng, cách vẽ những đối tượng linh hoạt, sinh động hơn. Đồng thời cũng là dịp để chúng em có thể biết được thêm nhiều kiến thức về thư viện Graphics.h cũng như những thuật toán thuật giải khác trong đồ họa máy tính.

## Mục tiêu đề tài

Mục tiêu của chúng em trong đề tài là hiện thực lại trò chơi bắn vịt trời, xây dựng và cài đặt những chức năng cần thiết của trò chơi, biết thêm nhiều hàm, thuật giải và cách tổ chức xây dựng một trò chơi trong thư viện Graphics.h cũng như trong ngôn ngữ lập trình nói chung.

## Giới hạn đề tài

Đề tài được chúng em xây dựng trong khuôn khổ ngôn ngữ C++ và thư viện đồ họa Graphics.h

Các đối tượng đồ họa được vẽ nhờ các thuật toán vẽ đường thẳng, đường tròn,…

## Bố cục của đề tài

Đề tài này được chia thành ba chương như sau:

* Chương 1 nhằm giới thiệu tổng quan về trò chơi bắn vịt trời, mục tiêu và phạm vi nghiên cứu của đề tài.
* Chương 2 tìm hiểu sơ lược đồ họa và thư viện sử dụng, các thuật toán vẽ hình và các đối tượng đồ họa trong đề tài.
* Chương 3 đưa ra kết luận chung, ưu và nhược điểm từ đó đưa ra hướng phát triển cho đề tài.

# TRÒ CHƠI BẮN VỊT TRỜI

## Cơ sở lý thuyết

### Thư viện Graphics.h

Graphics.h (Borland Graphics Interface - BGI) là một thư viện được biết đến nhiều trên tền tảng DOS, chạy trong hệ điều hành window các thời kỳ đầu. Thư viện này hỗ trợ mạnh mẽ cho ngôn ngữ C/C++, cung cấp cho người dùng 2 file graphics.h và graphics.lib gồm các lệnh và hàm trong đồ họa.

Thư viện khởi tạo và sử dụng khá đơn giản. Vì vậy đến bây giờ vẫn được nhiều người sử dụng và các trường học áp dụng vào trong việc giảng dạy.

#### Khởi tạo thư viện màn hình đồ họa BGI

Hệ trục tọa đồ Descartes: Gốc tọa độ O(0, 0), trục hoành là Ox, trục tung là Oy. Hệ trục tọa độ được giới hạn trong phạm vi hoành độ x và tung độ y.

Diagram

Description automatically generatedCó 2 quy ước: Hệ trục tọa độ bàn tay trái và hệ trục tọa độ bàn tay phải. Trong ngôn ngữ C/C++ được quy ước theo hệ trục tọa độ màn tay trái.

Hình 2‑1: Màn hình đồ họa theo hệ trục tọa độ bàn tay trái

Ở hình trên ta quan sát được, màn hình đồ độ có tọa độ mặc định x từ 0 đến 639 và y từ 0 đến 479.

Khai báo và gọi thư viện sử dụng:

#include "graphics.h"

#pragma comment(lib, "graphics.lib")

Khai báo chế độ đồ họa

void initgraph (int \*graphdriver, int \*graphmode, char \*s);

Trong đó:

* Graphdriver = VGA (hoặc 9) thì graphmode có thể là VGALO (hay 0) ứng với độ phân giải màn hình 640 x 200 pixels.
* Graphdriver = VGAHI (hoặc 2) ứng với độ phân giải màn hình 640 x 200 pixels.
* Graphdriver = DETECT (hoặc 0) ứng với chế đồ màn hình đồ họa với độ phân giải cao nhất trong graphmode.
* Shape

  Description automatically generated with low confidenceChuỗi s là đường dẫn đến tệp egavga.bgi, nếu s rỗng máy tự cung cấp đường dẫn tìm đến tệp trên thư mục máy chủ.

Hình 2‑2: Màn hình đồ họa BGI

### Các thuật toán vẽ đường

### Thuật toán vẽ đường thẳng DDA (Digital Differential Analyzer)

Thuật toán vẽ đường thẳng DDA là thuật toán vẽ đường thẳng xác định các điểm ảnh được vẽ trên đoạn thẳng bằng cách làm tròn tọa độ ví trí các điểm tiếp theo mà đường thẳng đó đi qua.

Giả sử vẽ đường thẳng đi qua 2 điểm A(x1, y1) và B(x2, y2)

Ta có phương trình đường thẳng d có dạng: y = mx + b

Trong đó:

m =

b = y1 - mx1Chart

Description automatically generated

Hình 2‑3: Thuật toán DDA

Ban đầu ta có điểm (xi, yi), ta thấy tọa độ điểm xi lên một đơn vị trên mỗi điểm vẽ, chọn yi+1 là yi  hay yi+1 dựa vào phương trình đường thẳng d, khi đó yi+1 được tính như sau:

yi +1 = mxi +1 + b = m(xi+ 1) + b = mxi+ b + m

Diagram

Description automatically generatedTa có lưu đồ thuật toán như sau:

Hình 2‑4: Lưu đồ thuật toán DDA

* Bước 1: Tính dx = x2 – x1, dy = y2 – y1
* Bước 2: Nếu giá trị tuyệt đối của dx lớn hơn giá trị tuyệt đối của dy thì ta gán biến step = abs(dx). Ngược lại, ta gán biến step = (dy).
* Bước 3: Gán biến x\_inc = dx / step, gán biến y\_inc = dy / step, gán biến x = x1, gán biến y = y1. Sau đó, ta vẽ một điểm tại tọa độ x1 và y1.
* Bước 4: Kiểm tra nếu k >= step thì x = x + x\_inc, y = y + y\_inc, ta vẽ điểm với tọa độ x và y được làm tròn số lên. Vẽ tiếp tục các điểm tiếp theo cho đến khi k < step thì chương trình kết thúc.

### Thuật toán vẽ đường tròn Bresenham

Thuật toán vẽ đường tròn Bresenhem là thuật toán vẽ đường tròn dựa trên các điểm pixel được vẽ theo đường tròn với bán kính R.

Chart, radar chart

Description automatically generatedXác định 8 điểm đối xứng trên đường tròn

Hình 2‑5: Tám điểm đối xứng trên đường tròn

A picture containing text, clock, antenna

Description automatically generatedGiả sử xác định được điểm (x, y), ta xác định được tọa độ 7 điểm còn lại nhờ tính chất đối xứng trong đường tròn. Ta vẽ các cung đường tròn qua các điểm nhờ tính đối xứng qua trục tọa độ và hai đường phân giác để vẽ được đường tròn.

Hình 2‑6: Thuật toán Bresenham

Giả sử biết trước điểm (xi, yi) như trong hình trên. Ta xác định thêm hai điểm S1 và S2, khi đó khoảng cách d1 và d2 được tính dựa vào phương trình đường tròn. Khoảng cách d1 và d2 lần lượt là khoảng cách giữa phương trình đường tròn với S1 và S2. Ta kiểm tra nếu d1 >= d2 ta chọn điểm S2 làm điểm tiếp theo đi tính toán. Ngược lại, ta chọn điểm S1 làm điểm tiếp theo cần được tính toán.

Diagram

Description automatically generatedTa có lưu đồ thuật toán Bresenham như sau:

Hình 2‑7: Lưu đồ thuật toán Bresenham

* Bước 1: Gán x = 0, y = R, p = 3 - 2R
* Bước 2: Kiểm tra nếu x > y thì chương trình kết thúc, ngược lại nếu x <= y ta sang bước tiếp theo.
* Bước 3: Đặt pixel tại điểm (x, y). Kiểm tra điều kiện nếu p < 0 thì p được tính p = p + 4\*x + 6. Ngược lại, p = p + 4\*(x – y) + 10 và y giảm một đơn vị.
* Bước 4: Tăng x lên một đơn vị và lặp lại bước 2.

### Sự kiện bắt chuột

Thư viện Graphics.h cung cấp nhiều phương thức để thao thức với chuột và bàn phím máy tính. Hai phương thức int mousex(); và int mousey(); là hai phương thức đơn giản và phổ biến nhất.

Trong đó

* int mousex(); (Lấy tọa độ chuột theo tung độ)
* int mousey(); (Lấy tọa độ chuột theo hoành độ)

Thư viện nhận các sự kiện chuột trong màn hình đồ họa như sau:

Sự kiện chuột khi nhấn chuột trái

* WM\_LBMOUSEMOVE: Sự kiện di chuyển chuột.
* WM\_LBUTTONDOWN: Sự kiện khi nhấn chuột trái.
* WM\_LBBUTTONDBLCLK: Sự kiện khi nhấn hai lần chuột trái.
* WM\_LBUTTONUP: Sự kiện thả khi nhấn chuột trái.

Sự kiện chuột khi nhấn chuột giữa

* WM\_MBUTTONDOWN: Sự kiện khi nhấn chuột giữa.
* WM\_MBBUTTONDBLCLK: Sự kiện khi nhấn hai lần chuột giữa.
* WM\_MBUTTONUP: Sự kiện thả khi nhấn chuột giữa.

Sự kiện chuột khi nhấn chuột phải

* WM\_RBUTTONDOWN: Sự kiện khi nhấn chuột phải.
* WM\_RBBUTTONDBLCLK: Sự kiện khi nhấn hai lần chuột phải.
* WM\_RBUTTONUP: Sự kiện thả khi nhấn chuột phải.

Thư viện có hỗ trợ các hàm để bắt sự kiện chuột trên:

* bool ismouseclick(int kind);
* void getmouseclick(int kind, int& x, int& y);
* void clearmouseclick(int kind);
* void registermousehandler(int kind, void h(int, int));

### Phép biến đổi 2D cơ bản

#### Tịnh tiến

Phép tịnh tiến theo vectơ T(Tx, Ty) sẽ dời vị trí điểm P(x, y) đến vị trí mới Q(x’, y’) theo công thức:

x’ = x + Tx

Shape, rectangle

Description automatically generated with medium confidencey’ = y + Ty

Hình 2‑8: Phép tịnh tiến điểm P thành điểm Q

#### Xoay/Quay (Rotation):

* Diagram

  Description automatically generatedQuay quanh gốc tọa độ: quay điểm P(x, y) quanh gốc tọa độ 1 góc theta

Hình 2‑9: Quay quanh một góc theta

* Diagram

  Description automatically generatedQuay quanh 1 điểm bất kỳ

Hình 2‑10: Quay quanh một góc bất kỳ

## Vẽ đối tượng đồ họa

### Vẽ vịt trời

- Ý tưởng: vẽ vịt trời dựa trên đường tròn và thêm các chi tiết như : mắt, mỏ, bụng, cánh, chân.

- Hàm vẽ vịt trời sẽ nhận vào giá trị tâm của đường tròn vàn bán kính (độ to của vịt)

+ Vẽ thân: vẽ đường tròn dựa vào tâm và bán kính đã nhận.

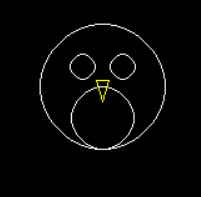
+ Vẽ mắt: từ tọa độ tâm đường tròn ta tính được vị trí tâm mắt trái và mắt phải sau đó tiến hành vẽ 2 đường tròn nhỏ.

Hình 2‑11: Vẽ thân và mắt vịt

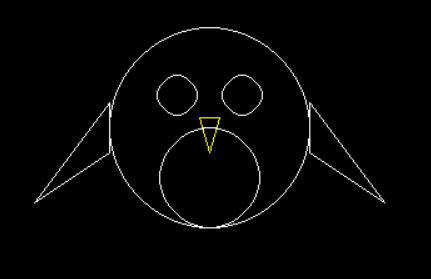
 + Vẽ bụng: từ tọa độ tâm đường tròn ta tính được vị trí tâm bụng bằng phân nữa bán kính và vẽ đường tròn nhỏ bằng ½ đường tròn thân

Hình 2‑12: Vẽ phần bụng vịt

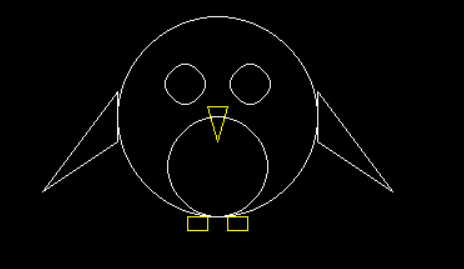
+ Vẽ mỏ : xác định 3 đỉnh của tam giác (mỏ) cách khoản đều với tâm đường tròn và tiến hành nối 3 đỉnh lại



Hình 2‑13: Vẽ miệng vịt

+ Vẽ cánh (trái và phải): từ vị trí điểm trái và phải của đường tròn tính được 3 điểm tạo nên cánh của vịt và tiến hành nối 3 đỉnh lại tạo nên cánh.

Hình 2‑14: Vẽ cánh vịt

+ Vẽ chân (trái và phải): từ vị trí dưới của đường tròn trừ đi 1 khoản nhỏ về 2 phía là có được tọa độ điểm đầu tiên của 2 chân và tiến hành xác định 4 đỉnh của chân rồi nối lại tạo thành hình chữ nhật.

Hình 2‑15: Vẽ chân vịt

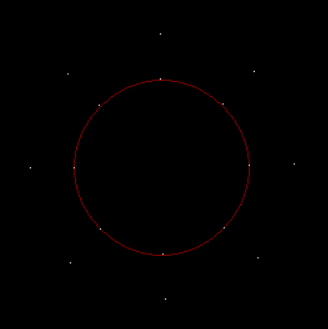
+ Thành phẩm sau khi tô màu

Hình 2‑16: Đối tượng vịt trời sau khi tô màu

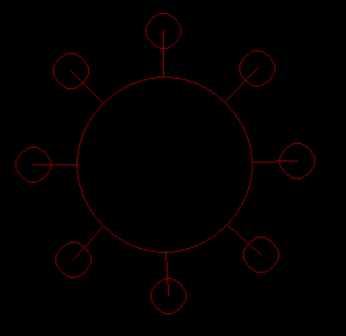
### Vẽ virus

- Ý tưởng: vẽ virus dựa trên đường tròn và thêm các gai protein (giống virus covid-19)

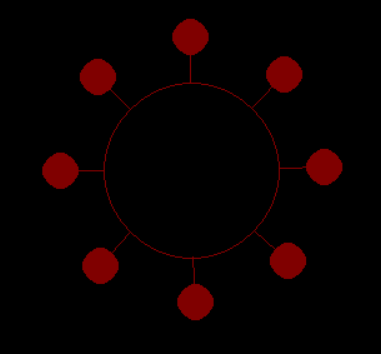
- Hàm vẽ vẽ virus sẽ nhận vào giá trị tâm của đường tròn vàn bán kính (độ to của virus)

 + Vẽ các đỉnh của gai protein: từ đỉnh dùng phương pháp xoay ra 8 điểm cách tâm bằng bán kính. Và xoay thêm 8 điểm nữa cách xa hơn bán kính của đường tròn ta được đỉnh của các gai

Hình 2‑17: Định hình các đỉnh gai

 + Vẽ các gai protein : nối 2 đỉnh của gai lại với nhau và ở đỉnh ngoài bắt đầu vẽ vòng tròn

Hình 2‑18: Vẽ các gai protein

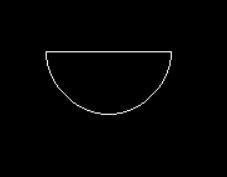
+ Sau khi tô màu

Hình 2‑19: Đối tượng virus khi tô màu

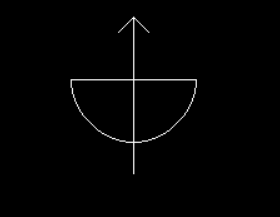
### Vẽ cung bắn

- Ý tưởng : vẽ cung bắn dựa trên thuật toán vẽ đường tròn

- Hàm vẽ vẽ virus sẽ nhận vào giá trị tâm của đường tròn vàn bán kính (độ to của cung)

 + Vẽ cung : vẽ nữa đường tròn và đường thẳng nằm ngan để tạo thành cung

Hình 2‑20: Vẽ nữa vòng tròn cung

+ Vẽ mũi tên: vẽ đường thẳng đướng để tạo trục mũi tên và tạo mũi tên bằng 2 đường thẳng nhỏ

Hình 2‑21: Đối tượng cung tên

## Thuật giải trò chơi

Các vấn đề cần xử lý trong bài:

* Giao diện menu người dùng: Gồm các nút bắt đầu chơi, thoát trò chơi và phổ biến luật chơi.
* Giao diện chơi:
* Xử lý vấn đề vịt trời bay và virus trong màn hình đồ họa
* Xử lý bắn vịt và tính điểm
* Xử lý viên đạn
* Giao diện kết thúc trò chơi:
* Hiện ra thông báo điểm cho người chơi
* Nút chơi lại
* Nút trở về lại màn hình menu bắt đầu

### Ý tưởng xử lý giao diện menu

Vẽ tên trò chơi bằng hàm outextxy()

Vẽ các hình chữ nhật tương ứng với các nút “Start”, “Rules” và “Exit”.

Sử dụng các sự kiện chuột, bằng cách cách định tọa độ được ấn định ở các nút trên sẽ gọi các hàm và giao diện người dùng tiếp theo tương ứng.

### Ý tưởng giao diện chơi

#### Xử lý vấn đề vịt bay và virus

Đối tượng vịt trời được xuất hiện ngẫu nhiên trên bằng hình đồ họa bằng hàm rand()

Sau khi vịt xuất hiện sẽ dựa vào giá trị bước nhảy để cập nhật tọa độ vịt mới và xóa vịt cũ. Đồng thời tạo lại vịt mới tạo cảm giác vịt đang di chuyển.

Khi vịt trời chạm đến giới hạn của cửa sổ đồ họa, sẽ cập nhật lại giá trị di chuyển để vịt chuyển động vào trong màn hình.

Tương tự với đối tượng virus, sẽ được xuất hiện ngẫu hiện trên màn hình và di chuyển bằng giá trị bước nhảy. Khi virus chạm đến giới hạn cửa sổ đồ họa sẽ di chuyển theo hướng khác.

#### Xử lý vấn đề bắn vịt và tính điểm

Kiểm tra vịt được bắn bằng cách xác định khoảng cách vị trí con vịt và con chuột. Khi trỏ chuột chọn đúng vào vị trí con vịt, con vịt sẽ hiện thông báo bắn trúng và cộng 10 điểm. Đồng thời xóa vị trí vịt và virus cũ, cập nhật lại vị trí đối tượng mới. Khi trỏ chuột chọn vào virus hoặc màn hình game thì sẽ mất một viên đạn.

Virus xuất hiện thêm dựa vào số điểm người chơi đạt được, điểm càng cao thì virus càng nhiều. Dựa vào mỗi móc điểm mà virus sẽ được tăng thêm. Tối đa thêm được 3 đối tượng virus và ít nhất 1 đối tượng virus tại thời điểm bắt đầu game.

Khi bắn hết đạn, chương trình sẽ chuyển đến giao diện kết thúc trò chơi.

### Ý tưởng giao diện kết thúc trò chơi

Thông báo điểm cho người chơi. Lấy biến điểm qua thủ tục chuyển kiểu số thành kiểu char \*textstring và xuất thông báo điểm bằng hàm outtextxy().

Khi người dùng nhấn nút “Replay”, chương trình trả về màn hình chơi game. Chọn nút trở lại giao diện menu.

## Hiện thực trò chơi

### Giao diện menu

Giao diện này gồm 3 nút điều khiển chính

* Start: Để bắt đầu chơi
* Exit: Thoát trò chơi
* A picture containing graphical user interface

  Description automatically generatedPhổ biến luật chơi

Hình 2‑22: Giao diện menu

### Giao diện chơi

Background pattern

Description automatically generated with medium confidenceMới bắt đầu chơi, có 2 đối tượng là con vịt với virus và 3 viên đạn. Khi bắn trúng con vịt sẽ hiện thông báo bắn trúng cộng 10 điểm và xóa đối tượng cũ, xuất hiện các đối tượng mới. Bắn trúng virus hoặc màn hình chơi sẽ bị mất một viên đạn

Hình 2‑23: Giao diện chơi ban đầu

A screenshot of a computer

Description automatically generated with medium confidenceKhi người chơi đạt được 50 điểm, xuất hiện thêm con virus.

Hình 2‑24: Giao diện chơi khi đạt trên 50 điểm

Background pattern

Description automatically generated with medium confidenceKhi người chơi đạt 200 điểm, trò chơi sẽ thêm một con virus.

Hình 2‑25: Giao diện chơi khi đạt trên 200 điểm

### Giao diện kết thúc trò chơi

Graphical user interface

Description automatically generatedKhi người chơi hết đạn. Trò chơi kết thúc, thông báo số điểm người dùng đạt được. Người chơi nhấn “Replay” để chơi lại, nhấn “Back to menu” để trở lại giao diện menu.

Hình 2‑26: Giao diện kết thúc trò chơi

# Kết luận

## Những kết quả đạt được

Qua đề tài này, chúng em xây dựng được trò chơi bắn vịt trời đơn giản bằng ngôn ngữ C++ với thư viện Graphics.h. Bên cạnh đó, chúng em biết được thêm nhiều kiến thức về môn học đồ họa máy tính này. Biết được nhiều hàm xử lý, cách triển khai các thuật toán. Chúng em biết vẽ những đối tượng với những đường nét riêng, xây dựng những họa tiết thêm sinh động và linh hoạt hơn. Tạo tiền đề để phát triển hơn về cách vẽ những đối tượng nâng cao và đời thật. Ngoài ra, chúng em hiểu và nắm bắt được ý tưởng xây dựng một trò chơi đơn giản nói chung và trò chơi bắn vịt trời này nói riêng. Nắm được cách phân bổ các hàm và tính năng cho hợp lý, rèn luyện khả năng suy nghĩ logic. Ngoài những kiến thức đồ họa và những kiến thức trong ngành công nghệ thông tin, chúng em phần nào còn được ôn lại cách bố cục, phân bổ bài báo cáo hợp lý. Từ đó, giúp chúng em trang bị thêm nhiều kỹ năng khi làm việc sau này.

## Ưu điểm của đề tài

* Màu sắc trong game khá hài hòa tạo cảm hứng cho người chơi.
* Luật chơi khá dễ, mang tính giải trí cao.
* Có các đối tượng virus tăng dần theo điểm thử thách người chơi nhiều hơn.

## Nhược điểm của đề tài

* Tại giao diện chơi, các đối tượng có hiệu ứng giật do mỗi lần di chuyển đều phải tăng tọa độ lên.
* Trò chơi khá đơn giản.
* Không có nhiều thử thách, thách đó người chơi nhiều.
* Các đối tượng đồ họa còn khá đơn giản. Thuật toán và code chưa được tối ưu.
* Càng chơi với số điểm càng cao, trò chơi càng giật, tạo cảm giác khó chịu. Đôi khi gây ra hiện tượng tràn stack gây đứng máy tạm thời phải tắt chương trình và chạy lại.

## Hướng phát triển trong tương lai

* Nhập thông tin vào lưu điểm người chơi.
* Xếp hạng người chơi.
* Tăng nhiều thử thách độ khó trò cho trò chơi.
* Phát triển trò chơi với đồ họa 3D.
* Giảm độ giật lag, các đối tượng game di chuyển mịn hơn.

**Tài liệu tham khảo**

[1] “Huy Super’s Blog: Giới thiệu về thư viện graphics.h”. [Trực tuyến]. Địa chỉ:

<http://huysuper.blogspot.com/2017/06/gioi-thieu-ve-thu-vien-graphicsh-va-mot.html> (truy cập tháng 12 28, 2021).

[2] “Nouvelle page 4”. [Trực tuyến]. Địa chỉ: <http://vietsciences2.free.fr/giaokhoa/tinhoc/dohoatrongturbopascal.htm> (truy cập tháng 01 02, 2022).

[3] “[OU]\_ComputerGraphics\_Chapter2.pdf”. Trực tuyến. Địa chỉ: <https://lms.ou.edu.vn/211/pluginfile.php/99708/mod_resource/content/2/%5BOU%5D_ComputerGraphics_Chapter2.pdf> (truy cập tháng 01 02, 2022).

[4] “Giáo trình kỹ thuật đồ họa: Giới thiệu thuật toán vẽ và tô các đường cơ bản - VOER”. [Trực tuyến]. Địa chỉ: <https://voer.edu.vn/c/giao-trinh-ky-thuat-do-hoa/6f8cfe5f> (truy cập tháng 01 03, 2022).

[5] “Xử Lý Sự Kiện Chuột và Phím với Borland Graphics Interface - BGI — Modern C++”. [Trực tuyến]. Địa chỉ: <https://voer.edu.vn/c/giao-trinh-ky-thuat-do-hoa/6f8cfe5f> (truy cập tháng 01, 05, 2022).