TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

---------------

BÀI TẬP LỚN MÔN: ĐỒ HỌA MÁY TÍNH

**TÊN ĐỀ TÀI**

**XÂY DỰNG TRÒ CHƠI ĐÀO VÀNG CỔ ĐIỂN**

**BÁO CÁO BÀI TẬP LỚN**

Ngành: Công Nghệ Thông Tin

Lớp: CS81

Sinh viên thực hiện:

Trần Lê Phương 1451050142

Trần Phương Nam 1751010094

Ngô Tuấn Linh 1851050080

Nguyễn Thị Diễm My 1851050091

Giáo viên giảng dạy:

ThS. Võ Thị Hồng Tuyết

Thành phố Hồ Chí Minh – Tháng 4 năm 2021

**LỜI MỞ ĐẦU**

Ngày nay, ngành công nghiệp Game trên trên đà phát triển rất mạnh mẽ. Game không chỉ là phương tiện giải trí cho người sử dụng mà còn là nguồn sáng tạo không ngừng từ nhà sản xuất. Từ đó thôi thúc rất nhiều Game hay và hấp dẫn ra đời. Thế nhưng để có thể cho ra đời một sản phẩm Game bạn sẽ phải thực hiện bằng một ngôn ngữ lập trình Game.

Sẽ không có gì đáng ngạc nhiên khi thấy C++ là ngôn ngữ lập trình Game phổ biến bởi khả năng kiểm soát các thành phần hệ thống cấp thấp đã góp phần kéo dài thời gian chạy chương trình – một yếu tố rất cần thiết trong lập trình Game.

Từ xu hướng phát triển và sáng tạo không ngừng, bài tập lớn này sẽ nghiên cứu để xây dựng trò chơi Đào vàng cổ điển bằng ngôn ngữ C++ với các hình vẽ đều vẽ từ graphics.h

***Mục lục***

[**CHƯƠNG 1: GIỚI THIỆU** 1](#_Toc69067928)

[**1.1** **Giới thiệu đề tài** 1](#_Toc69067929)

[**1.2** **Mục tiêu đề tài** 3](#_Toc69067930)

[**1.3** **Giới hạn đề tài** 4](#_Toc69067931)

[**1.4** **Bố cục báo cáo** 5](#_Toc69067932)

[**CHƯƠNG 2: TÊN ĐỀ TÀI** 6](#_Toc69067933)

[**2.1** **Các kiến thức cơ bản đi kèm** 6](#_Toc69067934)

[**2.1.1 Tổng quan về đồ họa máy tính** 6](#_Toc69067935)

[**2.1.2 Các giải thuật đồ họa cơ bản** 8](#_Toc69067936)

[**2.1.3 Phép biến đổi 2D cơ bản** 13](#_Toc69067937)

[**2.2** **Thuật giải đề xuất** 15](#_Toc69067938)

[**2.2.1 Vẽ đường thẳng** 15](#_Toc69067939)

[**2.2.2 Vẽ đường tròn** 18](#_Toc69067940)

[**2.2.3 Vẽ elip** 22](#_Toc69067941)

[**2.2.4 Vẽ đường cong** 23](#_Toc69067942)

[**2.2.5 Tô màu theo điểm** 24](#_Toc69067943)

[**2.3 Kết quả** 24](#_Toc69067944)

[**2.3.1 Màn hình menu Game** 24](#_Toc69067945)

[**2.3.2 Màn hình Game chính** 25](#_Toc69067946)

[**CHƯƠNG 3: KẾT LUẬN** 29](#_Toc69067947)

[**3.1** **Những kết quả đạt được** 29](#_Toc69067948)

[**3.2** **Ưu điểm của đề tài** 31](#_Toc69067949)

[**3.3** **Nhược điểm của đề tài** 32](#_Toc69067950)

[**3.4** **Hướng dẫn phát triển trong tương lai** 32](#_Toc69067951)

[**Tài liệu tham khảo** 33](#_Toc69067952)

**Danh mục hình ảnh**

[Hình 1‑0‑1 Ngôn ngữ C++ 1](#_Toc69067970)

[Hình 1‑0‑2 Nhân vật ông thợ mỏ 2](#_Toc69067971)

[Hình 1‑0‑3 Đào vàng cổ điển 3](#_Toc69067972)

[Hình 1‑0‑4 Visual Studio 2010 5](#_Toc69067973)

[Hình 2‑0‑1 Hình học 6](file:///D:\DHMT\DeTai6_TLPhuong_TPNam_NTLinh_NTDMy.docx#_Toc69067974)

[Hình 2‑0‑2 Hệ trục tọa độ bàn tay phải 7](#_Toc69067975)

[Hình 2‑0‑3 Hệ trục tọa độ bàn tay trái 8](#_Toc69067976)

[Hình 2‑0‑4 Điểm pixel 9](#_Toc69067977)

[Hình 2‑0‑5 Đoạn thẳng 10](#_Toc69067978)

[Hình 2‑0‑6 Đường gấp khúc 10](#_Toc69067979)

[Hình 2‑0‑7 Vùng tô 11](#_Toc69067980)

[Hình 2‑0‑8 Vector Bitmap 12](#_Toc69067981)

[Hình 2‑0‑9 Graphics.h 13](#_Toc69067982)

[Hình 2‑0‑10 Quay quanh gốc tọa độ 14](#_Toc69067983)

[Hình 2‑0‑11 Quay quanh 1 điểm bất kỳ 15](#_Toc69067984)

[Hình 2‑0‑12 DDA 16](#_Toc69067985)

[Hình 2‑0‑13 Lưu đồ DDA 17](#_Toc69067986)

[Hình 2‑0‑14 Đường tròn MidPoint 18](#_Toc69067987)

[Hình 2‑0‑15 Xác định 8 điểm 19](#_Toc69067988)

[Hình 2‑0‑16 Lưu đồ MidPoint 20](#_Toc69067989)

[Hình 2‑0‑17 Đường tròn Bresenham 21](#_Toc69067990)

[Hình 2‑0‑18 Elip MidPoint 22](#_Toc69067991)

[Hình 2‑0‑19 Lưu đồ elip MidPoint 23](#_Toc69067992)

[Hình 2‑0‑20 Kiểm tra góc thuộc đa giác 24](#_Toc69067993)

[Hình 2‑0‑21 Vẽ mắt, mũi 25](#_Toc69067994)

[Hình 2‑0‑22 Vẽ thân 26](#_Toc69067995)

[Hình 2‑0‑23 Tô màu đơn giản 26](#_Toc69067996)

[Hình 2‑0‑24 Vẽ dây móc 27](#_Toc69067997)

[Hình 2‑0‑25 Dây móc chuyển động quay 27](#_Toc69067998)

[Hình 2‑0‑26 Dây dài khi click chuột 27](#_Toc69067999)

[Hình 2‑0‑27 Vẽ 5 cục vàng 28](#_Toc69068000)

[Hình 2‑0‑28 Cục vàng tịnh tiến 28](#_Toc69068001)

[Hình 2-0‑29 Cục vàng biến mất 28](#_Toc69068002)

[Hình 3‑0‑1 Màn hình menu 29](#_Toc69068003)

[Hình 3-0‑2 Màn hình mục tiêu 29](#_Toc69068004)

[Hình 3-0‑3 Màn hình Play game 30](#_Toc69068005)

[Hình 3-0‑4 Mô tả việc xử lý móc 30](#_Toc69068006)

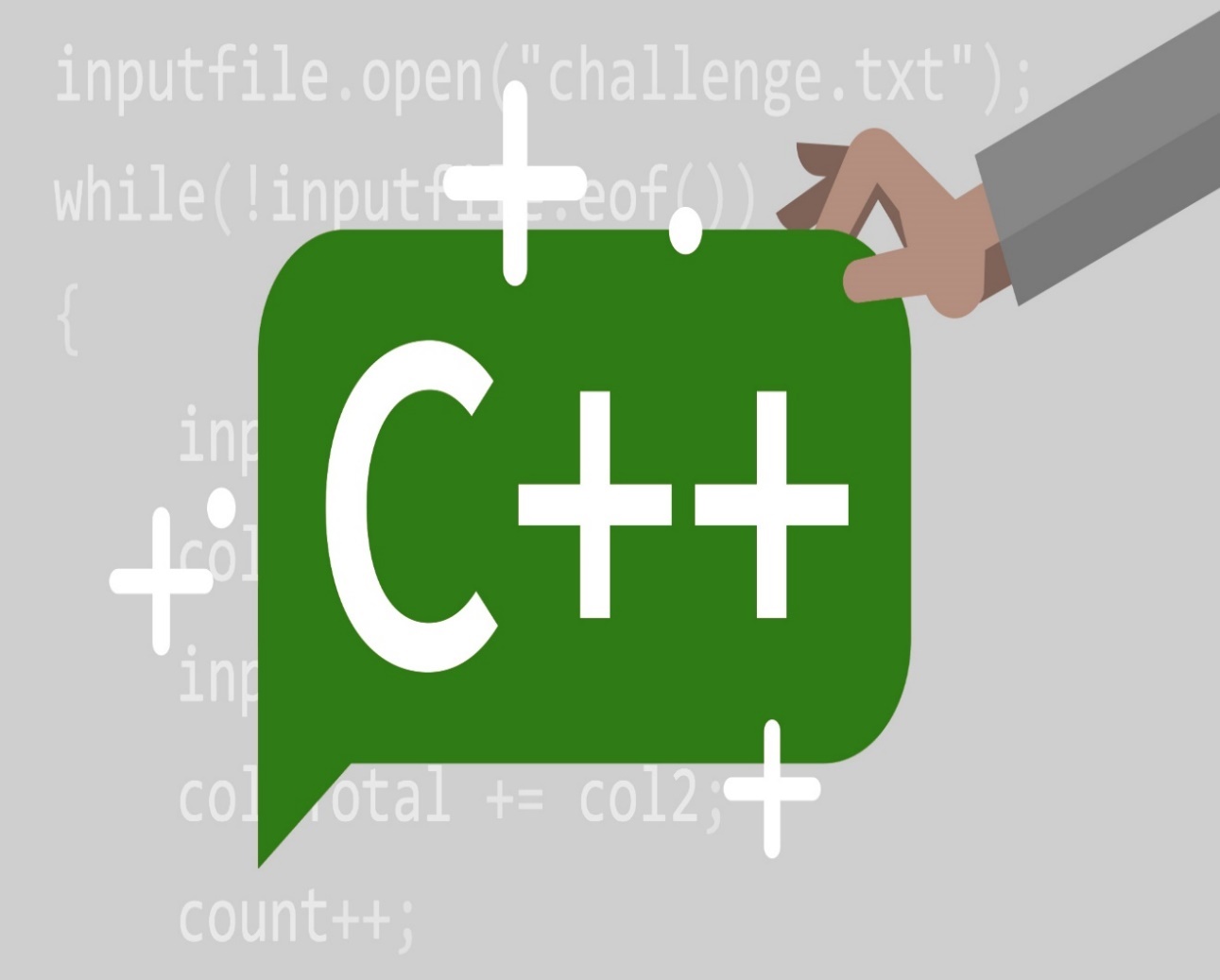
[Hình 3-0‑5 Màn hình hoàn thành mục tiêu 31](#_Toc69068007)

[Hình 3-0‑6 Màn hình Game Over 31](#_Toc69068008)

# **CHƯƠNG 1: GIỚI THIỆU**

## **Giới thiệu đề tài**

Xây dựng game Đào vàng kinh điển bằng ngôn ngữ C++ với các hình vẽ hoàn toàn từ graphics.h



Hình 1‑0‑1 Ngôn ngữ C++

Những thứ tạo nên một trò chơi đào vàng kinh điển sẽ có như: vàng, kim cương, thuốc nổ. Người dùng sẽ đóng vai một ông thợ mỏ, công việc của họ là đào thật nhiều vàng để kiếm thật nhiều tiền. Bên cạnh đó còn có những chướng ngại vật để cản trở tốc độ đào vàng như: đá, thuốc nổ …



Hình 1‑0‑2 Nhân vật ông thợ mỏ



Hình 1‑0‑3 Đào vàng cổ điển

Với ý tưởng muốn tiếp cận lĩnh vực thiết kế game – một trong số những nghề đang có nhu cầu nhân lực lớn, đem lại lợi nhuận khổng lồ nhưng lại còn khá mới đối với sinh viên chúng em thì trong đề tài này chúng em sẽ thực hiện xây dựng lại game Đào vàng kinh điển bằng ngôn ngữ C++ cùng các nét vẽ hoàn toàn bằng graphics.h sẽ cho ra những hình ảnh 2D đơn giản, mộc mạc.

## **Mục tiêu đề tài**

Cần đạt được những mục tiêu sau:

* Thiết kế màn hình Menu gồm có:

Bắt Đầu

Xem Điểm

Thoát

* Thiết kế màn hình Game gồm có:

Nhân vật ông thợ mỏ

Dụng cụ đào vàng

Các khối vàng

Các thuốc nổ

Khung điểm bên trên

Khung hiển thị thời gian

* Xử lý chuột với:

Các button ở menu

* Có tương tác với người dùng:

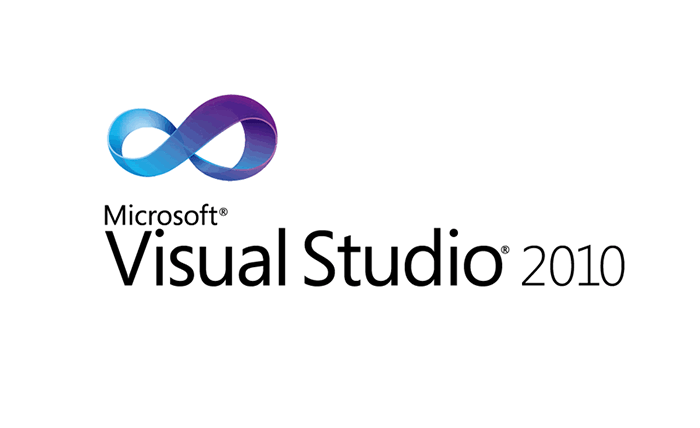
Xem Bảng Xếp Hạng

Lưu tên

* Có tính tiện dụng cho người dùng dễ sử dụng
* Giao diện đơn giản, đẹp mắt

## **Giới hạn đề tài**

* Sử dụng: Visual Studio 2010



Hình 1‑0‑4 Visual Studio 2010

* Các đối tượng xử lý bằng thuật toán vẽ với ngôn ngữ C++
* Giao diện đồ họa xử lý hoàn toàn bằng graphics.h
* Xử lý các chức năng cơ bản về đối tượng đồ họa
* Các thuật toán tọa độ, điểm, góc …

## **Bố cục báo cáo**

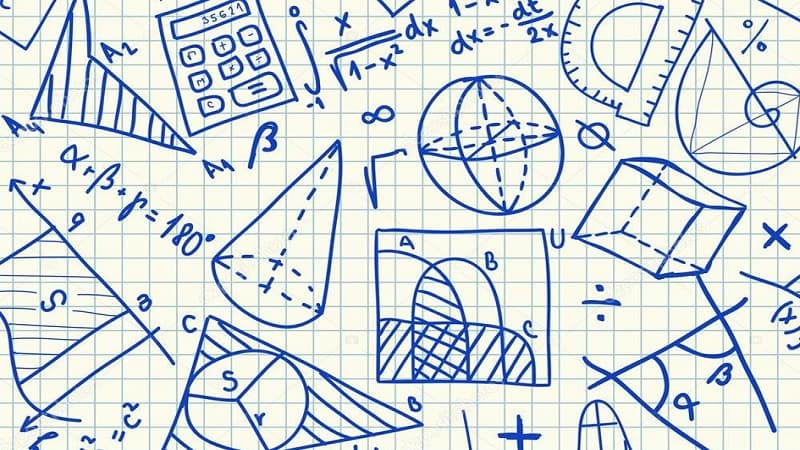
Nội dung báo cáo bài tập lớn gồm các thành phần sau:

* Mở đầu:
  + Chương 1. Giới thiệu
* Nội dung:
  + Chương 2. Tên đề tài
* Kết luận và hướng phát triển:
  + Chương 3. Kết luận
  + Tài liệu tham khảo

# **CHƯƠNG 2: TÊN ĐỀ TÀI**

## **Các kiến thức cơ bản đi kèm**

### **2.1.1 Tổng quan về đồ họa máy tính**

* Một hình ảnh: là một mảng (array)
* Pixel: tương ứng một điểm ảnh
* Khung đệm (frame buffer):
  + Nhiều điểm ảnh lưu trữ một phần của bộ nhớ
  + Là yếu tố cốt lõi của một hệ thống đồ họa
* Hình học: tổ chức các hình đa giác và đỉnh

Hình 2‑0‑1 Hình học

* Đỉnh: là một điểm.
* Bề mặt: tập hợp nhiều điểm
* Tọa độ Descartes:

Góc tọa độ O (0, 0)

0x là trục hoành

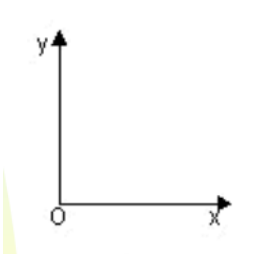
0y là trục tung

x là hoành độ

y là tung độ

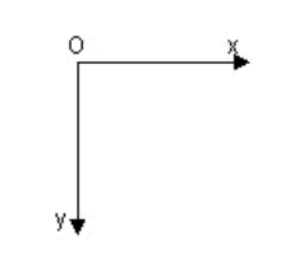
* Hệ trục tọa độ thiết bị:

Hệ trục tọa độ bàn tay phải



Hình 2‑0‑2 Hệ trục tọa độ bàn tay phải

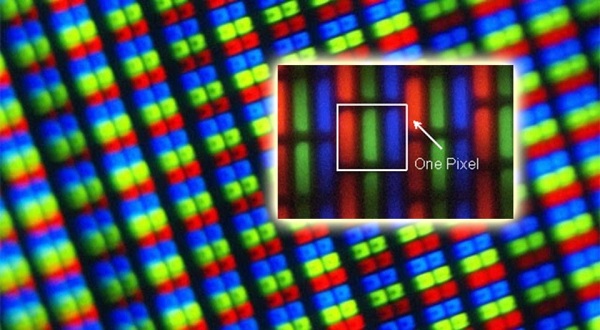
Hệ trục tọa độ bàn tay trái



Hình 2‑0‑3 Hệ trục tọa độ bàn tay trái

### **2.1.2 Các giải thuật đồ họa cơ bản**

* Điểm (pixel):



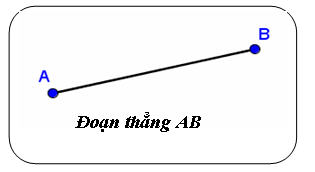
Hình 2‑0‑4 Điểm pixel

Là một thành phần cơ bản của hệ tọa độ

Được xác định bởi x và y

Có thuộc tính màu sắc

* Đoạn thẳng:



Hình 2‑0‑5 Đoạn thẳng

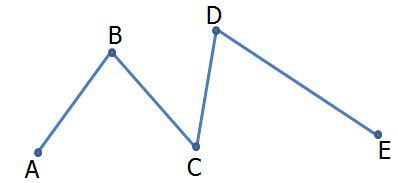
Được xác định bởi hai điểm có tọa độ (x1, y1) và (x2, y2)

Phương trình đường thẳng

Phương trình đoạn thẳng

Các thuộc tính: kiểu nét vẽ, độ dày, màu sắc

* Đường gấp khúc:



Hình 2‑0‑6 Đường gấp khúc

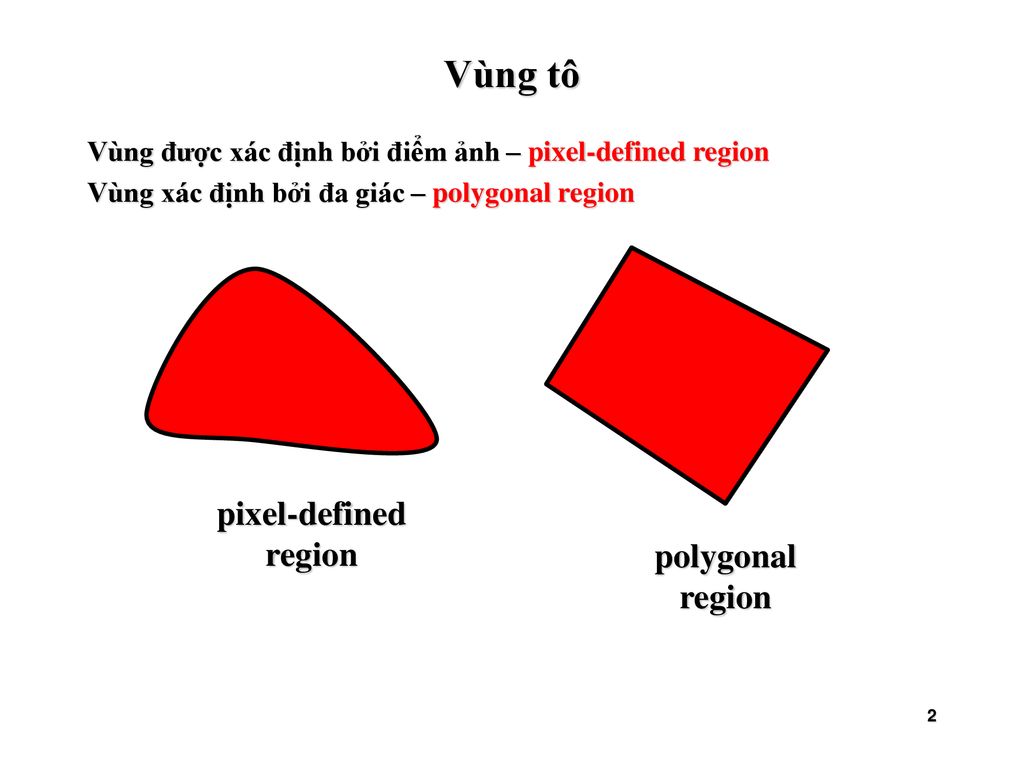
Tập hợp nhiều đoạn thẳng

Đường gấp khúc có thể kín hoặc hở

Điểm đầu và điểm cuối đường gấp khúc trùng nhau gọi là đa giác

Thuộc tính các đoạn thẳng trong đường gấp khúc là giống nhau

* Vùng tô:



Hình 2‑0‑7 Vùng tô

Gồm đường biên và vùng bên trong

Chỉ tô được màu cho đa giác

Thuộc tính: đường biên có thuộc tính của đoạn thẳng và vùng bên Trong có màu mẫu tô

* Ký tự, chuỗi:

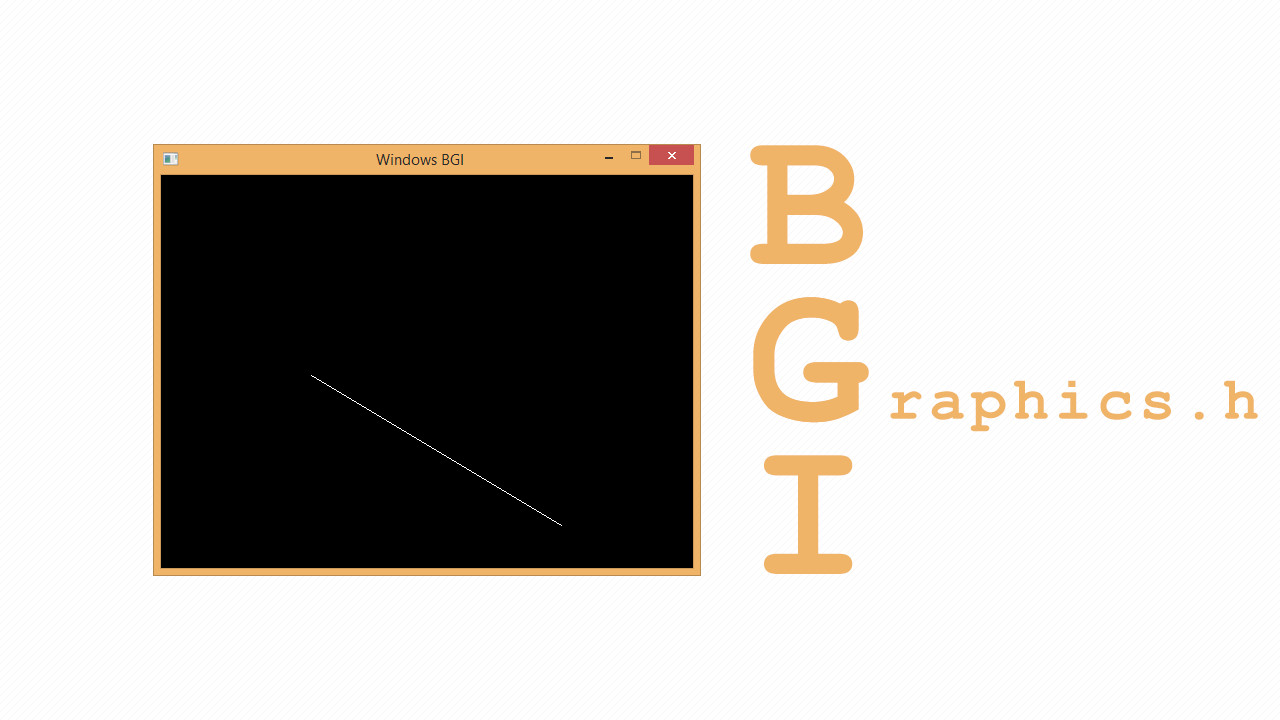


Hình 2‑0‑8 Vector Bitmap

Dạng bitmap và vector của font chữ

Thuộc tính: màu sắc, font, kích thước, lề, cách hiển thị, hướng

* graphics.h:



Hình 2‑0‑9 Graphics.h

Borland Graphics Interface (BGI)

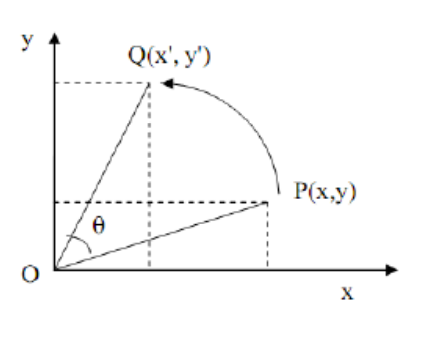
Thư viện đồ họa rất phổ biến DOS, các máy tính chạy Window đời đầu

Cung cấp cho người dùng hai file: graphics.h, graphics.lib

Đi kèm với IDE Borland C++ 3.1

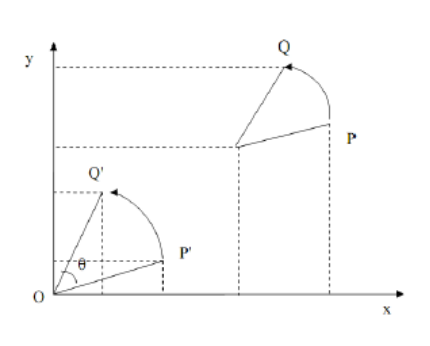
### **2.1.3 Phép biến đổi 2D cơ bản**

* + Xoay/Quay (Rotation):
    - Quay quanh gốc tọa độ: quay điểm P(x, y) quanh gốc tọa độ 1 góc theta



Hình 2‑0‑10 Quay quanh gốc tọa độ

* Quay quanh 1 điểm bất kỳ

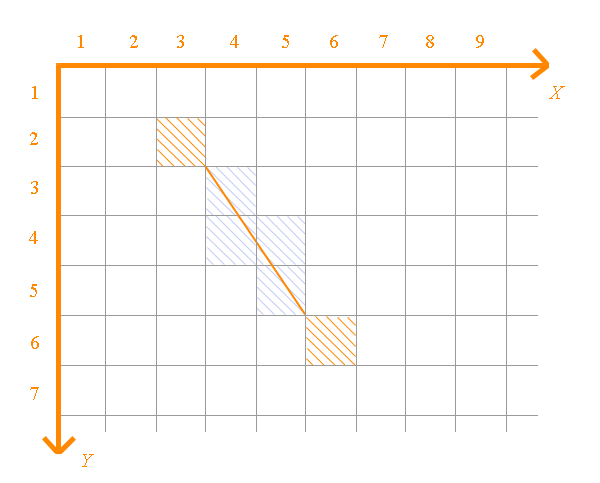


Hình 2‑0‑11 Quay quanh 1 điểm bất kỳ

## **Thuật giải đề xuất**

### **2.2.1 Vẽ đường thẳng**

* DDA (Digital Differential Analyzer) thuật toán tăng dần:



Hình 2‑0‑12 DDA

Ý tưởng:

Giả sử vẽ được xi, yi

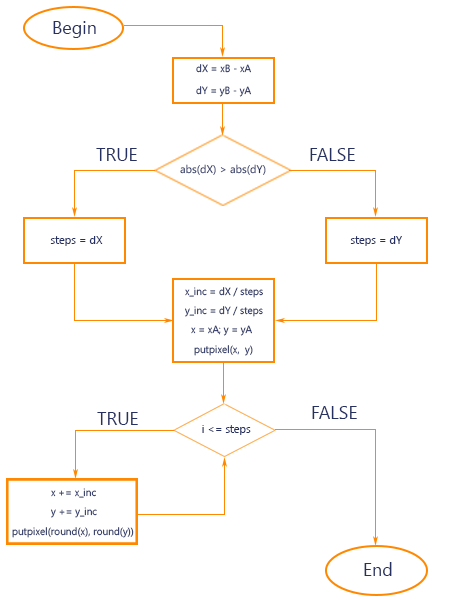
Chọn yi+1 là yi hay yi+1 dựa vào phương trình đường thẳng:

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

yi + 1 = mxi + b + m

yi + 1 = yi + m

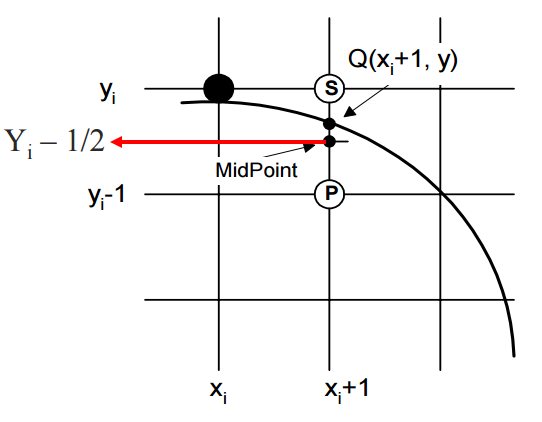
Lưu đồ:



Hình 2‑0‑13 Lưu đồ DDA

### **2.2.2 Vẽ đường tròn**

* Ý tưởng:
  + Cho x = 0, 1, 2 …, int (R \* sqrt(2) / 2) với R > 1
  + Tại mỗi giá trị x, tính int(y = sqrt(pow(R, 2) – pow(x, 2))
  + Vẽ điểm (x, y) cùng 7 điểm đối xứng của nó
* Vẽ đường tròn bằng thuật toán MidPoint:

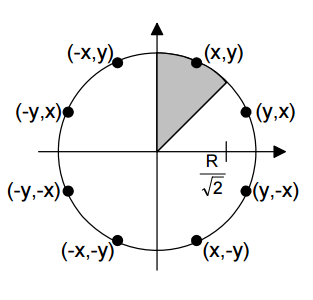


Hình 2‑0‑14 Đường tròn MidPoint

* Áp dụng vẽ 1/8 cung đường tròn và lấy đối xứng, khi đó:

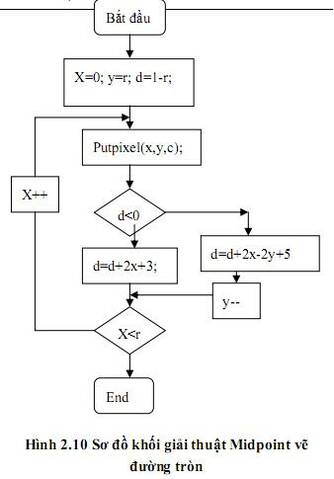
0 <= x <= R \* sqrt(2) / 2

R \* sqrt(2) / 2 <= y <= R



Hình 2‑0‑15 Xác định 8 điểm

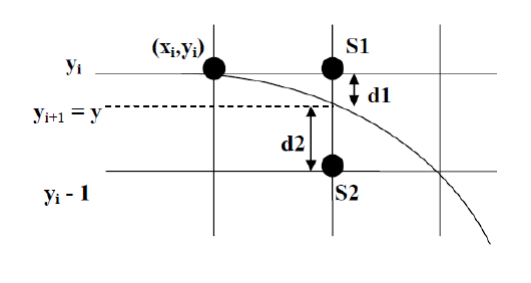
Lưu đồ:



Hình 2‑0‑16 Lưu đồ MidPoint

* Vẽ đường tròn bằng thuật toán Bresenham:

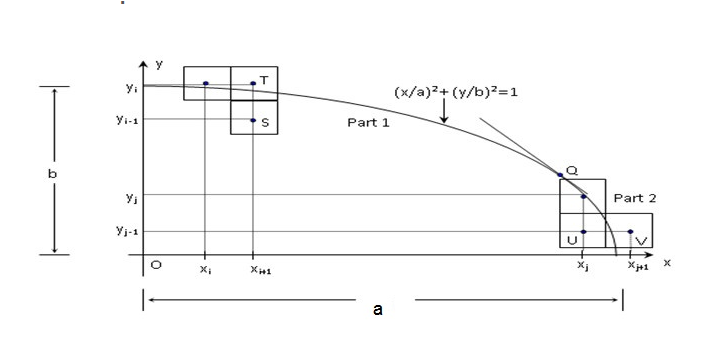
Các vị trí ứng với tọa độ nguyên nằm trên đường tròn có thể tính được bằng cách xác định 1 trong 2 pixel gần nhất với đường tròn thực như trong mỗi bước



Hình 2‑0‑17 Đường tròn Bresenham

### **2.2.3 Vẽ elip**

* Vẽ elip bằng thuật toán MidPoint:

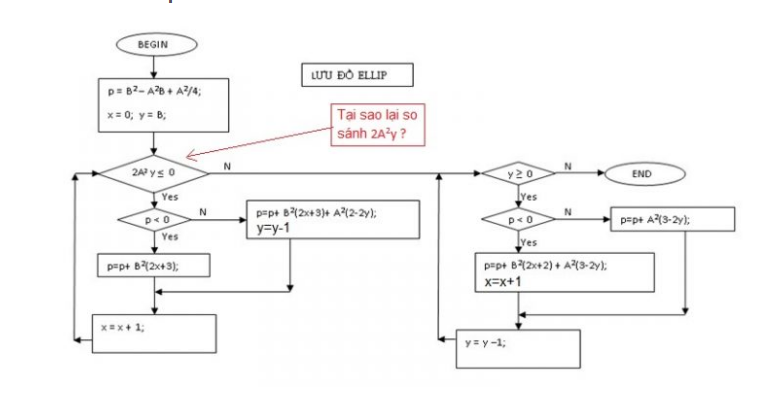


Hình 2‑0‑18 Elip MidPoint

Phương trình của đường elip:

f(x,y) = b^2 \* x^2 + a^2 \* y^2 – a^2 \* b^2

Lưu đồ:



Hình 2‑0‑19 Lưu đồ elip MidPoint

Tại sao lại so sánh 2A^2 \* y? => Chúng ta chia elip làm 2 phần để vẽ và lấy điểm Q có hệ số góc của tiếp tuyến với elip bằng -1 làm giao điểm. Theo công thức hệ số đã nêu ở trên: dx/dy = fx/fy = (2b^2 \*x) / (2a^2 \* y). Tại điểm đầu tiên có tọa độ (0, b), chúng ta thay x = 0, y = b vào cong thức trên được hệ số góc = 0/ (2a^2 \* b) lớn hơn -1. Nên ta chọn vẽ theo nhánh 1. Như vậy việc so sánh 2A^2 \* y < 0 là để chọn nhánh vẽ elip thôi.

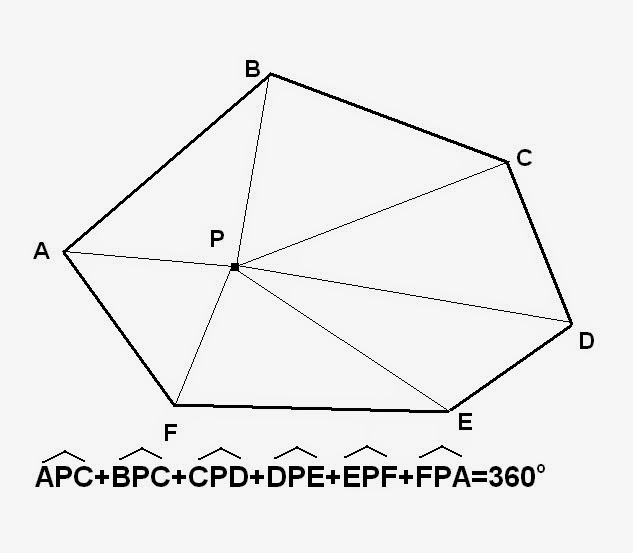
### **2.2.4 Vẽ đường cong**

* Vẽ đường cong theo thuật toán MidPoint:

Áp dụng từ vẽ đường tròn bằng thuật toán MidPoint những chỉ vẽ ¼ đường tròn để tạo thành đường cong.

### **2.2.5 Tô màu theo điểm**

* Kiểm tra một điểm có thuộc đa giác:



Hình 2‑0‑20 Kiểm tra góc thuộc đa giác

Kiểm tra góc:

Từ điểm cần xét nối với các đỉnh của đa giác tạo thành các góc theo thứ tự ngược chiều kim đồng hồ

Số đo các góc dương hoặc âm

Tính tổng số đo các góc. Nếu bằng 0 thì điểm cần xét nằm ngoài đa giác. Nếu bằng 360 độ thì điểm cần xét nằm trong đa giác

## **2.3 Kết quả**

### **2.3.1 Màn hình menu Game**

* Dùng hàm LineDDA để vẽ thành 1 khung (button) chứa label “Bat Dau”
* Dùng setcolor để đặt màu
* Dùng settextstyle để đặt font, hướng và cỡ chữ
* Dùng outtextxy để hiện các chữ “Bat Dau”, “Xem Diem” và “Thoat” tại các vị trí cố định trong menu trên màn hình BGI
* Dùng sự kiện chuột: khi click chuột trái vào button “Bat Dau” để chuyển sang màn hình game chính:

+ Chọn loại sự kiện chuột WM\_LBUTTONDOWN, WM\_LBUTTONUP

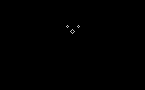
+ Sử dụng ismouseclick trả về true khi có sự kiện click chuột trái xuống, sau đó gọi getmouseclick để lấy tọa độ tại điểm sau khi thả chuột, nếu nằm trong phạm vi button “Bat Dau” thì gọi hàm cleardevice(); để toàn bộ màn hình đồ họa menu, sau đó gọi màn hình game chính đã thiết kế bên trong, sau đó dùng clearmouseclick hủy các sự kiện chuột đã gọi.

+ Tương tự cho “Xem Diem” và “Thoat”

### **2.3.2 Màn hình Game chính**

#### **2.3.2.1 Ông thợ mỏ**

* + - Tạo 1 hàm init chứa 1 mảng gồm nhiều phần tử, sau đó nhập vị trí cho các điểm
    - Tạo 1 hàm vẽ nhân vật ông thợ mỏ
    - Sử dụng thuật toán MidPoint để đường tròn làm 2 mắt, mũi cho nhân vật



Hình 2‑0‑21 Vẽ mắt, mũi

* + - Sử dụng lineDDA để nối các Point tạo thành hình nhân vật bằng các đoạn thẳng



Hình 2‑0‑22 Vẽ thân

* + - Tô màu nhân vật: tô đơn giản vài chi tiết như nón, thùng kéo, quần… theo giải thuật tô màu theo điểm bằng cách:
      * Tạo hàm tính khoảng cách 2 điểm
      * Tạo hàm tính 1 góc tam giác khi biết 3 cạnh
      * Tạo hàm kiểm tra 1 điểm có thuộc đa giác đó hay không
      * Tạo hàm tô màu theo điểm



Hình 2‑0‑23 Tô màu đơn giản

#### **2.3.2.2 Dây câu móc vàng**

* + - Vẽ dây câu móc vàng:
      * Vẽ sợi dây bằng lineDDA
      * Vẽ móc dây theo thuật giải MidPoint thành đường cong



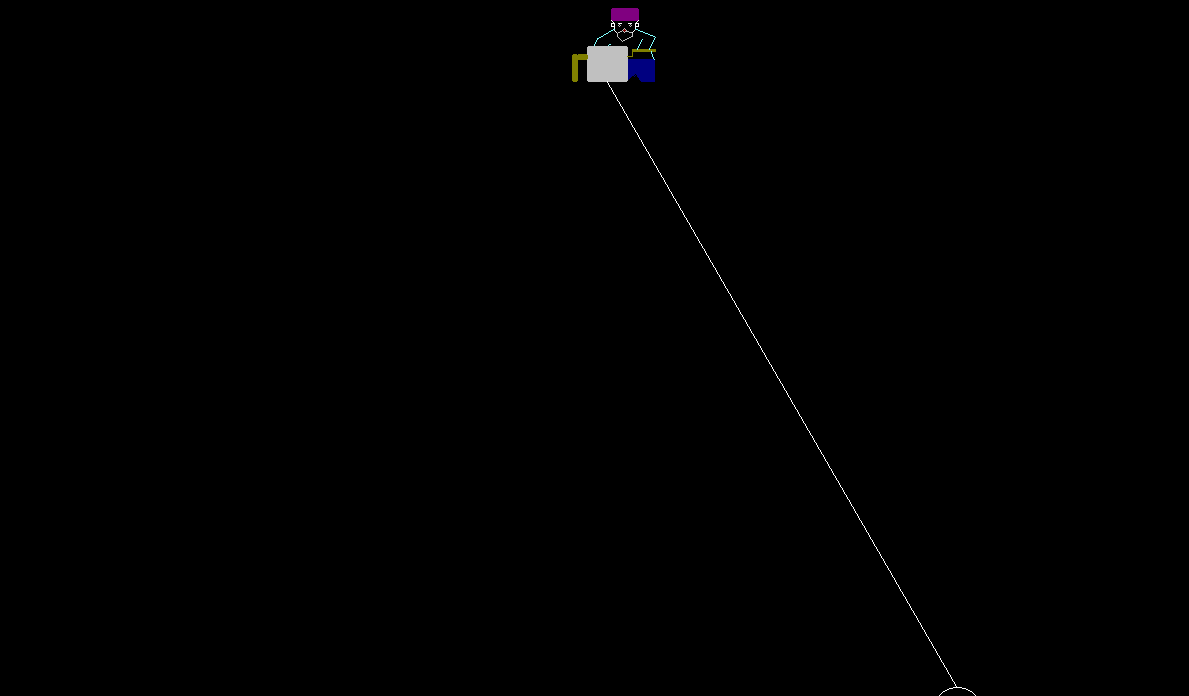
Hình 2‑0‑24 Vẽ dây móc

* + - Xử lý chuyển động dây:



Hình 2‑0‑25 Dây móc chuyển động quay

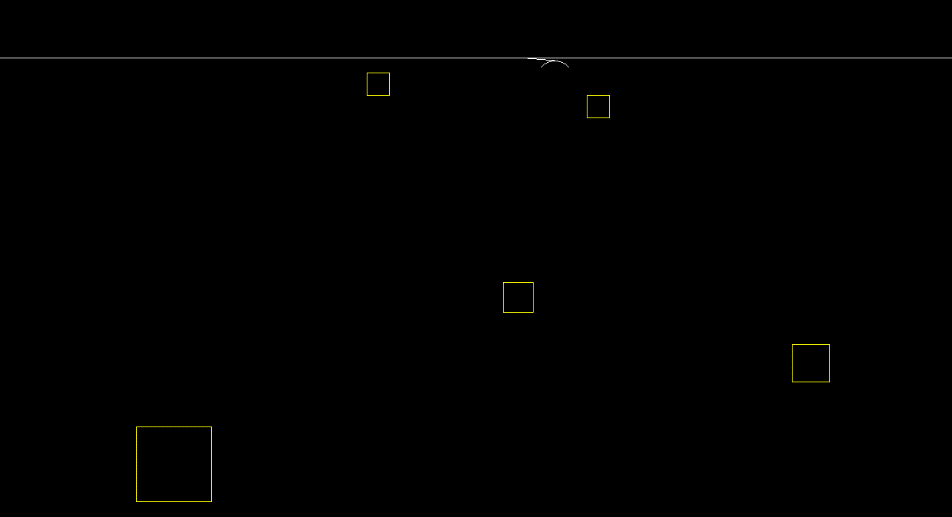
* + - Xử lý dây khi click chuột:



Hình 2‑0‑26 Dây dài khi click chuột

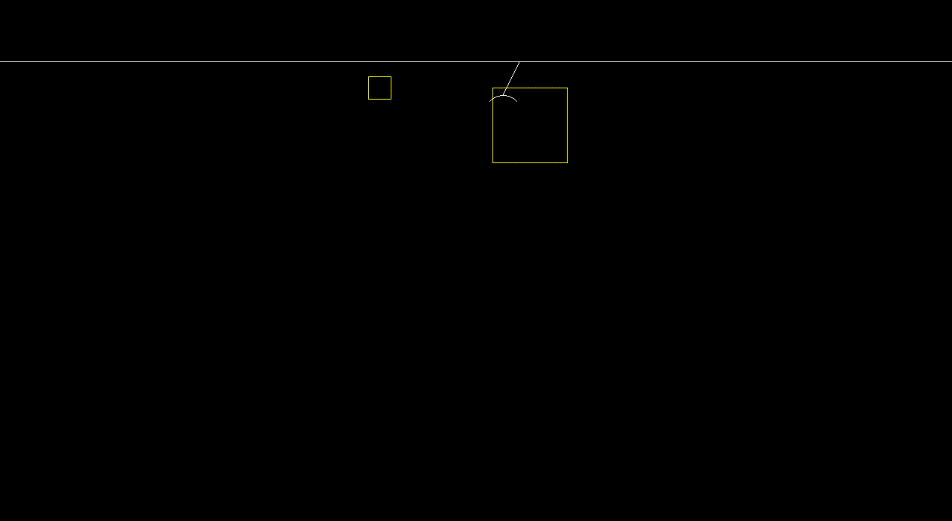
#### **2.3.2.3 Vàng**

* + - Số lượng: 5
    - Vị trí: phân tán phía dưới đường ranh giới



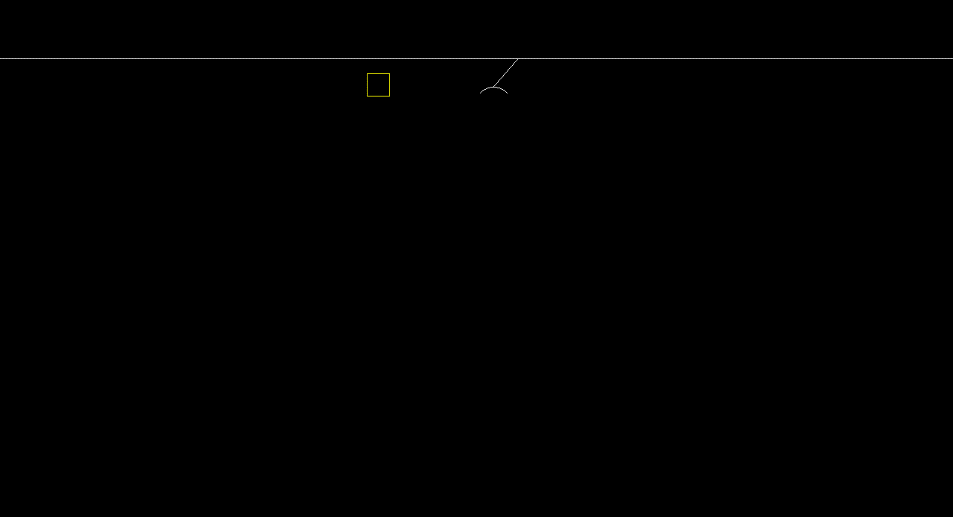
Hình 2‑0‑27 Vẽ 5 cục vàng

* + - Khi chạy dây móc câu đụng trúng cục vàng nào thì cục đó sẽ tịnh tiến đến gần ranh giới



Hình 2‑0‑28 Cục vàng tịnh tiến

* + - Cục vàng biến mất sau khi tịnh tiến



Hình 2-0‑29 Cục vàng biến mất

# **CHƯƠNG 3: KẾT LUẬN**

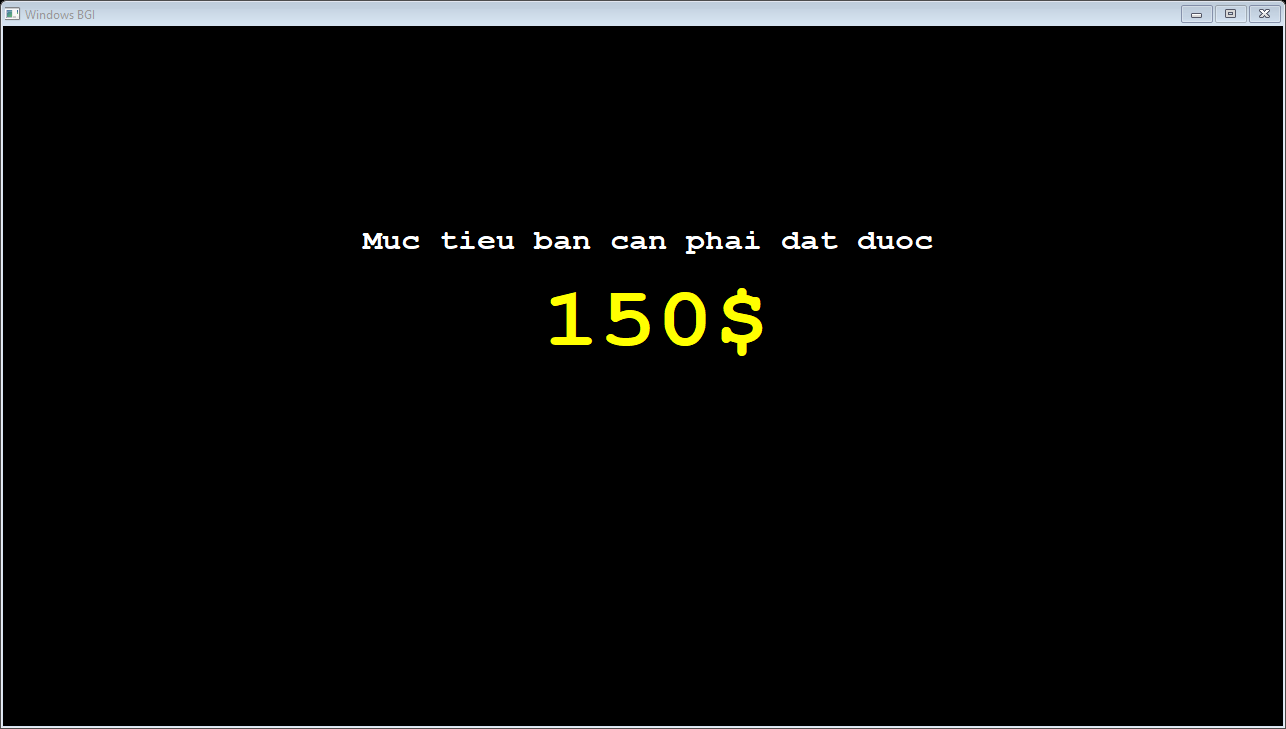
## **Những kết quả đạt được**

* Khi Run chương trình sẽ hiện ra màn hình BGI có kích thước 1280x700
* Đầu tiên, xuất hiện menu chương trình gồm button “Bat Dau”, “Xem Diem”, “Thoat”



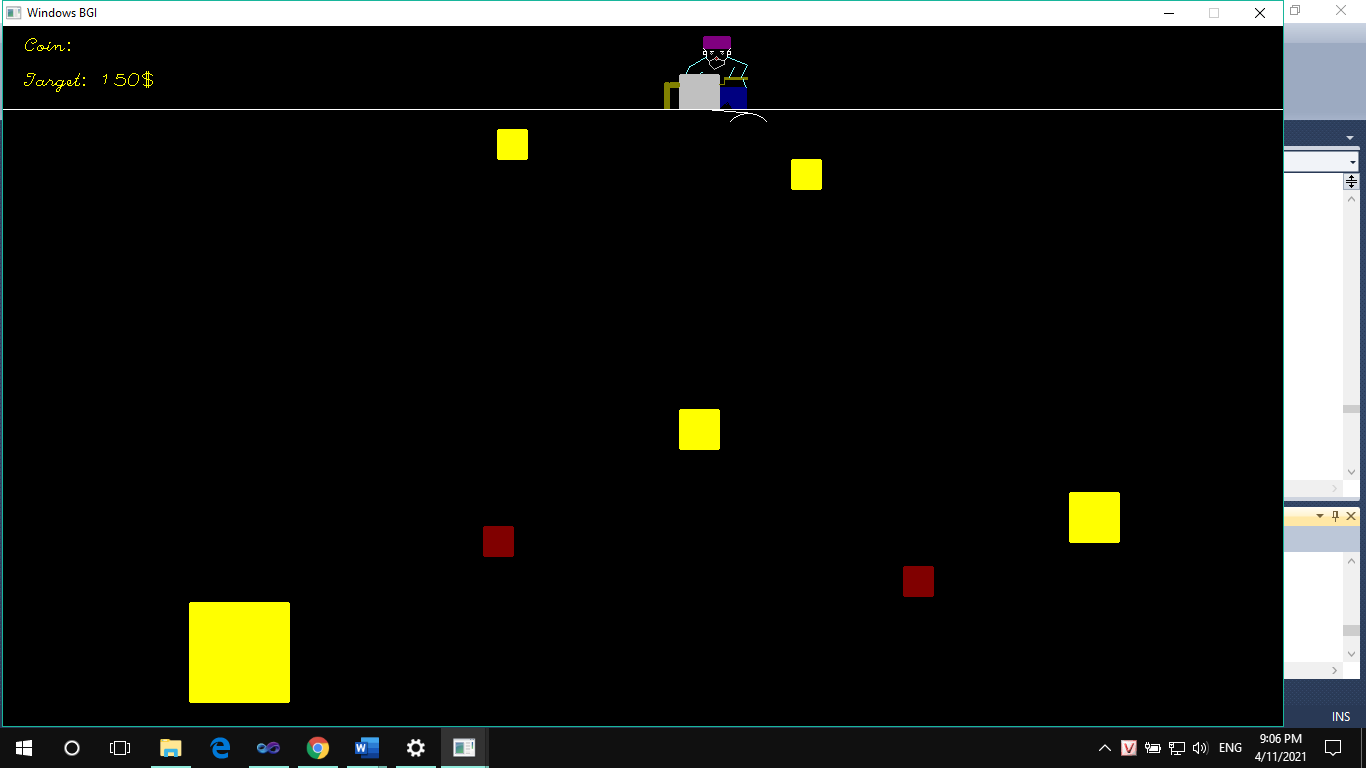
Hình 3‑0‑1 Màn hình menu

* Click chuột trái vào button “Bat Dau” sẽ mất màn hình menu và hiện màn hình mục tiêu trong 3s rồi chuyển sang màn hình Play game



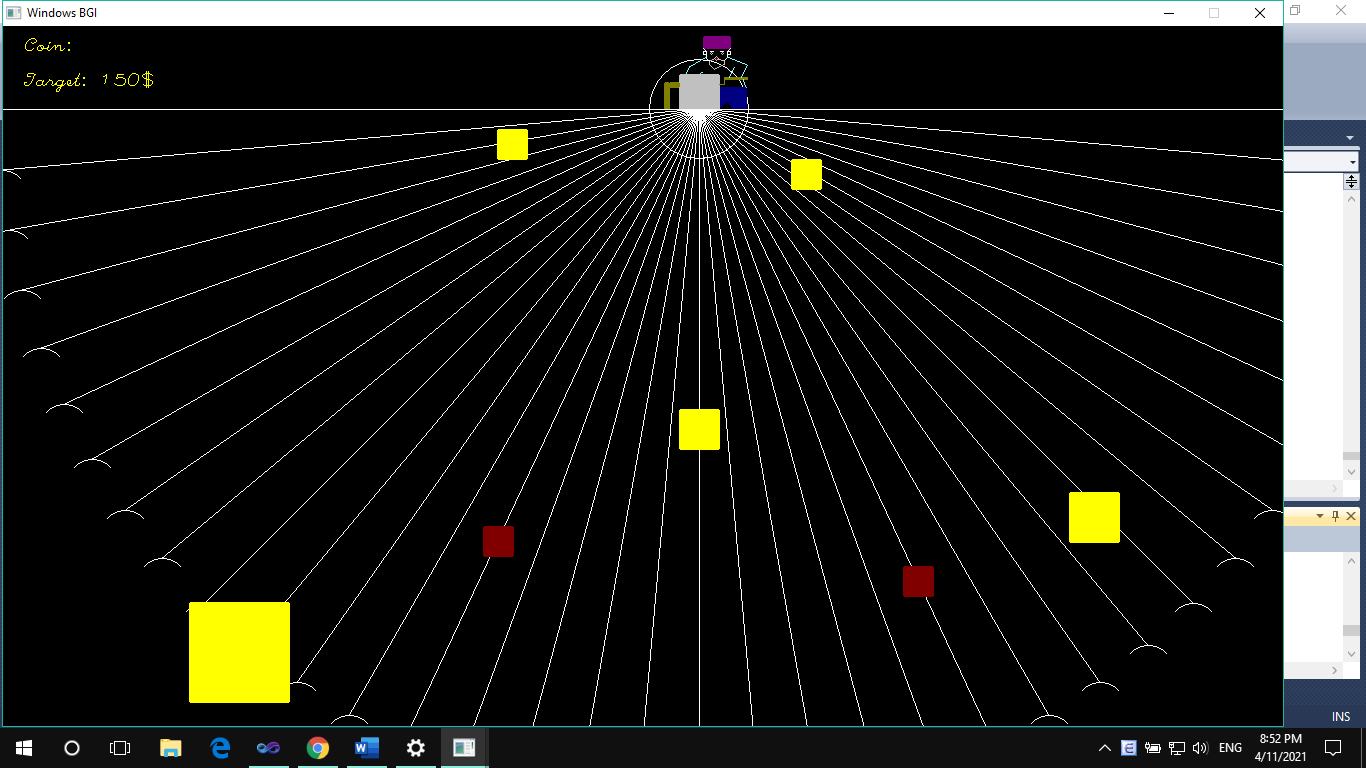
Hình 3-0‑2 Màn hình mục tiêu

* Màn play game chính có nhân vật ông thợ mỏ 2D đơn giản, 1 dây móc câu, 1 đường ranh giới, 5 cục vàng vuông vức nằm rải rác phía dưới và 2 mìn đỏ xuất hiện ở khoảng 24 giây đầu rồi biến mất và xuất hiện lại vào giây thứ 50 đến khi hết thời gian



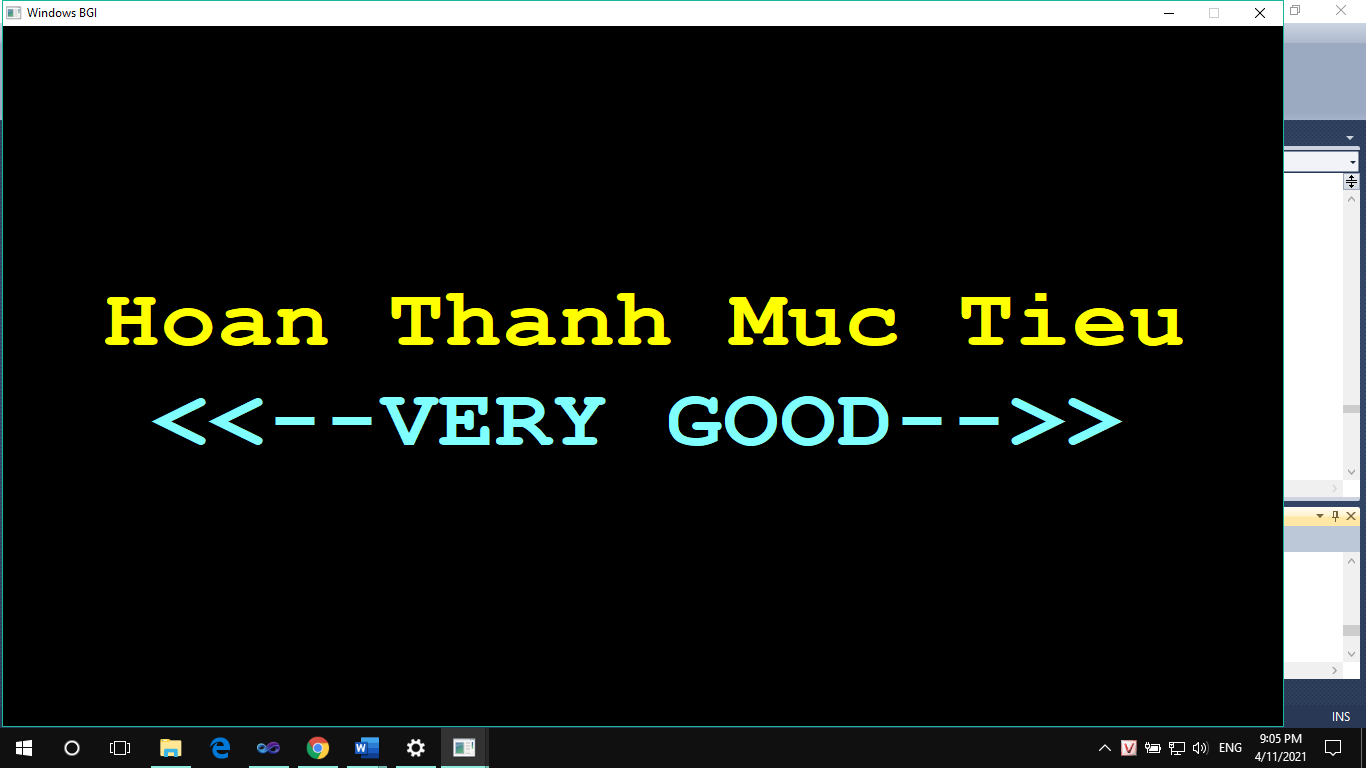
Hình 3-0‑3 Màn hình Play game

* Trong màn hình chính: dây móc câu quay tiến quay lùi 180 độ dưới đường ranh giới, bước nhảy là 5 độ
* Click chuột trái dây móc câu sẽ dài ra 1 khoảng và chiều dài sẽ trở lại như cũ khi thả chuột
* Nếu khi click chuột trái đầu dây rơi 1 vào khoảng xác định (x nằm trên đường tròn có đường thẳng kéo dài đi qua vật thể) sẽ khiến cục vàng tương ứng tịnh tiến đến gần ranh giới và biến mất, trúng mìn đỏ thì game over



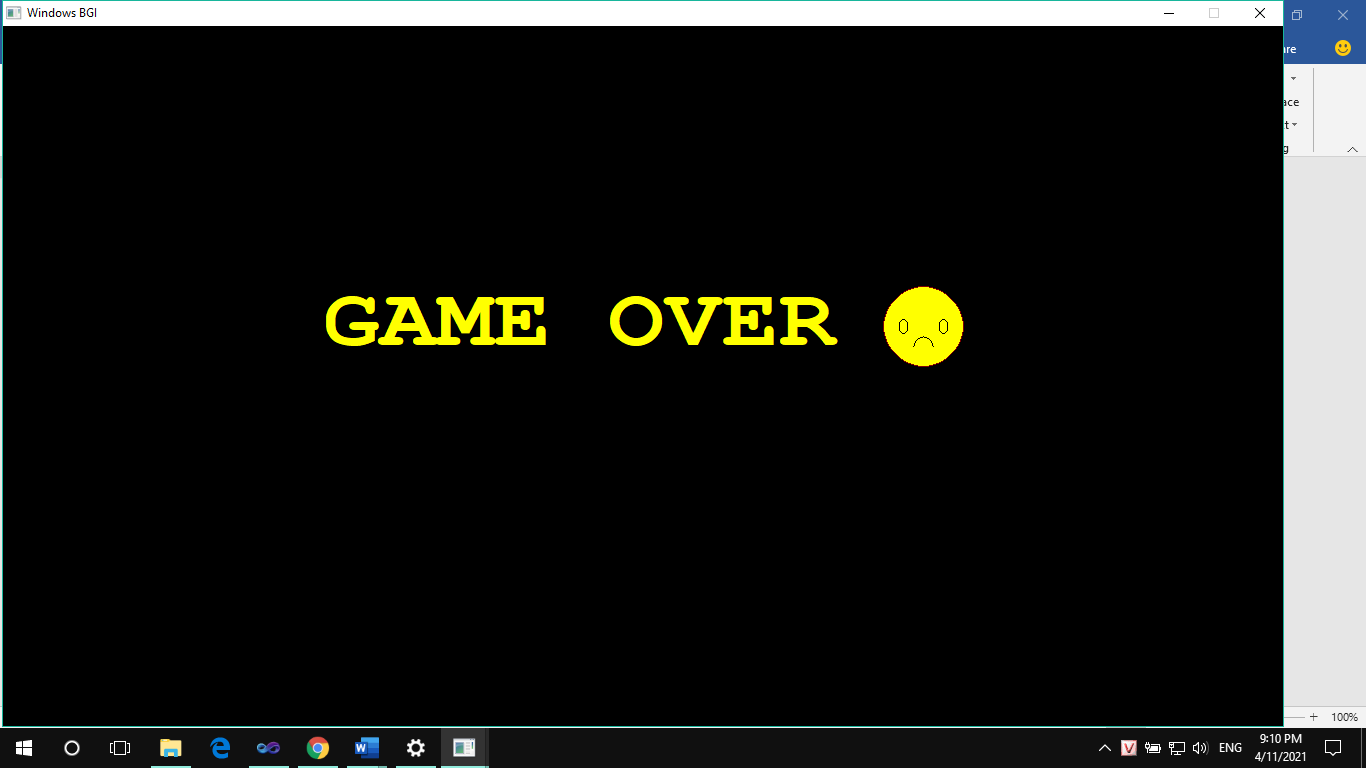
Hình 3-0‑4 Mô tả việc xử lý móc

* Người chơi sẽ được cộng thêm số điểm khác nhau tùy theo cục vàng móc được
* Nếu đạt được số điểm yêu cầu (150) thì sẽ hiện màn hình Hoàn Thành



Hình 3-0‑5 Màn hình hoàn thành mục tiêu

* Sau khoảng 60s không đạt đủ điểm sẽ game over



Hình 3-0‑6 Màn hình Game Over

## **Ưu điểm của đề tài**

* Tổng hợp các kiến thức đồ họa máy tính đã học trên lớp
* Ôn lại các nội dung lập trình đã được thực hành
* Bắt buộc phải tìm hiểu thêm kiến thức xử lý mới để chương trình được thực thi mượt hơn
* Là cơ hội rèn luyện cho sinh viên muốn theo hướng lập trình game bằng ngôn ngữ C++
* Rèn luyện kỹ năng làm việc nhóm
* Nâng cao trí tưởng tượng để có thể thiết kế nhân vật, vật thể hoàn toàn bằng các đường thẳng , đường tròn, đa giác… từ thư viện graphics.h
* Vận dụng kiến thức toán phổ thông để tính toán đặt vị trí cho các phép biến hình

## **Nhược điểm của đề tài**

* Gặp nhiều khó khăn khi sử dụng các hàm cleardevice() của thư viện graphics.h bởi nó sẽ xóa toàn bộ màn hình đồ họa và buộc bạn phải vẽ lại những chi tiết cố định ở sự kiện sau
* Hạn chế về màu sắc khi lập trình trên C++ BGI chỉ hiện được 16 màu
* Khó khăn về việc xử lý số liệu
* Khó khăn về việc lập trình các chức năng game

## **Hướng dẫn phát triển trong tương lai**

* Phát triển các hình ảnh trong game từ 2D lên 3D
* Hoàn thiện chương trình
* Thêm hiệu ứng
* Thêm âm thanh
* Thêm nhiều level chơi
* Lưu thông tin từng người chơi
* Bảng xếp hạng người chơi

# **Tài liệu tham khảo**

|  |  |
| --- | --- |
| [1] | A. Lê. [Online]. Available: https://www.stdio.vn/modern-cpp/xu-ly-su-kien-chuot-va-phim-voi-borland-graphics-interface-bgi-q2BL2. |