

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ﻣKHOA CÔNG NGHỆ THÔNG TINﻣ**

****

**BÀI TẬP LỚN**

**MÔN: LẬP TRÌNH WINDOWS**

**ĐỀ TÀI: XÂY DỰNG CHƯỜNG TRÌNH TRÒ CHƠI DÒ MÌN**



**Giáo viên hướng dẫn: *TS. Phạm Văn Hà***

**Lớp: Kỹ thuật phần mêm 2-K9**

**Hà Nội\_2016**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ﻣKHOA CÔNG NGHỆ THÔNG TINﻣ**

****

**BÀI TẬP LỚN**

**MÔN: LẬP TRÌNH WINDOWS**

**ĐỀ TÀI: : XÂY DỰNG CHƯỜNG TRÌNH TRÒ CHƠI DÒ MÌN**

**Giáo viên hướng dẫn: *TS. Phạm Văn Hà***

**Lớp: Kỹ thuật phần mêm 2-K9**

***Sinh viên thực hiện:***

1. ***Vũ Huy Hoàng***
2. ***Nguyễn Viết Nguyên***
3. ***Tài Lê Vi***

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc469213303)

[MỞ ĐẦU 2](#_Toc469213304)

[CHƯƠNG I – TỔNG QUAN VỀ BÀI TOÁN 3](#_Toc469213305)

[I. Đặt vấn đề 3](#_Toc469213306)

[II. Mô tả vấn đề 3](#_Toc469213307)

[1. Đầu vào 4](#_Toc469213308)

[2. Đầu ra 4](#_Toc469213309)

[CHƯƠNG 2 – PHÂN TÍCH VÀ GIẢI THUẬT 5](#_Toc469213310)

[I. Luật lệ của game 5](#_Toc469213311)

[1. Các trạng thái của ô 5](#_Toc469213312)

[2. Các trường hợp khi mở một ô 6](#_Toc469213313)

[3. Cắm cờ và đánh dấu 6](#_Toc469213314)

[II. Phân tích 6](#_Toc469213315)

[1. Thiết kế cấu trúc 6](#_Toc469213316)

[a. Biểu đồ use case tổng quát 6](#_Toc469213317)

[b. Mô tả các use case chính 7](#_Toc469213318)

[2. Giao diện và xây dựng chương trình 7](#_Toc469213319)

[a. Giao diện 7](#_Toc469213320)

[b. Xây dựng chương trình 8](#_Toc469213321)

[c. Cơ sở dữ liệu 15](#_Toc469213322)

[III. Các thuật toán quan trọng 18](#_Toc469213323)

[CHƯƠNG III – TỔNG KẾT 19](#_Toc469213324)

[I. Kết luận 19](#_Toc469213325)

[II. Tài liệu tham khảo 19](#_Toc469213326)

# ***LỜI NÓI ĐẦU***

**Ngày nay với sự phát triển của công nghệ thông tin, game đã trở thành hình thức giải trí rất phổ biến trên thế giới với sức lôi cuốn đặc biệt. Qua nhiều thời kỳ, các thể loại game không ngừng thay đổi đáp ứng nhu cầu đa dạng của người chơi. Tuy nhiên với sự xuất hiện của một số loại game kinh điển thì đến ngày nay vẫn được rất nhiều người biết đến.**

Để áp dụng những kiến thức đã được học trên lớp chúng em đã lên ý tưởng để thiết kế, xây dựng chương trình của game “dò mìn”. Do còn nhiều hạn chế về trình độ cũng như thời gian nên đề tài này không tránh khỏi sai sót. Chúng em rất mong nhận được sự góp ý của thầy cô và các bạn để giúp đề tài của chúng em hoàn thiện hơn. Qua đây chúng em xin chân thành cảm ơn Thầy TS. Phạm Văn Hà đã hướng dẫn chúng em hoàn thành bài tập lớn này.

Dưới đây là phần trình bày của chúng em!!!

# ***MỞ ĐẦU***

Tin học là nghành khoa học mũi nhọn phát triển hết sức nhanh chóng trong vài chục năm gần đây,ngày nay nghành khoa học này càng được mở rộng lĩnh vực nghiên cứu , ứng dụng trong mọi mặt của đời sống xã hội.

Trong vài thập kỷ gần đây , xã hội loài người có sự bùng nổ về thông tin , theo quan điểm truyền thống 3 nhân tố cơ bản của nền kinh tế là điều kiện tự nhiên , nguồn lao động và vốn đầu tư. Ngày nay , ngoài 3 nhân tố then chốt đó còn có 1 nhân tố mới rất quan trọng đó là thông tin\_một dạng tài nguyên mới.

Cùng với việc sáng tạo ra công cụ mới là máy tính điện tử , con người cũng tập trung trí tuệ xây dựng nghành khoa học tương ứng để đáp ứng nhu cầu khai thác thông tin và tài nguyên thông tin.

Trong bối cảnh đó, nghành công nghệ thông tin được phát triển thành một nghành khoa học với nội dung, đích phương pháp nghiên cứu riêng và ngày càng ứng dụng trong hầu hết các lĩnh vực hoạt động của xã hội loài người.

Đặc biệt trong đời sống thực tế ngày nay, tin học được ứng dụng rất nhiều trong đời sống giải trí. Nhiều trò chơi mới được viết bằng ngôn ngữ tin học xuất hiện, đem lại sự đam mê thích thú, khơi dậy sự sáng tạo, tìm tòi cho giới trẻ và toàn xã hội. đặc biệt hơn với những ngôn ngữ lập trình từ bậc thấp tới các ngô ngữ bậc cao như( pascal, java, c#...) các bạn trẻ có thẻ tự lập trình xây dựng lên những trò chơi đơn giản do chính mình thiết kế.

Ví dụ : trò chơi ném bóng, trò chơi bắn máy bay,trò chơi cờ caro… và điển hình là trò chơi Dò Mìn

# **CHƯƠNG I – TỔNG QUAN VỀ BÀI TOÁN**

1. **Đặt vấn đề**

Ít ai biết rằng, tựa game "dò mìn"( Minesweeper) nổi tiếng trên Windows một thời đã hình thành nên ý tưởng trí thông minh nhân tạo hoặc robot thay thế công việc của con người hiện nay. Sự ra đời của game dò mìn đã tạo nên một thành công không hề nhỏ của Microsoft. Cũng giống như hầu hết các game của Windows, dò mìn hay gỡ mìn chỉ đơn thuần chỉ là một game dùng để giết thời gian. Nó được ra đời cách đây khá lâu và dành cho một người chơi ở một thời điểm mà hầu hết các máy tính đều sử dụng các câu lệnh để hoạt động. Minesweeper có mục đích của riêng nó như là một cách tập cho người sử dụng máy tính sử dụng con trỏ chuột: Người chơi có nhiệm vụ mở hết các ô vuông mà không được bấm vào ô có chứa mìn, bấm chuột trái là mở ô, bấm chuột phải là cắm cờ, khi bạn bấm phải một ô có chứa bom thì bạn sẽ thua. Tuy là một game mà ai cũng biết nhưng để có thể chơi tốt và chơi giỏi dò mìn thì không phải là điều dễ dàng.

Bên cạnh đó game này đã góp phần giúp Microsoft hướng dẫn người dùng cách sử dụng thao tác chuột trên Windows, và khiến Windows trở nên phổ biến như hiện nay.Tuy nhiên, hơn thế nữa, với game dò mìn, người dùng còn được rèn luyện khả năng tư duy thông minh, tránh "mìn" trên bản đồ.

Bởi thay vì muốn Windows không chỉ được phục vụ trong doanh nghiệp, Microsoft còn kỳ vọng, hệ điều hành của hãng sẽ còn sử dụng trong các gia đình, trở nên gần gũi hơn với người dùng cá nhân, hoặc với mục đích giải trí. Để làm được điều này, các chuyên gia của Microsoft nhận định, "*một trò chơi trên Windows sẽ là cầu nối nhanh nhất tới mục tiêu trên*". Do đó game “Minesweeper” ra đời. Có một điều mà Microsoft không ngờ tới đó là Minesweeper lại thành công một cách đáng ngạc nhiên và trở thành một phần không thể thiếu của Windows.

1. **Mô tả vấn đề**

Bạn đã từng chơi Minesweeper chưa? Đó là một game nhỏ rất thú vị. Mục tiêu của trò chơi là tìm ra tất cả các quả mìn trong một vùng có kích thước M x N. Để giúp người chơi, trò chơi cho hiển thị ở mỗi ô vuông số mìn ở các ô liền kề. Ví dụ, một vùng có kích thước 4×4 chứa 2 quả mìn(được biểu diễn bằng kí tự \*) như sau:

\* . . .

. . . .

. \* . .

. . . .

Nếu chúng ta muốn thể hiện các con số để gợi ý số mìn cho vùng kể trên, chúng ta sẽ có kết quả sau:

\*100

2210

1\*10

1110

Như bạn thấy, mỗi ô trong vùng có tối đa là 8 ô bao quanh nó.

1. **Đầu vào**

Qua đây chúng ta có thể thấy đầu vào bao gồm một số lượng tùy ý các vùng chứa mìn khác nhau. Dòng đầu tiên của mỗi vùng chứa 2 số nguyên n và m (0<n, m<=100), tương ứng là số hàng và số cột của vùng đó, ở mỗi hàng trong n hàng tiếp theo chứa chính xác m ký tự biểu diễn cho mỗi ô. Mỗi ô an toàn (không chứa mìn) được biểu diễn bằng ký tự “.” (không có dấu nháy kép) và mỗi ô chứa mìn được biểu diễn bằng kí tự “\*” (không có dấu nháy kép). Khi dòng đầu tiên của mỗi vùng mà n=m=0 thì sẽ ngừng nhập dữ liệu đầu vào và không xử lý gì thêm nữa.

1. **Đầu ra**

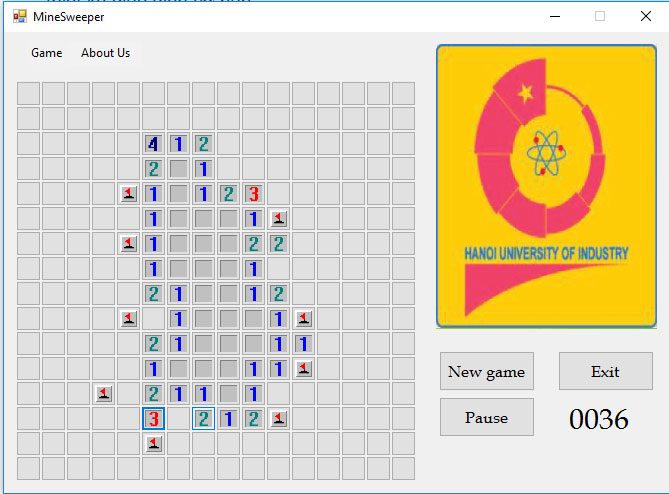
Với mỗi vùng, đầu tiên bạn sẽ in ra dòng thông báo như sau:

Vùng thứ #x:

Trong đó x là số thứ tự của vùng đó (bắt đầu từ 1, số vùng phụ thuộc dữ liệu đầu vào). n dòng tiếp theo sẽ bao gồm các ô mà kí tự “.” đã được thay thế bằng số lượng mìn xung quanh ô đó. Và phải có một dòng trắng (xuống dòng) để phân cách các vùng khác nhau.

# **CHƯƠNG 2 – PHÂN TÍCH VÀ GIẢI THUẬT**

1. **Luật lệ của game**



**\*Luật chơi**

Minisweeper (của Windows) có phần giao diện chính là một bảng các ô vuông xếp liền nhau tạo thành một hình chữ nhật có chiều rộng và dài tối thiểu là 9 ô (đơn vị là ô vuông) và số mìn tối thiểu là 10. Trong bài này chúng em thiết kế một bảng có kích thước 16x16 ô vuông và có số mìn là 40.

Trong bảng này sẽ có các ô được đặt mìn ngẫu nhiên và nhiệm vụ của người chơi là mở tất cả các ô không có mìn bằng cách click chuột trái vào các ô đó, khi chỉ còn các ô có mìn còn lại thì kết thúc màn chơi.

1. **Các trạng thái của ô**

Tùy theo quá trình khởi tạo và thao tác của người dùng mà các ô trong bảng có thể có một hoặc vài trạng thái trong các trạng thái sau:

* Có mìn: được đặt ngẫu nhiên lúc khởi tạo
* Đã mở: Khi người dùng nhấn chuột trái vào ô
* Được cắm cờ: Khi người dùng nhấn phải vào ô
* Được đánh dấu: Khi nhấn phải vào ô đã được “cắm cờ”
* Bình thường: không có tất cả các trạng thái trên

1. **Các trường hợp khi mở một ô**

Khi mở một ô **X** nào đó, có 3 trường hợp có thể xảy ra:

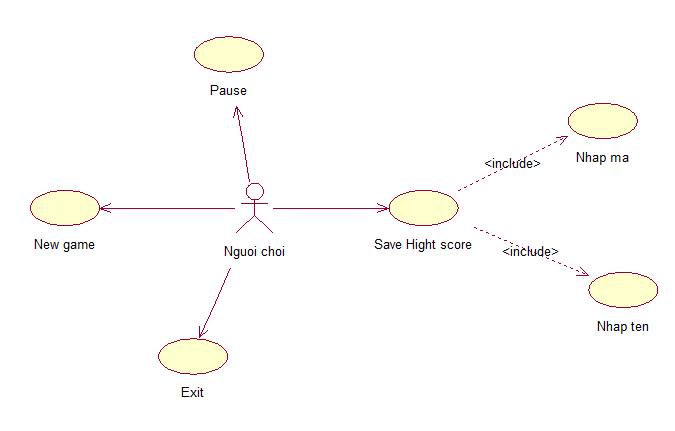
* X có mìn: hiện tất cả mìn trong bảng ra và ‘game over’.
* X không có mìn nhưng 8 ô xung quanh có mìn: hiện số mìn xung quanh vào X.
* X không có mìn và xung quanh cũng không có mìn: mở lần lượt các ô xung quanh X cho đến khi gặp các trường hợp 1 và 2.

1. **Cắm cờ và đánh dấu**

Minisweeper cho phép bạn đánh dấu các ô nghi ngờ có mìn bằng cách “cắm cờ” và “đánh dấu”. Khi bạn “cắm cờ”, tức là bạn xác định rằng ô đó có mìn và ô đó được hiển thị là một lá cờ. Bạn không thể mở ô đó bằng chuột trái được. Bạn chỉ có thể hủy bỏ trạng thái “cắm cờ” bằng cách click chuột phải, tùy theo thiết lập mà ô đó sẽ chuyển sang trại thái “đánh dấu” hoặc “bình thường”.

Khi bạn “đánh dấu”, tức là bạn đoán rằng ô đó có thể có mìn nhưng không chắc chắn.

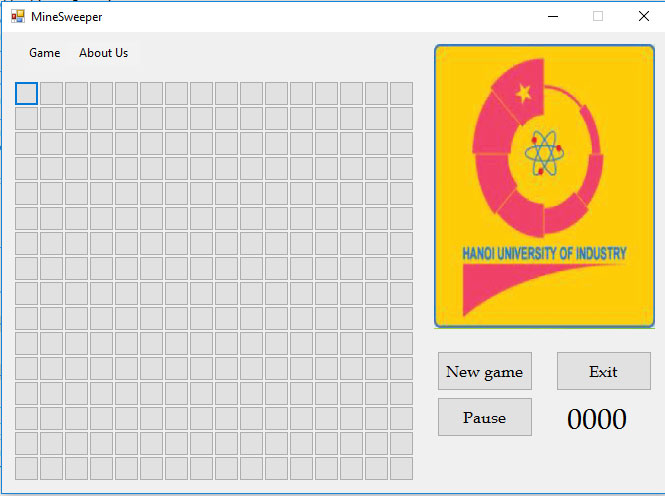
1. **Phân tích**
2. **Thiết kế cấu trúc**
3. **Biểu đồ use case tổng quát**



1. **Mô tả các use case chính**

* “New game” khi người chơi chọn thì một cửa sổ kích thước 16\*16 sẽ xuất hiện và toàn bộ những ô ở lần chơi trước sẽ mặc định bị xoá và quay lại trạng thái chơi ban đầu.
* “Exit” lập tức sẽ thoát game- không hiện thông báo.
* Khi người chơi thắng( tức là đặt được hết cờ vào các ô có mìn hoặc mở được hết các ô không có mìn) thì hệ thống sẽ hiện ra "You Win” và hiện ra một bảng với tiêu đề “EnterInformation” để cho người dùng “lưu điểm cao” và bắt người sử dụng “mã” và nhập “tên” . Khi người chơi nhập tên và chọn “xác nhận” thì điểm và tên người chơi sẽ được lưu cơ sở dữ liệu “HightScore” đồng thời hệ thống xuất ra thông báo “ Thêm thành công”.
* Khi người sử dụng chọn “Pause “ thì đồng hồ sẽ dừng.

1. **Giao diện và xây dựng chương trình**
2. **Giao diện**

****

Để xây dựng trò chơi này, em đã tiến hành thực hiện những công việc sau:

1. Tạo một đối tượng form làm giao diện chính của chương trình. Form này gồm 2 component Panel và 1MenuStrip

* Panel đầu tiên để chứa một mảng 2 chiều các MyButton được kế thừa từ component Button
* Panel thứ hai để chứa 1 PictureBox, 3 Button và 1 Label để hiển thị thời gian.
* .MenuStrip cho phép người chơi có những lựa chọn.

1. Tạo 1 class MyButton kế thừa từ Component có sẵn Button. Lớp này được bổ sung thêm một số thuộc tính để tiện thao tác về sau.
2. Cùng với một số form phụ để hiển thị những thông tin ứng dụng.
3. Để viết chương trình cần sử dụng các kiến thức về lập trình như: mảng 2 chiều, vòng lặp for, …
4. **Xây dựng chương trình**

\*Mở đầu

Phần này chúng em thiết kế một lớp **mybutton** đại diện cho một ô của bảng mìn trong trò chơi.

Lớp **mybutton**  dựa trên các trạng thái của một ô, người chơi có thể sử dụng enum thay cho class, trong bài này chúng em sử dụng class để người đọc dễ hiểu. Lớp này chỉ bao gồm các field (người chơi có thể dùng Property cho “máy móc”) sau:

Lớp **mybutton**  chứa dữ liệu cần thiết như đã mở, đã cắm cờ, đã đánh dấu, là mìn, ô bình thường.

// Đoạn code class MyButton

public partial class MyButton : System.Windows.Forms.Button

{

public bool isOpened;

public bool isNormal;

public bool isMark;

public bool isFlag;

public bool isMine;

public int minesAround = 0;

public MyButton()

{

InitializeComponent();

}

public MyButton(IContainer container)

{

container.Add(this);

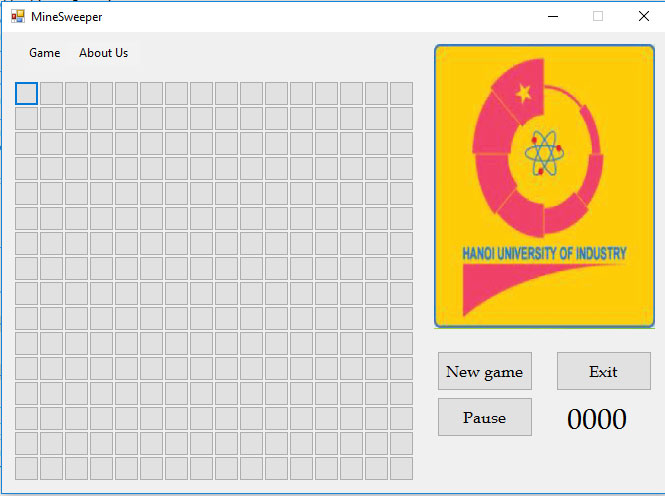
InitializeComponent();

}

}

**\***Vẽ giao diện

Khi thiết kế lớp này, chúng em đã tập trung vào hai phần chính: Hiển thị và xử lý thao tác của người dùng. Chúng em đã dùng mảng 2 chiều có kích thước 16x16 để thiết kế ra giao diện.

****

// Đoạn code tạo ra bãi mìn

void drawBoard()

{

for (int i = 0; i < Const.heighS; i++)

{

for (int j = 0; j < Const.widthS; j++)

{

btnArr[i,j] = new MyButton()

{

Width = Const.widthChess,

Height = Const.heightChess,

Location = new Point(Const.widthChess\*j,Const.heightChess\*i),

BackgroundImageLayout = ImageLayout.Stretch,

Tag = i\*16+j

};

btnArr[i, j].Click += MineSweeper\_Click;

btnArr[i, j].MouseDown += MineSweeper\_MouseDown;

pnlBoard.Controls.Add(btnArr[i,j]);

}

}

}

Tiếp theo khởi tạo vị trí của các quả bom. Bom sẽ được xuất hiện ở các vị trí ngẫu nhiên bằng cách sử dụng hàm random. Nếu số bom được tạo ra nhỏ hơn số bom quy định của thể loại 16\*16 thì tạo tiếp.



void initBoad()

{

Random rd = new Random();

int coutMines = 0;

while (Const.mines > coutMines)

{

int posMine = rd.Next(0, Const.widthS \* Const.heighS - 1);

int rMine = posMine / Const.widthS;

int cMine = posMine % Const.widthS;

if (!btnArr[rMine, cMine].isMine)

{

btnArr[rMine, cMine].isMine = true;

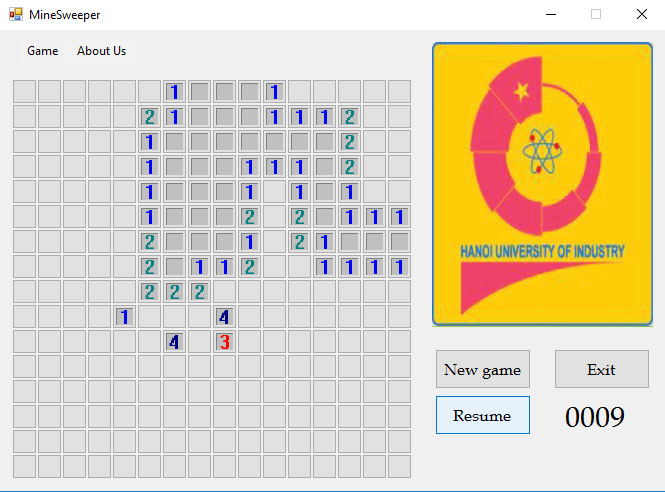
coutMines++;

}

}

}

Tiếp đến chúng em viết một phương thức để “mở” một ô trong bảng, như đã giới thiệu trong phần luật chơi, người chơi phải kiểm tra các trạng thái của ô đó và quyết định sẽ làm gì. Trong trường hợp thứ **3** người chơi phải dùng đệ quy để duyệt và mở các ô xung quanh. Nhưng trước tiên, chúng ta hãy xem về phương pháp duyệt các ô để đếm số mìn quanh một ô.



int coutMinesAround(int i,int j)

{

int countAround;

countAround = 0;

if (!btnArr[i, j].isMine)

{

int x1 = i == 0 ? 0 : -1;

int y1 = j == 0 ? 0 : -1;

int x2 = i == Const.heighS - 1 ? 1 : 2;

int y2 = j == Const.widthS - 1 ? 1 : 2;

for (; x1 < x2; x1++)

{

for (int k = y1; k < y2; k++)

{

if (btnArr[i + x1, j + k].isMine)

countAround++;

}

}

}

return countAround;

}

Ban đầu người chơi cần click chuột trái vào một ô bất kỳ để mở một ô. Khi ô đầu tiên được mở ra sẽ hiện ra số hoặc có thể là mìn( khả năng này sẽ ít hơn).

public bool OpenCell(int i, int j)

{

//int openedCount = 0;

if (btnArr[i, j].isOpened || btnArr[i, j].isFlag)

return false;

btnArr[i, j].isOpened = true;

if (btnArr[i, j].isMine)

{

return true;

}

//openedCount++;

Đếm số mìn xung quanh để kiểm tra các trường hợp

int count = coutMinesAround(i, j);

if (count > 0)

{

btnArr[i, j].minesAround = count;

btnArr[i, j].BackgroundImage = Image.FromFile(Application.StartupPath + "\\Resources\\" + coutMinesAround(i, j) + ".bmp");

}

else

{

int x1 = i == 0 ? 0 : -1;

int y1 = j == 0 ? 0 : -1;

int x2 = i == Const.heighS - 1 ? 1 : 2;

int y2 = j == Const.widthS - 1 ? 1 : 2;

btnArr[i,j].BackgroundImage = Image.FromFile(Application.StartupPath + "\\Resources\\" + coutMinesAround(i ,j) + ".bmp");

for (; x1 < x2;x1++ )

for (int k = y1; k < y2; k++)

{

OpenCell(i + x1, j + k);

btnArr[i + x1, j + k].isOpened = true;

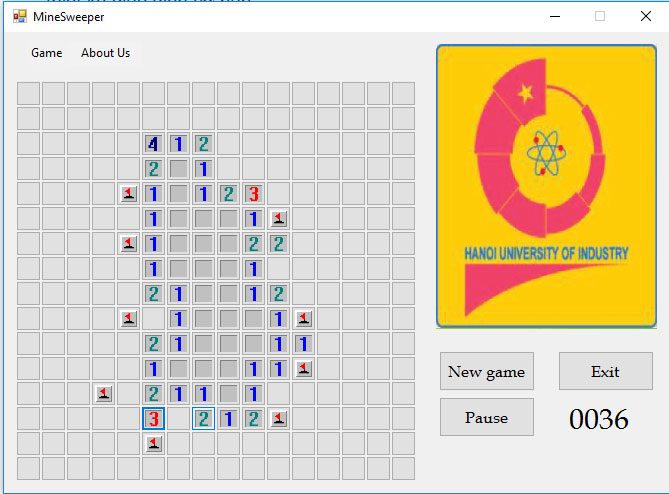
}

}

return false;

}

Để biết được vị trí của những ô chứa mìn bằng cách click chuột phải vào ô được đoán là có mìn để cắm cờ đánh dấu.



//Đoạn code xử lí sự kiện khi ấn chuột phải

void MineSweeper\_MouseDown(object sender, MouseEventArgs e)

{

MyButton btn = sender as MyButton;

timerPoint.Start();

if (btnPause.Text.Equals("Resume"))

btnPause.Text = "Pause";

if (!btn.isOpened && !btn.isFlag)

{

if (e.Button == MouseButtons.Right)

{

btn.BackgroundImage = Image.FromFile(Application.StartupPath + "\\Resources\\Flag.bmp");

btn.isFlag = true;

}

}

else

{

if (e.Button == MouseButtons.Right && !btn.isOpened)

{

btn.BackgroundImage = null;

btn.isFlag = false;

}

}

}

Khi người chơi đã đánh dấu được chính xác các vị trí của các ô chứa mìn và mở được hết các ô chúng ta cần kiểm tra xem người chơi đã thắng hay chưa. Nếu thắng, người chơi sẽ được nhập và lưu thông tin của mình vào cơ sở dữ liệu.

//Xử lí trực tiếp trong sự kiện không cần gọi checkWin()

int dem = 0;

for (int i = 0; i < Const.heighS; i++)

{

for (int j = 0; j < Const.widthS; j++)

{

if (btnArr[i, j].isOpened && !btnArr[i, j].isMine)

dem += 1;

}

}

if (dem == Const.heighS \* Const.widthS - 40)

{

lblResult.Text = "You Win";

EnterInformation ei = new EnterInformation();

timerPoint.Stop();

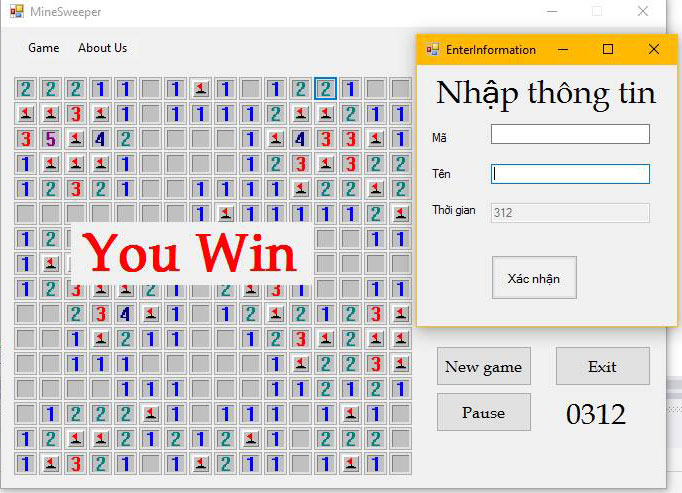
ei.setTextBoxTime = bienDemTG.ToString();

ei.Show();

}

1. **Cơ sở dữ liệu**

Sau khi mở được hết các ô không chứa bom. Lúc này người chơi đã thắng thì chương trình sẽ hiện ra giao diện thông báo “you win” và thời gian dừng. Lúc này giao diện sẽ hiện ra một form để người chơi nhập thông tin người chơi gồm mã, tên và thời gian dừng. Người chơi có thời gian chơi ít nhất sẽ là điểm cao nhất. Điểm của người chơi sẽ được lưu trong bảng cơ sở dữ liệu.



public HightScore()

{

InitializeComponent();

}

private void HightScore\_Load(object sender, EventArgs e)

{

HienThiDiem();

}

void HienThiDiem()

{

SqlConnection connect = new SqlConnection(@"Data Source=DESKTOP-V4QPNQU\SQLEXPRESS;Initial Catalog=QuanLiDiem;Integrated Security=True");

SqlDataAdapter adap = new SqlDataAdapter("select \* from Diem order by ThoiGian ",connect);

DataTable table = new DataTable();

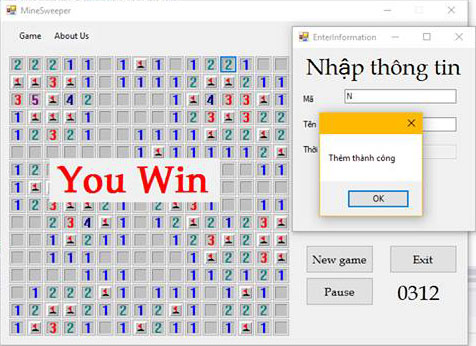
adap.Fill(table);

dgvHightScore.DataSource = table;

}

}

Sau khi nhập đầy đủ thông tin và người chơi click vào button xác nhận thì dữ liệu sẽ được thêm vào cơ sở dữ liệu. Người chơi đã lưu thành công!



private void btnSend\_Click(object sender, EventArgs e)

{

try

{

SqlConnection connect = new SqlConnection(@"Data Source=DESKTOP-V4QPNQU\SQLEXPRESS;Initial Catalog=QuanLiDiem;Integrated Security=True");

connect.Open();

SqlCommand cmd = new SqlCommand();

cmd.Connection = connect;

cmd.CommandType = CommandType.Text;

cmd.CommandText = System.String.Concat("insert into Diem values("+ "'"+ txtCode.Text.ToString() +"','" +txtName.Text.ToString() +"','"+ txtTime.Text.ToString() + "')");

cmd.CommandType = CommandType.Text;

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm thành công");

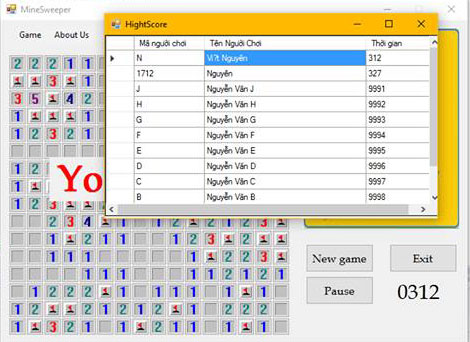
}catch(SqlException){

MessageBox.Show("Không thêm được");

}

}

Khi đã thêm thành công thì danh sách những người chơi thắng sẽ được hiện ra . Người chơi có thể xem được điểm và thứ tự xếp hạng của mình là bao nhiêu.



1. **Các thuật toán quan trọng**
2. Thuật toán loang

Thuật toán này được sử dụng để giải quyết vấn đề khi người sử dụng nhấn chuột trái vào 1 ô, nếu ô đó không có mìn và xung quanh nó cũng không có mìn nó sẽ loang cho đến khi gặp trướng hợp 1 và 2 như ở mục ‘Trường hợp mở một ô’.

1. Thuật toán sinh số ngẫu nhiên

Tạo một số nguyên trong khoảng từ 0 đến (16\*16-1) rồi tính toán để gán thuộc tính isMine của MyButton có vị trí tương ứng trong mảng 2 chiều là true. Lặp cho đến khi nào đủ 40 MyButton có isMine là true.

1. Thuật toán sử dụng vòn lặp lồng nhau

Thuật toán này được sử dụng để tính số bom xung quanh mỗi ô.

1. Thuộc tính có vai trò rất quan trọng : Tag

Thuộc tính này được gán giá trị khi tạo bãi mìn. Sở dĩ nó vai trò quan trọng là khi sự kiện click được gọi thì nó sẽ trả về vị trí của mybutton đấy trong mảng 2 chiều.

**CHƯƠNG III – TỔNG KẾT**

1. **Kết luận**

Cuối cùng, bài làm của chúng em xin được dừng ở đây. Chúng em đã cố gắng vận dụng những kiến thức mình có, cũng như tích cực tham khảo các tài liệu trên mạng Internet, của bạn bè cũng như các tài liệu liên quan. Kiến thức là vô cùng rộng lớn nên có lẽ, bài làm của chúng em còn nhiều thiếu sót cũng như chưa thực sự hoàn hảo. Chúng em rất mong sẽ nhận được những lời đóng góp chia sẻ ý kiến của thầy, cũng như các nhóm bạn để nhóm chúng em có thể hoàn thành bài một cách tốt nhất.

Nhờ việc làm bài tập lớn với đề tài "Game Dò Mìn- Minesweeper " chúng em đã phần nào hiểu được ý nghĩa, tầm quan trọng của môn học "Lập trình Windows "- là một môn học thú vị và có ý nghĩa quan trọng trong ngành CNTT, cũng như là nền tảng cho các môn học khác sau này. Chúng em xin chân thành cảm ơn thầy Phạm Văn Hà đã tận tình giúp đỡ chúng em trong quá trình học tập cũng như trong quá trình hoàn thành cuốn bài tập lớn này. Chúng em xin chân thành cảm ơn !

1. **Tài liệu tham khảo**

* Giáo trình “ Lập trình Windows – Trường Đại Học Công nghiệp Hà Nội”
* Cách xây dựng biểu đồ ca sử dụng trong National Rose ( Nguồn: Internet)