

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**  
**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

---



# **BÁO CÁO ĐỒ ÁN**

## **CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**ĐỀ TÀI: ỨNG DỤNG STACK VÀO GAME 2048**

**SVTH : VŨ NGỌC BỘI**

**MSSV : 17110102**

**SVTH : VŨ VĂN HIẾU**

**MSSV : 17133020**

*TP. Hồ Chí Minh, tháng 10 năm 2019*

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>1</b>
<b>DANH MỤC CÁC HÌNH .....</b>	<b>2</b>
<b>DANH MỤC CÁC BẢNG .....</b>	<b>3</b>
<b>1.GIỚI THIỆU.....</b>	<b>4</b>
<b>1.1. Game 2048 là gì? .....</b>	<b>4</b>
<b>1.2. Cách chơi .....</b>	<b>4</b>
<b>1.3. Mục tiêu đề ra .....</b>	<b>4</b>
<b>1.4. Lí do chọn đề tài.....</b>	<b>5</b>
<b>1.5. Ngôn ngữ cài đặt .....</b>	<b>5</b>
<b>2.PHÂN TÍCH GAME.....</b>	<b>6</b>
<b>2.1. Mô tả về game .....</b>	<b>6</b>
<b>2.2. Phân tích game .....</b>	<b>6</b>
<b>3. BẢNG KẾ HOẠCH THỰC HIỆN (DỰ KIẾN) .....</b>	<b>7</b>
<b>4. GIAO DIỆN PHẦN MỀM VÀ MÔ TẢ CHỨC NĂNG .....</b>	<b>9</b>
<b>4.1. Giao diện ngoài.....</b>	<b>9</b>
<b>4.2. Chức năng trong game .....</b>	<b>12</b>
<b>5. THIẾT KẾ CODE.....</b>	<b>13</b>
<b>5.1. Các lớp được sử dụng trong chương trình:.....</b>	<b>13</b>
<b>5.2. Các phương thức Form1.cs: .....</b>	<b>13</b>
<b>5.3. Ứng dụng Stack và giải thích.....</b>	<b>14</b>
<b>6. KẾT LUẬN .....</b>	<b>18</b>
<b>6.1. Mức độ hoàn thành.....</b>	<b>18</b>

<b>6.2. Các khó khăn gặp phải.....</b>	<b>18</b>
<b>6.3. Ưu điểm, nhược điểm của Game.....</b>	<b>18</b>
<b>6.4. Ý tưởng phát triển .....</b>	<b>19</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>20</b>

## **LỜI CẢM ƠN**

Trong lời đầu tiên của báo cáo đồ án môn Cấu Trúc Dữ Liệu Và Giải Thuật, đề tài “ Ứng dụng stack vào game 2048 ”, nhóm tác giả muốn gửi những lời cảm ơn và biết ơn chân thành nhất tới tất cả những người đã hỗ trợ, giúp đỡ về kiến thức và tinh thần trong quá trình thực hiện đồ án.

Trước hết chúng tôi xin chân thành cảm ơn Thầy Trần Công Tú, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ chúng tôi trong suốt quá trình thực hiện đồ án. Do thời gian thực hiện có hạn kiến thức còn nhiều hạn chế nên đồ án thực hiện chắc chắn không tránh khỏi những thiếu sót nhất định.

Nhóm tác giả rất mong nhận được ý kiến đóng góp của thầy cô giáo và các bạn để chúng tôi có thêm kinh nghiệm và tiếp tục hoàn thành đồ án của mình. Nhóm tác giả xin chân thành cảm ơn!

## **DANH MỤC CÁC HÌNH**

<b>Hình 1. Giao diện game.....</b>	<b>9</b>
<b>Hình 2. Lưu Điểm Cao.....</b>	<b>10</b>
<b>Hình 3. Game over .....</b>	<b>11</b>
<b>Hình 4. Chức Năng Undo .....</b>	<b>12</b>

## **DANH MỤC CÁC BẢNG**

**Bảng 1. Kế hoạch công việc dự kiến ..... 8**

**Bảng 2. Các lớp được sử dụng trong chương trình ..... 13**

**Bảng 3. Các phương thức được sử dụng trong lớp from.cs ..... 14**

## 1. GIỚI THIỆU

### 1.1. Game 2048 là gì?

- 2048 là một trò chơi giải đố do tác giả Gabriele Cirulli, một lập trình viên web trẻ 19 tuổi người Ý, tạo ra vào tháng 3 năm 2014. Mục tiêu của trò chơi là trượt các khối vuông có mang số trên một lưới vuông để kết hợp chúng lại và tạo ra khối vuông có giá trị 2048.
- Đây có thể xem như một dạng trò chơi giải đố trượt khối vuông và rất giống với ứng dụng Threes! ra mắt một tháng trước đó. Cirulli tạo ra trò chơi này chỉ trong một cuối tuần để tự kiểm nghiệm xem mình có thể viết một trò chơi từ bản phác thảo hay không.

### 1.2. Cách chơi

Game 2048 chơi trên một lưới vuông  $4 \times 4$ . Mỗi lần di chuyển là một lượt, người chơi sử dụng các phím mũi tên và các khối vuông sẽ trượt theo một trong bốn hướng tương ứng (lên, xuống, trái, phải). Mỗi lượt có một khối có giá trị 2 hoặc 4 sẽ xuất hiện ngẫu nhiên ở một ô trống trên lưới. Các khối vuông trượt theo hướng chỉ định cho đến khi chạm đến biên của lưới hoặc chạm vào khối vuông khác. Nếu hai khối vuông có cùng giá trị chạm vào nhau, chúng sẽ kết hợp lại thành một khối vuông có giá trị bằng tổng giá trị hai khối vuông đó (giá trị gấp đôi). Khối vuông kết quả không thể kết hợp với khối vuông khác một lần nữa trong một lượt di chuyển. Để dễ nhận biết thì các khối vuông giá trị khác nhau sẽ có màu sắc khác nhau.

Bảng điểm ở góc trên bên phải cho biết điểm của người chơi. Ban đầu điểm bằng 0. Khi hai khối vuông kết hợp thì người chơi sẽ tăng điểm là giá trị khối vuông mới. Bên cạnh điểm hiện tại là kỉ lục điểm cao nhất người chơi từng đạt được. Khi người chơi tạo được ô vuông có giá trị 2048 thì thắng cuộc. Lúc này người chơi có thể lựa chọn tiếp tục chơi để đạt các giá trị cao hơn 2048. Khi không còn nước đi hợp lệ (không còn ô trống và các ô kề nhau đều khác giá trị) thì trò chơi kết thúc.

### 1.3. Mục tiêu đề ra

*Xây dựng game 2048 có các chức năng như sau:*

- Thiết kế giao diện thân thiện
- Ứng dụng ngăn xếp (Stack) vào phím Undo giúp người chơi đi sai có thể quay lại bước đi trước.
- Tính năng lưu lại điểm cao sau mỗi lượt chơi.

#### **1.4. Lí do chọn đề tài**

Game 2048 được chúng tôi chọn vì đây là một game vui, mang tính giải trí cao, đồng thời có thể áp dụng được kiến thức học được . Nói rõ hơn đó là áp dụng stack vào Game để thực hiện đồ án.

#### **1.5. Ngôn ngữ cài đặt**

- Ngôn ngữ xây dựng game: Microsoft C#.NET với phiên bản .NET Framework4.5
- IDE và Compiler: Microsoft Visual Studio 2019



## **2.PHÂN TÍCH GAME**

### **2.1. Mô tả về game**

- Game 2048 là game khá đơn giản khi tất cả những gì bạn cần làm là dùng các phím mũi tên để di chuyển và kết hợp các ô số có cùng giá trị lại với nhau.
- Trò chơi sẽ kết thúc khi tất cả các ô trống đã được lấp đầy những con số.

### **2.2. Phân tích game**

- Các ô được tạo bởi các Label.
- Di chuyển lên trên, xuống dưới, sang trái, sang phải.
- Khi di chuyển, 2 ô kế nhau có cùng giá trị sau khi di chuyển sẽ tự động cập nhật lại thành ô mới có giá trị gấp đôi.
- Khi đi sai bước có thể dùng chức năng Undo để quay lại bước trước.
- Khi các ô được lấp đầy các con số thì game sẽ kết thúc.

### 3. BẢNG KẾ HOẠCH THỰC HIỆN (DỰ KIẾN)

DANH SÁCH CÔNG VIỆC	NỘI DUNG CHI TIẾT	Vũ Văn Hiếu	Vũ Ngọc Bội	Ngày bắt đầu	Ngày kết thúc
Khảo sát Game 2048	Khảo sát hình ảnh Game	X	X	20/9/2019	25/9/2019
	Tìm hiểu cách hoạt động Game	X		20/9/2019	25/9/2019
	Tìm hiểu ngăn xếp ứng dụng trong Game		X	20/9/2019	25/9/2019
Triển Khai Làm Game 2048	Chọn ngôn ngữ làm game (C#,C++, Java)	X	X	26/9/2019	30/9/2019
	Tạo hình ảnh đồ họa Game		X	1/10/2019	5/10/2019
	Xử lý random ra số ( 2, 4 ) và cộng dồn điểm	X		1/10/2019	9/10/2019
	Xử lý di chuyển các phím (Up, Down, Left, Right)	X		10/10/2019	15/10/2019
	Xử lý game kết thúc ( hiện thông báo, reset lại Game)		X	6/10/2019	15/10/2019
	Tạo chức năng undo		X	16/10/2019	10/25/2019

Viết Báo Cáo	Làm file báo cáo Word	X		16/10/2019	31/10/2019
	Làm file báo cáo PowerPoint	X	X	26/10/2019	31/10/2019

*Bảng 1. Kế hoạch công việc dự kiến*

## 4. GIAO DIỆN PHẦN MỀM VÀ MÔ TẢ CHỨC NĂNG

### 4.1. Giao diện ngoài



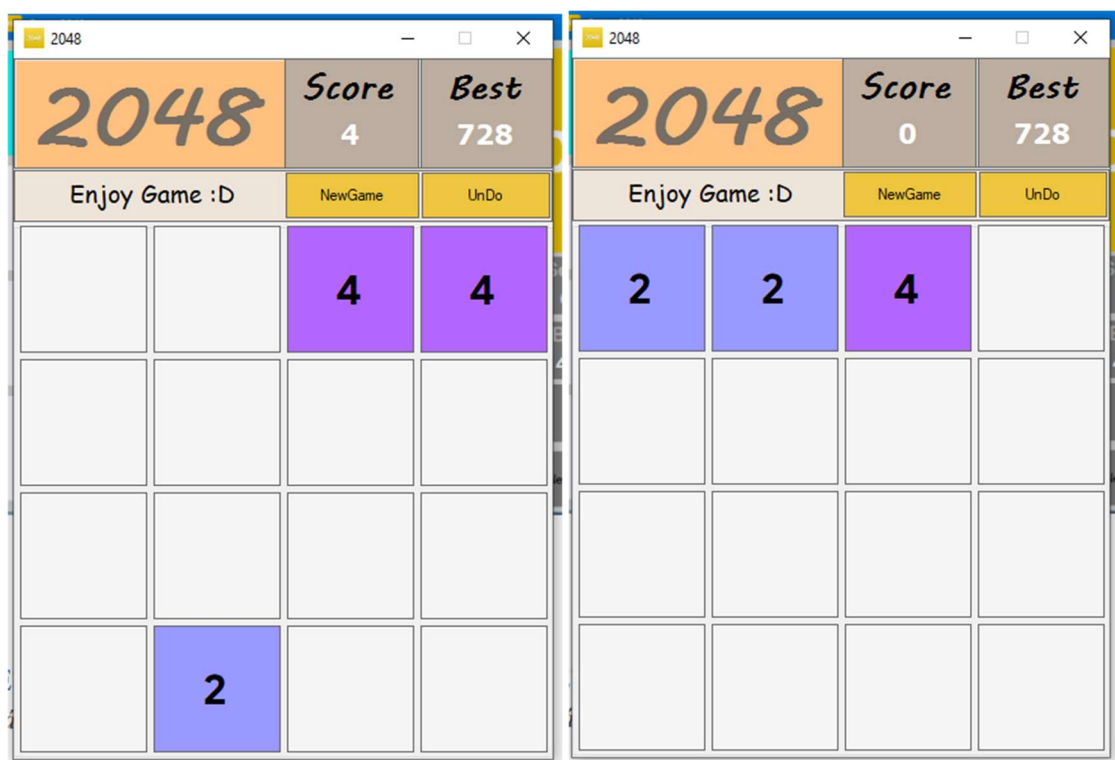
Hình 1. Giao diện game



Hình 2. Lưu Điểm Cao



Hình 3. Game over



Hình 4. Chức Năng Undo

#### 4.2. Chức năng trong game

- New Game để tạo mới lại game.
- Undo để quay lại bước đi trước đó.
- Score để báo điểm hiện tại của người chơi.
- Best để báo điểm cao nhất của người chơi.
- Khi lấp đầy tất cả các ô thì sẽ hiển thị thông báo như sau :  
 Có 2 sự lựa chọn là Yes hoặc No:
  - ✓ Chọn Yes để chơi game mới.
  - ✓ Chọn No để quay lại game ( có thể undo để tiếp tục game :D)

## 5. THIẾT KẾ CODE

### 5.1. Các lớp được sử dụng trong chương trình:

STT	Tên Class	Mục Đích
1	Form1.cs	Các hàm xử lý trong game
2	Form1.cs [Design]	Design giao diện game

*Bảng 2. Các lớp được sử dụng trong chương trình*

### 5.2. Các phương thức Form1.cs:

STT	Phương Thức	Chức Năng
1	Public Form1()	Hàm khởi tạo Các giá trị của biểu mẫu
2	private void LoadGame()	Thiết kế label , các biểu mẫu vào form và xuất hiện ngẫu nhiên 2 số hiển thị lần đầu tiên, đọc điểm cao từ file txt
3	public void Main()	Khởi tạo 2 giá trị ngẫu nhiên vào đầu tiên và khi thực hiện lệnh new game.Khởi tạo stack mảng và stack điểm
4	private void Form1_Load()	Chạy hàm LoadGame()
5	private void Form1_Paint()	Hiển thị các con số vào Label
6	private void RanDomNumbers()	Tạo ra 2 giá trị tại vị trí x,y bất kì trong bảng
7	private void SaveToStack()	Lưu giá trị của mảng và điểm vào stack



8	public void Check(bool Temp)	Lệnh kiểm tra các bước di chuyển có thực hiện không. Nếu có thực hiện sẽ tạo ra 1 số ngẫu nhiên ,nếu không sẽ pop mảng ra
9	private void UP(), DOWN(), RIGHT(), LEFT()	Lưu các bước đã đi vào mảng. Đưa các giá trị lên trên, xuống dưới, sang phải, sang trái. Nếu 2 giá trị bằng nhau thì sẽ cập nhật ô mới có giá trị gấp đôi sau đó thực hiện lệnh Check()
10	private void Form1_KeyDown()	Sử dụng event KeyDown để dùng các phím điều hướng mũi tên. Thông báo GameOver!
11	private bool CheckGameOver()	Xử lý game over !
12	private void SetColor(int x, int y)	Đặt màu cho các số 2,4,8, 16,...
13	private void NewGame_Click()	Tạo mới lại Game
14	private void Undo_Click()	Quay lại bước đi trước đó
15	private void Form1_FormClosing()	Kiểm tra có muốn thoát game không

*Bảng 3. Các phương thức được sử dụng trong lớp from.cs*

### 5.3. Ứng dụng Stack và giải thích

- *Khai báo Stack*

```
Stack<int[,]> pushnumber = new Stack<int[,]>();
Stack<int> pushscore = new Stack<int>();
```

- *Khi thực hiện hàm Main sẽ tạo ngẫu nhiên ra 2 số thì khởi tạo 2 stack mới : stack điểm và stack mảng*

```
public void Main()
{
    for (int x = 0; x < 4; x++)
        for (int y = 0; y < 4; y++)
            Boardsize[x, y] = 0;
    RandomNumbers();
    RandomNumbers();
    Score = 0;
    pushnumber = new Stack<int[,]>();
    pushscore = new Stack<int>();
    this.Refresh();
}
```

- *Hàm SaveToStack sẽ tạo ra 1 mảng x mới có các giá trị bằng mảng Boardsize .Khi đó push mảng x vào stack mảng và push điểm vào stack điểm*

```
private void SaveToStack()
{
    int[,] x = new int[4, 4];
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            x[i, j] = Boardsize[i, j];
    pushnumber.Push(x);
    pushscore.Push(Score);
}
```

- *Hàm Check kiểm tra xem nếu thực hiện được những bước đi thì Temp sẽ bằng true .Nếu Temp bằng true thì sẽ sinh ra ngẫu nhiên 1 số. Ngược lại sẽ pop mảng trong stack ra*

```
public void Check(bool Temp)
{
    if ((Temp) == true)
    {
        RandomNumbers();
    }
    else
    {
        pushnumber.Pop();
        pushscore.Pop();
    }
}
```

- *Khi thực hiện các bước di chuyển mà muốn quay lại bước đi trước ta nhấn undo, hàm undo\_Click. Nếu số lượng stack trong mảng lớn hơn 0 thì khởi*

tạo 1 mảng x mới, pop mảng trong stack và pop stack điểm ra chính bằng mảng x và cho lần lượt các giá trị trong mảng Boardsize bằng mảng x, cập nhật lại điểm.

```
private void Undo_Click(object sender, EventArgs e)
{
    if (pushnumber.Count > 0)
    {
        int[,] x = new int[4, 4];
        x = pushnumber.Pop();
        Score = pushscore.Pop();
        for (int i = 0; i < 4; i++)
            for (int j = 0; j < 4; j++)
                Boardsize[i, j] = x[i, j];
        lb_Score.Text = Score.ToString();
        this.Refresh();
    }
}
```

- Khi nhấn bất kì nút di chuyển thì việc đầu tiên là thực hiện hàm *SaveToStack* lưu trạng thái trước đó vào stack mảng. Đưa giá trị lên trên hoặc gộp 2 giá trị bằng nhau sẽ đánh dấu *Temp = true* tiếp theo là thực hiện hàm *Check* nếu *Temp = true* thì tạo ngẫu nhiên 1 số, nếu không sẽ pop ra 1 mảng trong stack. Tương tự các bước di chuyển *down*, *left*, *right* thực hiện tuần tự.

```
private void UP()
{
    SaveToStack();
    bool Temp = false;
    for (int x = 0; x < 4; x++)
    {
        for (int y = 0; y < 3; y++)
        {
            for (int y1 = y + 1; y1 < 4; y1++)
            {
                if (Boardsize[x, y1] > 0)
                {
                    if (Boardsize[x, y] == 0)
                    {
                        Boardsize[x, y] = Boardsize[x, y1];
                        Boardsize[x, y1] = 0;
                        y--;
                        Temp = true;
                    }
                    else if (Boardsize[x, y] == Boardsize[x, y1])
                    {
                        Boardsize[x, y] *= 2;
                        Boardsize[x, y1] = 0;
                        Score += Boardsize[x, y];
                    }
                }
            }
        }
    }
}
```

```
        Temp = true;
    }
    break;
}
}
}
}
Check(Temp);
}
```

## 6. KẾT LUẬN

### 6.1. Mức độ hoàn thành

- Chọn đề tài, phân tích đề tài: Hoàn thành 100%.
- Thiết kế Form giao diện game: Hoàn thành 100%.
- Code cho game: Hoàn thành 90%.
  - Còn chưa code được hàm khi game đạt số 2048 sẽ thông báo bạn đã thắng và có thể lựa chọn chơi tiếp hoặc bắt đầu lại.
  - Chưa tạo được level game như 6x6, 8x8,...

### 6.2. Các khó khăn gặp phải

- Thiết kế các label giống nhau bằng cách kéo thả làm mất thời gian
  - + Cách khắc phục : thiết kế bằng cách tạo code làm cho việc viết hàm và thực hiện các lệnh dễ dàng trực quan hơn
- Khi khởi tạo và thực hiện các bước di chuyển thì push các giá trị ngay vào stack nên khi undo pop giá trị ra bị sai khi áp dụng vào hàm UnDo\_Click
  - + Cách khắc phục: khi khởi tạo hàm Main thì khởi tạo ra một stack mới khi ta thực hiện hàm di chuyển việc đầu tiên là *lưu mảng trước đó* vào stack sau khi thực hiện lệnh Temp = true thì tạo ngẫu nhiên 1 số mới nhưng vẫn chưa lưu trạng thái mới. Nếu lệnh Temp = false thì pop mảng đó ra, và khi thực hiện các bước di chuyển tiếp theo thì ta mới lưu mảng trước nên việc pop giá trị ra ở hàm UnDo\_Click mới đúng
- Khi thực hiện lệnh unDo chưa unDo được điểm
  - + Cách khắc phục: cho 1 stack điểm vào tương tự như stack mảng
- Khi chơi thì điểm cao bị mất sau khi mở lại game
  - + Cách khắc phục: tạo thêm 1 biến string để lưu giá trị đọc được từ file txt

### 6.3. Ưu điểm, nhược điểm của Game

- Ưu điểm:
  - + Dễ chơi, chỉ cần dùng cái phím điều hướng mũi tên.
  - + Luyện khả năng tư duy tính toán.

- + Game nhẹ nhàng, không tốn tài nguyên máy.
- Nhược điểm:
  - + Game chưa cho chạy song song điểm cao và điểm đang đạt được.
  - + Chưa tăng được level cho game: 6x6 , 8x8,...

#### **6.4. Ý tưởng phát triển**

- Tạo bảng xếp hạng để lưu điểm cao.
- Phát triển game thành 6x6, 8x8, để tăng độ khó cho game.
- Đưa lên giao diện web để nhiều người chơi hơn.

## **TÀI LIỆU THAM KHẢO**

[1],[2],[3] [https://vi.wikipedia.org/wiki/2048\\_\(tr%C3%B2\\_ch%C6%A1i\)](https://vi.wikipedia.org/wiki/2048_(tr%C3%B2_ch%C6%A1i))