



**HCMUTE**

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

## BỘ MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

# **BÁO CÁO ĐỒ ÁN CUỐI KÌ** **ỨNG DỤNG STACK VÀO** **TÍNH NĂNG UNDO** **TRONG GAME 2048**

Nhóm sinh viên thực hiện :

Phạm Thị Thanh Hằng\_17110132

Nguyễn Công Chiến\_16110025

Giảng viên hướng dẫn :ThS. Trần Công Tú

Tp. Hồ Chí Minh, tháng 12 năm 2018

## Contents

LỜI CẢM ƠN.....	1
DANH MỤC CÁC HÌNH .....	2
DANH MỤC CÁC BẢNG.....	3
1.GIỚI THIỆU.....	4
1.1. Game 2048 là gì? .....	4
1.2. Cách chơi game.....	4
1.3. Mục tiêu đề ra .....	5
1.4. Lí do chọn đề tài .....	5
1.5. Ngôn ngữ cài đặt.....	5
2.ĐẶC TẢ, PHÂN TÍCH VỀ GAME 2048 .....	6
2.1.Mô tả về game .....	6
2.2.Phân tích game .....	6
3. KẾ HOẠCH THỰC HIỆN DỰ KIẾN.....	6
4. GIAO DIỆN PHẦN MỀM VÀ MÔ TẢ CHỨC NĂNG.....	7
4.1. Giao diện ngoài .....	7
4.2. Chức năng trong game .....	7
5. THIẾT KẾ CODE.....	10
5.1. Các lớp được sử dụng trong chương trình: .....	10
5.2. Các phương thức Form1.cs: .....	10
5.3. Tính năng stack được sử dụng trong game 2048.....	13
5.4. Giải thích code về stack .....	14
6. KẾT LUẬN.....	18
6.1. Mức độ hoàn thành .....	18
6.2. Các khó khăn gặp phải.....	18
6.3. Ưu điểm, nhược điểm của Game .....	18
6.4. Ý tưởng phát triển .....	19
TÀI LIỆU THAM KHẢO .....	20

## 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 tính năng undo trong 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, 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

Hình 1.Kế hoạch công việc dự kiến .....	6
Hình 1. Giao diện game.....	7
Hình 2: Tính năng Help.....	8
Hình 3: Lưu điểm cao.....	8
Hình 4: Game over .....	9
Hình 5: Game đang chơi.....	9
Hình 6: Undo lại bước đi.....	10

## DANH MỤC CÁC BẢNG

Bảng 1: Kế hoạch công việc dự kiến.....	6
Bảng 2: Các lớp được sử dụng trong chương trình .....	10
Bảng 3: Các phương thức được sử dụng trong form.cs.....	12

# 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. [1]
- 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. [2]

## 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. [3]

### *1.3. Mục tiêu đề ra*

Xây dựng game 2048 có các chức năng như sau:

- Thiết kế giao diện dễ nhìn.
- 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*

Nói đến game thì chắc phần lớn mọi người đều đồng ý đó là một loại hình giải trí có sức thu hút rất lớn, nhất là với giới trẻ hiện nay. Đặc biệt sau sự ra đời của Internet, game online và offline xuất hiện và nhanh chóng gây được sức hấp dẫn to lớn. Với tựa game 2048, không cần phải đồ họa bóng bẩy, cách chơi phức tạp... chỉ với đồ họa không có gì đặc biệt lắm cùng với lối chơi đơn giản nhưng vô cùng cuốn hút và đã phát sốt khi mới ra mắt. 2048 vừa có thể chơi trên web vừa có thể chơi dưới dạng các ứng dụng trên một số hệ điều hành di động phổ biến. Với 2048, bạn có thể giải trí, xả stress sau những giờ làm việc căng thẳng chinh phục, rèn luyện óc tư duy chinh phục game “ức chế” này. Với sự hấp dẫn của 2048 và những thách thức còn đang phía trước, với niềm đam mê công nghệ phần mềm và những ứng dụng game tuyệt vời, với khao khát khám phá và chinh phục những tri thức mới mẻ... chúng em đã chọn đề tài “ Ứng dụng stack vào tính năng undo trong game 2048 ”, để hiểu được cách thức hoạt động của game và áp dụng được kiến thức đã học của bộ môn Cấu trúc dữ liệu và giải thuật, ứng dụng ngăn xếp (Stack) vào đề tài của mình.

### *1.5. Ngôn ngữ cài đặt*

Để phục vụ cho việc thiết kế game và các sự kiện (event) trong game dễ dàng, việc viết code vô cùng trực quan chúng tôi chọn Window Form Application trong ngôn ngữ Microsoft C#.NET với phiên bản .NET Framework 4.5.2; IDE và compiler nằm trong bộ Microsoft Visual Studio 2017 để cài đặt game này.

## 2.ĐẶC TẢ, PHÂN TÍCH VỀ GAME 2048

### 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. KẾ HOẠCH THỰC HIỆN DỰ KIẾN

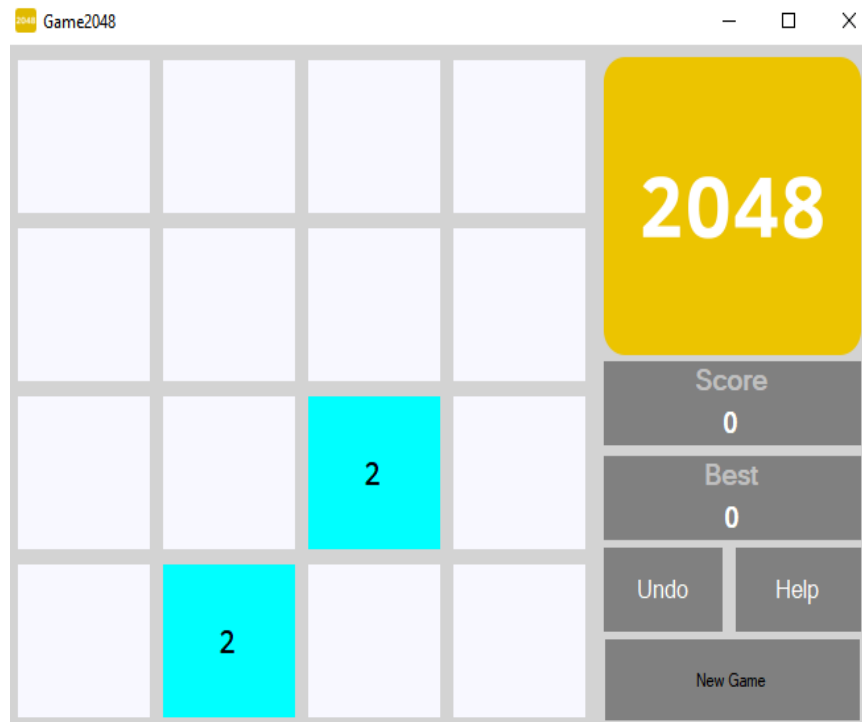
BẢNG KẾ HOẠCH ĐỒ ÁN GAME 2048					
STT	TÊN CÔNG VIỆC	NGƯỜI 1	NGƯỜI 2	NGÀY BẮT ĐẦU	NGÀY KẾT THÚC
1	1.Xây dựng ý tưởng		Chiến	21-9	
2	2.Mô tả game	Hằng			28-9
3	3.Tìm hiểu các hàm cơ bản:Textcolor,gotoxy,resizeconsole,xóa màn hình,...	Hằng		28-9	5-10
4	4.Tìm hiểu winform và tạo form 2048,tạo label và điền giá trị vào label	Hằng		5-10	11-10
5	5.Tìm hiểu các event game(up,down,left,right...)		Chiến	12-10	19-10
6	6.Hàm lưu các bước đi và hàm lưu		Chiến	19-10	26-10
7	7.Xử lý game over		Chiến	26-10	3-11
8	8.Test game	Hằng		3-11	7-11
9	9.Làm file word		Chiến	7-11	
10	10.Làm powerpoint	Hằng			14-11

Bảng 1.Kế hoạch công việc dự kiến



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

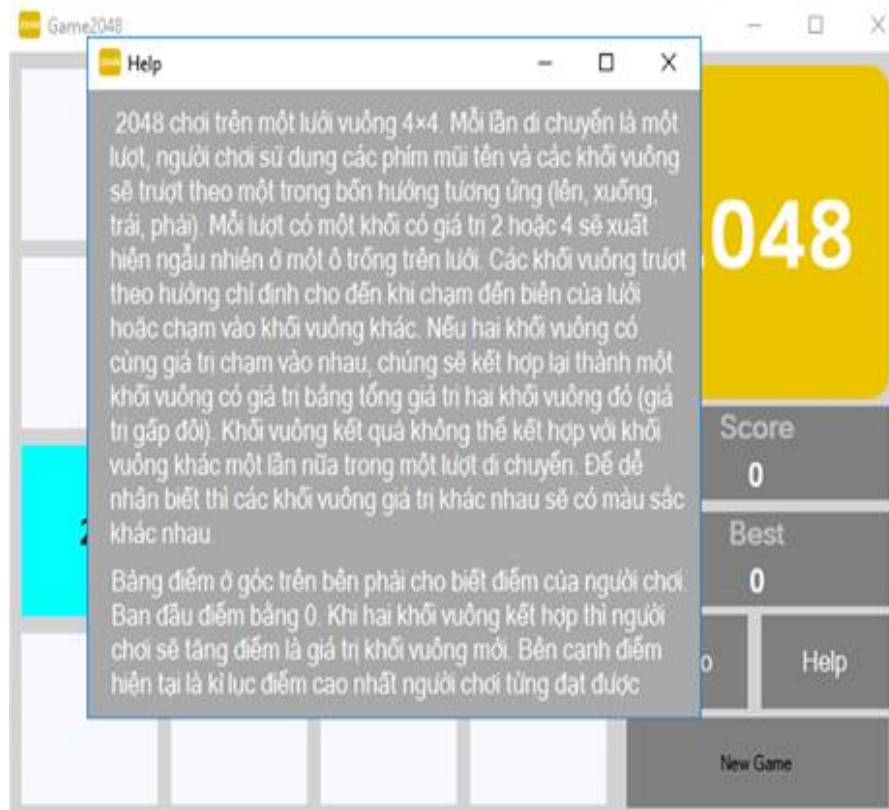
### 4.1. Giao diện ngoài



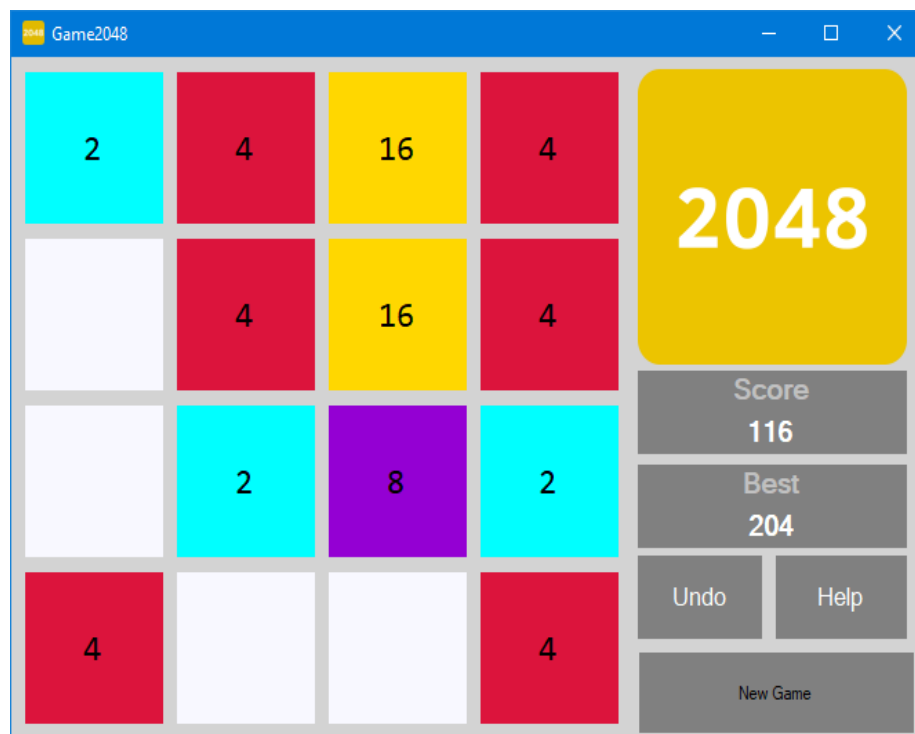
Hình 1. Giao diện game

### 4.2. Chức năng trong game

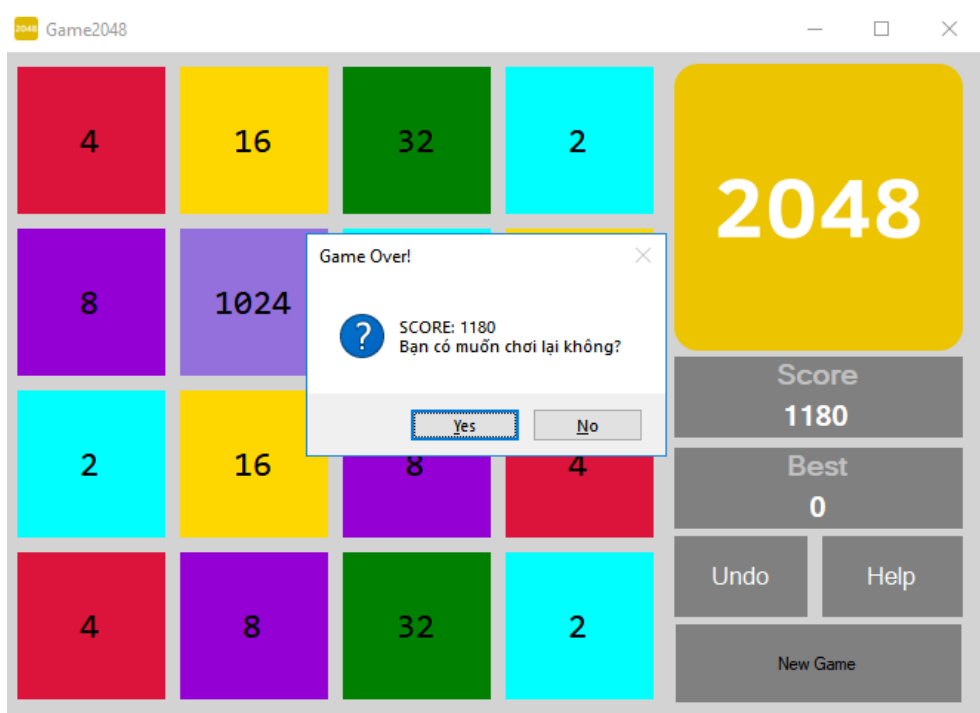
- **New Game** để tạo mới lại game.
- **Help** dùng để mô tả cách chơi.
- **Undo** để quay lại bước đi trước đó.
- **Score** để báo điểm hiện tại của người chơi.
- **Best** để báo điểm cao nhất của người chơi.
- Khi lấp đầy tất cả các ô thì sẽ hiển thị thông báo như sau :  
Có 2 sự lựa chọn là **Yes** hoặc **No**:
  - ✓ Chọn **Yes** để chơi lại game mới.
  - ✓ Chọn **No** để thoát game.



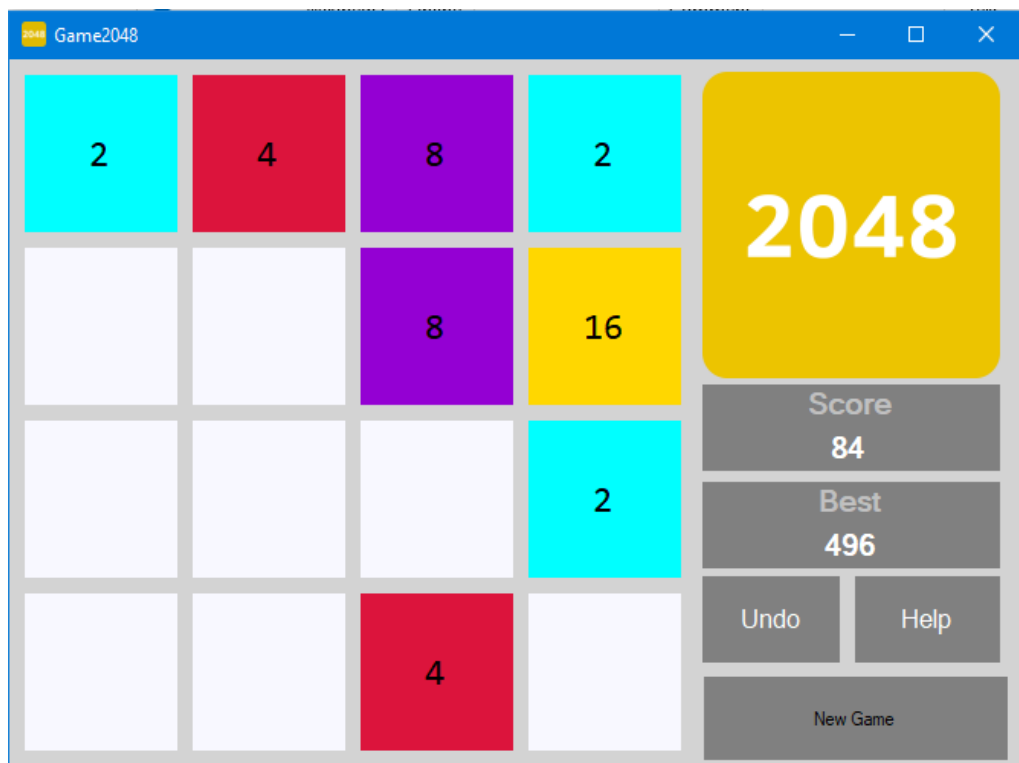
Hình 2: Tính năng Help



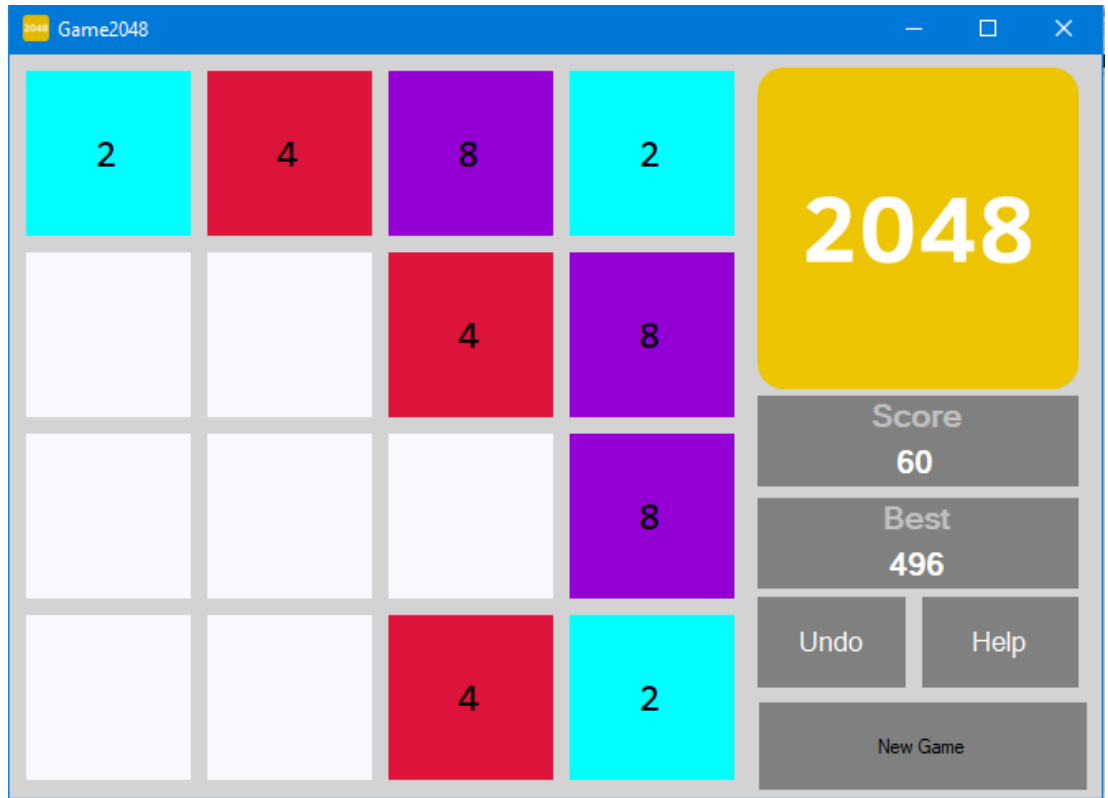
Hình 3: Lưu điểm cao



Hình 4: Game over



Hình 5: Game đang chơi



Hình 6: Undo lại bước đi

## 5. THIẾT KẾ CODE

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

TT	Tên Class	Mục đích
1	Help.cs [Design]	Hướng dẫn chơi game
2	Form1.cs	Các hàm xử lý trong game
3	Form1.cs [Design]	Design giao diện game

Bảng 2: Các lớp được sử dụng trong chương trình

### 5.2. Các phương thức Form1.cs:

TT	Phương Thức	Mục Đích	Tên file, số thứ tự chứa dòng khai báo
	<code>public Form1()</code>	Hàm khởi tạo Các giá trị của biểu mẫu	Form1.cs(25)
1	<code>private void Form1_Load(object sender, EventArgs e)</code>	Thiết kế label , các biểu mẫu vào form và xuất hiện ngẫu nhiên 2 số hiển thị lần đầu tiên	Form1.cs(29)
2	<code>private void Form1_design(object sender, PaintEventArgs e)</code>	Hiển thị các con số vào Label	Form1.cs(46)
3	<code>public void datMau(int i, int j)</code>	Vẽ màu	Form1.cs(64)
4	<code>public void creatRandomCard()</code>	Tạo ra 2 giá trị tại vị trí x,y bất kì	Form1.cs(80)
5	<code>public void init()</code>	Khởi tạo 2 giá trị ngẫu nhiên vào đầu tiên và khi thực hiện lệnh new game.Khởi tạo stack mảng và stack điểm	Form1.cs(94)
	<code>public void luuVaoStack()</code>	Lưu giá trị của mảng và điểm vào stack	Form1.cs(106)
	<code>public void kiemTra(bool isDo)</code>	Lệnh kiểm tra các bước di chuyển có thực hiện ko.Nếu có thực hiện sẽ tạo ra 1 số ngẫu nhiên ,nếu không sẽ pop mảng ra	Form1.cs(115)
6	<code>public bool Up()</code>	Lưu các bước đã đi vào mảng Đưa các giá trị lên trên cùng, nếu 2 giá trị bằng nhau thì sẽ cập nhật ô mới có giá trị gấp đôi sau đó thực hiện lệnh kiemTra	Form1.cs(127)

7	<code>public bool Down()</code>	Lưu các bước đã đi vào mảng. Đưa các giá trị xuống dưới cùng, nếu 2 giá trị bằng nhau thì sẽ cập nhật ô mới có giá trị gấp đôi sau đó thực hiện lệnh kiểm tra	Form1.cs(160)
8	<code>public bool Left()</code>	Lưu các bước đã đi vào mảng. Đưa các giá trị sang trái, nếu 2 giá trị bằng nhau thì sẽ cập nhật ô mới có giá trị gấp đôi sau đó thực hiện lệnh kiểm tra	Form1.cs(193)
9	<code>public bool Right()</code>	Lưu các bước đã đi vào mảng. Đưa các giá trị sang phải, nếu 2 giá trị bằng nhau thì sẽ cập nhật ô mới có giá trị gấp đôi sau đó thực hiện lệnh kiểm tra	Form1.cs(226)
10	<code>bool overgame()</code>	Xử lý GameOver!	Form1.cs(259)
11	<code>private void Form1_KeyDown(object sender, EventArgs e)</code>	Sử dụng event KeyDown để dùng các phím điều hướng mũi tên. Thông báo GameOver!	Form1.cs(273)
12	<code>private void undo_Click(object sender, EventArgs e)</code>	Quay lại bước đi trước đó	Form1.cs(295)
13	<code>private void newGame_Click(object sender, EventArgs e)</code>	Tạo mới lại Game	Form1.cs(309)
14	<code>private void help_Click(object sender, EventArgs e)</code>	Hướng dẫn người chơi	Form1.cs(318)

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

### 5.3. Tính năng stack được sử dụng trong game 2048

```
Stack<int[,]> st = new Stack<int[,]> ();
Stack<int> st_score = new Stack<int>();
public void init()
{
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            hide[i, j] = 0;
    creatRandomCard();
    creatRandomCard();
    score = 0;
    st = new Stack<int[,]>();
    st_score = new Stack<int>();
    this.Refresh();
}

public void luuVaoStack()
{
    int[,] k = new int[4, 4];
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            k[i, j] = hide[i, j];
    st.Push(k);
    st_score.Push(score);
}

public void kiemTra( bool isDo)
{
    if ((isDo) == true)
    {
        creatRandomCard();
    }
    else
    {
        st.Pop();
        st_score.Pop();
    }
}

public void Up()
{
    luuVaoStack();
    bool isDo=false;
    for (int x = 0; x < 4; x++)
    {
        for (int y = 0; y < 3; y++)
```

```

{
    for (int y1 = y + 1; y1 < 4; y1++)
    {
        if (hide[x, y1] > 0)
        {
            if (hide[x, y] == 0)
            {
                hide[x, y] = hide[x, y1];
                hide[x, y1] = 0;
                y--;
                isDo = true;
            }
            else if (hide[x, y] == hide[x, y1])
            {
                hide[x, y] *= 2;
                hide[x, y1] = 0;
                score += hide[x, y];
                isDo = true;
            }
            break;
        }
    }
}
kiemTra( isDo);
}

```

```

private void undo_Click(object sender, EventArgs e)
{
    if (st.Count > 0)
    {
        int[,] k = new int[4, 4];
        k = st.Pop();
        score = st_score.Pop();
        for (int i = 0; i < 4; i++)
            for (int j = 0; j < 4; j++)
                hide[i, j] = k[i, j];
        diem.Text = score.ToString();
        this.Refresh();
    }
}

```

#### 5.4. Giải thích code về stack

- Đầu tiên ta khai báo 2 stack : stack lưu mảng và stack lưu điểm

```

Stack<int[,]> st = new Stack<int[,]> ();
Stack<int> st_score = new Stack<int>();

```



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

```
public void init()
{
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            hide[i, j] = 0;
    creatRandomCard();
    creatRandomCard();
    score = 0;
    st = new Stack<int[,]>();
    st_score = new Stack<int>();
    this.Refresh();
}
```

- Hàm LưuVaoStack sẽ tạo ra 1 mảng k mới có các giá trị bằng mảng hide .Khi đó push mảng k vào stack mảng và push điểm vào stack điểm.

```
public void lưuVaoStack()
{
    int[, ] k = new int[4, 4];
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            k[i, j] = hide[i, j];
    st.Push(k);
    st_score.Push(score);
}
```

- Hàm kiểmTra kiểm tra xem nếu thực hiện được những bước đi thì isDo sẽ bằng true .Nếu isDo bằng true thì sẽ sinh ra ngẫu nhiên 1 số.Nếu isDo không bằng true thì sẽ pop mảng trong stack ra

```

public void kiemTra( bool isDo)
{
    if ((isDo) == true)
    {
        creatRandomCard();
    }
    else
    {
        st.Pop();
        st_score.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 k mới, pop mảng trong stack và pop stack điểm ra chính bằng mảng k và cho lần lượt các giá trị trong mảng hide bằng mảng k, cập nhật lại điểm.

```

private void undo_Click(object sender, EventArgs e)
{
    if (st.Count > 0)
    {
        int[,] k = new int[4, 4];
        k = st.Pop();
        score = st_score.Pop();
        for (int i = 0; i < 4; i++)
            for (int j = 0; j < 4; j++)
                hide[i, j] = k[i, j];
        diem.Text = score.ToString();
        this.Refresh();
    }
}

```

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

```
public void Up()
{
    lưuVaoStack();
    bool isDo=false;
    for (int x = 0; x < 4; x++)
    {
        for (int y = 0; y < 3; y++)
        {
            for (int y1 = y + 1; y1 < 4; y1++)
            {
                if (hide[x, y1] > 0)
                {
                    if (hide[x, y] == 0)
                    {
                        hide[x, y] = hide[x, y1];
                        hide[x, y1] = 0;
                        y--;
                        isDo = true;
                    }
                    else if (hide[x, y] == hide[x, y1])
                    {
                        hide[x, y] *= 2;
                        hide[x, y1] = 0;
                        score += hide[x, y];
                        isDo = true;
                    }
                    break;
                }
            }
        }
    }
    kiểmTra( isDo);
}
```

## 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 80%.
  - 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 lưu điểm cao mãi mãi được.

Mức độ hoàn thành: ~90%

### 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 init 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 isDo true thì tạo ngẫu nhiên 1 số mới nhưng vẫn chưa lưu trạng thái mới. Nếu lệnh isDo 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
- +st.Count tách ra thành st.Count =1 và st.Count >1 .st.Count =1 để peek giá trị ra.st.Count>1 thì thực hiện 2 lần pop giá trị
- Cách khắc phục : chỉ cần st.Count >0 rồi pop 1 lần giá trị ra
- +Khi thực hiện lệnh unDo chưa unDo được điểm
- Cách khắc phục :cho 1 stack điểm vào tương tự như stack mảng

### 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 lưu điểm cao mãi mãi được.
  - Chưa làm được tổ hợp phím Ctrl+Z để thay thế cho Click Undo.

#### *6.4. Ý tưởng phát triển*

- Tạo bảng xếp hạng để lưu điểm cao.
- Phát triển game thành 6x6, 9x9 để tăng độ khó cho game.
- Tạo sever cho mọi người chơi cùng chơi và solo về mức điểm cao của mình.
- Design lại giao diện nhìn bắt mắt hơn
- Thiết kế bắt mắt 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))