

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

ĐỒ ÁN 1

XÂY DỰNG GAME XẾP HÌNH

SVTH1 : TRỊNH NGUYỄN HOÀNG VĨNH PHÚC - 17110208

SVTH2 : HỒ QUỐC ĐẠT - 17110117

GVHD : ThS. TRẦN CÔNG TÚ

TP. Hồ Chí Minh, tháng 11 năm 2019

MỤC LỤC

LỜI MỞ ĐẦU	1
Danh mục các hình	2
Danh mục các bảng.....	2
CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN.....	3
1.1. Lịch sử hình thành.....	3
1.2. Giới thiệu game xếp hình	3
1.3. Lý do, mục đích, mục tiêu chọn đề tài	3
1.3.1. Lý do, mục đích chọn đề tài.....	3
1.3.2. Mục tiêu chọn đề tài.....	4
1.4. Lựa chọn ngôn ngữ cài đặt mô phỏng.....	4
CHƯƠNG 2 : NỘI DUNG	5
2.1. Quá trình và công việc thực hiện đồ án.....	5
2.2. Luật chơi.....	5
2.3. Thiết kế giao diện	6
2.4. Cách cài đặt và viết chương trình.....	6
2.5 Sản phẩm của đồ án.....	13
CHƯƠNG 3 : KẾT LUẬN.....	16

3.1 Kết luận	16
3.2 Ưu điểm.....	16
3.3 Nhược điểm.....	16
3.4 Đề xuất phương án cải thiện.....	16
PHỤ LỤC	18
TÀI LIỆU THAM KHẢO	20

LỜI MỞ ĐẦU

Để đồ án “Xây dựng game xếp hình” đạt kết quả tốt, nhóm đã cố gắng thực hiện và nhận được sự giúp đỡ tận tình của thầy Trần Công Tú. Nhóm đã khảo sát, xây dựng dữ liệu và tổng hợp thông tin từ nhiều hướng khác nhau. Trong quá trình hoàn thành chương trình, vì trình độ còn hạn chế nên có nhiều khó khăn làm chậm tiến độ. Mặt khác, đề tài tuy đã có sự giúp đỡ của GVHD nhưng vẫn không thể tránh khỏi những sai sót.

Do đó, nhóm mong sẽ nhận được sự góp ý chân thành của thầy và các bạn đọc. Những ý kiến đóng góp về đề tài sẽ được nhóm tiếp nhận, rút kinh nghiệm khi làm các đề tài nghiên cứu khác. Một lần nữa, nhóm em xin cảm ơn Thầy Trần Công Tú và các bạn.

Danh mục các hình

Hình 1 Bảng phân công	5
Hình 2 Thiết kế giao diện ban đầu.....	6
Hình 3 hàm UP().....	8
Hình 4 Hàm DOWN()	9
Hình 5 Hàm LEFT()	10
Hình 6 Hàm swap().....	11
Hình 7 Hàm RIGHT().....	11
Hình 8 Hàm ran() trong class RandomPlay.....	12
Hình 9 Hàm SwapRan() trong class RandomPlay	13
Hình 10 Giao diện chính.....	14
Hình 11 Giao diện nhập điểm.....	14
Hình 12 Giao diện người chiến thắng	15
Hình 13 Giao diện giới thiệu về Game.....	15

Danh mục các bảng

Bảng 1 Phân công công việc	19
----------------------------------	----

CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN

1.1. Lịch sử hình thành

Ban đầu loại game xếp hình (sliding puzzle) cổ nhất là mười lăm câu đố (4x4), được phát minh bởi Noyes Chapman vào năm 1880; Sam Loyd thường bị ghi sai khi làm cho các câu đố trượt trở nên phổ biến dựa trên tuyên bố sai lầm của mình rằng ông đã phát minh ra mười lăm câu đố. Phát minh của Chapman đã khởi đầu một cơn sốt câu đố vào đầu những năm 1880. Từ những năm 1950 đến những năm 1980, các câu đố trượt sử dụng các chữ cái để tạo thành các từ rất phổ biến. Những loại câu đố này có một số giải pháp khả thi, có thể được nhìn thấy từ các ví dụ như Ro-Let (một câu đố mười lăm dựa trên chữ cái), Scribe-o (4x8) và Lingo. [1]

1.2. Giới thiệu game xếp hình

Game xếp hình là game mà nó sẽ chia bức ảnh có sẵn thành nhiều ô vuông nhỏ có kích thước bằng nhau. Nhiệm vụ của người chơi là phải di chuyển các phần này, sao cho mỗi phần nằm đúng vị trí của nó và hoàn thành lại như bức hình gốc. Lưu ý: người chơi sẽ có một ô trống để di chuyển và chỉ di chuyển các ô nằm cạnh bên ô trống đó.. Các mảnh được di chuyển có thể bao gồm các hình đơn giản, hoặc chúng có thể được in bằng màu sắc, hoa văn, các phần của một bức tranh lớn hơn (như trò chơi ghép hình), số hoặc chữ cái.

1.3. Lý do, mục đích, mục tiêu chọn đề tài

1.3.1. Lý do, mục đích chọn đề tài

Sau khi học môn cấu trúc dữ liệu và lập trình hướng đối tượng, nhóm được tiếp xúc nhiều hơn với các khối kiến thức cơ sở, điều đó giúp nhóm có thể xây dựng các chương trình trên máy tính, áp dụng các kiểu dữ liệu trong thực tế. Thông qua đó nhóm vận dụng

kiến thức về lập trình hướng đối tượng và cơ sở dữ liệu để demo game xếp hình. Với mục đích xây dựng game xếp hình với tính tiện dụng, gần gũi, dễ chơi và tính giải trí cao, phù hợp với mọi lứa tuổi, bám sát kiến thức đã học. Và quan trọng là hiểu rõ hơn về các thuật toán sử dụng, tìm hiểu thêm về thiết kế giao diện trong game, rèn luyện kỹ năng lập trình, làm tiền đề cho các môn học sau này và ứng dụng mai sau. Vì vậy nhóm quyết định chọn đề tài game xếp hình để thực hiện đồ án 1.

1.3.2. Mục tiêu chọn đề tài

Với mục tiêu tạo ra một game caro có đầy đủ chức năng giúp người chơi dễ dàng sử dụng như lưu điểm, newgame, lựa chọn mức độ khó, tính số bước đi... Đặc biệt, áp dụng lập trình hướng đối tượng để xây dựng game, kết nối cơ sở dữ liệu để lưu điểm người chơi.

1.4. Lựa chọn ngôn ngữ cài đặt mô phỏng

Vì chương trình mô phỏng của nhóm biểu diễn thuật toán dưới dạng các đối tượng thuộc tính, hành vi rõ ràng nên nhóm quyết định sử dụng ngôn ngữ Microsoft C#.NET với phiên bản .NET Framework 4.7.2; IDE và compiler nằm trong bộ Microsoft Visual Studio 2019 để cài đặt mô phỏng này.

CHƯƠNG 2 : NỘI DUNG

2.1. Quá trình và công việc thực hiện đồ án

KẾ HOẠCH THỰC HIỆN ĐỒ ÁN 1 - 2019							
Đề tài : Game Xếp hình							
DANH SÁCH CÔNG VIỆC	NỘI DUNG CHI TIẾT CÔNG VIỆC	Hồ Quốc Đạt	Trịnh Nguyễn Hoàng Vinh Phúc	Ngày bắt đầu (dự kiến)	Ngày kết thúc (dự kiến)	Ngày bắt đầu (thực tế)	Ngày kết thúc (thực tế)
Khảo sát giao diện game							
	Giao diện nhập tên người chơi + tài khoản (nếu có)	X		5/10/2019	9/10/2019		
	Tìm hiểu cách tạo ảnh duy chuyển trong bản cờ game		X	10/10/2019	13/10/2019		
	Tìm hiểu các botton điều khiển cần thiết	X		13/10/2019	14/10/2019		
	Phát sinh cần thiết trong quá trình làm.	X	X	14/10/2019	20/10/2019		
Triển khai Code hoạt động cho game							
	tìm hiểu lập trình hướng đối tượng cho game	X	X	20/10/2019	22/10/2019		
	triển khai code hoạt động của game thông qua botton và click mouse		X	22/10/2019	27/10/2019		
	Triển khai code nhập lưu sửa xóa trong DataServer	X		27/10/2019	29/10/2019		
	Triển Khai code tính thời gian và điểm cho người chơi, Lưu kết quả sau khi người chơi đã chiến thắng vào DataServer	X		29/10/2019	31/10/2019		
	Phát sinh cần thiết trong quá trình làm.	X	X	31/10/2019	5/11/2019		
Kiểm tra Hoạt động của Game và xử lý các trường hợp có thể xảy ra lỗi trong quá trình chơi		X	X	5/11/2019	10/11/2019		

Hình 1 Bảng phân công

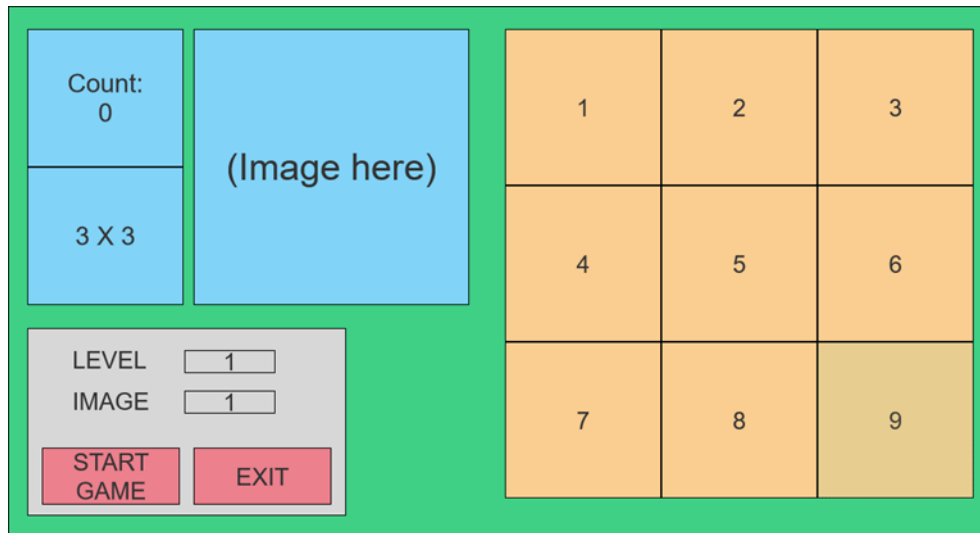
2.2. Luật chơi

Khi bắt đầu người chơi sẽ chọn độ khó và hình mà mình muốn chơi. Có ba mức độ chơi cho phép người dùng chọn : 1 (3x3), 2 (4x4), 3 (5x5).

Người chơi sẽ có một ô trống để di chuyển và chỉ di chuyển các ô nằm cạnh bên ô trống đó. Di chuyển các phần ô vuông bằng cách nhấp trái chuột vào nó để dịch chuyển sang ô trống kế bên, còn vị trí cũ sẽ trống. Các ô vuông có chứa phần ảnh, mà không có ô trống kế bên thì không thể di chuyển.

Điều kiện thắng : mỗi phần nằm đúng vị trí của nó và hoàn thành lại như bức hình gốc.

2.3. Thiết kế giao diện



Hình 2 Thiết kế giao diện ban đầu

Chương trình cho phép người chơi có thể chọn trò chơi mới (new game), chọn bật nhạc hay tắt nhạc, chọn độ khó chơi, chọn hình. Phía góc trái có thể hiện số bước người chơi đã đi.

Chức năng chọn hình cho phép người chơi chọn hình mình muốn hiển thị khi chơi (các hình được chọn dựa trên tài nguyên có sẵn của trò chơi).

Khi chiến thắng game sẽ cho phép người nhập tên của người chơi.

Khi người chơi thắng thì hiển thị ra thông báo thắng và cho phép chọn các tùy chọn lưu lại thành tích hay thoát.

Bên cạnh đó còn bao gồm phần xem lại thành tích cao của người chơi khi chơi game.

2.4. Cách cài đặt và viết chương trình

Đồ án của nhóm có 4 Form và 6 class :

- Form :

+ Form1 : *Form chính, bao gồm bàn chơi và những phím chức năng.*

+ YourNameWinner : *Form lưu điểm khi chơi*

+ ScorePlayList : *Form hiển thị thành tích người chơi.*

+ AboutGame : *Form hiển thị thông tin về game.*

- Class :

+ Class LoadButton: *lớp tạo button và các hàm trong nó.*

+ Class LoadCropImgae: *lớp xử lý cắt hình ảnh.*

+ Class LoadImageOriginal: *lớp xử lý hình ảnh tài nguyên.*

+ Class LoadSize: *lớp xử lý các mức game có thể khởi tạo (độ khó)*

+ Class MovePlay: *lớp xử lý di chuyển của button.*

+ Class RandomPlay: *lớp khởi tạo vị trí ngẫu nhiên các button.*

+ Class Program: *chương trình chính.*

- Một số hàm quan trọng :

Hàm UP():

- Hàm sẽ nhận vào giá trị của 2 button : ô trống ảnh và ô có ảnh.
- 2 Button sẽ kiểm tra xem có nằm đúng vị trí, đảm ứng luật chơi của game.
- Hàm sẽ trả về giá trị true nếu như vị trí của ô chứa ảnh nằm trên vị trí của ô trống. ngược lại sẽ trả về giá trị false.

```

public bool UP( Button Empty , Button full, LoadSize getsize)
{
    _size = getsize;
    emptyX = Empty.Location.X;
    emptyY = Empty.Location.Y;

    X = full.Location.X;
    Y = full.Location.Y;

    tabFull = full.TabIndex;
    tabEmpty = Empty.TabIndex;

    if (tabFull == (tabEmpty - 1) ||
        tabFull == (tabEmpty - _size.getSize()) ||
        tabFull == (tabEmpty + 1) ||
        tabFull == (tabEmpty + _size.getSize()))
    {
        if (emptyX == X && emptyY == (Y - 360 / _size.getSize()))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    return false;
}

```

Hình 3 hàm UP()

Hàm DOWN():

- Tương tự hàm UP(). Nhưng hàm sẽ trả về giá trị true nếu như vị trí của ô chứa ảnh nằm dưới vị trí của ô trống. ngược lại sẽ trả về giá trị false.

```

public bool DOWN(Button Empty, Button full, LoadSize getsizes)
{
    _size = getsizes;
    emptyX = Empty.Location.X;
    emptyY = Empty.Location.Y;

    X = full.Location.X;
    Y = full.Location.Y;

    tabFull = full.TabIndex;
    tabEmpty = Empty.TabIndex;

    if (tabFull == (tabEmpty - 1) ||
        tabFull == (tabEmpty - _size.getSize()) ||
        tabFull == (tabEmpty + 1) ||
        tabFull == (tabEmpty + _size.getSize()))
    {
        if (emptyX == X && emptyY == (Y + 360 / _size.getSize()))
        {
            //swap(Empty, full);
            return true;
        }
        else
        {
            return false;
        }
    }
    return false;
}

```

Hình 4 Hàm DOWN()

Hàm LEFT():

- Tương tự hàm DOWN(). Nhưng hàm sẽ trả về giá trị true nếu như vị trí của ô chứa ảnh nằm bên trái vị trí của ô trống. ngược lại sẽ trả về giá trị false.

```

public bool LEFT(Button Empty , Button full, LoadSize getsize)
{
    _size = getsize;
    emptyX = Empty.Location.X;
    emptyY = Empty.Location.Y;

    X = full.Location.X;
    Y = full.Location.Y;

    tabFull = full.TabIndex;
    tabEmpty = Empty.TabIndex;

    if (tabFull == (tabEmpty - 1) ||
        tabFull == (tabEmpty - _size.getSize()) ||
        tabFull == (tabEmpty + 1) ||
        tabFull == (tabEmpty + _size.getSize()))
    {
        if (emptyY == Y && emptyX == (X + 360 / _size.getSize()))
        {
            //swap(Empty, full);
            return true;
        }
        else
        {
            return false;
        }
    }
    return false;
}

```

Hình 5 Hàm LEFT()

Hàm RIGHT():

- Tương tự hàm LEFT(). Nhưng hàm sẽ trả về giá trị true nếu như vị trí của ô chứa ảnh nằm bên phải vị trí của ô trống. ngược lại sẽ trả về giá trị false.

```

public bool RIGHT(Button Empty, Button full, LoadSize getsize)
{
    _size = getsize;
    emptyX = Empty.Location.X;
    emptyY = Empty.Location.Y;

    X = full.Location.X;
    Y = full.Location.Y;

    tabFull = full.TabIndex;
    tabEmpty = Empty.TabIndex;

    if (tabFull == (tabEmpty - 1) ||
        tabFull == (tabEmpty - _size.getSize()) ||
        tabFull == (tabEmpty + 1) ||
        tabFull == (tabEmpty + _size.getSize()))
    {
        if (emptyY == Y && emptyX == (X - 360 / _size.getSize()))
        {
            //swap(Empty, full);
            return true;
        }
        else
        {
            return false;
        }
    }
    return false;
}

```

Hình 7 Hàm RIGHT()

```

public void swap(Button a, Button b)
{
    string temp = a.Text;
    a.Text = b.Text;
    b.Text = temp;

    Image i = a.Image;
    a.Image = b.Image;
    b.Image = i;

    int xx1 = a.Location.X;
    int yy1 = a.Location.Y;
    int xx2 = b.Location.X;
    int yy2 = b.Location.Y;
    int p1 = xx1;
    xx1 = xx2;
    xx2 = p1;
    int p2 = yy1;
    yy1 = yy2;
    yy2 = p2;
}

```

Hình 6 Hàm swap()

- Hàm swap có nhiệm vụ hoán vị 2 vị trí kề nhau theo điều kiện được giao.

```
public void ran( Button empty , Button full, LoadSize getsize)
{
    _size = getsize;
    int s = _size.getSize() * _size.getSize();

    if(empty == full)
    {
        return;
    }
    else if (_MovePlay.UP(empty, full, _size) == true)
    {
        mang(full);
    }
    else if (_MovePlay.DOWN(empty, full, _size) == true)
    {
        mang(full);
    }
    else if (_MovePlay.LEFT(empty, full, _size) == true)
    {
        mang(full);
    }
    else if (_MovePlay.RIGHT(empty, full, _size) == true)
    {
        mang(full);
    }
}
```

Hình 8 Hàm ran() trong class RandomPlay

- Hàm có nhiệm vụ xác định các ô liền kề của ô trống, sau đó cộng dồn tất cả vào một hàm gán mảng.

```

public void SwapRan(Button emp)
{
    do
    {
        ktra = rand.Next(1, count);

    } while (gan == ktra && gan2 == arr4Vitri[ktra]);

    gan = ktra;
    gan2 = arr4Vitri[ktra];

    if (arr4Vitri[ktra] == emp)
    {
        return;
    }
    else
    {
        _MovePlay.swap(arr4Vitri[ktra], emp);

        //lâm sạch hàm lưu trữ tạm thời
        count = 1;
        for (int x = 0; x < 5; x++)
        {
            arr4Vitri[x] = null;
        }
    }
}

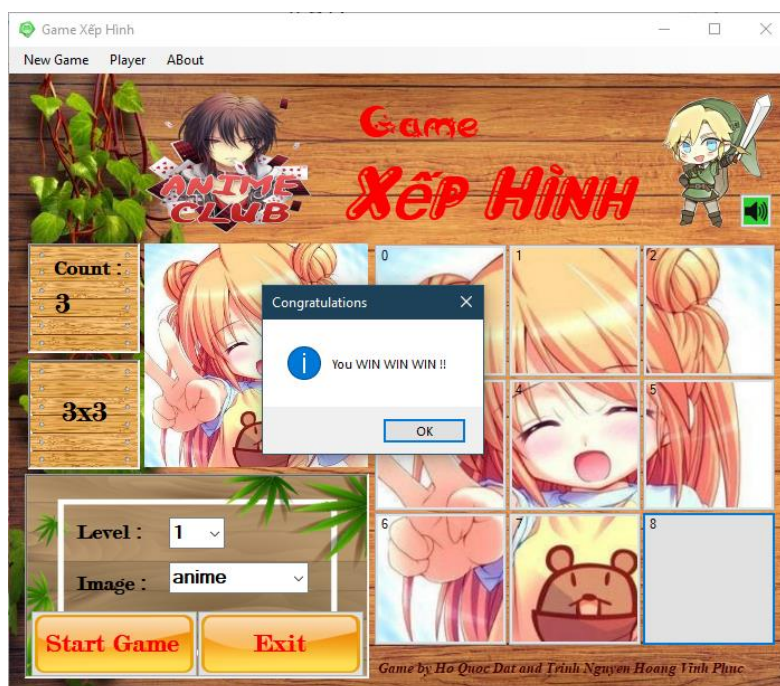
```

Hình 9 Hàm SwapRan() trong class RandomPlay

- Hàm sử dụng một biến “ktra” để lưu lại random một số bất kì trong khoảng từ 1 đến số giá trị trong mảng “mang()” được lưu. Việc này để cho ô trống có thể di chuyển với 4 vị trí xung quanh mà không theo một trật tự nào.
- 2 biến “gan” và “gan2” có nhiệm vụ lưu lại giá trị mà đã thực hiện từ trước và không cho phép thực hiện giá trị đó ở lần random tiếp theo, nhưng có thể lặp lại ở lần sau nếu như giá trị hiện tại không bằng giá trị vừa mới xuất hiện.
- Swap 2 Button khi đã thỏa điều kiện. Sau đó, mảng lưu trữ giá trị các ô xung quanh ô trống sẽ được xóa bỏ, giá trị sẽ được thêm vào khi thực hiện lại.

2.5 Sản phẩm của đồ án

Một game xếp hình hoàn chỉnh dành cho người chơi kết nối cơ sở dữ liệu để lưu điểm.



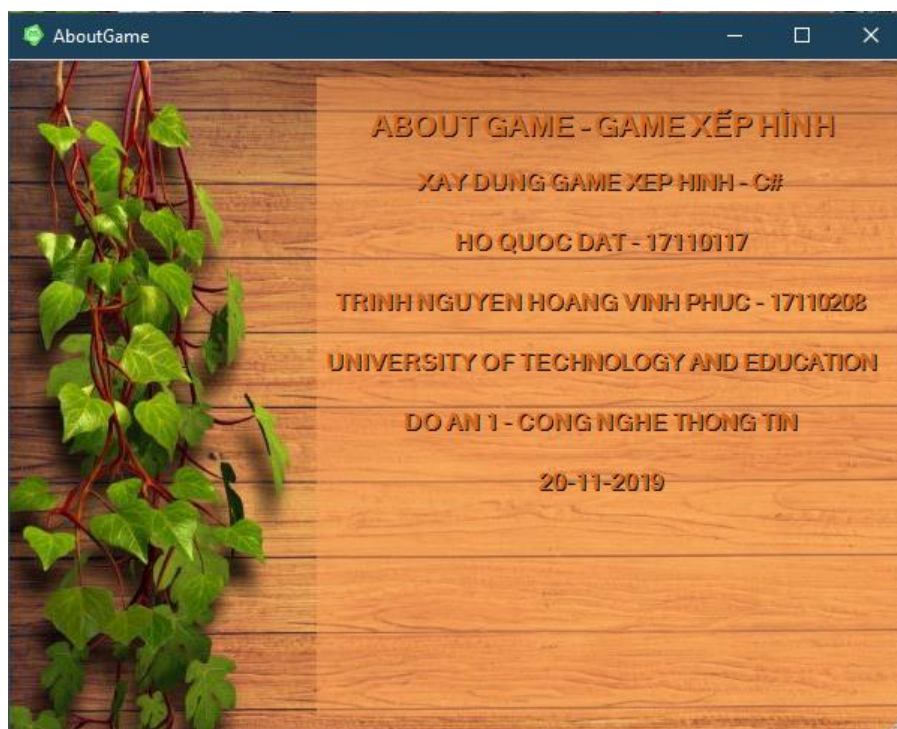
Hình 10 Giao diện chính



Hình 11 Giao diện nhập điểm



Hình 12 Giao diện người chiến thắng



Hình 13 Giao diện giới thiệu về Game

CHƯƠNG 3 : KẾT LUẬN

3.1 Kết luận

So với mục tiêu đã đặt ra thì nhóm tự tin đã hoàn thành được hơn 90% kế hoạch ban đầu. Nhóm đã xây dựng được một game xếp hình hoàn chỉnh cho người chơi và có thể lưu điểm.

3.2 Ưu điểm

Sử dụng hợp lý các class vào cài đặt game xếp hình với khả năng lưu trữ số bước hoàn thành của người chơi kết hợp với kết nối cơ sở dữ liệu. Nhóm đã tạo ra nhiều chức năng trong game xếp hình giúp người chơi dễ dàng sử dụng và thích thú với game hơn.

Phần mềm là một trò chơi điện tử ngoại tuyến đơn giản, dùng để giải trí sau những giờ làm việc căng thẳng, nâng cao tư duy, sáng tạo, và hướng giải quyết để chiến thắng.

Giao diện dễ nhìn, phù hợp với mọi lứa tuổi.

3.3 Nhược điểm

- Vẫn có thể xảy ra lỗi load hình không được trên một số máy.
- Cần phải chỉnh lại địa chỉ của data source nếu chạy trên máy khác.
- Máy phải có SQL Server thì mới có thể lưu điểm được.

3.4 Đề xuất phương án cải thiện

- Phát triển game có thể chơi trên web với những ngôn ngữ lập trình khác.
- Tạo thêm chức năng tự động chơi (tự giải câu đố).
- Phát triển giao diện game đẹp hơn.

- Phát triển về mặt mô phỏng hay lưu ván đã chơi.
- Có thể lưu điểm mà không cần SQL Sever.

PHỤ LỤC

DANH SÁCH CÔNG VIỆC	NỘI DUNG CHI TIẾT CÔNG VIỆC	PHÂN CÔNG VÀ ƯỚC TÍNH % ĐÓNG GÓP	
		Hồ Quốc Đạt	Trịnh Nguyễn Hoàng Vĩnh Phúc
Khảo sát giao diện game			
	Giao diện nhập tên người chơi + tài khoản (nếu có)		X
	Tìm hiểu cách tạo ảnh duy chuyển trong bàn cờ game	X	X
	Tìm hiểu các button điều khiển cần thiết	X	
	Phát sinh cần thiết trong quá trình làm.	X	X
Triển khai Code hoạt động cho game			
	tìm hiểu lập trình hướng đối tượng cho game	X	
	triển khai code hoạt động của game	X	

	thông qua button và click chuột		
	Triển khai code nhập, lưu, sửa, xóa, trong DataServer.		X
	Triển khai code tính thời gian và điểm cho người chơi, lưu kết quả sau khi người chơi đã chiến thắng vào DataServer.		X
	Phát sinh cần thiết trong quá trình làm.	X	X
Kiểm tra Hoạt động của Game và xử lý các trường hợp có thể xảy ra lỗi trong quá trình chơi		X	X

Bảng 1 Phân công công việc

TÀI LIỆU THAM KHẢO

[1] - https://en.wikipedia.org/wiki/Sliding_puzzle