



TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

Niên luận cơ sở ngành KTPM

**TRÒ CHƠI GHÉP HÌNH – SỬ DỤNG THUẬT TOÁN
DUYỆT THEO CHIỀU RỘNG
(BREADTH FIRST SEARCH - BFS)**

Cán bộ hướng dẫn

TS. Huỳnh Quang Nghi

Sinh viên thực hiện

Họ và tên: Kim Thái Phong

Mã số sinh viên: B1805803

Lớp: DI1896A3

Học kỳ 1, năm học 2021 - 2022

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ NGÀNH KTPM

(Học kỳ: 01, Năm học 2021-2022)

TÊN ĐỀ TÀI: TRÒ CHƠI GHEP HÌNH VỚI BFS**GIÁO VIÊN HƯỚNG DẪN:**

STT	HỌ VÀ TÊN	MSCB
1	Huỳnh Quang Nghi	2628

SINH VIÊN THỰC HIỆN:

HỌ VÀ TÊN	MSSV	THƯỜNG (Tối đa 1,0 điểm)	ĐIỂM (thang điểm 10)
Kim Thái Phong	B1805803		

I. HÌNH THỨC (0,5 điểm)	
Bìa (tối đa 0,25 điểm)	
<ul style="list-style-type: none">▪ Đầy đủ các thông tin▪ Đúng định dạng	
Bố cục (tối đa 0,25 điểm)	
<ul style="list-style-type: none">▪ Trang đánh giá kết quả thực hiện niên luận 1▪ Mục lục: cấu trúc chương, mục và tiểu mục▪ Phụ lục (nếu có)▪ Tài liệu tham khảo	
II. NỘI DUNG (3,5 điểm)	
Tổng quan (tối đa 0,5 điểm)	
<ul style="list-style-type: none">▪ Mô tả bài toán (0,25 điểm)▪ Mục tiêu cần đạt được, hướng giải quyết và kế hoạch thực hiện (0,25 điểm)	
Lý thuyết (tối đa 0,5 điểm)	
<ul style="list-style-type: none">▪ Các khái niệm sử dụng trong đề tài (0,25 điểm)▪ Kết quả vận dụng lý thuyết trong đề tài (0,25 điểm)	
Kết quả ứng dụng (tối đa 2,0 điểm)	
<ul style="list-style-type: none">▪ Phân tích yêu cầu, xây dựng các cấu trúc dữ liệu cần thiết (0,5 điểm)▪ Lưu đồ thuật toán, ngôn ngữ giả (1,0 điểm)▪ Giới thiệu chương trình (0,5 điểm)	
Kết luận – đánh giá (tối đa 0,5 điểm)	
<ul style="list-style-type: none">▪ Nhận xét kết quả đạt được▪ Hạn chế - nguyên nhân▪ Hướng phát triển	
III. CHƯƠNG TRÌNH DEMO (5,0 điểm)	
Giao diện thân thiện với người dùng (1,0 điểm)	
Hướng dẫn sử dụng (0,5 điểm)	
Kết quả thực hiện đúng với kết quả của phần ứng dụng (tối đa 3,5 điểm)	
<ul style="list-style-type: none">▪ Kết quả đúng (2,0 điểm)▪ Cách thức thực hiện hợp lý (1,0 điểm)▪ Chức năng bổ sung, sáng tạo (0,5 điểm)	

Cần Thơ, ngày 5 tháng 11 năm 2021

GIÁO VIÊN CHẤM

Huỳnh Quang Nghi

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	1
I. MÔ TẢ BÀI TOÁN:	1
II. MỤC TIÊU CẦN ĐẠT ĐƯỢC:	1
III. HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN:	1
3.3.1 Hướng giải quyết:	1
3.3.2 Kế hoạch thực hiện:	2
CHƯƠNG 2: LÝ THUYẾT	3
I. Các khái niệm sử dụng trong đề tài:	3
II. Kết quả vận dụng lý thuyết trong đề tài:	3
CHƯƠNG 3: KẾT QUẢ ỨNG DỤNG	4
I. PHÂN TÍCH YÊU CẦU BÀI TOÁN, XÂY DỰNG CÁC CẤU TRÚC DỮ LIỆU CẦN THIẾT:	4
3.1.1 Phân tích yêu cầu bài toán:	4
3.1.2 Các cấu trúc dữ liệu cần thiết:	5
II. THIẾT KẾ GIẢI THUẬT:	5
3.2.1. Ngôn ngữ giả:	5
3.2.2. Lưu đồ giải thuật:	7
III. GIỚI THIỆU CHƯƠNG TRÌNH	8
CHƯƠNG 4: KẾT LUẬN - ĐÁNH GIÁ	10
I. KẾT QUẢ ĐẠT ĐƯỢC:	10
II. HẠN CHẾ, NGUYÊN NHÂN:	10
III. HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI:	10
Phần phụ lục	11
I. Hướng dẫn sử dụng demo chi tiết:	11
a) Chức năng chơi trò chơi:	11
b) Chức năng chơi lại trò chơi :	12
c) Chức năng giải trò chơi (giải BFS):	14
d) Chức năng Đi Tới và Đi Lui:	14
e) Chức năng xem các bước đi sau khi giải trò chơi:	15
II. Các biểu mẫu, chứng từ, công thức được sử dụng để thực hiện đề tài.	15
Tài liệu tham khảo	15

CHƯƠNG 1: TỔNG QUAN

I. MÔ TẢ BÀI TOÁN:

- Bài toán yêu cầu viết một chương trình trò chơi ghép hình giúp người chơi có thể giải trí và bằng cách di chuyển ô khuyết bằng cách bấm vào, sắp xếp mảnh hình để hoàn chỉnh hình như đáp án đưa ra, sử dụng thuật toán duyệt theo chiều rộng (Breadth First Search - BFS).
- Yêu cầu chức năng như: Chơi chính thức, Chơi lại và Giải BFS.

II. MỤC TIÊU CẦN ĐẠT ĐƯỢC:

- Người chơi có thể chơi với thao tác thực hiện là bấm vào ô khuyết để di chuyển, sắp xếp mảnh hình hoàn chỉnh.
- Người chơi có thể chơi lại trò chơi, sau khi đã hoàn thành trò chơi hoặc sau khi xem xong các bước giải hoặc sau khi chơi gián đoạn và muốn chơi lại từ đầu.
- Người chơi có thể xem các bước giải khi người chơi không tìm ra lời giải cho trò chơi.
- Áp dụng có hiệu quả thuật toán duyệt theo chiều rộng.
- Viết code rõ ràng, dễ đọc và thời gian chạy chương trình được tối ưu nhất.
- Thiết kế giao diện bắt mắt, thân thiện và dễ sử dụng cho người dùng.

III. HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN:

3.3.1 Hướng giải quyết:

- Sử dụng Thuật toán Duyệt theo chiều rộng (Breadth First Search - BFS).
- Vị trí của các mảnh hình trong trò chơi sẽ nằm ngẫu nhiên trộn lẫn 9 ô, trong đó có một ô khuyết để người chơi dịch chuyển từng bước. Mỗi lần di chuyển ô khuyết, người chơi chỉ có thể di chuyển 1 bước sang trái hoặc phải hoặc lên hoặc xuống để sắp xếp mảnh hình hoàn chỉnh theo mẫu đã có sẵn. Người chơi không được di chuyển chéo.
- Trong quá trình chơi có thể có trường hợp người chơi không thể đi đến trạng thái hoàn chỉnh của hình. Vì vậy, ta áp dụng thuật toán vào trong trò chơi này, và cụ thể là thuật toán Breadth-First-Search. Mục đích là để đưa người chơi đi đến trạng thái hoàn chỉnh (giải ra được đường đi đến trạng thái hoàn chỉnh).
- Vậy bài toán ở đây là tìm ra đường đi đến trạng thái các mảnh hình hoàn chỉnh.

3.3.2 Kế hoạch thực hiện:

- Bước 1: Xây dựng thuật toán Duyệt theo chiều rộng.
- Bước 2: Xây dựng các hàm, cấu trúc cần thiết.
- Bước 3: Kiểm thử một vài trường hợp để đảm bảo rằng các hàm được viết không có sai sót.
- Bước 5: Gộp tất cả các hàm lại với nhau.
- Bước 6: Kiểm thử sau khi các hàm được tích hợp lại với nhau.
- Bước 7: Thiết kế giao diện cho trò chơi.
- Bước 8: Đóng vai trò là một kiểm thử viên. Kiểm thử lại toàn bộ hệ thống sau khi đã tích hợp tất cả các chức năng lại với nhau.

CHƯƠNG 2: LÝ THUYẾT

I. Các khái niệm sử dụng trong đề tài:

- Đề tài sử dụng một số khái niệm liên quan tới lý thuyết đã học và một số định nghĩa phụ khác được sử dụng trong đề tài.
 - Thuật toán Duyệt theo chiều rộng (Breadth First Search - BFS):
 - Đây là thuật toán tìm đường đi từ đỉnh xuất phát đến đỉnh kết thúc bằng cách duyệt theo chiều rộng.
 - Đây là thuật toán không quan tâm đến trọng số trên đường đi mà chỉ duyệt theo những đỉnh kề liên tiếp nó.
 - Xuất phát từ một đỉnh và đi tới các đỉnh kề nó, tiếp tục cho đến khi không còn đỉnh kề nào để đi.
 - Trong quá trình đi đến đỉnh kề, tiến hành lưu lại đỉnh kề để khi đi ngược lại từ đỉnh kết thúc đến đỉnh xuất phát ta có đường đi ngắn nhất.
 - Mảng, mảng hai chiều:
 - Mảng là một cấu trúc dữ liệu tập hợp các phần tử có cùng kiểu dữ liệu, mỗi phần tử trong mảng được xác định bằng các chỉ số (index) hoặc khóa (key).
 - Mảng hai chiều là một mảng một chiều của các mảng một chiều khác, trong toán học mảng hai chiều còn có biểu diễn một ma trận do đó mảng hai chiều cũng có thể gọi là một ma trận.
 - Sắp xếp nổi bọt (Bubble Sort): là một thuật toán sắp xếp đơn giản, với thao tác cơ bản là so sánh hai phần tử kề nhau, nếu chúng chưa đứng đúng thứ tự thì đổi chỗ. Có thể tiến hành từ trên xuống (bên trái sang) hoặc từ dưới lên (bên phải sang).
- Để thực hiện đề tài này, cần nắm vững khái niệm đó và thực hiện một cách nhuần nhuyễn giữa lý thuyết và thực hành. Nhằm đạt được kết quả cao nhất.

II. Kết quả vận dụng lý thuyết trong đề tài:

- Sau khi đã nắm vững các lý thuyết ở phần I. Việc thực hiện đề tài sẽ dễ dàng và có nhiều cách tiếp cận hơn đối với người đang thực hiện đề tài. Đồng thời, cũng là nền tảng để người thực hiện có thể tư duy và giải quyết một số bài toán khó khăn sau này.
- Sau khi đã biết vận dụng lý thuyết một cách có hiệu quả, người thực hiện đề tài sẽ trở nên dễ thực hiện và giải quyết một số bài toán trong đề tài sẽ dễ dàng hơn. Qua đó giúp người thực hiện hiểu rõ hơn về cách hoạt động cũng giúp người sử dụng dễ dàng nắm rõ về hệ thống.

CHƯƠNG 3: KẾT QUẢ ỨNG DỤNG

I. PHÂN TÍCH YÊU CẦU BÀI TOÁN, XÂY DỰNG CÁC CẤU TRÚC DỮ LIỆU CẦN THIẾT:

3.1.1 Phân tích yêu cầu bài toán:

- Yêu cầu chức năng:
 - Bài toán yêu cầu viết một chương trình trò chơi cho phép người chơi di chuyển bằng cách bấm vào ô khuyết, sắp xếp mảnh hình hoàn chỉnh.
 - Bên cạnh đó, yêu cầu thêm một số chức năng như:
 - Giải trò chơi, khi người chơi không tìm ra lời giải cho việc di chuyển ô khuyết, sắp xếp mảnh hình hoàn chỉnh để hoàn thành trò chơi giống với đáp án.
 - Sau khi người chơi bấm nút Giải trò chơi, thì người chơi không còn thao tác được với các mảnh hình, ô khuyết, mà thay vào đó là việc di chuyển các mảnh hình, ô khuyết bằng cách bấm vào nút Đi Tới hoặc Đi Lui để xem các bước giải.
 - Thực hiện chơi lại trò chơi, nếu người chơi đã hoàn thành trò chơi hoặc là đã xem xong lời giải trò chơi hoặc người chơi đang chơi gián đoạn mà muốn bắt đầu lại từ đầu.
- Yêu cầu phi chức năng: Chương trình phải đảm bảo được chất lượng cũng như độ tin cậy cần thiết.
- Yêu cầu về sản phẩm: Chương trình cần phải trả ra kết quả nhanh và chính xác, đảm bảo đúng yêu cầu người chơi. Ngoài ra, cần phải tiêu tốn ít tài nguyên bộ nhớ nhất.
- Yêu cầu về giao diện: Bài toán yêu cầu trình bày giao diện thân thiện, dễ sử dụng và dễ tiếp cận đối với người mới sử dụng.
- Yêu cầu về phát triển: Chương trình nguồn cần được viết ngắn gọn, chặt chẽ, dễ đọc mã nguồn nhằm để thuận lợi cho việc phát triển.

3.1.2 Các cấu trúc dữ liệu cần thiết:

- Một số cấu trúc dữ liệu và giải thuật được sử dụng xuyên suốt trong chương trình như:
 - Duyệt theo chiều rộng.
 - Bên cạnh đó còn có một số hàm đi kèm như:
 - Hàm tạo mảng hai chiều đáp án, đề bài.
 - Kiểm tra xem hai mảng hai chiều có giống nhau không.
 - Hàm trả về 1 mảng hai chiều giống với mảng truyền vào.
 - Hàm tìm vị trí ô khuyết.
 - Hàm in mảng hai chiều.
 - Hàm tìm các trạng thái có thể xảy ra khi di chuyển ô khuyết từ mảng hai chiều ban đầu (đề bài).
 - Hàm in lại bước đi của ô khuyết.
 - Hàm tạo mảng hai chiều.
 - Hàm kiểm tra người chơi di chuyển ô khuyết có hợp lệ không.
 - Hàm hoán đổi hai mảnh hình khi người chơi di chuyển ô khuyết.
 - Hàm tạo lại mảng hai chiều khi người chơi bấm nút chơi lại.
 - Ngoài ra còn có một số hàm khác nhưng không được sử dụng trong đề tài này nên có thể bỏ qua.
- Bên cạnh đó cũng không thể thiếu một số giải thuật căn bản như:
 - Một số kiểu dữ liệu được sử dụng như: mảng hai chiều [,], kiểu số nguyên Integer, đệ qui, một số đối tượng do lập trình viên tự định nghĩa, ...

II. THIẾT KẾ GIẢI THUẬT:

3.2.1. Ngôn ngữ giả:

```
Hàm BFS() {  
  <List> Queue //Lưu trạng thái được tạo ra và chờ duyệt  
  <List> Close //Lưu trạng thái đã duyệt qua  
  Queue <- u0 //u0 là trạng thái bắt đầu và thêm u0 vào Queue  
  Close = {rỗng} //  
  While (Queue != rỗng) {  
    //Kiểm tra Queue xem có còn phần tử nào không?  
    Xóa u ở đầu Queue  
    Thêm u vào Close  
    If (u là trạng thái đích) {  
      //Kiểm tra xem u có phải là trạng thái đích hay không?  
      Thông báo tìm kiếm thành công;  
      Exit;  
    }  
    For (mỗi trạng thái v kề u) {  
      //Duyệt các trạng thái v kề u  
      If (v không có trong Queue và Close) {  
        //Kiểm tra xem v có tồn tại trong Queue và Close hay không?
```

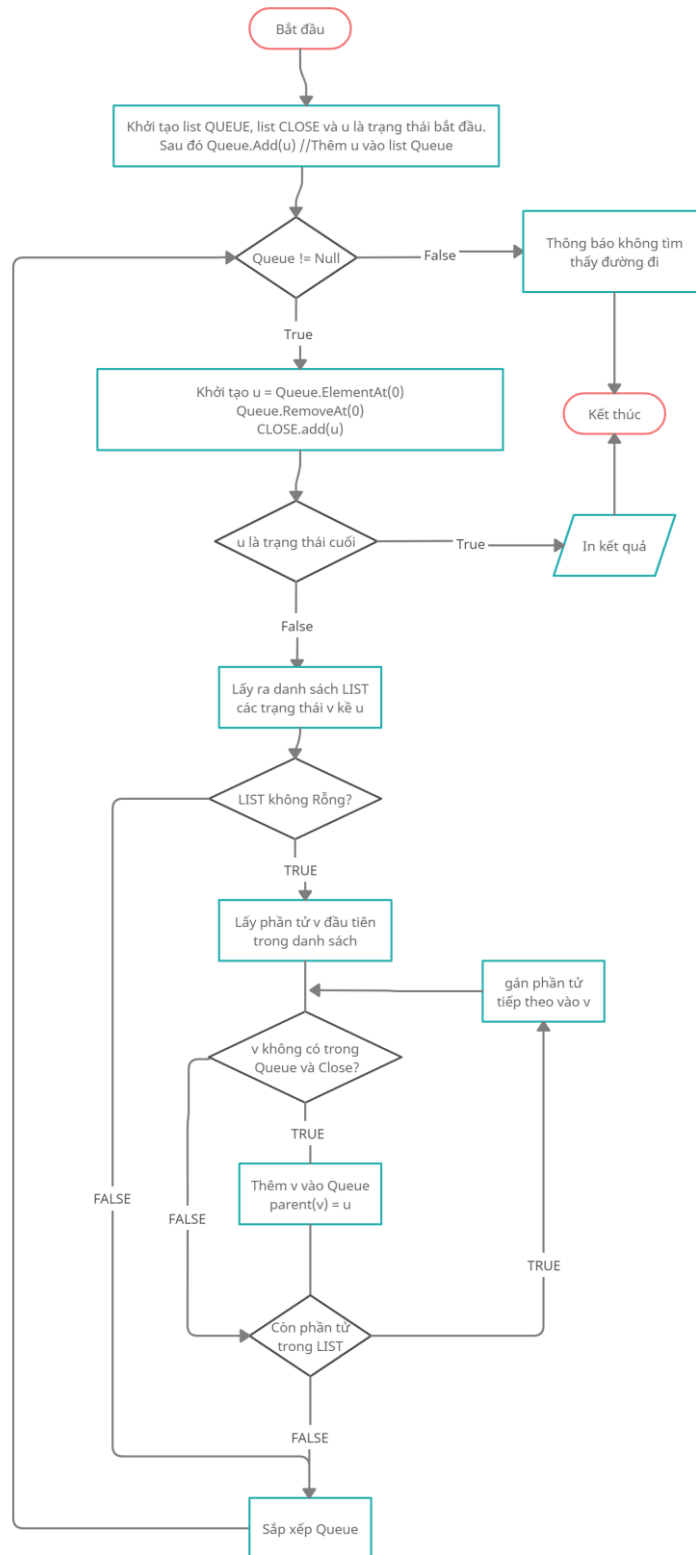


```

        Thêm v vào cuối Queue;
        Father(v) <- u; // Cập nhật Cha của đỉnh v
    }
}
Sắp xếp các trạng thái trong Queue theo thứ tự tăng dần theo số lượng
ô Sai so với trạng thái đích. Sau đó quay lại vòng lặp While cho đến
khi tìm ra đáp án.
}
Thông báo tìm kiếm thất bại
}

```

3.2.2. Lưu đồ giải thuật:



- Mô tả thuật toán: Các đỉnh đã được xét thì không thể xét lại lần hai nữa. Cơ chế lưu lại đỉnh kề sẽ là lưu lại các đỉnh kề thành một danh sách và lấy từ từ danh sách các đỉnh kề ra để xét, khi mà một đỉnh kề của một đỉnh đang xét được thêm vào danh sách thì nó sẽ được thêm vào cuối của danh sách hay còn gọi là cơ chế hàng đợi. Cơ chế này có nghĩa là khi thêm một phần tử thì phần tử đó sẽ được thêm ở cuối danh sách hàng đợi, còn lấy phần tử ra thì sẽ lấy ra ở đầu danh sách hàng đợi. Đó chính là điểm nổi bật để phân biệt thuật toán này với các thuật toán khác.
 - Ưu điểm:
 - Dễ cài đặt.
 - Nếu số đỉnh là hữu hạn, thuật toán chắc chắn tìm ra kết quả.
 - Khuyết điểm:
 - Mang tính chất vét cạn, không nên áp dụng nếu duyệt số đỉnh quá lớn.
 - Phải duyệt tất cả các đỉnh, không chú ý đến không tin trong các đỉnh để duyệt hiệu quả, dẫn đến duyệt qua các đỉnh không cần thiết.
 - Chiếm thời gian và không gian bộ nhớ khi số đỉnh duyệt quá nhiều.
- Ứng dụng thuật toán vào trò chơi ghép hình:
 - Áp dụng thuật toán như đã nói trên vào trò chơi ghép hình. Lúc này mỗi trạng thái hay mỗi đỉnh mà thuật toán duyệt qua chính là một danh sách các số từ một đến chín, trong đó có một ô số đánh dấu là ô khuyết để cho người chơi di chuyển,
 - Mỗi lần thuật toán duyệt qua một trạng thái, sẽ đưa vào trong hàng đợi, như vậy ta sẽ có danh sách chứa những danh sách.
 - Kết quả đường đi tìm được sẽ trả về danh sách của những trạng thái mà nó đã tìm ra (cũng là một danh sách các số từ một đến chín).

III. GIỚI THIỆU CHƯƠNG TRÌNH

- Khi người dùng khởi động chương trình lên. Màn hình sẽ hiện thị bên tay phải là hình gốc (hình hoàn chỉnh sau khi ghép đúng các mảnh hình), bên tay trái là chín ô số chứa từng phần nhỏ được cắt ra từ bức tranh hoàn chỉnh, chín ô số này sẽ có một ô khuyết trống để người chơi di chuyển các ô lân cận, chín ô này sẽ được sắp xếp ngẫu nhiên không hoàn chỉnh thành một bức tranh.
- Nhiệm vụ của người chơi là di chuyển từng mảnh hình một lên, xuống, trái, phải và không được di chuyển chéo dựa theo ô khuyết, sao cho sắp xếp hoàn thành một bức tranh hoàn chỉnh như hình bên tay phải.
- Khi người chơi bắt đầu chơi và số bước đi của người chơi sẽ được đếm nhằm thống kê số bước người chơi đã đi đến kết quả cuối cùng.
- Người chơi có thể chọn nút chơi lại để bắt đầu chơi lại từ đầu, lúc này chín ô số sẽ được trộn ngẫu nhiên lại một lần nữa và bước đi của người chơi sẽ được cài đặt về không như trạng thái ban đầu.

- Nếu người chơi không thể đi đến trạng thái cuối cùng để hoàn thành một bức tranh hoàn chỉnh thì người chơi có thể chọn nút giải theo BFS trên màn hình, sau khi chọn thuật toán giải thì tổng số bước đi để giải được bài toán sẽ được hiển thị lên, từ đây người chơi bấm vào nút đi tới để bắt đầu xem kết quả đi từng bước của thuật toán cho đến khi hoàn chỉnh bức tranh. Người chơi cũng có thể đi lui để xem lại bước đi trước đó.
- Người chơi có thể thoát khỏi chương trình.

CHƯƠNG 4: KẾT LUẬN - ĐÁNH GIÁ

I. KẾT QUẢ ĐẠT ĐƯỢC:

Chương trình, trò chơi giải quyết được hầu hết tất cả các vấn đề và chức năng đặt ra.

II. HẠN CHẾ, NGUYÊN NHÂN:

- Tuy đã giải quyết được các vấn đề, chức năng cơ bản đã được đặt ra. Chương trình còn gặp một số vấn đề cơ bản như: chương trình thô sơ, giao diện chưa thân thiện với người chơi.
- Bên cạnh đó, việc xây dựng một chương trình bằng công cụ Visual Studio 2019 cũng gặp một số khó khăn cần thiết như: Chương trình sẽ thô sơ, không có thu hút được người sử dụng. Chỉ đáp ứng được chức năng đặt ra nhưng không đáp ứng về vẻ thẩm mỹ của chương trình.

III. HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI:

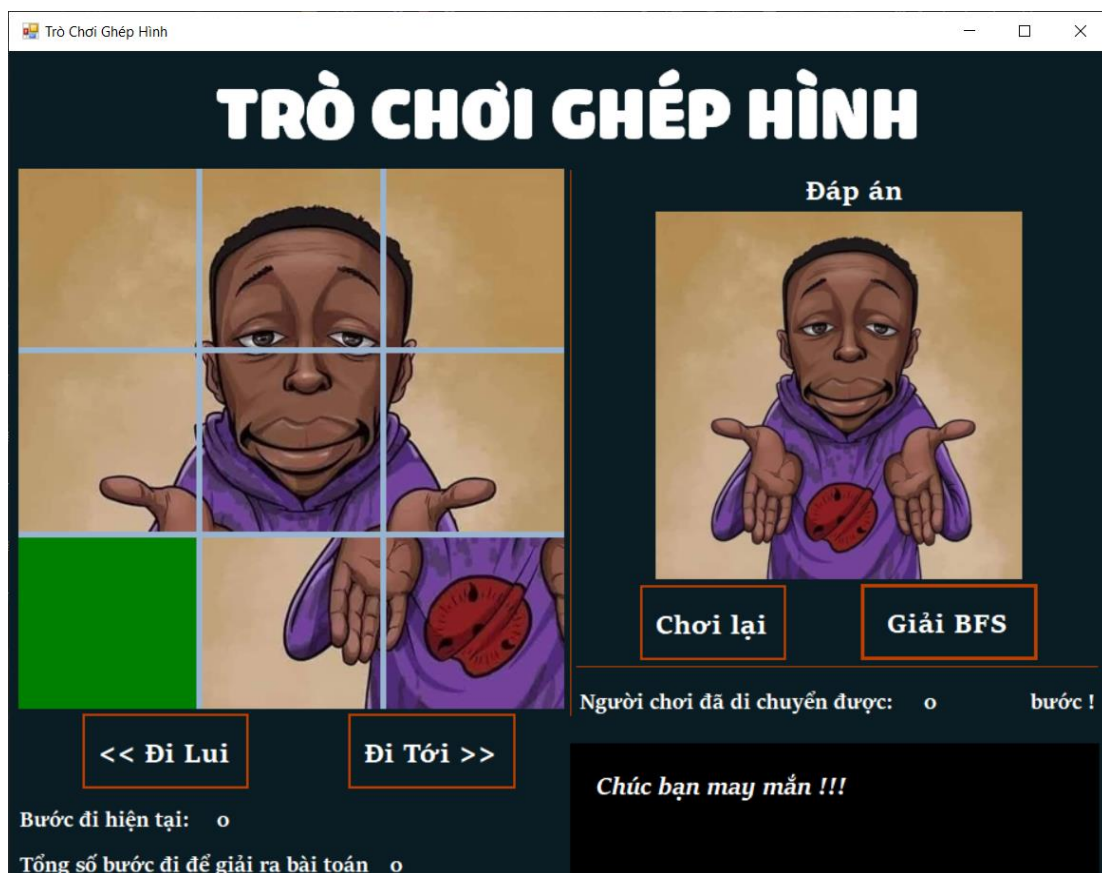
- Đối với trường hợp có quá nhiều mảnh hình (ô) bị sai thì thuật toán BFS vẫn sẽ tìm ra đường đi, nhưng thời gian tìm ra thì lâu hơn bình thường. Do đó cần áp dụng các thuật toán tối ưu khác để có sự quan sát độ tối ưu, thời gian nhanh hơn.
- Đề tài cần được phát triển theo chiều hướng hiện đại hơn. Có thể xây dựng đề tài trên một ngôn ngữ lập trình khác như: Java, PHP, JS, ... Để có một cái nhìn tổng quát về đề tài. Đồng thời có thể phát triển trò chơi mở rộng lên 16 ô, 25 ô, ...
- Ngoài ra, việc phát triển đề tài còn giúp người sử dụng hiểu rõ hơn về cách thức hoạt động của các chương trình, các mô-đun bên trong.

Phần phụ lục

I. Hướng dẫn sử dụng demo chi tiết:

a) Chức năng chơi trò chơi:

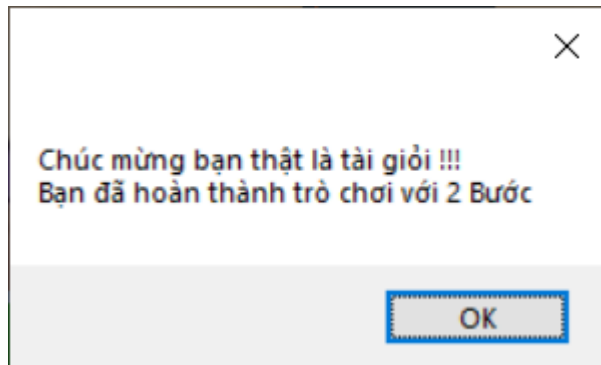
- Khi trò chơi bắt đầu vào trò chơi thì cách màn hình sẽ được random một cách ngẫu nhiên, yêu cầu người chơi di chuyển ô khuyết bằng cách bấm vào, sắp xếp (ghép) các mảnh hình lại thành một hình hoàn chỉnh như hình Đáp Án bên cạnh.



- Khi người chơi thực hiện bấm vào ô khuyết để di chuyển, sắp xếp các mảnh hình hoàn chỉnh, thì mỗi lần di chuyển sẽ được tính là một bước và phía bên phải sẽ hiện số lần di chuyển ô khuyết mà người chơi đã thực hiện bấm vào. Phần này giúp đánh giá được người chơi đã sử dụng di chuyển ô khuyết bao nhiêu lần để ghép các mảnh hình hoàn chỉnh. Ngoài ra, nó còn có thể giúp người chơi ghi nhận lại số bước đi để lần sau người chơi có thể cải thiện thành tích của bản thân để chiến thắng trò chơi. Một khi người chơi di chuyển ô khuyết càng ít thì người chơi sẽ đánh giá được khả năng chơi trò chơi của mình như thế nào?

Người chơi đã di chuyển được: o bước !

- Sau khi người chơi thực hiện di chuyển ô khuyết để sắp xếp các mảnh hình hoàn chỉnh đến kết quả cuối cùng hoàn thành trò chơi, thì trò chơi sẽ xuất hiện thông báo như hình bên dưới. Chứng tỏ người chơi đã hoàn thành xuất sắc trò chơi ghép hình với số lần di chuyển ô khuyết mà người chơi đã bấm vào.

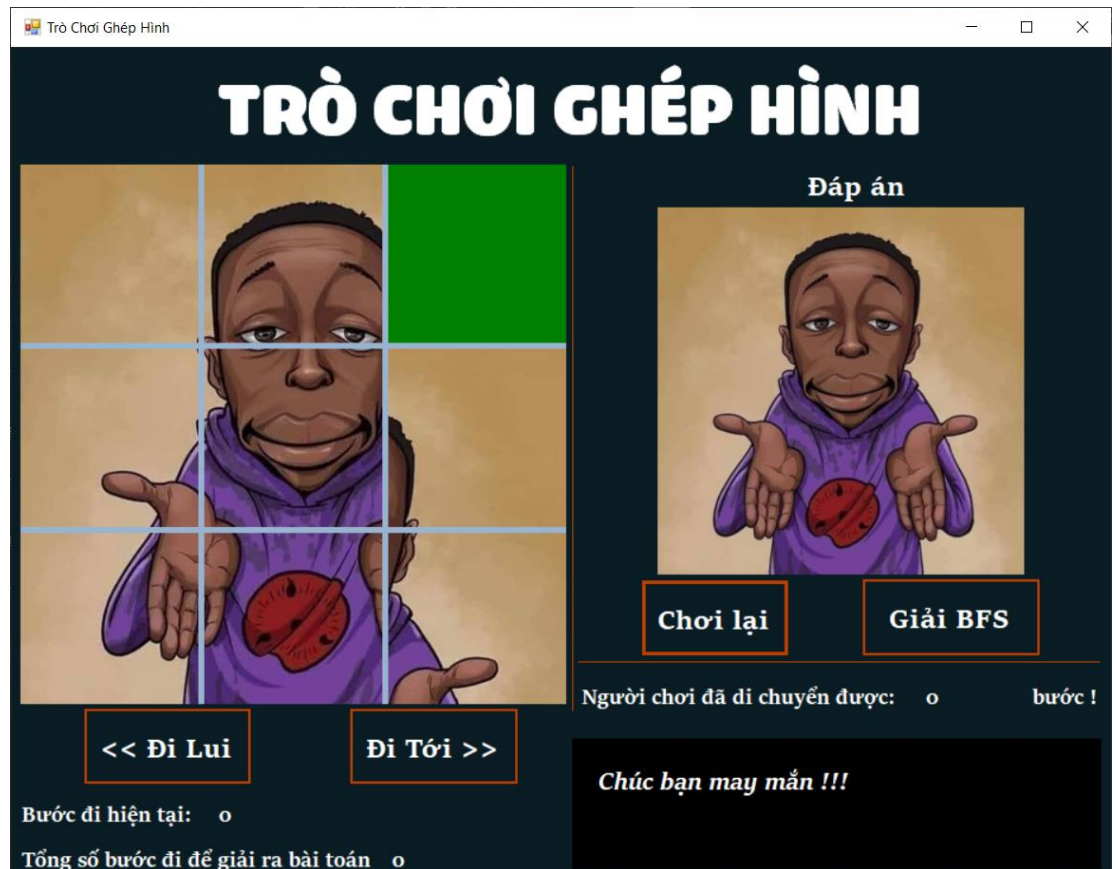


b) Chức năng chơi lại trò chơi :

- Sau khi hoàn thành trò chơi hoặc nếu gặp khó khăn trong việc di chuyển ô khuyết, sắp xếp mảnh hình hoàn chỉnh hoặc người chơi đã tìm ra cách để hoàn thành trò chơi mà ít tốn thời gian hơn cũng như ít số lần di chuyển ô khuyết hơn thì người chơi có thể chơi lại từ đầu.

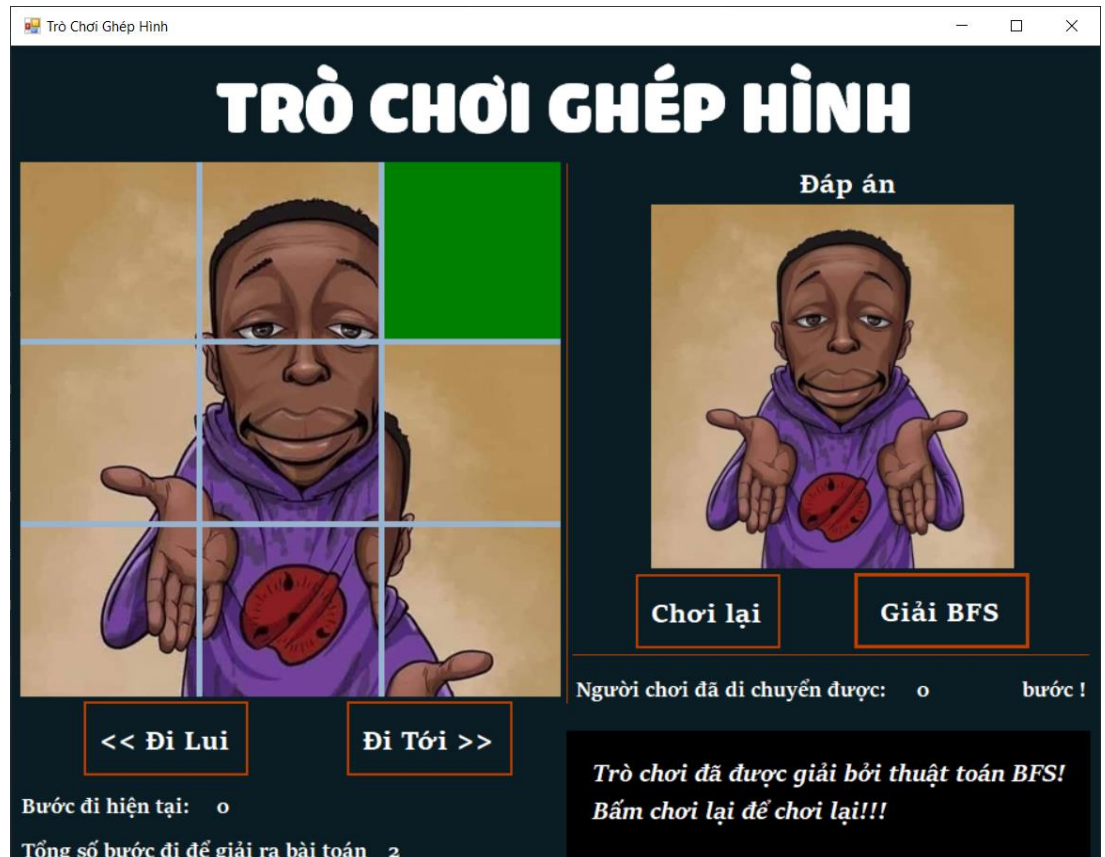


- Nếu người chơi bấm nút Chơi Lại có trên màn hình (giao diện trò chơi) thì các mảnh hình sẽ sắp xếp lại ngẫu nhiên. Sau đó Trò Chơi Ghép Hình sẽ được tạo lại để người chơi thực hiện các bước di chuyển ô khuyết, sắp xếp các mảnh hình để ghép hoàn chỉnh như lúc đầu.



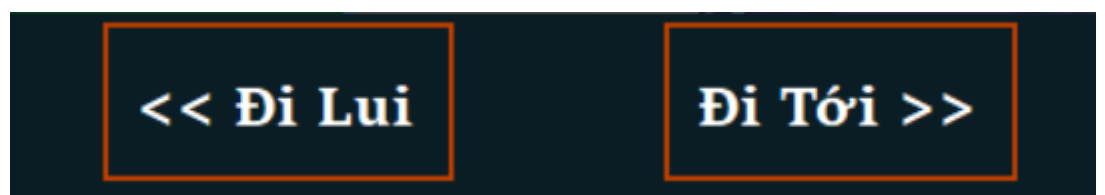
c) Chức năng giải trò chơi (giải BFS):

- Nếu người chơi không tìm ra được các bước di chuyển ô khuyết, sắp xếp các mảnh hình hoàn chỉnh. Thì người chơi có thể bấm vào **Giải BFS** để xem lời giải về số lần di chuyển ô khuyết, sắp xếp mảnh hình hoàn chỉnh. Lúc đó tại giao diện của người chơi, người chơi sẽ không thao tác bấm được với ô khuyết như lúc ban đầu. Mà chỉ được bấm nút **Đi Tới** và **Đi Lui** để xem các bước di chuyển ô khuyết, sắp xếp hình hoàn chỉnh. Khi đó thuật toán đã được giải và sẽ hiện thị ra màn hình tổng số bước đi để giải ra bài toán, giải ra được trạng thái cuối cùng để người chơi có thể đi đến chiến thắng.



d) Chức năng Đi Tới và Đi Lui:

- Khi người chơi chọn chức năng **Giải BFS** khi thuật toán đã thực thi xong thì bạn sẽ chọn hai nút **Đi Tới** hoặc **Đi Lui** để đi từng bước tới trạng thái cuối cùng. Chức năng này có thể giúp người chơi có thể hiểu được các bước giải của thuật toán thông qua việc đi từng bước, từng bước một. Khi đã đi hết tất cả các bước thì người chơi sẽ đi tới kết quả cuối cùng.



e) Chức năng xen các bước đi sau khi giải trò chơi:

- Và phía bên dưới 2 nút **Đi Tới** và **Đi Lui** sẽ là nơi hiện thị lời giải. Là cần bao nhiêu bước để di chuyển ô khuyết, sắp xếp hình hoàn chỉnh? Và hiện tại đang ở bước thứ bao nhiêu? Do đó **Bước đi hiện tại** – sẽ là nơi hiện thị số lần thực hiện di chuyển ô khuyết, sắp xếp mảnh hình hoàn chỉnh khi người chơi tiến hành bấm vào nút **Đi Tới** hoặc **Đi Lui** để xem sự di chuyển của ô khuyết, sắp xếp mảnh hình hoàn chỉnh. Còn **Tổng số bước đi để giải ra bài toán** – sẽ là nơi hiện thị tổng số lần di chuyển ô khuyết, sắp xếp mảnh hình hoàn chỉnh tính từ thời điểm cuối cùng người chơi bấm vào ô khuyết.

Bước đi hiện tại: 0

Tổng số bước đi để giải ra bài toán 2

- Đồng thời phía bên phải dưới cùng sẽ hiện ra thông báo là người chơi đã bấm vào nút **Giải BFS**. Nếu muốn chơi lại thì người chơi sẽ bấm vào **Chơi lại** và chơi lại từ đầu.

**Trò chơi đã được giải bởi thuật toán BFS!
Bấm chơi lại để chơi lại!!!**

II. Các biểu mẫu, chứng từ, công thức được sử dụng để thực hiện đề tài.

Các biểu mẫu: Không.

Chứng từ: Không.

Công thức: Giải thuật BFS cùng một số hàm cơ bản.

Tài liệu tham khảo

- [1] **Nguyễn Văn Linh.** *Giáo trình Phân tích và thiết kế thuật toán.* Nhà xuất bản Đại học Cần Thơ. 2010.
- [2] **ThS Võ Huỳnh Trâm.** *Slide bài giảng Phân tích và thiết kế thuật toán.* Nhà xuất bản Đại học Cần Thơ. 2010.
- [3] **Hồ Quang Thái.** *Slide bài giảng lập trình .NET.* Nhà xuất bản Đại học Cần Thơ. 2010.