

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



# **BÁO CÁO ĐỒ ÁN 1**

## **ĐỀ TÀI:**

## **GAME SUDOKU TRÊN WINFORM**

Nhóm sinh viên thực hiện:

Võ Minh Hiếu 17110136

Huỳnh Xuân Huy 17110147

GVHD: TS. Huỳnh Xuân Phụng

*Thành phố Hồ Chí Minh, tháng 11 năm 2019*

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



## **BÁO CÁO ĐỒ ÁN 1**

### **ĐỀ TÀI:**

## **GAME SUDOKU TRÊN WINFORM**

Nhóm sinh viên thực hiện:

Võ Minh Hiếu 17110136

Huỳnh Xuân Huy 17110147

GVHD: TS. Huỳnh Xuân Phụng

*Thành phố Hồ Chí Minh, tháng 11 năm 2019*

## ĐIỂM SỐ

TIÊU CHÍ	NỘI DUNG	TRÌNH BÀY	TỔNG
ĐIỂM	8.5	9	8.7

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Nhóm sinh viên hoàn thành các chức năng của đề tài, có xử lý thuật toán để giải game sodoku. Nhóm sinh viên có tinh thần làm việc tích cực.

Đánh giá: Giỏi

Điểm: Võ Minh Hiếu – 8.8

Huỳnh Xuân Huy – 8.0

.....  
.....  
.....  
.....

Giáo viên hướng dẫn

*(ký và ghi họ tên)*

(Đã ký)

**Huỳnh Xuân Phụng**

## LỜI NÓI ĐẦU

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, tiến sĩ Huỳnh Xuân Phụng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn Khoa đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kỹ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

# MỤC LỤC

<b>LỜI NÓI ĐẦU.....</b>	<b>4</b>
<b>DANH MỤC CÁC HÌNH.....</b>	<b>7</b>
<b>DANH MỤC CÁC BẢNG .....</b>	<b>8</b>
<b>I. ĐẶC TẢ.....</b>	<b>8</b>
1. Giới thiệu game Sudoku .....	8
2. Dữ liệu đầu vào.....	9
3. Các tình huống sử dụng .....	9
4. Giao diện dự kiến.....	9
<b>II. PHÂN CÔNG.....</b>	<b>10</b>
<b>III. THIẾT KẾ.....</b>	<b>11</b>
1. Thuật toán.....	11
2. Thiết kế lớp .....	13
A. Các lớp được sử dụng trong chương trình.....	13
B. Mẫu bảng mô tả các phương thức trong một lớp.....	14
3. Thiết kế database .....	17
4. Bảng mô tả các field trong bảng <i>NgnoiDung</i> .....	18
5. Thiết kế giao diện.....	18
<b>IV. CÀI ĐẶT KIỂM THỬ .....</b>	<b>21</b>
<b>V. KẾT LUẬN .....</b>	<b>26</b>
1. Kết quả đạt được.....	26
2. Khó khăn gặp phải.....	26
3. Ưu điểm phần mềm .....	26

4.	<i>Nhược điểm phần mềm.....</i>	26
5.	<i>Hướng phát triển phần mềm.....</i>	27
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>28</b>

# DANH MỤC CÁC HÌNH

Hình 1. Giao diện dự kiến .....	10
Hình 2. Sơ đồ thuật toán quay lui.....	12
Hình 3. Sơ đồ Flowchart thuật toán giải Sudoku .....	13
Hình 4. Sơ đồ khối AlwaysCheckIsOk.....	15
Hình 5. Mã giả isOK.....	16
Hình 6. Sơ đồ khối Solve_Sudoku().....	17
Hình 7. Giao diện Form1 .....	18
Hình 8. Giao diện Login .....	19
Hình 9. Giao diện InputPad.....	19
Hình 10. Giao diện FormWin.....	20
Hình 11. Kết quả dự kiến kiểm thử 1(1).....	21
Hình 12. Kết quả dự kiến kiểm thử 1 (2).....	21
Hình 13. Kết quả dự kiến kiểm thử 2 .....	22
Hình 14. Kết quả dự kiến kiểm thử 3 .....	22
Hình 15. Kết quả dự kiến kiểm thử 4 .....	23
Hình 17. Kết quả dự kiến kiểm thử 6(1).....	25
Hình 18. Kết quả dự kiến kiểm thử 6(2).....	25
Hình 19. Kết quả dự kiến kiểm thử 7 .....	25

## DANH MỤC CÁC BẢNG

Bảng 1. Bảng phân công .....	10
Bảng 1. Thiết kế lớp .....	13
Bảng 3. Bảng phương thức .....	14
Bảng 4: Bảng Database .....	18
Bảng 5. Field trong bảng NguoiDung .....	18
Bảng 6. Bảng giao diện.....	18
Bảng 7. Bảng kiểm thử .....	21

## I. ĐẶC TẢ

### *1. Giới thiệu game Sudoku*

Sudoku là một trò chơi từng gây nghiện của các quốc gia phát triển. Độ phức tạp của trò chơi tăng dần khi các số cho trước giảm dần. Vì vậy để chinh phục trò chơi này dễ dàng hơn, người ta đã tìm thuật toán (quay lui) và lập trình nó trên máy



tính. Nhờ vào sự phát triển của công nghệ mà máy tính có thể giải một ma trận Sudoku trong thời gian ngắn.[1]

Quy tắc chơi Sudoku, bạn chỉ cần đảm bảo và ghi nhớ điều duy nhất, đó là điền kín các chữ số còn thiếu vào các ô trống. Cụ thể:

- Các hàng ngang: Phải có đủ các số từ 1 đến 9, không cần đúng thứ tự.
- Các hàng dọc: Đảm bảo có đủ các số từ 1-9, không cần theo thứ tự.
- Mỗi vùng cũng phải có đủ các số từ 1-9.[2]

Từ đó ta cũng có thể viết hoàn thiện một game Sudoku và cho phép người dùng điền vào giá trị và máy tính liên tục kiểm tra các giá trị đó.

Sở dĩ nhóm chọn đề tài về game Sudoku là vì phần thuật toán của game vô cùng hấp dẫn, cần tư duy logic cao. Ngoài ra, các thành viên trong nhóm cũng từng rất hứng thú với trò chơi này trên giấy. Trò chơi thật sự khó, phải tốn nhiều thời gian và công sức để giải. Vì vậy việc giải trong vòng vài giây, đối với nhóm mang lại sức hấp dẫn rất lớn.

## ***2. Dữ liệu đầu vào***

Dữ liệu đầu vào là 1 ma trận 9x9 với 81 ô với giới hạn các số được điền trước theo từng Level.

## ***3. Các tình huống sử dụng***

Mục đích của trò chơi giúp người chơi rèn luyện các giác quan trở nên nhạy bén hơn, rèn luyện tư duy của người chơi khi họ phải liên tục tính toán để điền các con số một cách chính xác.

## ***4. Giao diện dự kiến***

Giao diện dự kiến sẽ là một bảng Panel với một list Button 2 chiều.



Hình 1. Giao diện dự kiến

## II. PHÂN CÔNG

Bảng 1. Bảng phân công

TT	Tên sinh viên	Mô tả công việc	Đóng góp
1	Võ Minh Hiếu	<ul style="list-style-type: none"> <li>- Thiết kế giao diện và chức năng trên form.</li> <li>- Thiết kế các lớp cho đối tượng và chức năng cho phần mềm.</li> <li>- Fix lỗi trong quá trình viết.</li> <li>- Viết báo cáo.</li> </ul>	70%

2	Huỳnh Xuân Huy	<ul style="list-style-type: none"> <li>- Viết báo cáo.</li> <li>- Viết chức năng đăng nhập.</li> <li>- Tạo các button cho Panel bảng game và InputPad</li> <li>- Kiểm thử.</li> </ul>	30%
---	----------------	---	-----

### III. THIẾT KẾ

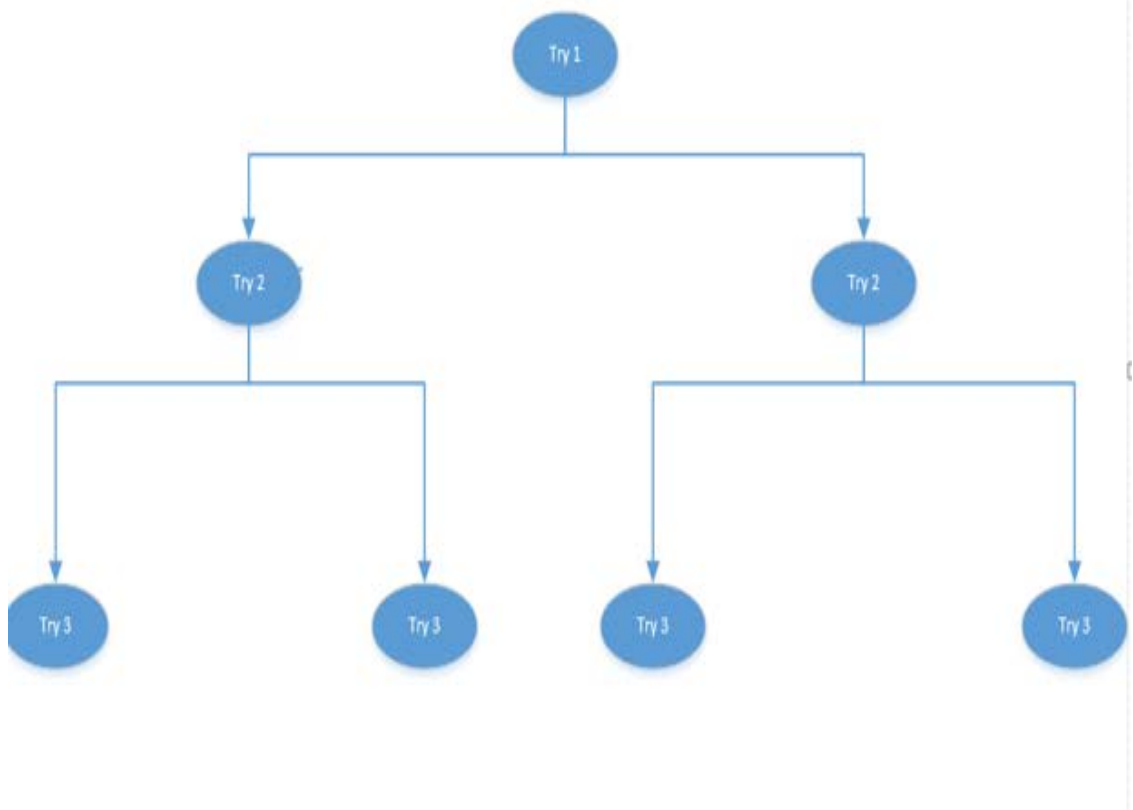
#### 1. Thuật toán

Xoay quanh cả trò chơi là thuật toán rất đơn giản đã học ở môn Cấu trúc dữ liệu và Giải Thuật đó là Quay lui (backtracking). Và thường thì thuật toán quay lui thường viết bằng lối đệ quy.

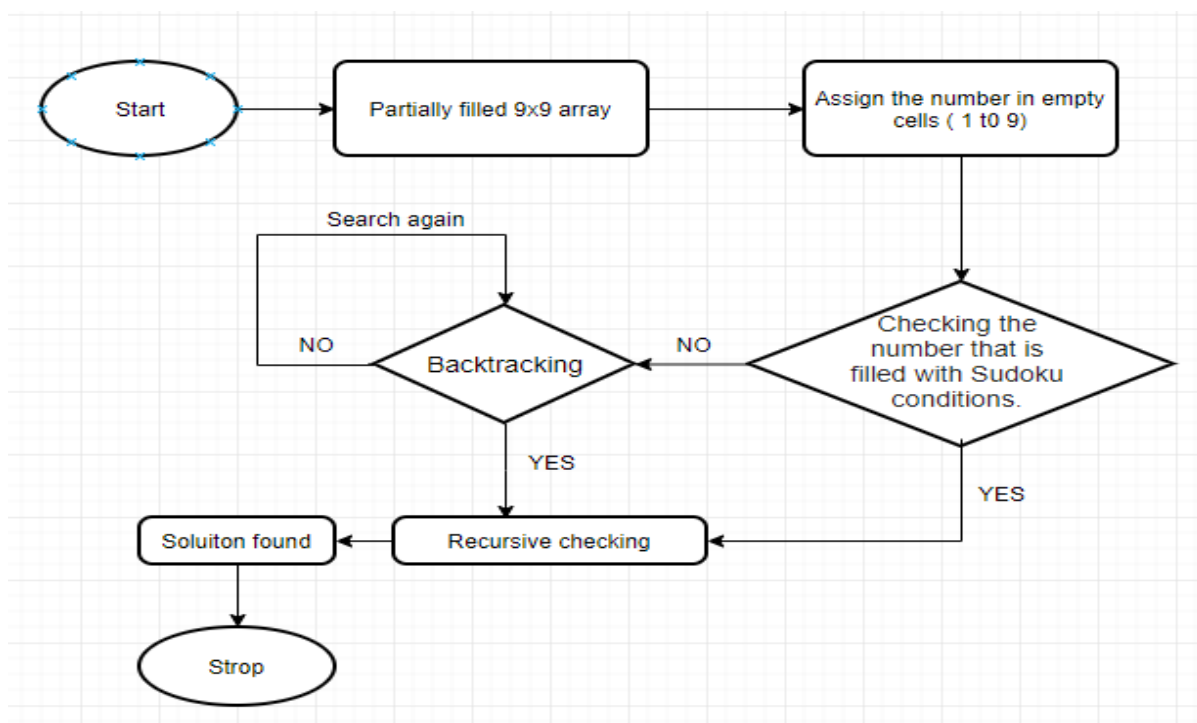
Thuật toán quay lui là một thuật toán điển hình để giải các bài toán ứng dụng trong tin học. Bằng việc liệt kê các tình huống, thử các khả năng có thể cho đến khi tìm thấy một lời giải đúng, thuật toán quay lui chia nhỏ bài toán, lời giải của bài toán lớn sẽ là kết quả của việc tìm kiếm theo chiều sâu của tập hợp các bài toán con. Trong suốt quá trình tìm kiếm nếu gặp phải một hướng nào đó mà biết chắc không thể tìm thấy đáp án thì quay lại bước trước đó và tìm hướng khác kế tiếp hướng vừa tìm kiếm đó. Trong trường hợp không còn một hướng nào khác nữa thì thuật toán kết thúc.

Khác với thuật toán tham lam, thuật toán quay lui có điểm khác là nó không cần phải duyệt hết tất cả các khả năng, nhờ đó tránh được các khả năng không đúng nên có thể giảm được thời gian giải. Thuật toán quay lui thường được cài đặt theo

lỗi đệ quy, hàm đệ quy được thực hiện để giải quyết các bài toán con để trả về kết quả của bài toán lớn. Mục đích của việc sử dụng hàm đệ quy là để thuật toán được rõ ràng, dễ viết, dễ hiểu hơn và cũng để bảo toàn các biến, các trạng thái lúc giải bài toán con.[3]



*Hình 2. Sơ đồ thuật toán quay lui.*



Hình 3. Sơ đồ Flowchart thuật toán giải Sudoku

## 2. Thiết kế lớp

Mục tiêu của chúng ta là tạo 1 bảng game, cho phép người dùng thao tác (điền số) trên đó. Bảng game như đã nói là ma trận các Button 9x9. Dữ liệu của Button được lấy từ ma trận được tạo sẵn.

### A. Các lớp được sử dụng trong chương trình

Bảng 1. Thiết kế lớp

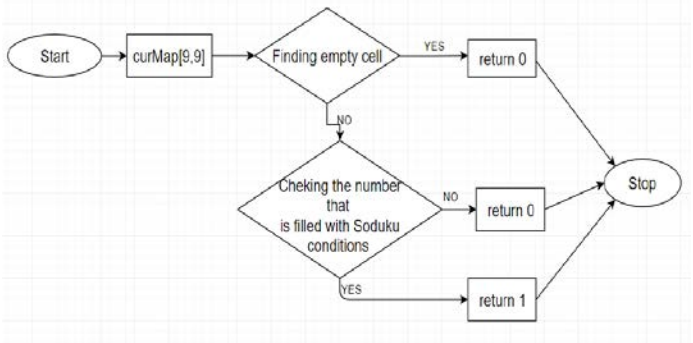
TT	Tên Lớp	Mục đích
1	ChessBoard(kế thừa Solution)	Tạo bảng game chứa các Button thể hiện được mảng 2 chiều 9x9 với các ô trống để người chơi nhập vào được quản lý bởi 1 panel. Và các chức năng có tác động trực tiếp tới bảng game.
2	Solution	Thuật toán giải game Sudoku.

3	Cell	Là đối tượng đại diện cho 1 vị trí của ma trận. Với các thuộc tính như: dòng, cột, giá trị.
4	Const	Chứa các biến static trong chương trình.
5	Player	Chứa thông tin người chơi và thời gian chơi của người chơi.

## B. Mẫu bảng mô tả các phương thức trong một lớp

*Bảng 3. Bảng phương thức*

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng khai báo	Sinh viên phụ trách
1	LoadChessBoard()	Khởi tạo 1 bảng game gồm 81 các Button với các thuộc tính.	ChessBoard.cs (53)	Huỳnh Xuân Huy
2	PrintSolution()	Chức năng của hàm Solve. Giải cả ma trận	ChessBoard.cs (91)	Võ Minh Hiếu

3	<p>AlwaysCheckIsOk(int [,] curMap)</p> <p>Input: curMap</p>  <pre> graph TD     Start([Start]) --&gt; curMap[curMap[9,9]]     curMap --&gt; Finding{Finding empty cell}     Finding -- YES --&gt; return0_1[return 0]     Finding -- NO --&gt; Cheking{Cheking the number that is filled with Soduku conditions}     Cheking -- YES --&gt; return1[return 1]     Cheking -- NO --&gt; return0_2[return 0]     return0_1 --&gt; Stop([Stop])     return1 --&gt; Stop     return0_2 --&gt; Stop   </pre> <p><i>Hình 4. Sơ đồ khối AllwaysCheckIsOk</i></p>	Liên tục kiểm tra điều kiện đúng của số nhập vào	ChessBoard.cs (107)	Võ Minh Hiếu
4	CreateNewMartix()	Reset lại các sự kiện và thuộc tính của Button. Đồng thời tạo các sự kiện và thuộc tính mới cho game mới.	ChessBoard.cs (137)	Võ Minh Hiếu
5	Hint()	Chức năng giải đúng một ô bất kì trong game.	ChessBoard.cs (204)	Võ Minh Hiếu
6	ProccessInsertText()	Lấy giá trị từ form Inputpad bỏ vào trong	ChessBoard.cs (211)	Võ Minh Hiếu

		button và đồng thời bỏ nó vào trong Stack để undo.		
7	DrawInput()	Tạo 9 button có giá trị từ 1 đến 9 ở form InputPad. Dùng để nhập số vào ChessBoard.	InputPad.cs (29)	Huỳnh Xuân Huy
8	<p>isOK(int curValue, int row, int col, int[,] matrix) [4]</p> <p>Input: curValue, row, col, matrix</p> <pre> isOK( curValue, row, col, S[9][9] )   for i←1 to 9     if S[row,i]= curValue       return 0   for i←1 to 9     if S[i,col]= curValue       return 0   a←⌊(x-1)/3⌋, b←⌊(y-1)/3⌋   for i←3a+1 to 3a+3     for j←3b+1 to 3b+3       if S[i,j]=kS[i,j]= curValue         return 0   return 1 </pre> <p><i>Hình 5. Mã giả isOK</i></p>	Kiểm số vừa nhập vào có hợp lệ với luật của trò chơi hay không?	Solution.cs (31)	Võ Minh Hiếu



9	<p>Solve_Sudoku() [5]</p> <p>Hình 6. Sơ đồ khối Solve_Sudoku()</p>	<p>Dùng thuật toán quay lui để giải ma trận Sudoku</p>	<p>Solution.cs (56)</p>	<p>Võ Minh Hiếu</p>
10	<p>ResetMatrix(int [,] martrix)</p>	<p>Reset về 0 các ma trận trước đó để chơi game mới.</p>	<p>Solution.cs (112)</p>	<p>Võ Minh Hiếu</p>
12	<p>DelCell</p>	<p>Xoá các số theo level</p>	<p>ChessBoard.cs (151)</p>	<p>Võ Minh Hiếu</p>
13	<p>Undo</p>	<p>Chức năng Undo</p>	<p>ChessBoard.cs (284)</p>	<p>Võ Minh Hiếu</p>
14	<p>Redo</p>	<p>Chức năng Redo</p>	<p>ChessBoard.cs (311)</p>	<p>Võ Minh Hiếu</p>

### 3. Thiết kế database

Database của game chỉ là 1 bảng chứa UserName và Password từ class Player.

Bảng 4: Bảng Database

TT	Tên bảng	Mục đích
1	NguoiiDung	Lưu thông tin của người chơi

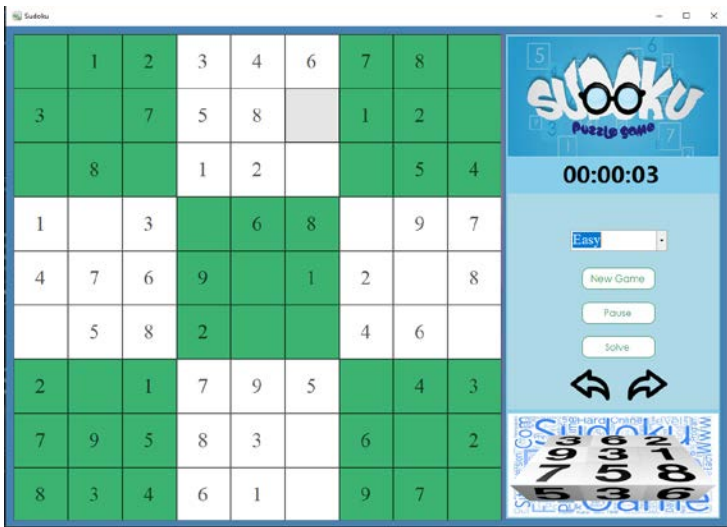
#### 4. Bảng mô tả các field trong bảng NguoiiDung

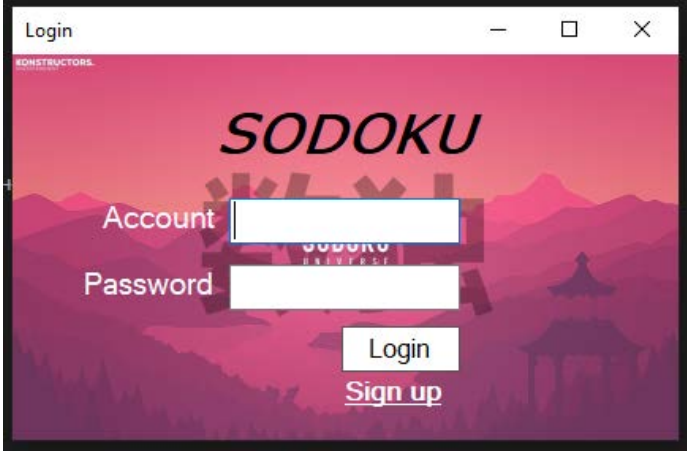
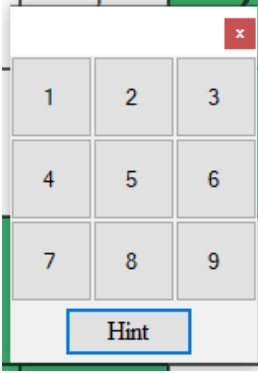
Bảng 5. Field trong bảng NguoiiDung

TT	Tên trường	Kiểu dữ liệu	Mục đích
1	TaiKhoan	Nchar(10)	Tên người chơi.
2	MatKhau	Nchar(10)	Mật khẩu người chơi.

#### 5. Thiết kế giao diện

Bảng 6. Bảng giao diện

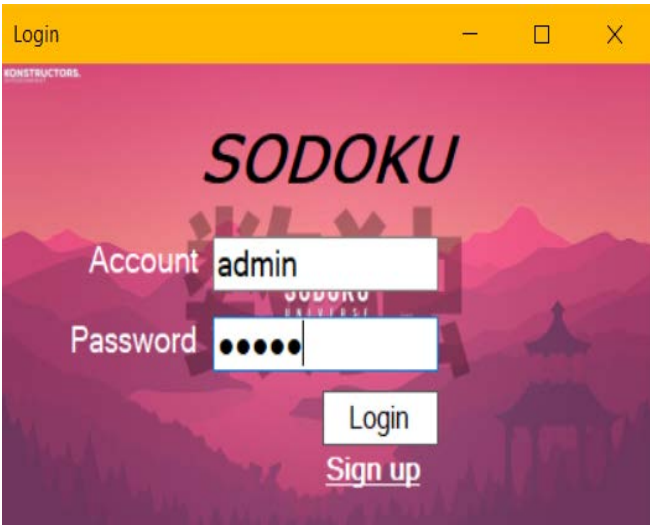

TT	Màn hình, cửa sổ	Mục đích	Giải thích
1	<p>Form1</p>  <p>Hình 7. Giao diện Form1</p>	Tạo form có bảng game để chơi.	Võ Minh Hiếu Form gồm 2 phần 1 panel chứa bảng game và 1 panel chứa các chức năng của game.


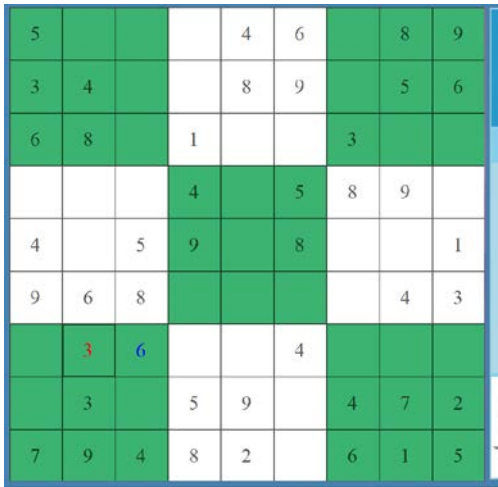
2	<p style="text-align: center;">Login</p>  <p style="text-align: center;"><i>Hình 8. Giao diện Login</i></p>	<p>Đăng nhập hoặc đăng ký để vào game</p>	<p>Huỳnh Xuân Huy Tạo ra đối tượng người chơi tương tác với game. Và lưu thông tin vào database cho lần sau</p>
3	<p style="text-align: center;">InputPad</p>  <p style="text-align: center;"><i>Hình 9. Giao diện InputPad</i></p>	<p>Ghi số vào trong bảng game.</p>	<p>Huỳnh Xuân Huy Tạo ra 1 form nhỏ để điền số vào trong bảng game, đồng thời kèm thêm chức năng Hint.</p>

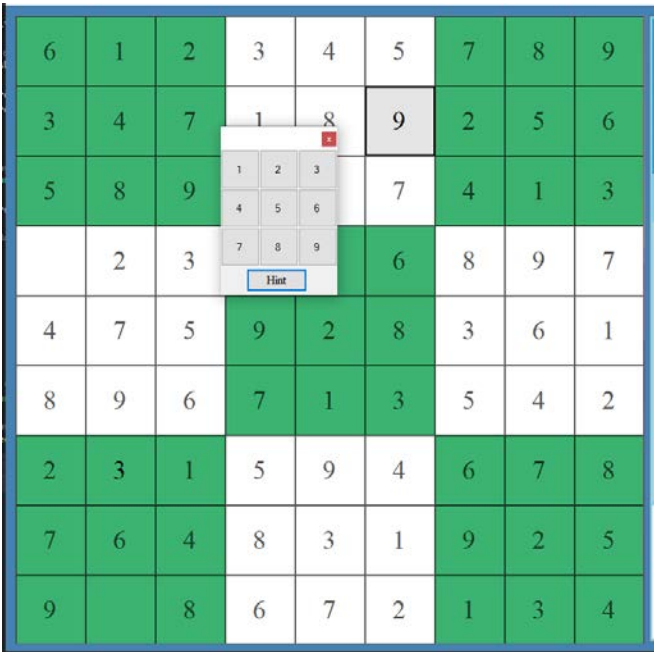
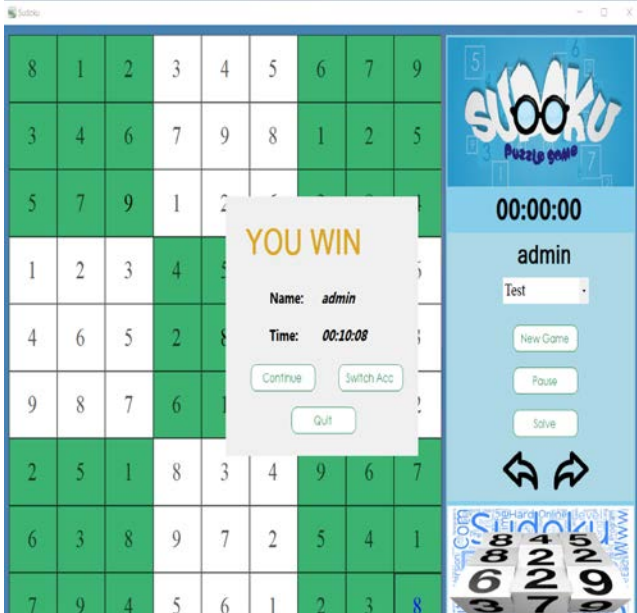
4	<div data-bbox="591 149 727 191" data-label="Section-Header"> <h3>FormWin</h3> </div> <div data-bbox="337 210 979 827" data-label="Form">  </div> <div data-bbox="488 848 899 890" data-label="Caption"> <p><i>Hình 10. Giao diện FormWin</i></p> </div>	<div data-bbox="1049 149 1206 369" data-label="Text"> <p>Hiện thông báo đã hoàn thành game.</p> </div>	<div data-bbox="1232 149 1468 722" data-label="Text"> <p>Võ Minh Hiếu          Khi ta điền đúng hết tất cả các ô trống, form sẽ hiện lên thông báo đã chiến thắng và hiện thời gian đã chơi và tên của người chơi.</p> </div>
---	---	--	---

## IV. CÀI ĐẶT KIỂM THỬ


Bảng 7. Bảng kiểm thử

TT	Tình huống dữ liệu vào và kết quả dự kiến	Mục đích	Giải thích
1	<p>Dữ liệu vào: username và password</p>  <p>Hình 11. Kết quả dự kiến kiểm thử 1(1)</p> <p>Kết quả dự kiến : hiện lên đăng nhập thành công , nếu không thành công thì người chơi phải nhập lại</p>  <p>Hình 12. Kết quả dự kiến kiểm thử 1 (2)</p>	Đăng nhập vào game	Kiểm tra kết nối với database

2	<p>Dữ liệu vào : chọn mức độ dễ khó của game ở mức test ( level =5)</p> <p>Kết quả dự kiến : hiện ra bảng game theo mức độ mà người chơi đã chọn ở trên.</p>  <p><i>Hình 13. Kết quả dự kiến kiểm thử 2</i></p>	<p>Kiểm tra xem game có Random đúng theo level không?</p>	<p>Vừa kiểm tra được tính đúng đắn của hàm tạo ma trận. Vì chỉ xoá 5 số, quá trình test sẽ nhanh hơn.</p>
3	<p>Dữ liệu vào : số 6 và 3</p> <p>Kết quả dự kiến: 6 thỏa mãn điều kiện game. sẽ có màu xanh, 3 không thỏa mãn sẽ có màu đỏ.</p>  <p><i>Hình 14. Kết quả dự kiến kiểm thử 3</i></p>	<p>Kiểm tra điều kiện của đúng của game.</p>	<p>Trong ô 3x3 thứ 7, 3 là số không thỏa mãn điều kiện game, 6 là số thỏa mãn điều kiện game.</p>

<p>4</p>	<p>Dữ liệu vào: Sự kiện click Button Hint</p> <p>Kết quả dự kiến: Hiện ra số đã được giải đúng.</p>  <p>Hình 15. Kết quả dự kiến kiểm thử 4</p>	<p>Kiểm tra chức năng Hint.</p>	<p>Ở vị trí trỏ chuột.</p> <p>Duy nhất số 2 là đúng so với điều kiện.</p>
<p>5</p>	<p>Dữ liệu vào: sự kiện click Button Solve</p> <p>Kết quả dự kiến: in ra tất cả các số được giải, báo thắng</p> 	<p>Kiểm tra chức năng Solve.</p>	<p>Các số được lưu trong solveMatrix đã được in ra màn hình.</p>

Hình 16. Kết quả dự kiến kiểm thử 5

6	<p>Dữ liệu đầu vào: sự kiện click Button Undo sau khi nhập 1 2 3 4 5 từ inputPad</p> <p>Kết quả dự kiến: Undo về số trước đó.</p> 	Kiểm tra chức năng Undo	Cell sau khi nhập đã được bỏ vào stack, và đã trả về được cell thứ 2 của đỉnh Stack.

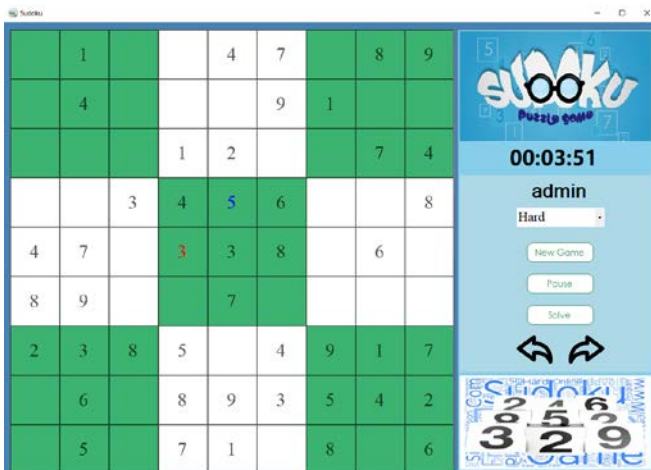


Hình 17. Kết quả dự kiến kiểm thử 6(1)



Hình 18. Kết quả dự kiến kiểm thử 6(2)

7 Dữ liệu vào: Sau khi bấm nút Undo, thì bấm nút Redo.  
Kết quả dự kiến: trả về số vừa Undo.



Hình 19. Kết quả dự kiến kiểm thử 7

Kiểm tra chức năng Redo

Cell được lấy ra từ Stack Undo đã được chuyển sang Stack Redo. Và Cell trên đỉnh Stack được lấy ra khi ấn phím Redo.

## V. KẾT LUẬN

### ***1. Kết quả đạt được***

Ứng dụng Window Form với ngôn ngữ C# để thiết kế một game.

Ứng dụng được các kiến thức của “lập trình hướng đối tượng”, “cấu trúc dữ liệu và giải thuật”, “cơ sở dữ liệu”, “kỹ thuật lập trình”, “lập trình windows” để giải quyết bài toán ứng dụng.

Chúng ta có 1 game Sudoku hoàn chỉnh. Có các chức năng cơ bản cho 1 game Sudoku như Undo, Redo, Giải cả ma trận, Giải một ô, Level.

### ***2. Khó khăn gặp phải***

Thiết kế các class theo hướng đối tượng và ứng dụng các tính chất của hướng đối tượng vào chương trình.

Hướng khắc phục là củng cố lại kiến thức về lập trình hướng đối tượng.

### ***3. Ưu điểm phần mềm***

Bất cứ một chương trình nào cũng sẽ có những ưu điểm riêng để mang lại trải nghiệm tốt nhất cho người chơi. Ưu điểm của phần mềm này là:

- Chức năng Hint giúp người chơi có thêm các gợi ý bổ ích trong quá trình chơi.
- Giao diện đơn giản dễ nhìn, dễ chơi.
- Lưu thông tin người chơi cho lần chơi tiếp theo.
- Có nhiều level thách thức người chơi.
- Có chức Undo và Redo khi nhập sai.
- Phân vùng 3x3 rõ ràng cho người chơi dễ phân biệt các số
- Có chức năng Hint giúp người chơi giải dễ dàng hơn.

### ***4. Nhược điểm phần mềm***

- Với những ma trận có nhiều trường hợp, thì hàm giải chỉ giải được 1 trường hợp.
- Vì hàm sinh ngẫu nhiên ma trận chạy chậm.
- Code chưa tối ưu hóa tốt nên tốn nhiều Ram của máy.

- Hiện tại Database chỉ có field UserName và Password, chưa thể lưu điểm bằng thời gian.

## ***5. Hướng phát triển phần mềm***

- Thiết kế giao diện đẹp hơn.
- Dùng User Controls thay vì mở từng form
- Database sẽ lưu thời gian người chơi.
- Update thời gian mới nếu người chơi đạt kỉ lục cao hơn
- Tạo chức năng viết nháp.
- Tạo bảng xếp hạng giữa các người chơi để so sánh.
- Thiết kế game trên WPF.

## TÀI LIỆU THAM KHẢO

- [1]. Trích dẫn\_Báo cáo đề án Cấu trúc Dữ Liệu và Giải Thuật của Lê Minh Tiến và Võ Minh Hiếu(03/12/2019): <https://github.com/minhtien300199/sudoku>
- [2]. Trích dẫn\_Tác giả Haha Học Nói\_Thế Giới Game(03/12/2019):  
<https://vn.trangcongnghes.com/the-gioi-games/179535-c225ch-choi-sudoku-cho-nguoi-moi.html>
- [3]. Trích dẫn\_Lớp 07T4, Nhóm 12A, Khoa công nghệ thông tin, Đại học Bách Khoa Đà Nẵng.(03/12/2019): <https://github.com/minhhieu023/Project1>
- [4]. Tham khảo ý tưởng từ tác giả Nguyen Van Truong(03/12/2019):  
<https://viblo.asia/p/thuat-toan-quay-lui-backtracking-bJzKmLbD59N>
- [5]. Nguồn thuật toán(03/12/2019): <https://www.geeksforgeeks.org/sudoku-backtracking-7/>

