**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**Môn học: Kỹ Thuật Lập Trình**

**ĐỀ TÀI:**

**GAME SUDOKU TRÊN WINFORM**

|  |  |
| --- | --- |
| Nhóm sinh viên thực hiện: Nhóm 9 | |
| 17110147 | Huỳnh Xuân Huy (nhóm trưởng) |
| 17110173 | Đặng Huỳnh Hoàng Long |
| 20110704 | Trần Minh Quang |
| 19110442 | Trần Nhất Quang |
| 19133022 | Hồng Tiến Hào |
| 19110397 | Phạm Minh Mẫn |

GVHD: TS. Nguyễn Thiên Bảo

*Thành phố Hồ Chí Minh, tháng 12 năm 2021*

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO CUỐI KỲ**

**ĐỀ TÀI:**

**GAME SUDOKU TRÊN WINFORM**

|  |  |
| --- | --- |
| Nhóm sinh viên thực hiện: Nhóm 9 | |
| 17110147 | Huỳnh Xuân Huy (nhóm trưởng) |
| 17110173 | Đặng Huỳnh Hoàng Long |
| 20110704 | Trần Minh Quang |
| 19110442 | Trần Nhất Quang |
| 19133022 | Hồng Tiến Hào |
| 19110397 | Phạm Minh Mẫn |

GVHD: TS. Nguyễn Thiên Bảo

*Thành phố Hồ Chí Minh, tháng 12 năm 2021*

ĐIỂM SỐ

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

**Giáo viên *hướng* dẫn**

**(*ký và ghi họ tên*)**

***Nguyễn* Thiên Bảo**

# 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, TS. Nguyễn Thiên Bảo, 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 Công Nghệ Thông Tin 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 kĩ 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**

[LỜI NÓI ĐẦU 4](#_Toc89104962)

[DANH MỤC CÁC HÌNH 7](#_Toc89104963)

[DANH MỤC CÁC BẢNG 8](#_Toc89104964)

[I. ĐẶC TẢ 9](#_Toc89104965)

[1. Giới thiệu game Sudoku 9](#_Toc89104966)

[2. Dữ liệu đầu vào 9](#_Toc89104967)

[3. Các tình huống sử dụng 10](#_Toc89104968)

[4. Giao diện dự kiến 10](#_Toc89104969)

[II. PHÂN CÔNG 11](#_Toc89104970)

[III. THIẾT KẾ 12](#_Toc89104971)

[1. Thuật toán 12](#_Toc89104972)

[2. Thiết kế lớp 14](#_Toc89104973)

[A. Các lớp được sử dụng trong chương trình 14](#_Toc89104974)

[B. Mẫu bảng mô tả các phương thức trong một lớp 14](#_Toc89104975)

[3. Thiết kế giao diện 18](#_Toc89104976)

[IV. CÀI ĐẶT KIỂM THỬ 20](#_Toc89104977)

[V. KẾT LUẬN 25](#_Toc89104978)

[1. Kết quả đạt được 25](#_Toc89104979)

[2. Khó khăn gặp phải 25](#_Toc89104980)

[3. Ưu điểm phần mềm 25](#_Toc89104981)

[4. Nhược điểm phần mềm 25](#_Toc89104982)

[5. Hướng phát triển phần mềm 26](#_Toc89104983)

[TÀI LIỆU THAM KHẢO 27](#_Toc89104984)

# DANH MỤC CÁC HÌNH

[Hình 1. Giao diện dự kiến 10](#_Toc89104993)

[Hình 2. Sơ đồ thuật toán quay lui. 13](#_Toc89104994)

[Hình 4. Sơ đồ khối AllwaysCheckIsOk 15](#_Toc89104995)

[Hình 5. Mã giả isOK 17](#_Toc89104996)

[Hình 6. Sơ đồ khối Solve\_Sudoku() 17](#_Toc89104997)

[Hình 7. Giao diện Form1 18](#_Toc89104998)

[Hình 9. Giao diện InputPad 19](#_Toc89104999)

[Hình 10. Giao diện FormWin 19](#_Toc89105000)

[Hình 13. Kết quả dự kiến kiểm thử 2 20](#_Toc89105001)

[Hình 14. Kết quả dự kiến kiểm thử 3 21](#_Toc89105002)

[Hình 15. Kết quả dự kiến kiểm thử 4 21](#_Toc89105003)

[Hình 17. Kết quả dự kiến kiểm thử 6(1) 23](#_Toc89105004)

[Hình 18. Kết quả dự kiến kiểm thử 6(2) 23](#_Toc89105005)

[Hình 19. Kết quả dự kiến kiểm thử 7 24](#_Toc89105006)

# DANH MỤC CÁC BẢNG

[Bảng 1. Bảng phân công 11](#_Toc89105007)

[Bảng 2. Thiết kế lớp 14](#_Toc89105008)

[Bảng 3. Bảng phương thức 14](#_Toc89105009)

[Bảng 6. Bảng giao diện 18](#_Toc89105010)

[Bảng 7. Bảng kiểm thử 20](#_Toc89105011)

# ĐẶC TẢ

## Giới thiệu game Sudoku

Sudoku là một trò chơi từng gây nghiện của các quốc gia phát triển. Độ phức tạpcủa 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.

## Dữ liệu đầu vào

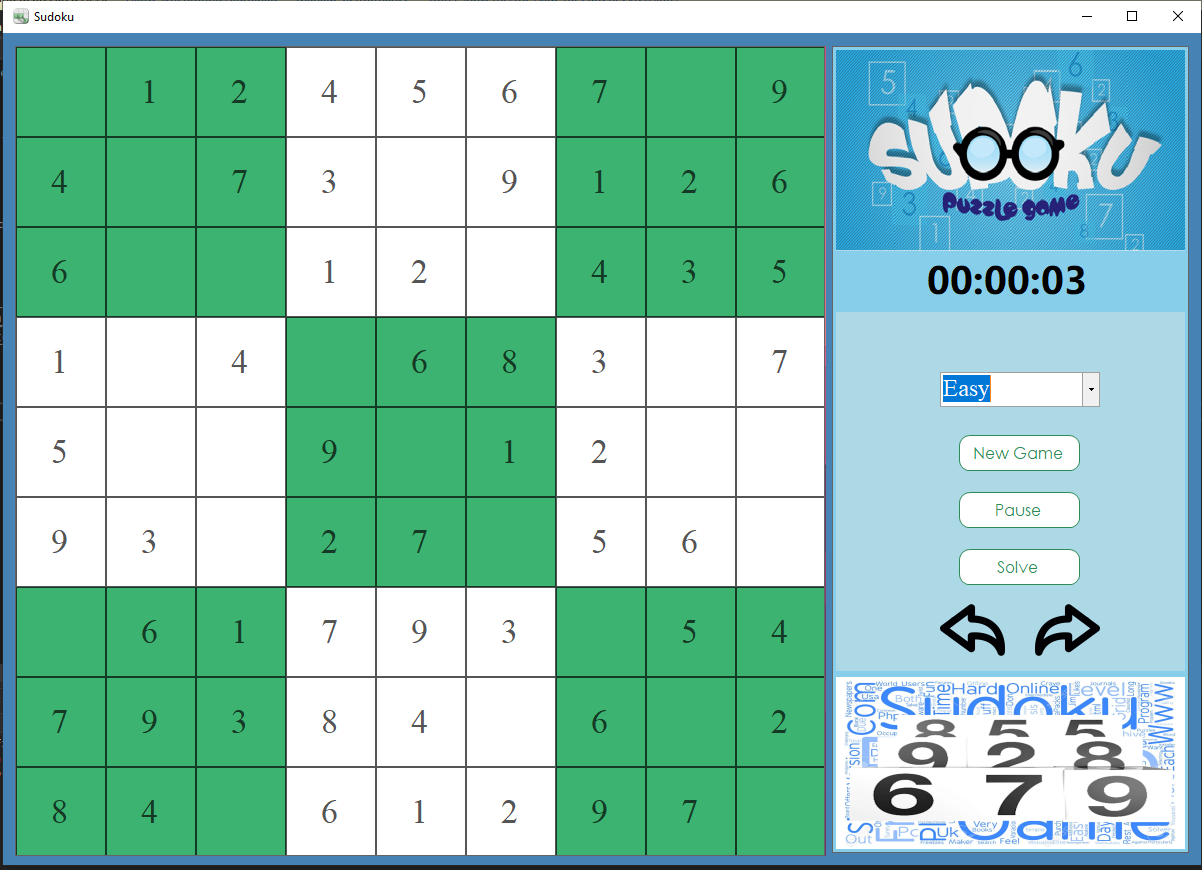
Dữ liệu đầu vào là 1 ma trận 9x9 với 81 ô với giới hạn các số được điền trước theo từng Level.

## 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.

## 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

# PHÂN CÔNG

###### Bảng 1. Bảng phân công

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên sinh viên | Mô tả công việc | Đóng góp |
| 1 | Huỳnh Xuân Huy | * Thiết kế các lớp cho đối tượng và chức năng cho phần mềm. * Fix lỗi trong quá trình viết. | 100% |
| 2 | Đặng Huỳnh Hoàng Long | * Thiết kế các chức năng cho phần mềm * Thiết kế các giao diện | 100% |
| 3 | Trẩn Minh Quang | * Viết báo cáo * Viết chức năng cho game | 100% |
| 4 | Trần Nhất Quang | * Viết báo cáo * Thiết kế các giao diện cho game | 100% |
| 5 | Hồng Tiến Hào | * Kiểm thử. * Thiết kế các chức năng cho game | 100% |
| 6 | Phạm Minh Mẫn | * Viết báo cáo * Tạo các button cho Panel bảng game và InputPad | 100% |

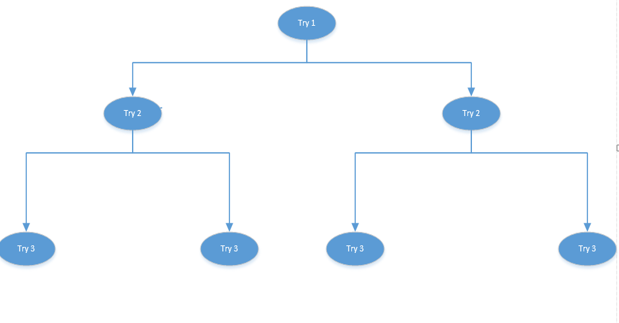
# THIẾT KẾ

## Thuật toán

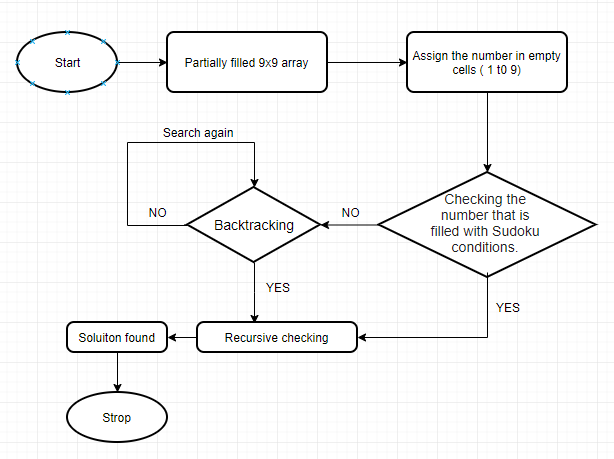
Xoay quanh cả trò chơi là thuật toán rất đơn giản đã học ở môn Cấu trúc dữ liệu và Giải Thuật đó là Quay lui (backtracking). Và thường thì thuật toán quay lui thường viết bằng lối đệ quy.

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 nửa 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 lối 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

## 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 sẳn.

### Các lớp được sử dụng trong chương trình

###### Bảng 2. 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. |

### 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  Thực  Hiện |
| 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) | Huỳnh  Xuân  Huy |
| 3 | AlwaysCheckIsOk(int [,] curMap)  Input: curMap    Hình 4. Sơ đồ khối AllwaysCheckIsOk | Liên tục kiểm tra điều kiện đúng của số nhập vào | | ChessBoard.cs  (107) | Đặng  Huỳnh  Hoàng  Long |
| 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) | Đặng  Huỳnh  Hoàng  Long |
| 5 | Hint() | Chức năng giải đúng một ô bất kì trong game. | | ChessBoard.cs  (204) | Trần  Minh  Quang |
| 6 | ProccessInsertText() | Lấy giá trị từ form Inputpad bỏ vào trong button và đồng thời bỏ nó vào trong Stack để undo. | | ChessBoard.cs  (211) | Trần  Minh  Quang |
| 7 | DrawInput() | Tạo 9 button có giá trị từ 1 đến 9 ở form InputPad. Dùng để nhập số vào ChessBoard. | | InputPad.cs  (29) | Trần  Nhất  Quang |
| 8 | isOK(int curValue, int row, int col, int[,] matrix) [4]  Input: curValue, row, col, matrix    Hình 5. Mã giả isOK | Kiểm số vừa nhập vào có hợp lệ với luật của trò chơi hay không? | | Solution.cs  (31) | Trần  Nhất  Quang |
| 9 | Solve\_Sudoku() [5]  Hình 6. Sơ đồ khối Solve\_Sudoku() | Dùng thuật toán quay lui để giải ma trận Sudoku | | Solution.cs  (56) | Hồng  Tiến  Hào |
| 10 | ResetMatrix(int [,] martrix) | Reset về 0 các ma trận trước đó để chơi game mới. | | Solution.cs  (112) | Hồng  Tiến  Hào |
| 11 | DelCell | Xoá các số theo level | | ChessBoard.cs  (151) | Phạm  Minh  Mẫn |
| 12 | Undo | Chức năng Undo | | ChessBoard.cs (284) | Phạm  Minh  Mẫn |
| 13 | Redo | Chức năng Redo | | ChessBoard.cs  (311) | Phạm  Minh  Mẫn |

## Thiết kế giao diện

###### Bảng 6. Bảng giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình, cửa sổ | Mục đích | Giải thích |
| 1 | Form1    Hình 7. Giao diện Form1 | Tạo form có bảng game để chơi. | Form gồm 2 phần 1 panel chứa bảng game và 1 panel chứa các chức năng của game. |
| 2 | InputPad    Hình 9. Giao diện InputPad | Ghi số vào trong bảng game. | Tạo ra 1 form nhỏ để điền số vào trong bảng game, đồng thời kèm thêm chức năng Hint. |
| 4 | FormWin    Hình 10. Giao diện FormWin | Hiện thông báo đã hoàn thành game. | 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. |

# 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 | Dữ liệu vào : chọn mức độ dễ khó của game ở mức test ( level =5)  Kết quả dự kiến : hiện ra bảng game theo mức độ mà người chơi đã chọn ở trên.    Hình 13. Kết quả dự kiến kiểm thử 2 | Kiểm tra xem game có Random đúng theo level không? | Vừa kiểm tra được tính đúng đắn của hàm tạo ma trận. Vì chỉ xoá 5 số, quá trình test sẽ nhanh hơn. |
| 2 | Dữ liệu vào : số 6 và 3  Kết quả dự kiến: 6 thoả mãn điều kiện game. sẽ có màu xanh, 3 không thoả mãn sẽ có màu đỏ.    Hình 14. Kết quả dự kiến kiểm thử 3 | Kiểm tra điều kiện của đúng của game. | Trong ô 3x3 thứ 7, 3 là số không thoả mãn điều kiện game, 6 là số thoả mãn điều kiện game. |
| 3 | Dữ liệu vào: Sự kiện click Button Hint  Kết quả dự kiến: Hiện ra số đã được giải đúng.    Hình 15. Kết quả dự kiến kiểm thử 4 | Kiểm tra chức năng Hint. | Ở vị trí trỏ chuột. Duy nhất số 2 là đúng so với điều kiện. |
| 4 | Dữ liệu vào: sự kiện click Button Solve  Kết quả dự kiến: in ra tất cả các số được giải, báo thắng    *Hình 16. Kết quả dự kiến kiểm thử 5* | Kiểm tra chức năng Solve. | Các số được lưu trong solveMatrix đã được in ra màn hình. |
| 5 | Dữ liệu đầu vào: sự kiện click Button Undo sau khi nhập 1 2 3 4 5 từ inputPad  Kết quả dự kiến: Undo về số trước đó.    Hình 17. Kết quả dự kiến kiểm thử 6(1)    Hình 18. Kết quả dự kiến kiểm thử 6(2) | 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. |
| 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 tra từ Stack Undo đã được chuyển sang Stack Redo. Và Cell trên đỉnh Stack được lấy ra khi ấn phím Redo. |

# KẾT LUẬN

## 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”, “kĩ 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.

## 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.

## 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 phầm 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.
* 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.

## 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 ngốn nhiều Ram của máy.

## Hướng phát triển phần mềm

* Thiết kế giao diện đẹp hơn.
* Dùng User Controls thay vì mở từng form
* Kết hợp sử dụng với Database
* 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\_Tác giả Haha Học Nói\_Thế Giới Game(03/12/2019): <https://vn.trangcongnghe.com/the-gioi-games/179535-c225ch-choi-sudoku-cho-nguoi-moi.html>

[2]. 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>

[3]. Tham khảo ý tưởng từ tác giả Nguyen Van Truong(03/12/2019): <https://viblo.asia/p/thuat-toan-quay-lui-backtracking-bJzKmLbD59N>

[4]. Nguồn thuật toán(03/12/2019): <https://www.geeksforgeeks.org/sudoku-backtracking-7/>