**Tên đề tài:**

**<Áp dụng thuật toán k láng giềng gần vào bài toán mã đi tuần>**

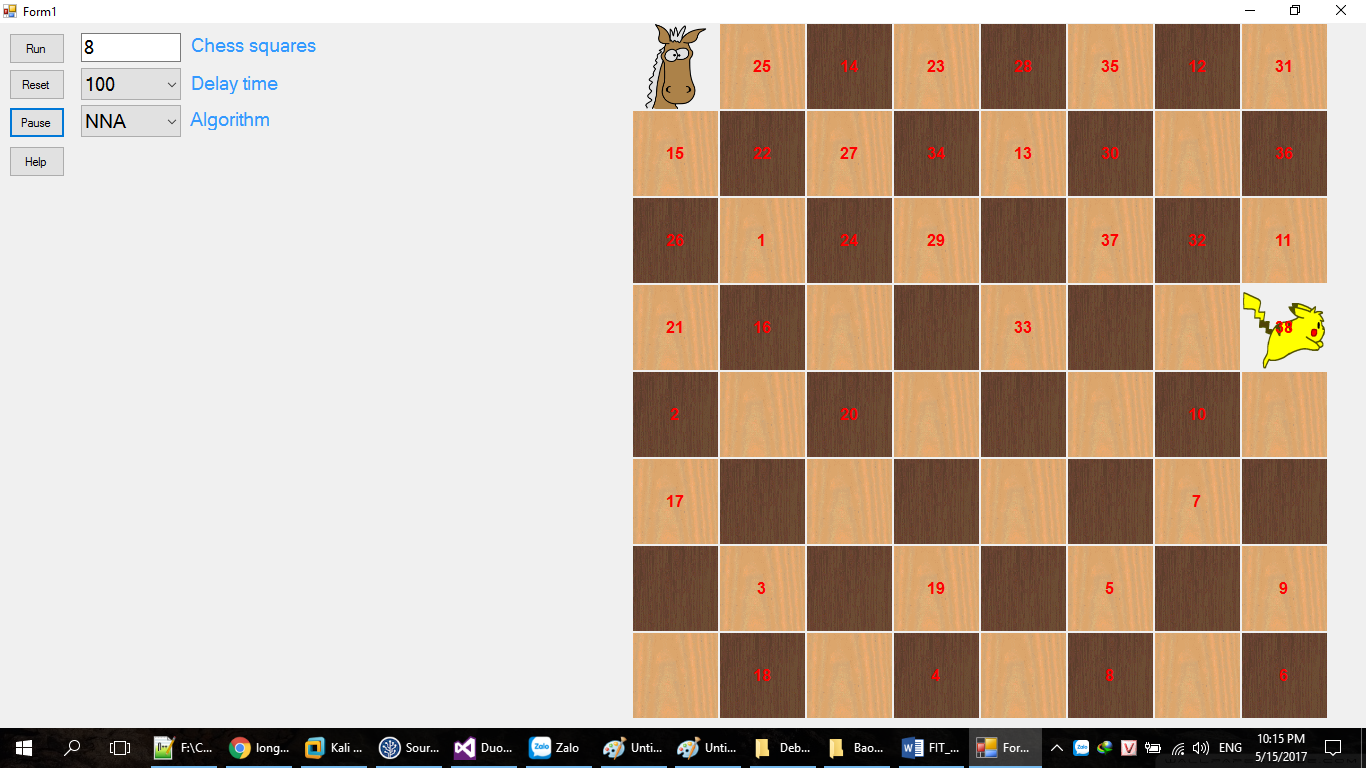
**Danh sác sinh viên thực hiện:**

**<1642042 – Huỳnh Phương Nam>**

**<1642067 – Nguyễn Lê Quang Thắng>**

**<1642071 – Trịnh Quốc Thịnh>**

**<1642084 – Trần Lê Tuấn>**



# Giới thiệu

* Mã đi tuần là bài toán mô tả về việc di chuyển một quân mã trên bàn cờ vua (n x n). Quân mã được đặt ở một ô trên một bàn cờ trống nó phải di chuyển theo quy tắc của cờ vua để đi qua mỗi ô trên bàn cờ đúng một lần. Trong đề tài thú vị này, nhóm xin giới thiệu về bài toán và những điều có thể khai thác được qua bài toán.

# Nội dung

* Bài toán mã đi tuần có rất nhiều phương pháp giải khác nhau như vét cạn, quay lui, quay lui áp dụng duyệt ưu tiên, thuật toán k láng giềng gần. Nhóm cài đặt thuật toán NNA (Nearest Neighbor Algorithm – Láng giềng gần nhất) để giải quyết bài toán này.

# Tóm tắt

* Ý tưởng giải bài toán : Tìm đường đi hamilton
* Từ vị trí hiện tại của quân mã, chọn ô láng giềng kế cận sao cho số nước đi có thể có từ ô này là ít nhất.
* Mô tả thuật toán NNA cho bài toán Knight Tour:

B1: Từ ô X hiện tại tìm ô kế cận Y sao cho số nước đi có thể của Y là khác 0 và ít nhất.

B2: Nếu Y không tồn tại dừng thuật toán.

B3: Đánh dấu đã đi qua Y.

B3: Ngược lại gán X = Y lặp lại B1

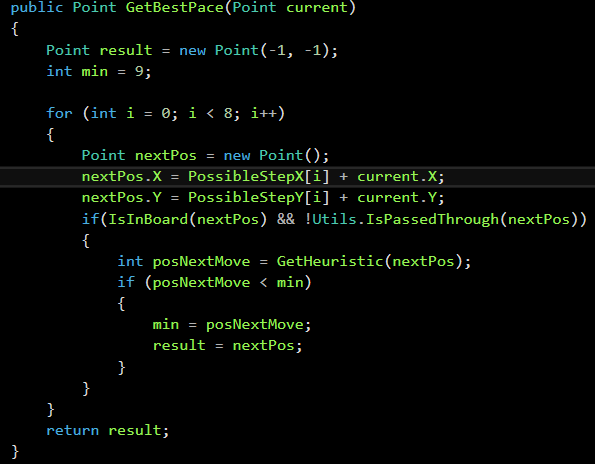
* Chi tiết:

Thuật toán được đặt trong 1 vòng lặp là Timer với interval được thiết lập trên màn hình, dùng queue để lưu nước đi kế tiếp, mục đích là để hỗ trợ real-time interactive trên UI.

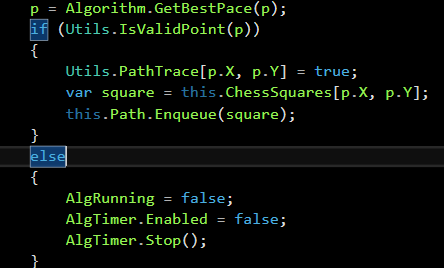
* Tìm nước đi kế tiếp (NextPace) cho quân mã:

Hàm GetBestPace trả về 1 object Point là nước đi tốt nhất kế tiếp, cụ thể

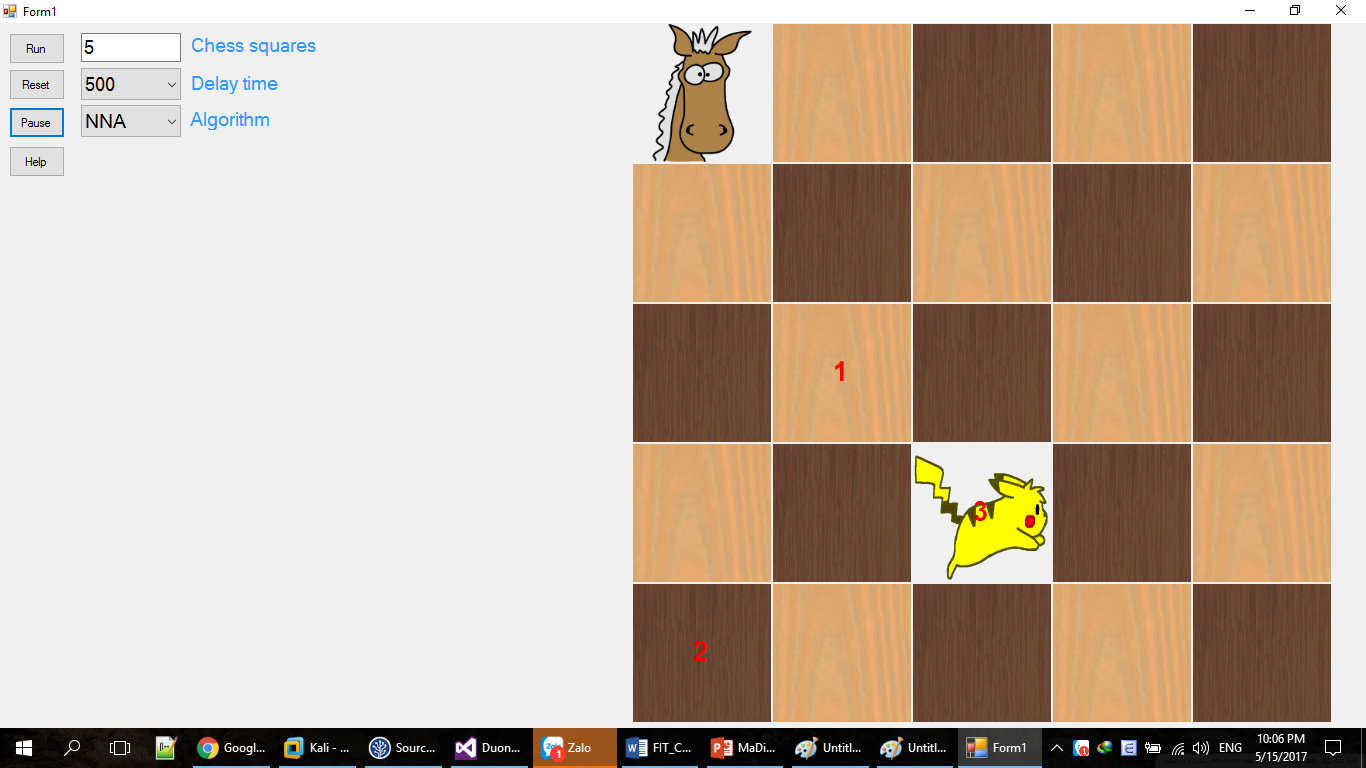
1. Khởi tạo result là ô (-1,-1)
2. Duyệt qua 8 ô con mã có thể đi tiếp, gọi là Pi
3. Nếu Pi chưa từng được đi qua và nằm trên bàn cờ, kiểm tra số nước đi (H) của Pi với min
4. Nếu H < min, gán min = H, gán result = Pi
5. Trả về result



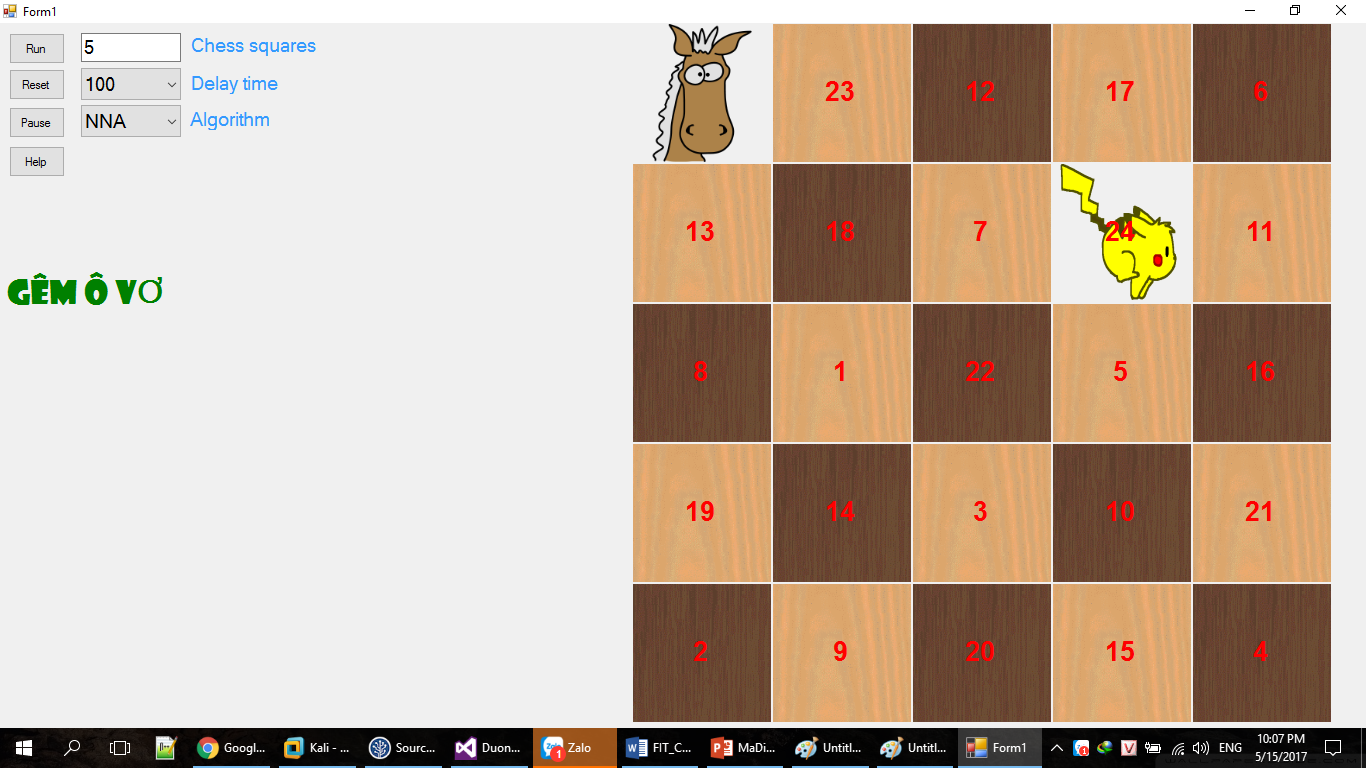
* Nếu NextPace khác (-1;-1), push NextPace vào queue, đánh dấu đã đi qua NextPace
* Nếu NextPace bằng (-1;-1) nghĩa là không còn đường đi cho quân mã, kết thúc thuật toán, stop Timer



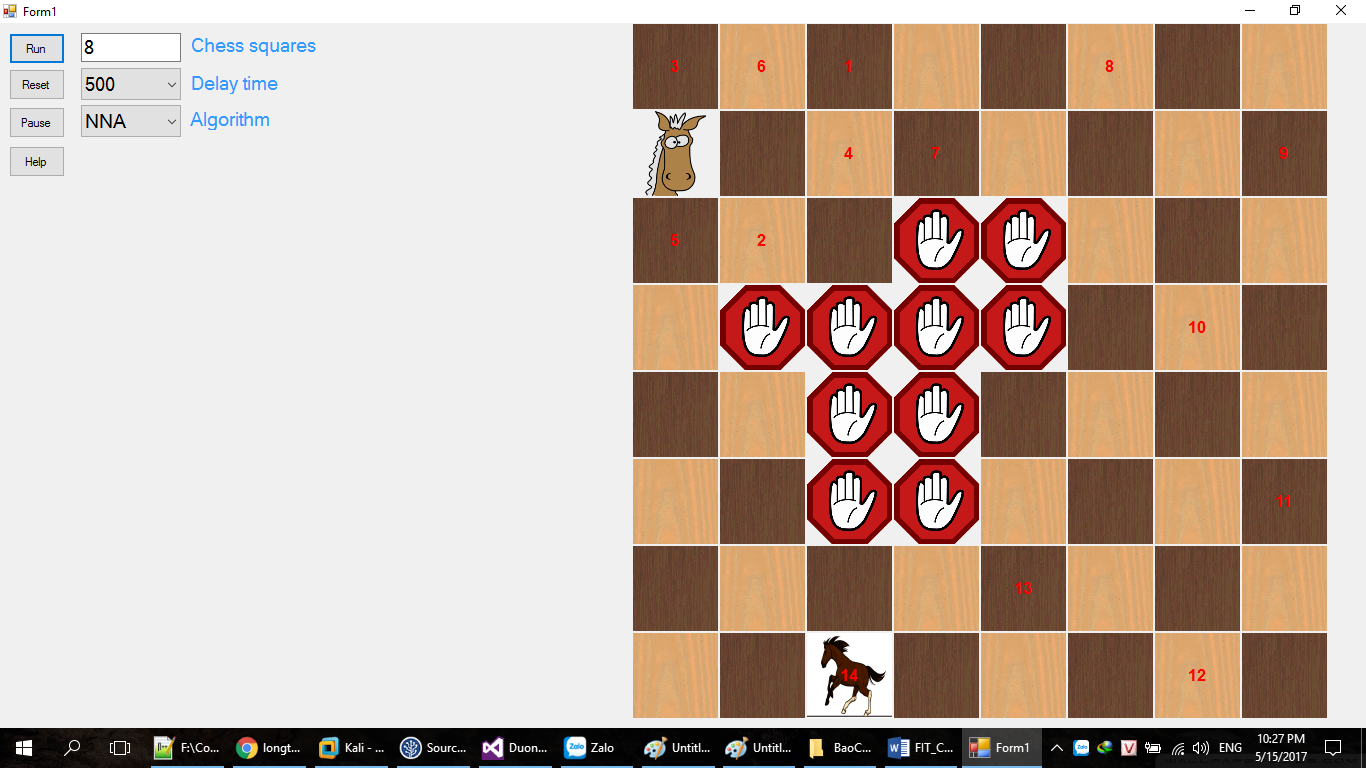
* Trong khi queue vẫn còn items, Timer event pick item ra và vẽ lên UI



* Nếu queue không còn item, kết thúc drawing, dừng Timer.



* Trong quá trình chạy thuật toán có thể setup các chướng ngại vật (khóa ô)



* Khi kết thúc, tồn tại 1 đường đi cho con mã khi tất cả các ô trên bàn cờ sẽ được đánh số (trừ các ô đã khóa), ngược lại là không có một đường đi nào như vậy.

# Đánh gi

* NNA chạy nhanh hơn các thuật toán truyền thống (brute-force, backtracking, backtracking)
* Trong hầu hết các trường hợp đều tìm đường đường đi (đi hết bàn cờ)

# Mã nguồn

* IDE: Visual studio 2015
* Ngôn ngữ: C#
* File thực thi nằm tại bin/Debug/DuongDiConNgua.exe
* Yêu cầu .NET framework v4.0 hoặc v4.5

# Tài liệu tham khảo

* <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>
* https://vi.wikipedia.org/wiki/B%C3%A0i\_to%C3%A1n\_m%C3%A3\_%C4%91i\_tu%E1%BA%A7n