**BÁO CÁO TIẾN ĐỘ BTL GIỮA KÌ + CUỐI KÌ**

**MÔN HỌC: TRÍ TUỆ NHÂN TẠO VÀ ỨNG DỤNG**

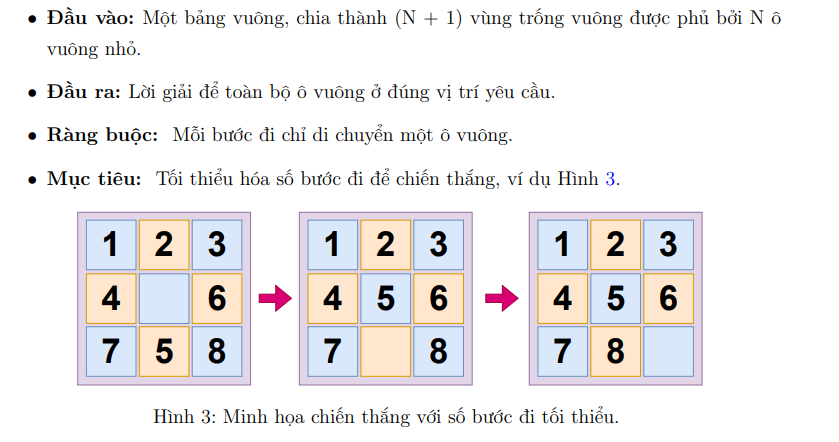
**Nhóm 17: MÃ LỚP 142076: HỌC KÌ 2022.2**

Trần Tiến Duy : 20192807 : Nhóm trưởng

Vũ Mạnh Hà : 20192817

* **Đề tài :** Ứng dụng trí tuệ nhân tạo tìm số bước đi tối thiểu chiến thắng trò chơi N-puzzle
* **Mô hình bài toán :**
* Đầu vào : một bảng vuông , chia thành (N+1) vùng tống vuông đc phủ bưởi N ô vuông nhỏ ( *được lựa chọn số lần xáo trộn các ô vuông* )
* Đầu ra : Lời giải để toàn bộ ô vuông ở đúng vị trí yêu cầu
* Ràng buộc : Mỗi bước đi chỉ di chuyển một ô vuông
* Mục tiêu : Tối thiểu hóa số bước đi để chiến thắng

Vd minh hoạt như hình sau



* **Thuật toán đề xuất :**

**1 : Các chiến lược tìm kiếm cơ bản** ( tìm kiếm theo chiều rộng BFS và sâu dần )

**2 : Các chiến thuật tìm kiếm tri thức** ( Tìm kiếm tham lam và thuật oán A\* liên quan ( không trọng số , trọng số ... ) )

* **Ứng dụng của đề tài :**
* Phù hợp với nhiệm vụ, khối lượng công việc cho BTL môn học.
* Mô hình hóa bài toán N-puzzle dưới dạng tìm kiếm trên đồ thị.
* So sánh được điểm mạnh của từng thuật toán để áp dụng vào từng bài toán.
* Áp dụng đúng những kiến thức đã được học trên lớp và tự tìm hiểu thêm .
* **Kết quả mong muốn :**
* Xây dựng một app cơ bản giao diện phù hợp dễ tương tác và sử dụng, có thể tự chạy ra kết quả
* Kết quả đạt được phù hợp với những mục tiêu ban đầu đặt ra ( Mức độ phù hợp của từng thuật toán và bài toán N với N=8 15 24 , cùng với số lần xáo )
* Sẽ có một báo cáo chung tổng kết các kết quả chung đạt được sau nhiều lần chạy để tìm ra kết luận ( thuật toán nào tối ưu. Chấp nhận những trường hợp máy tính không tính toán ra được do hạn chế năng lực tính toán của phần cứng … )
* **Framework dự kiến**
* Ngôn ngữ lập trình **Python** + **Thư viện PyQt5** trong đó PyQt nói chung là một liên kết Python cho Qt , một công cụ phát triển xây dựng giao diện nguồi dùng
* **Kế hoạch phân chia công việc ( updated 1/6 )**

|  |  |
| --- | --- |
| **Vũ Mạnh Hà** | **Trần Tiến Duy** |
| * Tìm hiểu và xây dựng thuật toán ( chiều rộng và sâu dần ) vào BTL * Đề xuất thuật toán phù hợp * Xây dựng trương trình và các thuật toán sẽ sử dụng trong trương trình   *Cập nhật thêm* | * Tìm hiểu và áp dụng các thuật toán( Tham lam và A\* ) vào BTL * Ưng dụng của các hàm Heuristic vào BTL * Xây dựng kịch bản ( 8 15 24 Puzzle ) * Xây dựng trương trình ( Sử dụng thư viện PyQt5 ) và các thuật toán sẽ sử dụng trong trương trình   *Cập nhật thêm …* |

* **Kế hoạch các tuần tiếp theo ( dự kiến )**
* Tiếp tục thực hiện các công việc đã được phân
* Xây dựng trương trình , giao diện cơ bản của app

Số trạng thái có thể giải được mà một N-Puzzle có thể có được tính bằng (N+1)!/2, do đó đối với 8-Puzzle, chúng ta có 9!/2 trạng thái có thể giải được là 181.440.

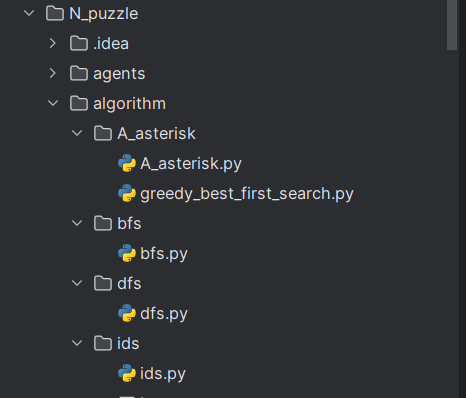
Xây dựng bảng xắp xếp các giải thuật với 3 kịch bản

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **THUẬT TOÁN** | | **Kịch bản 1** | | **Kịch bản 2** | | **Kịch bản 3** | |
| Thời gian  (s) | Số bước | Thời gian  (s) | Số bước | Thời gian  (s) | Số bước |
| **1** | **Chiều rộng :BFS** |  |  |  |  |  |  |
| **2** | **Sâu dần : IDS** |  |  |  |  |  |  |
| **3** | **Tham lam**  **(Manhattan)** |  |  |  |  |  |  |
| **4** | **A\* (Số ô sai vị trí )** |  |  |  |  |  |  |
| **5** | **A\*(Manhattan)** |  |  |  |  |  |  |
| **6** | **A\***  **(Manhattan trọng số )** |  |  |  |  |  |  |
| **7** | **A\*( Mâu thuẫn tuyến**  **Tính )** |  |  |  |  |  |  |
| **8** | **Tham lam (Mâu thuẫn**  **tuyến tính )** |  |  |  |  |  |  |

Update : 01/06/2023 :

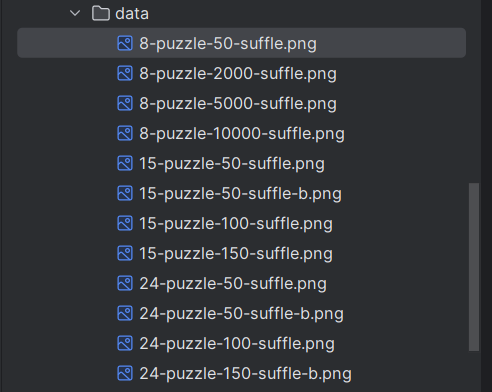
Tìm kiếm và xây dựng các giải thuật phục vụ cho bài toán gồm : A\* Greedy, BFS , IDS , DFS …

Chú ý : Ngôn ngữ lập trình Python



Xây dựng cơ sở dữ liệu tạm thời cho bài toán (gồm các trường hợp N=3, 4, 5 cùng số lần tráo .)

*Chú ý : từ N=4 trở đi là số lần tráo tối đa là 150 trong bài toán này vì số lần tráo càng nhiều . bài toán càng phức tạp => số lần quét các trường hợp nhiều hơn gây tốn dung lượng RAM- >Chương trình dễ bị Dump :*



Ảnh có chứa ảnh chụp màn hình, văn bản, phần mềm

Mô tả được tạo tự động

Xây dựng Khung giao diện của trương trình (ui1 )Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Tiếp tục cải thiện giao diện chính của chương trình (ui5)



Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

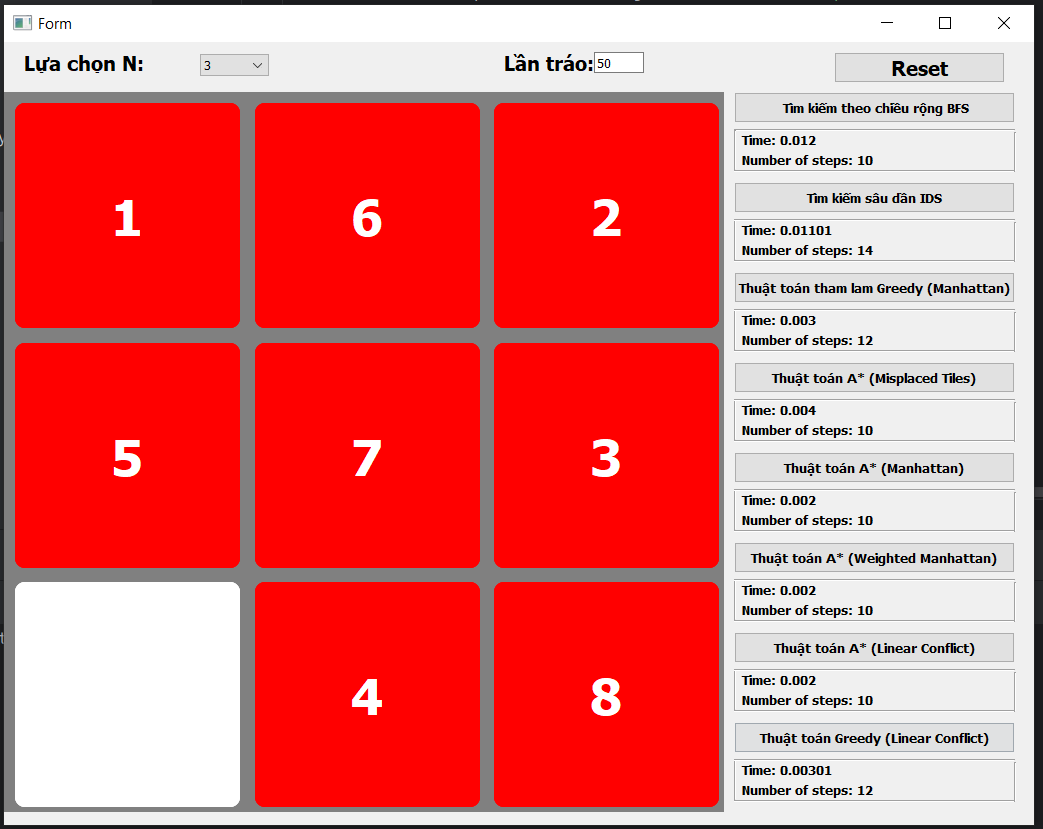
Giao diện khi chạy độc lập game

Chạy thử chương trình với N=3 cùng kết quả một số thuật toán tính ra được :

Thời gian thuật toán tìm ra số bước đi tối thiểu và số bước đi tối thiểu .

Trong ví dụ dưới :

* Thuật toán BFS, A\* ( trọng số và không có trọng số và mâu thuẫn tuyến tính ) cho bước đi tối thiểu là 10 .
* Thuật toán IDS cho bước đi nhiều hơn là 14 bước .
* Thuật toán Tham lam (Greedy ) cho bước đi là 12 bước .

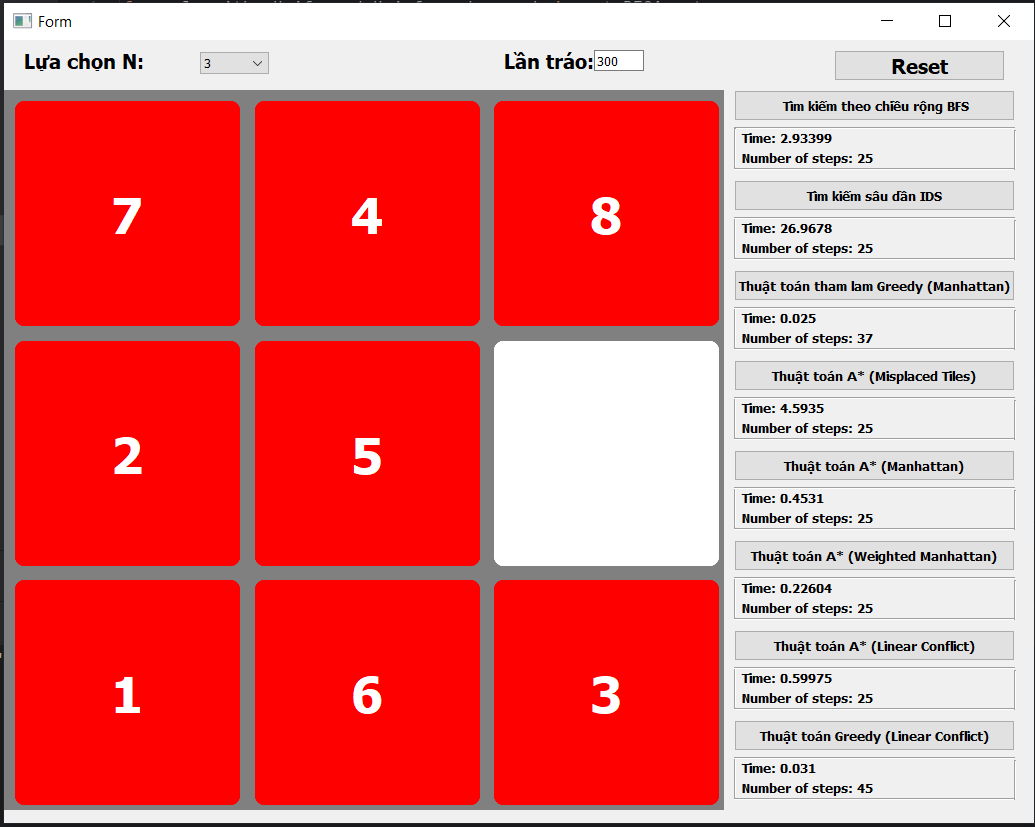


Trong thời gian tới : nhóm tiếp tục xây dựng trương trình để khi ấn các phím trên bàn phím trương trình sẽ tự chạy lần lượt các bước… theo 2 cách sau :

**Cách 1 :** Về đích với số bước đi tối thiểu, bằng đúng với số bước đi mà chương trình tính toán ra ( phù hợp với tất cả các trường hợp N= 3 4 5 )

**Cách 2 :** Về đích với số bước đi , cùng hướng di chuyển do người dùng tự điều khiển … ( phù hợp với trường hợp N=3 )

**Updated 20/06/2023**

****

Kết quả thu được khi chạy thử trương trình N=3 và số lần tráo 300

**Nhận xét :**

* Số lần tráo càng nhiều thì đa phần thuật toán sẽ chạy lâu hơn, cả thời gian và số bước di chuyển
* Số lần tráo < 100 thì thuật toán sẽ chạy nhanh hơn, cả thời gian và số bước di chuyển