|  |  |
| --- | --- |
| **Công việc** | **Người làm** |
| Hình ảnh cho giao diện | Như |
| Code tầm nhìn | Hiển |
| AI level 1,2 cho 1 food | Như |
| AI level 1,2 cho nhiều food | Hiển |
| AI level 3 | Hiển |
| AI level 4 | Luân |
| Tạo map | Phương |
| Hàm đọc file | Phương |
| Code chỉnh lại file input phù hợp level | Luân |
| Code A\* cho ghost | Phương |
| Hàm quay hướng cho pacman | Như |
| Code di chuyển cho người chơi | Luân |
| Code tạo giao diện map | Phương |
| Ghi file | Phương |
| Chạy test | Hiển |
| Viết báo cáo | Luân, Hiển |
| Viết kiểm tra win, surrender, game over | Như |

**Project 1 Môn AI**

Danh sách nhóm:

|  |  |
| --- | --- |
| **Tên thành viên** | **MSSV** |
| Hà Thế Hiển | 18127097 |
| Khưu Vĩ Luân | 18127146 |
| Dương Ngọc Nguyên Phương | 18127187 |
| Đặng Nguyễn Quỳnh Như | 18127172 |

Phân công

Môi trường: là python

Độ hoàn thành công việc:

Level 1: Thành công, them được có nhiều food nhưng nhiều food chưa tối ưu lắm và dùng cho level 4

Level 2: Thành công. Như level 1 nhưng xem ghost giống tường

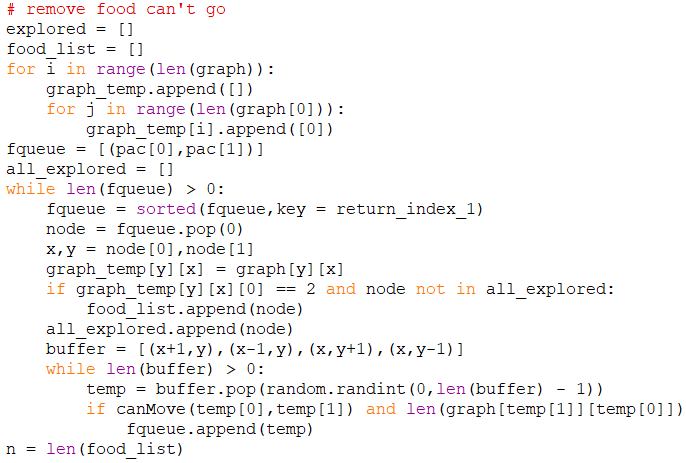
Level 3: Chưa thành công lắm. Do khi nó chưa xác định trung tâm của ma thì mặc định những nước đi xung quanh ma đều là trung tâm

Level 4: Chưa thành công lắm

Giải thích thuật toán:

**Level 1,2:**

**Xóa những food không thể đi**

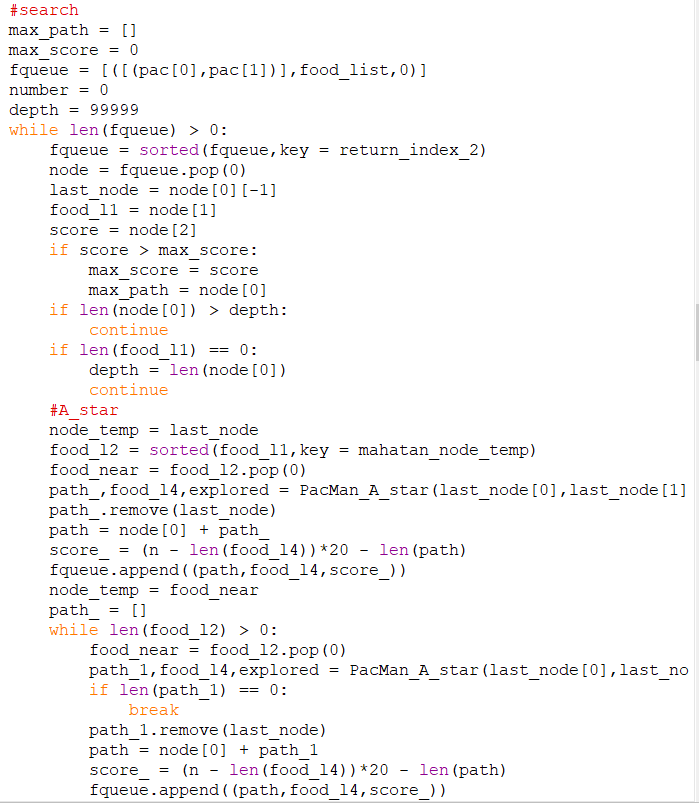


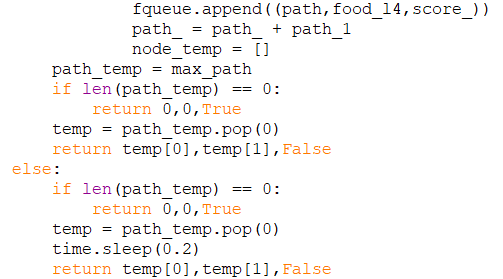
**Search theo score của từng đường đi (nó sẽ tính đường đi trước rồi) chạy pacman trong 1 lần**

Sẽ tìm con đường theo node mahatan nhỏ nhất thì chạy A\* (h = mahatan + g)

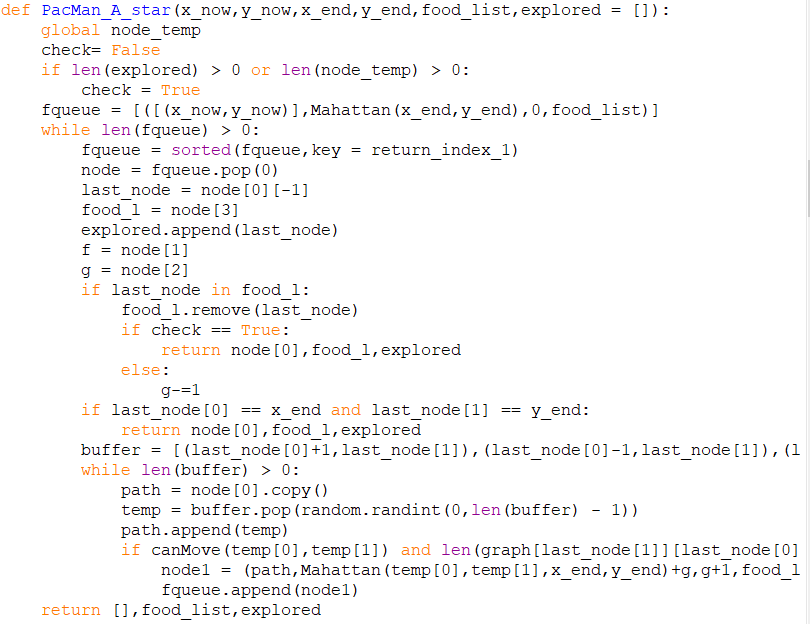
Còn những node còn lại sẽ không được đường trùng nhau và chạy breath-first-search trong hàm PacMan\_A\_star với (h = g)

Cuối cùng sẽ return đường có nhiều điểm nhâts





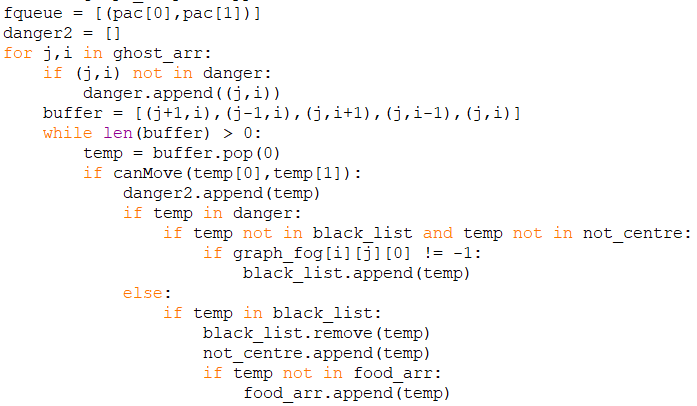
**Hàm PacMan\_A\_star**



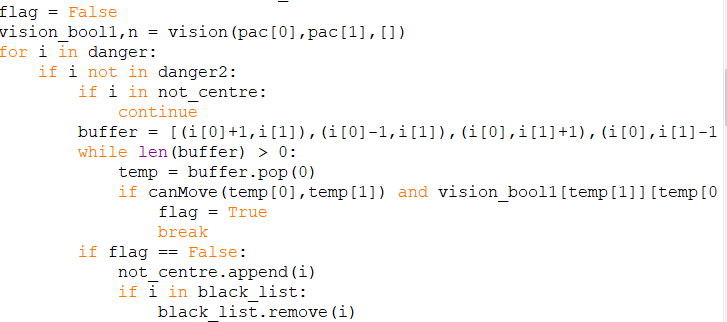
Chuyển thành BFS

**Level 3:**

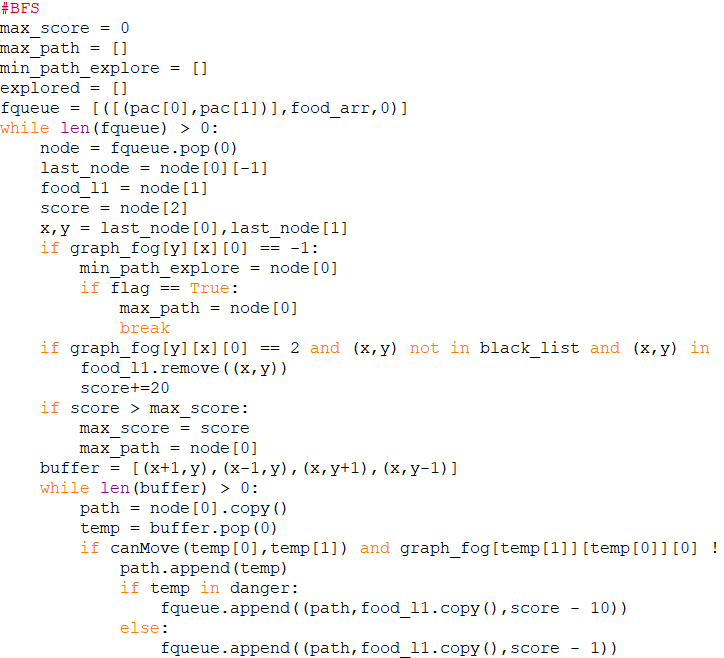
**Lưu các vị trí xung quanh ghost vào danger2 và danger lưu vị trí ma đứng lên**

****

**Nếu vị trí là danger và nó ko nằm trong danger2 thì nó không phải là trung tâm của ma**



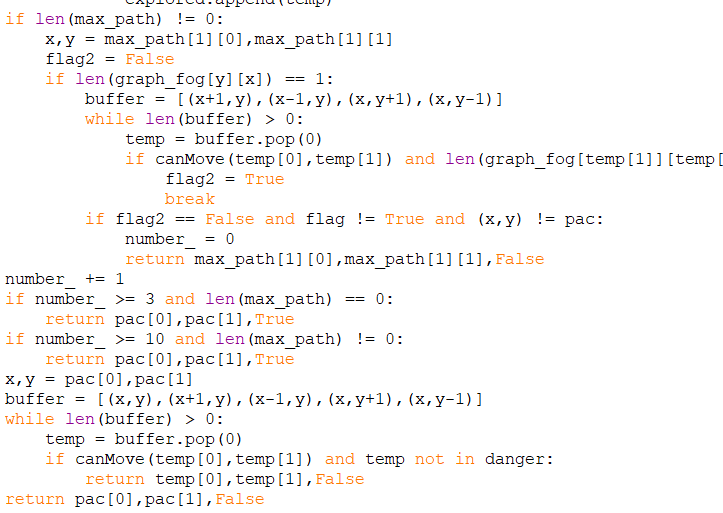
**Search tìm food theo BFS có A\* với đường nào có điểm lớn hơn thì chạy trước**

****

Nếu qua nơi danger thì giảm score

**Nếu có đường đi gặp ghost hoặc không đường đi thì nó đi random những chố không trong danger**

**Max random không đường đi là 3 và random do ghost chặn đường là 10**



**Level 4:**

**Tìm một list những food và ghost nhìn thấy được và có một đường đi xác định từ vị trí của pacman ở thời điểm đang xét (food có thể xác định đường đến ăn hoặc ghost có khả năng gây hại cho pacman)**

**A screenshot of a cell phone

Description automatically generated**

**Nếu ở vị trí hiện tại không thấy được food lẫn ghost, ta thực hiện tìm đường đi có thể mở rộng tầm nhìn nhiều nhất có thể cho đến khi nhìn thấy được food hoặc ghost**

**A screenshot of a cell phone

Description automatically generated**

**Nếu chỉ thấy food mà không thấy ghost, ta dùng thuật toán A\* search để tìm đến vào ăn tất cả food mà ta tìm thấy**

**A screenshot of a cell phone

Description automatically generated**

**Nếu ở điểm hiện tại có sự xuất hiện của ghost thì ta dùng reflex agent với hàm f(s) = (khoảng cách giữa pacman và food gần nhất) + 3\*(tổng số food còn lại) – 4\*(khoảng cách giữa pacman và ghost gần nhất) , khi đó ta càng đi xa khỏi ghost và càng gần với food thì càng tốt, lúc đó giá trị hàm f sẽ càng thấp và ta chỉ cần chọn tới nước đi có giá trị f thấp nhất trong số nước đi có thể đi**

* Hàm f:

**A screenshot of a cell phone

Description automatically generated**

* Thuật toán search sử dụng hàm f bên khi có ghost xuất hiện trong tầm nhìn:

**A screenshot of a social media post

Description automatically generated**

**Tạo map:**

Ta tạo map theo format sau vào file input.txt:

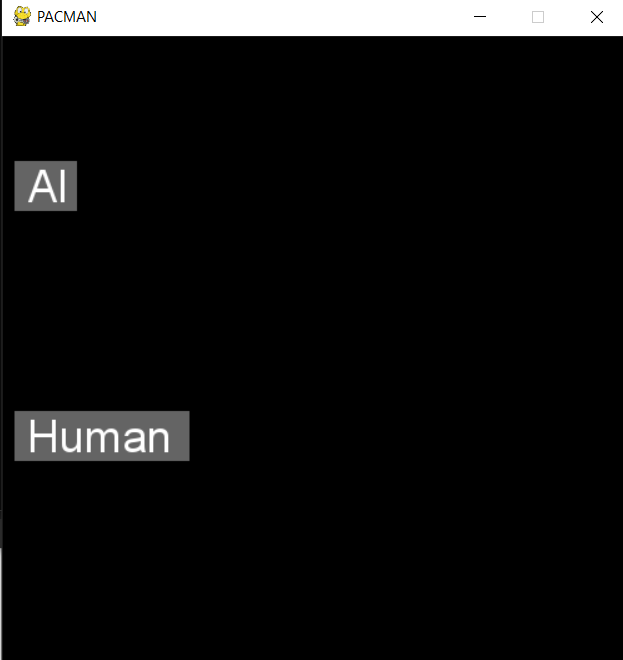
A close up of a logo

Description automatically generated

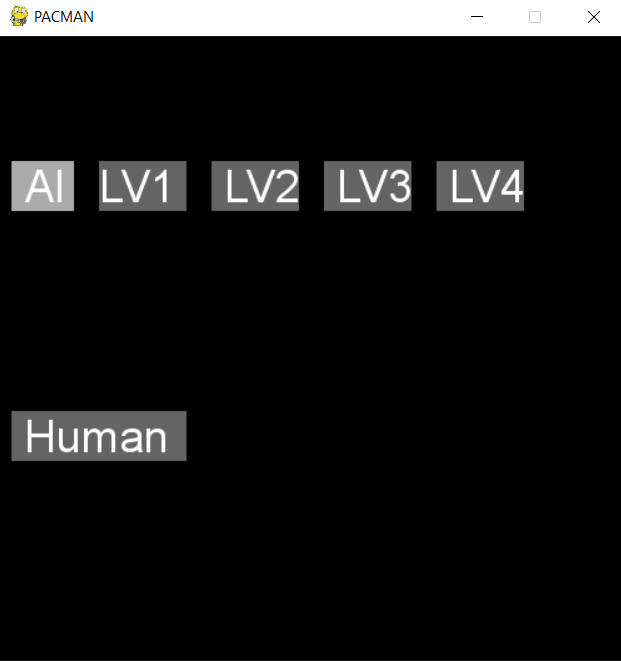
* Trong đó, dòng đầu tiên biểu thị kích cỡ của map, số dòng đứng trước và sau đó là số cột
* Tiếp đến là map được biểu thị theo không gian 2 chiều, trong đó, 0 là đường đi, 1 là tường, 2 là food, 3 là ghost
* Dòng cuối cùng chính là vị trí bắt đầu của pacman theo tọa độ được xác định là “X Y”, X biểu thị cột được bắt đầu bằng 0 từ trái sang phải, Y biểu thị dòng được bắt đầu bằng 0 từ trên xuống dưới

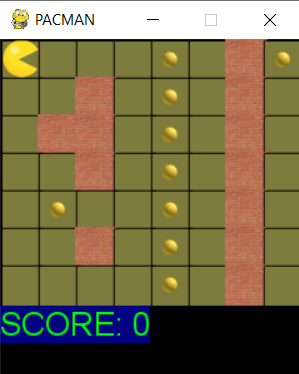
**Chạy chương trình:**

Double-click vào file project01.py sẽ hiện lên 1 window như sau



Sau khi bấm AI (AI có 4 level, còn Human chỉ có level 4)

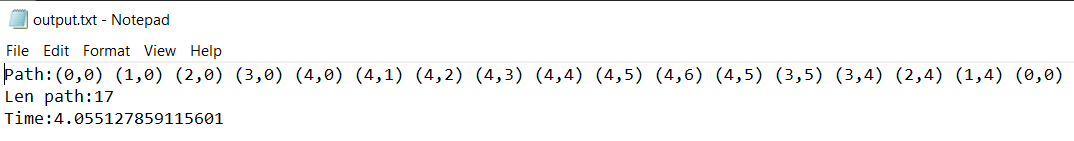




Nếu graph sương mù

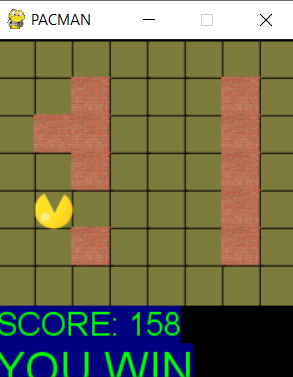
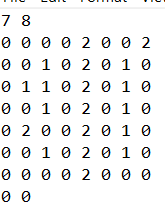
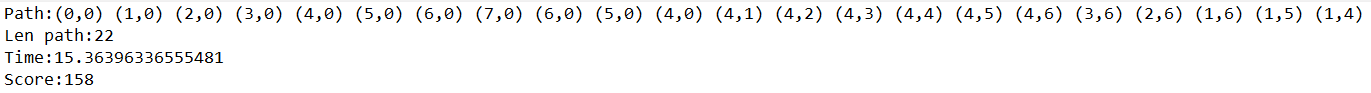


File output

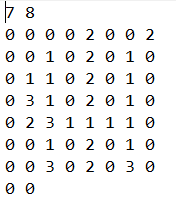
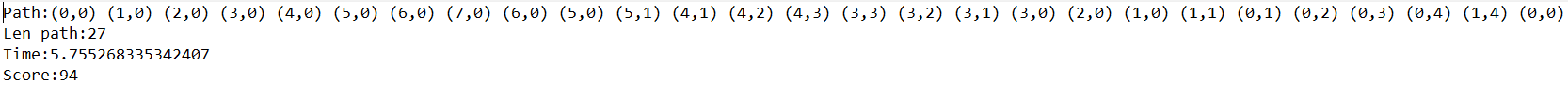


Test case

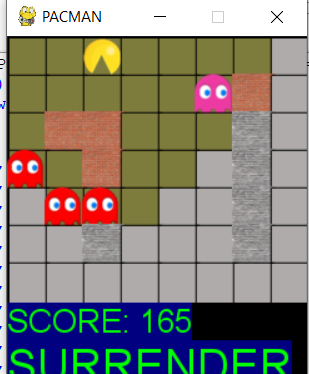
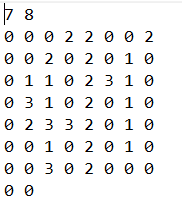
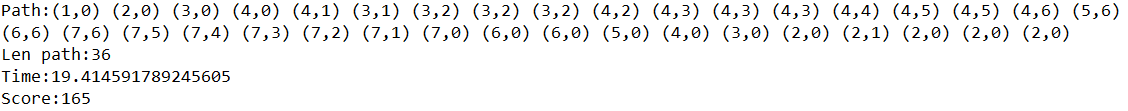
**Level 1:**

****

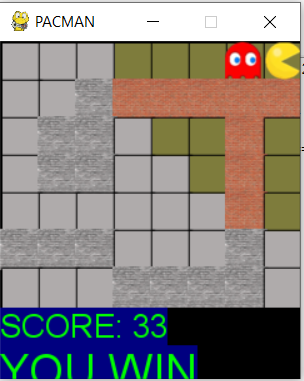
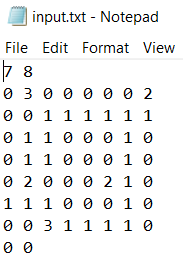
**Level 2:**

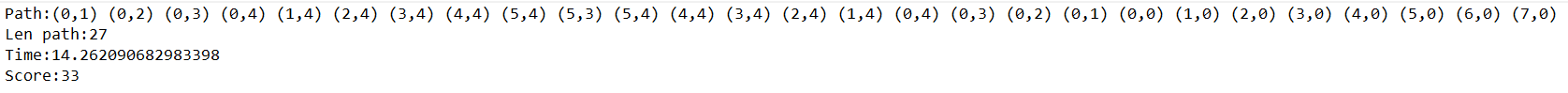
****

**Level 3:**

****

**Level 4:**

****

****

**Tham khảo:**

Level 4 được tham khảo dựa trên project multiagent pacman của Berkeley để tìm thuật toán adversarial search hợp lý cho pacman, link tham khảo: http://ai.berkeley.edu/multiagent.html