**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------o0o---------



BÁO CÁO ĐỒ ÁN

Bộ môn: Cơ sở trí tuệ nhân tạo

Giáo viên hướng dẫn: Lê Ngọc Thành, Hồ Thị Thanh Tuyền

**Đề tài: Search in Pac-man World**

1. **Thông tin nhóm**
2. 18127072 Huỳnh Lâm Hoàng Đại
3. 18127202 Đinh Lê Trọng Tài
4. 18127265 Nguyễn Hoàng Minh
5. 18127226 Bùi Thị Anh Thư
6. **Ý tưởng thực hiện**
7. **Level 1**

Sử dụng thuật toán tìm kiếm A\* với hàm heuristic tính bằng khoảng cách Manhattan từ vị trí đích đến (thức ăn)

Mỗi trạng thái (state) là vị trí tương ứng của pacman trên bản đồ.

Mỗi successor là vị trí xung quanh vị trí hiện tại mà pacman có thể di chuyển đến (không ngoài phạm vi bản đồ và không phải tường chắn)

Mục tiêu là vị trí của đồ ăn trong bản đồ

1. **Level 2**

Sử dụng thuật toán tìm kiếm A\* tương tự level 1

Mỗi successor là vị trí xung quanh vị trí hiện tại mà pacman có thể di chuyển đến (không ngoài phạm vi bản đồ, không phải vị trí quái vật và không phải tường chắn)

1. **Level 3**

Level 3 gồm 2 giai đoạn đan xen là khám phá và tìm kiếm.

* Khám phá:

Giai đoạn khi pacman chưa xác định được vị trí thức ăn, hoặc không có giải pháp để đi đến vị trí thức ăn đã biết.

Giải pháp: Trong phạm vi quan sát được, chọn ô có khoảng cách Manhattan lớn nhất từ vị trí hiện tại và có đường đi ngắn nhất từ ví trí hiện tại đến đó.

* Tìm kiếm:

Giai đoạn tìm đường đi đến vị trí thức ăn (trong tầm nhìn hiện tại hoặc tầm nhìn trước đó đã được lưu lại)

Sử dụng thuật toán tìm kiếm A\* với phạm vi đã quan sát được.

Xuyên suốt quá trình di chuyển, pacman duy trì 1 map riêng.

Map riêng của pacman sẽ được cập nhật mỗi khi mở rộng phạm vi đã quan sát được.

Trong 1 phạm vi quan sát bất kì, nếu có quái vật trong tầm nhìn, pacman sẽ lưu lại vị trí quái vật lúc đó. Nếu quái vật di chuyển tối thiểu 3 ô trong phạm vi mà pacman quan sát được thì phạm vi di chuyển của quái vật được xác định.

Pacman sẽ ưu tiên chỉ di chuyển vào ô nếu phạm vi 2 bước xung quanh ô đó không có quái vật.

Sau 4 lần thử nghiệm nếu luôn có quái vật trong phạm vi 1 bước quanh ô sắp đi đến, pacman sẽ cập nhật lại map và di chuyển tìm đường khác.

Nếu không còn đường khác hoặc đồ ăn khác khả thi thì pacman sẽ tự dừng game.

1. **Level 4**

Sử dụng thuật toán Minimax kết hợp Alpha-Beta pruning

Mỗi trạng thái game được lưu trong 1 dictionary GameState:

gameState = {"map": [a 2d array map], "pacmanPos": (x,y), "ghostsPos": [(x,y)] "gameScore": int, "nFoods": int}

Monster: Sử dụng thuật toán tìm kiếm A\* với đích đến là vị trí pacman

Pacman: Sử dụng hàm đánh giá độ tối ưu của từng trường hợp dựa trên các yếu tố:

* Điểm của pacman: Điểm của Pacman ảnh hưởng bởi số bước đi và số lượng thức ăn đã ăn được. Điểm số của trạng thái được cập nhật sau mỗi bước đi. Mục tiêu của pacman là đạt được số điểm cao nhất có thể. Trong hàm đánh giá, điểm của Pacman được nhân với 1 trọng số dương.
* Số thức ăn còn lại: Số thức ăn còn lại trong 1 trạng thái càng nhiều thì độ tối ưu của trạng thái đó càng ít. Nên trong hàm đánh giá, số thức ăn còn lại được lấy nghịch đảo.
* Tổng khoảng cách từ pacman đến quái vật: Pacman cố gắng đi đến vị trí mà khoảng cách đến quái vật là lớn nhất. Khoảng cách được dùng để tính toán là khoảng cách Manhattan. Trong hàm đánh giá, tổng khoảng cách đến vị trí các quái vật được tính toán, lấy nghịch đảo và nhân với 1 trọng số âm. => Khoảng cách càng gần thì độ ưu tiên càng thấp, trạng thái đó càng không tối ưu.

Tỉ lệ giữa các trọng số sử dụng trong hàm đánh là có được từ thực nghiệm dựa trên nhiều map và các test case khác nhau. Tỉ lệ này chỉ mang tính tương đối và chưa hoàn toàn tối ưu.

1. **Môi trường biên dịch và thực thi**

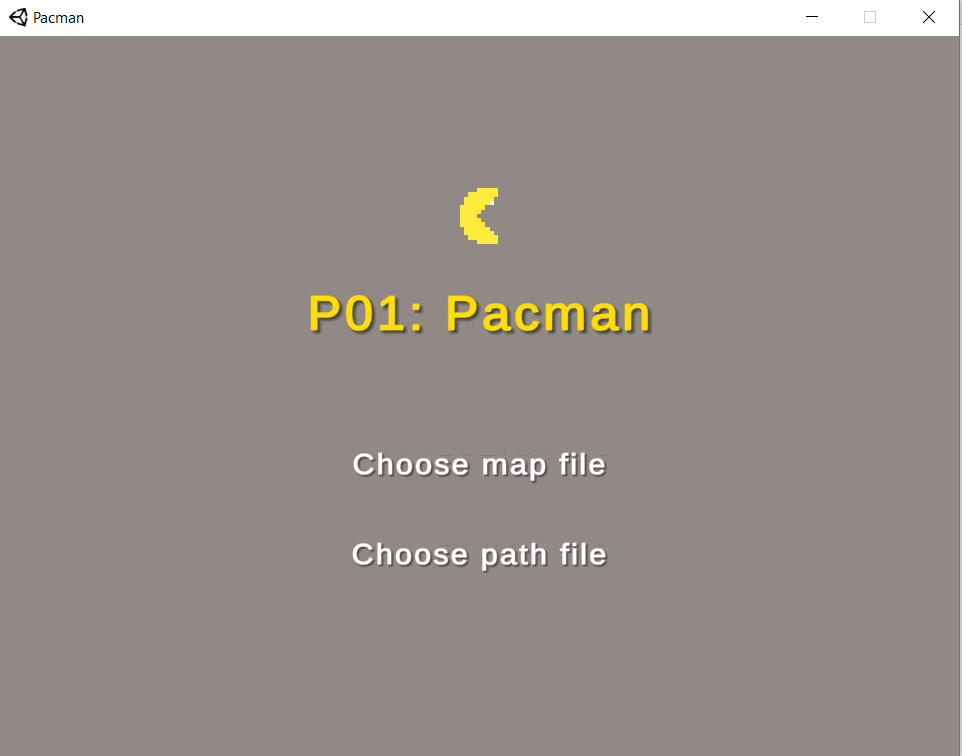
Kết quả minh họa cụ thể trong file txt đính kèm. Thực thi bằng file *‘Pacman.exe’*

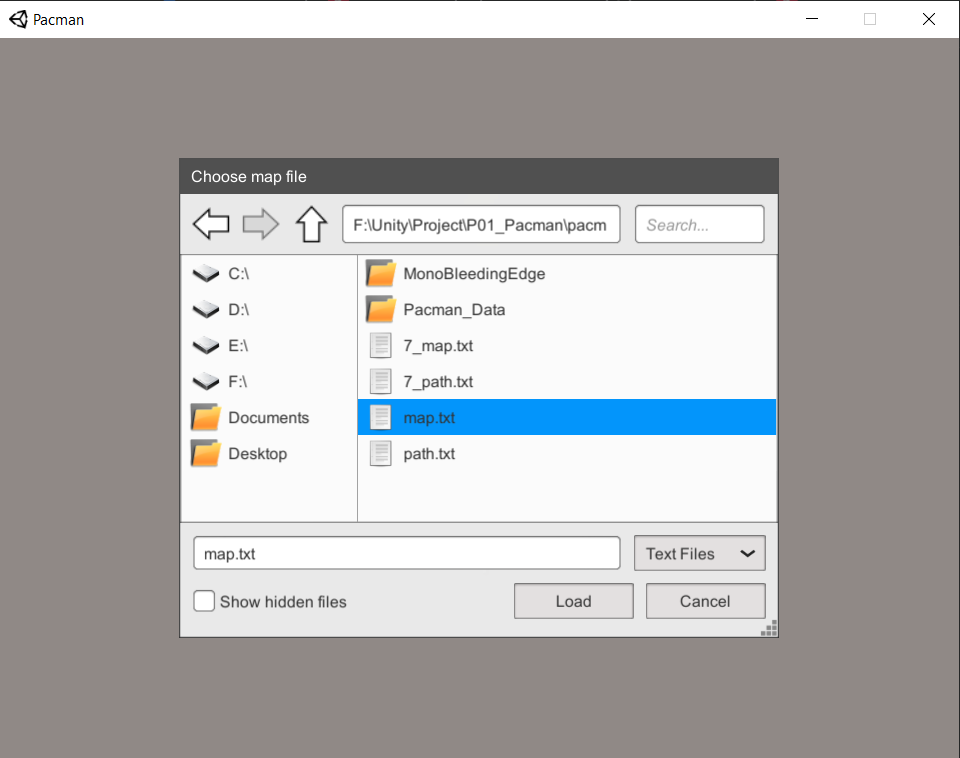
Chương trình gồm 2 phần:

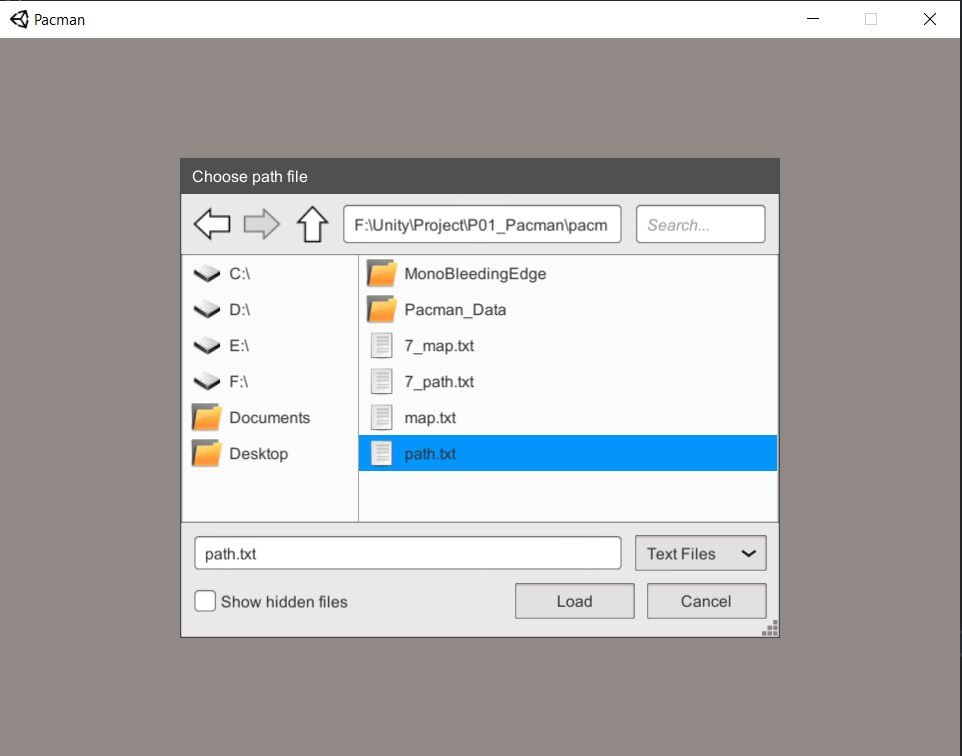
1. Thực thi thuật toán và trả về kết quả dưới dạng txt file
   * Môi trường và ngôn ngữ: ***Python 3***
2. Sử dụng thư viện đồ họa để minh họa kết quả trả về
   * Ngôn ngữ: ***C#***
   * Engine: ***Unity3D***
   * Môi trường thực thi: ***Windows***

Hướng dẫn:

* **B1**: Khởi chạy file .py và nhập đường dẫn file map và level tương ứng. Sau khi đã thực thi xong tập tin .py, chương trình sẽ trả về kết quả dưới dạng txt file có tên là ‘path.txt’.
* **B2**: Sau đó, khởi chạy tập tin ‘Pacman.exe’
* **B3**: Chọn đường dẫn của file map và file path tương ứng để xem minh họa cho kết quả trả về





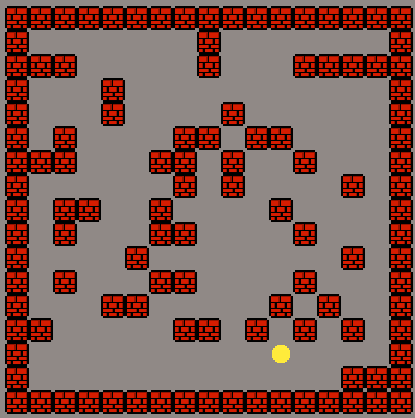
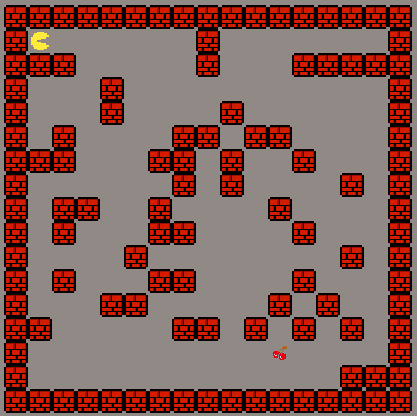


1. **Nhận xét, đánh giá**

|  |  |  |  |
| --- | --- | --- | --- |
| Level | Nhận xét | Mức độ hoàn thành | Mức độ tối ưu |
| 1 | Sử dụng A\* graph search.  BFS: Tối ưu chỉ khi đồ ăn ở gần vị trí pacman  DFS: Nếu có nhiều đường đi đến đồ ăn, DFS có thể trả về đường đi dài, không tối ưu | 100% | 100% |
| 2 | Sử dụng A\* graph search.  BFS: Tối ưu chỉ khi đồ ăn ở gần vị trí pacman  DFS: Nếu có nhiều đường đi đến đồ ăn, DFS có thể trả về đường đi dài, không tối ưu | 100% | 100% |
| 3 | Thuật toán sẽ không tối ưu nếu trong tầm quan sát chỉ ghi nhận được ít hơn 3 vị trí của 1 quái vật, dẫn đến không xác định được phạm vi di chuyển của quái vật. | 100% | 95% |
| 4 | Thuật toán sẽ chậm đối với những map có state space lớn (ít tường, kích thước map lớn). Depth càng lớn, thuật toán càng tối ưu đồng thời thời gian càng lớn. | 100% | 95% |

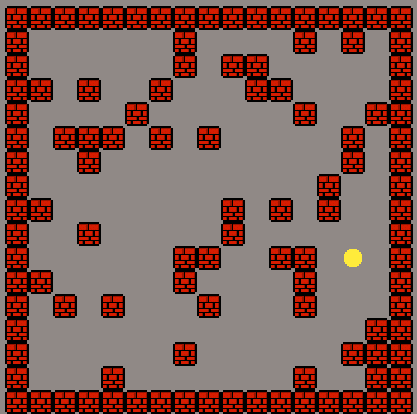
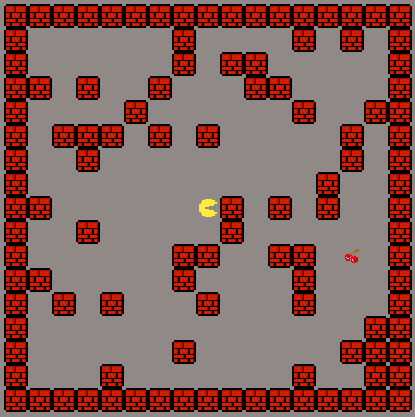
**Một số kết quả minh họa:**

**LEVEL1**



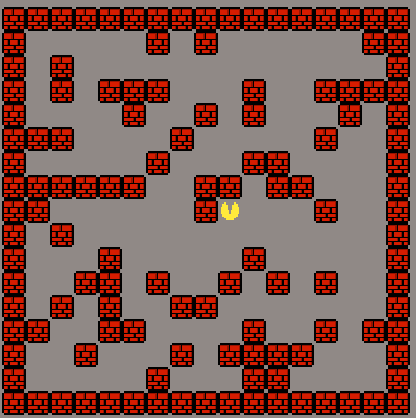
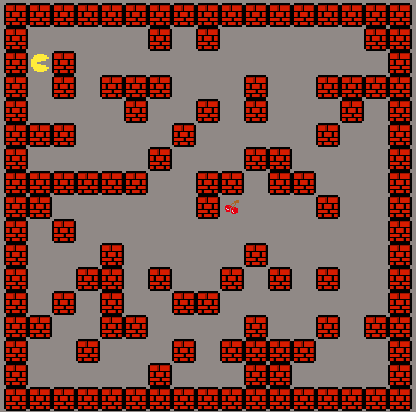
Độ dài đường đi: 26

Điểm: -6



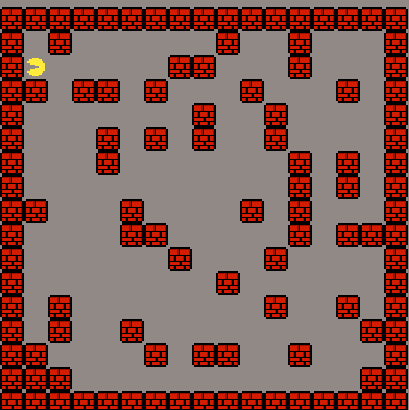
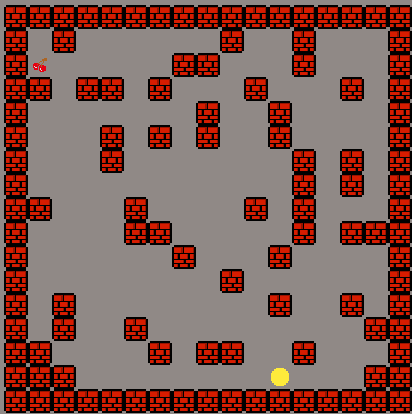
Độ dài đường đi: 11

Điểm: 9



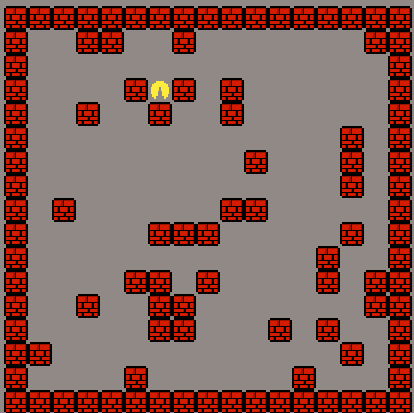
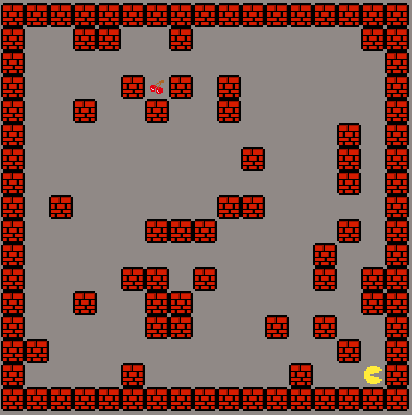
Độ dài đường đi: 23

Điểm: -3



Độ dài đường đi: 24

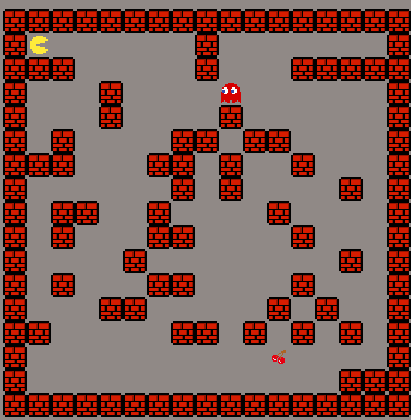
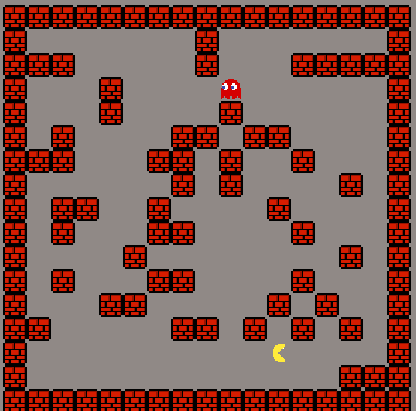
Điểm: -4



Độ dài đường đi: 24

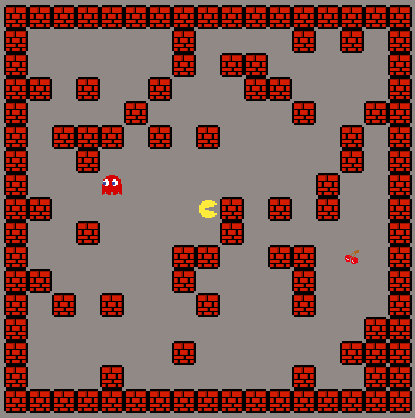
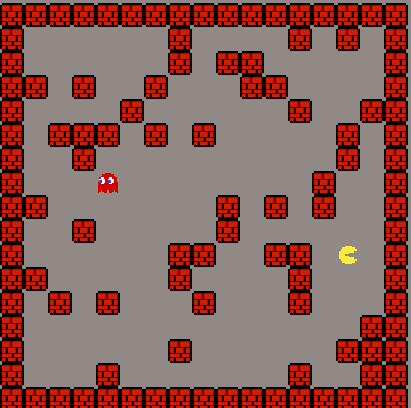
Điểm: -4

**LEVEL2**

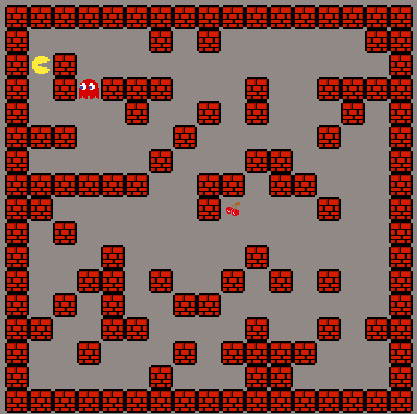
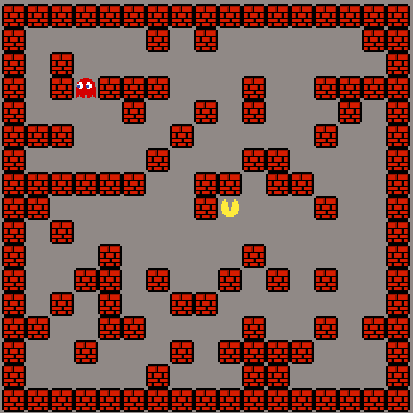
Độ dài đường đi: 26

Điểm: -6

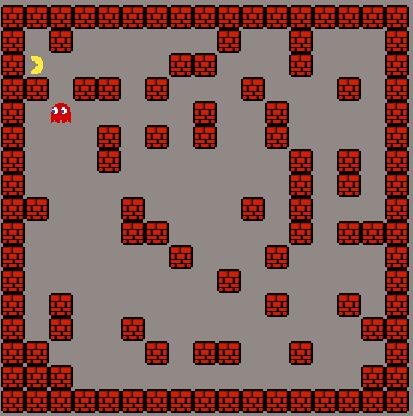
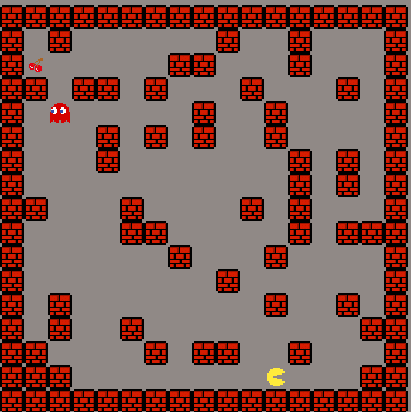
Độ dài đường đi: 11

Điểm: 9

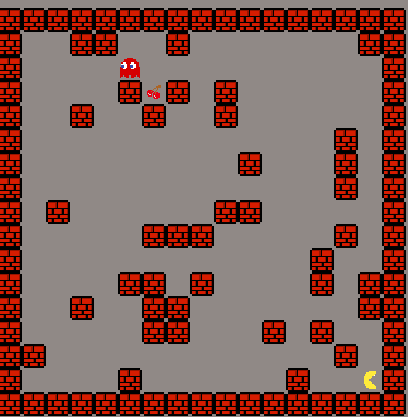
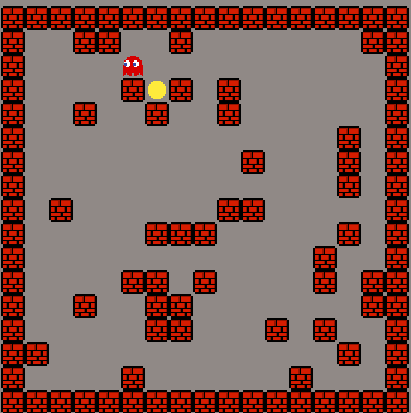
Độ dài đường đi: 23

Điểm: -3



Độ dài đường đi: 24

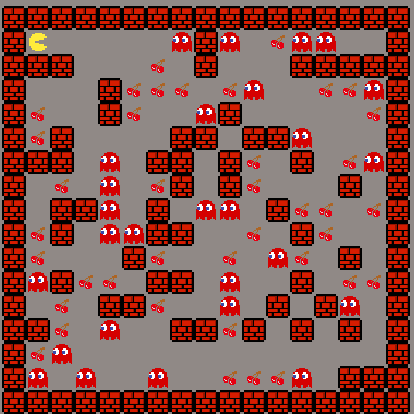
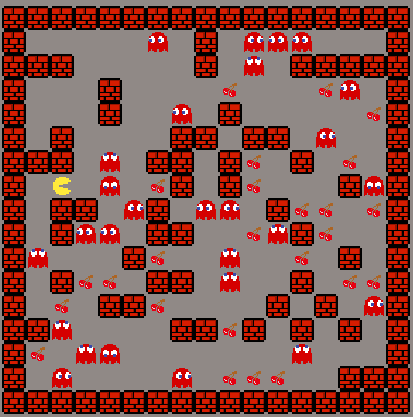
Điểm: -4

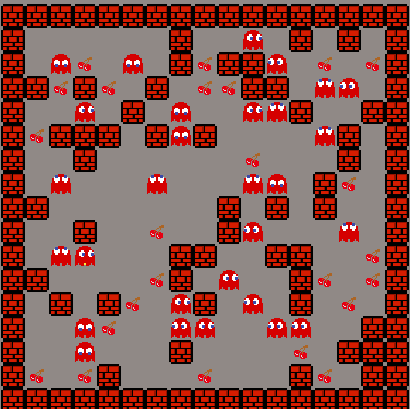
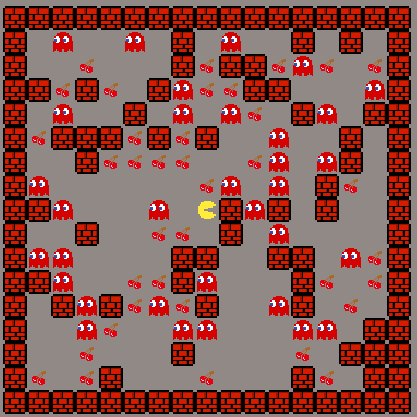
 

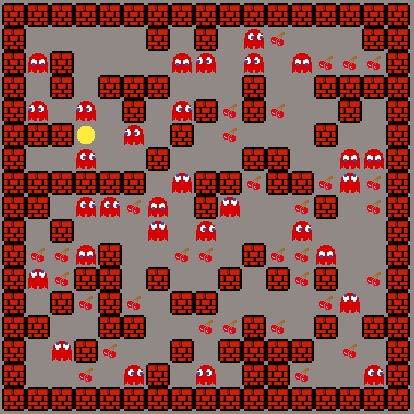
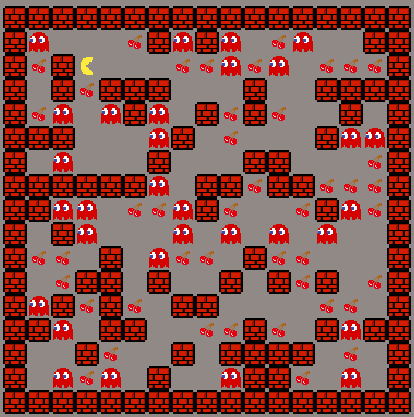
Độ dài đường đi: 24

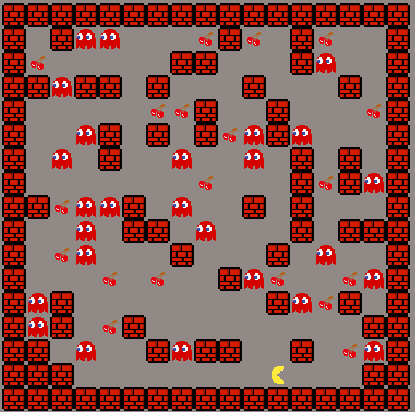
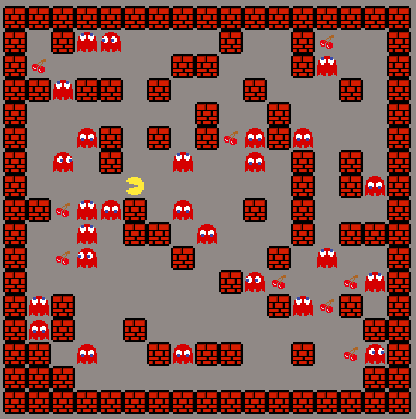
Điểm: -4

**LEVEL3**

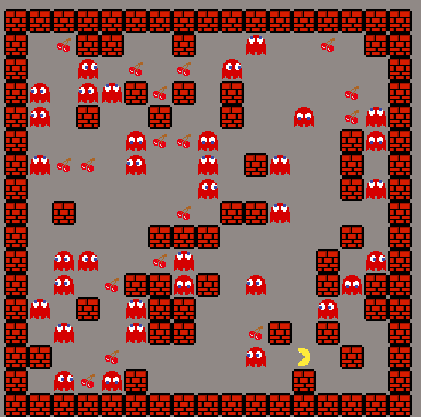
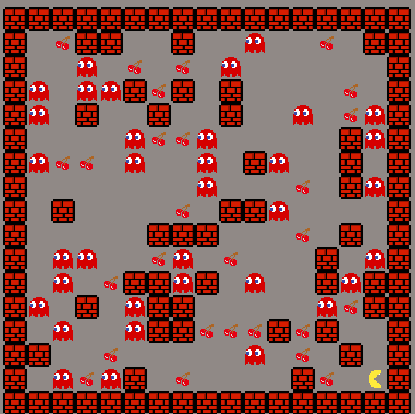
****Map 1. Tỉ lệ Thắng :Thua:Hòa = 0%: 53%:47%

Map 2. Tỉ lệ Thắng :Thua:Hòa = 0%: 18%:82%

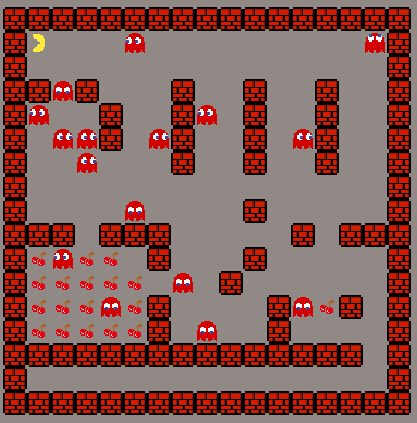
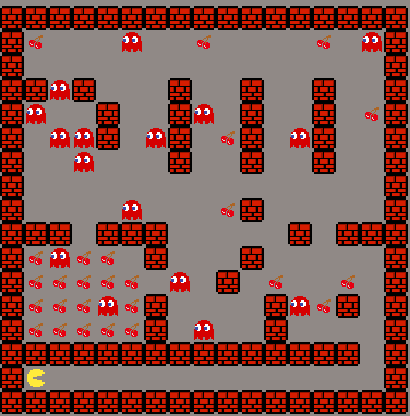
Map 3, level 3. Tỉ lệ Thắng :Thua:Hòa = 0%: 0%:100%

Map 4, level 3. Tỉ lệ Thắng :Thua:Hòa = 0%: 29%:71%

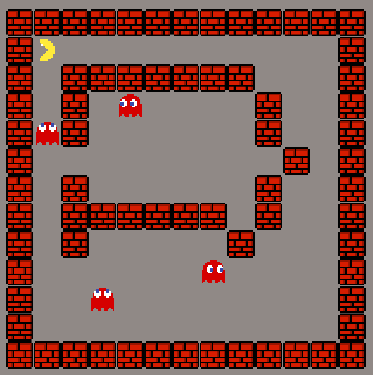
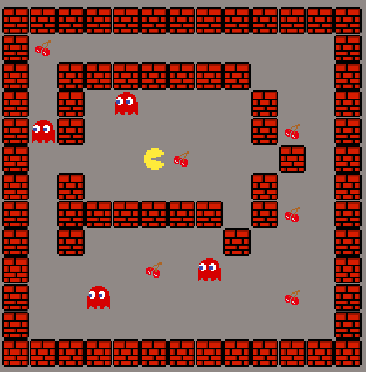
Map 5, level 3. Tỉ lệ Thắng :Thua:Hòa = 0%: 20%:80%



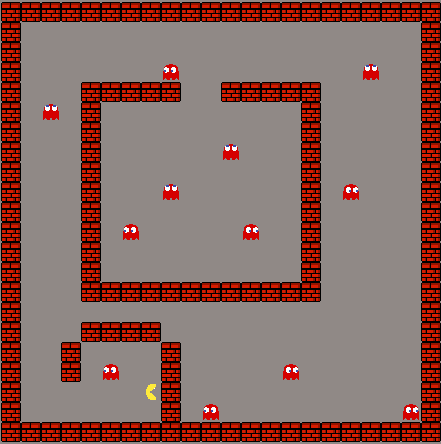
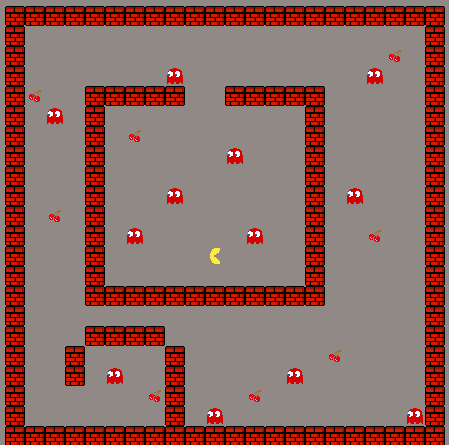
Map 6, level 3. Tỉ lệ Thắng :Thua:Hòa = 0%: 19%:81%



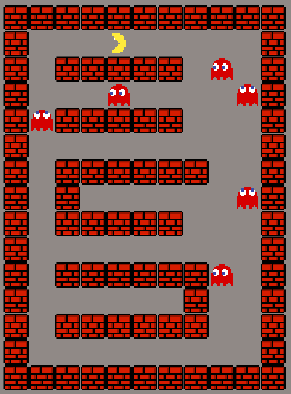
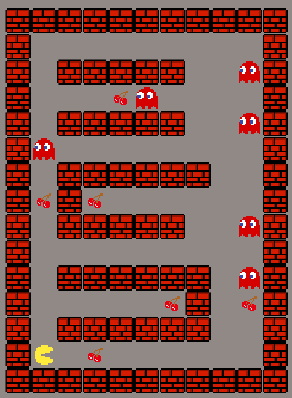
Map 7, level 3. Tỉ lệ Thắng :Thua:Hòa = 55%: 4%:41%



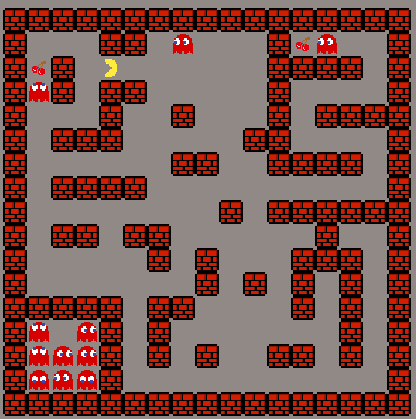
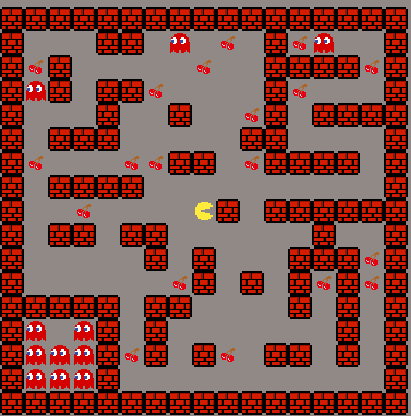
Map 8, level 3. Tỉ lệ thắng : thua: hòa = 95%:0%:5%



Map 9, level 3. Tỉ lệ Thắng :Thua:Hòa = 0%: 20%:80%

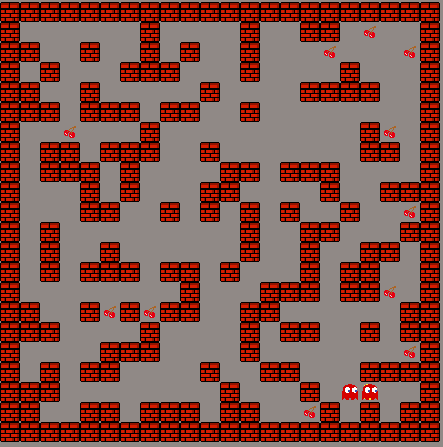
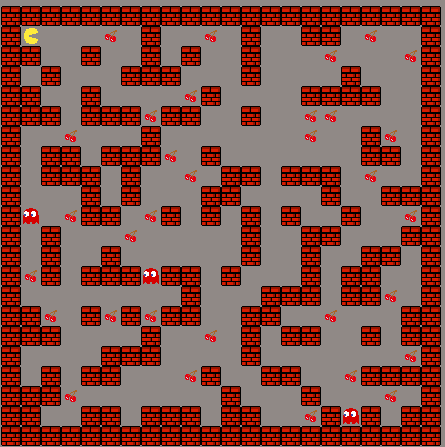


Map 10, level 3. Tỉ lệ Thắng :Thua:Hòa = 0%: 13%:87%



**LEVEL4:**

**Map 1:**



DEPTH:18

LOSE

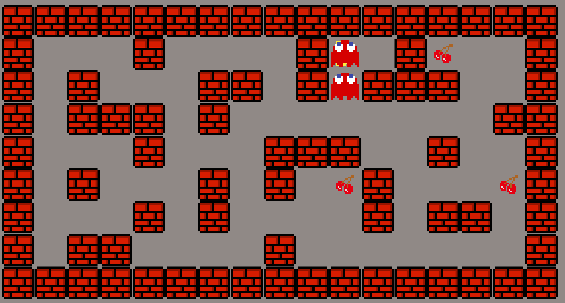
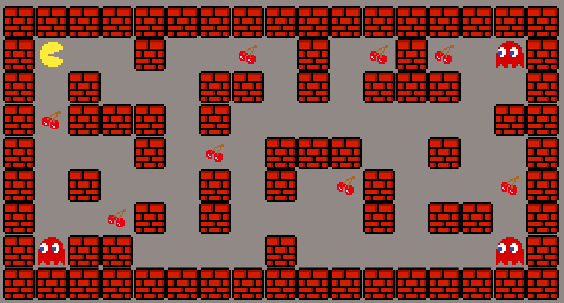
Time to finish: 767.2016191482544

Length of Pacman's discovered path: 110

Game score: 311

Foods left: 11

**Map2:**



DEPTH:20

LOSE

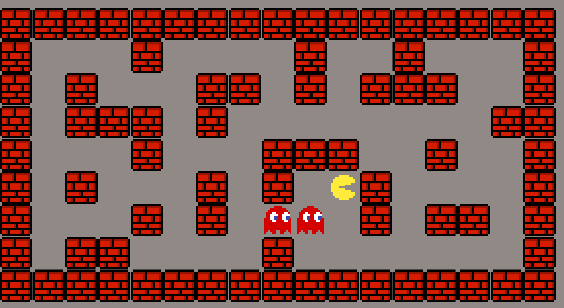
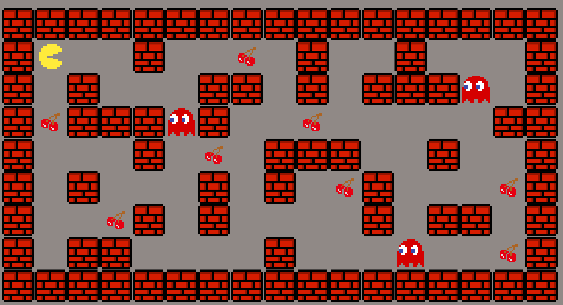
Time to finish: 33.879302978515625

Length of Pacman's discovered path: 40

Game score: 61

Foods left: 3

**Map3:**



DEPTH:30

WIN

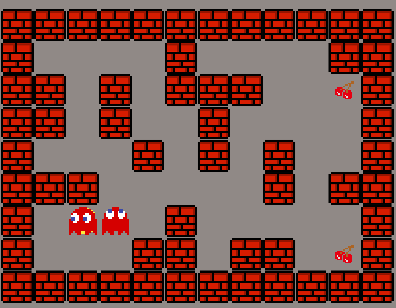
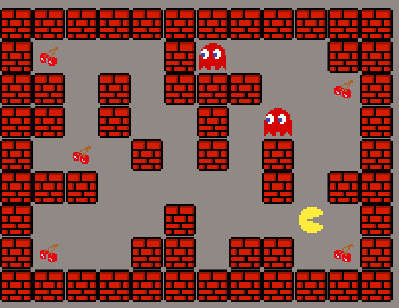
Time to finish: 112.98213267326355

Length of Pacman's discovered path: 76

Game score: 85

Foods left: 0

**Map4:**



DEPTH:30

LOSE

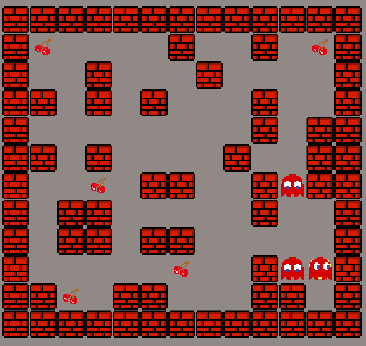
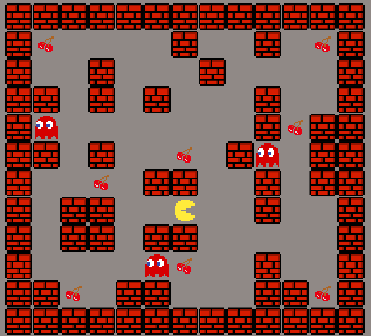
Time to finish: 12.274065017700195

Length of Pacman's discovered path: 32

Game score: 29

Foods left: 2

**Map5**:



DEPTH:20

LOSE

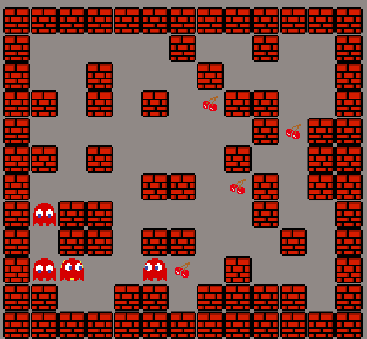
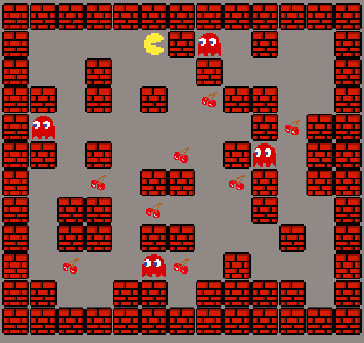
Time to finish: 127.28278160095215

Length of Pacman's discovered path: 41

Game score: 60

Foods left: 2

**Map 6:**



DEPTH:20

LOSE

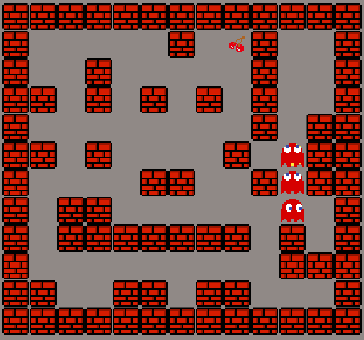
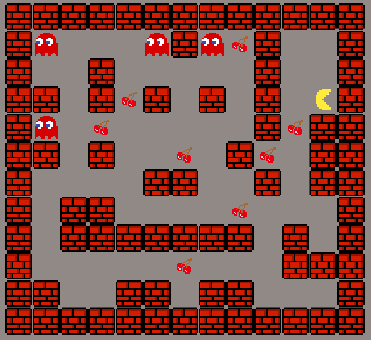
Time to finish: 309.71491622924805

Length of Pacman's discovered path: 22

Game score: 59

Foods left: 4

**Map7:**



DEPTH:25

LOSE

Time to finish: 2241.1086280345917

Length of Pacman's discovered path: 52

Game score: 89