

Introdução à Inteligência Artificial
Trabalho Prático nº1-Reactive Pacman
Meta Final

Grupo Constituido Por:
Bruno Miguel Oliveira Rolo (2010131200)
João Artur Ventura Valério Nobre(2010131516)
Daniel Luís Caetano Pereira 2006124813



Grupo/Tarefas

1.Constituição do Grupo:

Nome:Bruno Rolo

Número de Estudante: 2010131200

Email:brolo@student.dei.uc.pt

Turma Prática: PL5

Esforço Total para a realização deste trabalho: Aproximadamente 3 horas de estudo e 10 de implementação.

Tarefas Desempenhadas: Desenvolvimento no ficheiro IIAAgents.py, criação dos fantasmas e pacman,e testes sobre o desempenho de cada Agente, criação do relatório.

Nome:Daniel Pereira

Número de Estudante 2006124813

E-mail dlcp@student.dei.uc.pt

Turma Prática: PL5

Esforço:3 horas de estudo, 4 de implementação.

Tarefas Desempenhadas: Desenvolvimento no ficheiro IIAAgents.py, criação dos fantasmas e pacman,e testes sobre o desempenho de cada Agente,criação do relatório.

Nome:João Nobre

Número de Estudante:2010131516

E-mail janobre@student.dei.uc.pt

Turma Prática: PL5

Esforço Total para a realização deste trabalho:Aproximadamente 3 horas de estudo, 10 de implementação.

Tarefas Desempenhadas: Desenvolvimento no ficheiro IIAAgents.py, criação dos fantasmas e pacman,e testes sobre o desempenho de cada Agente, criação do relatório.



Introdução

Realizámos este com o objectivo de criar um agente reactivo adequado para o problema apresentado, neste caso, o conhecido jogo do Pacman. Com o objectivo de atribuir um comportamento de alto nível ao agentes do jogo, pacman e os respectivos fantasmas. Para resolução do problema criámos um sistema de produções que diz-nos como os agentes devem reagir em determinado ambiente, tendo estes mesmos agentes percepções, criámos sensores para que o agente em questão consiga desempenhar a tarefa pretendida. Ao longo deste documento iremos descrever a solução que desenvolvemos.



Modelização e Design

1.1 Percepções Comuns

Percepções Comuns a todos os Agentes Reactivos:

Segue-se em baixo uma tabela de percepções comuns a todos os agentes reativos á excepção do Agente Pacman que não tem sensores de si próprio.

Localização:	Comida	Pacman	Fantasma
Frente	CF	PF	FF
Esquerda	CE	PE	FE
Direita	CD	PD	FD

Anexo 01: Sensores de Localização de Comida,Capsulas,Paredes,Fantasma

EF -> Estado do Fantasma

SP-> Sensor vê o Pacman no campo de visão

FR = FF v FR v FE <=> Se avista algum fantasma nas redondezas

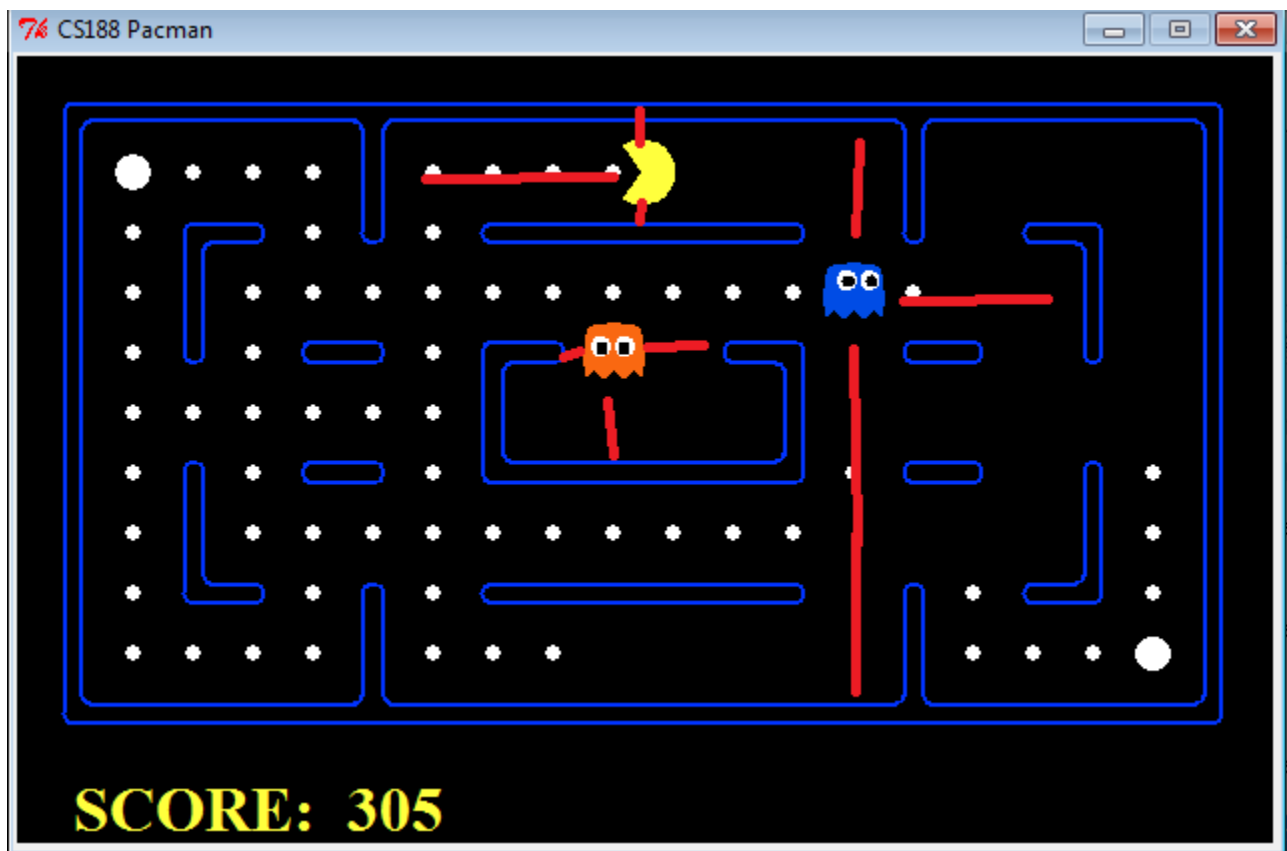


1.2 Sensores

Os agentes conseguem ver à sua frente, no seu lado esquerdo e direito.

A profundidade destes sensores é ilimitada até encontrar uma parede, as suas percepções são os caminhos das três coordenadas até encontrar uma parede.

Exemplo de profundidade:



Anexo 2: Exemplo de sensores de profundidade

As linhas vermelhas representam neste caso o alcance de cada Agente, neste caso, em particular o Pacman tem profundidade 0 no sensor esquerdo e direito, o fantasma laranja sem profundidade do seu lado direito e o azul com bom alcance.



1.3 Personalidade dos Agentes

1. Fantasma (CrazyStrategy):

Este fantasma irá ser agressivo, ou seja irá estar constantemente à procura do Pacman(movendo-se de forma aleatória), e se o encontrar irá atrás dele enquanto estiver no seu campo de visão(sensores).

2.Fantasma (O medroso):

Como o nome diz este fantasma é “medricas” , ou seja tentará antes de tentar encontrar o Pacman irá tentar encontrar um parceiro(um outro fantasma) para não estar sozinho , só irá procurar o Pacman quando tem um fantasma no seu campo de visão.

1.4 Comportamentos de alto nível esperados

O Pacman irá procurar comida no seu campo de visão, enquanto houver comida numa determinada direcção (frente,esquerda,direita), ele irá seguir essa direcção até encontrar uma parade,e assim sucessivamente.

Caso encontre um fantasma(sem este estar medricas*) numa direcção tentará encontrar uma direcção alternativa a seguir, lembrando que o pacman no seu campo visão só têm três direcções, frente, esquerda e direita.

Caso não encontre comida, (entenda-se comida como a comida que está presente ao longo do tabuleiro como as capsulas),o Pacman irá efectuar um movimento aleatório excluindo a direcção de onde veio, tentando assim evitar ciclos. É de salientar que o Pacman não diferencia comida de capsulas, para ele tanto faz.



Quando o pacman está em modo empowered,(comeu capsula grande), se ele encontrar um fantasma ele irá ignorá-lo(o Pacman dá prioridade á comida independentemente se pode comer fantasmas ou não)

Os Fantasmas(em que cada um tem a sua personalidade explicado anteriormente), tentarão cada um á sua maneira comer o pacman.

1.5 Memória

Na implementação que seguimos não foi necessário memória extra, isto é, para além utilizada nos sensores, como por exemplo, guardar a última posição dos agentes para não voltar a uma posição já visitada. Não utilizámos essa opção, orientamos a nossa implementação mais para os sensores, para que sejam capazes reagir como o previsto.



1.6. Sistema de Produções

Percepções:

SF : Sensor vê Fantasmas no campo de visão

SC : Sensor vê Comida no respectivo campo de visão

EP : Estado do Pacman (1 == empower / 0 == normal)

SCAP : Sensor de capsula no campo de visão

Accões:

MF: Mover para Frente;

MC: Mover para sentido contrário (N -> S; S -> N; E -> O; O -> E)

ME: Mover para a esquerda;

MR: Mover para a direita;

CM: Comer

1. PACMAN:

Cenário 1:Campo de visão sem fantasmas, sem comida mas com capsulas

$SCAP(F) \wedge !SF(F) \rightarrow MF$

$SCAP(E) \wedge !SF(E) \rightarrow ME$

$SCAP(R) \wedge !SF(R) \rightarrow MR$



Cenário 2:Campo de visão sem fantasmas,sem comida e sem capsulas

$!SP(F) \wedge !SP(E) \wedge !SP(R) \wedge !CF \wedge !COR \wedge !CE \wedge !SCAP(F) \wedge !SCAP(R) \wedge !SCAP(E) \rightarrow$
 $Random\{MF,ME,MO\}$

Nota: Se existir uma parede para um determinado movimento escolhido esse movimento é obviamente excluído sendo também excluído a ultima posição visitada a não ser que este seja o único movimento possível (dead-end)

Cenário 3:Encontra comida e sem capsulas

$CF \wedge !SF(F) \rightarrow MF$

$CE \wedge !SF(E) \rightarrow ME$

$CR \wedge !SF(R) \rightarrow MR$

Cenário 4:Pacman em estado “normal” tenta fugir dos fantasmas

$SF(F) \wedge !PF \wedge !EP \rightarrow MC(F)$

$SF(E) \wedge !PR \wedge !EP \rightarrow MC(E)$

$SF(R) \wedge !PE \wedge !EP \rightarrow MC(R)$

2.Fantasma:

Cenário 1:Vê o pacman em determinada direção

$SP(F) \wedge !PF \wedge EP \rightarrow MC(F)$

$SP(E) \wedge !PR \wedge EP \rightarrow MC(E)$

$SP(R) \wedge !PE \wedge EP \rightarrow MC(R)$

$SP(F) \wedge !EP \rightarrow MF$

$SP(E) \wedge !EP \rightarrow ME$

$SP(R) \wedge !EP \rightarrow MR$



Fantasma 1 “CrazyStrategy”:

$!SP(F) \wedge !SP(E) \wedge !SP(R) \rightarrow \text{Random}\{MF,ME,MR\}$

Nota: Se existir uma parede para um determinado movimento escolhido esse é excluído e irá ser gerado um novo movimento aleatório.

Nota2: A última posição visitada também é excluída dos movimentos possíveis, à exceção de este movimento ser o único movimento possível (dead-end)

Fantasma 2 “O medroso”:

$SF(F) \rightarrow MF$

$SF(E) \rightarrow ME$

$SF(R) \rightarrow MR$

$!SF(F) \wedge !SF(E) \wedge !SF(R) \rightarrow \text{Random}\{MF,ME,MR\}$

Nota: Se existir uma parede para um determinado movimento escolhido esse é excluído e irá ser gerado um novo movimento aleatório.

Nota2: A última posição visitada também é excluída dos movimentos possíveis, à exceção de este movimento ser o único movimento possível (dead-end)



2.1 Testes e Análises

Realizámos vários testes ao jogo que desenvolvemos, de destacar os três mapas, identificados a seguir, para percebermos o comportamento de alto nível dos agentes em diferentes ambientes.

Fantasma “CrazyStrategy”:

Mapa Medium classic:

Verificando os resultados podemos analisar que o pacman têm uma probabilidade baixa de ganhar. Como o pacman tem visão limitada do tabuleiro e estando os fantasmas em maior número a dificuldade aumenta, por exemplo, não existe um sensor do pacman que tenha visão para ver no sentido contrário à direcção em que ele segue, logo se um fantasma se encontrar atrás do pacman, o pacman não sente esse mesmo fantasma, o que implica que no caso do pacman não encontrar comida no seu campo de visão ele poderá ir na direcção do fantasma o que implica a perda do jogo se o fantasma tiver “encostado” a ele.

Original Classic:

Verificamos que a probabilidade de ganhar neste mapa é baixa, devido ao tamanho do mapa, sendo que o sensor do pacman se orienta pela comida, quando existe parte do mapa sem comida, anda numa direcção escolhida aleatoriamente, logo é mais difícil encontrar comida no seu campo de visão, como neste mapa é aumentado o número de fantasmas (quatro) a dificuldade aumenta.



Open classic:

Analisando os resultados, como o mapa é aberto, o pacman têm um campo de visão mais alargado e também só existe um fantasma em jogo, a probabilidade de ganhar é elevada (quase 100%). Existindo uma pequena hipótese de derrota, como por exemplo, sem ver comida reage aleatoriamente, como o campo de visão do pacman não alcança o sentido contrário, pode voltar para direção contrário onde se encontra um fantasma.

Foram testadas 15 partidas nos 3 seguintes mapas:

MediumClassic(mapa default),originalClassic e openClassic

Segue em anexo a tabela de probabilidades do pacman ganhar ao fantasma com a personalidade crazystrategy:

Jogos/Mapa:	MediumClassic	originalClassic	OpenClassic
1	Perdeu	Perdeu	Ganhou
2	Perdeu	Perdeu	Ganhou
3	Ganhou	Perdeu	Ganhou
4	Perdeu	Ganhou	Ganhou
5	Perdeu	Perdeu	Ganhou
6	Perdeu	Perdeu	Ganhou
7	Perdeu	Perdeu	Ganhou
8	Perdeu	Perdeu	Ganhou
9	Perdeu	Perdeu	Ganhou
10	Perdeu	Perdeu	Ganhou
11	Perdeu	Perdeu	Ganhou
12	Perdeu	Perdeu	Ganhou
13	Perdeu	Perdeu	Ganhou
14	Ganhou	Perdeu	Ganhou
15	Perdeu	Perdeu	Ganhou
% Wins:	0,13	6.6%	99.9%

Anexo 3- Win Rate Pacman versus “crazystrategy”



Fantasma “medricas”:

Testando os mesmos mapas testados para o fantasma anterior, verificamos que o pacman tem uma elevada probabilidade de ganhar. Sendo que o fantasma “medricas” está mais preocupado em ter companhia do que comer o pacman, e só procura o pacman se estiver acompanhado. Consequentemente o pacman sente-se será menos perseguido pelos fantasmas e seguirá comendo mais facilmente.

Jogos/Mapa: ▼	MediumClassic ▼	originalClassic ▼	OpenClassic ▼
1	Ganhou	Ganhou	Ganhou
2	Ganhou	Ganhou	Ganhou
3	Ganhou	Ganhou	Ganhou
4	Ganhou	Ganhou	Ganhou
5	Ganhou	Ganhou	Ganhou
6	Ganhou	Ganhou	Ganhou
7	Ganhou	Perdeu	Ganhou
8	Ganhou	Ganhou	Ganhou
9	Ganhou	Ganhou	Ganhou
10	Ganhou	Perdeu	Ganhou
11	Perdeu	Ganhou	Ganhou
12	Ganhou	Ganhou	Ganhou
13	Ganhou	Ganhou	Ganhou
14	Ganhou	Ganhou	Ganhou
15	Ganhou	Ganhou	Ganhou
% Wins:	93.3%	86.6%	100%

Anexo 4- Pacman versus “medricas”



3. Conclusão

Ao longo do desenvolvimento encontramos alguns problemas, como por exemplo, implementar no Pacman o sensor do fantasma que transmite o estado deste (com medo do pacman ou não). Para isso encontramos um método que nos retorna o número de segundos restantes do fantasma com medo, logo quando este valor for maior que zero o fantasma detectado está com medo logo o pacman pode ignorá-lo ou comê-lo se este estiver a interferir na sua missão. Como no caso do fantasma estar à sua frente e onde está a comida que ele quer comer.

Depois do programa estar implementado, após alguns testes vemos que existem algumas melhorias que poderiam ser efectuadas, como por exemplo, quando o pacman têm dois ou mais fantasmas na sua zona de visão, estando estes mesmos fantasmas em direcções diferentes, ele poderá não optar pela melhor opção, ele reage como só visse um fantasma, não separando os casos.

Como a profundidade do campo de visão do pacman é ilimitada até encontrar uma parede nessa direcção, se a distância entre o Pacman e um determinado fantasma detectado for longo ele imediatamente irá fugir dele o que não poderá ser a melhor opção por exemplo se a distância entre o Pacman e o fantasma for 4, e estiver nessa mesma direcção uma cápsula, uma boa reacção a este cenário seria comer essa cápsula e só depois fugir mas o que ele faz é fugir imediatamente.

Poderíamos pensar em diminuir o campo de visão, mas isso traria vantagens e desvantagens, mas optámos por seguir esta implementação descrita.

Outra melhoria seria quando o pacman não encontra comida, nem nenhum fantasma no seu campo de visão reage aleatoriamente, o que poderia ser evitado, Podendo entrar em pequenos ciclos.