**Desenho do Agente**

**Sistema Referencial**

Optámos por trabalhar segundo as direcções relativas FRONT, RIGHT, LEFT e BACK, ao invés das direcções absolutas NORTH, EAST, WEST e SOUTH. Esta decisão teve por base por duas razões essenciais:

* Primeiro, facilita em termos de descrição do agente ter a possibilidade de indicar e verificar posições relativamente à direcção pontual deste, como por exemplo, indicar que deve verificar primeiro se pode seguir em frente no seu percurso e não voltar para trás.
* Segundo, pensamos que a utilização de direcções relativas é mais coerente com a definição de um agente reactivo, que apenas tem uma visão local do ambiente e não uma percepção da sua posição em relação ao mundo.

**Capacidades dos Sensores:**

Cada sensor consegue percepcionar 5 elementos diferentes: Paredes, Fantasmas, Fantasmas Assustados, Comida e Cápsulas.

Relativamente à percepção de paredes, este apenas detecta paredes que inviabilizem a navegabilidade nessa direcção. Desta forma, se existir uma parede na casa imediatamente a seguir, sensor(Parede) assumirá o valor de True, caso contrário o valor Falso. Mesmo que o sensor consiga visualizar mais casas, este estado reflecte apenas a existência de paredes imediatamente a seguir à posição do agente na direcção do sensor.

Os restantes elementos – Fantasmas, Fantasmas Assustados, Comida ou Cápsulas – são activados a True na existência de um objecto desse tipo em qualquer casa pertencenta ao campo de visão do sensor.

**Descrição dos Sensores**

Inicialmente idealizámos um agente minimalista. Neste sentido, definimos um conjunto de sensores bastante limitados, no sentido de avaliar o desempenho do agente na sua forma mais elementar. Desta forma, definimos um sensor para cada direcção relativa (FRONT, LEFT, RIGHT e BACK) com a capacidade de visualizar apenas uma casa. A Figura 1 reflecte esta abordagem, mostrando os sensores, as sua amplitude e direcções.

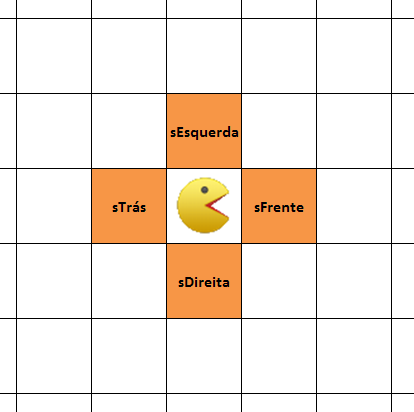


Figura - Sensores do Agente (versão 1)

**Descrição das Acções**

As acções já são definidas inicialmente pelo jogo e são inalteráveis. Estas correspondem a movimentos de uma casa na direcção Norte, Este, Oeste ou Sul. Como optámos por trabalhar segundo direcções relativas, as acção disponíveis são portanto, movimento de uma casa em Frente, Direita, Esquerda e para Trás, que correspondem no sistema de produções a MoveFrente, MoveDireita, MoveEsquerda e MoveTrás. Adicionalmente, existe também a acção STOP, que corresponde ao agente ficar parado durante uma jogada.

**Sistema de Produções**

Para construirmos o sistema de produções tivemos em consideração as seguintes permissas:

1. Nunca deslocar em direcção que contenha um Fantasma (não assustado) ou Parede;
2. Dar prioridade a Fantasmas Assustados, por representarem uma maquia elevada de pontos;
3. De seguida, dar prioridade a Comida ou Cápsula (como as Cápsulas se encontram normalmente em cantos ou corredores, ou não são detectadas em simultâneo com comida, ou se o são, é na mesma direcção. Desta forrma, não sentimos a necessidade de as diferenciar da Comida normal);
4. Caso não haja Comida, Capsula ou Fantasma Assustado, deslocar para uma direcção possível;
5. Em último recurso, parar!

Desta forma, criámos o seguinte sistema de produções:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| #Prioridade a Fantasmas Assustados | | | | | | | | |
| ¬­sFrente(P) | Λ | sFrente(FA) | Λ | ¬sFrente(F) | 🡪 | MoveFrente |  |  |
| ¬­sEsquerda(P) | Λ | ­sEsquerda(FA) | Λ | ¬­sEsquerda(F) | 🡪 | MoveEsquerda |  |  |
| ¬­sDireita(P) | Λ | sDireita(FA) | Λ | ¬­sDireita(F) | 🡪 | MoveDireita |  |  |
| ¬sTras(P) | Λ | sTras(FA) | Λ | ¬sTras(F) | 🡪 | MoveTras |  |  |
|  |  |  |  |  |  |  |  |  |
| #Prioridade a Comida ou Cápsula | | | | | | | | |
| ¬­sFrente(P) | Λ | ¬sFrente(F) | Λ | ( sFrente(C) | V | sFrente(Cap) ) | 🡪 | MoveFrente |
| ¬­sEsquerda(P) | Λ | ¬­sEsquerda(F) | Λ | ( sEsquerda(C) | V | sEsquerda(Cap) ) | 🡪 | MoveEsquerda |
| ¬­sDireita(P) | Λ | ¬­sDireita(F) | Λ | ( sDireita(C) | V | sDireita(Cap) ) | 🡪 | MoveDireita |
|  |  |  |  |  |  |  |  |  |
| #Segue direcção possível | | | | | | | | |
| ¬­sFrente(P) | Λ | ¬sFrente(F) | 🡪 | MoveFrente |  |  |  |  |
| ¬­sEsquerda(P) | Λ | ¬­sEsquerda(F) | 🡪 | MoveEsquerda |  |  |  |  |
| ¬­sDireita(P) | Λ | ¬­sDireita(F) | 🡪 | MoveDireita |  |  |  |  |
| ¬sTras(P) | Λ | ¬ sTras (F) | 🡪 | MoveTras |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| #Pára | | | | | | | | |
| True | 🡪 | STOP |  |  |  |  |  |  |

**Comportamento**

O facto de termos criado um sistema tão minimalista levou-nos a alguns problemas e a estatísticas de resultados bastante baixos (ver Anexo I).

O primeiro problema detectado que nos levou à primeira alteração dos sensores encontra-se descrito na Figura 2. Neste caso, se tivéssemos um Fantasmas a 2 casas do nosso agente, este não seria detectado pelos sensores. Desta forma, se ambos se deslocassem na mesma direcção e sentidos opostos, o agente e o Fantasma acabam na mesma casa, levando à derrota no jogo.

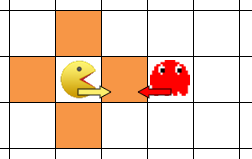


Figura - Problema associado a alcance dos sensores

Com o intuito de resolver esta situação, procedemos ao aumento do campo de visão de cada sensor para duas casas. A Figura 3 esquematiza esta nova abordagem.

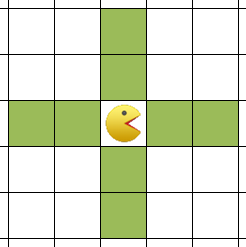
****

Figura – Sensores do Agente (versão 2)

Esta alteração levou a melhorias, embora não muito significativas (ver Anexo I). Detectámos então um novo problema, que mais uma vez nos levou a reavaliar o nosso esquema de sensores. Se numa esquina se encontrar um Fantasma a dirigir-se ao encontro do nosso agente, este não o consegue detectar, gerando colisão e, consecutivamente, derrota do nosso agente. Este problema encontra-se descrito visualmente na Figura 4.

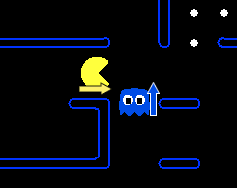
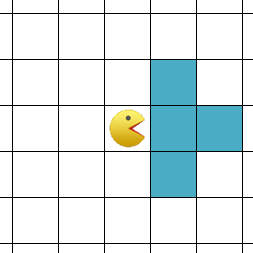
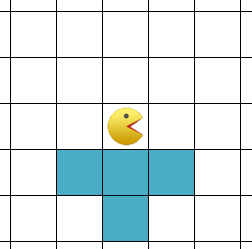
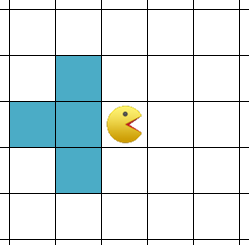
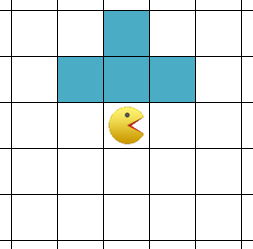


Figura - Colisão em Esquina

Este problema levou-nos novamente a uma alteração à visão dos sensores. Decidimos manter os mesmos sensores, alterar apenas o que cada sensor consegue ver. A Figura 5 representa a nova forma como os sensores percepcionam o ambiente.

Sensor de frente Sensor da direita Sensor de trás Sensor da esquerda

Figura - Sensores do Agente (versão 3)

É preciso salientar que agora os sensores não diferenciam entre um elemento na casa imediatamente à frente e um elemento na casa frente-esquerda (por exemplo). São tudo resultados do mesmo sensor (à excepção do elemento Parede que, como foi explicado acima, apenas é detectado se estiver na casa imediatamente a seguir ao agente). Esta indiferenciação entre as casas da visão do sensor permitiu-nos manter o sistema de produções, que se manteve inalterado desde o início até esta versão.

Verifica-se que as casas dos “cantos” são “vistas” sempre por 2 sensores. Assim, a casa Frente-Esquerda é percepcionada pelos sensores sFrente e sEsquerda. Desta forma, se um Fantasma se encontrar nesta casa, ambas as direções Frente e Esquerda são bloqueadas. Como mostra a Figura 6, ambas as direcções podiam levar à morte do agente.

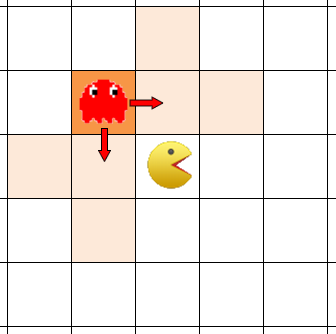


Figura - Ataque pelo Canto

Os resultados foram significativamente melhores, como se pode constactar no Anexo I. No entanto, detectámos um problema crucial no nosso sistema de produções: podia gerar ciclos! (ver Figura 7).

O nosso sistema de produções, na ausência de elementos (exceptuando as Paredes), segue sempre para a frente, se impedido para a direita, se impedido para a esquerda, se impedido para trás. Desta forma, o nosso agente podia entrar em ciclos, perdendo tempo e consecutivamente pontos, sendo apenas liberto do ciclo pelo aparecimento de um Fantasma.

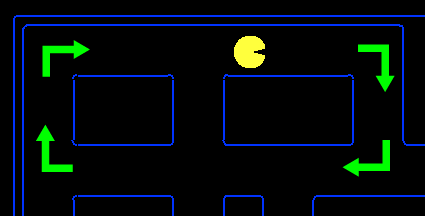


Figura - Ciclo produzido pelo agente

Isto levou-nos à única alteração no nosso sistema de produções. Decidimos que a melhor maneira para resolver este problema seria uma escolha aleatória da direcção a seguir (entre frente, esquerda e direita) quando não há qualquer elemento captado pelos sensores (à excepção de Paredes). Para tal, inserimos um sensor aleatório sAleat que assume valores de 1 a 3. Desta forma, conseguimos uma escolha aleatória de uma direcção. Esta direcção pode ser inválida (haver uma Parede ou Fantasma), no entanto, mais cedo ou mais tarde esta decisão é válida e permite-nos evitar os ciclos.

Uma particularidade do comportamento resultante do agente ocorre quando se dirige paralelamente a uma linha de comida. Neste caso, o agente atravessa primeiro toda a linha de comida, e só depois se dirige realmente a esta. As figuras seguintes exemplificam este comportamento:

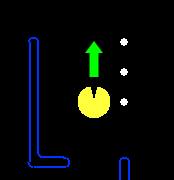
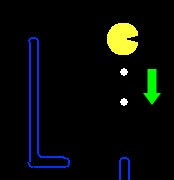
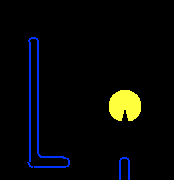
  

Figura – Comportamento do agente perante uma fila de comida

Na grande parte dos casos, este comportamento não leva a um pior desempenho, porque o agente acaba por comer de qualquer forma todos os elementos. Mas o adiar dessa acção pode levar ao surgimento de um fantasma que afaste o agente da comida.

O agente só perderá o jogo em duas situações diferentes em que pode ficar encurralado. Poderá encontrar-se num corredor sem saída e encontrar-se um fantasma a movimentar-se na sua direcção, ou pode ser encurralado por 2 ou mais fantasmas num corredor ou espaço aberto. As figuras seguintes exemplificam estas situações:

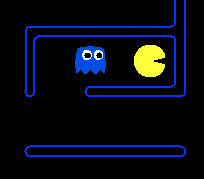
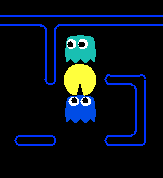
 

Figura – Situações em que o agente se encontra encurralado

**VERSÃO FINAL**

**Sensores**: sFrente, sEsquerda, sDireita, sTras, sAleat

**Valores dos Sensores:** P – Parede, FA – FantasmaAssustado, F – Fantasma, C – Comida e Cap – Capsula

**Acções**: MoveFrente, MoveEsquerda, MoveDireita, MoveTras, STOP

**Sistema de Produções Final**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| #Prioridade a Fantasmas Assustados | | | | | | | | |
| ¬­sFrente(P) | Λ | sFrente(FA) | Λ | ¬sFrente(F) | 🡪 | MoveFrente |  |  |
| ¬­sEsquerda(P) | Λ | ­sEsquerda(FA) | Λ | ¬­sEsquerda(F) | 🡪 | MoveEsquerda |  |  |
| ¬­sDireita(P) | Λ | sDireita(FA) | Λ | ¬­sDireita(F) | 🡪 | MoveDireita |  |  |
| ¬sTras(P) | Λ | sTras(FA) | Λ | ¬sTras(F) | 🡪 | MoveTras |  |  |
|  |  |  |  |  |  |  |  |  |
| #Prioridade a Comida ou Cápsula | | | | | | | | |
| ¬­sFrente(P) | Λ | ¬sFrente(F) | Λ | ( sFrente(C) | V | sFrente(Cap) ) | 🡪 | MoveFrente |
| ¬­sEsquerda(P) | Λ | ¬­sEsquerda(F) | Λ | ( sEsquerda(C) | V | sEsquerda(Cap) ) | 🡪 | MoveEsquerda |
| ¬­sDireita(P) | Λ | ¬­sDireita(F) | Λ | ( sDireita(C) | V | sDireita(Cap) ) | 🡪 | MoveDireita |
|  |  |  |  |  |  |  |  |  |
| #Escolhe direcção aleatória | | | | | | | | |
| sAleat = 1 | Λ | ¬­sFrente(P) | Λ | ¬­sFrente(F) | 🡪 | MoveFrente |  |  |
| sAleat = 2 | Λ | ¬­sEsquerda(P) | Λ | ¬­sEsquerda(F) | 🡪 | MoveEsquerda |  |  |
| sAleat = 3 | Λ | ¬­sDireita(P) | Λ | ¬­sDireita(F) | 🡪 | MoveDireita |  |  |
|  | | | | | | | | |
| #Segue direcção possível | | | | | | | | |
| ¬­sFrente(P) | Λ | ¬sFrente(F) | 🡪 | MoveFrente |  |  |  |  |
| ¬­sEsquerda(P) | Λ | ¬­sEsquerda(F) | 🡪 | MoveEsquerda |  |  |  |  |
| ¬­sDireita(P) | Λ | ¬­sDireita(F) | 🡪 | MoveDireita |  |  |  |  |
| ¬sTras(P) | Λ | ¬ sTras (F) | 🡪 | MoveTras |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| #Pára | | | | | | | | |
| True | 🡪 | STOP |  |  |  |  |  |  |