

Implementação de Um Jogo Com Inteligência Artificial e Aprendizado Probabilístico



Rogério Cazelato Papetti

Supervisor: Marcelo Finger, Co-Supervisor: Flávio Soares Corrêa

Departamento de Ciência da Computação IME-USP
MAC-499 Trabalho De Formatura Supervisionado



1. Introdução

O desenvolvimento de jogos representa uma área em crescimento na computação. É um catalisador do desenvolvimento de técnicas de inteligência artificial e computação gráfica.

Entretanto, é um pouco obscuro, mesmo para um programador, como é o processo de desenvolvimento do jogo e de um algoritmo de inteligência artificial para o mesmo.

Esse projeto tem como objetivo desenvolver e implementar um algoritmo próprio de inteligência artificial e aprendizado e aplicá-lo num jogo, para testá-lo de forma prática.

1.1 O jogo PACMAN

Pacman é um dos jogos mais famosos em toda a história dos games. Foi criado pelo japonês Toru Iwatani, na década de 1980. O jogo consiste na tentativa da personagem Pacman coletar todas as pac-dots (bolinhas) espalhadas pelo cenário sem ser capturado por um dos fantasmas distribuídos pelo labirinto.

1.2 Motivação

Esse jogo nos abre uma possibilidade de implementação de inteligência artificial, já que os fantasmas são (na versão original) movimentados por um algoritmo determinístico. Ou seja, uma sequência de movimentos vencedora será sempre uma estratégia vencedora.

2. Ferramentas Utilizadas

Foram utilizadas na concepção do jogo duas ferramentas: O Maya (para modelagem 3D) e o Panda-3D (como framework para desenvolvimento da aplicação).

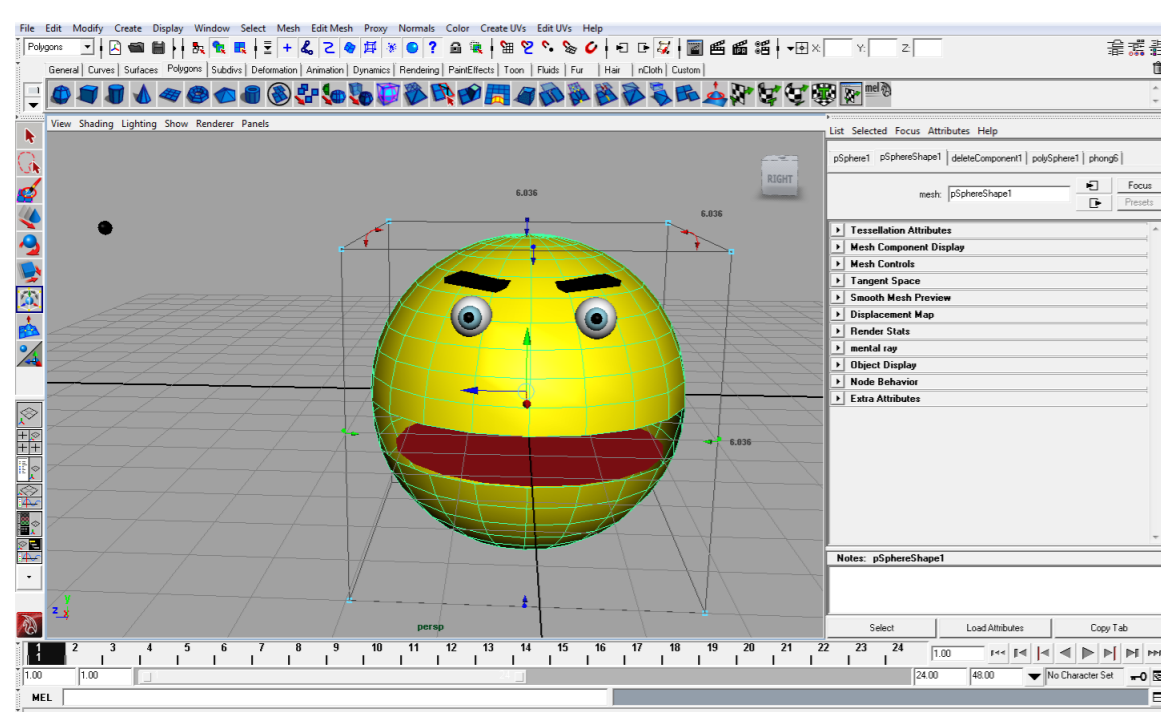


Figura 1: Modelagem do Pacman feita no Maya

Com o Maya foram modelados o mapa (por meio do Maya é possível, inclusive, indicar onde existem paredes ou outros objetos de colisão e até mesmo deixar indicados pontos chave do mapa, como o local onde as personagens devem iniciar, passagens secretas, etc) e as personagens (pacman e fantasmas).

Já o Panda-3D foi o framework base para todos os aspectos de implementação do jogo, como colisões, movimentação dos personagens, execução de algoritmos de Inteligência Artificial a cada frame, renderização de imagens, controle de tempo de jogo.

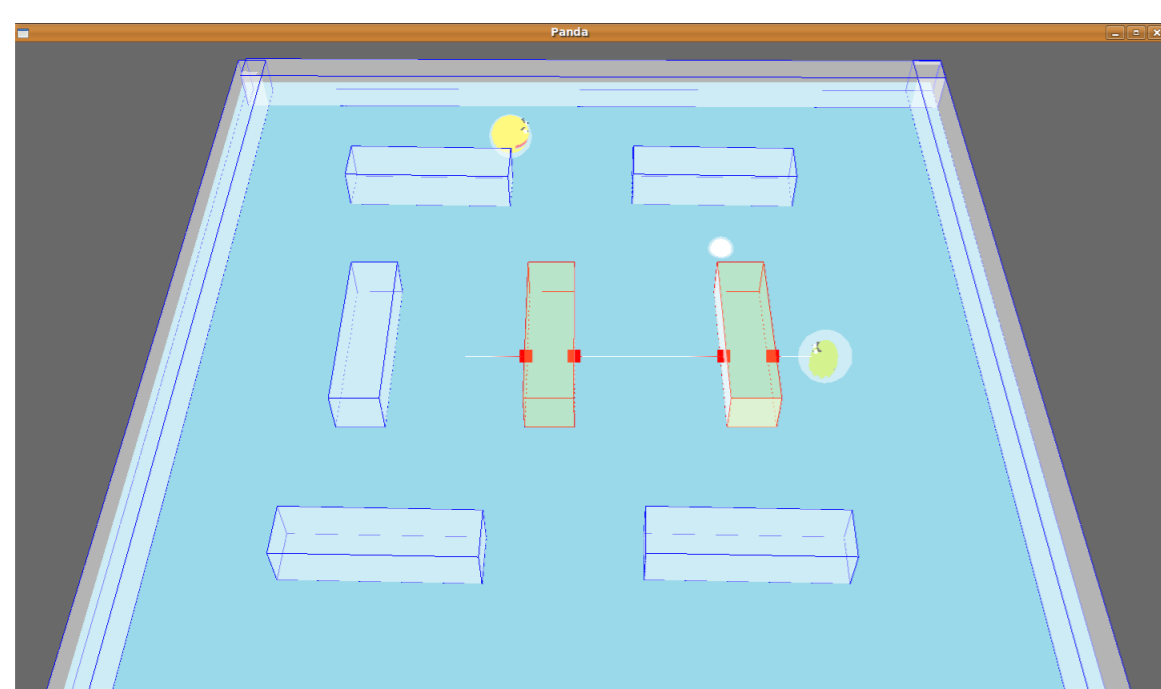


Figura 2: Sistema de colisões do Panda-3D funcionando

3. Os Algoritmos

3.1 Raios de Visão Frontal e Periférica

O Raio de Visão do fantasma é um segmento de reta que sai da altura do olho do mesmo e pode interceptar qualquer objeto do mapa. A cada frame, um raio de visão é lançado e um gerenciador de colisões associado ao raio de visão ordena as colisões (da mais perto do fantasma para a mais distante) e devolve a colisão mais próxima. Esse raio frontal também permite ao fantasma medir a distância do mesmo para uma parede à sua frente, evitando uma colisão com a mesma.

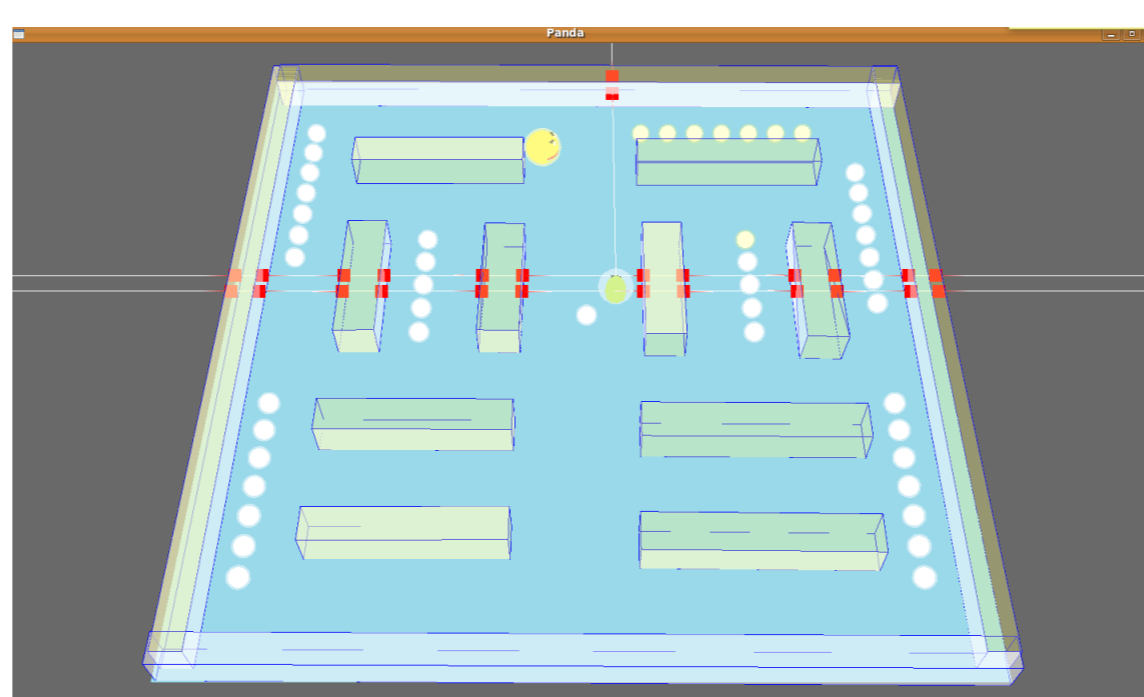


Figura 3: Raios laterais e frontal

O fantasma também lança raios para detectar qual distância está de uma parede tanto à sua esquerda como à sua direita (caso essa distância seja muito pequena, o fantasma não se move para os lados, evitando colidir com a parede).

A medida da distância atual com a parede também é comparada com a medida obtida no frame anterior. Caso essa distância tenha aumentado, há uma esquina no mapa e o fantasma pode escolher virar para aquele lado. Caso a distância tenha diminuído, a entrada não existe mais e o fantasma não tentará movimentar-se para os lados.

3.2 Sorteio

Após o fantasma identificar, por meio do raio de visão, qual objeto está mais próximo ou colidindo com ele, o mesmo deve decidir o que fazer a partir da informação coletada. Essa escolha é feita por meio de probabilidades. Para cada objeto próximo, existem algumas ações possíveis e, para cada, uma probabilidade associada.

Por exemplo, ao identificar que está sobre uma 'pac-Dot', o fantasma pode permanecer sobre a mesma, sair para a frente, para trás ou para os lados (caso haja uma entrada disponível). Como uma probabilidade é associada a uma ação e como ela é modificada está explicado na seção aprendizado.

3.3 Movimentação do Fantasma

Variáveis:

- fantasma: instância da Classe Fantasma
- raioDeVisao: raio de visão frontal
- colHandler: gerenciador de colisão associado ao raio de visão frontal

```
1: repeat
2:   lança raioDeVisao
3:   objetoMaisPerto ← colHandler.devolveMaisPerto()
4:   if fantasma.estaSobrePacDot() = TRUE then
5:     sorteia ação e executa
6:   else
7:     if fantasma.estaPertoDeParedeFrontal() = TRUE
8:       then
9:         if fantasma.existeEntradaLateral() = TRUE
10:        then
11:          Vira na entrada Lateral
12:        else
13:          Vira 180 graus e volta
14:        end if
15:      else
16:        Sorteia ação baseado no objeto que está mais próximo e executa
17:      end if
18:    end if
19:  until jogo terminar
```

3.4 Aprendizado

Cada ação tem associada a si uma probabilidade de ser executada. A cada fim de jogo, todas as jogadas executadas, que foram armazenadas, são analisadas e recebem uma pontuação, que depende da vitória do fantasma ou não e do tempo que o jogo durou. Quanto mais longo o jogo, mais pontos o fantasma ganha. Se derrotou o pacman, ganha um bônus.

Foram testadas duas situações: probabilidades sem viés (ou seja, todas as ações foram inicializadas com chances iguais) e com viés (ou seja, o fantasma já sabia que algumas ações eram melhores que outras).

4. Resultados

O algoritmo de movimentação foi implementado com sucesso. O fantasma consegue evitar paredes, identificar entradas e esquinas e se movimentar com razoável destreza pelo cenário, principalmente se levarmos em conta que ele não tem nenhuma ajuda externa ou marcação do mapa.

Quanto ao aprendizado, foram feitas simulações com 2 fantasmas e com probabilidades enviesadas e não enviesadas. Sem viés de probabilidades, o fantasma não venceu em nenhum dos 15 jogos testados e não houve muitas mudanças nos valores das probabilidades (nenhum valor foi alterado em mais de 10%).

Com viés, houve 5 vitórias dos fantasmas em 15 jogos testados e houve um considerável aumento nos valores de probabilidades de jogadas cruciais (como avançar para uma 'pac-dot'): cerca de 20%.

5. Conclusão

Baseado nos objetivos descritos, pode-se concluir que o desenvolvimento do algoritmo de movimentação e de aprendizado atingiu os objetivos esperados. O fantasma se movimenta pelo cenário identificando entradas, paredes, objetos sem precisar de marcações no mapa. Entretanto, ainda pode ser melhorado para ficar mais próximo em termos de jogabilidade do algoritmo determinístico aplicado normalmente.

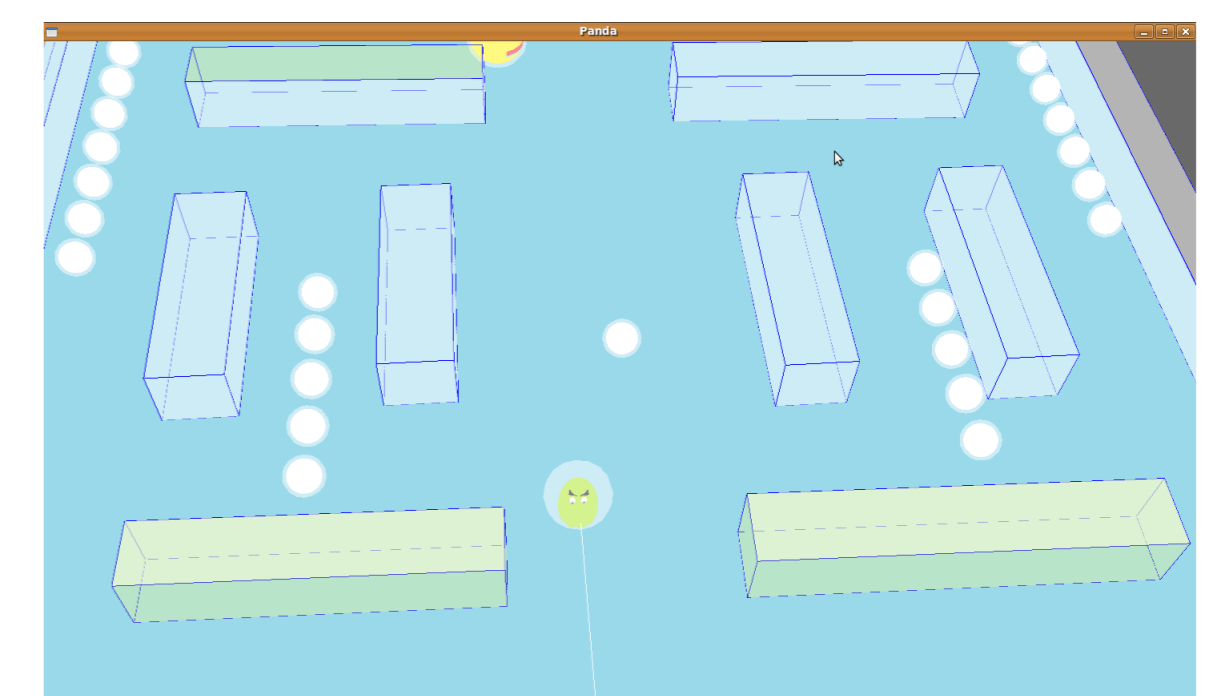


Figura 4: Fantasma se movendo pelo cenário

Quanto ao aprendizado, sem nenhum viés nas probabilidades, a chance de escolher uma jogada errada ou de escolher jogadas completamente opostas em dois frames consecutivos é muito alta. Assim, os fantasmas se comportam de forma muito errante e permitem ao pacman vitórias muito fáceis.

Já com as probabilidades enviesadas, o fantasma age mais vezes corretamente, sobrevive por mais tempo e, assim, aprende mais rápido. Entretanto, ainda são necessárias melhorias nos algoritmos para que os mesmos sejam competitivos contra um humano.

Referências

[1] RUSSELL, Stuart ; NORVIG, Peter - "Artificial Intelligence: A Modern Approach" -Prentice Hall Series in Artificial Intelligence, 1995 .

[2] Site Panda-3D: www.panda3d.org

[3] Site Maya: www.everything4maya.com