

Relatório de Computação Gráfica - Pac-man 3D

Douglas Venâncio - 28164
Engenharia de Computação
Universidade Federal de Itajubá
campus Itabira
dougsov@gmail.com

Filipe Euzebio Borges Pena - 30213
Engenharia de Computação
Universidade Federal de Itajubá
campus Itabira
epilif2011@live.com

Marco Antonio Rocha Vinha - 30749
Engenharia de Computação
Universidade Federal de Itajubá
campus Itabira
marvsdd.01@live.com

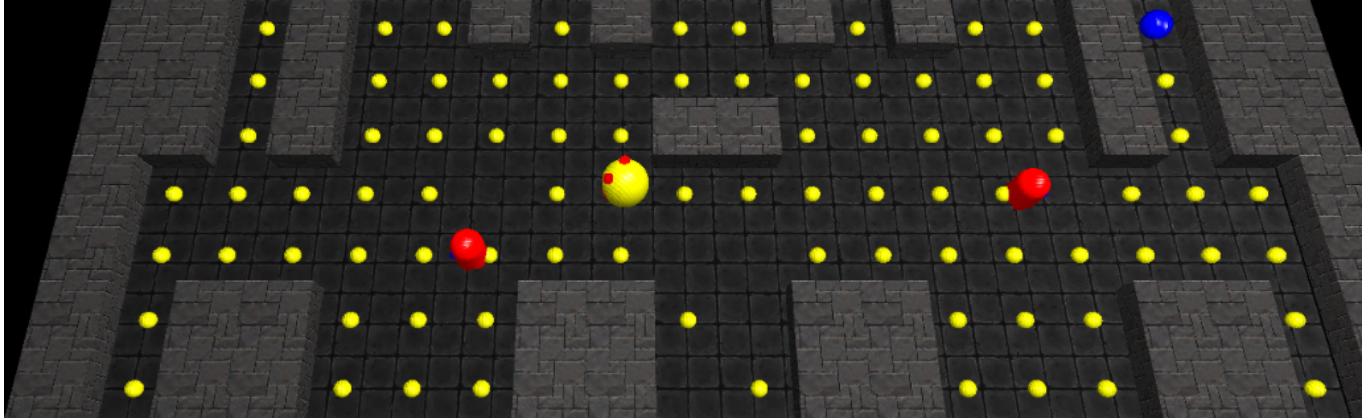


Figura 1. Amostra dos resultados obtidos com o desenvolvimento do projeto: nesse estágio, já foram aplicados conceitos de iluminação, colorização, textura e câmera com projeção perspectiva.

Abstract—The Pac-man game is a classic of video games, released in 1980 by Atari for its console at the time. The game well known, as well as his mechanics and general gameplay, with a few changes over the decades and with different versions of the game. In this paper, it's shown the process of development of a version of that game. The proposal is to create a 3D version of the game for the study and implementation of concepts in the field of Computer Graphics, as to be known: modeling process, colouring process, lightning process and some mechanics related to this processes.

Keywords-Computer Graphics; Pac-man; lightning process; colouring process; Open-GL.

Resumo—O jogo Pac-man é um clássico dos videogames, lançado em 1980 pela Atari para seu console na época. O jogo é bem conhecido, assim como sua mecânica e jogabilidade geral, com algumas mudanças ao longo das décadas e com diferentes versões do jogo. Neste trabalho, é mostrado o processo de desenvolvimento de uma versão desse jogo. A proposta é criar uma versão 3D do jogo para o estudo e implementação de conceitos no campo da Computação Gráfica, como é de se conhecer: processo de modelagem, processo de coloração, processo de iluminação e algumas mecânicas relacionadas a eles.

Keywords-Computação Gráfica; Pac-man; processo de iluminação; processo de coloração; Open-GL.

I. INTRODUÇÃO

O presente relatório é respectivo ao projeto desenvolvido como forma de se aplicar conceitos relacionados a princípios

da computação gráfica, como transformações geométricas, transformações de projeção, iluminação de cenas e texturização. O projeto se trata de um jogo baseado no clássico *Pac-man*, e foi desenvolvido usando as bibliotecas do *open-GL* para C++.

A. Transformações Geométricas e Projetivas

Um dos tópicos fundamentais em computação gráfica, é o que diz respeito às transformações geométricas e projetivas. Para que, em uma cena, se possa manipular de forma variada os objetos que a compõem, existem uma série de ferramentas derivadas da geometria analítica que, quando aplicadas ao contexto da construção de cenas tri-dimensionais, são amplamente utilizadas nas transformações geométricas, às quais possam estar sujeitos os objetos na cena, e nas transformações projetivas, que dizem respeito à projeção dessa cena para o observador. Na aplicação desenvolvida, foram usadas apenas as primitivas disponíveis no *open-GL*, como triângulos, quadrados e polígonos no geral, sendo necessário o uso de ferramentas mais complexas para se desenhar formas mais complexas - como a junção de polígonos ou conceitos de coordenadas esféricas e cilíndricas e polares para se desenhar formas como cilindros ou esferas. Para o jogo desenvolvido, com o objetivo de se manipular os elementos gráficos na cena, são aplicados os conceitos de transformações geométricas, principalmente translação e rotação. Da forma como são utilizadas, essas transformações são realizadas por operações matriciais ou

vetoriais aplicadas em pontos de interesse, ou seja, nos pontos (ou objetos) que, na cena, possuem mobilidade.

Ainda, foram aplicadas conceitos de transformação projetiva, mais especificamente de modelos de câmera, para que, de certa forma, os recursos de um jogo tri-dimensional fossem melhor aproveitados. Dessa forma, o jogo dá a possibilidade de que se alterem projeções, de forma que o jogador possa visualizar ângulos diferentes da mesma cena. Segundo Ting [1] um modelo utilizado em computação gráfica possui como exemplo quatro parâmetros, posição de um ponto do plano de projeção, vetor normal do plano de projeção, orientação da câmera, posição da câmera. Baseando-se nesses conceitos, foram escolhidos três configurações de câmeras para o jogo, e as definições serão mostradas posteriormente, mas já vale ressaltar que alterações justamente nesses parâmetros é o que permite uma variação nos tipos de projeção utilizados.

B. Modelos de Cor e Iluminação

Outro tópico que merece destaque, é o relativo à cor e à iluminação de uma cena, responsáveis por, juntamente com o uso de certos tipos de projeção, por causar efeitos de profundidade para o observador. O *Open-GL* usa como modelo de representação de cor o RGB para definir a cor do objeto, representada como um vetor com 3 componentes, relacionadas às cores da representação em RGB: vermelho (R), verde (G) e azul (B).

Entretanto, a cor, como elemento único em uma cena, não é capaz de fornecer, ao observador, todas as informações de profundidade. Para permitir que a visualização da cor do objeto, ou até mesmo alterá-la com o passar do tempo, é necessário que haja sobre o mesmo uma iluminação. Dando base a esse fato, Thing [1] enfatiza a importância da iluminação. Segundo ela, "a nossa percepção de cores se dá como resultado das interações entre as radiações luminosas oriundas da superfície dos objetos de interesse com as células fotossensíveis na retina". O *open-GL* dispõe de uma série de ferramentas para cálculos relacionados à cor e à iluminação, muitas destas, utilizadas no desenvolvimento do projeto.

C. Texturas

Para conseguir a representação de um objeto ou elemento de cena que se aproxime o máximo quanto possível do objeto real, é necessário um grande poder de processamento, que pode não ser viável dependendo da aplicação. Essa necessidade se dá, entre outras coisas, devido à complexidade encontrada em cálculos de cor iluminação, aspectos determinantes na forma como o observador enxerga um objeto. Para contornar esse problema, utiliza-se de um artifício para que se tente representar o material do qual é composta a superfície do elemento de cena representado por meio de uma malha que descreve esse material (geralmente, um arquivo de imagem). Essa malha é chamada de textura, e ela permite, justamente, dar ao objeto características que seriam complexas para se modelar, como rugosidade, porosidade ou outras características intrínsecas ao objeto representado. Assim, cobre-se o sólido

que representa o objeto com uma textura para que se consiga essas propriedades.

II. MÉTODOS

Nesta seção serão detalhados os aspectos relacionados diretamente à computação gráfica e que foram implementados no jogo. Além desses aspectos, existem também outros, não relacionados diretamente à computação gráfica, mas que são igualmente importantes e que, portanto, serão abordados posteriormente.

A. Modelagem dos Objetos

Para desenhar as paredes é necessário apenas definir os vértices é usar alguma primitiva disponível no *Open-GL*, para desenhar as paredes do jogo foi utilizado o *GL_QUAD*. Porém para desenhar figuras geométricas mais complexas, é necessário usar fundamentos do cálculo, mais especificamente mudança de sistemas de coordenadas. Para desenhar o *PAC-MAN* os Fantasmas e as frutinhas do jogo, foram necessários usar o sistema de coordenadas esféricas, equação 1, cilíndricas, equação 2, e esféricas, equação 3

$$x = r\cos(\Theta) \quad y = r\sin(\Theta)\sin(\Phi) \quad z = r\sin(\Theta)\cos(\Phi) \quad (1)$$

$$x = r\sin(\Theta)\cos(\Phi) \quad y = r\sin(\Theta) \quad z = r\sin(\Theta)\sin(\Phi) \quad (2)$$

$$x = r\cos(\Phi) \quad y = r\sin(\Phi) \quad z = r\sin(\Theta)\cos(\Phi) \quad (3)$$

B. Transformações Geométricas

Para realizar manipulações em um conjunto de pontos que formam um objeto, é necessário realizar transformações geométricas, como rotação e traslação. Por exemplo para virar o *PAC-MAN* é necessário aplicar uma rotação sobre o eixo Y, lembrando que o mapa do jogo está ortogonal ao eixo y, a matriz apresentada na equação 4. Para deslocar um objeto é necessário realizar uma translação do conjunto de pontos que compõe o objeto, a equação 5 mostra a matriz de transformação. No *Open-GL* existem funções que implementam essas transformações geométricas.

$$\begin{bmatrix} P_{x2} \\ P_{y2} \\ P_{z2} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta & 0 \\ 0 & 0 & 0 & 0 \\ -\sin\Theta & 1 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} P_{x2} \\ P_{y2} \\ P_{z2} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (5)$$

C. Iluminação

Para a iluminação é importante entender o conceito de iluminação do modelo de *phong*, onde a iluminação é dividida em três componentes, a reflexão ambiente, difusa e especular. A equação 5 mostra a expressão que define a intensidade luminosa em um ponto. No Open-GL são definidos os parâmetros de cada componente, a posição da luz e outras características.

$$I(x, y, z) = I_a(x, y, z) + f_{at}(k_d \sum_{i=1}^m I_{d,i}(L_i n) + k_s \sum_{i=1}^m I_{s,i}(VR_i)^n) \quad (6)$$

D. Colisão

Para realizar a colisão entre os objetos do jogo, foi utilizada uma simples comparação entre as coordenadas dos objetos que tem que colidir, por exemplo, o *PAC-MAN* precisa colidir com a parede, então é verificado se os pontos da origem do *PAC-MAN* somado ao seu raio, ocupa o mesmo espaço que a parede. A colisão entre o fantasma e a parede e o o *PAC-MAN* e fantasma segue o mesmo princípio.

III. MECÂNICAS DE JOGO

Nesta seção serão apresentadas as principais mecânicas implementadas no jogo. Essas mecânicas influenciam diretamente na percepção que se tem da aplicação, uma vez que, juntas, consistem na maneira de como o usuário interage com a aplicação. Por isso, por mais que não estejam relacionadas diretamente à computação gráfica (embora , por muitas vezes, se utilizem de seus conceitos), devem ser abordadas.

A. Menus e fases

Para a criação de um menu para o jogo em que ficariam armazenadas informações acerca de pontuação, por exemplo, foi preciso pensar em uma lógica que implementasse, basicamente dois estados de jogo: um estado "menu" e um estado "jogando". O primeiro estado ao se executar a aplicação é o estado "menu". Os estados são exclusivos um ao outro, ou seja, enquanto um executa, o outro não. Dessa forma, é possível alterar entre esses estados no decorrer da execução (implementando-se, assim, a possibilidade de um botão de *pause*, por exemplo.)

B. Movimentação do Pac-man

Um diferencial em relação ao jogo original do Pac-man, é a forma como se dá a movimentação do personagem no programa desenvolvido. No jogo original, o movimento do Pac-man é discretizado, de forma que ele somente consiga se movimentar de maneira perpendicular, ou seja, só pode fazer curvas no mapa em ângulos de 90°. Nessa nova versão do jogo, o movimento também é discretizado, já que não há como não de ser-lo. Entretanto, com uma discretização, agora, de 10°, a liberdade de movimento aumenta.

C. Fantasmas

Os fantasmas se movimentam pelo mapa de maneira aleatória, e sua direção é atualizada quando ocorre uma colisão com alguma parede. Para evitar que o fantasma escolha a mesma direção que ele colidiu, foi necessário bloquear essa possibilidade por determinado tempo, assim se o fantasma colidir com uma parede ele fica limitado a escolher três caminhos, se ele colidir com duas paredes em um pequeno intervalo ele só pode tomar duas decisões e assim sucessivamente. Quando os fantasmas se aproximam do *PAC-MAN* eles começam a perseguir, sempre tentando diminuir a maior distância perpendicular, quando o *PAC-MAN* come uma fruta especial os fantasmas se afastam.

IV. RESULTADOS

Esta seção trata da apresentação dos resultados obtidos ao final do desenvolvimento do projeto. Serão abordados os resultados relativos aos processos de definição de projeções, e processos colorização, iluminação e texturização. Serão brevemente abordados aspectos relacionados à mecânica do jogo, mas que não dizem respeito diretamente aos conceitos de Computação Gráfica.

A. Projeções

Para o projeto, foram pensados três tipos de projeções iniciais. Todas as projeções são perspectivas, uma vez que distorcem os objetos em cena para se ter uma noção de profundidade. Uma primeira é uma visão *table top* da cena, que simula a visão que se tem ao se jogar o Pac-man original em um *arcade*. A figura 2 mostra essa visão.

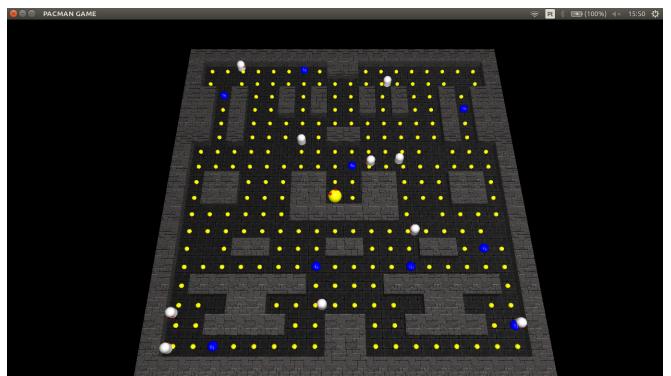


Figura 2. Visão *table top* do jogo

O segundo modelo de câmera é baseado no primeiro, mas aplicado um *zoom* para que se possa obter mais detalhes da cena. A figura 3 exemplifica esse tipo de câmera.

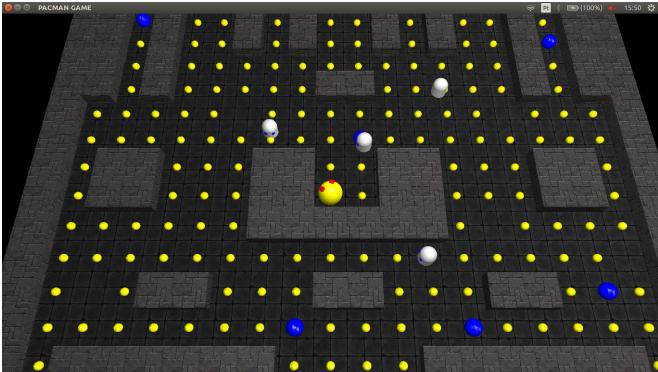


Figura 3. Visão *table top* com *zoom* do jogo

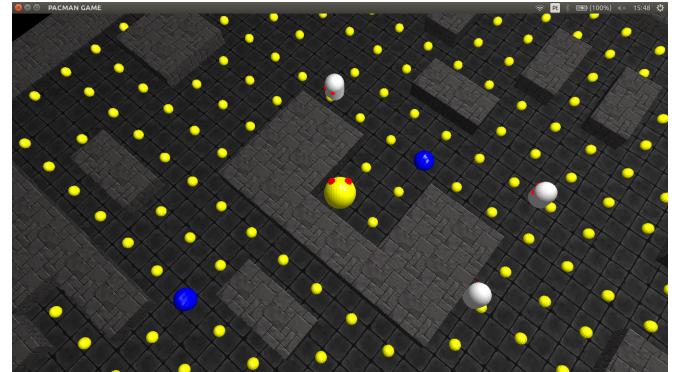


Figura 5. Iluminação nos diferentes blocos das paredes

Por fim, o terceiro modelo de câmera é um modelo que segue o terceiro modelo de câmera é um modelo que segue o Pac-man em *terceira pessoa*. Assim, é aplicado um *zoom* com foco no Pac-man, e a câmera tem uma dinâmica que, à medida que o Pac-man se move, no plano de projeção ele aparenta estar parado enquanto a cena se move ao seu redor. A figura 4 mostra essa implementação.

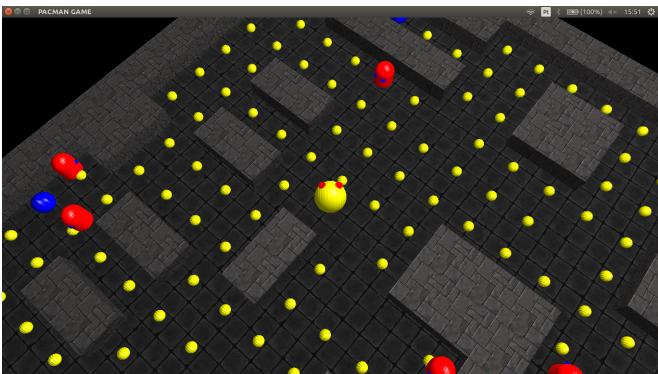


Figura 4. Visão em "terceira pessoa" do jogo

C. Texturização

Para tentar simular uma superfície irregular sem que, ao mesmo tempo, se consuma processamento, pode-se usar de técnicas de texturização, que usam uma malha que representa a superfície do objeto coberto. Para a aplicação desenvolvida, o objetivo era simular uma superfície *rochosa*, ou seja, irregular, rugosa e que refletia de uma forma muito característica a luz que incide sobre o respectivo objeto. Ao recobrir as paredes e o piso da cena com essa malha, o resultado obtido é o mostrado na figura 6.



Figura 6. A texturização causa certo efeito de irregularidade nos blocos da parede

B. Colorização e Iluminação

O processo de colorização da aplicação desenvolvida se deu de forma que os objetos, num primeiro momento, fossem coloridos e iluminados com uma única componente de luz ambiente (excluindo, nesse momento, componentes difusas e especulares). Dessa forma, entretanto, noções de profundidade são grandemente prejudicadas. Por isso, se fez necessário incluir essas componentes especulares e difusas. Após essa configuração, o sistema de luzes e cores fica como é mostrado na figura 5.

D. Mecânicas Gerais

As particularidades na implementação desse projeto que não envolviam conceitos diretos de Computação Gráfica, aspectos cuja apresentação é importante, se deram, basicamente, na forma de dinâmicas dentro do jogo. Alguns desses aspectos valem ser mencionados. O primeiro diz respeito à movimentação do Pac-man na cena 3D. Foi desenvolvido um algoritmo para a movimentação do Pac-man que permitia que ele se movesse, não mais de forma perpendicular, mas sim em ângulos múltiplos de 10° no seu próprio eixo. Para complementar essa movimentação, o Pac-man pode andar no sentido positivo ou negativo do eixo perpendicular à face mais externa ao "rosto" do Pac-man. A figura 7 demonstra esse aspecto.

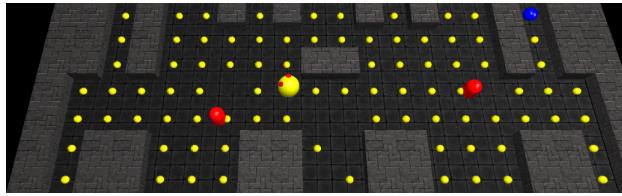


Figura 7. Movimentação do Pac-man em seu próprio eixo

V. CONCLUSÕES

Como conclusão do projeto, podem ser citados e reavaliados todos os aspectos abordados neste documento. Isso se deve ao fato de que existem diferenças claras em uma proposta de projeto de cena em três dimensões (se se compara a uma cena 2D).

Uma das principais diferenças está no fato de que uma cena 3D possui muitos elementos e informações do "mundo real", e que, nesse aspecto, são responsáveis por uma maior noção de profundidade para o observador. Intrinsecamente a essa maior noção de profundidade, entretanto, estão relacionados uma série de conceitos de luz e iluminação, por exemplo, que são os responsáveis pela tentativa de se representar fielmente um ambiente real (ou simplesmente tri-dimensional) em um plano de projeção 2D.

Por mais que se trate de um projeto limitado, a ideia de se representar o comportamento da luz, das cores ou dos objetos nas cenas desenvolvidas toma como base comparações com a forma com que esses mesmos aspectos se dão no mundo real. Por isso, trabalhar em projetos simplificados, como este, é uma base sólida para projetos em que se busca, por exemplo, maior realismo nas cenas criadas computacionalmente.

REFERÊNCIAS

- [1] S. T. Wu, *Sistemas de Informações Gráficas*, 1st ed. Campinas: UNICAMP, 2009.