Evento: XXVII Jornada de Pesquisa

FERRAMENTA INTERATIVA PARA ENSINO DE TRANSFORMAÇÕES GEOMÉTRICAS TRIDIMENSIONAIS APLICADAS À COMPUTAÇÃO GRÁFICA[1]

INTERACTIVE TOOL TO TEACH TRIDIMENSIONAL GEOMETRIC TRANSFORMATIONS APPLIED TO COMPUTER GRAPHICS

Maikon Cismoski dos Santos[2], Bruno Bearsi da Paixão[3]

[1] Pesquisa desenvolvida no IFSUL - Instituto Federal de Educação Ciência e Tecnologia Sul Rio-Grandense, Câmpus Passo Fundo.
[2] Docente do Curso de Bacharelado em Ciência da Computação do IFSUL - Câmpus Passo Fundo.
[3] Discente do Curso de Bacharelado em Ciência da Computação do IFSUL - Câmpus Passo Fundo.

RESUMO

Transformações geométricas são operações largamente empregadas no desenvolvimento de diferentes aplicações que necessitem o uso de rotação, escala e translação, tais como editores de imagem, ferramentas CAD e aplicações gráficas como jogos e simuladores. Este trabalho apresenta o desenvolvimento de uma aplicação que auxilia o ensino de transformações geométricas no contexto da computação gráfica, permitindo a visualização dessas operações em um ambiente tridimensional, bem como a exibição das matrizes que produzem essas transformações. Utilizando a aplicação desenvolvida, o estudante pode compreender os conceitos envolvendo transformações geométricas tridimensionais, possibilitando a visualização das matrizes de cada tipo de transformação, bem como o resultado em um canvas, o qual é apresentado em uma página HTML.

Palavras-chave: Transformações Geométricas. Computação Gráfica. Ferramenta de Ensino.

ABSTRACT

Geometric transformations are operations widely used in the development of different applications that require the use of rotation, scale and translation, such as image editors, CAD tools and graphic applications such as games and simulators. This work presents the development of an application that helps the teaching of geometric transformations in the context of computer graphics, allowing the visualization of these operations in a three-dimensional environment, as well as the display of the matrices that produce these transformations. Using the developed application, the student can understand the concepts involving three-dimensional geometric transformations, allowing the visualization of matrices of each type of transformation, as well as the result in a canvas, which is presented in an HTML page.

Keywords: Geometric Transformations. Computer Graphics. Learning Tool.

INTRODUÇÃO

A visualização de modelos tridimensionais ou bidimensionais é um excelente método para auxiliar o estudo de conceitos da computação gráfica ou álgebra linear. Diversas ferramentas foram criadas nesse sentido, algumas podendo ser acessadas pelo próprio navegador, assim podendo ser utilizadas por professores e alunos de diferentes níveis de escolaridade.

As transformações geométricas, no contexto da computação gráfica, são operações aplicadas através de cálculos de álgebra linear com matrizes, para modificar a posição dos vértices de um objeto tridimensional ou bidimensional. Assim permitindo a alteração da posição, tamanho, ou rotação de um objeto em uma cena.

Durante o estudo da computação gráfica, especificamente no desenvolvimento de uma aplicação gráfica, compreender a aplicação de transformações geométricas pode ser complexo, muitas vezes o estudante pode não conseguir o efeito esperado, ao utilizar uma transformação, uma vez que há diferentes eixos e a ordem em que tais transformações pode impactar no resultado final. Os conceitos relacionados às transformações geométricas podem ser explicados de forma mais clara com a utilização de uma ferramenta que demonstra os tipos de transformações sendo realizadas em um objeto e o efeito que elas causam conforme sua ordem.

Considerando esse problema, foi desenvolvida uma aplicação com WebGL 2 que permite adicionar objetos em um ambiente tridimensional e modificar os valores de um conjunto de transformações sendo aplicadas no mesmo, assim como sua ordem.

REFERENCIAL TEÓRICO

Nesta seção é apresentado o embasamento teórico sobre o qual se fundamentou o trabalho, sendo abordado os fundamentos da Computação Gráfica, o processo de renderização, a aplicação da álgebra linear em transformações geométricas, seguidos de uma introdução a API (*Application Programming Interface*) WebGL 2 e bibliotecas Javascript.

COMPUTAÇÃO GRÁFICA

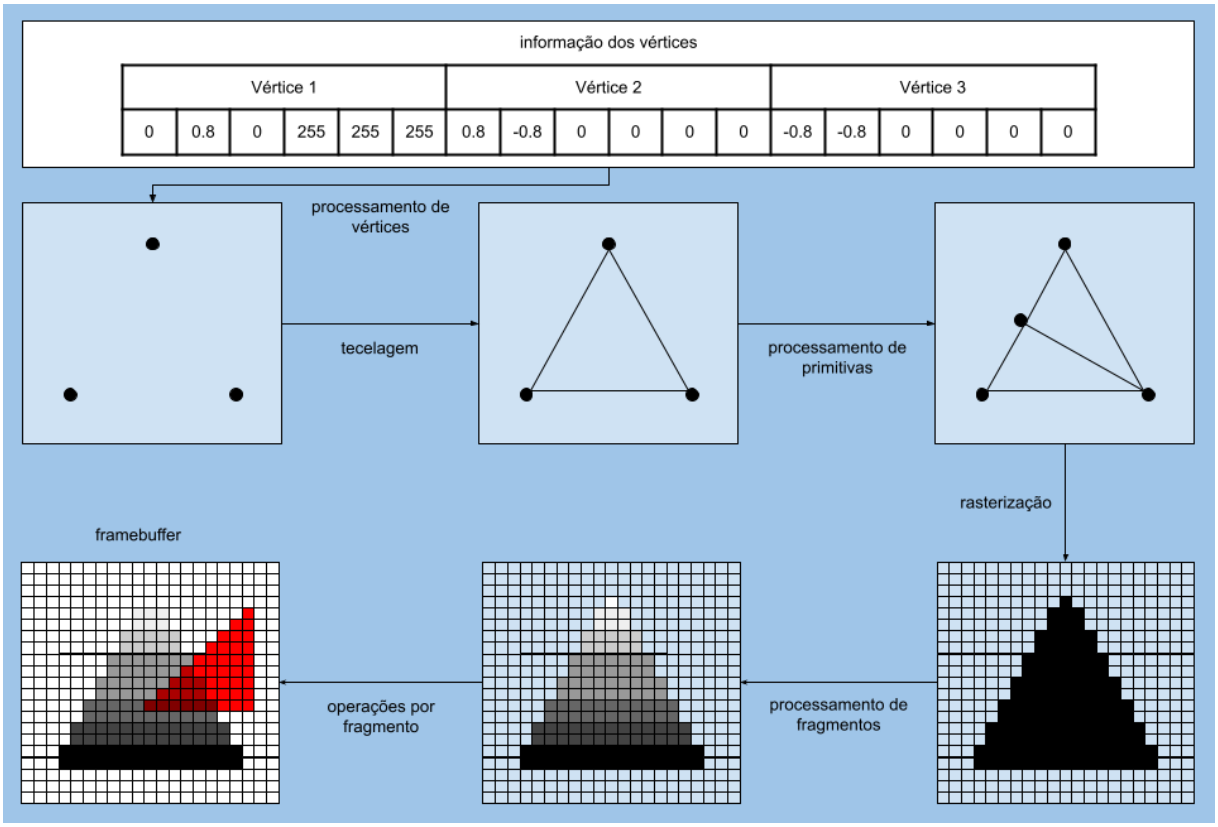
Segundo Angel(2011) a Computação Gráfica é o estudo de todos os aspectos voltados à produção de imagens. Podemos incluir nessa definição tanto as fundações matemáticas e computacionais quanto as técnicas desenvolvidas para a geração e processamento de imagens.

O processo de renderização é o conjunto de passos necessários para a conversão de um modelo 2D ou 3D em uma imagem. Esse processo pode ser baseado em software, quando as instruções são executadas na CPU (*Central Processing Unit*) do computador, ou hardware, quando o processamento é realizado em uma GPU (*Graphics Processing Unit*). No contexto web principalmente, a renderização pode ser realizada de forma remota, renderização baseada em servidor, ou pode ocorrer localmente, renderização baseada em cliente (JONES, 2012).

A renderização no WebGL 2 ocorre por hardware de forma local, sendo realizada através de uma série de passos: processamento dos vértices, tecelagem, processamento de primitivas, rasterização, processamento de fragmentos, operações por fragmento. Desses passos, o processamento dos vértices e fragmentos é realizado através de programas chamados *shaders* desenvolvidos pelos próprios programadores conforme a necessidade da aplicação.

Na Figura 1 está exemplificado o processo de renderização de um simples triângulo: Um *array* contendo as informações de posição e cor dos vértices do triângulo são enviadas a GPU; a posição de cada vértice é processada; as linhas são desenhadas através do processo de tecelagem com o objetivo de dividir o polígono sendo renderizado em partes padronizadas; o próximo passo é tentar dividir em triângulos menores; o objeto então é convertido em fragmentos utilizando um algoritmo de rasterização; a cor de cada fragmento é processada; por fim são realizadas diversas operações como mesclar cores de objetos semitransparentes e desconsiderar objetos sobrepostos a outros para então as informações serem gravadas no *framebuffer*, local onde as informações dos pixels exibidos na tela são armazenados.

Figura 1 - *Pipeline* de renderização



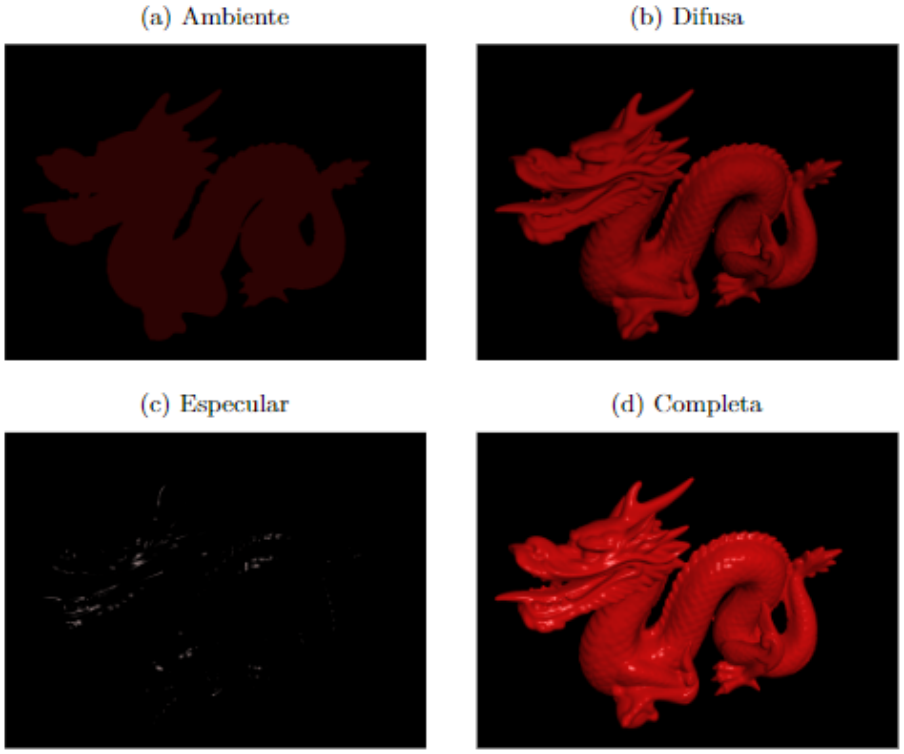
Fonte: Autor

Apenas as características de forma e cor não são suficientes para produzir uma boa visualização de um ambiente tridimensional. A luz afeta a aparência do mundo de diferentes formas. Nossa percepção, quanto à natureza física de um objeto ou o realismo de uma cena ou imagem, é muito influenciada pela interação da luz com os objetos no ambiente (MATSUDA, 2013). Os modelos mais comuns utilizados hoje são baseados no modelo ADS (*Ambient, Diffuse, Specular*):

- Reflexão de ambiente: iluminação fraca do ambiente em todos os objetos;
- Reflexão difusa: iluminação de um ou mais objetos dependendo do ângulo e da incidência de um ponto de luz;
- Reflexão especular: simula pontos de luminosidade onde a luz refletiria mais diretamente nos olhos do observador;

Na Figura 2 são demonstrados cada componente do modelo ADS e o resultado de sua união.

Figura 2 - Componentes do modelo ADS



Fonte: Autor

Dois modelos de sombreamento básicos são Goraud que calcula a cor final durante o processamento dos vértices, assim sendo mais rápido, porém menos preciso, e Phong que calcula a cor final no processamento dos fragmentos utilizando o modelo ADS (GHAYOUR, 2018).

TRANSFORMAÇÕES GEOMÉTRICAS TRIDIMENSIONAIS

A álgebra linear é a área da matemática que estuda matrizes e vetores, suas aplicações e operações. O processo de renderização na Computação Gráfica utiliza representações em forma de matrizes e vetores e realiza manipulações através das operações criadas para ambas as estruturas. Jones (2012) descreve que o entendimento dos conceitos

aplicados às operações na álgebra linear é essencial durante os estudos envolvendo luzes e projeções de câmera.

Transformações geométricas como translação, rotação e escalonamento são realizadas através da multiplicação de matrizes (MARSCHNER, 2021), o resultado então é multiplicado pelo vetor que representa os valores de posição X, Y e Z de cada vértice durante o processamento de vértices na pipeline de renderização para aplicar as transformações no objeto, alterando a visualização dos objetos.

Todas as transformações começam com uma matriz identidade, uma matriz formada de zeros com exceção de sua diagonal principal, a qual é composta de uns. Essa matriz recebe então os fatores de uma transformação, sendo que a transformação que ela realizará depende do posicionamento desses fatores. Normalmente, para aplicação em objetos tridimensionais, utiliza-se na computação gráfica matrizes identidade de dimensão 4x4 por comportarem todos os tipos de transformações.

A transformação de escala é utilizada para alterar o comprimento de um objeto, na Figura 3 é possível visualizar o posicionamento dos fatores de escala de cada eixo (Sx, Sy, Sz), bem como a matriz para essa transformação.

Figura 3 - Matriz de Escala

$$\text{escala}(Sx,Sy,Sz) = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fonte: Autor

Uma rotação utiliza um ângulo para movimentar um vetor ao redor do ponto de origem, utiliza-se uma matriz específica para calcular a rotação em cada eixo. As Figuras 4, 5 e 6 apresentam as matrizes para rotação com base em um ângulo φ e nos eixos X, Y e Z, respectivamente.

Figura 4 - Matriz de rotação no eixo X

$$\text{rotaçãoX}(\varphi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fonte: Autor

Figura 5 - Matriz de rotação no eixo Y

$$\text{rotaçãoY}(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fonte: Autor

Figura 6 - Matriz de rotação no eixo Z

$$\text{rotaçãoZ}(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fonte: Autor

A operação de translação é usada para mover objetos, multiplicando sua matriz com um vetor, adiciona os fatores da translação aos elementos do vetor. Na Figura 7 é possível visualizar a posição dos fatores dentro da matriz.

Figura 7 - Matriz de Translação

$$translação(Tx,Ty,Tz) = \begin{bmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fonte: Autor

WEBGL 2 E BIBLIOTECAS JAVAVASCRIPT

O WebGL 2 é uma API gráfica 3D baseada na web (JONES, 2012). Foi desenvolvido com base no OpenGL ES 3.0 (*Open Graphics Library for Embedded Systems*), uma API para renderização 2D e 3D utilizando recursos de GPUs, e projetado para sistemas embarcados como *smartphones*, *tablets*, consoles de *video-game* (KHRONOS, 2015). Atualmente, as versões mais atualizadas dos navegadores Apple Safari, Google Chrome, Microsoft Edge e Mozilla Firefox já são compatíveis com a tecnologia (KHRONOS, 2017).

Jones (2012) afirma que o processo de renderização pode ocorrer de diferentes maneiras. A renderização pode ser baseada tanto em software ou em hardware considerando a existência ou não de uma GPU, quanto em cliente ou servidor considerando o local onde será realizado o processo. O WebGL 2 traz uma abordagem que utiliza o componente gráfico da máquina local.

WebGL 2 foi programado em Javascript, isso oferece conjunto de vantagens em sua utilização como: facilidade de integração com bibliotecas Javascript e tecnologias do HTML5, gerenciamento de memória automático reduz a quantidade de código deixando-o mais limpo e compreensível, não necessitando compilar o código mudanças podem ser rapidamente visualizadas, suporte para diferentes plataformas como Android e IOS.

WebGL 2 é um novo recurso para o desenvolvimento de aplicações gráficas na web, uma biblioteca que trabalha sobre o elemento canvas do HTML5 (*Hyper Text Markup*

Language, versão 5) sem exigir software adicional. Segundo (MATSUDA, 2013), com a evolução dos padrões da web, os navegadores deixam de ser um simples mecanismo de apresentação para se tornarem plataformas para aplicações.

O WebGL 2 permite apenas a utilização da GPU para renderizar em um canvas, o desenvolvimento de uma aplicação precisa de uma interface para interação com o usuário. Nesse sentido, a biblioteca dat.GUI pode ser empregada para a criação de interfaces. Através dela é possível desenvolver um menu com *sliders* ou caixas de texto, vinculados a objetos de um *script*, e configurar o evento que vai ocorrer após a interação com o usuário.

A criação de matrizes e vetores de diferentes tamanhos são elementos necessários no desenvolvimento de aplicações gráficas. A glMatrix é uma biblioteca de funções matemáticas relacionadas à álgebra linear, possuindo um conjunto de funções que realizam operações em matrizes e vetores de diferentes dimensões.

TRABALHOS RELACIONADOS

Com o crescimento da informática nas salas de aula, a utilização de recursos audiovisuais torna-se um grande suporte para melhorar a qualidade do ensino. O ensino de transformações geométricas, no contexto da computação gráfica, através de ferramentas que permitam a visualização de exemplos já foi abordado em trabalhos passados.

Um exemplo é o InfoGraph 3D, um ambiente gráfico desenvolvido em Java para auxiliar no ensino da Computação Gráfica (CERQUEIRA, 2009). A aplicação apresenta 9 lições sobre tópicos da Computação Gráfica, cada uma contendo uma visualização interativa e o código-fonte que gerou a visualização.

Uma das lições introduz os conceitos de transformações geométricas, durante as aulas o professor pode utilizá-lo para demonstrar as transformações geométricas sendo realizadas em objetos 3D e comentar o código utilizado para implementar esse exemplo. Alguns pontos negativos do infograph são: não permitir a adição de objetos na tela, apenas a manipulação de um objeto já existente; e não mostrar as matrizes sendo aplicadas para realizar as transformações.

Outro trabalho a ser destacado é o ADUBOGL (ARAÚJO, 2012), uma aplicação desenvolvida com o objetivo de exercitar o entendimento, quanto a utilização de transformações geométricas, dos alunos. O professor pode desenvolver atividades para os alunos exercitarem a utilização de transformações e compreenderem a hierarquia das operações sendo aplicadas aos objetos. Sua utilização pode ser confusa para algumas pessoas, pois é necessário posicionar blocos de uma forma específica no programa para criar objetos e aplicar as transformações.

Ambos os trabalhos obtiveram resultados positivos como ferramentas para demonstrar a utilização de transformações geométricas, mas não abordam a fundo os conceitos relacionados à álgebra linear. Esses trabalhos não demonstram as matrizes de cada transformação sendo aplicada ou a matriz resultante de serem acumuladas por multiplicação.

METODOLOGIA

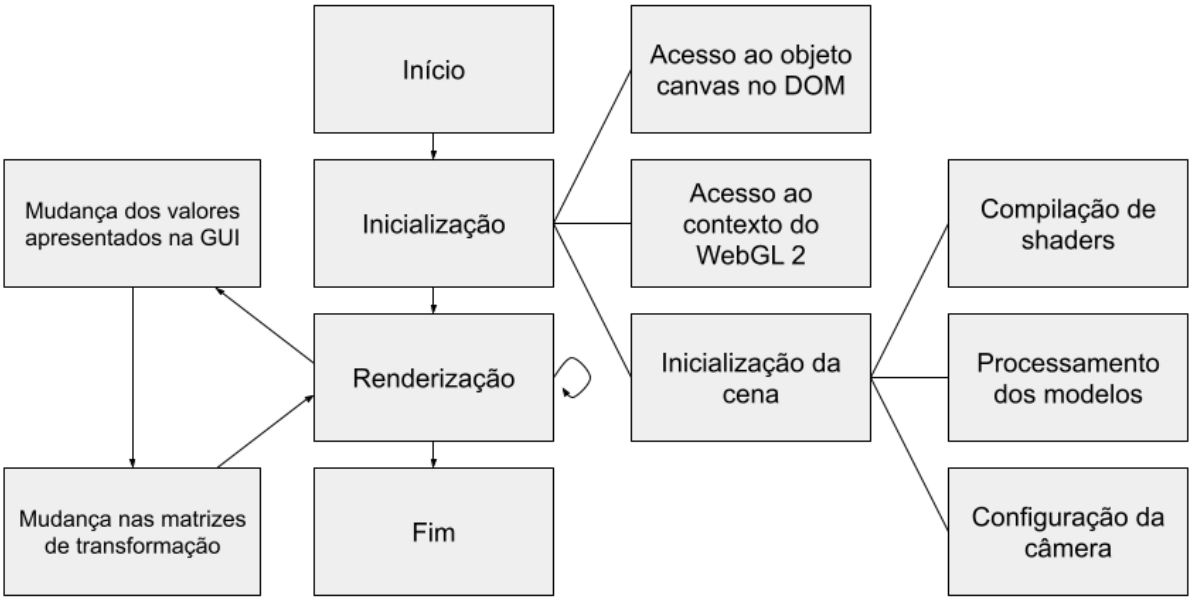
Nesta seção é apresentada a metodologia que foi utilizada para alcançar o objetivo proposto. Pode-se dividir o processo de desenvolvimento em duas partes: a aplicação do ambiente 3D e a interface de usuário.

A aplicação do ambiente 3D foi implementada utilizando a API WebGL 2 e Javascript para realizar o processo de renderização. O ambiente desenvolvido utiliza uma câmera perspectiva e um shader com o modelo de iluminação phong para melhor visualizar as faces dos objetos na cena. Foram desenvolvidas funções auxiliares que carregam arquivos JSON com informações dos modelos 3D mostrados na aplicação e arquivos contendo os shaders utilizados. Para implementar o código que aplica transformações geométricas nos objetos é necessário a utilização de um conjunto de funções que realizam cálculos de álgebra linear com matrizes e vetores, para essas funções foi utilizada a biblioteca glMatrix.

Na Figura 8 é demonstrado o funcionamento da aplicação durante o processo de renderização. Após a página ser carregada, uma função recebendo um ID retornará o contexto WebGL 2 do *canvas* com mesmo ID. Com o contexto, objetos inicializados na cena podem ser carregados, nesse momento as funções implementadas na aplicação são utilizadas para: compilação de *shaders*, criação de *buffers* com os dados dos objetos, calcular as matrizes de

transformação do modelo e passar as informações necessárias para renderizar os objetos. Por fim, a inicialização de um loop de renderização que vai produzir a cena no *canvas* conforme especificado e atualizar as matrizes, das transformações aplicadas no objeto, mostradas abaixo do canvas.

Figura 8 - Processo de Renderização



Fonte: Autor

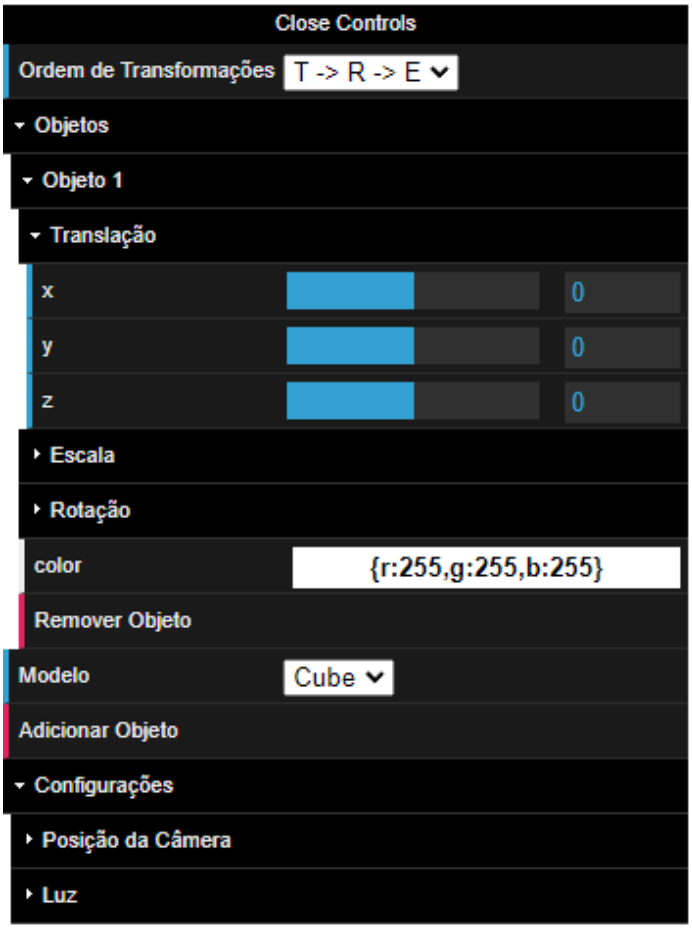
São aplicadas em cada objeto as transformações de translação, rotação e escalonamento. Para proporcionar a interação com o usuário, foi utilizada a biblioteca dat.GUI para desenvolver uma interface de usuário, que permite alterar os fatores das transformações em cada objeto, sendo que os valores iniciais não causam alterações na visualização do objeto em comparação com o modelo original.

RESULTADOS

Com base na metodologia proposta, a aplicação permite a adição de objetos no ambiente tridimensional e possibilita a realização das transformações geométricas de translação, rotação e escalonamento.

A Figura 9 apresenta o menu da aplicação utilizado para interagir com a cena. O primeiro campo é um seletor que permite alterar a ordem na qual as transformações geométricas são aplicadas nos objetos da cena, seu valor indica que a ordem inicial será translocação, rotação e escalonamento . Abaixo do seletor são listados os objetos contidos na cena, em que é possível alterar, através de *sliders*, os parâmetros utilizados para realizar as transformações geométricas (translação, escalonamento e rotação) em cada eixo(X, Y, Z), e ainda modificar a cor do objeto ou removê lo através de um botão. Os próximos componentes do menu permitem adicionar um objeto na cena a partir do seletor e do botão contidos na interface. Por fim, os últimos elementos do menu são empregados para a configuração de câmera e luz da cena.

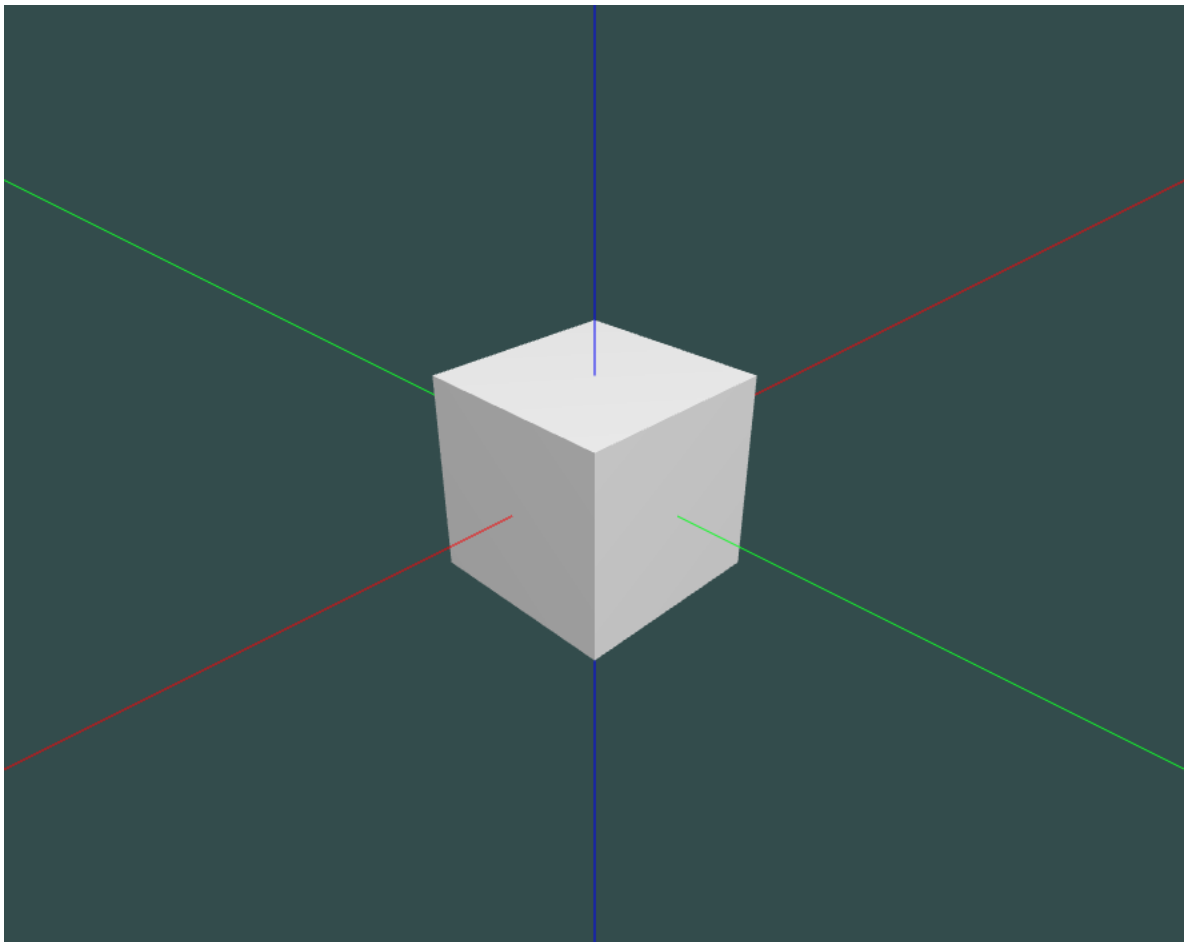
Figura 9 - Menu da Aplicação



Fonte: Autor

Na Figura 10 é possível visualizar o ambiente 3D, nele a câmera é iniciada a 20 unidades de distância do ponto de origem do ambiente (posição 0, 0, 0), -45 graus do eixo X e 60 graus do eixo Y. A luz é iniciada com a cor branca na posição X=75, Y=125 e Z=-100. Para melhorar a percepção do espaço do ambiente tridimensional, os eixos X, Y e Z são apresentados por linhas nas cores vermelho, verde e azul, respectivamente. Um cubo é apresentado sem nenhuma transformação sendo aplicada, ou seja, com os fatores de translação e rotação zerados e fatores de escala em um.

Figura 10 - Canvas de Visualização

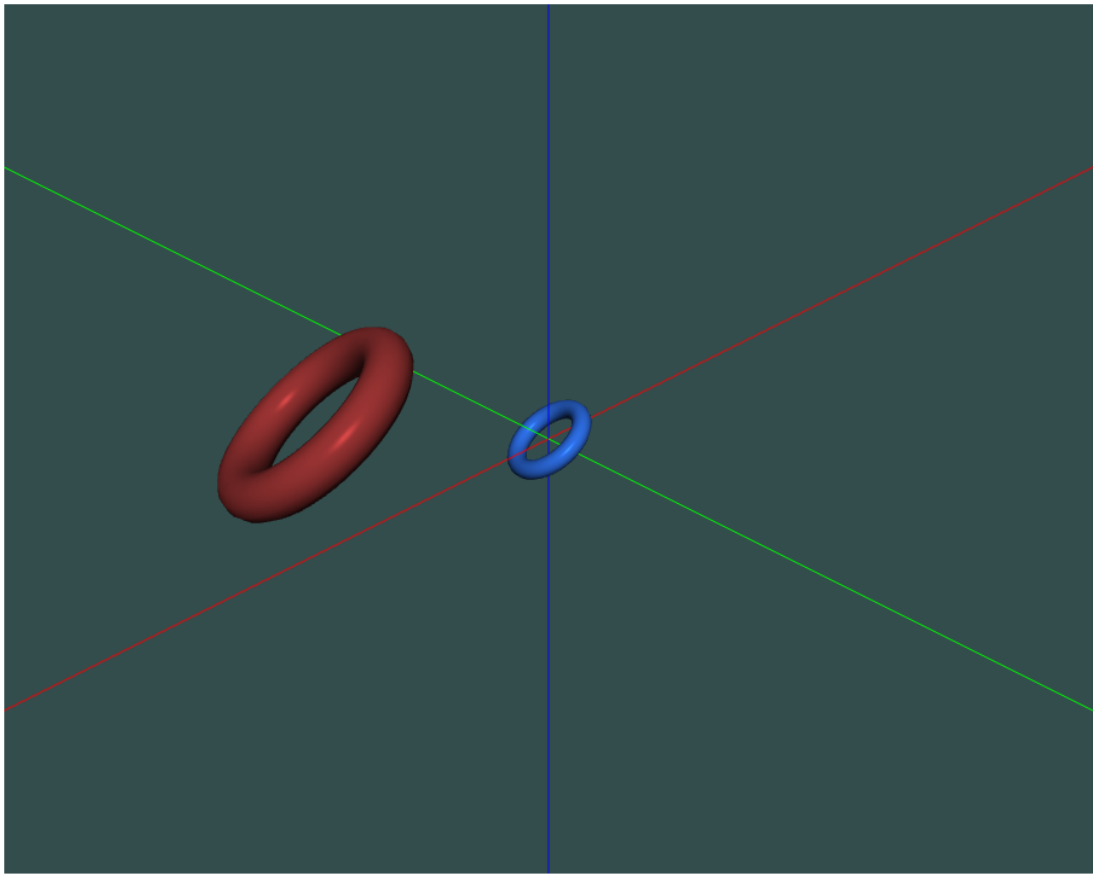


Fonte: Autor

Na Figura 11 é demonstrado a visualização do ambiente ao ser removido o cubo adicional e adicionado 2 novos objetos: o primeiro recebeu um tom de vermelho como cor,

uma translação de 2 unidade em X, 3 unidades em Y e 5 unidades em Z, uma escala de 2 unidades em todos os eixos e uma rotação de 40 graus em X, 30 graus em Y e 80 graus em Z; o segundo recebeu um tom de azul como cor e os mesmos valores de rotação do primeiro torus, não foram alterados seus valores de translação e escala. Na Figura 12 são mostradas as matrizes de translação, escalonamento e rotação geradas a partir das variáveis da interface, e a matriz resultante da multiplicação das três matrizes, para cada objeto na cena.

Figura 11 - Visualização do ambiente



Fonte: Autor

Através dos elementos apresentados pela aplicação, o estudante pode então validar seus conhecimentos: testando as transformações que pretende utilizar e verificando seu efeito em um objeto tridimensional, dessa forma auxiliando seu aprendizado ao apresentar a aplicação prática dos conceitos sendo estudados.

Figura 12 - Matrizes de Transformação

Matrizes de Modelo

Objeto 2

Translação	Escala	Rotação
1.000 0.000 0.000 2.000	2.000 0.000 0.000 0.000	0.386 -0.509 0.769 0.000
0.000 1.000 0.000 3.000	0.000 2.000 0.000 0.000	0.774 -0.275 -0.571 0.000
0.000 0.000 1.000 5.000	0.000 0.000 2.000 0.000	0.503 0.815 0.288 0.000
0.000 0.000 0.000 1.000	0.000 0.000 0.000 1.000	0.000 0.000 0.000 1.000
Resultante	|	|
0.772 -1.019 1.538 0.000	|	|
1.547 -0.551 -1.142 0.000	|	|
1.005 1.630 0.576 0.000	|	|
5.606 2.231 1.265 1.000	|	|

Objeto 3

Translação	Escala	Rotação
1.000 0.000 0.000 0.000	1.000 0.000 0.000 0.000	0.386 -0.509 0.769 0.000
0.000 1.000 0.000 0.000	0.000 1.000 0.000 0.000	0.774 -0.275 -0.571 0.000
0.000 0.000 1.000 0.000	0.000 0.000 1.000 0.000	0.503 0.815 0.288 0.000
0.000 0.000 0.000 1.000	0.000 0.000 0.000 1.000	0.000 0.000 0.000 1.000
Resultante	|	|
0.386 -0.509 0.769 0.000	|	|
0.774 -0.275 -0.571 0.000	|	|
0.503 0.815 0.288 0.000	|	|
0.000 0.000 0.000 1.000	|	|

Fonte: Autor

CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de uma ferramenta que auxilia no aprendizado de transformações geométricas aplicadas à computação gráfica. Constituída por uma aplicação 3D interativa e da visualização das matrizes aplicadas nos objetos da cena, o estudante é capaz de melhorar seu entendimento de como utilizar cada transformação e da ordem em que devem ser empregadas para alcançar o resultado esperado.

Na aplicação é possível interagir com um menu que permite a alteração da ordem das transformações, o valor de cada transformação por objeto, adição e remoção de objetos do ambiente, configuração da câmera e luz do ambiente. A interação com essas variáveis permite visualizar com clareza os efeitos das transformações sendo aplicadas aos objetos.

Como trabalhos futuros pretende-se ampliar as opções da aplicação para abordar outros conceitos da computação gráfica como algoritmos de sombreamento, mapeamento de textura e projeções de câmera.

REFERÊNCIAS BIBLIOGRÁFICAS

ANGEL, D. S. E. Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL. [S.l.]: Pearson, 2011.

ARAÚJO, L. P. D. Adubogl – aplicação didática usando a biblioteca opengl. 2012.

CERQUEIRA, V. C. d. S. R. G. Aprendendo conceitos de computação gráfica através de um ambiente multimídia e interativo com opengl. Anais do Workshop de Informática na Escola, 2009.

GHAYOUR, D. C. F. Real-Time 3D Graphics with WebGL 2. [S.l.]: Packt Publishing, 2018.

JONES, D. C. B. WebGL Beginner's Guide. [S.l.]: Packt Publishing, 2012.

KHRONOS. OpenGL ES Overview. 2015. Disponível em: <https://www.khronos.org/opengles/>. Acesso em: 3 julho 2022.

KHRONOS. WebGL Overview. 2017. Disponível em: <https://www.khronos.org/webgl/>. Acesso em: 3 julho 2022.

MARSCHNER, S.; SHIRLEY, P. Fundamentals of computer graphics. Boca Raton: Crc Press, Taylor & Francis Group, 2021.

MATSUDA, R. L. K. Webgl Programming Guide: Interactive 3D Graphics Programming with Webgl. [S.l.]: Addison-Wesley, 2013.