

# UNIVERSIDADE FEDERAL DA PARAÍBA

Discente: Carlos Fabrício da Silva Pontes, Matrícula: 20190020637

Discente: Davi José Lucena Luiz, Matrícula: 20190115490

Docente: Christian Azambuja Pagot

Disciplina: Introdução à Computação Gráfica

## ATIVIDADE PRÁTICA 2

João Pessoa, 2021

# 1. Desenvolvimento da atividade:

Primeiramente realizamos um passo a passo para criar cada matriz que fosse necessária para a composição dos cálculos e levar os vértices de um espaço para o outro. Para desenvolver as estratégias que foram aplicadas dentro da atividade utilizamos como base as aulas da disciplina, para que fosse possível entender como cada matriz e as suas operações funcionam dentro do pipeline gráfico. Como o espaço para o objeto ser descrito já estava criado, inicializamos a atividade partindo da criação de um bloco “for” para aplicar a matriz model em todos os vértices do cubo, fazendo a transição do espaço do objeto para o espaço do universo, tínhamos em mãos as possíveis transformações a serem feitas para o procedimento da atividade.

Realizado os cálculos com a matriz model e já dentro do espaço do universo, configuramos o vetor posição da câmera ( $X_{cam}$ ,  $Y_{cam}$  e  $Z_{cam}$ ) e o ponto para onde ela aponta para visualizar o objeto dentro do espaço. Após isso realizamos também o cálculo do seu vetor de direção, para podermos configurar o vetor “up” e colocar a câmera na posição correta para visualizar o objeto. Sendo assim, foi possível aplicar a mudança de base dos vértices levando-os do espaço do universo para o espaço da câmera e assim tornando possível o uso da matriz view nos próximos procedimentos da atividade.

Concluída toda a mudança anterior e podendo fazer uso da matriz view após a transformação dos vértices, aplicamos a matriz view em todos os vértices do objeto, sendo possível realizar rotação e translação através dos vetores de posição, direção e “up” da câmera.

Em seguida, iniciamos a construção da matriz de projeção, ou seja, transformar os vértices do espaço da câmera para o espaço de recorte para que seja possível projetar os vértices no plano de projeção. Nesta etapa do pipeline gráfico é onde ocorre a projeção dos vértices contra o viewplane e o resultado esperado por meio deste processo é uma sensação de profundidade do objeto rasterizado na tela. Para o desenvolvimento desta etapa utilizamos um modelo apresentado em aula, substituímos o “d”, ou seja, a direção da câmera por 1 e aplicamos a matriz de projeção em cada vértice do objeto e tornando possível a projeção dos vértices no viewplane.

Após essa etapa, partimos para a realização da homogeneização, transformar os vértices do espaço de recorte para o espaço canônico. Simplesmente dividimos as coordenadas dos vértices que estavam no espaço de recorte por sua

coordenada homogênea, sendo assim, alteramos a geometria do objeto, possivelmente tornando-o mais distante em relação à câmera, porém, transferindo o objeto para o espaço canônico.

Para o próximo passo, fizemos a multiplicação da matriz escala pela matriz de translação para obtermos a matriz viewport e aplicá-la aos vértices do cubo e levá-lo do espaço canônico para o espaço de tela, onde através da multiplicação de matrizes feita anteriormente transformamos as coordenadas dos vértices para realizar a rasterização da cena na tela.

Por fim, implementamos o algoritmo do ponto médio para rasterizar o modelo transformado através do pipeline gráfico. Em seguida, aplicamos a função para rasterizar cada aresta do cubo.

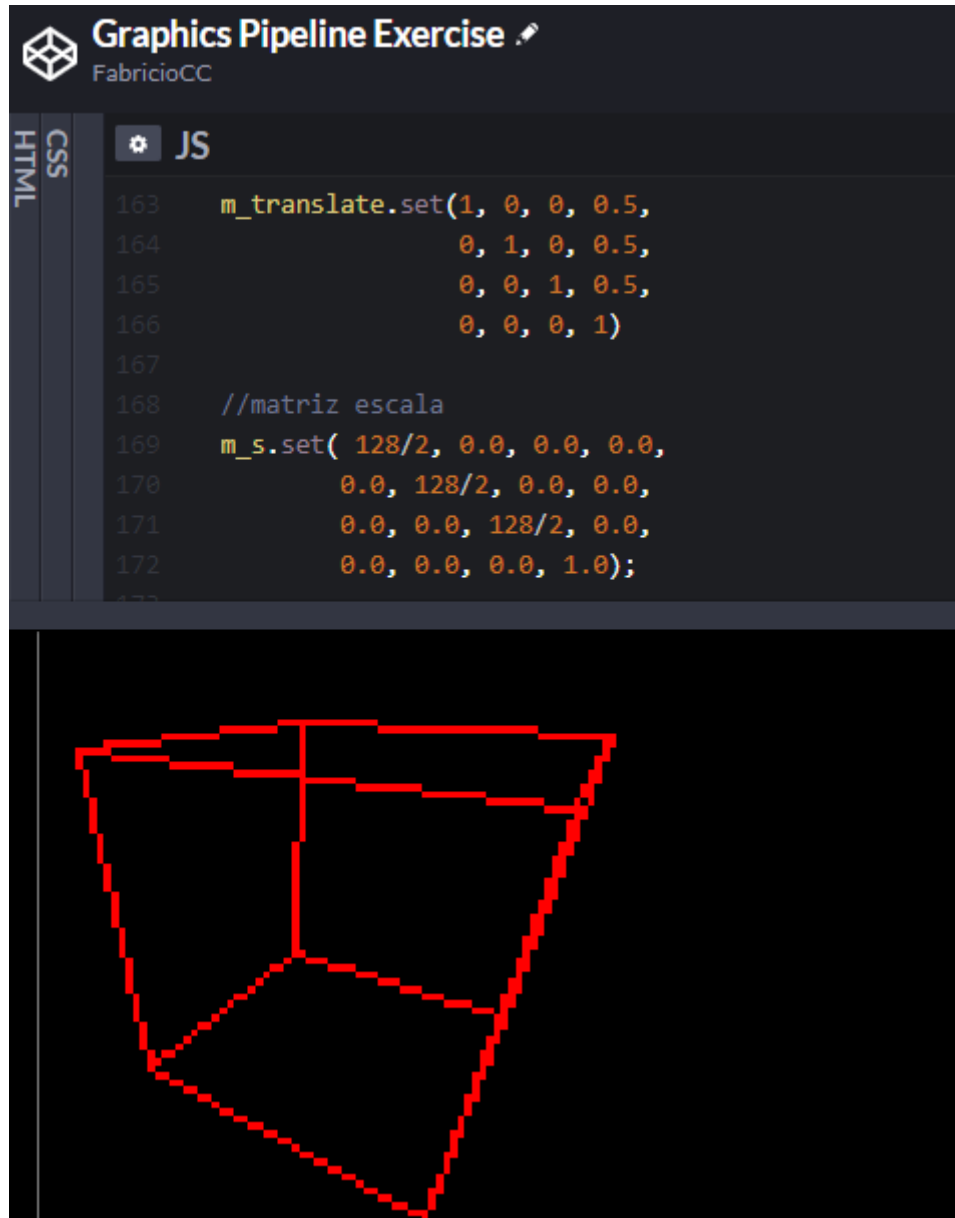
## **2. Explicação breve sobre as estratégias adotadas para o desenvolvimento da atividade:**

Nesta atividade fizemos a utilização do conceito de pipeline gráfico e dos estágios de cada matriz que são responsáveis por realizar a modelação de coordenadas de um objeto levando-o até o espaço de tela, o qual é bidimensional. Dentro deste desenvolvimento implementamos o pipeline gráfico, realizando cada passo necessário para que fosse possível obter a rasterização de linhas na tela dentro do espaço solicitado, a cada parte do pipeline gráfico que estava sendo desenvolvido fizemos uso do console para ter uma noção de como a implementação estava ocorrendo e se os cálculos estavam corretos. Para este desenvolvimento utilizamos também o algoritmo do ponto médio para rasterizar as arestas do cubo.

## **3. Dificuldades:**

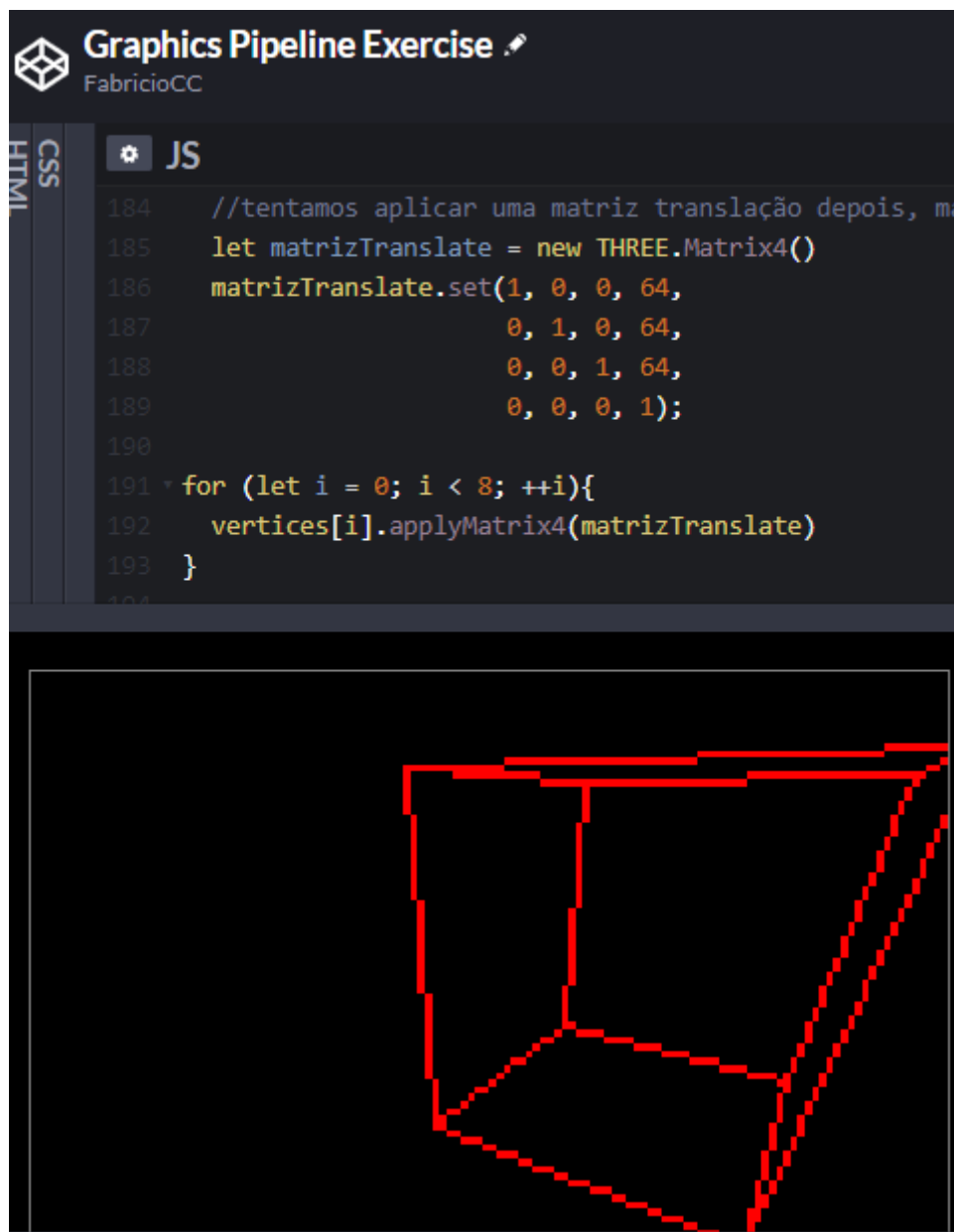
A principal dificuldade nesta atividade que tivemos foi na hora de mudar os valores da última coluna da matriz de translação do objeto para realizarmos testes e vermos como ele se comportaria no processo de translação após a rasterização. Fizemos diversas mudanças nesta matriz, porém o mesmo erro sempre persistia, ou seja, o objeto se demonstrava deformado na tela, não

sendo possível visualizá-lo de forma correta e não podendo ver também uma parte de sua face.



**Figura 1** – Exibição incorreta do cubo e erro na matriz de translação

Após essa etapa, tentamos mais uma vez ver como o cubo se comportaria alterando novamente os valores da última coluna da matriz de translação.

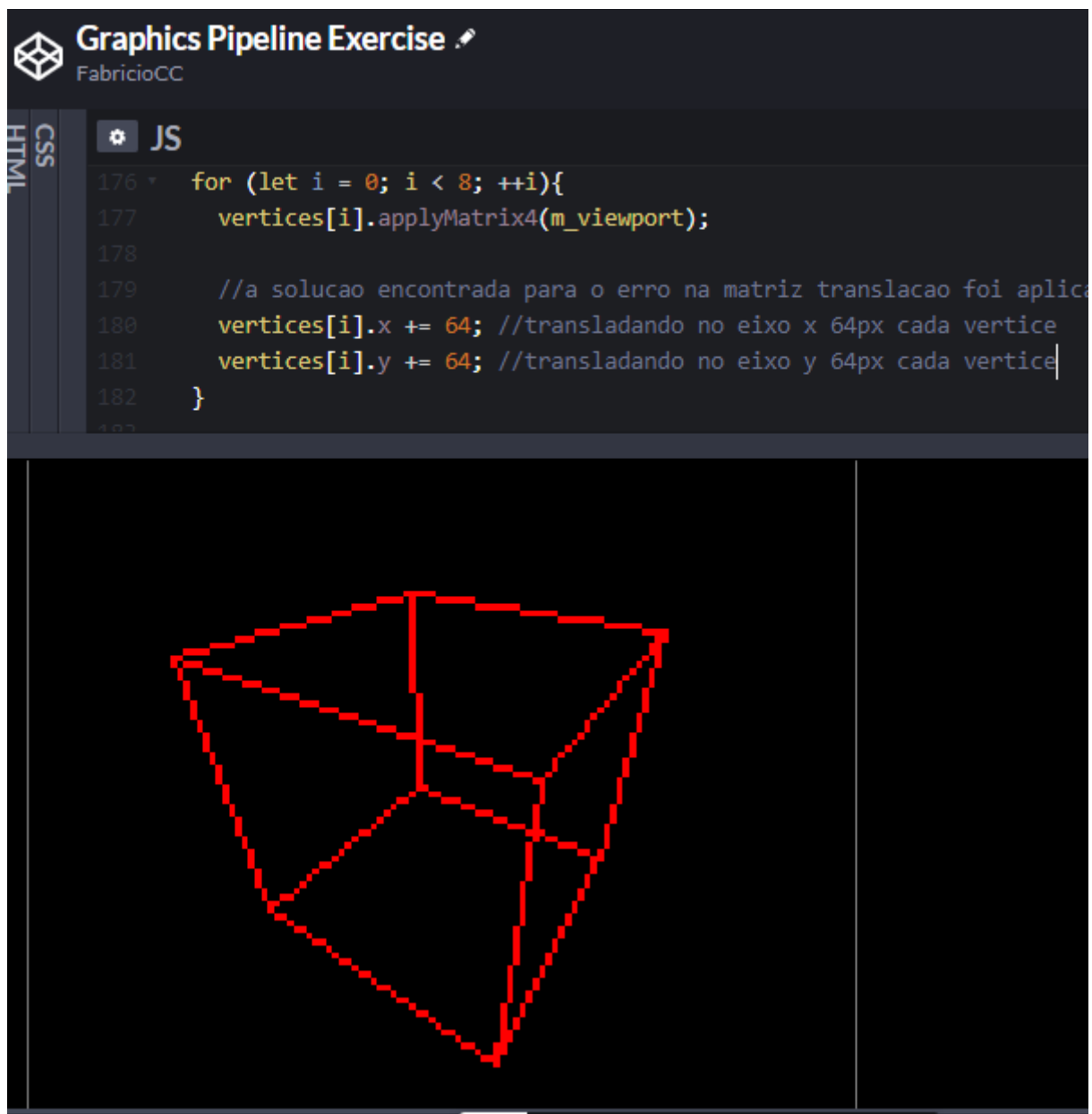


**Figura 2** – Exibição incorreta do cubo e mais uma alteração dos valores na matriz de translação

Essa foi a principal dificuldade que tivemos no desenvolvimento da atividade, mesmo após algumas tentativas de mudança de valores na matriz de translação o problema continuou a persistir e buscamos uma solução válida para tornar a exibição do cubo de forma correta como solicitado no PDF.

## 4. Resultados gerados e soluções:

Após os erros que tivemos no desenvolvimento da atividade, buscamos desenvolver uma solução para demonstrar o cubo da mesma forma como está apresentado no PDF. Primeiramente criamos uma matriz de translação e tentamos aplicá-la, mas mesmo assim o erro persistiu, no entanto, encontramos uma solução adequada para corrigir o erro e demonstrar de forma correta. A solução encontrada para o erro na matriz de translação foi realizar a aplicação da translação em cada vértice separadamente, transladando no eixo X 64 pixels cada vértice e transladando também no eixo Y 64 pixels em cada vértice.



**Figura 3** – Exibição correta do cubo e aplicação da translação em cada vértice do objeto

## 5. Possíveis melhoras:

Uma possível melhoria que poderia ser feita, seria diretamente na matriz de translação, onde está presente o principal erro que tivemos. Modificá-la, para que seja possível realizar a translação do objeto diretamente por ela, sem precisar transladar o objeto em cada vértice, tanto no eixo X, quanto no eixo Y separadamente.

## **6. Referências bibliográficas:**

[https://johannesca.github.io/cg\\_t2/](https://johannesca.github.io/cg_t2/)

<http://wesnydyribeiro.blogspot.com/2017/04/pipeline-grafico.html>

## **7. Link para repositório online CodePen:**

<https://codepen.io/fabricao/cc/pen/dyNrdMX?editors=0010>