

**LUCAS SIMÕES DE ALMEIDA**

**MATRÍCULA - 1712101**

**PROJETO 2**

CENA RENDERIZADA COM PIPELINE PROGRÁMAVEL

Relatório da Disciplina Computação INF1761, Segundo Semestre do ano de 2021.

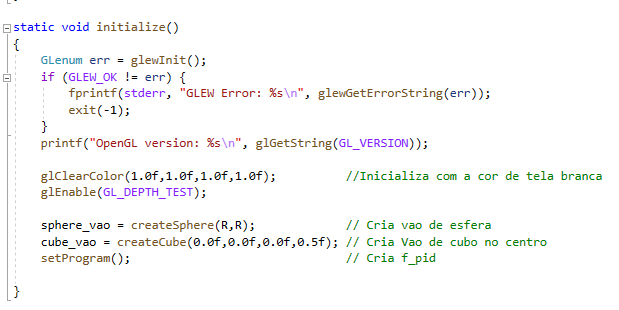
Professor:Waldemar Celes

RIO DE JANEIRO

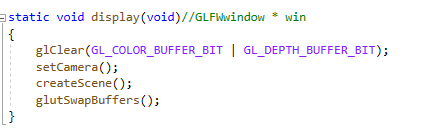
2021

Neste projeto, tivemos que renderizar uma cena utilizando um pipeline programável e através da biblioteca openGL, programar em placa gráfica, esta cena deve ter obrigatoriamente: instâncias de cubos e esferas, iluminação por pixel, e mapeamento de textura de pelos menos 2 objetos da cena.

Meu maior problema inicial eram os shaders, eu criei diversos tipos de shaders para teste, pois acreditava que meus shaders eram o problema principal do código, afinal quando comecei, se quer conseguia gerar um objeto.

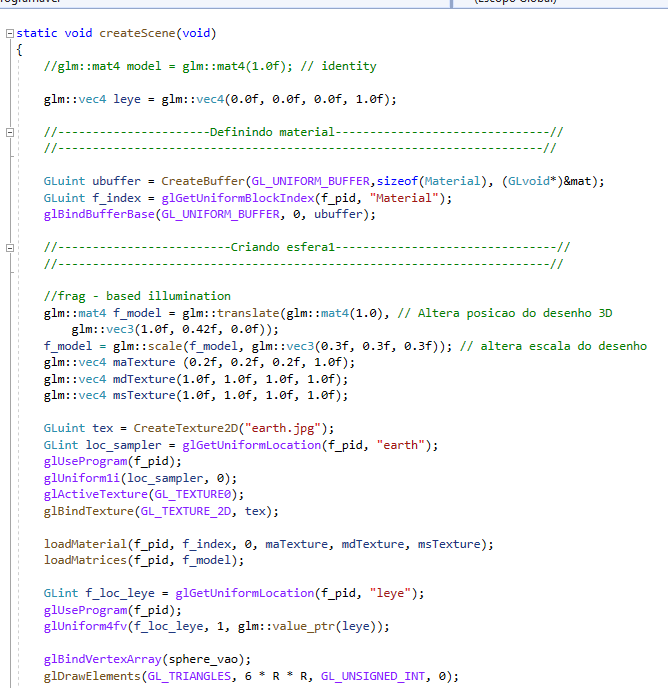
Após corrigir diversos problemas no código fonte, percebi que os shaders estavam na verdade corretos, meu problema era em outro assunto, a manipulação dos vértices e buffers necessários e a interação uniform das variáveis. 

Essa é a função de inicializar, ela cria tudo que será necessário na função display, o programa de shader, os VAO dos objetos utilizados e o mais importante, inicializa a glew.

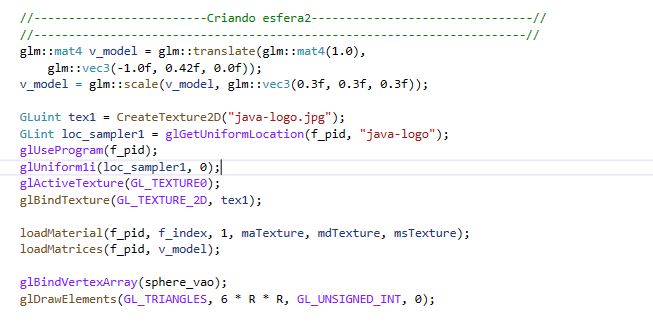


A função de display ficou extremamente curta, afinal ela só chama funções, a definição da câmera,a que cria a cena e a que atualiza a janela em tempo real.

Tive bastante dificuldade para enteder o processo de criação de objeto através do pipeline programável, mas quando finalmente entendi a coisa ficou bem mais natural:



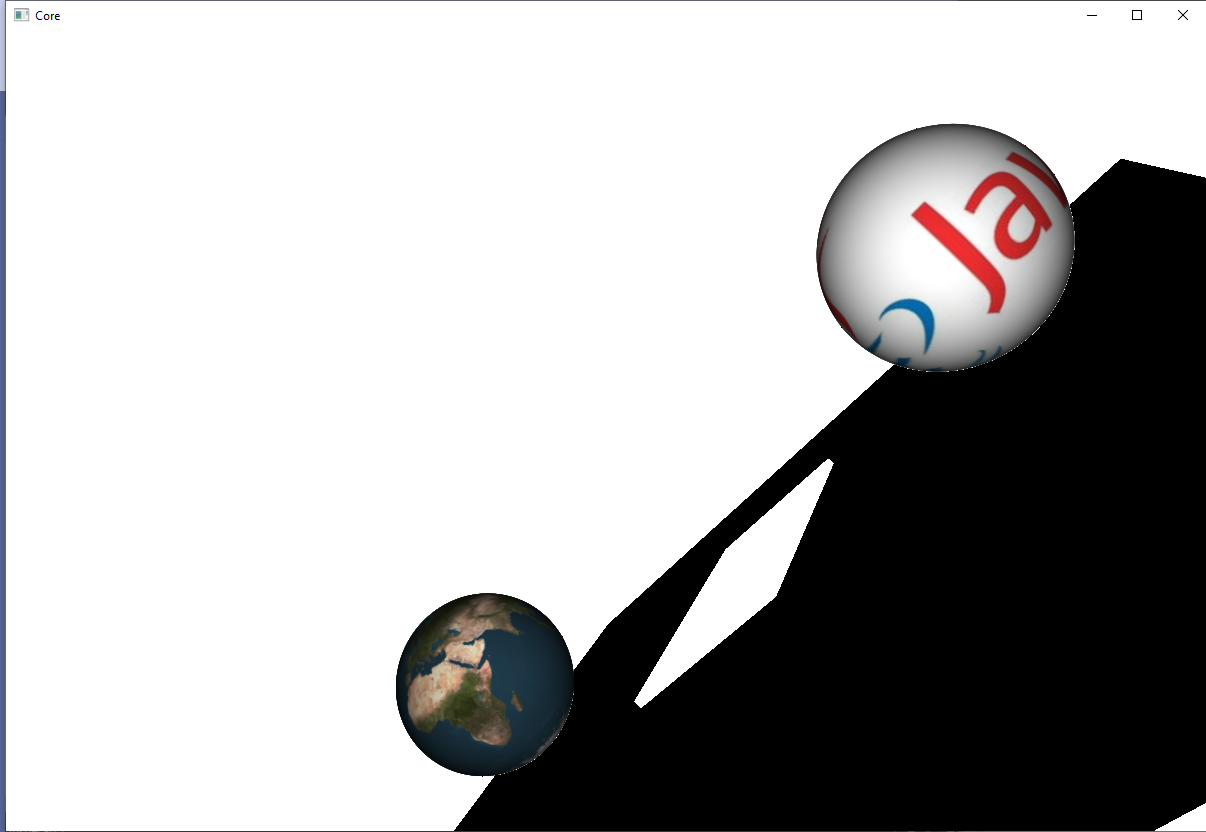
Esse trecho, é o ínicio da create scene, a parte mais importante desse trecho é que utilizo do “leye” para definir a sua posição, mas como só precisamos definir ela uma vez para cada programa de shader utilizado, a criação de objetos depois disso pode ser resumida dessa forma:



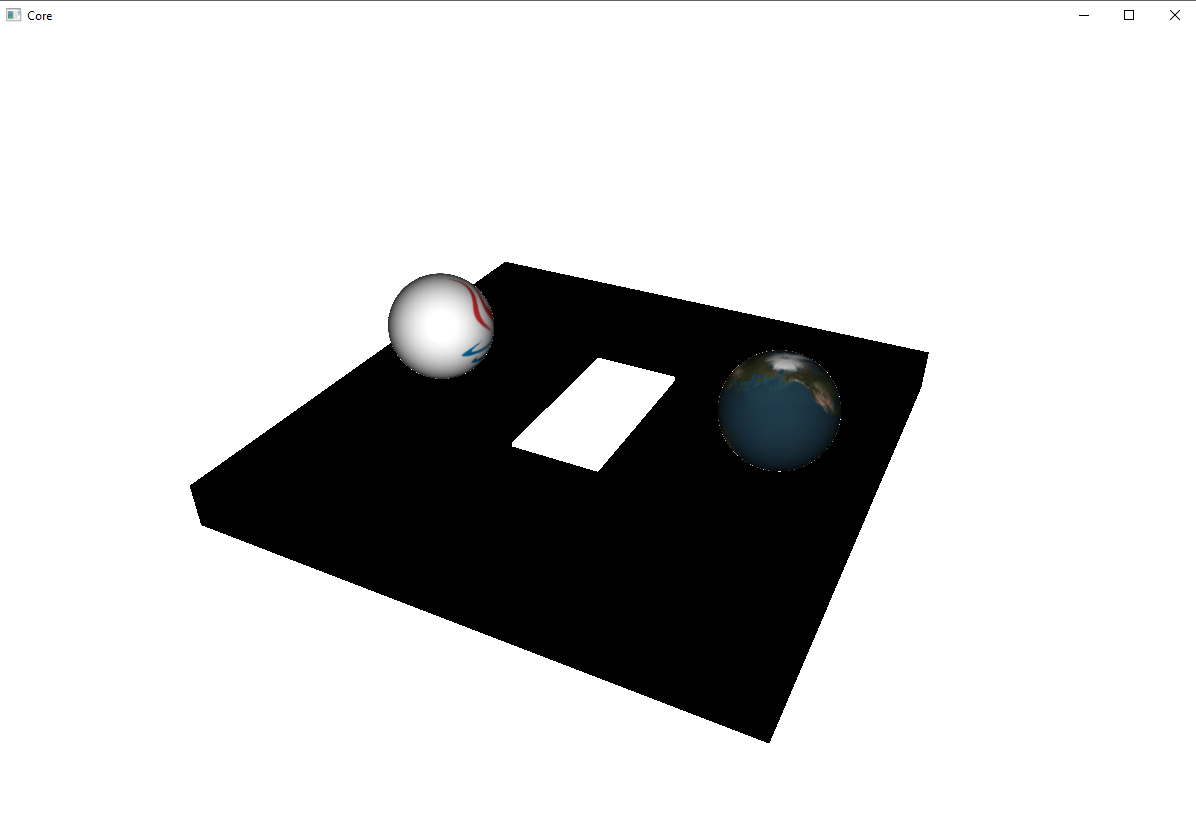
Precisamos sempre bindar após definir essas mudanças glUniform\*, pois o shader precisa saber a qual objeto aquela alteração foi aplicada. Se não utilizarmos do bind, a ultima alteração, modificara todo o projeto.

Podemos ver o resultado final abaixo:

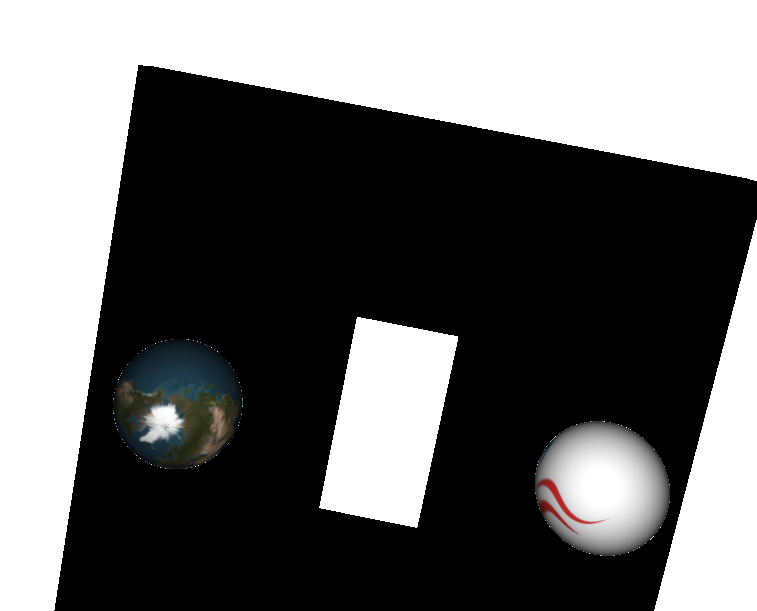
**Imagem 1 -**



**Imagem 2 -**



**Imagem 3 -**



Ficou parecendo uma tomada analisando pela vista superior, graças a implementação do controle de câmera arcball, é possível ver as cenas de qualquer ângulo, utilizei a estrutura da minha implementação do projeto 1 porém a modifiquei para funcionar no novo pipeline.

Sobre as texturas testei diversas imagens, inclusive uma com o meu rosto, e todas deram certo, porém como já estava atrasado com a entrega, não desenvolvi uma maneira de aplicar texturas aos objetos cubo escalonados, ou seja, a mesa e a folha de papel. Mas consegui aplicar às esferas, então utilizei 2 texturas diferentes para ter certeza que estava tudo okay.

Concluindo, o trabalho foi extremamente complexo no início, tive diversas dificuldades, e eu quase desisti, mas o resultado final valeu muito a pena e fico muito feliz que tenha seguido em frente. É bem complicado entender como o pipeline programável funciona, mas as peças acabam se encaixando eventualmente e a coisa fica bem mais natural.

Gostaria de agradecer ao professor **Waldemar Celes**, pela oportunidade e por estar sempre disponível para esclarecer dúvidas pontuais.