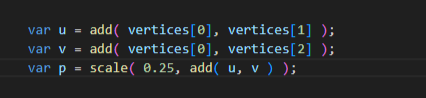
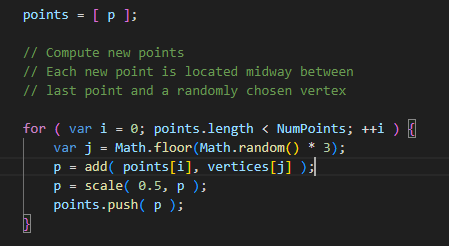
Gasket1:

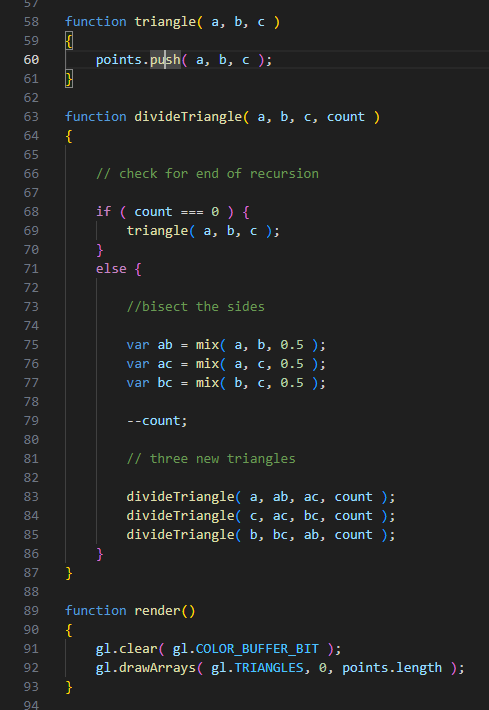


Cria um ponto P inicial entre os vertices 0, 1 e 2.



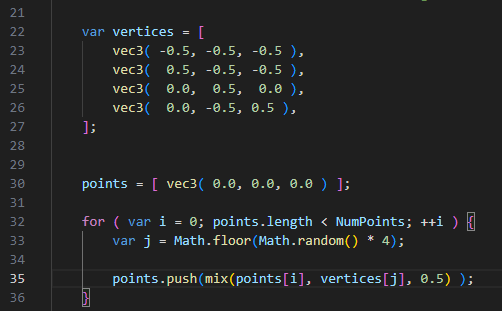
Adiciona o ponto P inicial num vetor, de acordo com o NumPoints dado no início do código (neste exemplo 5000), em cada iteração é escolhido um número aleatório j, no ponto médio entre points[i] e vertices[j] é adicionado um novo ponto ao vetor points, esse processo é repetido até terem 5000 pontos.

Gasket2:



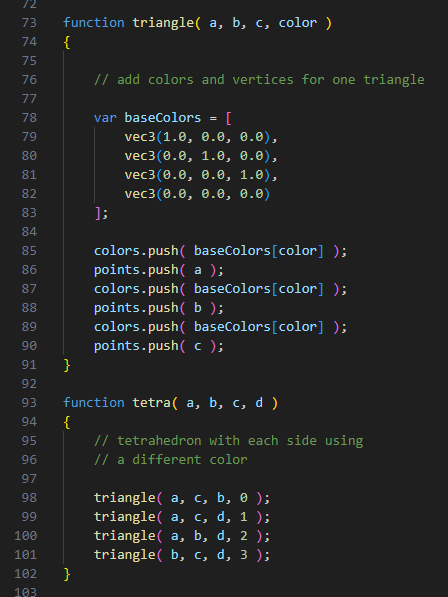
Função principal do código, dado 3 vértices e o número de vezes que eles têm que ser subdivididos, será pego o ponto médio dos 3 lados do triangulo formado pelos vértices, e será repetido o processo para os triângulos formados pelos vértices iniciais e os novos pontos médios até que a variável count seja 0, no final da recursão será adicionado no vetor points os vértices de todos os triângulos a serem desenhados.

Gasket3:



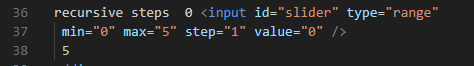
Mesmo raciocínio de Gasket1, só que adicionando 1 dimensão a mais (como Gasket1 era 2d este será 3d), necessitando assim 4 vertices com posições x, y, z para formar uma pirâmide.

Gasket4:

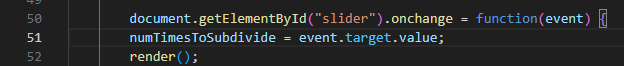


Mesmo raciocínio de Gasket3, só que subdivide tetraedros ao invés de triângulos, pintando cada lado do tetraedro com a função triangle, de vermelho, verde e azul.

Gasket5:



Mesmo raciocínio de Gasket3 porém a quantidade de vezes que o triangulo será subdividido é controlado por um slider presente no arquivo html.



Parte do código em que a variável numTimesToSubdivide recebe o valor atual do slider.