**Tipos de Dados:**

G01-03

**APP.JS**

* **MAX\_LIGHTS** – Numero Maximo de luzes – Agora definido a 3 (definido aqui para nao conseguirmos criar mais do que o numero maximo de luzes)
* **Lights[]** – Vetor com todas as lights e os seus atributos (active – se esta ativa, ambient – luz ambiente, diffuse – difusao da cor depende do material, specular – reflexao especular, position – posiçao da luz( w indica se a luz é direcional), axis – direçao da luz , aperture - abertura, cutoff – atenuador da intensidade da luz)

**FRAGMENT SHADER**

* **MAX\_LIGHTS** – Numero Maximo de luzes – Agora definido a 3 (usado porque “i o i não pode ser comparado com uma expressão não constante num for loop)
* **nLights** – Numero de luzes que efetivamente existem
* **Uniforms** – variaveis para calcular como a luz reflete nos materiais (material ambient, material diffusion, material speculation, shininess), variaveis para as luzes (mode, active, ambient, diffuse, specular, position, axis, aperture, cutoff).

**Ter mais do que 3 luzes**:

Teríamos de alterar o MAX\_LIGHTS para um número maior, tanto no fragment shader como na app.js, o que poderá causar problemas de incompatibilidade. No entanto, por não poder passar uma constante para o fragment shader, nesta versão do webgl, tivemos de fazer assim.

**Implementação de Spotlight:**

**Desafio:**Para o desafio, nós fizemos um eventListener que vai alterando duas variaveis locais (mouseX, mouseY). Depois calculamos a nova posição da camera.eye, camera.at e camera.up. De seguida, se de facto o rato for pressionado e movido, alteramos o lookAt para a mView (view matrix), e fazemos load dessa mesma matrix.

Projeto 3

G01-03

Ricardo Gonçalo(60519) & Rita Barbosa(64925)