

Relatório 3ª Entrega CG

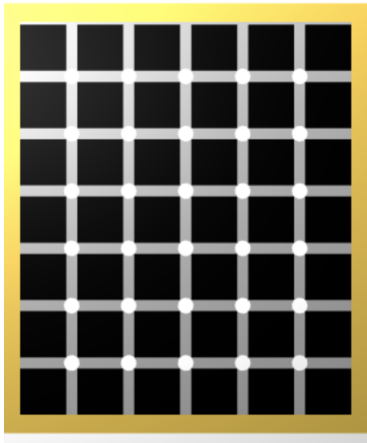
Discente: Henrique Dias, 89455
Docente: Prof. João Brisson
5 de novembro de 2019



O trabalho foi desenvolvido em 8 diferentes classes, maioritariamente estendendo a classe `THREE.Object3D` da biblioteca em uso:

- `Mesh`, estende a classe `THREE.Mesh` e permite a utilização de três tipos de materiais diferentes (`Basic`, `Lambert` e `Pong`) através dos métodos `toggleLight`, que alterna entre `Basic` e `Lambert/Pong`, e `toggleLightMaterial` que alterna entre `Lambert` e `Pong`.
- `Frame`, utilizada para a moldura da imagem, em dourado.
- `Icosahedron`, onde é criado um icosaedro irregular (explicação mais em baixo).
- `Painting`, onde a pintura com a ilusão ótica é criada através de paralelepípedos e cilindros.
- `Pedestal`, utilizado para o pedestal do icosaedro.
- `Room`, que inclui as paredes e o chão.
- `Spotlight`, utilizado para criar as luzes direcionais.
- `Scene`, que permite gerir toda a cena através dos métodos `resize` (as câmaras, quando a janela é redimensionada), `toggleLight` (para desligar ou ligar o cálculo da luz), `toggleLightMaterial` (para alternar entre os dois tipos de material com cálculo de luz - `Lambert` e `Pong`), `toggleGlobalLight` (para desligar e ligar a luz global), `toggleSpotlight(i)` (para desligar a luz direcional `i`).

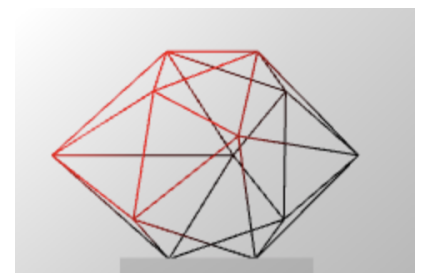
Para o material sem cálculo de luz, utilizei o `MeshBasicMaterial`, enquanto que para sombreamento *diffuse* foi utilizado o `MeshLambertMaterial`. Já para o `Phong`, utilizei o `MeshPongMaterial`. A tecla 'E' alterna entre estes dois últimos tipos de material, enquanto que a tecla 'W' alterna entre o `MeshBasicMaterial` e o último material com luz utilizado.



Para a luz global é utilizada uma `THREE.DirectionalLight`, que é ligada ou desligada através da tecla 'Q'. Qualquer uma das outras luzes (spotlights) podem ser desligados ou ligados através das teclas 1-4. Todos os holofotes possuem a forma de uma esfera e um cone e produzem sombras.

A pintura (`Painting`) é feita através de um fundo cinzento, com paralelepípedos pertos e círculos brancos alinhados de forma a produzir o efeito desejado.

Relativamente ao icosaedro, comecei por utilizar a fórmula dada ($[0, \pm 1, \pm \phi]$) e produzir os 12 vértices necessários. Depois adicionei ou removi aleatoriamente valores entre 0 e 1 de vértices aleatórios de forma a produzir um icosaedro irregular. Finalmente, criei as 20 faces necessárias recorrendo a `THREE.Face3`. Tudo isto poderá ser encontrado na classe `Icosahedron`.



Quando a janela é redimensionada (evento `resize`), ativo uma *flag* que é utilizada na próxima iteração de `requestAnimationFrame`. Caso tenha sido redimensionada, chamo o método `scene.resize`

que efetua os novos cálculos para as câmaras. Da mesma forma faço todas as ações relativamente a outras teclas (evento `keyup`).

Neste terceiro trabalho, decidi criar uma classe para tratar da cena toda (classe `Scene`) de forma a que as chamadas a funções na função `animate` fossem mais concretas e objetivas. Além disso, é uma melhor forma de separar o que a cena é e faz do que está a gerar as alterações. Esta classe trata da criação de toda a cena, desde as paredes, chão, pedestal, pintura, icosaedro, holofotes e luz global.