

## CG 2324 – Sugestões para o Desenvolvimento do Trabalho C

### Sugestões para a Tarefa 1

Em alternativa ao THREE.TubeGeometry podem recorrer ao THREE.CylinderGeometry para modelar o cilindro (como alternativa ao THREE.TubeGeometry).

Cada anel deve ser modelado THREE.ExtrudeGeometry usando para uma THREE.Shape com o formato de anel (como alternativa ao THREE.RingGeometry).

Quanto às teclas '1', '2' e '3', podem equivaler a botões de 'toggle'. Não se esqueçam que a última tarefa tem como colocar o utilizador em Realidade Virtual, mas que não têm acesso às teclas, pelo que sugerimos que a web app arranque sempre com os anéis em movimento, servido as teclas '1', '2' e '3' para parar e recomeçar a sua animação.

### Sugestões para a Tarefas 2

Antes de mapear texturas que reajam à luz, adicionem à cena uma ou mais luzes to tipo pontual, direccional ou spotlight.

O seguinte trecho aplica-se a uma Mesh de um plano mas é facilmente extensível a uma Mesh com a forma de uma esfera:

```
// Realizar o loading de diferentes imagens e convertê-las
// em objectos 'texture'
let map = new THREE.TextureLoader().load("brick.jpg");
let bmap = new THREE.TextureLoader().load("brick-bump.jpg");
let dmap = new
THREE.TextureLoader().load("brick-displacement.jpg");

// Passar 'textures' como parâmetro(s) de um material.
// Não esquecer de mapear o lado certo (FrontSide ou BackSide) da
geometria // ou então ambos (DoubleSide).
let material = new THREE.MeshPhongMaterial({
  bumpMap: bmap,
  bumpScale: 1.3,
  side: THREE.DoubleSide,
  displacementMap: dmap,
  displacementScale: 5,
  map: map
});

// Uma vez na posse de um material texturado, definir a geometria
e a
// respectiva Mesh que deve ser adicionada à cena
```

```
let geometry = new THREE.PlaneGeometry(400,450,50,50);
let brickWall = new THREE.Mesh(geometry,material);
scene.add(brickWall);
```

Sources:

- <https://redstapler.co/three-js-displacement-map-from-single-image-tutorial/>
- <https://www.youtube.com/watch?v=u-7SNWeZKk8>

## Sugestões para a Tarefa 4

Para ligar ou desligar luzes, alterem o estado do booleano “.visible” do objecto luz em questão:

```
switch (e.keyCode) {
  case 68: //D
  case 100: //d
    directional_light.visible = !directional_light.visible
    ambient_light.visible = hemisphere_light.visible = true
    break;
  ...
}
```

## Sugestões para a Tarefa 5

Definir o objecto como BufferGeometry ao qual têm de explicitar os vértices e as faces; certifiquem que triangulam todos os lados da faixa. A faixa deverá ter pelo menos 1 torção. Os vértices não devem ser calculados com base numa expressão matemática (e.g., expressão paramétrica). Com base em 2 ou mais vistas de uma faixa (e.g., vistas de tipo, lateral e de baixo), devem assinalar vértices, medir as coordenadas e introduzir esses dados como array de vértices no BufferGeometry.

```
const geometry = new THREE.BufferGeometry();
// listar vértices (vectores 3D com as coordenadas de cada vértice)
const vertices = new Float32Array(
  [ -1.0, -1.0, 1.0, // v0
    1.0, -1.0, 1.0, // v1
    1.0, 1.0, 1.0, // v2
    -1.0, 1.0, 1.0, // v3 ] );
geometry.setAttribute( 'position', new
THREE.BufferAttribute(vertices, 3) );
// listar tripletos de índices por forma a definir cada
face/triângulo
// notem que a sequência de índices deve indicar o sentido da
normal
const indices = [ 0, 1, 2, 2, 3, 0 ];
geometry.setIndex( indices );
```

```
// não esquecer de calcular as normais de cada face
geometry.computeVertexNormals();

// uma vez na posse de uma geometria, definir um material e criar
uma Mesh
const material = new THREE.MeshBasicMaterial( { color: 0xff0000 }
);
const mesh = new THREE.Mesh( geometry, material );
```

## Sugestões para a Tarefa 8

Por forma a certificar que o VRButton está a funcionar devidamente, instalar a WebXR API Emulator no vosso browser (recomenda-se o Chrome)

- <https://chrome.google.com/webstore/detail/webxr-api-emulator/mjddjgeghkdijejnciaefnkjmkafnnje>

Uma vez instalado o WebXR API Emulator, seguir as instruções contidas neste vídeo por forma a aceder ao emulador dentro do browser:

- <https://www.youtube.com/watch?v=Twenzp-LEMkU>

Seguir a documentação oficial em como criar conteúdo VR para uma aplicação web (How to create VR content)

- <https://threejs.org/docs/#manual/en/introduction/How-to-create-VR-content>

```
// Dentro do init(), fazer o "enable" do modo xr do renderer e
// adicionar o elemento VRButton ao documento DOM
function init() {
  'use strict';

  renderer = new THREE.WebGLRenderer({ antialias: true });
  renderer.setPixelRatio(window.devicePixelRatio);
  renderer.setSize(window.innerWidth, window.innerHeight);
  document.body.appendChild(renderer.domElement);

  renderer.xr.enabled = true;
  document.body.appendChild(VRButton.createButton(renderer));
  ...

  // Dentro do animate(), ajustar o requestAnimationFrame para
  setAnimationLoop
  function animate(){
    "use strict";

    update();
    render();
```

```
// requestAnimationFrame(animate);  
renderer.setAnimationLoop(animate);  
...
```