



universidade
de aveiro

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Antiques Showroom

Three.js – 1º Projeto

Introdução à Computação Gráfica - 2020/2021

Joaquim Madeira (jmadeira@ua.pt)

93310 - Gonçalo Pereira

Índice

Introdução	3
Correr o Programa	3
Modelos e Texturas	4
Movimentação	4
Câmera	4
Iluminação	5
Resumo das Implementações	5
Conclusão	8

Introdução

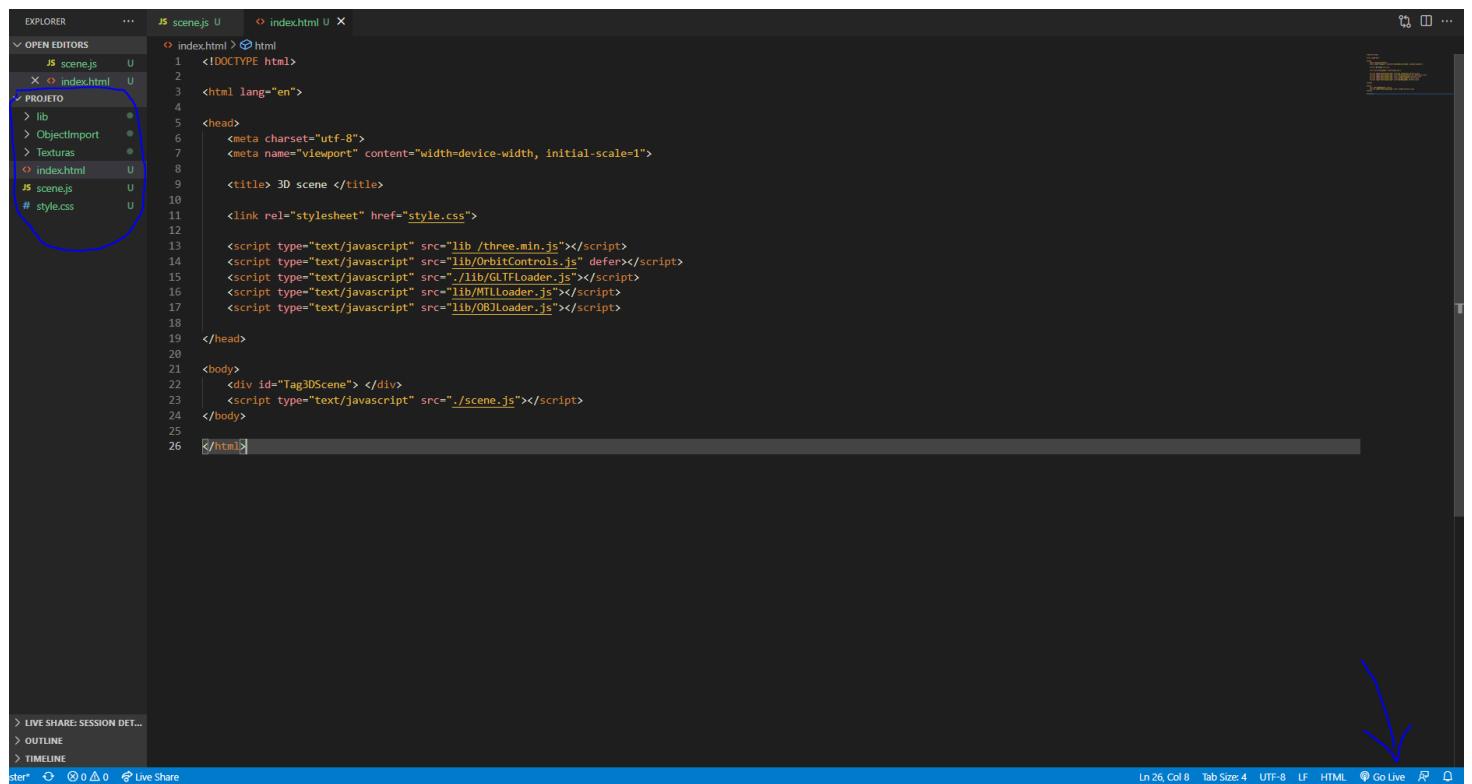
Neste projeto decidi criar uma showroom de objetos antigos, como se fosse um museu de antiguidades.

O personagem pode percorrer a sala para ver cada modelo de perto, andar mais rápido e fazer zoom.

Neste relatório vou descrever as minhas opções durante o trabalho e os aspetos principais da aplicação.

Correr o Programa

Para correr o programa é necessário abrir a pasta num editor de código para as pastas necessárias carregarem, depois abrir o projeto como um todo, e não apenas um ficheiro.



The screenshot shows a code editor interface with the following details:

- EXPLORER** panel on the left showing the project structure:
 - OPEN EDITORS: scene.js, index.html
 - PROJETO: lib, ObjectImport, Texturas, indexHTML, scene.js, style.cssA blue oval highlights the PROJETO section.
- index.html** tab is active, displaying the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title> 3D scene </title>
    <link rel="stylesheet" href="style.css">
    <script type="text/javascript" src="lib/three.min.js"></script>
    <script type="text/javascript" src="lib/OrbitControls.js" defer></script>
    <script type="text/javascript" src="_/lib/GLTFLoader.js"></script>
    <script type="text/javascript" src="lib/MTLLoader.js"></script>
    <script type="text/javascript" src="lib/OBJLoader.js"></script>
</head>
<body>
    <div id="Tag3DScene"> </div>
    <script type="text/javascript" src="./scene.js"></script>
</body>
</html>
```

A blue arrow points from the bottom right towards the status bar at the bottom of the editor.

Bottom status bar: LIVE SHARE: SESSION DET... > OUTLINE > TIMELINE. Bottom right: Line 26, Column 8, Tab Size: 4, UTF-8, LF, HTML, Go Live, Refresh, Close.

Modelos e Texturas

Para criar um ambiente realista de museu/showroom, decidi usar modelos de objetos antigos, por exemplo tanques de óleo e um cofre antigo. Usei também texturas para o chão, teto e paredes da sala para a sala ter a personalidade adequada ao meu objetivo.

Os modelos encontram-se na pasta ObjectImport, as texturas estão na pasta Texturas.

Importei os modelos do site *Sketchfab* e as texturas são imagens retiradas da internet.

Movimentação

O movimento da personagem é controlado com as teclas 'WASD' e o movimento da câmara é efetuado com as setas do teclado.

A showroom é feita em primeira pessoa, e os objetos da sala têm tamanho proporcional ao personagem para criar imersão e realismo.

Existem também *hitboxes* que não permitem ao personagem atravessar os modelos ou as paredes.

As *hitboxes* são aplicadas através de uma caixa, *cameraBox*, que acompanha a câmera do jogador e verifica as colisões com os restantes modelos e paredes. Resumidamente, quando a *cameraBox* atinge um objeto a posição do jogador volta à anterior e aparece uma mensagem na consola.

Câmera

Como disse, a câmera é controlada com as setas do teclado.

O utilizador pode olhar à sua volta, para cima e para baixo, havendo limite na rotação vertical, de modo a não dar a volta.

Illuminação

Para reforçar o efeito de antiguidade, não usei muita iluminação propositadamente. Assim, os modelos enquadram-se melhor no tema e a experiência da visita é mais adequada.

Apesar disso, a sala tem luz suficiente para analisar os pormenores de cada modelo, criando equilíbrio com a escuridão.

Resumo das Implementações

Nesta última secção do relatório vou tentar resumir os itens da showroom, falando um pouco de cada implementação minha e dos motivos das mesmas.

Pretendo também mostrar partes do código que considero relevantes.

Na showroom é possível ver algumas máquinas antigas, que considerei fazerem sentido tendo em conta o meu objetivo para este trabalho. Também coloquei mesas com aspecto desgastado, que podem ser consideradas peças históricas e um cofre com ar deteriorado.





Os controlos da câmera são realizados na função animate(), através das seguintes linhas (exemplo para 'W' -> frente):

```
if (keyboard[87]) { // W key
    camera.position.x -= Math.sin(camera.rotation.y) * player.speed;
    camera.position.z += -Math.cos(camera.rotation.y) * player.speed;

    cameraBox.position.x -= Math.sin(camera.rotation.y) * player.speed;
    cameraBox.position.z += -Math.cos(camera.rotation.y) * player.speed;

    if (checkCollisions()) {
        cameraBox.position.x = oldPos.x
        cameraBox.position.z = oldPos.z

        camera.position.x = oldPos.x
        camera.position.z = oldPos.z
    }

    if (checkCollisionDoorWall()) {
        cameraBox.position.x = oldPos.x
        cameraBox.position.z = oldPos.z

        camera.position.x = oldPos.x
        camera.position.z = oldPos.z
    }

    if (checkCollisionsObjects()) {
        cameraBox.position.x = oldPos.x
        cameraBox.position.z = oldPos.z

        camera.position.x = oldPos.x
        camera.position.z = oldPos.z
    }
}
```

Neste projeto a parte que deu mais trabalho e problemas a implementar foi sem dúvida a câmera, precisei de bastante ajuda para acertar com os movimentos e rotações mas compensou pois ficou uma *feature* bem implementada.

Conclusão

Considero que com este trabalho consegui aplicar os conhecimentos adquiridos em Introdução à Computação Gráfica e tive a oportunidade de aprender novos conceitos que pude usar para a realização do próximo projeto.

Realizei o trabalho sem muitas limitações, apesar de ter tido dificuldades que atrasaram a realização de algumas funcionalidades que gostaria de implementar.

Repositório git

O código do projeto, relatório e apresentação encontram-se no meu repositório git de ICM:

https://github.com/pereira-goncalo/93310_ICG/tree/master/Projeto

