

Realizado por:

Alexandre Costa, 1211225

José Teixeira, 1210965

Rafael Santos, 1221267

Rodrigo Almeida, 1220776

Professor: João Paulo Jorge Pereira

Este documento contém uma breve descrição da tarefa que foca na modelagem/importação de modelos 3D, visualização, interação com o usuário, animação/lógica, iluminação e mapeamento de textura, a identificação das tecnologias utilizadas, a interação adotada (identificação das teclas utilizadas e das ações do mouse, se aplicável) e quaisquer outras peculiaridades consideradas dignas de referência. Para falar da breve descrição foi separado pelas user stories:

Primeiro Sprint:

- **6.5.1** – Nesta us, foi foram criadas várias formas geométricas para gerar um modelo de um andar de um hospital. Foi criado 4 planos para serem pelo método *createGround()* com uma textura de um chão de um hospital. Depois num ficheiro JSON onde foram dispostas todas as todas configurações das paredes *createWall()* com uma textura apropriada, onde foram colocadas quatro salas em cada lado com um corredor no meio. Foram criados outros dois ficheiros JSON para conter as informações das portas, mesas de cirurgias e modelos de corpo humano nos métodos *loadModel()* e *loadModelWithRotation()* que apenas diferem no parâmetro extra para rodar o modelo.
A câmara foi definida com coordenadas para se ajustar ao andar para que ao inicializar a interação ela se fique ajustada numa perspetiva real para facilitar o utilizador.
- **6.5.2** - Todos os modelos e texturas usados foram colocadas numa pasta "models" e "textures", onde dentro dos métodos que geram cada forma geométrica foram colocadas as texturas apropriadas que posteriormente nos ficheiros JSON estão as imagens de cada parte específica para cada figura geométrica criada pelo THREE.js.
- **6.5.3** - Para criar uma luz ambiente e uma luz direcionada foi adicionada uma *THREE.AmbientLight* e uma *THREE.DirectionalLight* preenchidos com campos/parâmetros realistas a um ambiente hospitalar.
- **6.5.4** – Nesta user story foram definidos vários botões com determinadas funcionalidades usando o OrbitControls para que o Botão direito: órbita e o Roda: Zoom ou dolly e o Botão esquerdo ficou para ser definido na última entrega:
"LEFT: null,
RIGHT: THREE.MOUSE.ROTATE,
MIDDLE: THREE.MOUSE.DOLLY".

Segundo Sprint:

- **7.5.1** – Neste segundo Sprint foi definido o botão do lado esquerdo para que ao clicar no botão esquerdo na cama, a câmara movia-se para o centro da sala e para isso foram definidas figuras geométricas com opacidade de 0, “*opacity: 0*” dentro das mesas de cirurgias para que ao clicar na mesa de cirurgia, na realidade está a clicar numa box invisível para que se mova para as posições definidas nas “*targetPositions*” que são as posições do centro da sala para cada mesa de cirurgia, onde foi criado um “*window.addEventListener*” para esse click no botão esquerdo.
- **7.5.2** – Para esta user storie foi outro “*window.addEventListener*” para que ao colocar no carácter “i”, em uma determinada sala, seja mostrada uma breve informação sobre a sala em especifica. As informações de cada sala estão descritas na constante “*roomData*”. Exemplo para a sala A: “*{name: "Room A", description: "This is room A, a spacious area with modern amenities."}*”. No ficheiro HospitalFloor.html foi adicionado um container (<div>) para identificar onde vai ser disposta toda a informação sobre uma determinada sala.
- **7.5.3** – Não era para fazer porque o nosso grupo é de quatro elementos.
- **7.5.4** – Nesta user story era necessário criar uma animação para que ao clicar com o botão esquerdo em uma sala, a câmara em vez de se mover de forma instantânea, existe uma animação suave para realizar essa transição entre salas.
Para realizar a animação foi escrito o método “*updateCameraPosition()*” onde ele usa o “*camera.position.lerpVectors()*” que recebe como parâmetro a posição inicial (“*cameraStartPosition*”), a posição da sala para onde se deve deslocar (“*targetCameraPosition*”) e por fim também recebe a uma variável para ir garantindo a cada momento a situação em que está (“*progress*”), onde o “*controls.target.lerp()*” que recebe dois parâmetros para ir atualizando a câmara para que ela também olhe para o ponto que se irá direcionar, para tornar a transição mais suave.

Por fim é de realçar que as pastas do projeto bem como todos os ficheiros usados foram dispostos da seguinte forma:

- Existem dentro do projeto do Hospital, duas pastas contendo uma, as texturas usadas para definir paredes, e chão do piso do hospital, e outra contendo todos os modelos usados, como por exemplo, as portas, mesas de cirurgias e modelos de corpo humano. Depois no restante projeto existem ficheiros JSON para alocar a informação de uma forma mais organizada, um ficheiro JavaScript chamado main.js onde continha o código principal e vários ficheiros também JavaScript que têm métodos como por exemplo a geração das paredes (como sendo uma *THREE.BoxGeometry*) para simplificar tanto a leitura como a interpretação e continuação do código nos seguintes Sprints. E por fim um ficheiro HTML (HospitalFloor.html) que chama o ficheiro main.js para realizar a demonstração da visualização o piso do hospital.