



## **Relatório de Sistemas Gráficos e Interação**

### **Sprint C**

#### **3DI \_ Grupo 52**

1211299 – André Barros  
1211277 – Carlos Lopes  
1211285 – Ricardo Moreira  
1211289 – Tomás Lopes  
1211288 - Tomás Russo

**Data: 02/01/2024**

## Trabalho Desenvolvido

O trabalho desenvolvido consiste na simulação gráfica de um campus de uma instituição de ensino na qual a plataforma RobDroneGO atua. O campus é um conjunto de edifício, que por sua vez é um conjunto de pisos. Cada piso possui corredores e salas ou gabinetes. Para navegar entre os edifícios são usados elevadores e corredores que ligam os pisos dos vários edifícios entre si, permitindo assim ao robot fazer a transição entre os mesmos.

Quem conduz a simulação pode controlar um robot que se destina à realização de tarefas requisitadas pelos vários utilizadores do sistema.

## Tecnologias Utilizadas

A solução está embebida na restante plataforma desenvolvida em **React** usando **Typescript**. Para a simulação 3D foi usada a biblioteca **ThreeJS** que nos permite manipular elementos gráficos através de Javascript (neste caso Typescript).

Quanto aos elementos do **ThreeJS** usados:

- **PlaneGeometry** – Paredes
- **BoxGeometry** – Chão
- **Directional Light** – Luz global
- **SpotLight** – Lanterna do robô
- **TextureLoader** – Carregar as texturas das paredes e chão
- **GLTFLoader** – Carregar os modelos 3D (portas, elevador, *player*, etc)
- **AudioLoader** – Carregar os efeitos sonoros

## Interações Adotadas

Inicialmente, são apresentadas 4 câmaras em simultâneo ao utilizador, através de um **menu lateral** é possível escolher quais das câmaras são visíveis a qualquer momento (esta escolha é persistida entre sessões). Através do **menu superior** é possível fazer **ajustes manuais** às câmaras, manipulando valores como *offset*, *zoom*, *height*, *fog* etc...

É possível manipular a câmara com que se visualiza o robot através de ações com o **rato**, como por exemplo, ao efetuar **scroll** é aplicado “**zoom**” à medida que é efetuado. Além disso, é possível mover a posição do *viewport* arrastando o rato com o **botão direito** e rodar a câmara arrastando o **botão esquerdo**.

Já no **teclado** é controlado o *robot*, utilizando as teclas as **4 setas do teclado**, seguindo a direção das mesmas (cima, baixo, esquerda direita). Na tecla **F** é possível ligar/desligar a lanterna do robô, sendo assim, ao pressionar a tecla é apresentada uma nova fonte de luz com origem no jogador e com alcance reduzido. Também é possível efetuar o *cycle* entre as câmaras disponíveis usando a tecla **V**. Ao pressionar é alterada a câmara atual em **fullscreen**. Usando o **menu superior**, é possível efetuar o **reset** do *viewport*, voltando assim ao formato original (4 câmaras).

Além disso, ao movimentar o rato pelo *viewport* é possível exibir uma **tip flutuante** que identifica o objeto para o qual estamos a apontar (entrada de salas/gabinetes ou elevador). É também apresentado no canto superior esquerdo o piso atual do edifício em que nos encontramos.

### Solução Desenvolvida

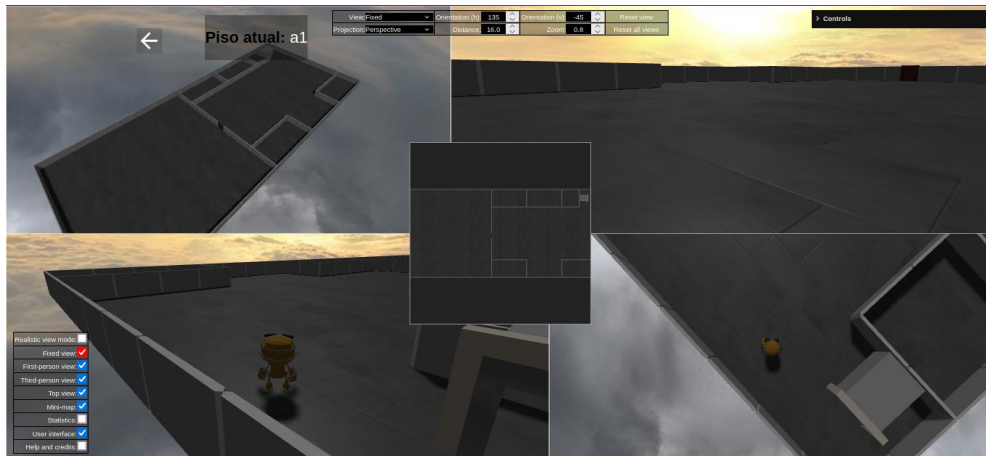


Figura 1 - Viewports base



Figura 2 - Elevador e tip flutuante



Figura 1 - Porta e tip flutuante