

# BeachDefense

João Alcatrão

<https://github.com/Alcatrao/BeachDefense>

<https://alcatrao.github.io/BeachDefense/>



## • Objetivos

O objetivo do trabalho é construir um cenário de praia, com dunas, areia plana, mar (com ondas, ou pelo menos ondulação), céu e sol semi-realistas, e com alguns personagens simples, e permitir ao utilizador interagir com este ambiente sob a forma de um jogo.

Para o jogo em si, pretende-se fazer um first-person shooter (no cenário de praia), com o jogador a ter que defender a praia antes de criaturas inimigas (que surgem do mar) chegarem às dunas. O jogo seria de duração indeterminada, acabando quando algum inimigo 1) matar o jogador ou 2) chegar às dunas, sendo a pontuação do jogador proporcional ao tempo que conseguir defender a praia antes de alguma destas condições se verificar.

## • O que já está feito

A versão atual do projeto consiste num cenário simples de praia, mar e céu, sendo que a única interatividade provém do movimento do rato (para mover a câmara), e das teclas de movimento W,A,S,D e SPACE (para movimentos horizontais e para saltar).

Mais detalhadamente, o mapa está dividido a metade por dois planos (que na verdade estão sobrepostos): o plano do mar e o plano da praia (que se sobrepõe ao do mar). Além disso, o objeto do céu incorpora todo este mapa.

O céu foi feito com recurso a um objeto do tipo 'Sky' (que pertence à biblioteca ThreeJS, mas que tem que ser importada separadamente), e que incorpora um Vector3 para representar a posição do sol. O objeto do céu foi configurado para simular um céu realista, com parâmetros de turbidez, dispersão Rayleigh e Mie ajustados de maneira a criar um céu limpo, claro, com um sol muito brilhante (o vetor 3d não simula luz, simula apenas a posição do sol no céu, e o shader próprio do objeto 'Sky' ajusta o gradiente de cores e o brilho posicional no céu usando as coordenadas desse vetor)

Também existem outras duas fontes de luz, uma geral que ilumina todos os objetos, e outra direcional, a representar a luz do sol, sincronizada com o vetor 3d que o representa no céu, e esta sim é que cria as sombras e reflexos associados ao sol.

O mar foi criado com o objeto 'Water', também da biblioteca Three.js (e que também teve de ser importado separadamente), e aqui utiliza-se um normal map ('waternormals.jpg') para simular a ondulação e a distorção da luz. A superfície da água é dinâmica e reflete tanto o céu como a luz solar.

A praia em si é um Mesh que consiste de um plano horizontal adicionado por cima da superfície do mar, e usa uma texture ('sand\_1.avif') para simular a areia da praia.

A personagem que o jogador controla é um simples Object3D com uma câmara agarrada a ele (a câmara que se usa para ver o cenário), e portanto quando o personagem se mexe com as teclas de movimento, a câmara também mexe.

Além disso, é usado o 'PointerLockControls' (componente ThreeJS que também precisa de ser importada separadamente) para controlar a câmara (ou mais corretamente, o personagem).

## • O que falta fazer

Um dos objetivos mencionados acima é fazer com que a areia e o mar sejam realistas. No momento, a areia tem uma textura que pode ser melhorada (foi usada uma foto de areia da praia para servir como textura), e o mar, embora aparente ter ondulação, na verdade é um plano estático, que se pretende alterar para um plano dinâmico, com flutuações ao longo do tempo, para simular um mar realista.

Além disso, não existe ainda nenhum NPC inimigo, nem nenhuma opção do jogador poder atacar estes personagens, que é o que terá de ser feito para a próxima entrega.

- Sources

- Three.js: <https://threejs.org/>
- Módulos 'Sky', 'Water' incluídos no repositório oficial.
- Mapa normal da água 'waternormals.jpg':  
<https://github.com/mrdoob/three.js/blob/dev/examples/textures/waternormals.jpg>
- Textura da areia: retirada do Google Images.