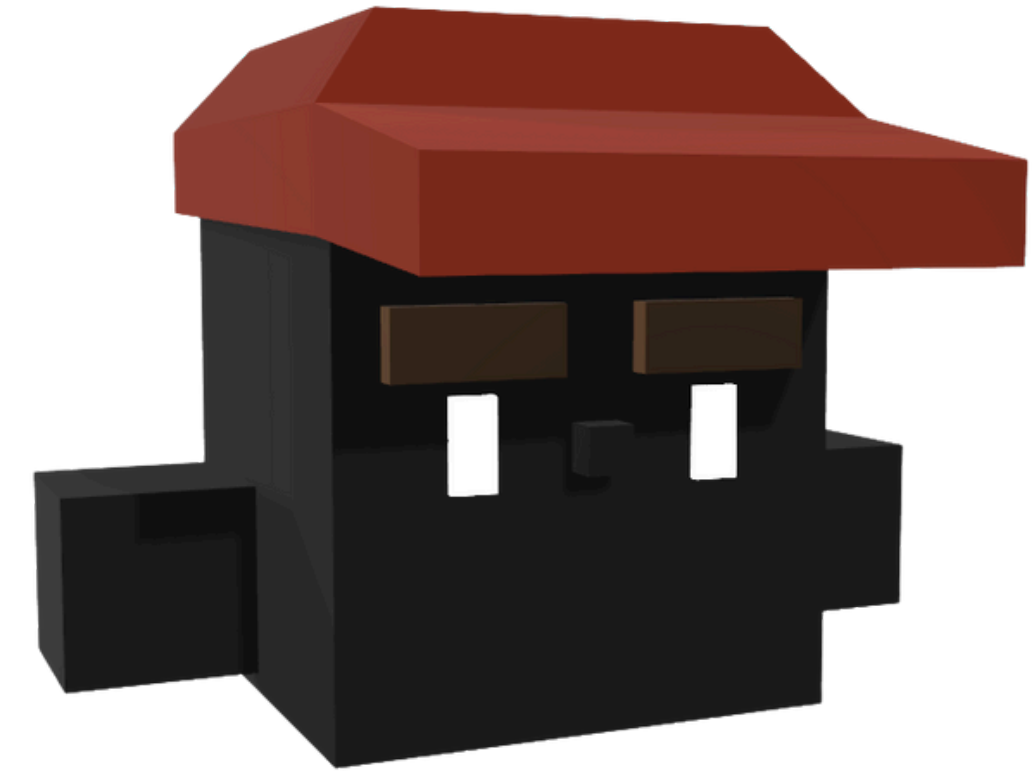


Cube Runner



Introdução à Computação Gráfica
DETI, Universidade de Aveiro

Tomás Fonseca 107245 LECI

Visão Geral

- **Sobre o Projeto**

- Endless runner
- Objetivo: desviar de cubos inimigos
- Dificuldade aumenta com o tempo

- **Biblioteca principal:** Three.js

- Responsável pela renderização 3D, cena, câmera, luzes e objetos.

- **GLTFLoader**

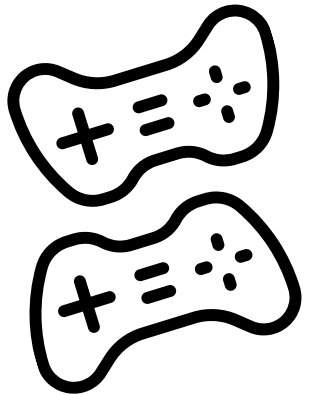
- Utilizado para carregar modelos 3D no formato .glb

- **O que o jogador pode fazer**

- 1 ou 2 jogadores (multiplayer local)
- Acumular pontuação individual
- Consultar ranking global com as 10 melhores pontuações

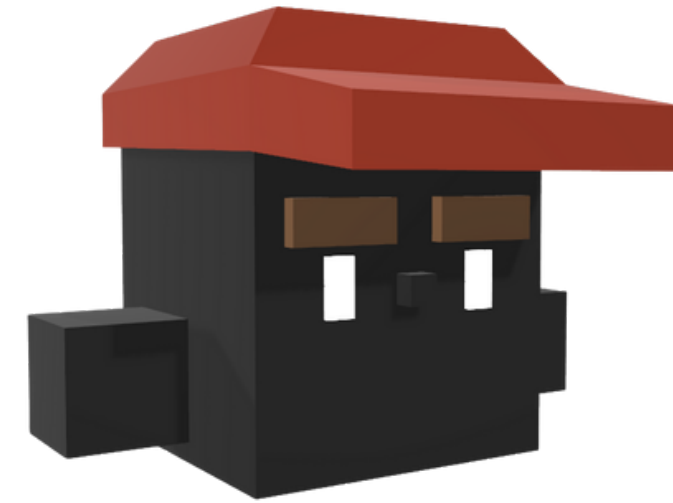
- **URL**

- <https://cuberunner.pt/>



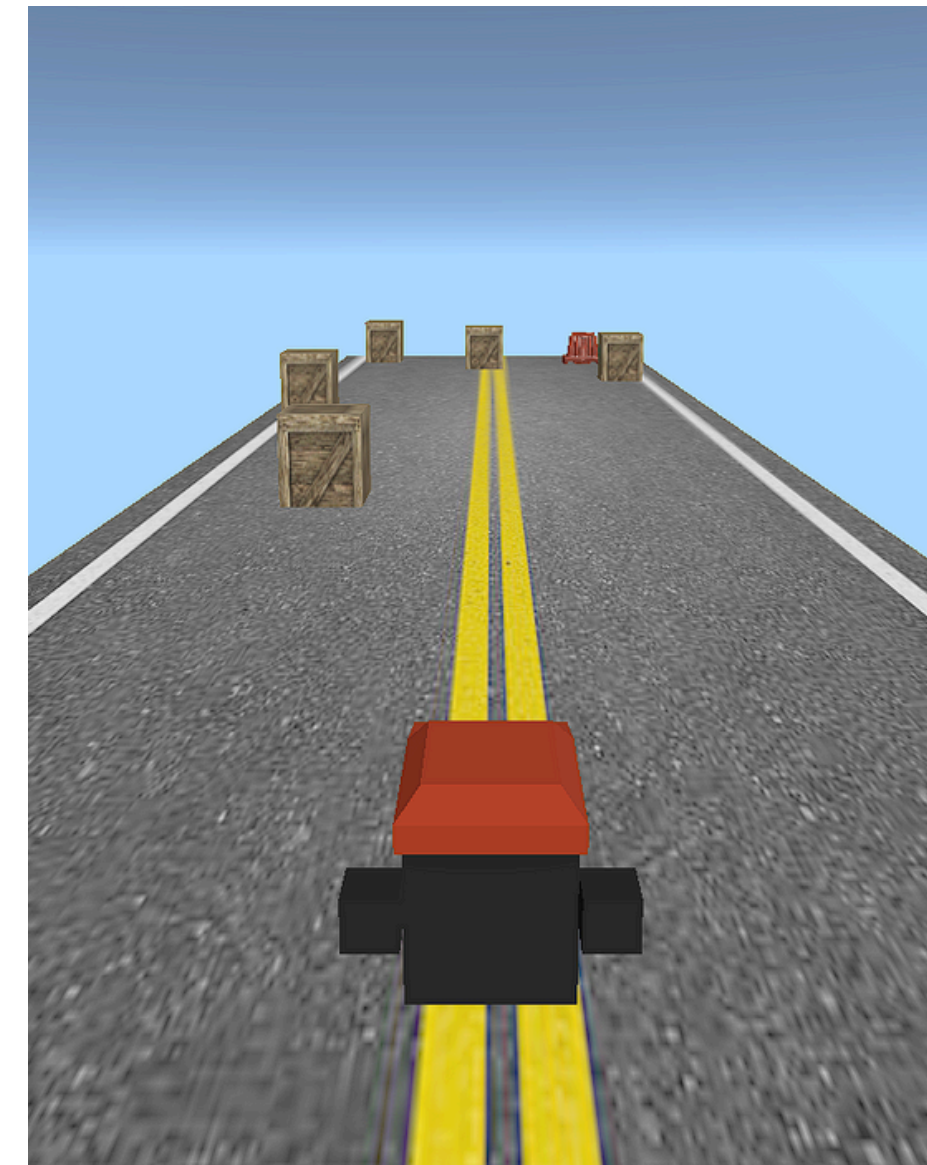
Modelos

- **Modelos 3D utilizados:**
 - hat_cube.glb (Player 1 e Player 2)
 - obstacleX.glb (obstáculos)
- Cada jogador tem um modelo 3D e uma caixa invisível para a física do jogo (colisões)



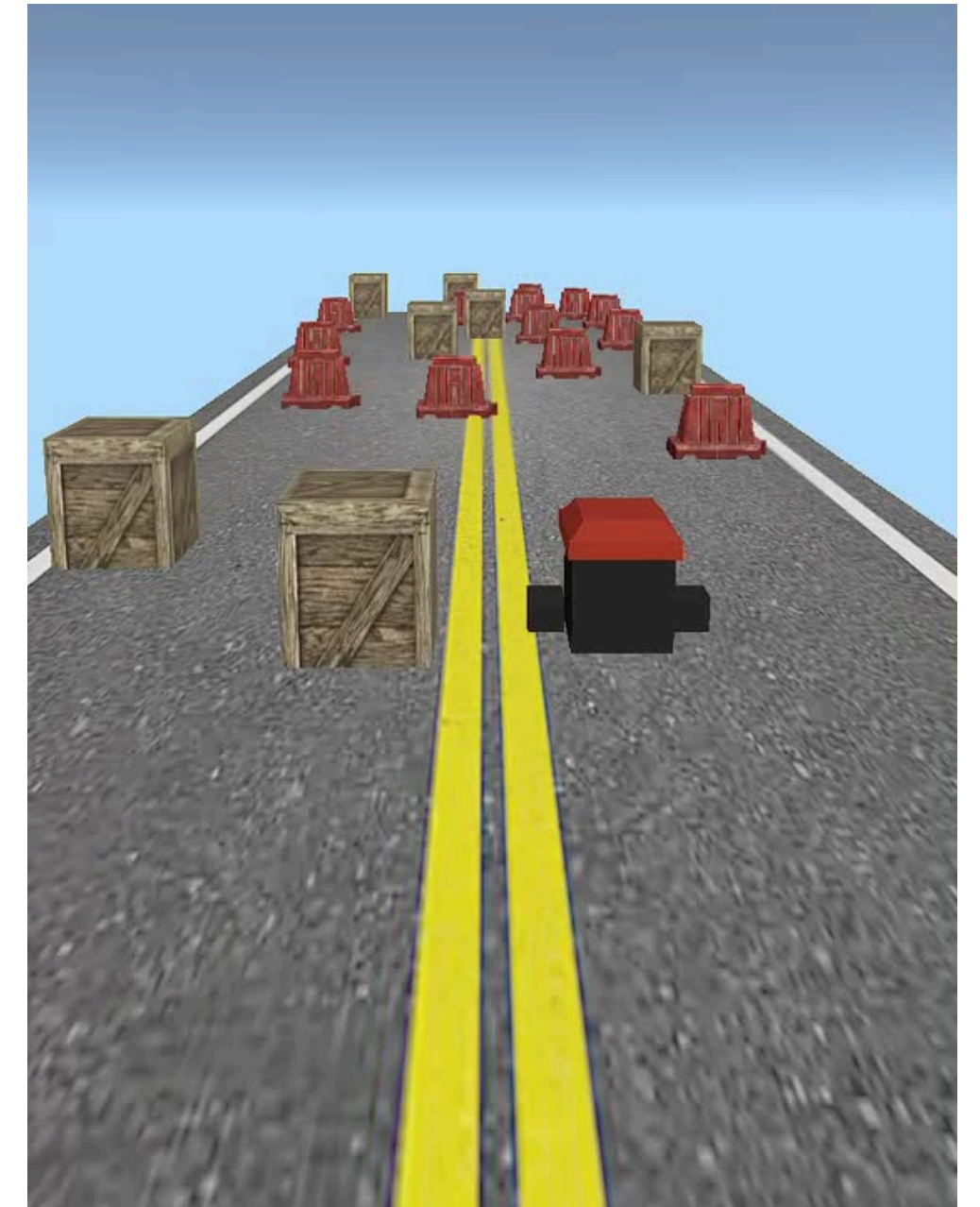
Organização da Cena

- Cena principal com:
 - Modelos dos jogadores
 - Cubos invisíveis para colisão (player1Cube, player2Cube)
 - Obstáculos (enemies) com modelo e caixa
- Objetos adicionados ou removidos conforme o jogo avança



Animações e Funcionamento

- Loop principal com **requestAnimationFrame**
- Atualiza posições dos jogadores com base nas teclas pressionadas
- Move obstáculos e remove os que saem da área visível
- Verifica colisões a cada frame para detectar eventos do jogo



Iluminação

- **DirectionalLight:** luz principal que simula o sol.
 - Projeta sombras realistas com alta definição;
 - Configuração detalhada da área de sombras para melhorar a qualidade.
- **AmbientLight:** luz ambiente que ilumina uniformemente a cena.
- **Sky:** objeto que cria o céu, adicionando realismo visual.
 - Controla parâmetros como turbidez, posição do sol (conecta-se à DirectionalLight)

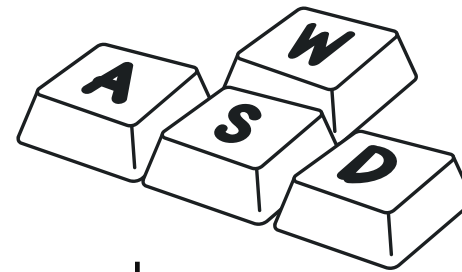
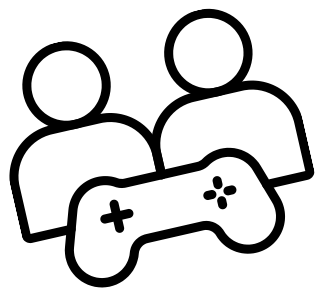
Interação do Utilizador

- **Single Player:**

- Movimento com W, A, S, D ou Setas do teclado
- Saltar com Space

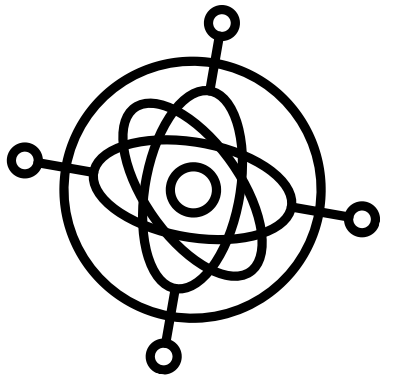
- **Multiplayer:**

- Player 1: W, A, S, D para mover e Space para saltar
- Player 2: Setas para mover e P para saltar



- **Controles no telemóvel:**

- Movimento controlado pelo giroscópio (inclinação do dispositivo)
- Toque na tela para saltar



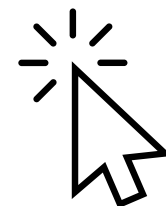
Organização e Implementação

- **Estrutura do Código:**

- Todo o código está num único ficheiro JavaScript.

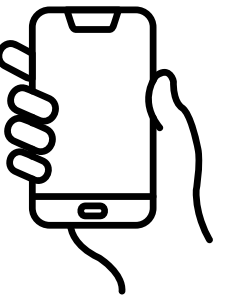
- **Controlo da Câmara:**

- Tecla V alterna entre posições predefinidas da câmara.
- Rato (PC): Usa **OrbitControls** para rodar a câmara livremente.
- Telemóvel: A câmara responde ao giroscópio do dispositivo.



- **Adaptação ao Telemóvel:**

- O jogo deteta se está a correr num dispositivo móvel.
- O giroscópio controla o movimento do cubo e da câmara.
- Lógica específica implementada para distinguir entre inputs de PC e telemóvel.

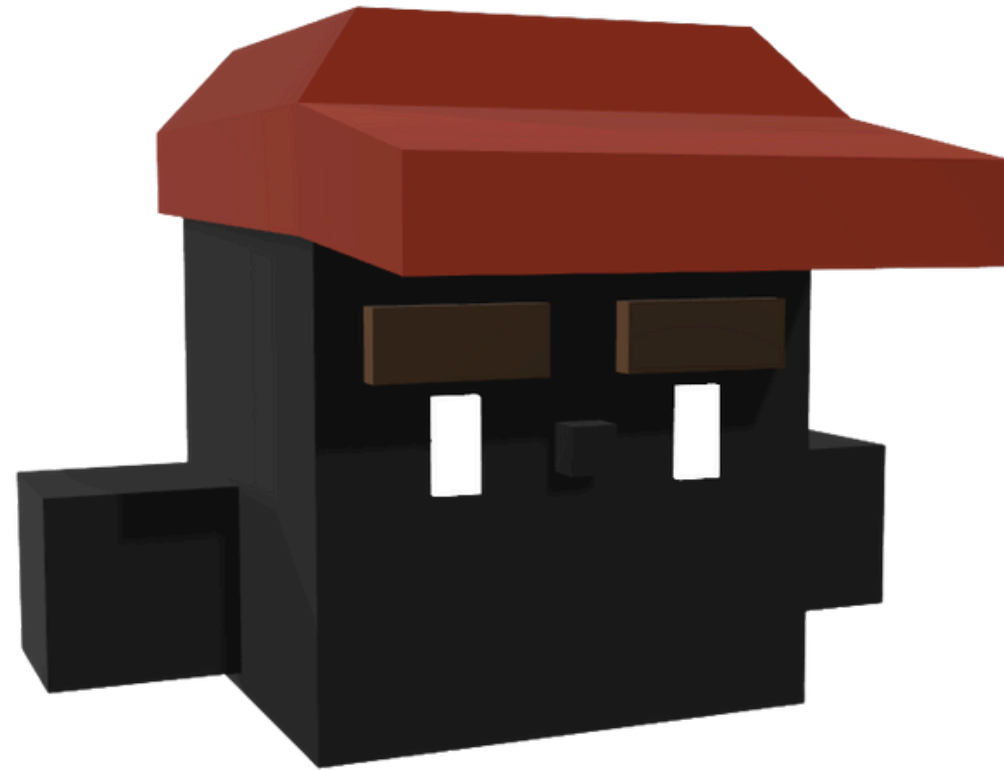


Problemas e Dificuldades



- Calibrar o **giroscópio** foi desafiante pois causava movimentos bruscos e imprecisos.
- Foi necessário implementar lógica distinta para **compatibilidade entre desktop e telemóvel**.
- Ocorreram **problemas de performance**, especialmente com muitos obstáculos ou em modo multiplayer.
- Ter todo o código num único ficheiro dificultou a **organização e manutenção** do projeto.

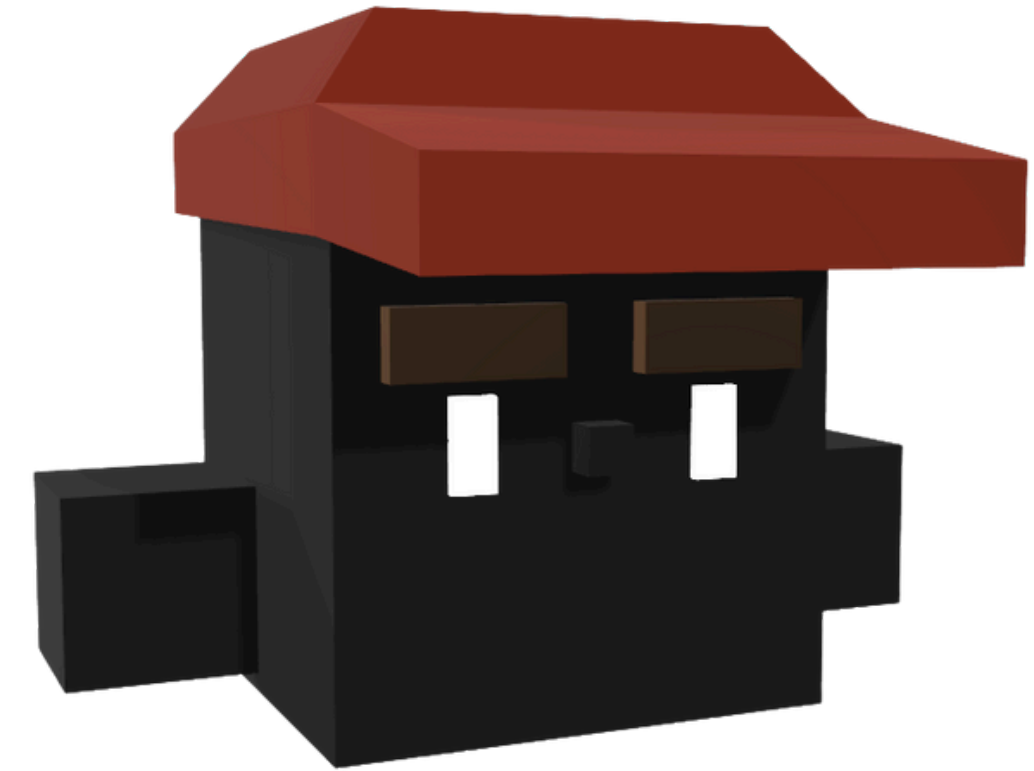
DEMO



Conclusão

- Este projeto permitiu aplicar conceitos de modelação e renderização **3D, interação** e **multiplayer** num jogo web.
- Foi desafiante adaptar o jogo para **desktop** e **telemóvel**, com suporte a teclado, rato e giroscópio.
- A **performance** e **organização** do código foram cruciais, sobretudo no **modo multiplayer**.
- Apesar da limitação de ter tudo num **único ficheiro**, o resultado é um jogo **funcional** e **envolvente**.

Cube Runner



Introdução à Computação Gráfica
DETI, Universidade de Aveiro

Tomás Fonseca 107245 LECI