Simulador de estacionamento

Luís Diogo - nº 108688

Introduction to Computer Graphics – 2024/2025

Ideias principais

- Simulador de estacionamento
- O jogador terá de conduzir os veículos com o objetivo de os estacionar

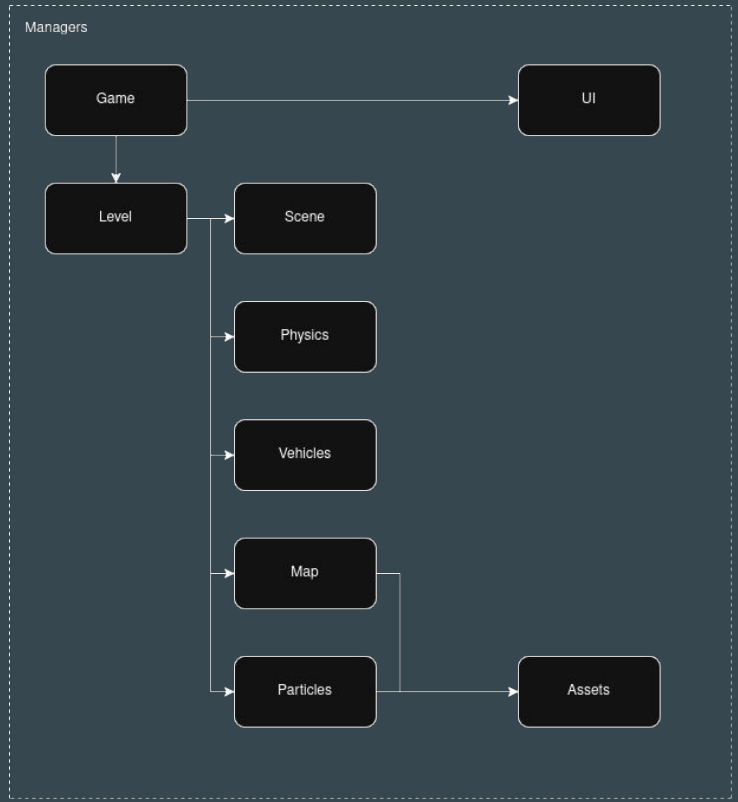
- Módulos utilizados:
  - Threejs
  - Rapier3d - simulação da física
  - mitt - comunicação de eventos

- Jogo disponível em: https://luisdiogo12.github.io/Simulador-de-estacionamento/

Desenvolvimento

Estrutura principal com base numa arquitetura modelar:

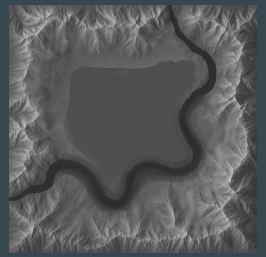
- GameManager
  - gestão da lógica do jogo, inputs e loop de frames para a física e renderização
- LevelManager
  - gestão da lógica do nível: criação, objetivos concluídos, ...
- SceneManager
  - criação e gestão do cenário e operações ao cenário
- PhysicsManager
  - criação e gestão física e operações à física
- VehiclesManager
  - gestão dos veículos
- MapManager
  - criação e gestão do mapa
- AssetsManager
  - criar, padronizar e gerir modelos, entidades, ...



Modelos e Texturas

- Textura do céu
  - autoria alheia
- Flowmap da àgua
  - autoria alheia
- Modelos das estradas
  - autoria própria
- Heightmap do terreno
  - autoria própria

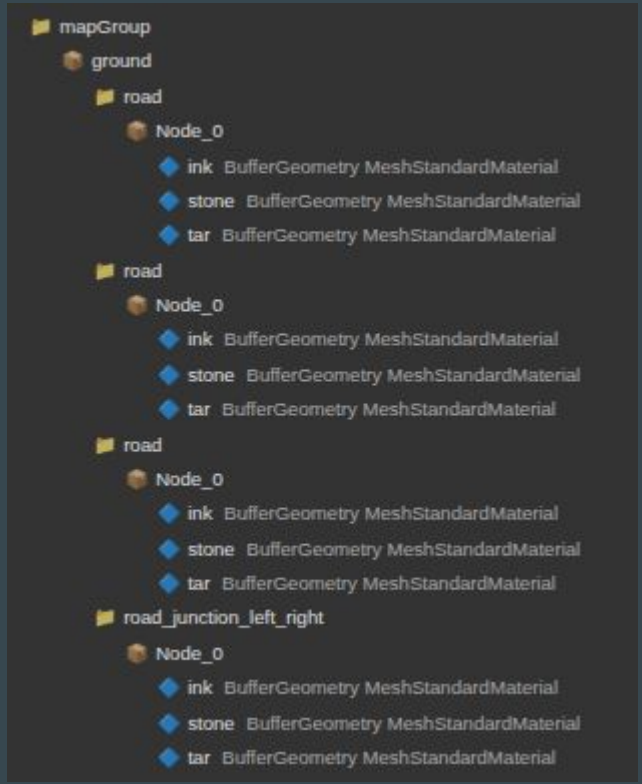
Heightmap para testes



Heightmap utilizado

Grafo da cena

- Montado de acordo com uma convenção que formei para melhor manipular a simulação da física
  - Corpos estáticos montados apenas para efeitos de organização
  - Corpos dinâmicos montados de modo a que Three.Group seja analogo a Rapier.RigidBody e Three.Mesh analogo a Rapier.Collider

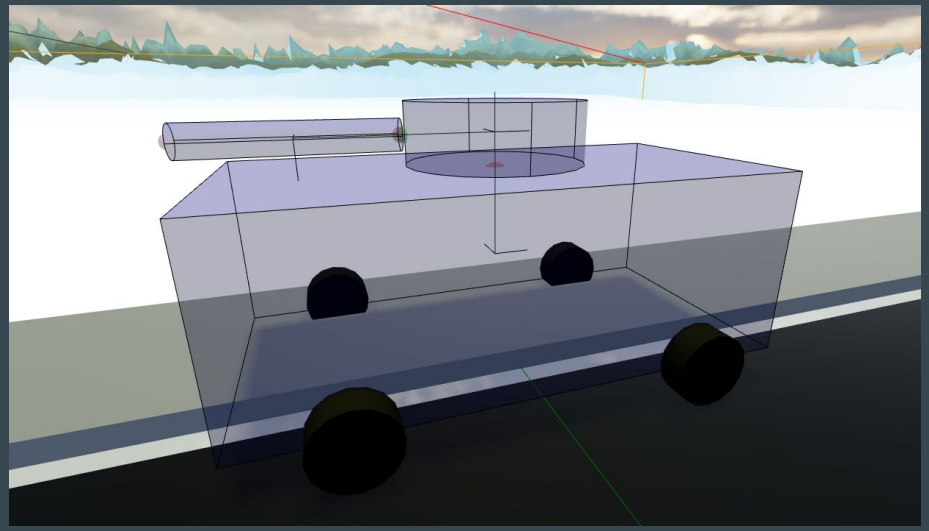


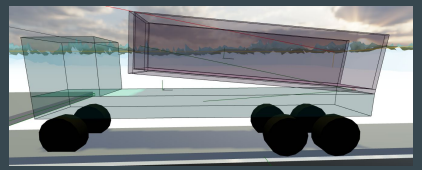
Animações

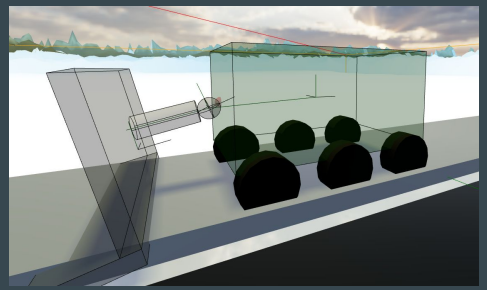
- Todas as animações resultam da simulação da física.
- GameManager implementa o ciclo de animação que chama todas as classes que necessitem de ser animadas
- Cada classe de objetos além de tratar da criação do objeto trata também do seu controlo e declaração de posição, velocidade e força.
- PhysicsManager trata da sincronização das meshes com os bodies.

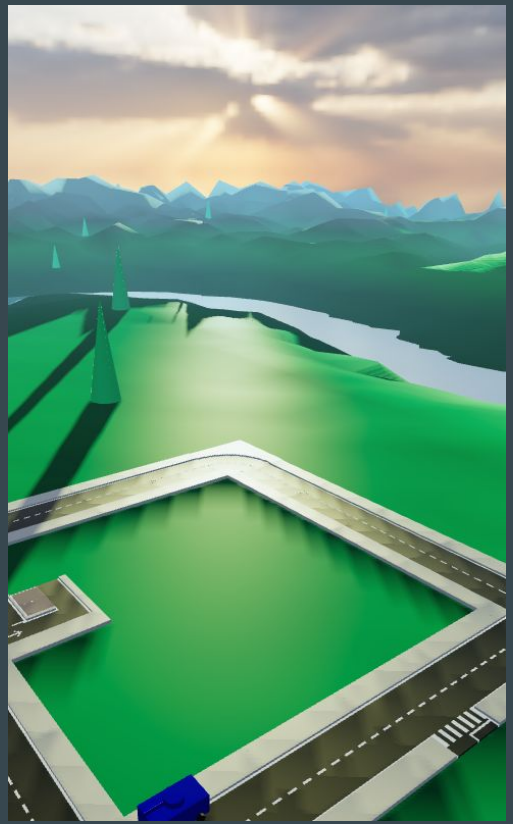
Veículos

- DynamicRayCastVehicleController
  - raycast para detetar contacto com o solo
- ImpulseJoint
  - conexões entre elementos dos veículos
- ImpulseJoint.configureMotorVelocity
  - controlar as articulações
- Pivots
  - marcação de pontos de referência
- RigidBody.setLinvel
  - aplicar velocidade aos projéteis



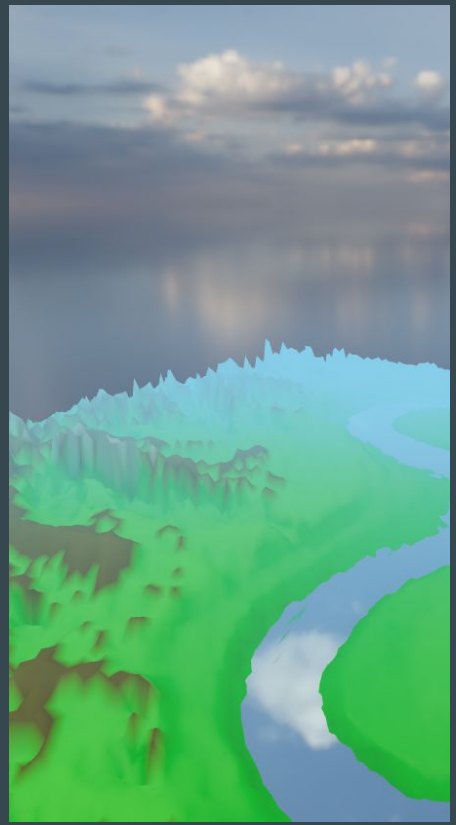






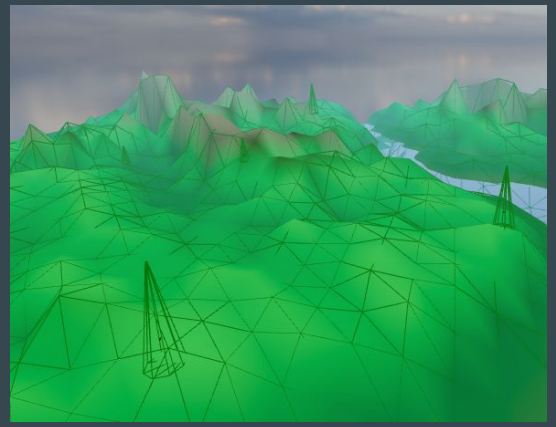
Cenário

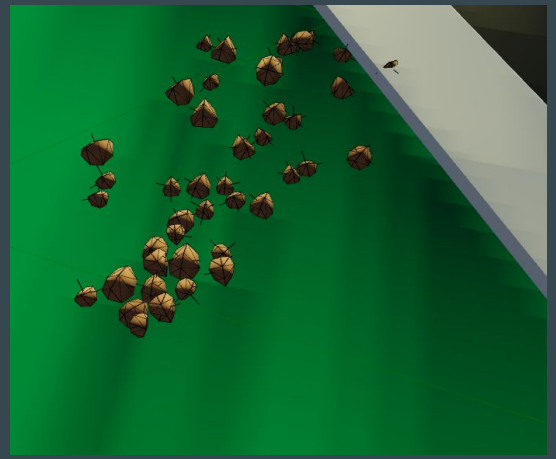
- Iluminação
  - Ambient Light - preencher sombras
  - Spot Light - simular o sol
  - Textura no background e environment - iluminação e reflexões
- Câmara
  - Perspective Camera - simular visualização em 3D
  - Atualização da sua posição de acordo com o veículo selecionado
  - OrbitControls - permitir navegação pelo mapa



Mapa - terreno, estradas e objetos

- Descrição do mapa em json
  - heightmap e posição/rotação das estradas
- Criação da estrada a partir de modelos glTF
- Criação de objetos estáticos
  - Raycast para não ficam localizados onde há estrada e para saber a altura a serem posicionados
  - InstancedMesh para melhor desempenho
  - RigidBodies fixos
- Criação de objetos dinâmicos
  - InstancedMesh para melhor desempenho
  - RigidBodies dinâmicos





Interação do utilizador

- Uso do rato para seleção do veículo
  - Utilizado raycast para localizar a posição do rato
- Uso do rato para navegar pelo mapa
  - Utilizado OrbitControls
- Uso das teclas w, a, s, d, f, arrow up,arrow down, arrow right e arrow left para comandar o veículo selecionado
- Sistema de eventos que permite os vários Managers comunicarem interações

Demo

Conclusões

- Grande parte do tempo investido acabou por ser utilizado para a exploração e aprendizagem.
- A maior dificuldade no desenvolvimento foi a criação da física dos veículos.
- Na proposta de projeto ambiciosa, apesar de não estar tudo implementado, o conceito e base foi concluído com sucesso.

Referências

- https://github.com/mrdoob/three.js
  - Exemplos em threejs e rapier
- https://waelyasmina.net/articles/category/three-js/
  - Tutoriais em threejs
- https://sbcode.net/threejs
  - Tutoriais em threejs
- https://github.com/pmndrs/react-three-rapier
  - Exemplos em rapier
- https://polyhaven.com/a/kloppenheim_06_puresky
  - Textura do céu