E-fólio A | Folha de resolução para E-fólio

UNIDADE CURRICULAR:

CÓDIGO: 21020

DOCENTE: António Araújo / Pedro Pestana

A preencher pelo estudante

NOME: Ivo Vieira Baptista

N.º DE ESTUDANTE: 2100927

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 20 de novembro de 2023

TRABALHO / RESOLUÇÃO:

1. Introdução

O projeto centrou-se na implementação do algoritmo do ponto médio e no desenvolvimento de uma interface gráfica interativa com a biblioteca Three.js. Este algoritmo, essencial para a rasterização de linhas, foi aplicado na criação de linhas numa grelha 3D, proporcionando uma demonstração prática e visual da sua funcionalidade. A interface gráfica ofereceu uma experiência interativa, permitindo aos utilizadores explorar elementos gráficos em 3D.

2. Estrutura do Projeto

a. Estrutura do HTML

O projeto utilizou HTML para estruturar a interface, dividindo-a em uma cena Three.js e um painel de instruções. A tag ** ** foi empregada para a cena 3D e as instruções, facilitando a interação e compreensão dos utilizadores. As tags **<script>** e **<link>** integraram módulos JavaScript e estilos CSS, respectivamente, contribuindo para a coerência visual e técnica da página.

b. Estilo CSS

O design visual foi aprimorado com CSS, escolhendo-se cores contrastantes para clareza e estética. A posição absoluta do painel de instruções e as animações sutis em elementos específicos enriqueceram a interface, tornando-a intuitiva e dinâmica.

3. Implementação do Algoritmo do Ponto Médio

O algoritmo do ponto médio foi essencial para desenhar linhas retas na interface 3D. Baseado no cálculo de erros incrementais, determinava o pixel mais próximo para formar a linha. A implementação incluiu loops para calcular pontos intermediários, ajustes de erro e incrementos direcionais, assegurando a precisão e eficácia do algoritmo na interface.

4. Desenvolvimento da Interface Gráfica com Three.js

a. Criação de Cena e Objetos

A cena 3D foi construída com Three.js, incorporando uma câmera e um renderizador para visualização em 3D. A grelha de quadrados, elemento central da interface, foi criada com quadrados alternadamente coloridos, proporcionando uma experiência visual rica.

b. Controles e Interação

Os controles OrbitControls permitiram uma navegação fluida pela cena 3D. Eventos de teclado e mouse foram implementados para interações como seleção de quadrados, remoção de objetos e reconfiguração da câmera, facilitando uma experiência interativa completa.

5. Animação e Renderização

Um loop de animação contínuo, implementado com a função **requestAnimationFrame**, manteve a cena 3D dinamicamente atualizada. As constantes atualizações e a renderização eficiente da cena asseguraram que as interações dos utilizadores fossem visualmente representadas em tempo real.

6. Desafios e Soluções:

6.1. Desafio de Raycasting:

Barreira: Inicialmente, enfrentamos dificuldades em implementar a funcionalidade de raycasting de forma eficiente para detectar interações do mouse com os quadrados na cena 3D. **Solução:** Após uma investigação detalhada e testes com diferentes abordagens, conseguimos otimizar a lógica de raycasting. Isso não só melhorou a detecção de interações, mas também aumentou o desempenho geral da aplicação.

6.2 Gestão de Estados dos Objetos:

Barreira: Manter o estado correto dos quadrados (como destacado, selecionado, etc.) provou ser complexo, especialmente ao lidar com múltiplas interações consecutivas. **Solução:** Implementamos uma estrutura de dados mais robusta, utilizando listas e variáveis de controle, como **lastHighlightedSquare**, para gerenciar de forma eficaz o estado dos quadrados. Isso permitiu uma maior clareza no código e reduziu os erros de lógica.

6.3 Desempenho na Renderização:

Barreira: Inicialmente, a aplicação enfrentou problemas de desempenho, especialmente ao desenhar linhas e ladrilhos amarelos em tempo real. **Solução:** Melhoramos o desempenho otimizando o código de renderização e reduzindo o número de operações necessárias para desenhar e atualizar a cena. Isso foi alcançado através de uma revisão cuidadosa do código e aplicando práticas recomendadas de otimização do Three.js.

6.4 Intuitividade dos Controles:

Barreira: Recebemos feedback de que os controles da câmera e a interatividade não eram tão intuitivos quanto poderiam ser. **Solução:** Realizamos sessões de teste com usuários e, com base em seus feedbacks, ajustamos a sensibilidade dos **OrbitControls** e melhoramos as instruções na interface do usuário para tornar a navegação mais amigável e intuitiva.

6.5 Gestão de Objetos Adicionados:

Barreira: Um desafio crítico foi gerenciar os objetos adicionados à cena de forma eficiente. Inicialmente, havia dificuldades em rastrear e manipular esses objetos, especialmente ao implementar a funcionalidade de remoção. **Solução:** Superamos este obstáculo desenvolvendo uma lista chamada **addedObjects**. Essa estrutura permitiu um controle mais preciso dos objetos adicionados, facilitando sua gestão e remoção conforme necessário. A abordagem simplificou a lógica de manipulação dos objetos e melhorou a clareza do código.

6.6 Cópias de Objetos para Algoritmo de Linhas:

Barreira: Durante a implementação do algoritmo lineMP, deparamo-nos com a necessidade de manipular os pontos de entrada (P e Q) sem alterar os dados originais. **Solução:** A solução encontrada foi criar cópias desses objetos antes de processá-los no algoritmo. Utilizando a sintaxe **P = { ...P }** e **Q = { ...Q }**, conseguimos garantir que as operações dentro do algoritmo não afetassem os dados originais. Essa abordagem preservou a integridade dos dados e garantiu um comportamento previsível e correto do algoritmo.

7. Conclusão

O projeto EfolioA foi uma experiência enriquecedora, destacando-se no aprofundamento técnico, na resolução de problemas e no desenvolvimento de habilidades em programação. Os desafios enfrentados, como a complexidade técnica e a otimização de desempenho, contribuíram para um aprendizado valioso em computação gráfica e desenvolvimento de software e como funciona o algoritmo do ponto medio.

Enfrentar e superar as barreiras não só melhorou significativamente a funcionalidade da nossa aplicação, mas também enriqueceu nosso conhecimento e habilidades técnicas. Este processo foi essencial para transformar desafios iniciais em oportunidades de aprendizado e crescimento profissional.

8. Referências

Webgrafia:

http://server.ivo.com.pt:8080 (programa online no meu servidor para testar)

https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

https://threejs.org/docs/#manual/en/introduction/Installation

https://threejs.org/docs/#api/en/objects/Mesh

https://threejs.org/docs/#api/en/core/BufferGeometry

https://threejs.org/docs/#api/en/scenes/Scene

https://stackoverflow.com/questions/24437296/keeping-camera-above-ground-using- orbitcontrols-js

https://stackoverflow.com/questions/19426559/three-js-access-scene-objects-by-name-or-id

https://chat.openai.com (como debugger do código, e consultas)