

Introdução à Computação Gráfica

Atividade Prática 0 - JavaScript e Three.js

Francisco Siqueira Carneiro da Cunha Neto¹

20190029015

¹francisconeto@eng.ci.ufpb.br

19 de agosto de 2021

QUESTÕES RESPONDIDAS

Todas as questões propostas foram respondidas. Isso inclui os problemas 1 – 5 da seção 3 (Problemas JavaScript) e os problemas 1 – 3 da seção 4 (Problemas Three.js). O código dos problemas da seção 3 se encontra na pasta `//JavaScript/js/`, em que cada problema possui um arquivo `.js` diferente, enquanto `//JavaScript/index.html` é um documento HTML que os exemplifica. O código referente a seção 4 se encontra no arquivo único `//ThreeJS/js/cubes.js`, similarmemente junto com o documento HTML `//ThreeJS/index.html`.

O código também pode ser encontrado em seu [repositório no GitHub](#).

ABORDAGEM

Todo o código para os problemas da seção 3 foram estruturados na forma de funções. Algumas dessas funções resolvem o problema como descrito na atividade e outras fazem uma interface entre o código JavaScript e o usuário que interage com ele através do documento HTML.

Os problemas 1 e 2 da seção 3 necessitaram apenas do conhecimento prévio de JavaScript do autor. O algoritmo Quick-sort usado para resolver o problema 3 foi baseado no artigo [1]. A teoria matemática usada para o problema 4 foi consultada do livro [2], e a criação das classes de JavaScript do artigo [3]. Embora o enunciado solicite a criação de “uma pequena biblioteca”, o uso da *keyword export* gerou um erro no documento HTML de *Cross-Origin Resource Sharing* (CORS), descrito no artigo [4]. Esse erro não foi resolvido ao definir `type="module"` na *tag script* que importa o código no documento e, segundo [4], para resolvê-lo é necessário usar uma URL HTTPS, o que foge do escopo da atividade. Portanto, as classes foram escritas, mas sem a criação de uma biblioteca. O uso do elemento HTML *canvas* para a resolução do problema 5 foi norteado pelo artigo [5].

O código que resolveu os problemas da seção 4 foi estruturado da seguinte forma: Foram declaradas variáveis globais *camera*, *scene* e *animation* para armazenar respectivamente uma referência à câmera, à cena e a alguma função anônima,

e foi declarada uma função *animate()*, chamada a cada *frame*, que executa a função referenciada por *animation* e renderiza a cena referenciada por *scene* usando a câmera referenciada por *camera*. Para cada problema foram, então, criadas funções que modificavam a posição da câmera referenciada por *camera* e substituíam tanto a cena referenciada por *scene* quanto a função referenciada por *animation*. Assim, foi possível manter todas as resoluções em uma mesma exibição, escolhendo qual exibir de acordo com a função chamada (controlada por botões no documento HTML). Todos os problemas da seção 4 foram resolvidos com base na documentação do Three.js [6].

Um problema extra também foi proposto e resolvido pelo autor, em que três cubos com materiais distintos e sendo rotacionados em tempo real são desenhados na tela do *browser*.

REFERÊNCIAS

- [1] T. Cormen e D. Balkcom. (2014). “Khan Academy - Algorithms: Quick Sort,” endereço: <https://www.khanacademy.org/computing/computer-science/algorithms/quick-sort> (acesso em 16/08/2021).
- [2] W. K. Nicholson, *Álgebra Linear*. AMGH Editora, 2014.
- [3] MDN. (2021). “Classes - JavaScript,” endereço: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes> (acesso em 17/08/2021).
- [4] —, (2021). “Reason: CORS request not HTTP,” endereço: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS/Errors/CORSRequestNotHttp> (acesso em 17/08/2021).
- [5] —, (2021). “<canvas>: The Graphics Canvas element,” endereço: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas> (acesso em 17/08/2021).
- [6] Three.js. (2015). “three.js docs,” endereço: <https://threejs.org/docs/> (acesso em 18/08/2021).