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

UNIDADE CURRICULAR: Computação Gráfica

CÓDIGO: 21020

DOCENTE: António Araújo e Pedro Pestana

A preencher pelo estudante

NOME: Mário Pedro Capela Rodrigues Carvalho

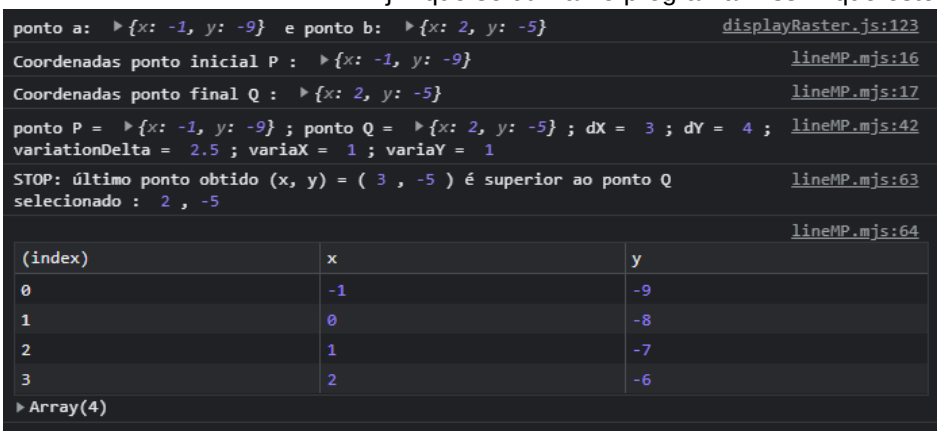
N.º DE ESTUDANTE: 2000563

CURSO: Licenciatura em Engenharia Informática

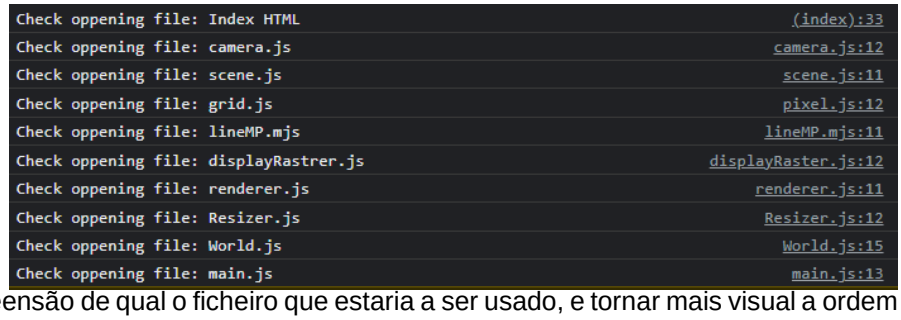
DATA DE ENTREGA: 22 de novembro de 2022

TRABALHO / RESOLUÇÃO:

Este relatório tem como objetivo de explicar o trabalho realizado em paralelo, até ter sido alcançado o objetivo final do e-fólio em questão. Este, demonstra assim, o que fora pensado em cada etapa, e como se foi ponderando cada parte da sua construção, para ser realizado com sucesso.
Assim, antes de se iniciar qualquer parte de código, ou reproduzir as condições necessárias, teve-se em atenção em criar a arquitetura daquilo que foi pedido, para não ser critério de exclusão (organização das pastas, nomes, recursos aos "imports" necessários – CDN, three.js, ...).
Além de que, se realizaram préviamente vários cálculos de diferentes pontos médios, para que se pudesse ter uma base de comparação correta sobre os resultados que se teriam de atingir segundo a fórmula, para calcular cada ponto da reta em cada quadrante diferente.
Para facilitar o cálculo de obtenção dos vários pontos, iniciais e finais diferentes, de forma imediata sem se ter de calcular cada ponto para cada uma das novas retas, criou-se em primeiro lugar um ficheiro Excel, que permitiria de forma exata calcular todos os pontos segundo a fórmula do "ponto médio". Recorreu-se a esta fórmula para comparar e validar os resultados obtidos através do código realizado no ficheiro "lineMP.mjs" que se utiliza no programa. Assim que este ficheiro foi implementado, associou-se um novo ficheiro de construção de pontos de modo automatizado e randomizado, para cada vez que se atualizava o programa, se pudesse verificar várias possibilidades aleatórias. Esta possibilidades podiam ser e ainda são consultadas no browser, em modo "DevTools/Console". Aqui foi encontrada a maior dificuldade, pois, certas funções não funcionava para os quadrantes negativos, ou para os eixos verticais, ou então, quando os pontos eram selecionados no tabuleiro, de baixo para cima. Isto é, imaginado que o primeiro ponto é inferior à seleção do segundo ponto num eixo dos xx. Após várias tentativas conseguiu-se colocar a função a imprimir os valores certos. Mas esta função, teve que quer novamente revista durante a implementação da Parte II, para garantir que os pontos vermelhos eram imprimidos no tabuleiro corretamente.



Uma vez que a Parte I – Implementação do algoritmo do ponto médio foi concluída, iniciou-se a implementação da Parte II – Interface Gráfico (1, e 2). Para melhor compreensão de qual o ficheiro que estaria a ser usado, e tornar mais visual a ordem em que cada parte se processava, foi introduzido um "console.log" indicativo de cada ficheiro.



Toda a restante estrutura foi sendo realizada por partes nos respetivos ficheiros, com importação de uma estrutura idêntica e bem dividida segundo a documentação de "threejs.org", com os ficheiros pedidos (index.html, lineMP.mjs, ReadMe.pdf) na sua base, e os restantes, subdivididos dentro de "./src", onde constam três grupos : "./styles" para o ficheiro correspondente a ".css", "./World", que tem "./components" e "systems", e "main.js", que importa o ficheiro "World.js". Esta subdivisão, tal como indicada na documentação de "three.js", permite de forma mais segura,

limitar o acesso às propriedades de cada objeto a partir de uma fonte externa, e simultaneamente, facilitar a sua importação para outro projeto de forma idêntica, sem que esta sofra modificações ou altere o projeto seguinte.
Deste modo, em "./systems", correm os ficheiros "renderer.js", e "Resizer.js". Em "./components", correm os ficheiros "camera.js", "scene.js", "./pixel.js" e "./displayRaster.js".

Estes dois últimos, servem para criar um tabuleiro. Em que, dentro de "pixel.js" se cria um quadrado dimensionado para 2D, com as condições e dimensões pedidas (quanto a tamanho dos lados e cores) que são originalmente impostas a partir da função de "displayRaster.js". Este ficheiro, tem em si a responsabilidade de criar um tabuleiro "pixel a pixel", com as instruções de cada pixel, dado pela função existente no ficheiro "pixel.js". Nova dificuldade, mas ultrapassada para que se pudesse imprimir os "pixel's" com diferentes cores, e corretas posições. Teve-se de realizar uma manipulação mais profunda sobre a ordem em que cada parte do código ocorria no "displayRaster.js", juntamente com o "pixel.js". Assim, "pixel.js", fico uma função mais generalizada, e o "createBoard()" mais simplificado.
Ainda aqui, foram incorporadas as funcionalidades relativas aos "events" para que pudesse movimentar o tabuleiro com o rato, selecionar cada "pixel" com a tecla "x" e construir a linha respetiva, e, se apagar a seleção de pontos realizada anteriormente. Apagar pontos, mais uma vez obrigou a implementar um "array" que ficava novamente vazio, mas dificuldade foi ultrapassada quando se colocou no lugar certo. A implementação de "OrbitControls" (CDN) foi criada em "World.js", pois esta tem a funcionalidade de conectar o utilizador à interface. No entanto, não ficou a funcionar de forma completamente fluída. Isto é, só se verifica a interação do utilizador, assim que se seleciona o segundo ponto com "x" no tabuleiro, depois de o movimentar. Em cada função ao longo do resto do "displayRaster.js", atribuiu-se um nome o mais aproximado possível do seu objetivo, apresentando-se a função respetiva e associando um comentário, para melhor compreensão.
Encontraram-se algumas dificuldades com a implementação de eventos, para que funciona-se sistematicamente e corretamente. Dificuldade foi ultrapassada, quando se percebeu que certas variáveis deveriam ser reinicializadas de alguma maneira, ou então, deveriam ser convertidas a variáveis temporárias (ex: em "drawLine()").
Apesar dos cálculos para a criação da "middlePoint line" estarem corretos, foi difícil conseguir fazer bater certa a posição entre o rato e o tabuleiro para a seleção dos pontos.

Para fazer funcionar o projeto, utilizou-se como servidor o aconselhado no fórum da disciplina ('Web Server for Chrome').
E-fólio encontra-se em modo partilha a partir do dia seguinte à data limite de entrega, em repositório do estudante, para facilitar acesso ao trabalho realizado mesmo em caso de dificuldades de abertura deste documento partilhado em modo "Carvalho2000563.zip".