

**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

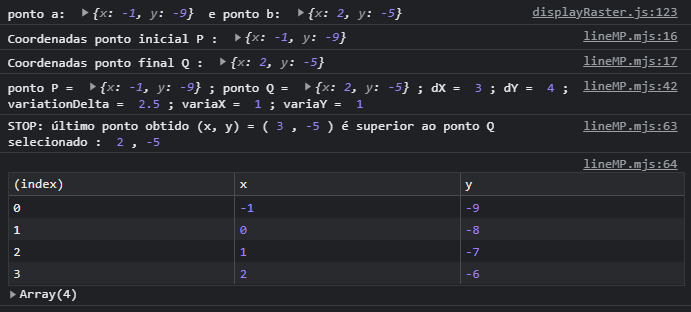
**DATA DE ENTREGA:** 22 de novembro de 2022

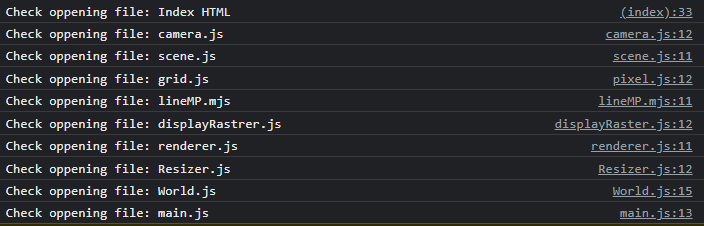
**TRABALHO / RESOLUÇÃO:**

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 calculo 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”.

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”. 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. Em cada função, descreveu-se o seu objetivo principal, apresentando-se a função respetiva com um nome mais aproximado possível do seu objetivo, e associando um comentário, para melhor compreensão.

A possibilidade de manipulação do tabuleiro foi criada em “World.js”, pois esta é a funcionalidade que tem por objetivo ligar o utilizador com a interface.

Sempre que um pixel do tabuleiro for selecionado, este muda de cor (vermelha), e quando dois pixels forem selecionados, uma linha fica formada em cima da seleção e os pixéis correspondentes mudam para a cor amarela.

Encontraram-se algumas dificuldades com a implementação de eventos, para que funciona-se sistematicamente e corretamente. Dificuldade ultrapassada quando foi compreendido 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.

E-fólio encontra-se em modo partilha a partir do dia seguinte à data limite de entrega, em [repositório do estudante](mailto:https://github.com/MP-C/UaB.ComputacaoGraficaEfolioA?subject=repositório%20do%20estudante), para facilitar acesso ao trabalho realizado mesmo em caso de dificuldades de abertura deste documento partilhado em modo “.zip”.