



## E-fólio A | Instruções para a realização do E-fólio



# COMPUTAÇÃO GRÁFICA | 21020

### Período de Realização

Decorre de 10 de dezembro de 2021 a 3 de janeiro de 2022

### Data de Limite de Entrega

3 de janeiro de 2022, até às 23:55 de Portugal Continental

### Trabalho a desenvolver:

Sejam dados 4 pontos no espaço tridimensional. Sejam eles  $C_i = (x_i, y_i, z_i)$ ,  $i = 0, \dots, 3$ . Chama-se Bezier cúbica definida pelos pontos  $C_i$  à curva seguinte:

$$f(t) = (1-t)^3 C_0 + 3(1-t)^2 t C_1 + 3(1-t) t^2 C_2 + t^3 C_3, 0 \leq t \leq 1$$

Este trabalho consiste de duas partes: primeiro deverá implementar uma função que calcula a posição no instante  $t$  de uma curva Bezier cúbica de parâmetros especificados por pontos de controle dados. Na segunda parte implementará um interface gráfico para gerar plots dessa função a partir de pontos de controlo especificados graficamente.

Recordo que uma Bezier cúbica é uma curva parametrizada em  $t \in [0,1]$ , que fica definida por quatro pontos de controlo.

### PARTE I- Implementação do algoritmo do ponto médio (2 valores):

Deverá implementar uma função chamada `bezier3`. Esta função tem como input um objecto literal da forma `{c0:Vector3, c1: Vector3, c2:Vector3, c3:Vector3, t:float}` onde  $t$  é um float no intervalo  $[0,1]$ , e  $c_0$ ,  $c_1$ ,  $c_2$ ,  $c_3$ , são vectores `three.js` (<https://threejs.org/docs/#api/en/math/Vector3>).

O output da função `bezier3` é um `Vector3`, que contém a posição no espaço, para o valor  $t$  do seu parâmetro, da Bezier cúbica definida pelos pontos de controle  $c_0$ ,  $c_1$ ,  $c_2$ ,  $c_3$ .

A função deverá ser exportada a partir de um módulo chamado `bezier3.mjs` localizada na mesma pasta que o `index.html`. Para standardizar o export, peço ainda que o façam na forma explícita, assim:

```
export {bezier3};
```

Nota: tal como no e-fólio A, os interfaces não são negociáveis. A função tem que correr, e os inputs e outputs têm que estar no formato especificado. Não respeitar estes dois pontos obrigará a um desconto entre 30% e 100% da nota do trabalho, consoante a gravidade do desvio.

## PARTE II- Interface Gráfico (2 valores)

Deverá fazer uma página web em three.js que implemente um interface gráfico que permite seleccionar quatro pontos de controle e a partir destes gerar a curva Bezier definida por esses quatro pontos.

Vamos reaproveitar, e generalizar, o interface do e-fólio A. Faça um tabuleiro de xadrez como da última vez, mas desta vez não vamos simular pixels. Vamos em vez disso usar o tabuleiro como referencia, por isso centre o referencial no vértice comum dos quatro ladrilhos centrais, e faça o tabuleiro simétrico em relação a estes, com 10 ladrilhos para cada lado, portanto um tabuleiro de lado 20. Desta vez faça o referencial completo com os três eixos. Faça os ladrilhos de lado igual a 1, para que os vértices correspondam às coordenadas inteiras. Faça os ladrilhos semi-transparentes para conseguirmos ver a curva que desenharemos se ela atravessar o plano. Implemente Orbit Controls.

Interacção: Dispostas sobre o tabuleiro, inicialmente a formar um quadrado, estão 4 pequenas bolas (raio=0.5) semi-transparentes (mas bem visíveis), com o centro sobre o plano. O centro de cada bola corresponde às coordenadas de um ponto de controle da curva Bezier que vamos desenhar. As bolas correspondentes aos pontos  $C_0, C_1, C_2, C_3$  são respectivamente de cor amarela, vermelha, verde, azul. Quando carrego nas teclas 1, 2, 3, ou 4, selecciono a bola correspondente ao ponto de controle  $C_0, C_1, C_2, C_3$  respectivamente (notem por favor a habitual divergência entre a numeração ordinal dos pontos (1, 2, 3, etc.) e a representação interna (0, 1, 2, etc.)). Quando uma bola está seleccionada perde a transparência, ficando opaca. É então possível mudar a posição da bola sobre o plano clicando sobre o ponto do plano para onde pretendemos que ela se mova. A sua altura relativa ao plano pode também ser mudada carregando nas teclas W ou S, que a fazem subir ou descer na perpendicularmente ao plano de referência. Não há limite superior ou inferior para esta “altura”, e a variação deve ser relativamente suave, rápida, e precisa, para permitir uma fácil manipulação da altura de cada ponto (por exemplo, um movimento de 0,1 unidades por clique que acelera se a tecla for mantida sob pressão). Quando uma bola está acima ou abaixo do tabuleiro, deve haver uma recta fina que a une ao mesmo na perpendicular, para sabermos sobre qual ponto do plano ela se encontra e estimarmos a altura. Quando mudamos a posição da bola clicando no tabuleiro com o mouse, a sua altitude em relação ao tabuleiro deve permanecer invariante (ou seja, a posição sobre o plano e a altura são controladas independentemente pelo mouse e teclas respectivamente). Deve aparecer um texto num canto da cena a mostrar qual a bola seleccionada e quais as suas coordenadas no momento.

Se o utilizador carregar na tecla “x” o programa desenhará a Bezier definida pelos quatro pontos de controle definidos pelos centros das bolas (utilizando necessariamente a função que definimos na secção anterior).

A Bezier desenhada deverá necessariamente ser um objecto TubeGeometry ( <https://threejs.org/docs/#api/en/geometries/TubeGeometry> ) de parâmetros adequados, que segura uma Curve de three.js definida pela nossa função bezier3.

Será possível desenhar várias Beziers, carregando em x. Cada vez que carregamos geramos uma nova bezier para os pontos actuais. Façam a bezier fina mas bem visível (ligeiramente mais fina que as bolas).

Carregar em Backspace faz *reset* à cena.

### Especificações adicionais:

Programa os algoritmos e métodos usando JavaScript, Html, CSS e Three.js e comente adequadamente o código fonte.

Deve juntar um relatório explicativo de todas as decisões tomadas. Este relatório constituirá uma memória descritiva do trabalho realizado, devendo ser sucinto (máximo de 2 páginas A4, fonte 10, espaçamento simples) e complementar os comentários inseridos no código fonte;

Faça entrega de todos os ficheiros comprimindo-os num único ficheiro zip, adotando o seguinte formato: <Ultimo nome do aluno><numero de estudante>.zip.

Deve carregar o referido ficheiro para a plataforma no dispositivo E-fólio B até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas.

Indique se detectar qualquer problema com o enunciado. Como o trabalho é individual, utilize o meu email em vez dos fóruns para qualquer dúvida cujo esclarecimento possa dar pistas impróprias aos seus colegas (na dúvida, assumo que é o caso): antonio.araujo@uab.pt

A página deverá correr a partir de um ficheiro index.html localizado na directoria de base do seu trabalho.

O ficheiro a enviar não deve exceder 8 MB.

O código deverá ser importado a partir do index.html, na forma de módulos. **Preferencialmente** o código JavaScript deve estar totalmente contido em módulos e não inline no html. O html deve apenas importar e correr um módulo inicial, que chama os demais.

Um desses módulos terá por sua vez que importar o three.js e o OrbitControls. Essas importações deverão ser feitas a partir de CDN, da seguinte forma:

```
import * as THREE from 'https://unpkg.com/three@0.124.0/build/three.module.js';  
  
import { OrbitControls } from  
'https://unpkg.com/three@0.124.0/examples/jsm/controls/OrbitControls.js'
```

É essencial que todas estas dependências funcionem (exceptuando claro os casos de eventuais indisponibilidades do servidor). **Só será alvo de avaliação código que corre.**

Deverá ainda importar o módulo bezier3.mjs, que desenhou na alínea acima para implementar o cálculo da curva que serve de base ao TubeGeometry. Este módulo deve estar na directoria base. Os outros módulos deverão estar numa directoria “./src”.

Volto a frisar que o seu programa deverá estar **pronto para correr, sem dependências partidas**. Inclua na directoria base o ficheiro **lineMP.mjs**, o ficheiro **index.html** que corre a segunda parte do trabalho, e

o relatório do trabalho (chamado ReadMe.pdf); quaisquer outros ficheiros, incluindo os seus outros módulos, eventuais ficheiros css, etc, devem ser colocados em pastas, sendo que os módulos JavaScript devem estar numa pasta “./src”.

Não deve utilizar outras libraries além das especificadas, nem código copiado em bloco de outras fontes. O código que faz deve ser razoavelmente perto do mínimo necessário para o objectivo, e não uma construção em cima de framework pré-definidas. Isso será tido em conta na avaliação. Em particular não há nenhum motivo para que o código ultrapassar os 8MB (na verdade nem mais que uns 100 KB).

O seu trabalho será avaliado por vários factores:

- a implementação das features pedidas.
- a correcção, elegância, e eficiência do código e da documentação do mesmo.
- **requisito essencial (eliminatório):** o programa tem que estar pronto a correr, sem alterações nem dependências quebradas, seguindo as normas estabelecidas acima para os interfaces e objectos.

Votos de bom trabalho!

A. Araújo