

”

**E-fólio B** | Folha de resolução para E-fólio

**UNIDADE CURRICULAR:** Computação Gráfica

**CÓDIGO:** 21020

**DOCENTE:** António Araújo / Pedro Pestana

**A preencher pelo estudante**

**NOME:** Gonçalo Caraça

**N.º DE ESTUDANTE:** 2000130

**CURSO:** Engenharia Informática

**DATA DE ENTREGA:** 09/01/2023

## TRABALHO / RESOLUÇÃO:

### 1ª Parte:

Como solicitado no enunciado, primeira parte do trabalho consistia em construir uma Bezier Quártica que recebe um objeto com 5 vetores e um “t” que será um float no intervalo [0,1].

Com a função fornecida no enunciado, calcular “x”, “y” e “z” final, para de seguida com essas coordenadas construir um novo vector3 e devolver esse resultado que contém a posição no espaço, para o valor “t”.

### 2ª Parte:

Comecei por criar uma função para desenhar cada quadrado (drawSquare), e de seguida a função (drawGrid) que chama a função (DrawSquare) para, conforme o controlo efetuado nos meus ciclos “for” desenhar quadrados de forma intercalada.

Tendo em conta que neste trabalho foi solicitado o referencial completo com os 3 eixos, recorri ao (axisHelper) do three.js que desenha automaticamente os 3 eixos necessitando apenas de lhe passar o comprimento da linha, neste caso específico “15”.

De seguida, como se pretendia pequenas esferas de raio 0,5, utilizei a função (drawSphere) que recebe as coordenadas, a cor, scene e o nome que lhe será atribuído para facilitar o seu acesso quando necessário.

Função criada, decidi colocar as esferas pelo tabuleiro de forma aleatória usando “Math.random”, assim, a cada inicialização teremos as bolas posicionadas de maneira diferente.

Para poder seleccionar as bolas como pretendido, ao premir os números 1,2,3,4,5, criei uma função que é acionada ao clicar numa destas teclas, (onKeyPress), que ao ser ativado um destes números, vai buscar à scene a bola correspondente, altera a sua opacidade para 1 e de seguida poderemos alterar

a sua posição premindo “Space”, em que irá ativar o event “Space” que por sua vez, com o auxílio do raycaster recebe as coordenadas onde o rato se situa e altera a posição da bola, nota para que neste “event” já estamos a desenhar uma linha vermelha (drawline) sempre até ao plano  $z=0$  como solicitado no enunciado.

Event.key “w” é um evento bastante simples que apenas incrementar em 0.1 a cada clique a posição da bola em Z + e o event.key “s” faz o incremento em direção oposta, Z -.

Ambos este 3 evento tem sempre em comum o desenho da linha vermelha, encontrar a posição da bola no ecrã e passar as novas coordenadas para o html.

Para auxílio da função BackSpace, que é limpar o ecrã, foi criado uma função auxiliar de nome “initialize()”, que ao ser chamada limpa o tabuleiro, desenha novamente os eixos e coloca as bolas no tabuleiro.

Por fim a chamada da função Bezier quártica com o event “x”, começa por criar o “obj” com as coordenadas de todos os 5 pontos, de seguida criei a função (curve) para que com os pontos em questão e 3 valores de “t” definidos por mim no intervalo pretendido entre [0,1], efetuar a chamada da função bezier4 e criar a curva, curva essa que será desenhada com “TubeGeometry”.

#### **Anexos:**

- Programa (a correr).
- Relatório.
- Demo.

**FIM**