



## Computação Gráfica — 21020

### Período de Realização

Consultar os prazos de entrega indicados pelos serviços.

### Objetivos

O e-fólio global cobre potencialmente a totalidade da matéria lecionada.

A prova é composta por 4 questões divididas por 2 grupos, contém 2 páginas e termina com a palavra **FIM**.

### Recursos

A prova é individual, com consulta bibliográfica livre.

### Critérios de Avaliação e cotação

Todas as respostas devem ser justificadas, salvo instrução em contrário. Respostas não devidamente justificadas são inválidas e terão cotação zero.

Todas as questões têm a mesma cotação, somando um total de 12 valores.

### Normas as respeitar

Deve redigir o seu E-fólio na Folha de Resolução disponibilizada e preencher os dados do cabeçalho. A prova deve ser entregue como um único ficheiro zip ou rar contendo um pdf e um ficheiro html, com um máximo de 8 megabytes. Não são aceites outros formatos.

O nome do ficheiro pdf deve ser: número de estudante seguido do seu apelido, seguido de EfolioG. Exemplo: 123456SilvaEfolioG.pdf; o nome do zip e do html deve ser o mesmo, mas com a devida extensão relativa a cada tipo de ficheiro;

Utilize letra legível, se a prova for manuscrita. Atente à qualidade e legibilidade da digitalização.

No ato da entrega, assegure a integridade do ficheiro. Ficheiros que não abrem não podem ser corrigidos.

O e-fólio global dura 90 minutos, tendo uma tolerância de 60 minutos adicionais para digitalizar e carregar o ficheiro na plataforma.

Deve carregar o referido ficheiro para a plataforma no dispositivo disponibilizado para o efeito, até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas técnicos.

Votos de bom trabalho!

António Araújo

# Grupo I

1. Considere o segmento de recta que une o ponto  $(0, 1)$  ao ponto  $(1001, 0)$ . Qual é o menor valor de  $x$  tal que um pixel  $(x, 0)$  é iluminado pelo algoritmo de Bresenham? Explique porquê, de acordo com as propriedades do algoritmo.
2. Utilize o algoritmo scan-line para preencher o polígono definido por  $A = (1, 0), B = (3, 2), C = (5, 0), D = (6, 3), E = (0, 2)$ . Apresente a ET e AET em cada iteração, bem como os *pixels* de preenchimento. Represente o resultado final graficamente.
3. Considere a curva de Bézier  $t \mapsto f(t)$ , com três pontos de controlo  $A = (A_x, A_y), B = (B_x, B_y), C = (C_x, C_y)$ . Suponha que sabe que  $A = (0, 0)$  e  $C = (6, 2)$ , e que sabe ainda que o valor da derivada da curva em  $t = 0$  é  $f'(0) = (0, 4)$ . Determine o ponto de controle  $B$  bem como a expressão da parametrização  $f(t)$ . Represente graficamente a curva.

# Grupo II

Resolva este grupo num ficheiro html. Comentários adicionais à resolução podem ser colocados no pdf da prova.

1. Considere a seguinte cena a implementar em three.js (Figura 1).

Coloque duas esferas vermelhas em cena. As esferas são idênticas e estão colocadas uma directamente em cima da outra, tocando-se num ponto.

Em torno da esfera inferior há um cubo verde, semi-transparente, com opacidade de 50%. O lado do cubo tem o comprimento igual ao diâmetro da esfera, e os seus centros coincidem, pelo que a esfera está inscrita no cubo, tocando-o no centro de cada uma das suas faces.

O cubo descreve um movimento de rotação uniforme em torno do eixo vertical, no sentido oposto ao dos ponteiros do relógio (quando visto de cima), descrevendo uma volta completa a cada quatro segundos.

Coloque uma fonte de luz e camera adequada para ter aproximadamente o aspecto que se apresenta na figura; a camera está situada numa altura superior à do cubo, e aponta ao ponto de contacto entre as duas esferas. O viewport é um quadrado.

Formato de entrega:

O código deve ser escrito directamente num único ficheiro html, que idealmente deverá correr sem erros. Resolva o problema por fases, que terão classificação parcial, mesmo que não cumpra todos os objectivos. Por exemplo, colocar as esferas dá alguma cotação, colocar o cubo transparente dá mais uns pontos, fazê-lo rodar outros tantos; por isso avance por fases, assegurando que o código corre em cada fase.

Não inclua a library three.js nem ficheiros js auxiliares no seu zip, mas apenas o ficheiro index.html com o código inline, e que deverá chamar o three.module.js num repositório

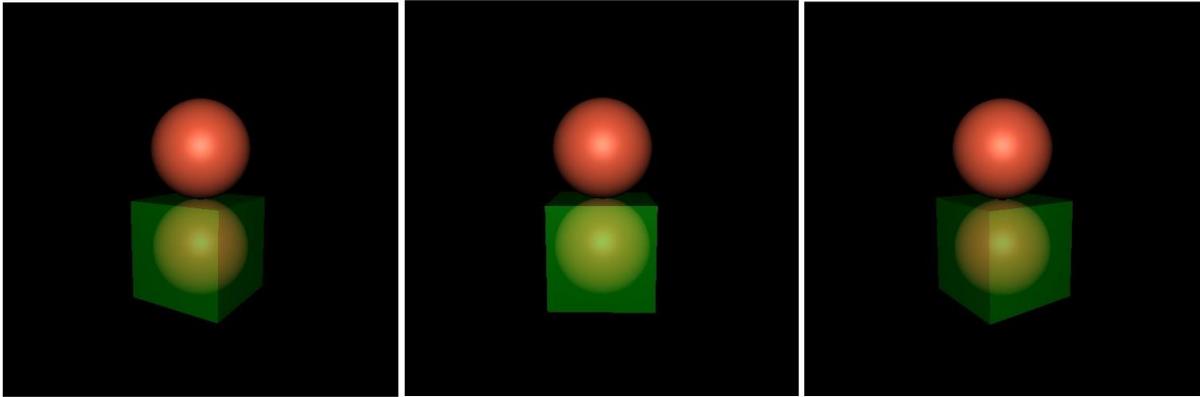


Figura 1: Modelo a implementar. Três frames da animação.

online, por exemplo assim:

```
<script type="module">
import * as THREE from 'https://unpkg.com/three@0.124.0/build/three.module.js';
// coloque o seu script aqui
</script>.
```

Comente o código adequadamente (sem exageros - poupe tempo!).

Não deve depender de nenhuma outra library nem de assets auxiliares. No zip só vem a folha de exame e o html.

FIM (Boa Sorte!)