



Computação Gráfica — 21020

Período de Realização

Consultar os prazos de entrega indicados pelos serviços.

Objetivos

O exame cobre potencialmente a totalidade da matéria lecionada.

A prova é composta por dois grupos de questões, 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.

Cada questão do grupo 1 vale 3.5 valores, somando um total de 14 valores. O grupo 2 vale 6 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 Exame. Exemplo: 123456SilvaExame.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.

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.

Grupo I

1. Considere os pixels que seriam acesos pelo algoritmo de ponto médio aplicado aos pontos extremos $A = (12, 1)$ e $B = (100, 5)$. Liste as coordenadas y de todos os pixels que são acesos com coordenada $y = 3$. Justifique. A sua justificação deve envolver um cálculo simples utilizando a equação da recta e as propriedades do algoritmo. Não deve tentar executar o algoritmo passo a passo. A elegância e simplicidade da justificação será tida em conta na cotação.
2. Utilize o algoritmo scan-line para preencher o polígono definido por $A = (1, 0)$, $B = (2, 0)$, $C = (4, 1)$, $D = (5, 3)$, $E = (2, 3)$, $F = (0, 2)$. Apresente a ET e AET em cada iteração, bem como os *pixels* de preenchimento. Represente o resultado final graficamente.
3. Explique porque são úteis as coordenadas homogéneas no cálculo das transformações geométricas afins. Em particular, demonstre que existe um tipo importante de transformação geométrica afim de \mathbb{R}^3 que não pode ser representada como uma transformação linear de \mathbb{R}^3 (ou seja não pode ser representada por uma matriz 3×3).
4. Considere a curva de Bézier com os pontos de controlo $P_0 = (0, 0)$, $P_1 = (2, 5)$, $P_2 = (9, 7)$, $P_3 = (5, 0)$.
 - a) Apresente a parametrização da curva. Calcule os pontos da curva que correspondem a $t = 1/3$ e $t = 2/3$.
 - b) Represente graficamente a curva (desenhada à mão).

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. No momento inicial as esferas são idênticas e estão colocadas uma directamente em cima da outra, tocando-se num ponto. À medida que o tempo passa a esfera superior altera o seu raio, que se reduz até zero e depois volta a crescer, num ciclo infinito (dica: utilize um seno ou coseno em função do tempo). Durante este escalamento a posição da esfera também se altera, de forma que ela esteja sempre a tocar a esfera inferior num único ponto (ver figura).

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.

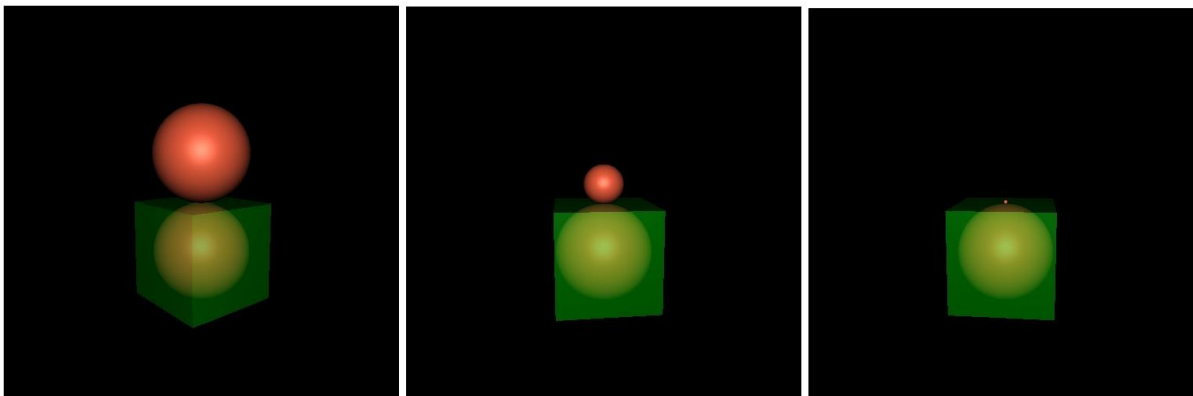


Figura 1: Modelo a implementar. Três frames da animação.

Formato de entrega:

O código deve ser escrito directamente num único ficheiro html, que 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 html com o código inline, e que deverá chamar o three.module.js num repositório 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