

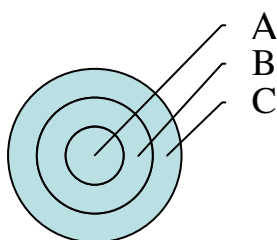
Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

Este Exame tem 9 perguntas e está cotado para 20 valores. Cada pergunta tem a seu valor expresso entre parêntesis a seguir ao seu número.

Responda às perguntas de uma forma sintética.

1 – (3 Valores) Implemente em Flash e ActionScript um programa que permita ler 4 pontos na área activa do documento em Flash, desenhe três circunferências designadas A, B e C, concêntricas, e em seguida calcule a área correspondente a zona B:



Resposta : O aluno deveria ter expresso sob a forma de uma representação algorítmica a resolução do problema (representação esta que podia ser de mais alto ou baixo nível), exemplo numa linguagem de mais alto nível:

1. Proceder a “leitura” usando o rato de 4 pontos (2D pares de pontos na forma $p_i = (x, y)$), o primeiro correspondente ao centro dos círculos, e mais um ponto por cada raio das circunferências que delimitam as áreas A, B e C.
2. Em seguida deveria calcular as distâncias entre o primeiro ponto e cada um dos outros pontos lidos usando a fórmula da distância entre dois pontos:

$$d_{cp} = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} .$$

3. Desenhar as três circunferências.
4. Finalmente calcular as áreas correspondente ao círculo mais interior (A) e ao círculo mais exterior usando a fórmula $Area = \pi * r^2$, em que r é cada uma das distâncias já calculadas no ponto anterior. Finalmente calcular o valor da área B usando

$$Area_B = Area_{Exterior} - Area_A .$$

Implementar em ActionScript o programa:

```
// ler os pontos que definem os círculos
//ler_ponto_central ponto=0
//ler_ponto_1 ponto=1
//ler_ponto_2 ponto=2
//ler_ponto_3 ponto=3

pc=new Object();
p1=new Object();
p2=new Object();
p3=new Object();
p=new Object();
var ponto=-1;
```

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

```

var r1;
var r2;
var r3;

function onMouseDown():Void {
    if (ponto==1){
        //le ponto central
        pc._x=_xmouse;
        pc._y=_ymouse;
        ponto=1;
        //trace("Entra...pc");
    }

    else if (ponto==2){
        //le ponto 1
        p1._x=_xmouse;
        p1._y=_ymouse;
        ponto=2;
        //trace("Entra...p1");
        //
        trace(p1._x);
    }

    else if (ponto==3){
        //le ponto 2
        p2._x=_xmouse;
        p2._y=_ymouse;
        ponto=3;
        //trace("Entra...p2");
    }

    else if (ponto==4){
        //le ponto 3
        p3._x=_xmouse;
        p3._y=_ymouse;
        ponto=0;
        //trace("Entra...p3");
        calcula_raios();
        desenha_circulos();
        calcular_area();
    }
}

//Calcular distâncias
function calcula_raios():Void{
    if (ponto==0) {
        trace("Entra...calcula_raios");
        r1=Math.sqrt((p1._x-pc._x)*(p1._x-pc._x)+(p1._y-pc._y)*(p1._y-pc._y));
        trace("r1="+r1);
        r2=Math.sqrt((p2._x-pc._x)*(p2._x-pc._x)+(p2._y-pc._y)*(p2._y-pc._y));
        trace("r2="+r2);
        r3=Math.sqrt((p3._x-pc._x)*(p3._x-pc._x)+(p3._y-pc._y)*(p3._y-pc._y));
        trace("r3="+r3);
    }
}

//Desenhar Circunferencias concentricas

```

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

```
function desenha_circulos():Void{
//trace("Entra...desenha_circulos");
lineStyle(1, 0, 100);
desenha_c1();
desenha_c2();
desenha_c3();
function desenha_c2():Void {
    moveTo(p2._x,p2._y);
    for (angulo=0;angulo<2*Math.PI;angulo+=0.1) {
        trace(angulo);
        trace(r2);
        p._x=pc._x+r2*Math.cos(angulo);
        p._y=pc._y+r2*Math.sin(angulo);
        trace(p._x);
        trace(p._y);
        lineTo(p._x,p._y);
    }
}
function desenha_c3():Void {
    moveTo(p3._x,p3._y);
    for (angulo=0;angulo<2*Math.PI;angulo+=0.1) {
        //trace(angulo);
        //trace(r3);
        p._x=pc._x+r3*Math.cos(angulo);
        p._y=pc._y+r3*Math.sin(angulo);
        //trace(p._x);
        //trace(p._y);
        lineTo(p._x,p._y);
    }
}
function desenha_c1():Void {
    moveTo(p1._x,p1._y);
    for (angulo=0;angulo<2*Math.PI;angulo+=0.1) {
        //trace(angulo);
        //trace(r1);
        p._x=pc._x+r1*Math.cos(angulo);
        p._y=pc._y+r1*Math.sin(angulo);
        //trace(p._x);
        //trace(p._y);
        lineTo(p._x,p._y);
    }
}
}
}

// Imprimir área
function calcular_area():Void{

area_exterior= Math.PI*r3*r3;
area_interior=Math.PI*r1*r1;
area=area_exterior-area_interior;
```

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

```
trace("area interior (B) = "+area);
}
```

2 - (2 valores) O que é um “sprite”? Quais as suas propriedades e funcionalidades mais importantes? Dê exemplos da aplicação dos mesmos em aplicações gráficas. Como implementaria em Flash uma estrutura com comportamento semelhante a um “Sprite”?

Resposta: Em computação gráfica e desenvolvimento de VideoJogos um sprite é uma imagem ou animação bidimensional integrada numa cena com determinado comportamento (wikipédia). Originalmente os sprites foram inventados como um método de compôr rapidamente várias imagens que pudessem ser reproduzidos em videoJogos por hardware especial. Hoje um sprite representa concretamente um objecto gráfico tipicamente 2D com vários comportamentos gráficos (animações ou imagens). Exemplo deste tipo de objecto gráfico é por exemplo o característico Sonic que faz ainda hoje as delícias de grande parte dos jogadores mais jovens. Os comportamentos como andar, saltar, correr explodir são comportamentos que podem estar associados a um determinado Sprite.



Em Flash/ActionScript podemos criar o comportamento de um sprite através de composições de Movies, convenientemente etiquetados, estes movies reproduzem diversos comportamentos gráficos tipicamente animados como um personagem a “andar”. A reprodução de um dos comportamentos suportado pelo sprite em Flash/ActionScript pode ser feito por uma simples chamada ao movie que possua a animação pretendida. Muitas vezes os sprites estão organizados graficamente em “sprite Sheets” ou folhas com sprites:



3 – (2 valores) Explique como criar uma imagem composta da “menina” sobre uma foto da cidade de Budapeste, substituindo o fundo verde pela foto. Considere que a cor do verde é conhecida (RGB) e que é uniforme (todos os pixels verdes são da mesma cor – mesmo RGB).

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____



Foto cidade de Budapeste



Foto menina

Resposta: A técnica de substituição de uma cor por uma outra imagem, animação ou vídeo é genericamente designada em vídeo por cromakeying. No entanto existem muitas formas de conseguir o resultado pretendido: Cromakey, GreenScreen, LumaKey, AlfaKey etc. No caso da pergunta, claramente a técnica a usar deveria ser um GreenScreen, ou seja a substituição de uma determinada tonalidade de verde (transformando estes pixels em transparentes) pela imagem da cidade de Budapeste. O algoritmo para o fazer poderia ser descrito da seguinte maneira:

1. Estabelecer acesso às duas imagens bitmap. Criar dois ciclos embricados que permitam ler todos os pixels de cada um dos bitmap. Um a um, seriam lidos os três ou quatro canais de cor de cada pixel (o flash suporta ambos quer o sistema ARGB quer RGB) de cada imagem.
2. Para cada pixel lido comparar a cor com o intervalo permitido de tolerância de cor (não exigido na pergunta porque a cor verde é homogénea (e normalmente obtida por ferramenta de amostragem). Se a cor do pixel se encontrar nesse intervalo então substituir esse pixel por o pixel correspondente na fotografia de Budapeste (a representada na figura da pergunta).
3. Repetir o passo 2 para todos os pixels do bitmap.

Nota: se ambas as fotografias tiverem dimensões diferentes duas hipóteses se colocam:

- Ou usar as imagens com as dimensões reais e portanto a imagem de fundo não se verá totalmente ou não existirão pixels para preencher completamente o fundo a substituir.
- Ou tentar compatibilizar as dimensões das fotos por exemplo interpolando ou estimando valores para os pixels (compatibilizar a dimensão das fotos).

Revisão da resposta do aluno:

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

4 – (2 Valores) Identifique cinco dos principais princípios de animação criados e popularizados pela “Escola Disney”, mas usados extensivamente na criação de animação com meios digitais. Explique cada um dando um exemplo.

Resposta: Os estúdios Disney nas primeiras décadas do século passado criaram uma “escola” de como fazer animação com sucesso. Em 1981 os animadores da Disney [Ollie Johnston](#) e [Frank Thomas](#) no livro *The Illusion of Life: Disney Animation* descrevem os 12 princípios fundamentais da animação. Esta forma de animar ainda hoje é usada para animar personagens em filmes de animação digitais. Entre os muitos princípios popularizados pelos animadores da Disney temos:

1. Princípio da antecipação

Este princípio, estabelece que na animação de um determinado movimento de um personagem devemos preceder o movimento em si de um outro que permita antecipar qual o movimento que se vai realizar.



2. Princípio do exagero do movimento.

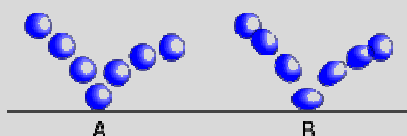
Este princípio estabelece que para cada movimento executado se exagere o mesmo para dar ênfase a narrativa visual que inclui o movimento.

3. Arcos - Descrição do movimento de personagens sempre em arco (não linearidade)

Na natureza o movimento nunca é linear. A melhor forma de o descrever é em arco. Por exemplo a ponta do pé a executar um chute numa bola descreve uma trajetória em arco.

4. Squash and Stretch

O objectivo deste principio é dar a sensação de peso e flexibilidade aos objectos animados/desenhados. Pode ser aplicado a objectos como uma bola ou a construções animadas mais complexas como a musculatura humana da face. Aplicado de forma extrema a figura esticada ou esmagada a um nível exagerado pode ter um efeito comico. Muito importante é no entanto que o volume do objecto esmagado ou esticado não se altera.



Um exemplo diferente pode ser uma pessoa gorda a andar em que deve ser evidente o movimento da gordura sobre efeito da gravidade e da velocidade a que a pessoa se desloca.

5. Easing in / Easing out (ou Slow in Slow out) - Alteração da velocidade de objectos ao longo de um percurso (ou seja com recurso a aceleração/desaceleração): O movimento do corpo humano e da maioria dos objectos precisa de tempo para acelerar e desacelerar. Ou seja todo o objecto para atingir uma determinada velocidade sofre uma aceleração ou desaceleração, podendo este

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

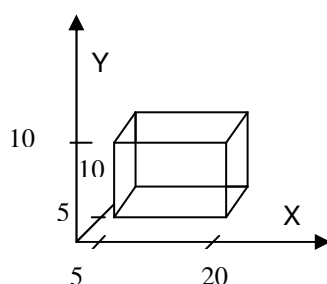
Numero: _____ Nome : _____

valor da velocidade ser descrito por curvas como splines. Em animação devemos usar este princípio para tornar verosímil animação do movimento. Quer seja um objecto inanimado como um carro ou um movimento de um ser humano. Também se designa por “slow in” “slow down” e correspondia em animação tradicional a possuir mais frames junto a frame de início do movimento e junto a frame final do movimento.

Para completar os doze princípios enumerados pelos autores referidos eram:

1. [Squash and Stretch](#) – define a rigidez e massa de um objecto distorcendo a sua forma e acção.
2. [Timing and Motion](#) – Espaçamento das acções de forma a definir o peso e tamanho de objectos e personalidade dos personagens.
3. [Anticipation](#) – Preparação da acção
4. [Staging](#) – Apresentar uma ideia que seja evidentemente clara
5. [Follow Through and Overlapping Action](#) – A continuidade entre o termino de uma acção e o inicio da próxima.
6. [Straight Ahead Action and Pose-to-Pose Action](#) – As duas formas de criar movimento tradicionalmente.
7. [Slow In and Out](#) – Espaçamento das frames de forma obter o “timing” apropriado do movimento.
8. Arcs – O percurso visual de uma acção com movimento
9. [Exaggeration](#) – Acentuar a ideia por via do design e da acção.
10. [Secondary Action](#) - the action of an object resulting from another action
11. [Appeal](#) – criar um desenho e acção que a audi~encia goste.
12. [Personality](#) Dar personalidade aos personagens

5 – (3 valores) Represente a matriz genérica de transformação a 3D capaz de executar a operação de rotação sobre o centroíde da figura e em torno do eixo dos x (horizontal direito) de um ângulo de 30 graus, observe a figura Figura:



Resposta: a Descrição matricial do movimento pedido através de uma matriz de tranformação genérica resultaria da multiplicação de três matrizes representando a primeira a **Translação (deslocação) do centróide da caixa para a origem do sistema global de eixos**, uma **segunda correspondente a rotação pretendida da caixa** e finalmente uma **terceira que repõem a caixa na sua posição original**. Ou seja para cada ponto (vertice) que define o objecto deveria ser aplicada a transformação:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = T_{origem} * R_x * T_{posição_original} * \begin{bmatrix} x & y & z & 1 \end{bmatrix}$$

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

Para os valores dados seria,

$$\begin{vmatrix} x' & y' & z' & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & -12.5 \\ 0 & 1 & 0 & -7.5 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(30) & -\sin(30) & 0 \\ 0 & \sin(30) & \cos(30) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 & 12.5 \\ 0 & 1 & 0 & 7.5 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x & y & z & 1 \end{vmatrix}$$

6 –(2 valores) – Relacione o conceito de animação em flash com o conceito de **easing in** e **easing out** da animação tradicional, explicando o conceitos destes últimos. Dê exemplos, e/ou use gráficos para ilustrar a sua resposta.

Resposta: Easing é uma forma de mover um objecto de uma posição para outra com alteração de velocidade do objecto durante o percurso.

‘Easing Out’ → é forma de easing que permite ao objecto diminuir a velocidade conforme este se aproxima do seu destino.

‘Easing In’ → é forma de easing que permite ao objecto aumentar a velocidade conforme este se aproxima do seu destino.

Um objecto sem easing apresenta velocidade constante.

Exemplo de implementação de uma função em actionscript que permita “easing”.

```

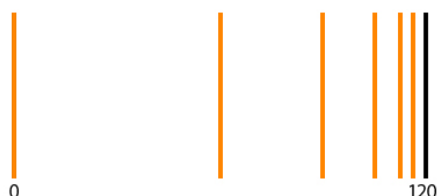
var velocidade:Number = 2;
ease = function( Objecto:MovieClip , PosicaoX:Number ){
    var PosicaoPrevia:Number = Objecto._x;
    Objecto.onEnterFrame = function(){
        this._x = PosicaoX - (PosicaoX - this._x ) / velocidade;
        if( this._x == PosicaoPrevia ){
            this._x = to;
            delete this.onEnterFrame;
        }
        Posicao_previa = this._x;
    }
}

```

O valor da variavel velocidade deve ser sempre maior que 1, porque representa o número de divisões que fazemos da “distância” que falta percorrer em frames, o PosicaoX representa a posição x para onde queremos mover o objecto. A figura ilustra o conceito para 120 frames.

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____



7- (2 Valores) Compare as características da API gráfica avançada como o XNA (DirectX 9C) e o ambiente gráfico do Flash/ActionScript em termos de aplicações gráficas no domínio dos videojogos 2D. Enumere vantagens e inconvenientes nas diversas vertentes que abordar.

Resposta:

O ambiente gráfico do Flash/ActionScript está associado ao Flash/Flex [A versão CS3 é a mais actual à data] e aos seus players (que são gratuitos e vêm embebidos na generalidade dos browsers para pc e Mac, em dispositivos moveis como telemóveis e consolas de jogos).

O flash usa gráficos vectoriais 2D como base do seu ambiente multimédia, permitindo “footprints” para as aplicações/programas muito pequenos comparativamente a outros ambientes. Este ambiente com mais de dez anos de existência sofreu uma evolução muito grande desde o seu surgimento e é considerado já um standard de facto para animações 2D na Internet (web).

As últimas versões do Flash permitem trabalhar com vídeo, som (mp3) e interagir com ambientes servidor na Internet de forma cada vez mais avançada

estando a Adobe (empresa detentora do Flash) a promover ambientes programáticos baseados na tecnologia Flash com muitas funcionalidades e com elevada interoperabilidade com outros programas e ambientes, caso do Flex. O flash é fácil de aprender, possui um ambiente de desenvolvimento interactivo de fácil aprendizagem e as aplicações geradas podem ser distribuídas para os mais diversos tipos de plataformas com muito poucas adaptações.

Como os gráficos e animações geradas são pseudo-interpretadas a sua velocidade é limitada comparativamente a ambientes gráficos nativos como o DirectX e OpenGL. É um ambiente gráfico 2D que apresenta como pontos fortes a portabilidade, estabilidade e facilidade de desenvolvimento.

O XNA baseado na API DirectX9 é desenvolvido e distribuído pela Microsoft para ser utilizado como ambiente de desenvolvimento de video jogos quer a 2D quer a 3D nas suas linhas de sistemas operativos (Win98, win2000, WinXP, Windows Vista, Xbox e Zune ...) só podendo ser utilizado em projectos estinados a estas plataformas. A API DirectX 9.C é multi-dispositivo (ou seja possui recursos para a parte gráfica, som, jogos multiplayer, dispositivos de input – como joystick, ou outros). Tem evoluído regularmente e actualmente é sado não só em PC’s, como em consolas (xbox) e a curto prazo dispositivos móveis é GRATUITO mas possui como elementos a desfavor o não ser ‘open source’ ou Multi-Plataforma fora dos sistemas operativos da Microsoft. É uma api gráfica muito avançada quer para 2D quer para 3D que exige que se saiba programar num ambiente que possua capacidade de usar o DirectX, caso da generalidade dos ambientes gráficos de desenvolvimento para PC. O facto de não ser open-source limita a sua generalização a outros ambientes operativos. Apresenta como pontos fortes ser um ambiente robusto a 3D, possuir recursos para som, vídeo, ambientes ligados em rede, e a generalidade dos fabricantes de placas de vídeo avançadas para PC e portáteis optimizarem as suas placas para este API. O XNA acrescenta a esta API de base um ambiente que permite

Engenharia Multimédia	2º Ano - 2º Semestre
Correcção do exame de SMII e TAGI	Exame
Data :24-07-2008	Duração : 45 Minutos
Parte Teórica	Prof. : Jorge Mota

Numero: _____ Nome : _____

o desenvolvimento de jogos de forma fácil pela existencia de um ambiente de Game Studio que integra para além das funcionalidades gráficas, um gestor de conteúdos e bibliotecas que executam a maioria das funcionalidades requeridas a um motor de jogo.

8 - (2 Valores) Crie um procedimento em actionscript que permita mover um sprite como o da figura fig. 1 ao longo de um caminho do tipo da fig 2:



Fig. 1

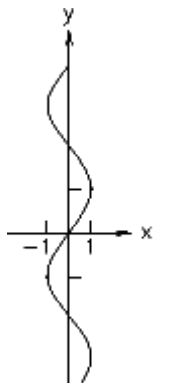


Fig 2.

Resposta:

Versao simplificada

```
angulo = -2*Math.PI;
centroX = 200;
limite = 50;
yvelocidade = 1;
xvelocidade = .05;
```

```
onEnterFrame = function(){
    nave._y += yvelocidade;
    nave._x = centroX+ Math.sin(angulo) * limite;
    angulo += xvelocidade;
}
```

Versao com desenho do trajecto

```
angulo = 0;
centroY = 200;
limite = 50;
xvelocidade = 0.05;
yvelocidade = 1;
```