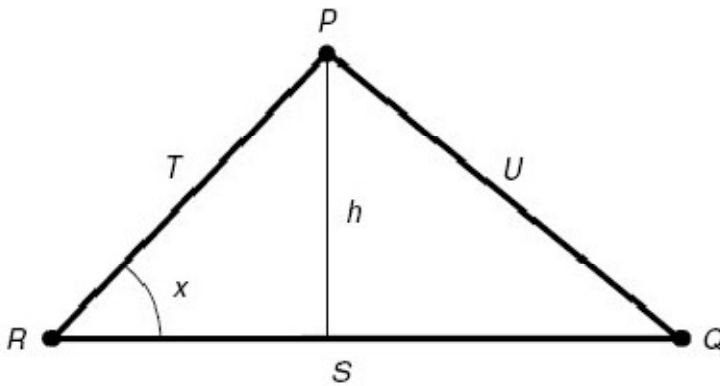


Pergunta 1

(3 valores) Implemente em Flash e ActionScript um programa que permita ler 3 pontos na área activa do documento em Flash, desenhe o triângulo e em seguida calcule a sua área e a imprima na área de (3 valores).



Notas:

- A área de um triângulo é dada por: $A = sh/2$
- Pode considerar uma ordem pré-definida para a digitação (click) dos pontos para facilitar o Cálculo do h.
- Pode utilizar o método onMouseDown para activar a função do tecla X da consola.

RESPOSTA:

```
// Desenha Triangulo
//Ponto é uma variavel que indica que ponto estamos a ler
var Ponto:Number=0;
var x1:Number=0;
var x2:Number=0;
var x3:Number=0;
var x4:Number=0;
var y1:Number=0;
var y2:Number=0;
var y3:Number=0;
var y4:Number=0;

_root.onMouseDown = function () {
```

```
if (Ponto==0) {  
    x1=_xmouse;  
    y1=_ymouse;  
    Ponto=1  
}  
else if (Ponto==1) {  
    x2=_xmouse;  
    y2=y1;  
    Ponto=2;  
  
} else if (Ponto==2) {  
    x3=_xmouse;  
    y3=_ymouse;  
    Ponto=0;  
    CalculaArea();  
}  
}  
//Desenha Triangulo  
function desenha():Void {  
    //tipo de linha  
    lineStyle(1, 0, 100);  
    moveTo(x1,y1);  
    lineTo(x2,y2);  
    lineTo(x3,y3);  
    lineTo(x1,y1);  
}  
//Calcula a Area do Triangulo
```

```

function CalculaArea():Void {

    desenha();

    Calcula();

}

//Calcula a area

function Calcula():Void {

    var s:Number=0;

    var h:Number=0;

    var area:Number=0;

    x4=x3;

    y4=y1;

    s=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));

    h=Math.sqrt((x4-x3)*(x4-x3)+(y4-y3)*(y4-y3));

    area=s*h/2;

    Trace("Area =" +area);

}

```

Pergunta 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:

Um sprite é objecto gráfico que apresenta vários comportamentos gráficos. Exemplo o sprite “Sonic” com comportamentos com {“Parado”, “Correr”, “Saltar”, “Explodir”}. Normalmente este tipo de objecto gráfico tem aplicação em videoJogos (nomeadamente 2D), simulação, Conteúdos interactivos para aplicação como eLearning, ou mesmo webdesign. Para implementar uma estrutura tipo sprite podíamos usar um movie contentor (“Sprite”) e dentro deste movie vários movies com os comportamentos gráficos pretendidos. Estes devem estar organizados na Timeline de forma que possamos chamar cada um individualmente. Para tal podemos usar etiquetas ou mesmo o numero da frame. Para evitar passar para outro comportamento podemos usar funções como stop();gotostart().

Pergunta 3

(2 Valores) As consolas portáteis e os dispositivos móveis, tais como, telemóveis de última geração autenticamente invadiram o mercado, sendo hoje um equipamento de uso corrente. O desenvolvimento de conteúdos e aplicações para este tipo de dispositivos, nomeadamente em português, no entanto continua a ser ainda do domínio de uns poucos especialistas, constituindo-se como uma oportunidade para uma nova geração de profissionais que a queiram aproveitar.

Descreva de forma comparativa com o ambiente de um PC, que limitações e que tipo de abordagem poderiam ser seguidas para o desenvolvimento de conteúdos, usando ActionScript/Flash para um dispositivo como a consola PSP (PlayStation Portable).

Resposta:

A PSP tem um browser equivalente ao player 6/VAM 1.2. Ora esta versão não implementa funcionalidades como XML, Video constituindo portanto uma limitação a determinado tipo de aplicações. Actualmente o Player 9 para PC usa uma VAM 3.0 com funcionalidades como Video nativo, comunicação Scokets em XML, Uma organização do programa 100% object oriented e suporte para bibliotecas extensas de componentes. Outra limitação do player da PSP é o limitado número de botões/funções interacção comparativamente a um PC. Neste último temos teclado "Full size" Rato, WebCam completamente acessível e um ecran com resolução limitada. A velocidade de processamento em dispositivos como a PSP é muito limitada e a capacidade de armazenamento e o acesso a esta também muito limitada.

Pergunta 4

(2 valores) Explique como converter uma imagem bitmap colorida usando um padrão 24 bits (RGB), numa imagem Bitonal (Preto e Branco). (3 Valores).

Resposta:

Para converter uma imagem bitmap colorida a 24 bits (RGB) numa imagem bitonal ou seja apenas a preto e branco, podemos proceder da seguinte forma:

-Criar um procedimento que leia pixel a pixel cada elemento da imagem bitmap e converta as suas componentes R, G e B em níveis de cinzento usando uma média pesada semelhante à usada no sistema televisivo NTSC/PAL para converter imagem cor em PB (luminância):

Valor do nível de cinzento para cada pixel= $R*0,2+G*0,7+B*0,1$

- Em seguida aplicar um procedimento de comparação ao valor calculado para os seguintes intervalos:

[0,127] e [128,255].

Se o valor estiver no primeiro intervalo então colocamos o valor de 0 em cada componente RGB,.

Se o valor estiver no segunndo intervalo colocamos o valor 255 nas componentes.

A imagem assim obtida é bitonal possui apenas o Branco Branco ou o Preto Preto.

Pergunta 5

(3 valores) Analise o seguinte código em ActionScript, que implementa o comportamento de uma bola, num campo, usando um vector velocidade representado pelas componentes xv e yv:

```
0: // Programação do Comportamento da Bola.
1: onClipEvent(load){
2: //Criar as variáveis que representam o vector velocidade
3: var xv=5;
4: var yv=5;
5: }
6: //Este código é executado sempre que entra numa frame
7: onClipEvent(enterFrame){
8: //Verificar colisões com os limites do campo
9: if((this._y+this._height)>Stage.height){
10: //Recolocar a bola no fim do campo
11: this._y=Stage.height-this._height;
12: //Activar movimento da Bola para cima
13: yv=yv*(-1);
14: }
15: // Verificar colisões com os limites do campo
16: if(this._y<0){
17: //Colocar a bola no canto superior
18: this._y=0;
19: //Activar o movimento da bola para cima
20: yv=yv*(-1);
21: }
22: //Verificar colisões com os limite lateral
23: if((this._x+this._width)>Stage.width){
24: //Recolocar a bola no fim do campo
25: this._x=Stage.width-this._width;
26: //Activar movimento da Bola para cima
27: xv=xv*(-1);
28: }
29: //verificar se ultrapassa os limites laterais do campo
30: if((this._x+this._width<0) || (this._x>Stage.width)){
31: //Somar um ponto ao jogador
32: if(this._x+this._width<0){_root.score++;}
33: //Colocar a bola a meio campo
34: this._x=Stage.width/2;
35: //Alterar a direcção da bola
36: xv=xv*(-1);
37: }
38: //Incrementar a posição da Bola
39: _x=_x+xv;
40: _y=_y+yv;
41: }
```

```

42: // Programação do Comportamento da Raquete
43: onClipEvent(enterFrame){
44: // Se a tecla UP é premida move a raquete da esquerda para cima
45: if(Key.isDown(Key.UP)){
46: this._y= this._y -5;
47: }
48: // Se a tecla DOWN é premida move a raquete da esquerda para Baixo
49: if(Key.isDown(Key.DOWN)){
50: this._y= this._y +5;
51: }
52: //Se a Raquete ultrapassa a área do Campo, então é colocada
novamente no 53: //fundo do campo
54: if((this._y+this._height)>Stage.height){
55: this._y=Stage.height-this._height;
56: }
57: //Se ultrapassa o campo parte superior então move para a parte superior
58: if(this._y<0){
59: this._y=0;
60: }
61: //Verificar a colisão da bola com a raquete
62: if(this.hitTest(_root.Bola)){
63: //Altera a direcção da bola
64: _root.Bola.xv=_root.Bola.xv*(-1);
65: //Move a bola para os limites da raquete
66: _root.Bola._x=this._x+this._width;
67: }
69: }

```

Implemente o efeito de Zona de Aceleração e Variação de direcção da bola nas extremidades da raquete conforme representado na figura.

Resposta:

```

61: //Verificar a colisão da bola com a raquete
62: if(this.hitTest(_root.Bola)){
If ((_root.Bola._x>this._x-this._height/4) or (_root.Bola._x>this._xthis._height/4*3)){
//Altera a direcção da bola
_root.Bola.xv=_root.Bola.xv*(-random()*3);
else
65: //Move a bola para os limites da raquete
66: _root.Bola._x=this._x+this._width;
67: }

```

Notas para facilitar a implementação :

- Numere as linhas de código para melhor se perceber as ALTERAÇÕES QUE FIZER

Pergunta 6

(3 valores) Represente a matriz genérica de transformação de **ROTAÇÃO em 2D** (MTRX(θ)), de forma que esta permita por uma simples operação de transformação, **rodar na posição actual** o ponto P(x,y,1) de um ângulo θ :

Resposta:

$$\begin{matrix} \text{Matriz de transformação} \\ \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\text{sen}(\alpha) & 0 \\ \text{sen}(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -d_x & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\text{sen}(\alpha) & d_x(1-\cos(\alpha)) + d_y\text{sen}(\alpha) \\ \text{sen}(\alpha) & \cos(\alpha) & d_y(1-\cos(\alpha)) - d_x\text{sen}(\alpha) \\ 0 & 0 & 1 \end{bmatrix} \\ T(d_x, d_y).R(\alpha).T(-d_x, -d_y) \end{matrix}$$

Em vez θ usamos α para representar o ângulo de rotação.

Pergunta 7

(2 valores) Relacione o conceito de animação em flash com o conceito de **easing in** e **easing out** da animação tradicional, explicando o que é isto de easing in e out em termos de animação de movimentos. Dê exemplos, e/ou use gráficos para ilustrar a sua resposta(2 valores).

Resposta:

Na natureza não existem movimentos lineares puros. Os movimentos tem um determinado tempo de aceleração ou desaceleração. Com a animação baseada em Keyframes, em que o movimento é obtido por interpolação entre duas keyframes usamos formulas de interpolação não linear para simular estes movimentos, easing in e easing out. Exemplo uma bola a cair sobre efeito da gravidade (easing out).

Pergunta 8

(2 Valores)Compare as características da API gráfica avançada como 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.(2 valores)

Resposta:

O ambiente gráfico do Flash/ActionScript está associado ao Flash [versão 8 é a mais actual a 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) [versão 9.0] a data.

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 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 API DirectX9 é desenvolvido e distribuído pela Microsoft para ser utilizado nas suas linhas de sistemas operativos (Win98, win2000, WinXP, Windows Vista, Xbox ...) só podendo ser utilizado em projectos destinados a estas plataformas. É um API 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 usado não só em PC's, como em consolas (xbox) 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 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.

Pergunta 9

(2 valores) Crie um procedimento em actionscript que permita carregar um textura Jpeg da pasta .\imagens e a mostre no palco do Flash.(2 valores)