ISTEC 2010 TGAI – 2º SEMESTRE	CURSO DE ENGENHARIA MULTIMÉDIA
Data 15-07-2010	Duração: 60 minutos
Professor: Jorge Mota	EXAME ESCRITO CORRECÇÃO

TEMA 1: Processing

Pergunta 1 - (3 Valores) Implemente em Processing um método (função) que permita determinar se dois objectos em movimento, num campo, *O1* e *O2* (bolas) estão em colisão.

Notas para a implementação:

- Os dois objectos têm raio *r1* e *r2*
- *As esquações que descrevem a trajectória dos objectos são dadas por:*

Objecto 1:

$$\begin{cases} ox_1 = ox_1 + vxo_1 \\ oy_1 = oy_1 + vyo_1 \end{cases}$$

Objecto 2:

$$\begin{cases} ox_2 = ox_2 + vxo_2 \\ oy_2 = oy_2 + vyo_2 \end{cases}$$

Em que Vxo1,Vxo2,Vyo1,Vyo2 representam as componentes do vector velocidade para o objecto 1 e 2.

Resolução:

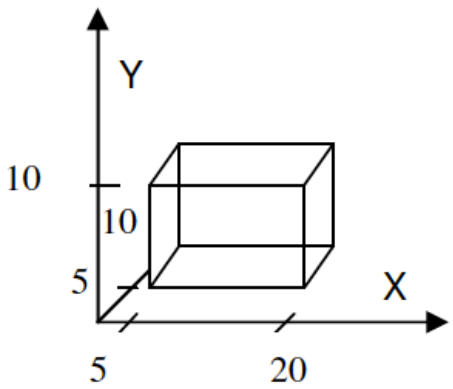
```
// Exame de TGAI Setembro
// Correcção
// Jorge Mota, 2010
// Pergunta 1 - Resolvida com programação simples

  float r1; // raio1
  float r2; // raio 2
  float o1x,o2x,o1y,o2y;
  float vxo1,vxo2,vyo1,vyo2;
  color c;

//Move as bolas no canvas  na janela gráfica
  void move() {
    o1x += vxo1; // Incremento x objecto 1
    o1y += vyo1; // Incremento y
    o2x += vxo2; // Incremento x objecto 2
    o2y += vyo2; // Incremento y

    // Detecção com as paredes horizontais
    if (o1x > width || o1x < 0) {
      vxo1 *= - 1;
    }
```

```
    if (o2x > width || o2x < 0) {
      vxo2 *= - 1;
    }

    // Detecção com as paredes verticais
    if (o1y > height || o1y < 0) {
      vyo1 *= - 1;
    }
     if (o2y > height || o2y < 0) {
      vyo2 *= - 1;
    }
  }

  // Função de realce

  void highlight1() {
    c = color(0,150);
  }


  // Desenhar a bola
  void desenha_bola() {
    stroke(0);
    fill(c);
    ellipse(o1x,o1y,r1*2,r1*2);
    ellipse(o2x,o2y,r2*2,r2*2);
     // Repor a cor da bola
    c = color(100,50);
  }

  // Função que devolve verdadeiro ou falso se as bolas se interceptam
  // Testa se a distnacia entre os dois centros é menor que a soma dos raios das bolas

  boolean intersecta(float r1, float x1, float y1, float r2, float x2, float y2 ) {

    // Os objectos tambem podem ser passados como parametros
    float distancia = dist(x1,y1,x2,y2); // Calcular a distância entre centros

    // Compara a distancia entre as soma dos raios
    if (distancia < r1 + r2) {
      return true;
    } else {
      return false;
    }
  }


void setup() {
  size(400,400);
  color c = color(100,50);
  smooth();

  // Initialize bolas
  r1=50;
  r2=30;
  vxo1 = random( -5,5);
  vxo2 = random( -5,5);
  vyo1 = random( -5,5);
```

```
  vyo2 = random( -5,5);
  o1x=200;
  o2x=200;
  o1y=200;
  o2y=200;

}

void draw() {
  background(255);
  // Mover e mostrar as bolas
  move();


  if (intersecta(r1,o1x,o1y,r2,o2x,o2y)) {
    highlight1();
   }

  desenha_bola();

}
```

TEMA 2: Transformações no espaço

Pergunta 2 – (3 valores) A seguinte expressão representa a matriz genérica de transformação a 3D, capaz de executar a operação de rotação sobre o centroíde da figura (Paralelipipedo), em torno do eixo dos x (horizontal direita) de um ângulo de 30 graus, conforme ilustrado na figura:



$$\begin{vmatrix} x' & y' & z' & 1 \end{vmatrix} = T_{origem} * R_x * T_{posição\_orginal} * \begin{vmatrix} x & y & z & 1 \end{vmatrix}$$

Para os valores dados seria,

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

Represente em MATLAB esta operação passo a passo.

Resposta:

```
>>To=[1 0 0 -12.5; 0 1 0 -7.5; 0 0 1 -5; 0 0 0 1]
>>Rx=[1 0 0 0 ; 0 cos(30) –sen(30) 0; 0 sen(30) cos(30) 0; 0 0 0 1]
>>Tp=[1 0 0 12.5; 0 1 0 7.5; 0 0 1 5; 0 0 0 1]
>> MT=To*Rx*Tp
```

TEMA 3: Cor

Pergunta 4 – (2 valores) Diga como poderia codificar a seguinte cor, representada pelas suas componentes RGBA, de 32 Bits, em Processing?

R←0xAA
G←0x99
B←0x10
A←0xFF

Resposta:

Em processing se pretendermos codificar o canal alfa temos de converter a representação Hexadecimal para decimal. Para uma representação apenas RGB bastaria o seguinte:

```
Color cor;

void setup()
{
cor=#AA9910;
// para atribuir o alfa teriamos que o fazer nos commandos de cor fill, background e stroke mas em decimal

fill(cor, 255);

}
```

Pergunta 5 – (2 valores)

Considere que a seguinte formula representa a luminância de uma cor representada pelas suas componentes RGB :

Y=0.2999R+0.59G+0.114B

Implemente em Processing um programa que permita converter um imagem bitmap colorida em monocromática usando esta equação.

Resposta:

```
// Resposta Pergunta 5 do teste de TGAI JULHO 2010 ISTEC 2010

// Implemente em Processing um programa que permita converter um imagem bitmap colorida em monocromática usando esta equação.
// Y=0.2999R+0.59G+0.114B
```

```
// declarar um objecto da classe imagem

PImage imagem;



void setup() {
  imagem = loadImage("Foto.jpg");
  // inicialiar a janela gráfica capaz de mostrar a dimensão da imagem
  size(imagem.width,imagem.height);
  // Esta foto deve estar na pasta do programa em processing uma numa subpasta chamada data
   // evita que a função Draw se repita
  noLoop();
}

void draw() {
 loadPixels();

 // Devemos chamar o loadPixels da classe PImage já que vamos ler os pixeis.
 imagem.loadPixels();
 for (int y = 0; y < height; y++ ) {
   for (int x = 0; x < width; x++ ) {
     int loc = x + y*width;
     // The functions red(), green(), and blue() pull out the three color components from a pixel.
     float r = red(imagem.pixels [loc]);
     float g = green(imagem.pixels[loc]);
     float b = blue(imagem.pixels[loc]);

     // Usa a formula da luminâcia para converter o RGB de cada pixel em Y
     //Coloca os três canais com o mesmo valor

     r=r*0.2999+0.59*g+0.114*b;
     g=r;
     b=r;

     // Atribui o novo valor a cada pixel da imagem
     imagem.pixels[loc] = color(r,g,b);
   }
 }

 updatePixels();
 //Desenha a imagem em tons de cinza para a formula dada
 image(imagem, 0,0);
}
```

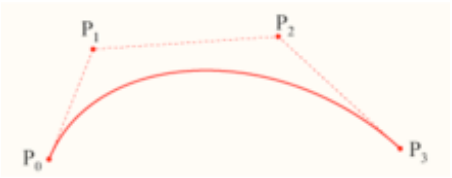
TEMA 4: Rasterização de Curvas

Pergunta 6 – (2 valores) Defina de forma simples spline?

Resposta:

Uma **spline** é uma curva definida matematicamente por dois ou mais *pontos de controle*. Os pontos de controle que ficam na curva são chamados de *nós*. Os demais pontos definem a tangente à curva em seus respectivos *nós*. Por exemplo, a curva de Bézier definida pelos pontos (A, B, C e D) é delimitada pelos nós A e D e nesses nós, a curva é tangente ao vectores AB e DC respectivamente. Variando as posições dos pontos B e C, a curva apenas varia sua inclinação, mas continua passando pelos pontos A e D.
As splines podem ser divididos em duas categorias:

- **Splines de interpolação** que passam por todos os pontos de controle
- **Splines de aproximação** que passam perto de todos os pontos de controle



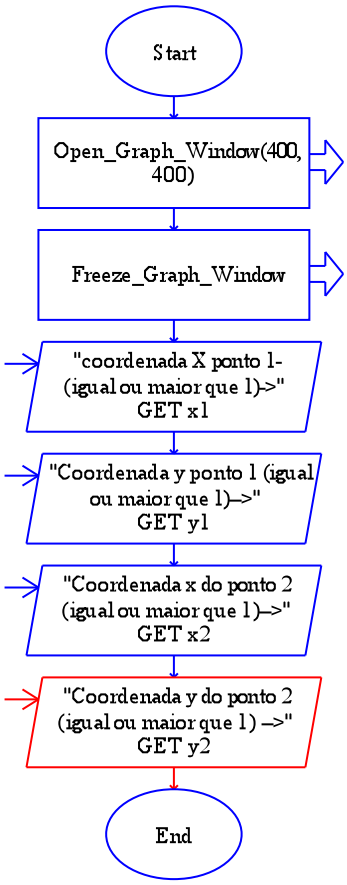
TEMA 5 : Raptor

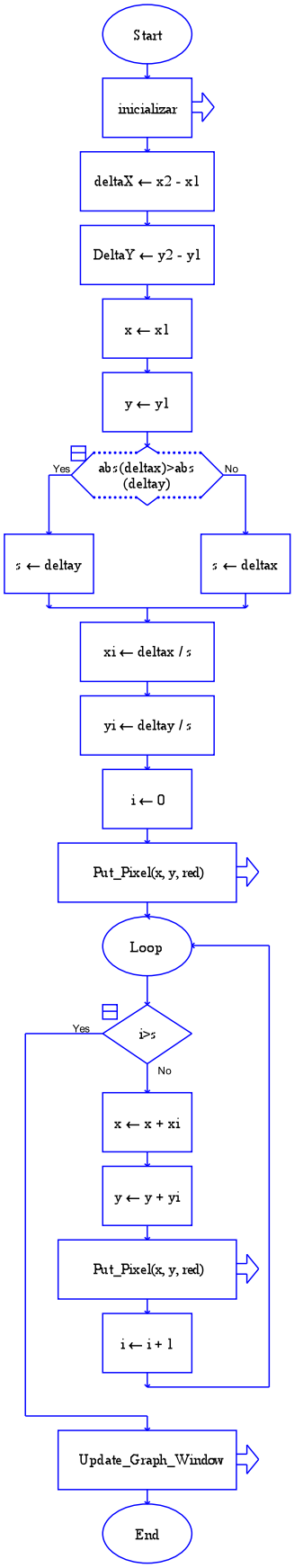
Pergunta 7 – (3 valores) Implemente em Raptor e RaptorGraph um programa que rasterize uma linha no primeiro octante usando um algoritmo DDA.

Resposta:

O algoritmo DDA é conhecido pelo nome de **D**igital **D**ifferencial **A**nalyser, ou analisador Diferencial Digital, e baseia-se na aplicação directa das formulas que descrevem uma linha recta no plano. Assim, para traçar um segmento de recta que vai do ponto P1 ao ponto P2, pode considerar-se o seguinte algoritmo em linguagem de alto nível:

1 – Pinta-se o pixel do ponto P1, isto é o de coordenadas (x1,y1) e atribuem-se as variaveis de trabalho (x,y) os valores (x1,y1).
2 – Passa-se para o pixel seguinte (x,y) em que x← x + 1 e y ←y+m, e pinta-se o pixel.
3 – Repete-se a passagem 2 até que o ponto P2 seja alcançado.

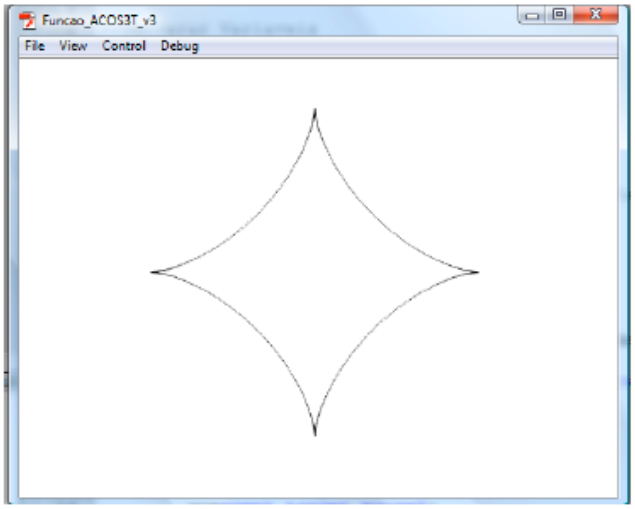



Start
inicializar
deltaX ← x2 - x1
DeltaY ← y2 - y1
x ← x1
y ← y1
abs(deltax)>abs (deltay)
Yes
No
s ← deltay
s ← deltax
xi ← deltax / s
yi ← deltay / s
i ← 0
Put_Pixel(x, y, red)
Loop
i>s
Yes
No
x ← x + xi
y ← y + yi
Put_Pixel(x, y, red)
i ← i + 1
Update_Graph_Window
End

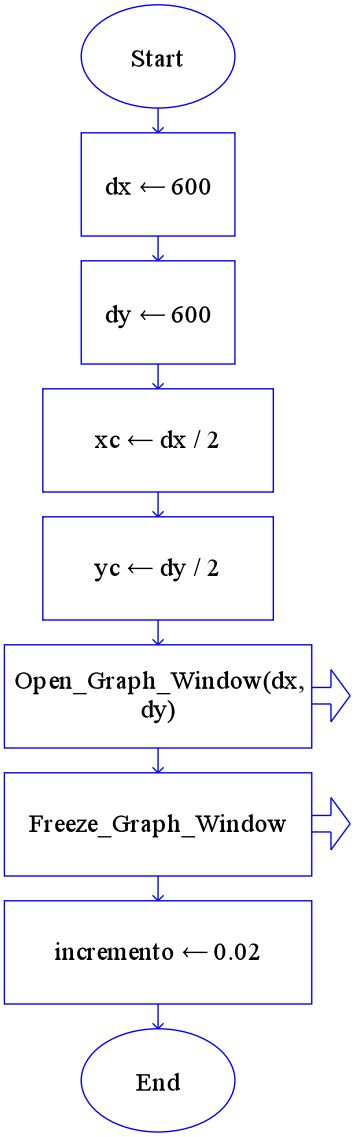
Pergunta 8 – (3 Valores) Implemente em Raptor e RaptorGraph um programa que desenhe a seguinte função:

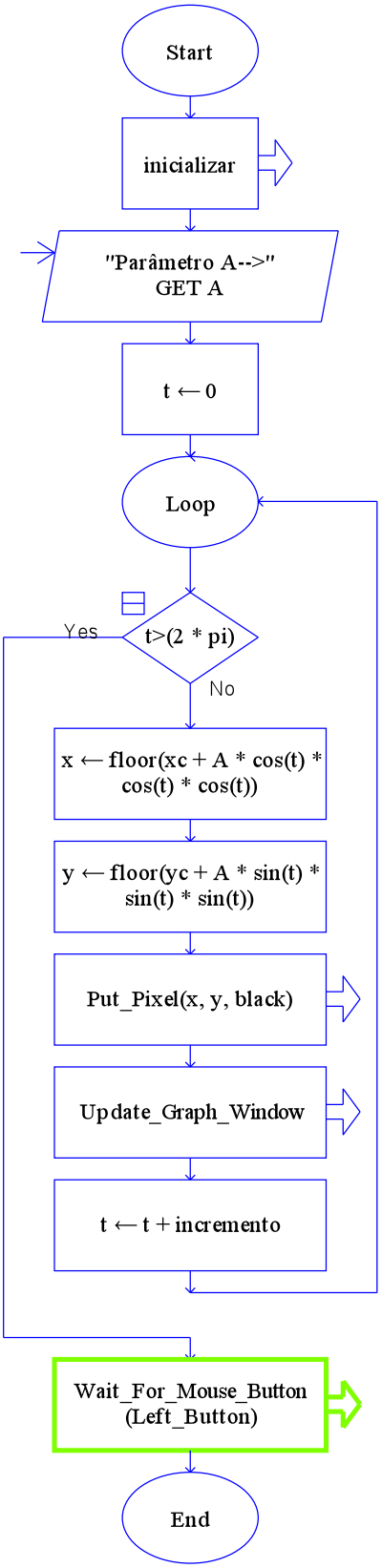
$$\begin{cases} x = A * \cos^3 t \\ y = A * \sin^3 t \end{cases}$$

Representação visual:



Resposta:


Start
dx ← 600
dy ← 600
xc ← dx / 2
yc ← dy / 2
Open_Graph_Window(dx, dy)
Freeze_Graph_Window
incremento ← 0.02
End


Start
inicializar
"Parâmetro A-->"
GET A
t ← 0
Loop
t>(2 * pi)
Yes
No
x ← floor(xc + A * cos(t) * cos(t) * cos(t))
y ← floor(yc + A * sin(t) * sin(t) * sin(t))
Put_Pixel(x, y, black)
Update_Graph_Window
t ← t + incremento
Wait_For_Mouse_Button (Left_Button)
End

TEMA 6: Conceitos Aplicações e Historia da Computação Gráfica

Pergunta 8 – (2 valores) Commpare as caracteristicas do ambiente XNA com o ambiente Processing em termos de aplicações gráficas no dominio dos videojogos 2D. Enumere Vantagens e Inconvenientes nas diversas vertentes que abordar.

Resposta:

O XNA é um ambiente desenvolvido pela Microsoft para os seus sistemas operativos em computadores, consolas de jogos (XBOX 360) e dispositivos móveis. Isto significa que não funciona em outros ambientes como OSX, Linux ou UNIX para enumerar alguns. É um ambiente especialmente preparado para o desenvolvimento de Jogos a 3D, tem incluido no seu workflow as funções para incluir e usar recursos 3D, criar o ambiente de um motor gráfico e usar como linguagem de referência o C# desenvolvido pela Microsoft. Como está preparado para gráficos 3D possui implementado classes de objectos para criar camaras, luzes e importar recursos 3D com facilidade. Isto não impede que possamos por exemplo desenvolver jogos a 2D, já que possui igulamente recursos para tal. Como é um ambiente fechado só pode ser desenvolvido pela própria Microsoft, embora seja gratuito.

O Processing é um ambiente desenvolvido em Java no MIT, destinado a não programadores (Artistas e Criativos), que implementa um IDE de desenvolvimento simplificado de forma a permitir muito fácilmente a aprendizagem da linguagem e do seu ambiente de trabalho. É Opensource e multiplataforma o que significa que pode ser portado para qualquer SO ou processador onde exista Java (Quase Todos). Especialmente preparado para gráficos 2D possui muitas bibliotecas desenvolvidas por terceiros (libraries) e novas são criadas constantemente. É usado por pessoas sem uma preparação muito avançada em engenharia de software especialemente em artes visuais. Tal como o XNA é gratuito.