- UNIVERSIDADE DO VALE DO RIO DOS SINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS – GRADUAÇÃO TECNOLÓGICA – GT JEDi
CURSO DE DESENVOLVIMENTO DE JOGOS E ENTRETENIMENTO DIGITAL

Projeto de Jogos
Ferramentas de Desenvolvimento Rápido de Jogos 3D
RAD3D - DarkBasicPro

Disciplina: **Desenv. Rápido de Jogos 3D** **E-mail:** ***fosorio@ unisinos.br***
Professor responsável: ***Fernando Santos Osório*** **Web:** ***http://inf.unisinos.br/~osorio/jogos-gt.html***
Semestre: 2006/2 **Horário:** ***34/54 [34-6B302, 54-6L121 Lab. de Prog.]***

TRANSFORMAÇÕES 2D e 3D

Transformações Básicas em 2D:

1. **Translação**: Posição + Deslocamento
   Coordenada inicial do objeto 2D: X#, Y#
   Coordenada do objeto 2D Transladado: NovoX#, NovoY#
   Translação = deslocamento em X e Y: DX#, DY#
   Operação:
   NovoX# = X# + DX#
   NovoY# = Y# + DY#

2. **Escala**: Alterar o tamanho de um objeto, significa alterar seus componentes
   Coordenada inicial do Ponto de Referência 2D: X#, Y#
   Coordenada do Ponto 2D após Escalamento: NovoX#, NovoY#
   Fator de escala: SX#, SY# (escala no eixo X e no eixo Y)
   Operação:
   NovoX# = X# * SX#
   NovoY# = Y# * SY#

3. **Rotação**: Posição + Ângulo de Rotação (em relação a um centro e com um determinado raio)
   Coordenada inicial do objeto 2D: X#, Y#
   Coordenada do objeto 2D Rotacionado: NovoX#, NovoY#
   Centro da rotação (pivot): CX#, CY#
   Raio de rotação: Raio#
   Ângulo de Rotação: Ang#
   Operação:
   Rotação sobre a origem (0,0) no círculo unitário (raio 1.0)
   NovoX# = X# * Cos(Ang#) - Y# * Sin(Ang#)
   NovoY# = X# * Sin(Ang#) + Y# * Cos(Ang#)
   Usualmente esta operação é seguida de um ajuste de escala (raio) e de translação (centro)
   PtX# = CX# + (NovoX# * Raio#)
   PtY# = CY# + (NovoY# * Raio#)

Exemplo de Código:

```
REM ***** Código Fonte DarkBasicPro *****
REM UNISINOS - Curso GT-JEDi
REM Autor: Fernando Osório - DATA: Agosto/2005  [Código completo disponível na Internet]

Rem Inicializa o modo gráfico em 800x600 com 32 bits/pixel (opcional)
SET DISPLAY MODE 800,600,32

ImgAbelha=1
load image "abelha.png", ImgAbelha

cls
paste image ImgAbelha,0,0,1
get image ImgAbelha,1,1,59,65

Rem Posicao no círculo de raio 1.0 (usada na rotação)
px_ini#=1.0
py_ini#=0.0
Rem Tamanho do raio usado para alterar a escala da circunferência de raio 1.0
Rem Com uma escala de Raio#=100.0 obtemos uma circunferência de raio 100.0
raio#=100.0
Rem Ângulo inicial da rotação
angulo#=0.0

Rem Translação a ser realizada do centro da circunferência de raio unitário (1.0)
cx#=400
cy#=300
Rem Deslocamento (coordenada obtida por rotação) aplicado junto ao centro da circunferência
dx#=0
dy#=0

cls
DO

  Rem Calculo da rotação em sentido horário. Lembre-se: o Eixo Y cresce para baixo na tela...
  dx#=(px_ini#*cos(angulo#)-py_ini#*sin(angulo#) )*raio#
  dy#=(px_ini#*sin(angulo#)+py_ini#*cos(angulo#) )*raio#

  Rem A rotação se considerarmos um eixo Y crescendo para cima será no sentido anti-horário

  Cls
  sprite ImgAbelha, cx#+dx#, cy#+dy#, ImgAbelha
  Rem  Invertendo o Eixo Y
  Rem  sprite ImgAbelha, cx#+dx#, 600-(cy#+dy#), ImgAbelha

  show sprite ImgAbelha
  angulo#=angulo#+1

  if escapekey() then END

LOOP

END
```

Transformações Básicas em 3D:

1. **Translação**: Posição + Deslocamento
   Objeto a ser transladado: Nobj (número que identifica o objeto)
   Posição inicial do objeto no espaço 3D: X#, Y#, Z#
   Deslocamento aplicado ao objeto: DX#, DY#, DZ#
   Posição do objeto após translação: NovoX#, NovoY#, NovoZ#
   Operação:
   X# = Object Position X(Nobj)
   Y# = Object Position Y(Nobj)
   Z# = Object Position Z(Nobj)
   NovoX# = X# + DX#
   NovoY# = Y# + DY#
   NovoZ# = Z# + DZ#
   Position Object Nobj, NovoX#, NovoY#, NovoZ#

   Funções usadas para operações de Translação do DarkBasicPro:
   Position Objet Nobj, X, Y, Z (posicionamento absoluto, não altera orientação)
   Move Object Nobj, Deslocamento (move na direção corrente, para onde está orientado)
   Move Object Up
   Move Object Down Nobj, Deslocamento (move relativo a posição e orientação atual!)
   Move Object Left
   Move Object Right

2. **Escala**: Alterar o tamanho de um objeto
   Objeto a ser transladado: Nobj (número que identifica o objeto)
   Fator de escala a ser aplicado no objeto 3D: SX#, SY#, SZ#
   Operação:
   Scale Object Nobj, SX#, SY#, SZ#

3. **Rotação**: Posição + Centro + Raio + Ângulo de Rotação
   Objeto a ser rotacionado: Nobj (número que identifica o objeto)
   Posição inicial do objeto no espaço 3D: X#, Y#, Z# (Círculo Unitário: 0.0 <= Coord <= 1.0)
   Posição final do objeto após realizada a rotação: NovoX#, NovoY#, NovoZ#
   Centro da rotação (pivot): CX#, CY#, CZ#
   Raio de rotação: Raio#
   Ângulo de Rotação: AngX#, AngY#, AngZ#
   Operação: (inclui translação e escala)

   ROTAÇÃO NO ***PLANO XY*** - SOBRE O ***EIXO Z***
   NovoX# = CX# + (X# * Cos(AngZ#) - Y# * Sin(AngZ#)) * Raio#
   NovoY# = CY# + (X# * Sin(AngZ#) + Y# * Cos(AngZ#)) * Raio#
   NovoZ# = CZ# + Z# * Raio# (não sofre alteração)

   ROTAÇÃO NO ***PLANO XZ*** - SOBRE O ***EIXO Y***
   NovoX# = CX# + (X# * Cos(AngY#) - Z# * Sin(AngY#)) * Raio#
   NovoY# = CY# + Y# * Raio# (não sofre alteração)
   NovoZ# = CZ# + (X# * Sin(AngY#) + Z# * Cos(AngY#)) * Raio#

   ROTAÇÃO NO ***PLANO YZ*** - SOBRE O ***EIXO X***
   NovoX# = CX# + X# * Raio# (não sofre alteração)
   NovoY# = CY# + (Y# * Cos(AngX#) - Z# * Sin(AngX#)) * Raio#
   NovoZ# = CZ# + (Y# * Sin(AngX#) + Z# * Cos(AngX#)) * Raio#

Caso a coordenada inicial (X#,Y#,Z#) não seja tão importante, ou seja, a trajetória é circular ao redor de um objeto, onde não importa o ponto da trajetória onde este começa... então podemos assumir as coordenadas de X#, Y# e Z# iniciais como sendo 1.0 (lembre-se estamos considerando um círculo de raio 1.0), e as equações ficam simplificadas assim:

ROTAÇÃO NO ***PLANO XY*** - SOBRE O ***EIXO Z***
NovoX# = CX# + (Cos(AngZ#) - Sin(AngZ#)) * Raio#
NovoY# = CY# + (Sin(AngZ#) + Cos(AngZ#)) * Raio#
NovoZ# = CZ# + Raio# (não sofre alteração – não precisaria ser atualizado)

ROTAÇÃO NO ***PLANO XZ*** - SOBRE O ***EIXO Y***
NovoX# = CX# + (Cos(AngY#) - Sin(AngY#)) * Raio#
NovoY# = CY# + Raio# (não sofre alteração)
NovoZ# = CZ# + (Sin(AngY#) + Cos(AngY#)) * Raio#

ROTAÇÃO NO ***PLANO YZ*** - SOBRE O ***EIXO X***
NovoX# = CX# + Raio# (não sofre alteração)
NovoY# = CY# + (Cos(AngX#) - Sin(AngX#)) * Raio#
NovoZ# = CZ# + (Sin(AngX#) + Cos(AngX#))* Raio#

Funções usadas para operações de Translação do DarkBasicPro:

Rotação em relação a um pivot: (ver equações acima)
Object Position + Sin/Cos Rx,Ry,Rz + Position Object

Rotação ao redor do centro do próprio objeto:
XRotate Object Nobj, Ângulo#
YRotate Object Nobj, Ângulo#
ZRotate Object Nobj, Ângulo#
Rotação composta nos 3 eixos:
Rotate Object Nobj, AngX#, AngY#, AngZ#
Para obter os ângulos atuais de rotação:
AngX# = Object Angle X(obj)
AngY# = Object Angle Y(obj)
AngZ# = Object Angle Z(obj)

Alterando a rotação do objeto através da definição de sua orientação:
Position Object Nobj, X#, Y#, Z#
Point Object Nobj, X1#, Y1#, Z1#

Outras rotações... Rotações relativas ao ângulo atual
(O efeito final é o mesmo de usar o XRotate, YRotate, ZRotate)
Turn Object Left Nobj, Valor#
Turn Object Right Nobj, Valor#
Pitch Object Up Nobj, Valor#
Pitch Object Down Nobj, Valor#
Roll Object Left Nobj, Valor#
Roll Object Right Nobj, Valor#

Outras funções importantes:
Wrapvalue => Evita que o ângulo saia fora da faixa de 0 a 360 graus
Uso: valor# = wrapvalue(valor#)

Exemplo de programa usando as funções de rotação: YROTATE

```
Load Object "carro.x",1
Autocam off
Make Matrix 1,1000,1000,50,50
Position Matrix 1,-500,0,-500

DO
   Control camera using arrowkeys 0,5.0,2.0

   IF Inkey$()="Y" THEN  YRotate Object 1, Object Angle Y(1)+0.5

   IF Inkey$()="y"  THEN  YRotate Object 1, Object angle Y(1)-0.5

LOOP
```

Exemplo de programa usando as funções de rotação: TURN

```
Load Object "carro.x",1
Autocam off
Make Matrix 1,1000,1000,50,50
Position Matrix 1,-500,0,-500

DO
   Control camera using arrowkeys 0,5.0,2.0

   IF Inkey$()="Y" THEN  Turn Object Left 1, 0.5

   IF Inkey$()="y"  THEN  Turn Object Right 1, 0.5

LOOP
```

Exemplo de programa fazendo um planeta orbitar ao redor de outro: SIN/COS

```
Terra=1 : Lua=2 : Sol=3
Make object sphere Terra, 100    :   Color object Terra, rgb(0,0,255)
Make object sphere Lua, 20       :   Color object Lua, rgb(200,200,200)
Make object sphere Sol, 100      :   Color object Sol, rgb(244,122,0)

Rem Centro de Rotação (Terra), Raio e Ângulo Inicial
Tx#= 300  : Ty#= 0 :  Tz#= 0
Position Object Terra,Tx#,Ty#,Tz#
Raio#=100  :   AngY#=0

DO
  control camera using arrowkeys 0,5.0,2.0

  Rem Rotacao no PLANO XZ  - Ao redor do EIXO Y
  if Spacekey()
    AngY#=AngY#+1
    NovoX# = TX# + (Cos(AngY#)-Sin(AngY#))*Raio#
    NovoZ# = TZ# + (Sin(AngY#)+Cos(AngY#))*Raio#
    position object Lua,NovoX#,0.0,NovoZ#
  Endif

LOOP
```

Exercício

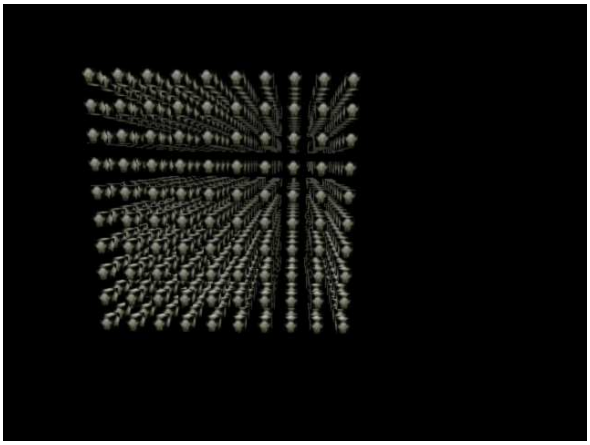
Implemente um SCREENSAVER!

Use Translações, Escalas e/ou Rotações para animar a cena, onde você pode deslocar tanto os objetos da cena como também a câmera (os comandos de posição e rotação também podem ser aplicados na câmera.

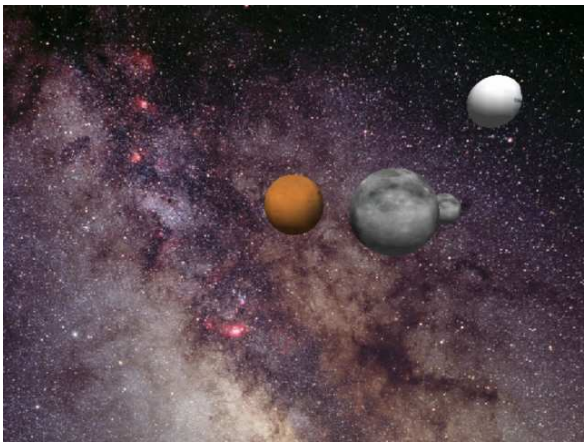
Bom trabalho!

Exemplos...

Cubo de Cubos



Planetas



`

Outros...
Aquário, Relógio 3D, Fotos de Jogos Famosos ou Pessoas, Naves, Temático (inspirado em filmes, jogos, cenas de terror), Efeitos especiais com objetos, formas, luzes, transparências e cores. Use sua imaginação!