(C++) Animierte Sprites 2D

Moderator: Moderation

neuesThema antworten **Seite 1 von 1** [1 Beitrag]

Vorheriges Thema | Nächstes Thema

Autor **Nachricht**

frodenius

offline

Moderator

Registriert: 11.03.2007, 20:25
Beiträge: 556
Wohnort: Frankfurt/Main

Betreff des Beitrags: (C++) Animierte Sprites 2D **Verfasst:** 14.05.2007, 23:05

soo.. basierend auf einem post im englischen forum(hier), hier den code eines scnenodes der einen animierten sprite rendert. der originalcode ist auf windows basiert, hab ihn plattformunabhÃ¤ngig gemacht und ein bisschen Ã¼bersichtlicher..

CAnimSprite.h

Code:

```
#ifndef CANIMSPRITE_H_INCLUDED
#define CANIMSPRITE_H_INCLUDED

#include "irrlicht.h"
using namespace irr;

class CAnimSprite : public scene::ISceneNode
{

     public:
          CAnimSprite(scene::ISceneNode* parent,
scene::ISceneManager* mgr, s32 id, ITimer* tim);
          virtual void Load(char* filename,s32 frmWidth,s32
frmHeight,bool useClrKey=false);
          virtual void Load(char* filename,s32 Ax,s32 Ay,s32 Aw,s32
Ah,s32 frmWidth,s32 frmHeight,bool useClrKey=false);
```

```
         virtual void PlayForward() {forward = true;}
         virtual void PlayBackward() {forward = false;}
         virtual void setSpeed(s32 spd)  {speed = spd;}
         virtual void OnRegisterSceneNode();
         virtual void setFrame(s32 n);
         virtual void OnAnimate();
         virtual void setStartEndFrame( s32 st, s32 ed);
         virtual s32 GetMaxFrames() { return TotalFrm; }
         virtual void render();
         virtual const core::aabbox3d<f32>& getBoundingBox() const
{return Box;}
         virtual u32 getMaterialCount(){return 1;}
         virtual video::SMaterial& getMaterial(u32 i){return
Material;}


     private:
          core::aabbox3d<f32> Box;
          video::S3DVertex    Vertices[4];
          u16                 Indices[12];
          video::SMaterial    Material;
          video::ITexture*    Texture;
          f32                 fWidth,fHeight;
          s32                 crntFrm,TotalFrm;
          s32                 stepww,stephh;
          bool                forward;
          s32                 speed;
          u32                 oldtick;
          s32                 startFrame,endFrame;
          f32                 xCoord,yCoord;
          core::matrix4       Ortho;
          ITimer*             timer;
};


#endif // CANIMSPRITE_H_INCLUDED
```

CAnimSprite.cpp

Code:

```
#include "irrlicht.h"
#include "CAnimSprite.h"
```

```
using namespace irr;
using namespace core;
using namespace scene;
using namespace video;
using namespace io;
using namespace gui;

CAnimSprite::CAnimSprite(ISceneNode* parent, ISceneManager* mgr,
s32 id, ITimer* tim):ISceneNode(parent, mgr,
id),timer(tim),oldtick(0),speed(0)
{
         Material.Wireframe = false;
         Material.Lighting = false;

         u16 ind[] = { 0,1,3, 3,1,2, 1,0,2, 2,0,3 };
         for(u8 i=0;i<12;i++)
         Indices[i] = ind[i];

         IVideoDriver* driver = SceneManager->getVideoDriver();
         dimension2d<s32> Screensize = driver->getScreenSize();

         Ortho(0,0) = (double)2/(double)Screensize.Width;
         Ortho(1,0) = 0;
         Ortho(2,0) = 0;
         Ortho(3,0) = 0;
         Ortho(0,1) = 0;
         Ortho(1,1) = (double)2/(double)Screensize.Height;
         Ortho(2,1) = 0;
         Ortho(3,1) = 0;
         Ortho(0,2) = 0;
         Ortho(1,2) = 0;
         Ortho(2,2) = 1;
         Ortho(3,2) = 0;
         Ortho(0,3) = 0;
         Ortho(1,3) = 0;
         Ortho(2,3) = 0;
         Ortho(3,3) = 1;
}

void CAnimSprite::Load(char* filename,s32 frmWidth,s32
frmHeight,bool useClrKey)
          {
              IVideoDriver* driver =
SceneManager->getVideoDriver();
              dimension2d<s32> Screensize =
driver->getScreenSize();
              float x = (float)frmWidth/2.0f;
              float y = (float)frmHeight/2.0f;
              Vertices[0] = S3DVertex(-x,-y,0,
0,0,0,SColor(255,255,255,255),0,1);
              Vertices[1] = S3DVertex( x,-y,0,
0,0,0,SColor(255,255,255,255),1,1);
              Vertices[2] = S3DVertex( x, y,0,
0,0,0,SColor(255,255,255,255),1,0);
              Vertices[3] = S3DVertex(-x, y,0,
0,0,0,SColor(255,255,255,255),0,0);

              Box.reset(Vertices[0].Pos);
              for (s32 i=1; i<4; ++i)
Box.addInternalPoint(Vertices[i].Pos);

              Texture = driver->getTexture(filename);
              if (useClrKey==true)

driver->makeColorKeyTexture(Texture,position2d<s32>(0,0));
              Material.MaterialType =
EMT_TRANSPARENT_ALPHA_CHANNEL;
              Material.Textures[1] = Texture;

              dimension2d<s32> size = Texture->getOriginalSize();
              fWidth  = (float)frmWidth/(float)size.Width;
              fHeight = (float)frmHeight/(float)size.Height;
              crntFrm = 0;
              stepww = size.Width / frmWidth;
              stephh = size.Height /frmHeight;
              TotalFrm =(s32)(stepww * stephh);
              forward = true;
              startFrame = 0;
              endFrame   = TotalFrm;
              xCoord = yCoord = 0.0;

              Vertices[0].TCoords.X = 0;
              Vertices[0].TCoords.Y = fHeight;
              Vertices[1].TCoords.X = fWidth;
              Vertices[1].TCoords.Y = fHeight;
              Vertices[2].TCoords.X = fWidth;
              Vertices[2].TCoords.Y = 0;
              Vertices[3].TCoords.X = 0;
              Vertices[3].TCoords.Y = 0;
          }

void CAnimSprite::Load(char* filename,s32 Ax,s32 Ay,s32 Aw,s32
Ah,s32 frmWidth,s32 frmHeight,bool useClrKey)
          {
             IVideoDriver* driver =
SceneManager->getVideoDriver();
              dimension2d<s32> Screensize =
driver->getScreenSize();
              float x = (float)frmWidth/2.0f;
              float y = (float)frmHeight/2.0f;
              Vertices[0] = S3DVertex(-x,-y,0,
0,0,0,SColor(255,255,255,255),0,1);
              Vertices[1] = S3DVertex( x,-y,0,
0,0,0,SColor(255,255,255,255),1,1);
              Vertices[2] = S3DVertex( x, y,0,
0,0,0,SColor(255,255,255,255),1,0);
              Vertices[3] = S3DVertex(-x, y,0,
0,0,0,SColor(255,255,255,255),0,0);

              Box.reset(Vertices[0].Pos);
              for (s32 i=1; i<4; ++i)
Box.addInternalPoint(Vertices[i].Pos);

              Texture = driver->getTexture(filename);
              if (useClrKey)

driver->makeColorKeyTexture(Texture,position2d<s32>(0,0));
              Material.MaterialType =
EMT_TRANSPARENT_ALPHA_CHANNEL;
              Material.Textures[1] = Texture;

              dimension2d<s32> size = Texture->getOriginalSize();
              fWidth  = (float)frmWidth/(float)size.Width;
              fHeight = (float)frmHeight/(float)size.Height;
              crntFrm = 0;
              stepww = Aw / frmWidth;
```

```
               stephh = Ah / frmHeight;
               TotalFrm = stepww * stephh;
               forward = true;
               startFrame = 0;
               endFrame   = TotalFrm;
               xCoord = (float)Ax/(float)size.Width;
               yCoord = (float)Ay/(float)size.Height;

               Vertices[0].TCoords.X = xCoord + 0;
               Vertices[0].TCoords.Y = yCoord + fHeight;
               Vertices[1].TCoords.X = xCoord + fWidth;
               Vertices[1].TCoords.Y = yCoord + fHeight;
               Vertices[2].TCoords.X = xCoord + fWidth;
               Vertices[2].TCoords.Y = yCoord + 0;
               Vertices[3].TCoords.X = xCoord + 0;
               Vertices[3].TCoords.Y = yCoord + 0;
          }


         void CAnimSprite::OnRegisterSceneNode()
          {
              if (IsVisible)
SceneManager->registerNodeForRendering(this);
              ISceneNode::OnRegisterSceneNode();
          }

         void CAnimSprite::setFrame(s32 n)
          {
               float x = (n % stepww)*fWidth;
               float y = (n / stepww)*fHeight;
               Vertices[0].TCoords.X = xCoord + x;
               Vertices[0].TCoords.Y = yCoord + y+fHeight;
               Vertices[1].TCoords.X = xCoord + x+fWidth;
               Vertices[1].TCoords.Y = yCoord + y+fHeight;
               Vertices[2].TCoords.X = xCoord + x+fWidth;
               Vertices[2].TCoords.Y = yCoord + y;
               Vertices[3].TCoords.X = xCoord + x;
               Vertices[3].TCoords.Y = yCoord + y;

          }

         void CAnimSprite::OnAnimate()
```

```
          {
             if(timer->getRealTime()-oldtick > speed)
             {
                  oldtick = timer->getRealTime();
                  if (forward)
                  {
                      crntFrm++;
                      if (crntFrm > endFrame-1)crntFrm =
startFrame;
                  }
                  else
                  {
                      crntFrm--;
                      if (crntFrm < startFrame)crntFrm =
endFrame-1;
                  }

                  float x = (crntFrm % stepww)*fWidth;
                  float y = (crntFrm / stepww)*fHeight;
                  Vertices[0].TCoords.X = xCoord + x;
                  Vertices[0].TCoords.Y = yCoord + y+fHeight;
                  Vertices[1].TCoords.X = xCoord + x+fWidth;
                  Vertices[1].TCoords.Y = yCoord + y+fHeight;
                  Vertices[2].TCoords.X = xCoord + x+fWidth;
                  Vertices[2].TCoords.Y = yCoord + y;
                  Vertices[3].TCoords.X = xCoord + x;
                  Vertices[3].TCoords.Y = yCoord + y;
             }
          }

         void CAnimSprite::setStartEndFrame( s32 st, s32 ed)
          {
                startFrame = st;
                endFrame   = ed;
          }

         void CAnimSprite::render()
          {
             IVideoDriver* driver = SceneManager->getVideoDriver();
             driver->setMaterial(Material);

             matrix4 Trns,Scl,Rot,wrld;
             wrld.makeIdentity();
             Trns.makeIdentity();
             Scl.makeIdentity();
             Rot.makeIdentity();

             Trns.setTranslation(RelativeTranslation);
             Scl.setScale(RelativeScale);
             Rot.setRotationRadians(RelativeRotation);

             driver->setTransform(ETS_VIEW, wrld);
             driver->setTransform(ETS_PROJECTION, wrld);

             // update ortho matrix to new screen size {
             core::dimension2d<s32> Screensize =
driver->getScreenSize();
             Ortho(0,0) = (f32)
((double)2/(double)Screensize.Width);
             Ortho(1,1) = (f32)
((double)2/(double)Screensize.Height);
             // }

             wrld = Trns * Ortho * Rot * Scl;

             driver->setTransform(ETS_WORLD, wrld);
             driver->drawIndexedTriangleList(&Vertices[0], 4,
&Indices[0], 4);
          }
```

und hier ein beispiel wie man das ding benutzt:

Code:

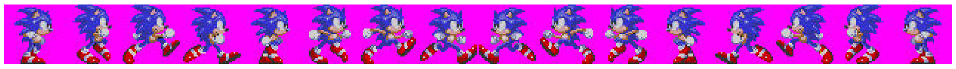
```
#include "irrlicht.h"
#include "CAnimSprite.h"
using namespace irr;
using namespace core;
using namespace scene;
using namespace video;
using namespace io;
using namespace gui;
```

```
int main()
{
     IrrlichtDevice* irrDevice =
createDevice(EDT_OPENGL,dimension2d<s32>
(640,480),32,false,false,false,0);
     IVideoDriver*   irrVideo   = irrDevice->getVideoDriver();
     ISceneManager*  irrSceneMgr = irrDevice->getSceneManager();

     CAnimSprite* Sprite = new
CAnimSprite(irrSceneMgr->getRootSceneNode(), irrSceneMgr, 666,
irrDevice->getTimer());
     Sprite->Load("sonwalk.jpg",0,0,40*8,40,40,40,true);
     Sprite->setSpeed(100);
     Sprite->PlayBackward();
     Sprite->setScale(vector3df(2,2,0));
     Sprite->setPosition(vector3df(-0.5,0.1,0));
     f32 rt=0;

     while  (irrDevice->run())
       {
           irrVideo->beginScene(true, true, SColor(0,200,200,200));
           rt += 0.01;
           Sprite->setRotation(vector3df(0,0,rt));
           irrSceneMgr->drawAll();
           irrVideo->endScene();
       }
     irrDevice->drop();

     return 0;
}
```

und das bild im anhang..
ist leider ziemlich verpixelt wegen .. Ã¶hh ist halt so!

[edit] soo mal auf das neue interface von ISceneNode aktualisiert.. (wurde auch zeit..)

Dateianhänge:



sonwalk.jpg [17.35 KiB | 2348-mal betrachtet]

Nach oben Profil E-Mail

Beiträge der letzten Zeit anzeigen: Alle Beiträge Sortiere nach Erstellungsdatum Aufsteigend Los

neues Thema antworten **Seite 1 von 1** [1 Beitrag]

Foren-Übersicht » **Irrlicht-Coding** » **Code Snippets** Alle Zeiten sind UTC + 1 Stunde

Wer ist online?

Mitglieder in diesem Forum: 0 Mitglieder und 1 Gast

Du darfst **keine** neuen Themen in diesem Forum erstellen.
Du darfst **keine** Antworten zu Themen in diesem Forum erstellen.
Du darfst deine Beiträge in diesem Forum **nicht** ändern.
Du darfst deine Beiträge in diesem Forum **nicht** löschen.
Du darfst **keine** Dateianhänge in diesem Forum erstellen.

Suche nach:
Los Gehe zu: Code Snippets Los