(C++) Render To Texture Skybox

newtopic postreply Irrlicht Engine Forum Index -> Code Snippets

View previous topic :: Vie

Author	Message
bitplane Admin Joined: 28 Mar 2005 Posts: 2981 Location: England	Posted: Thu May 18, 2006 11:24 pm Post subject: (C++) Render To Texture Skybox

Nothing more than the usual skybox with some tcoord changes and a camera added. useful for my project and I'm well chuffed with the results, so i thought
I dont understand the fov of the camera working at 0.93, I just changed it til it looked right.
the idea is, of course, to render lots of distant terrain/mesh/etc into the skybox outside your main loop, and render the skybox each frame instead

Code:

```
// CRTTSkyBox.h
#ifndef __C_RTT_SKY_BOX_SCENE_NODE_H_INCLUDED__
#define __C_RTT_SKY_BOX_SCENE_NODE_H_INCLUDED__

#include "ISceneNode.h"
#include "ICameraSceneNode.h"
#include "S3DVertex.h"

namespace irr
{
namespace scene
{

   // Skybox, rendererd with zbuffer turned off, before all other nodes.
   class CRTTSkyBoxSceneNode : public ISceneNode
   {
   public:

      //! constructor
      CRTTSkyBoxSceneNode(core::dimension2d<s32> size, ISceneNode* parent, ISceneManager* mgr, s32 id);

      //! destructor
      virtual ~CRTTSkyBoxSceneNode();

      // renders the view from the said position to the skybox
      void renderToSkyBox(core::vector3df pos, video::SColor col=video::SColor(0,0,0,255), bool self=false);

      virtual void OnPreRender();

      //! renders the node.
      virtual void render();

      //! returns the axis aligned bounding box of this node
      virtual const core::aabbox3d<f32>& getBoundingBox() const;

      //! returns the material based on the zero based index i. To get the amount
      //! of materials used by this scene node, use getMaterialCount().
      //! This function is needed for inserting the node into the scene hirachy on a
      //! optimal position for minimizing renderstate changes, but can also be used
      //! to directly modify the material of a scene node.
      virtual video::SMaterial& getMaterial(s32 i);

      //! returns amount of materials used by this scene node.
      virtual s32 getMaterialCount();

      scene::ICameraSceneNode *skyCam;

   private:

      core::aabbox3d<f32> Box;
      u16 Indices[6];
      video::S3DVertex Vertices[4*6];
      video::SMaterial Material[6];
      video::ITexture *rt[6];
   };

} // end namespace scene
} // end namespace irr

#endif
```

Code:

```
// Copyright (C) 2002-2005 Nikolaus Gebhardt
// This file is part of the "Irrlicht Engine".
// For conditions of distribution and use, see copyright notice in irrlicht.h

#include "CRTTSkyBoxSceneNode.h"
#include "IVideoDriver.h"
```

```
#include "ISceneManager.h"
#include "ICameraSceneNode.h"
#include "S3DVertex.h"


namespace irr
{
namespace scene
{

//! constructor
CRTTSkyBoxSceneNode::CRTTSkyBoxSceneNode(core::dimension2d<s32> size, ISceneNode* parent, ISceneManager* mgr, s32 id)
: ISceneNode(parent, mgr, id)
{
   #ifdef _DEBUG
   setDebugName("CRTTSkyBoxSceneNode");
   #endif

   AutomaticCullingEnabled = false;

   // create indices

   Indices[0] = 0;
   Indices[1] = 1;
   Indices[2] = 2;
   Indices[3] = 0;
   Indices[4] = 2;
   Indices[5] = 3;

   // create material

   video::SMaterial mat;
   mat.Lighting = false;
   mat.ZBuffer = false;
   mat.ZWriteEnable = false;
   mat.BilinearFilter = true;

   /* Hey, I am no artist, but look at that
      cool ASCII art I made! ;)  (Niko)

       -111         111
          /6--------/5        y
         /  |      / |        ^  z
        /   |   11-1 |        | /
  -11-1 3---------2  |        |/
        |   7- - -| -4 1-11    *---->x
        | -1-11   |  /       3-------|2
        |/        | /         |    //|
        0---------1/          |  //  |
     -1-1-1     1-1-1         |//    |
                            0--------1
   */

   f32 onepixel = 1.0f / (size.Width * 1.5f);

    // create the render targets
    for (s32 n=0; n<6; ++n)
      rt[n] = mgr->getVideoDriver()->createRenderTargetTexture(size);
    // create camera (remember last camera)
    scene::ICameraSceneNode *currentcam = mgr->getActiveCamera();
    skyCam = mgr->addCameraSceneNode();
    skyCam->setFarValue(100000.0f);
    skyCam->setAspectRatio(1.0f);
    skyCam->setFOV(0.93f);       // dunno why, but it works here
    mgr->setActiveCamera(currentcam);

   f32 l = 10.0f;
   f32 t = 1.0f - onepixel;
   f32 o = 0.0f + onepixel;

   Material[0] = mat;
   Material[0].Texture1 = rt[4];
   Vertices[0] = video::S3DVertex(-l,-l,-l, 0,0,1, video::SColor(255,255,255,255), t, o);
   Vertices[1] = video::S3DVertex( l,-l,-l, 0,0,1, video::SColor(255,255,255,255), o, o);
   Vertices[2] = video::S3DVertex( l, l,-l, 0,0,1, video::SColor(255,255,255,255), o, t);
   Vertices[3] = video::S3DVertex(-l, l,-l, 0,0,1, video::SColor(255,255,255,255), t, t);

   // create left side

   Material[1] = mat;
   Material[1].Texture1 = rt[2];
   Vertices[4] = video::S3DVertex( l,-l,-l, -1,0,0, video::SColor(255,255,255,255), t, o);
   Vertices[5] = video::S3DVertex( l,-l, l, -1,0,0, video::SColor(255,255,255,255), o, o);
   Vertices[6] = video::S3DVertex( l, l, l, -1,0,0, video::SColor(255,255,255,255), o, t);
   Vertices[7] = video::S3DVertex( l, l,-l, -1,0,0, video::SColor(255,255,255,255), t, t);

   // create back side

   Material[2] = mat;
   Material[2].Texture1 = rt[5];
   Vertices[8]  = video::S3DVertex( l,-l, l, 0,0,-1, video::SColor(255,255,255,255), t, o);
   Vertices[9]  = video::S3DVertex(-l,-l, l, 0,0,-1, video::SColor(255,255,255,255), o, o);
   Vertices[10] = video::S3DVertex(-l, l, l, 0,0,-1, video::SColor(255,255,255,255), o, t);
   Vertices[11] = video::S3DVertex( l, l, l, 0,0,-1, video::SColor(255,255,255,255), t, t);

   // create right side

   Material[3] = mat;
   Material[3].Texture1 = rt[3];
   Vertices[12] = video::S3DVertex(-l,-l, l, 1,0,0, video::SColor(255,255,255,255), t, o);
   Vertices[13] = video::S3DVertex(-l,-l,-l, 1,0,0, video::SColor(255,255,255,255), o, o);
   Vertices[14] = video::S3DVertex(-l, l,-l, 1,0,0, video::SColor(255,255,255,255), o, t);
   Vertices[15] = video::S3DVertex(-l, l, l, 1,0,0, video::SColor(255,255,255,255), t, t);

   // create top side

   Material[4] = mat;
   Material[4].Texture1 = rt[0];
   Vertices[16] = video::S3DVertex( l, l, l, 0,-1,0, video::SColor(255,255,255,255), t, t);
   Vertices[17] = video::S3DVertex(-l, l, l, 0,-1,0, video::SColor(255,255,255,255), t, o);
   Vertices[18] = video::S3DVertex(-l, l,-l, 0,-1,0, video::SColor(255,255,255,255), o, o);
   Vertices[19] = video::S3DVertex( l, l,-l, 0,-1,0, video::SColor(255,255,255,255), o, t);

   // create bottom side

   Material[5] = mat;
   Material[5].Texture1 = rt[1];
   Vertices[20] = video::S3DVertex(-l,-l, l, 0,1,0, video::SColor(255,255,255,255), o, o);
   Vertices[21] = video::S3DVertex( l,-l, l, 0,1,0, video::SColor(255,255,255,255), o, t);
   Vertices[22] = video::S3DVertex( l,-l,-l, 0,1,0, video::SColor(255,255,255,255), t, t);
   Vertices[23] = video::S3DVertex(-l,-l,-l, 0,1,0, video::SColor(255,255,255,255), t, o);

}



//! destructor
CRTTSkyBoxSceneNode::~CRTTSkyBoxSceneNode()
{
    // drop render targets
    for (s32 n=0; n<6; ++n)
      rt[n]->drop();

    // drop sky camera
    skyCam->remove();
}

void CRTTSkyBoxSceneNode::renderToSkyBox(core::vector3df pos, video::SColor col, bool self)
{

     ICameraSceneNode *lastcam = SceneManager->getActiveCamera();
     SceneManager->setActiveCamera(skyCam);
     bool vis; // is visible
     if (!self)
     {
         vis = isVisible();
         setVisible(false);
     }
     skyCam->setPosition(pos);

     // render up
     SceneManager->getVideoDriver()->setRenderTarget(rt[0], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,10,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render down
     SceneManager->getVideoDriver()->setRenderTarget(rt[1], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,-10,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render left
     SceneManager->getVideoDriver()->setRenderTarget(rt[2], true, true, col);
     skyCam->setTarget(pos + core::vector3df(10,0,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render right
     SceneManager->getVideoDriver()->setRenderTarget(rt[3], true, true, col);
     skyCam->setTarget(pos + core::vector3df(-10,0,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render front
     SceneManager->getVideoDriver()->setRenderTarget(rt[4], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,0,-10));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render back
     SceneManager->getVideoDriver()->setRenderTarget(rt[5], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,0,10));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     if (!self) setVisible(vis);
     SceneManager->getVideoDriver()->setRenderTarget(0);

     SceneManager->setActiveCamera(lastcam);
}


//! renders the node.
void CRTTSkyBoxSceneNode::render()
{
   video::IVideoDriver* driver = SceneManager->getVideoDriver();
   scene::ICameraSceneNode* camera = SceneManager->getActiveCamera();

   if (!camera || !driver)
      return;

   core::matrix4 mat;
   mat.setTranslation(camera->getAbsolutePosition());

   driver->setTransform(video::ETS_WORLD, mat);

   for (s32 i=0; i<6; ++i)
   {
      driver->setMaterial(Material[i]);
      driver->drawIndexedTriangleList(&Vertices[i*4], 4, Indices, 2);
   }
}
```

```
//! returns the axis aligned bounding box of this node
const core::aabbox3d<f32>& CRTTSkyBoxSceneNode::getBoundingBox() const
{
   return Box;
}


void CRTTSkyBoxSceneNode::OnPreRender()
{
   if (IsVisible)
      SceneManager->registerNodeForRendering(this, ESNRP_SKY_BOX);

   ISceneNode::OnPreRender();
}


//! returns the material based on the zero based index i. To get the amount
//! of materials used by this scene node, use getMaterialCount().
//! This function is needed for inserting the node into the scene hirachy on a
//! optimal position for minimizing renderstate changes, but can also be used
//! to directly modify the material of a scene node.
video::SMaterial& CRTTSkyBoxSceneNode::getMaterial(s32 i)
{
   return Material[i];
}


//! returns amount of materials used by this scene node.
s32 CRTTSkyBoxSceneNode::getMaterialCount()
{
   return 6;
}

} // end namespace scene
} // end namespace irr
```

one way you could use it...

Code:

```
// make a 512x512 texture skybox
scene::CRTTSkyBoxSceneNode *sky = new
    CRTTSkyBoxSceneNode(core::dimension2d<s32>(512,512), smgr->getRootSceneNode(),smgr,0);

// make your main scene invisible
gameSceneRoot->setVisible(false);

// render sky, sun, clouds, whatever with default blue bg
backgroundsky->setVisible(true);
sky->renderToSkyBox(core::vector3df(0,0,0));
backgroundsky->setVisible(false);

// render distant landscape with sky
distantTerrainRoot->setVisible(true);
sky->renderToSkyBox(zoneposition, video::SColor(0,0,0,0), true);
distantTerrainRoot->setVisible(false);

// carry on with game
gameSceneRoot->setVisible(true);
```

____
Submit bugs/patches to the tracker!
Need help right now? Visit the chat room

Last edited by bitplane on Mon Jun 26, 2006 1:49 am; edited 1 time in total

Back to top profile pm email www msnm

niko
Site Admin

Joined: 22 Aug 2003
Posts: 1757
Location: Vienna, Austria

Posted: Sun May 21, 2006 1:35 pm Post subject:

Nice. Hey, you copied my ASCII-Art box! Copyright infringement anyone?! 😛

Back to top profile pm www

bitplane
Admin

Joined: 28 Mar 2005
Posts: 2981
Location: England

Posted: Sun May 21, 2006 2:40 pm Post subject:

hehe I didn't have the heart to take it out, gotta love ascii art 😀

____
Submit bugs/patches to the tracker!
Need help right now? Visit the chat room

Back to top profile pm email www msnm

zillion42

Joined: 29 Aug 2007
Posts: 304
Location: Hamburg,

Posted: Fri Mar 13, 2009 7:03 am Post subject:

updated for 1.6...
FOV is now 1.573f ❓

Germany

CRTTSkyBoxSceneNode.cpp:

Code:

```
// Copyright (C) 2002-2005 Nikolaus Gebhardt
// This file is part of the "Irrlicht Engine".
// For conditions of distribution and use, see copyright notice in irrlicht.h

#include "CRTTSkyBoxSceneNode.h"
#include "IVideoDriver.h"
#include "ISceneManager.h"
#include "ICameraSceneNode.h"
#include "S3DVertex.h"


namespace irr
{
namespace scene
{

//! constructor
CRTTSkyBoxSceneNode::CRTTSkyBoxSceneNode(core::dimension2d<u32> size, ISceneNode* parent, ISceneManager* mgr, s32 id)
: ISceneNode(parent, mgr, id)
{

   this->setAutomaticCulling(scene::E_CULLING_TYPE::EAC_OFF);
   // create indices

   Indices[0] = 0;
   Indices[1] = 1;
   Indices[2] = 2;
   Indices[3] = 0;
   Indices[4] = 2;
   Indices[5] = 3;

   // create material

   video::SMaterial mat;
   mat.Lighting = false;
   mat.ZBuffer = false;
   mat.ZWriteEnable = false;
   //mat.BilinearFilter = true;
   mat.setFlag(video::EMF_BILINEAR_FILTER, true);
   /* Hey, I am no artist, but look at that
      cool ASCII art I made! ;)  (Niko)

       -111         111
          /6--------/5        y
         /  |      / |        ^  z
        /   |   11-1 |        | /
  -11-1 3---------2  |        |/
        |   7- - -| -4 1-11    *---->x
        | -1-11   |  /       3-------|2
        |/        | /         |    //|
        0---------1/          |  //  |
     -1-1-1     1-1-1         |//    |
                            0--------1
   */

   f32 onepixel = 1.0f / (size.Width * 1.5f);

    // create the render targets
    for (s32 n=0; n<6; ++n)
      rt[n] = mgr->getVideoDriver()->addRenderTargetTexture(size);
    // create camera (remember last camera)
    scene::ICameraSceneNode *currentcam = mgr->getActiveCamera();
    skyCam = mgr->addCameraSceneNode();
    skyCam->setFarValue(100000.0f);
    skyCam->setAspectRatio(1.0f);
    skyCam->setFOV(1.573f);       // dunno why, but it works here
    mgr->setActiveCamera(currentcam);

   f32 l = 10.0f;
   f32 t = 1.0f - onepixel;
   f32 o = 0.0f + onepixel;

   Material[0] = mat;
   Material[0].setTexture(0,rt[4]);
   Vertices[0] = video::S3DVertex(-1,-1,-1, 0,0,1, video::SColor(255,255,255,255), t, t);
   Vertices[1] = video::S3DVertex( 1,-1,-1, 0,0,1, video::SColor(255,255,255,255), o, t);
   Vertices[2] = video::S3DVertex( 1, 1,-1, 0,0,1, video::SColor(255,255,255,255), o, o);
   Vertices[3] = video::S3DVertex(-1, 1,-1, 0,0,1, video::SColor(255,255,255,255), t, o);

   // create left side

   Material[1] = mat;
   Material[1].setTexture(0,rt[2]);
   Vertices[4] = video::S3DVertex( 1,-1,-1, -1,0,0, video::SColor(255,255,255,255), t, t);
   Vertices[5] = video::S3DVertex( 1,-1, 1, -1,0,0, video::SColor(255,255,255,255), o, t);
   Vertices[6] = video::S3DVertex( 1, 1, 1, -1,0,0, video::SColor(255,255,255,255), o, o);
   Vertices[7] = video::S3DVertex( 1, 1,-1, -1,0,0, video::SColor(255,255,255,255), t, o);

   // create back side

   Material[2] = mat;
   Material[2].setTexture(0,rt[5]);
   Vertices[8]  = video::S3DVertex( 1,-1, 1, 0,0,-1, video::SColor(255,255,255,255), t, t);
   Vertices[9]  = video::S3DVertex(-1,-1, 1, 0,0,-1, video::SColor(255,255,255,255), o, t);
   Vertices[10] = video::S3DVertex(-1, 1, 1, 0,0,-1, video::SColor(255,255,255,255), o, o);
   Vertices[11] = video::S3DVertex( 1, 1, 1, 0,0,-1, video::SColor(255,255,255,255), t, o);

   // create right side

   Material[3] = mat;
   Material[3].setTexture(0,rt[3]);
   Vertices[12] = video::S3DVertex(-1,-1, 1, 1,0,0, video::SColor(255,255,255,255), t, t);
   Vertices[13] = video::S3DVertex(-1,-1,-1, 1,0,0, video::SColor(255,255,255,255), o, t);
   Vertices[14] = video::S3DVertex(-l, l,-l, 1,0,0, video::SColor(255,255,255,255), o, o);
   Vertices[15] = video::S3DVertex(-l, l, l, 1,0,0, video::SColor(255,255,255,255), t, o);

   // create top side

   Material[4] = mat;
   Material[4].setTexture(0,rt[0]);
   Vertices[16] = video::S3DVertex( l, l, l, 0,-1,0, video::SColor(255,255,255,255), t, o);
   Vertices[17] = video::S3DVertex(-l, l, l, 0,-1,0, video::SColor(255,255,255,255), t, t);
   Vertices[18] = video::S3DVertex(-l, l,-l, 0,-1,0, video::SColor(255,255,255,255), o, t);
   Vertices[19] = video::S3DVertex( l, l,-l, 0,-1,0, video::SColor(255,255,255,255), o, o);

   // create bottom side

   Material[5] = mat;
   Material[5].setTexture(0,rt[1]);
   Vertices[20] = video::S3DVertex(-l,-l, l, 0,1,0, video::SColor(255,255,255,255), o, t);
   Vertices[21] = video::S3DVertex( l,-l, l, 0,1,0, video::SColor(255,255,255,255), o, o);
   Vertices[22] = video::S3DVertex( l,-l,-l, 0,1,0, video::SColor(255,255,255,255), t, o);
   Vertices[23] = video::S3DVertex(-l,-l,-l, 0,1,0, video::SColor(255,255,255,255), t, t);

}



//! destructor
CRTTSkyBoxSceneNode::~CRTTSkyBoxSceneNode()
{
    // drop render targets
    for (s32 n=0; n<6; ++n)
      rt[n]->drop();

    // drop sky camera
    skyCam->remove();
}

void CRTTSkyBoxSceneNode::renderToSkyBox(core::vector3df pos, video::SColor col, bool self)
{

     ICameraSceneNode *lastcam = SceneManager->getActiveCamera();
     SceneManager->setActiveCamera(skyCam);
     bool vis; // is visible
     if (!self)
     {
         vis = isVisible();
         setVisible(false);
     }
     skyCam->setPosition(pos);

     // render up
     SceneManager->getVideoDriver()->setRenderTarget(rt[0], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,10,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render down
     SceneManager->getVideoDriver()->setRenderTarget(rt[1], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,-10,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render left
     SceneManager->getVideoDriver()->setRenderTarget(rt[2], true, true, col);
     skyCam->setTarget(pos + core::vector3df(10,0,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render right
     SceneManager->getVideoDriver()->setRenderTarget(rt[3], true, true, col);
     skyCam->setTarget(pos + core::vector3df(-10,0,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render front
     SceneManager->getVideoDriver()->setRenderTarget(rt[4], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,0,-10));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     // render back
     SceneManager->getVideoDriver()->setRenderTarget(rt[5], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,0,10));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();

     if (!self) setVisible(vis);
     SceneManager->getVideoDriver()->setRenderTarget(0);

     SceneManager->setActiveCamera(lastcam);
}


//! renders the node.
void CRTTSkyBoxSceneNode::render()
{
   video::IVideoDriver* driver = SceneManager->getVideoDriver();
   scene::ICameraSceneNode* camera = SceneManager->getActiveCamera();

   if (!camera || !driver)
      return;

   core::matrix4 mat;
   mat.setTranslation(camera->getAbsolutePosition());

   driver->setTransform(video::ETS_WORLD, mat);

   for (s32 i=0; i<6; ++i)
   {
      driver->setMaterial(Material[i]);
      driver->drawIndexedTriangleList(&Vertices[i*4], 4, Indices, 2);
   }
}



//! returns the axis aligned bounding box of this node
const core::aabbox3d<f32>& CRTTSkyBoxSceneNode::getBoundingBox() const
{
   return Box;
}


void CRTTSkyBoxSceneNode::OnRegisterSceneNode()
{
   if (IsVisible)
      SceneManager->registerNodeForRendering(this, ESNRP_SKY_BOX);

   ISceneNode::OnRegisterSceneNode();
}


//! returns the material based on the zero based index i. To get the amount
//! of materials used by this scene node, use getMaterialCount().
//! This function is needed for inserting the node into the scene hirachy on a
//! optimal position for minimizing renderstate changes, but can also be used
//! to directly modify the material of a scene node.
video::SMaterial& CRTTSkyBoxSceneNode::getMaterial(s32 i)
{
   return Material[i];
}


//! returns amount of materials used by this scene node.
s32 CRTTSkyBoxSceneNode::getMaterialCount()
{
   return 6;
}

} // end namespace scene
} // end namespace irr
```

CRTTSkyBoxSceneNode.h

Code:

```
// CRTTSkyBox.h
#ifndef __C_RTT_SKY_BOX_SCENE_NODE_H_INCLUDED__
#define __C_RTT_SKY_BOX_SCENE_NODE_H_INCLUDED__

#include "ISceneNode.h"
#include "ICameraSceneNode.h"
#include "S3DVertex.h"

namespace irr
{
namespace scene
{

   // Skybox, rendererd with zbuffer turned off, before all other nodes.
   class CRTTSkyBoxSceneNode : public ISceneNode
   {
   public:

      //! constructor
      CRTTSkyBoxSceneNode(core::dimension2d<u32> size, ISceneNode* parent, ISceneManager* mgr, s32 id);

      //! destructor
      virtual ~CRTTSkyBoxSceneNode();

      // renders the view from the said position to the skybox
      void renderToSkyBox(core::vector3df pos, video::SColor col=video::SColor(0,0,0,255), bool self=false);

      virtual void OnRegisterSceneNode();

      //! renders the node.
      virtual void render();

      //! returns the axis aligned bounding box of this node
      virtual const core::aabbox3d<f32>& getBoundingBox() const;

      //! returns the material based on the zero based index i. To get the amount
      //! of materials used by this scene node, use getMaterialCount().
      //! This function is needed for inserting the node into the scene hirachy on a
      //! optimal position for minimizing renderstate changes, but can also be used
      //! to directly modify the material of a scene node.
      virtual video::SMaterial& getMaterial(s32 i);

      //! returns amount of materials used by this scene node.
      virtual s32 getMaterialCount();

      scene::ICameraSceneNode *skyCam;

   private:

      core::aabbox3d<f32> Box;
      u16 Indices[6];
      video::S3DVertex Vertices[4*6];
      video::SMaterial Material[6];
      video::ITexture *rt[6];
   };

} // end namespace scene
} // end namespace irr

#endif
```

and a way to render my old skybox and stars into them, turning them off afterwards...
http://irrlicht.sourceforge.net/phpBB2/viewtopic.php?p=189337#189337

Code:

```
// make a 1024x1024 texture skybox
scene::CRTTSkyBoxSceneNode* sky = new scene::CRTTSkyBoxSceneNode(core::dimension2d<u32>(1024,1024), smgr->getRootSceneNode(),smgr,(

// render sky, sun, clouds, whatever with default blue bg
sky->renderToSkyBox(core::vector3df(0,0,0));

allstars->setVisible(false);
SkyBox->setVisible(false);
```

Back to top profile pm

christianclavet Posted: Fri Mar 13, 2009 11:41 pm Post subject:

Joined: 30 Apr 2007
Posts: 1014
Location: Montreal, CANADA

So you've made the 3D Skybox? Wow! Thanks! I'll try to keep a look on that thread.

SourceSDK as implemented this on their engine and I'm using it a lot on some levels using Hammer. I'm very happy to see I could do the same in IRRlicht!

Do you think you'll have the time to do a complete example?
How do you set the RTT cam (position of the fake environment)?

In Hammer, we create 2 "box", one for the main level with the player and another smaller that contain a reduced (1/16) scale of a standard object (We could "maquette"). A camera entity named (SKY_Camera) is placed where the player would be in the real level, and the camera will match the player orientation anc 1/16 scale and render the output on the skybox convering the real level.

Is it working like that? Can you give us more details on the way your method work and the way we use it with the code?

One interesting aspect of the way they do the things with the "SKYBOX" is a shader material that we put on meshes directly. It's rendered as a skybox but o models part that contain that material.

____
Current project:

http://first-king.sourceforge.net

Back to top profile pm www msnm

zillion42 Posted: Sat Mar 14, 2009 8:42 am Post subject:

Joined: 29 Aug 2007
Posts: 304
Location: Hamburg, Germany

well christian, with all respect, bitplane did it... all I did was update it for 1.6, fix compile issues, fine tune the new FOV, and make sure text on my original sky (TL,TR,BL,BR) came out readable... why the new FOV value, not dont ask me, but it looked quite good (1.574 is too much 1.573 looks fine) using a check grid does as described render to skybox or skyboxrtt.... well done bitplane...

EDIT: I'm quite sure I'll find time to do a whole example using a sky and a terrain, but basically its self explanatory... it renders whatever to a skybox...

Back to top profile pm

bitplane Posted: Sat Mar 14, 2009 4:53 pm Post subject:
Admin

Joined: 28 Mar 2005
Posts: 2981
Location: England

Nice to see old code like this being re-used. I should probably look at that FOV thing and tidy it up for the irrext project.

Christian, no example code but here's an example of it in use-
http://dump.bitplane.net/monotheism/rendertoskybox.rar

Its very slow because its rendering to all 6 sides of the skybox every frame and the Irrlicht version is from before hardware buffers, and mipmapping is disabl above the map and move around, it will generate a skybox for that map location on the fly.

____
Submit bugs/patches to the tracker!
Need help right now? Visit the chat room

Back to top profile pm email www msnm

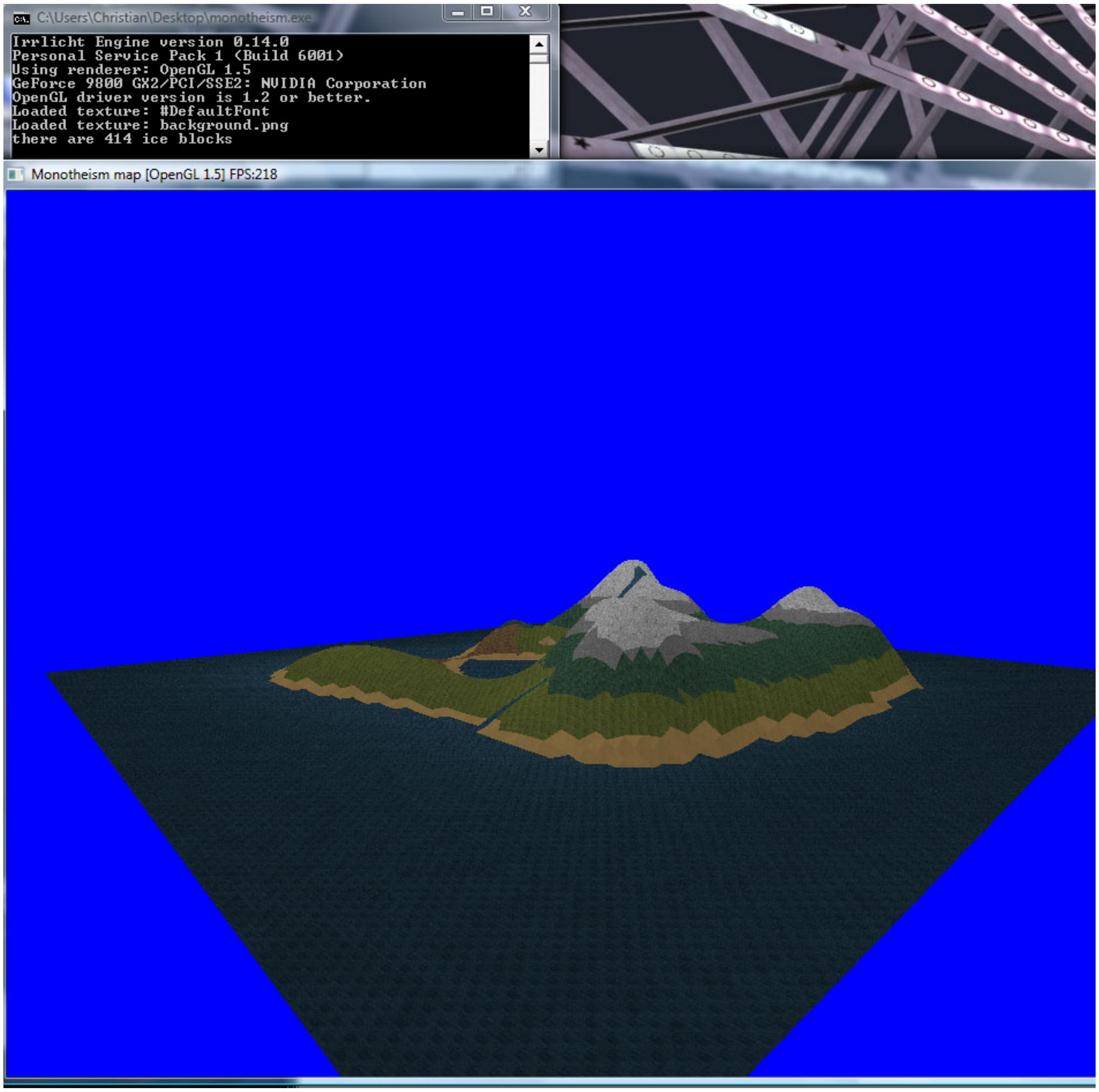
christianclavet Posted: Sat Mar 14, 2009 7:06 pm Post subject:

Thanks!

I've executed the little APP. Seem to work the same way I used to see on SourceSDK.

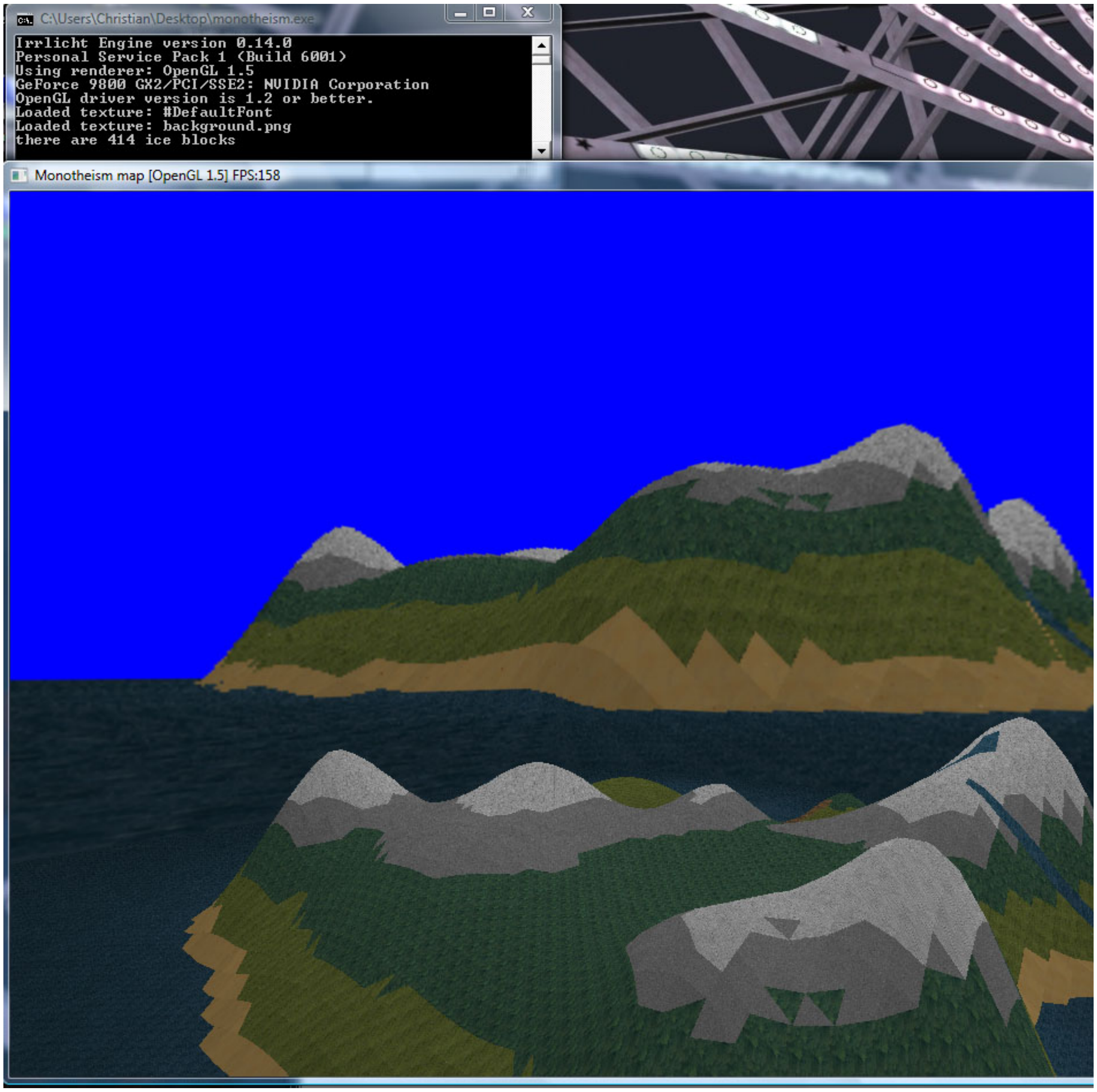
I saw some more things. When I'm outside the "map" area, there seem to be nothing rendered (blue only):

Joined: 30 Apr 2007
Posts: 1014
Location: Montreal, CANADA


C:\Users\Christian\Desktop\monotheism.exe
Monotheism map [OpenGL 1.5] FPS:218

```
Irrlicht Engine version 0.14.0
Personal Service Pack 1 (Build 6001)
Using renderer: OpenGL 1.5
GeForce 9800 GX2/PCI/SSE2: NVIDIA Corporation
OpenGL driver version is 1.2 or better.
Loaded texture: #DefaultFont
Loaded texture: background.png
there are 414 ice blocks
```

But it appear Ok, (Render a camera view to to skybox):


C:\Users\Christian\Desktop\monotheism.exe
Monotheism map [OpenGL 1.5] FPS:158

```
Irrlicht Engine version 0.14.0
Personal Service Pack 1 (Build 6001)
Using renderer: OpenGL 1.5
GeForce 9800 GX2/PCI/SSE2: NVIDIA Corporation
OpenGL driver version is 1.2 or better.
Loaded texture: #DefaultFont
Loaded texture: background.png
there are 414 ice blocks
```

Are the 6 faces of the skybox rendered? Why the resolution of the skybox so low? (It take a long time to render?)

From what I see from the demo you showed me, I could put that RTT cam inside another environnment (far from the viewer camera, copies the data from th camera (mostly angles, and perhaps a reduced value from the position of it (like an offset)) and I think I could do the same technique I'm used to do in Sourc

This will need to be tested, since I'm worried about 2 things:
- Frame rate could be low on complex scenes (like 50K Polys scenes)
- Resolution of the skybox texture show the trick. On source SDK, the skybox resolution is at the resolution of the current display and is sharp.

But that code is great, it show at least a possibility of doing it! We could use that to trick the eye that a maps is huge when in fact is it a lot smaller. (Like sim cityscape in real time in the background)

EDIT:
I don't know how IRRlicht use the rendertarget so you can put that as a very newbie question:

Looked the code and saw that repeating:

Code:

```
// render up
     SceneManager->getVideoDriver()->setRenderTarget(rt[0], true, true, col);
     skyCam->setTarget(pos + core::vector3df(0,10,0));
     skyCam->updateAbsolutePosition();
     SceneManager->drawAll();
```

Can the 6 sides of the skybox be rendered in one pass? There is only one *setRenderTarget* per render pass?

EDIT2: Here is a YouTube video showing the potential of that technique. And how we could apply it.

http://www.youtube.com/watch?v=ObtDnPlX30I&hl=fr

Here is a funny one, showing a NPC put near the SKY Camera.
You see him like a giant, and he's rendered into the skybox:

http://www.youtube.com/watch?v=b_a5UaZMc8g&hl=fr

____
Current project:

http://first-king.sourceforge.net

Back to top profile pm www msnm

bitplane
Admin

Joined: 28 Mar 2005
Posts: 2981
Location: England

Posted: Sun Mar 15, 2009 9:46 pm Post subject:

> **christianclavet wrote:**
> I saw some more things. When I'm outside the "map" area, there seem to be nothing rendered (blue only):

Yeah they get culled for some reason, not sure why, it's been a long time since I used it.

> **christianclavet wrote:**
> Are the 6 faces of the skybox rendered? Why the resolution of the skybox so low? (It take a long time to render?)

They're low resolution because that's the effect I wanted for my game. The map is created using an algorithm, the idea was to render it using a cartoon shad
you travel to that location in-game, you have a cartoony low-res background showing the "real" map location.

> **christianclavet wrote:**
> This will need to be tested, since I'm worried about 2 things:
> - Frame rate could be low on complex scenes (like 50K Polys scenes)

The idea is to not render it every frame. You render to the skybox when you load the level or do a huge change in location. I was going to use multiple passes
for example
Pass 1: Render the sun/moon/sky/stars into the skybox, (taking a few ms)
Pass 2: Render several hundred CCloudSceneNodes into a second rtt skybox (taking maybe half a second or longer)
Pass 3: Render distant terrain, overwriting the first skybox (taking a few ms)
Pass 4: Render distant trees and objects, thousands of them, (taking another half a second or longer)

Then use this as the background of the current location. The result is it costs a few seconds to set up the level, but after that I have maybe a hundred millior

> **christianclavet wrote:**
> - Resolution of the skybox texture show the trick. On source SDK, the skybox resolution is at the resolution of the current display and is sharp.

Each game has its own graphical style, right? Well mine looks like programmer art and pixels. You can use whatever RTT resolution you want, just try it.

> **christianclavet wrote:**
> Can the 6 sides of the skybox be rendered in one pass? There is only one *setRenderTarget* per render pass?

No. You need to move the camera around and render to each side.

This code has kind-of been superseded by an idea for a generic environment map renderer, which isn't complete but you can snag the current code from her
Video example here

____
Submit bugs/patches to the tracker!
Need help right now? Visit the chat room

Back to top profile pm email www msnm

Display posts from previous: All Posts Oldest First Go

newtopic postreply **Irrlicht Engine Forum Index -> Code Snippets**

Page 1 of 1

Watch this topic for replies

Jump to: Code Snippets

You **can** post new to
You **can** reply to to
You **can** edit your p
You **can** delete your p
You **can** vote in

SOURCEFORGE.NET

Powered by phpBB © 2001, 2005 phpBB Group