

开发环境:

Win7 x64, maya2014x64, MSVC2010

命令算法说明:

用 RTT(render to texture)的方法把图片渲染到纹理, 同时用 camera 的旋转值控制图片的旋转.
具体函数是:quadricShapeUI::test2_rtt()

```
void quadricShapeUI::test2_rtt(const quadricGeom *geom)const
{
    glPushAttrib(GL_ALL_ATTRIB_BITS);

    //仅初始化一次 glew
    static bool rtt_initd = false;
    if( !rtt_initd )// initialize RTT buffers if it is not initialized
    {
        if (GLEW_OK != glewInit())
        {
            __debug("Couldn't initialize GLEW");
        }
        initRTTFrameBuffer();
        rtt_initd = true;
    }

    //渲染图片到纹理
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    renderTeapotScene(geom->camRotateX, geom->camRotateY); // Render our teapot scene
into our frame buffer
    glPopAttrib();

    //把纹理贴到面片上
    glBindTexture(GL_TEXTURE_2D, fbo_texture); // Bind our frame buffer texture

    glNormal3f( 0.0f, 0.0f, 1.0f);
    glBegin(GL_QUADS);

    glTexCoord2f(0.0f, 0.0f);
    glVertex3f(-10.0f, -10.0f, -2.0f); // The bottom left corner

    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(10.0f, -10.0f, -2.0f); // The bottom right corner

    glTexCoord2f(1.0f, 1.0f);
    glVertex3f(10.0f, 10.0f, -2.0f); // The top right corner

    glTexCoord2f(0.0f, 1.0f);
    glVertex3f(-10.0f, 10.0f, -2.0f); // The top left corner
    glEnd();

    glBindTexture(GL_TEXTURE_2D, 0); // Unbind any textures
```

```
glPopAttrib();
```

```
}
```

也尝试过修改 uv 去做(见函数 `quadricShapeUI::test1_manipulateUV()`),但结果不好.

使用说明:

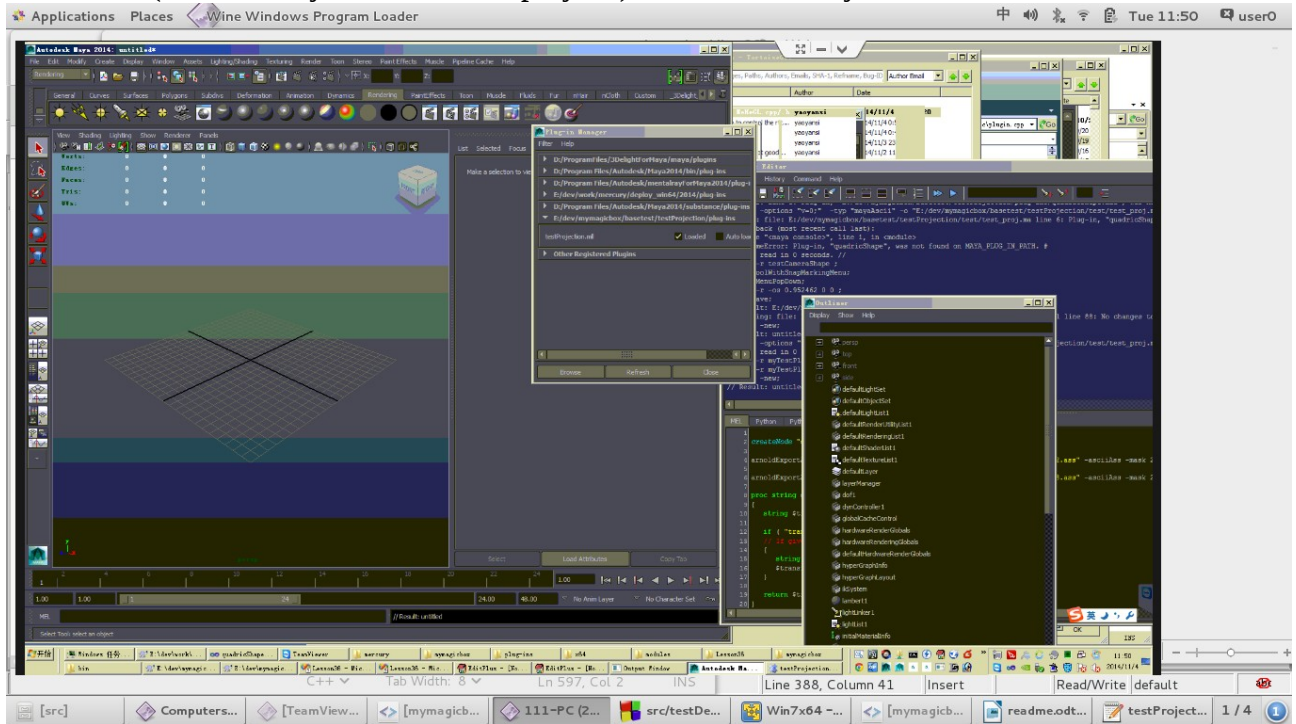
0. 复制 `testProjection.mod` 到'我的文档/maya/2015-x64/modules'目录下,并修改字符串

[E:/dev/mymagicbox/basetest/testProjection](#) 为插件所在的根目录

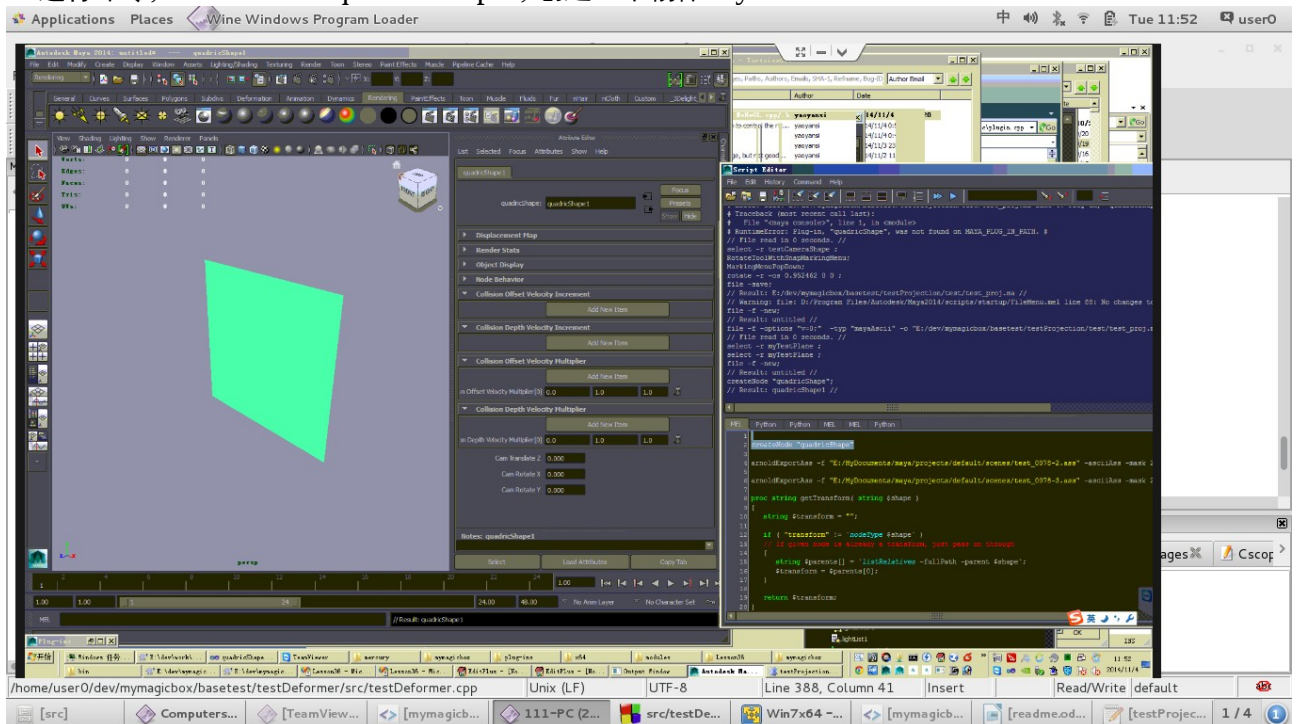
以及

[E:/dev/mymagicbox/basetest/testProjection/bin](#) 为根目录下的 `bin` 目录

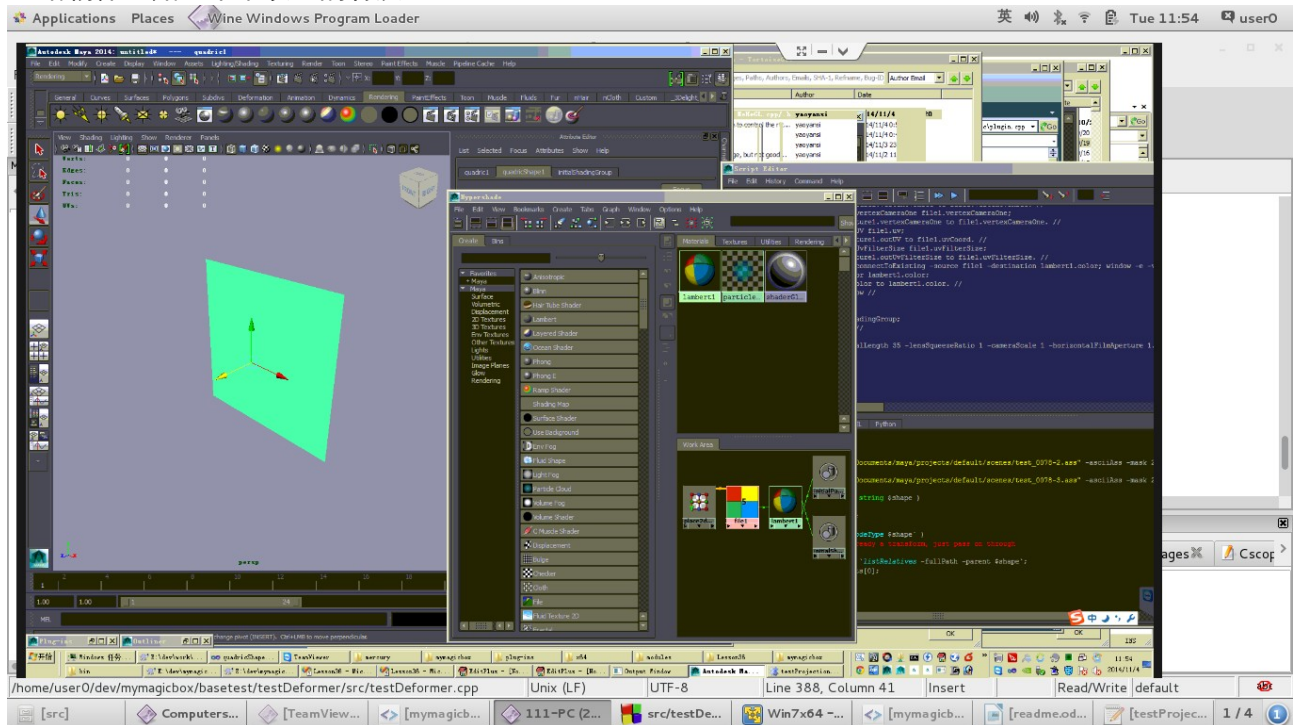
1. 打开场景(例如 `testProjection/test/test_proj.ma`),加载插件 `testProjection.mll`



2. 运行命令, `createNode "quadricShape",` 创建一个物体 `myTestPlane`



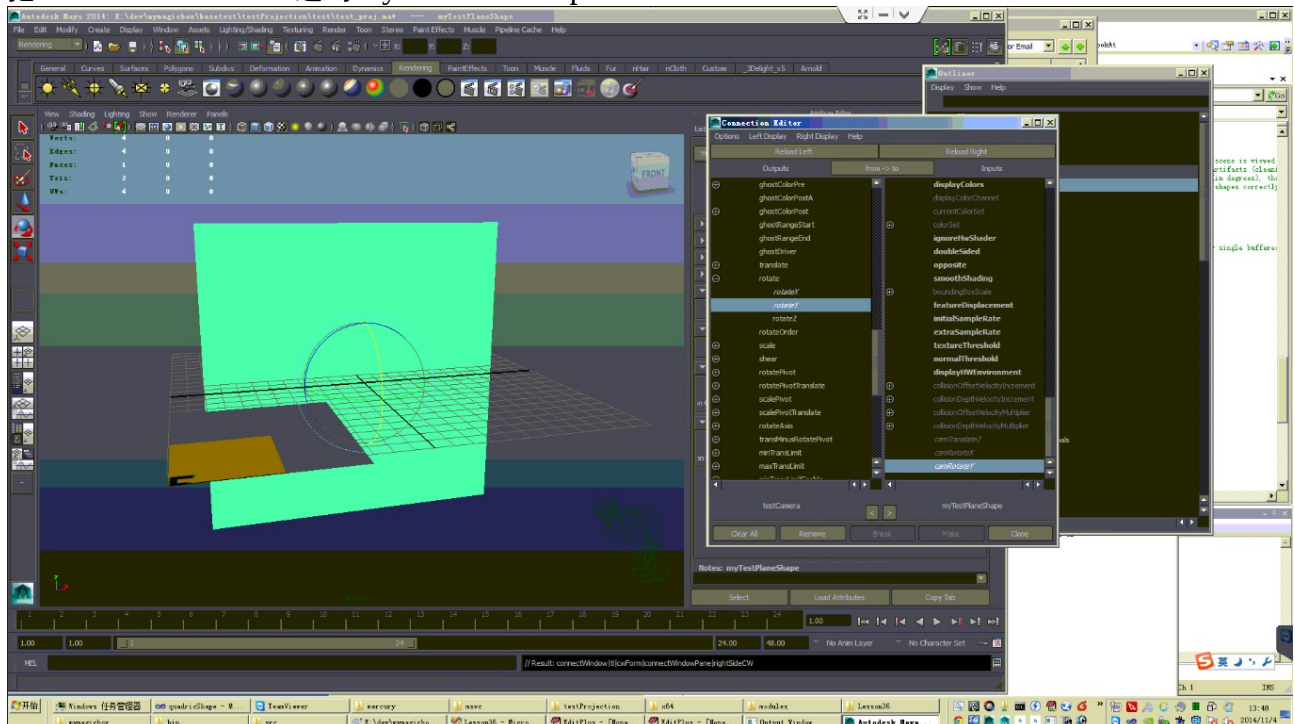
3. 给物体上贴一个带纹理的材质

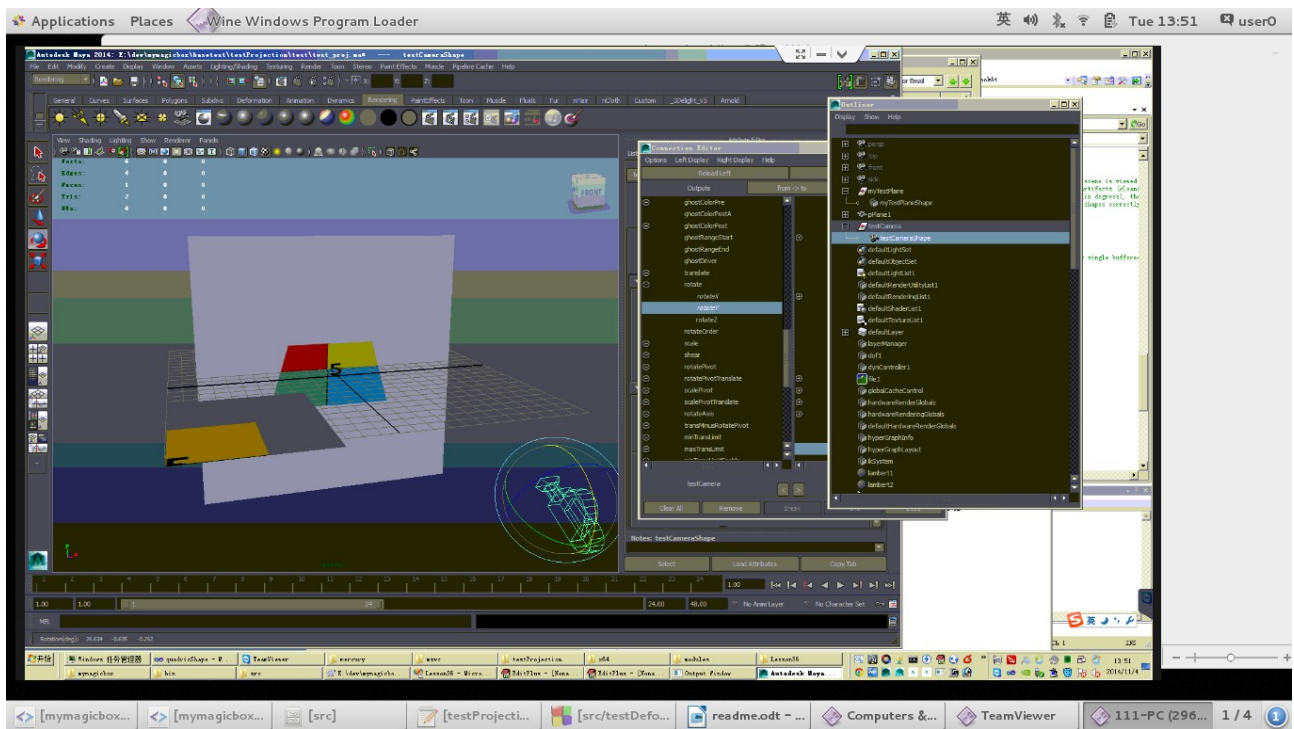


4. 创建一个相机 testCamera.

把 testCamera.rotateX 连到 myTestPlaneShape.camRotateX;

把 testCamera.rotateY 连到 myTestPlaneShape.camRotateY





5. 选择 testCamera 并旋转, 就可以看到效果

