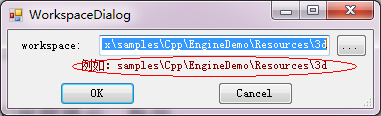
# 模型编辑软件使用说明

**版本：V 0.0.2**

## 使用前须知：

首先，安装.net framework 4.0或以上版本。

第一次启动工具时，会提示输入工作目录



通常你需要把工作目录指向存放资源的文件夹目录，在这里我们按照提示，指向samples\Cpp\EngineDemo\Resources\3d，这么指定是因为目前工具需要载入一些公用美术资源（灯光模型型，骨骼模型等），直接指定到EngineDemo中会避免自己配制的麻烦。

## 概览：



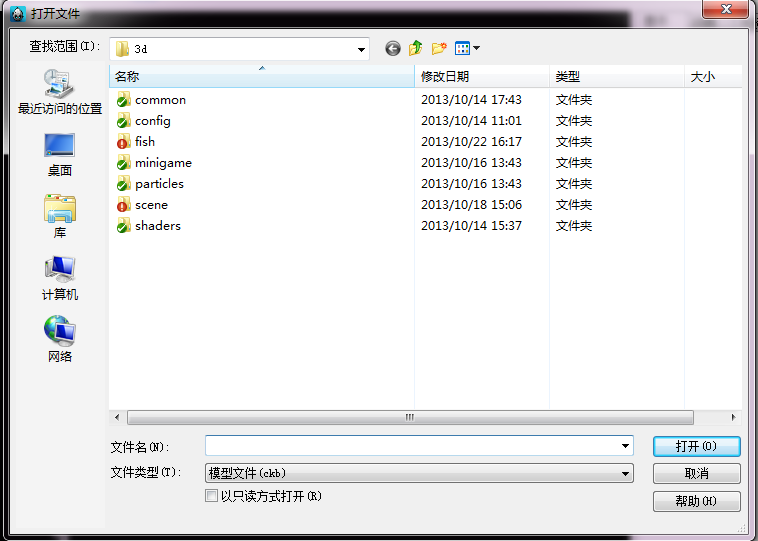
1. 标题栏。
2. 菜单栏。
3. 快捷栏。
4. 显示区。
5. 编辑区。
6. 状态栏。

## 菜单栏介绍：

### 文件：

* 载入模型：

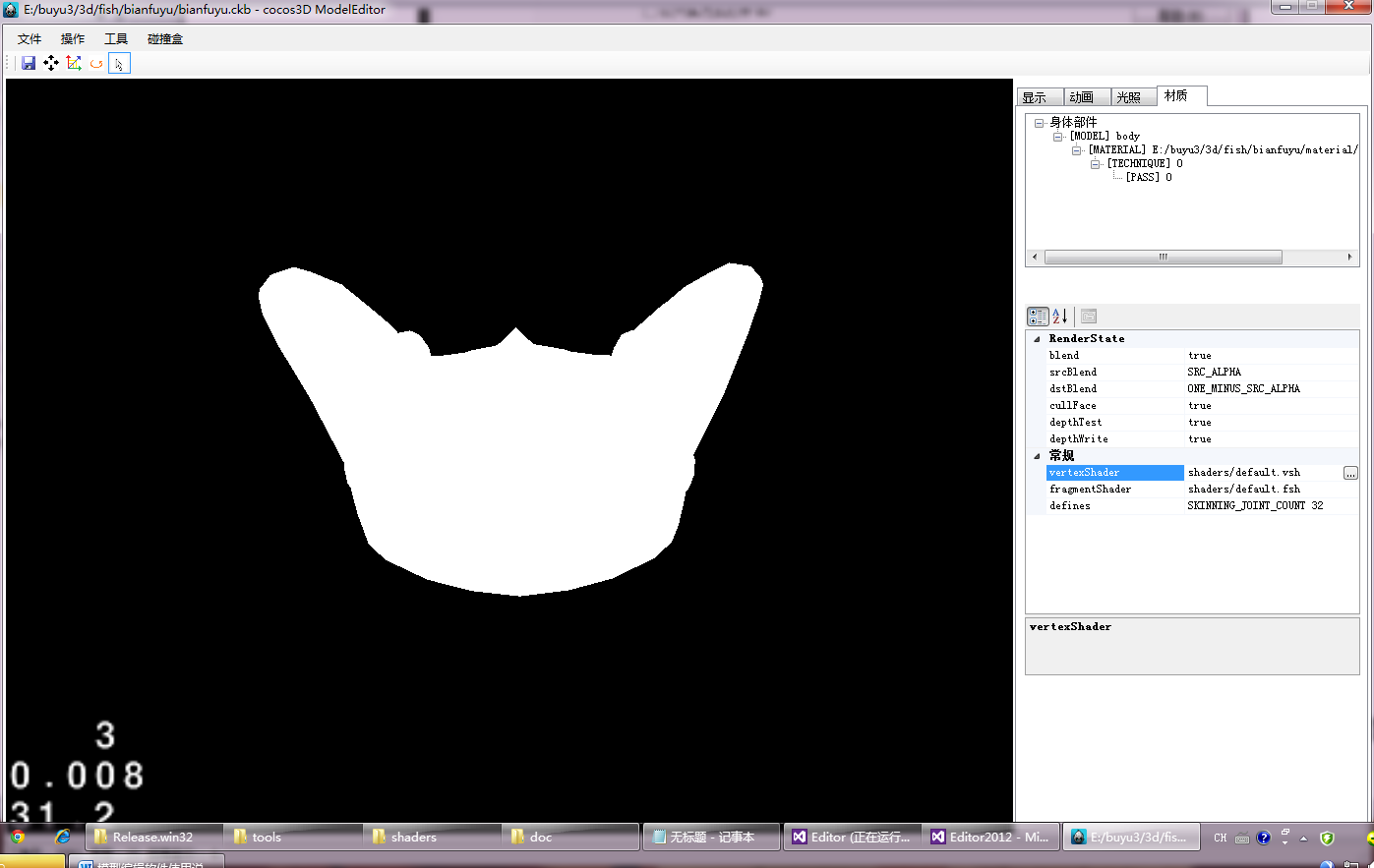
点击后弹出载入窗口，如下图：



选择载入以\*.ckb或\*.fbx为后缀的模型文件，默认显示\*.ckb文件，可通过“文件类型”下拉列表选择文件类型，对于载入的是静态模型或动态模型，编辑器会自动对其进行区分。选定要打开的模型文件后，点击打开按钮，这样模型便会加载到模型编辑软件中，并根据提示确定是否赋予它默认材质。

**注：**载入\*.fbx文件时，编辑器会自动生成对应的\*.ckb文件，并进行载入，也就是说，当你载入\*.fbx文件时，实际上是先生成\*.ckb文件，然后载入这个\*.ckb文件。

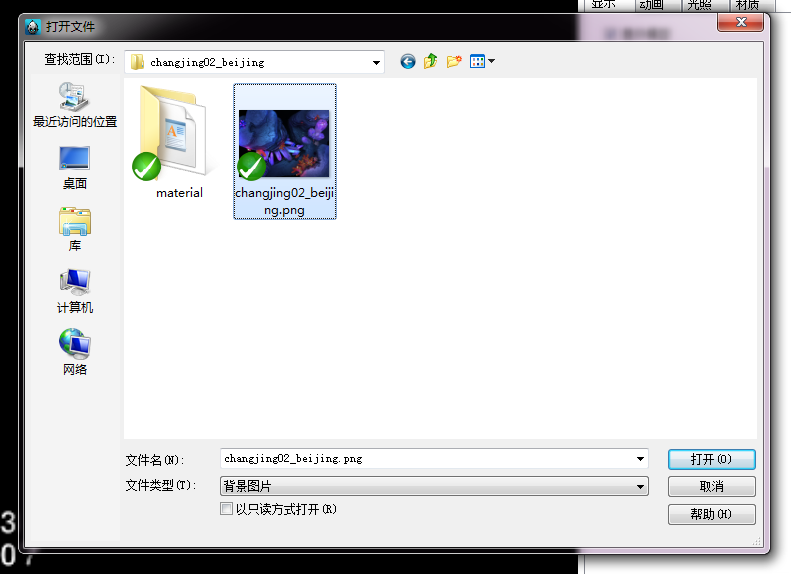
默认材质的shader是default.vsh 和default.fsh，这两个shader是不处理贴图的，所以载入默认材质后，编辑器中的模型是白色的，如下如：



如果想赋予模型贴图，首先要在材质面板切换可以处理贴图的shader。

#### 载入背景：

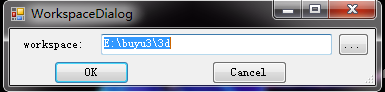
点击后弹出载入窗口，如下图：



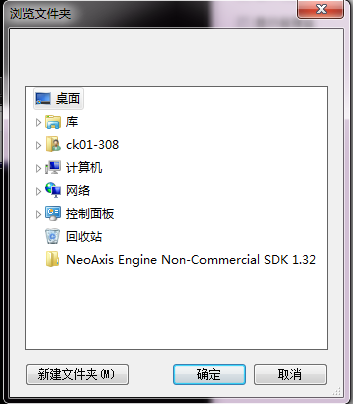
选择将要作为背景的图片，选定后，点击打开按钮，所选的图片就会作为模型编辑软件的背景呈现出来，背景图是按照贴图的原始尺寸进行载入的，不会对图片的大小进行任何改动，所以当你需要大的背景时，必须载入比较大的背景图。

#### 工作目录：

点击后，弹出设置工作目录窗口，如下图：



点击右边的浏览按钮，会弹出目录选择对话框，如下图：



通过该对话框，确定模型编辑软件的工作目录，设置好模型编辑软件的工作目录后，模型编辑软件所处理的资源都是相对于当前设置的工作目录，包括模型的载入和材质的生成等。

模型编辑软件的工作目录会保存在运行程序所在的目录，文件名是ModelEditorConfig.xml，如果用户没有设置工作目录，编辑器会以当前应用程序所在的目录为工作目录。

#### 保存：

保存用户所作的修改，例如：当用户对模型的材质信息进行修改后，点击保存，将保存材质所作的修改。

### 操作：

#### 重置：

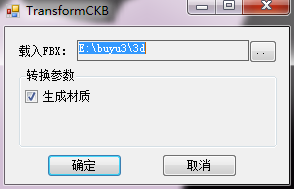
将模型和视口恢复到原始状态，由于用户在使用过一段时间后，会对模型进行各种各样的改动，点击重置按钮，会将模型恢复为载入时的状态，包括模型的位置，旋转，缩放，以及摄像机的位置和角度。

### 工具：

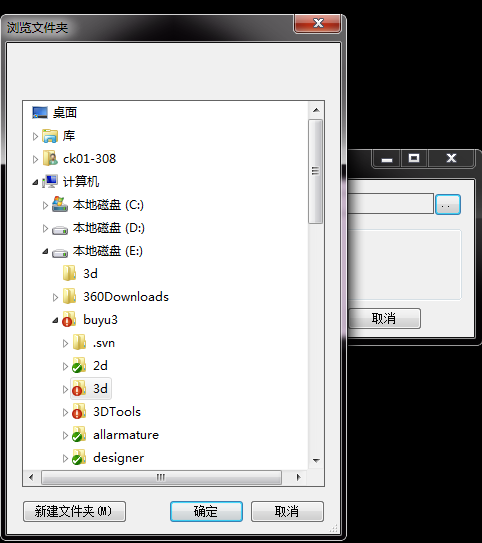
#### 批量转换ckb：

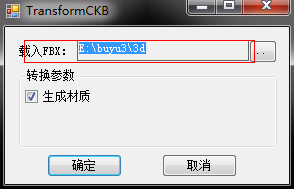
转换会将\*.fbx转换\*.ckb文件，\*.ckb文件才是在游戏中使用的文件格式，

点击“工具->转换ckb”菜单，会弹出批量转换对话框，如下图：



点击右边的浏览按钮，会弹出目录选择窗口，如下图：

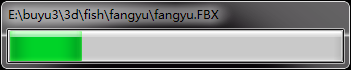


选择需要进行转换的目录后，点击“确定”就会将目录设置到批量转换窗口的路径区域，如图：

**转换参数**：

1. 生成材质：

是否在转换过程中，生成默认材质，点击“确定”按钮，开始进行批量转换，同时弹出批量转换进度窗口，如图：



### 碰撞盒：

#### 编辑：

开启对模型碰撞盒的编辑器，当勾选该选项后，模型编辑软件进入碰撞盒编辑状态，接下来所有的操作都是针对模型碰撞盒进行的，包括对碰撞盒的移动，缩放，添加和删除。

#### 添加：

选中骨骼后点击菜单的添加按钮，碰撞盒就会被绑定到对应的骨骼上，骨骼的运动会带动碰撞盒的运动，进而达到精确碰撞。如果是静态模型，也就是没有骨骼的情况下，碰撞盒会绑定到模型的中心点。

#### 删除：

从模型身上删除已经选中的碰撞盒。

## 快捷栏：

：文件保存。

：移动目标。

：缩放目标。

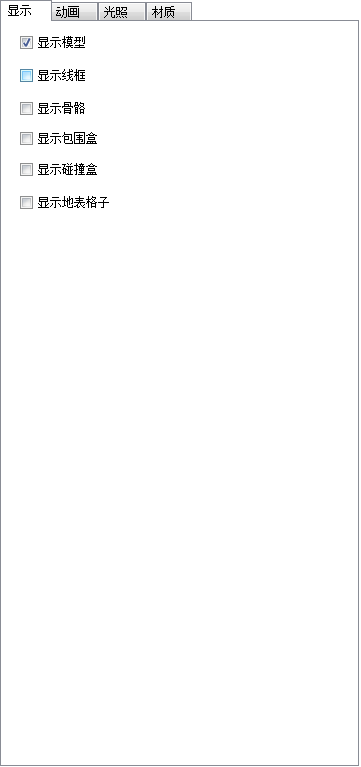
：旋转目标。

：正常鼠标状态。

## 控制面板：

### 显示标签：

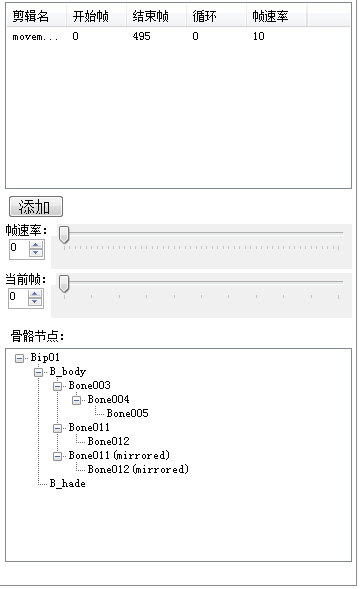
面板结构如图：



*  ：是否绘制模型，有的时候，用户可能需要把模型隐藏掉，来便于查看某些信息。
*  ：是否为线框绘制，线框绘制将模型用三角形勾勒出来，以便于查看模型导出后，三角形信息是否正确。
*  ：是否绘制骨骼，骨骼的绘制用于对模型动作的调试以及挂接碰撞盒及其他实体。
*  ：是否显示包围盒，包围盒是模型进行拾取的依据，正确的包围盒才能进行正确的拾取操作。
* ：是否显示碰撞盒子，碰撞盒子是进行精确碰撞的重要条件，模型身上的每个盒子都会对模型碰撞产生影响。
* ：是否显示地表网格，目的是用来衡量模型的大小。

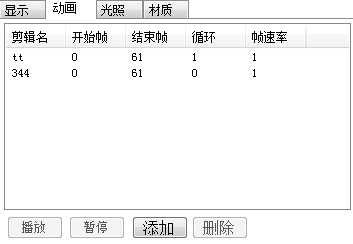
### 动画标签：

面板结构如图：



#### 动画剪辑列表：

如下图：



* 添加：

点击添加按钮后，会弹出“添加动画剪辑”对话框，如下图：



填入动画剪辑信息，然后“确定”添加到动画剪辑列表中，在对话框中，你可以填入如下信息：

1. 名称：动画剪辑的名字。
2. 开始帧：动画剪辑在整个动画序列中的开始帧。
3. 结束帧：动画剪辑在整个动画序列中的结束帧。
4. 是否循环：循环播放，还是单次播放。
5. 速度信息：动画剪辑的播放速度。

添加后，点击动画剪辑，即可播放对应的动画剪辑。默认的在载入模型后，会将整个动画序列作为一个动画剪辑添加到动画列表中，这样做的目的是，方便用户查看真个动画信息。

#### 帧控制：

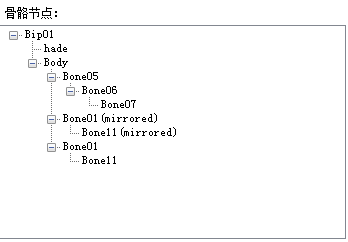
对当前动画剪辑进行控制，可以调节帧速率和单独播放某一帧，如下图。



1. 帧数率：鼠标拖动滚轴，可以调节当前动画的帧速率，方便用户对动画剪辑在不同速度播放时的状态。
2. 当前帧：显示动画剪辑播放的当前帧，这个当前帧显示的是相对于动画剪辑的当前帧，而不是整个动画序列的当前帧。

#### 骨骼树结构：

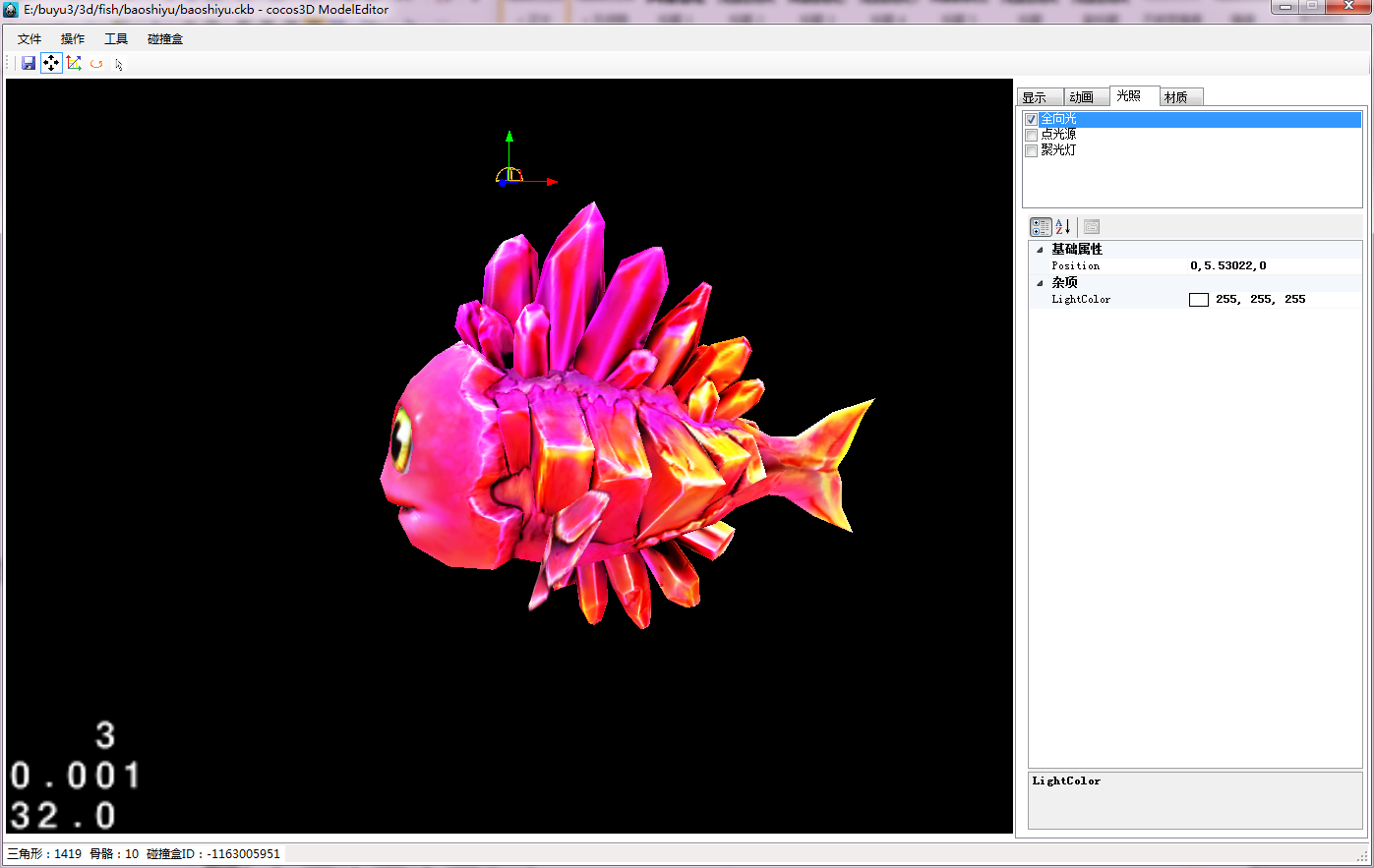
显示模型骨骼的树状结构，如下图：。



显示模型的骨骼树结构的目的是，用户可以选定某一骨骼后对骨骼进行绑定，例如：碰撞盒的挂接等。

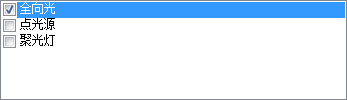
### 光照标签：

如下图：



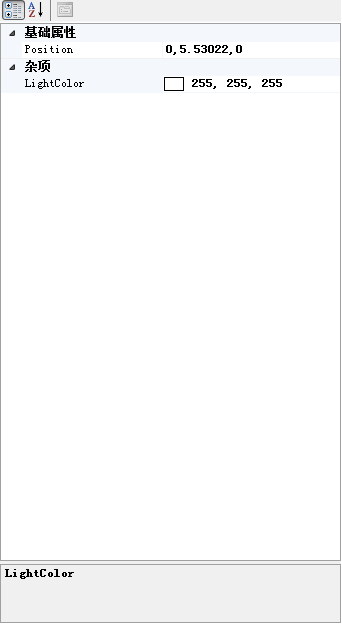
* 灯光列表：各种灯光的列表，勾选即为开启，否则为关闭。

如下图：



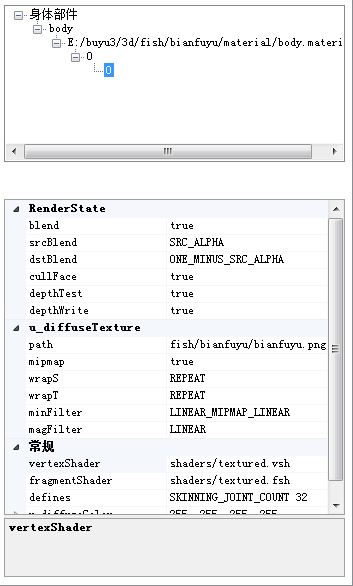
开启灯光后，对应的灯光模型会出现在编辑器中，可以通过鼠标对其进行移动，或旋转等操作。

* 灯光属性：不同的灯光可以配置不同的属性，目前支持3种光，分别是平行光，点光源，聚光灯，如下图：



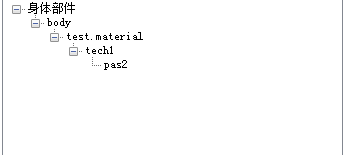
## 材质标签：

如下图：



#### 部件列表：

如下图：



部件树从上到下依次为：

1. 身体部件节点（身体部件）
2. 子模型节点（body）
3. 材质节点（test.material）
4. Technique节点（tech1）
5. Pass节点（pas2）

目前对材质的编辑支持添加，创建，删除材质，以及对材质中Technique和Pass属性的添加和删除,鼠标右键点击树节点，有相应的操作菜单。

###### Technique节点：

节点属性如下图：

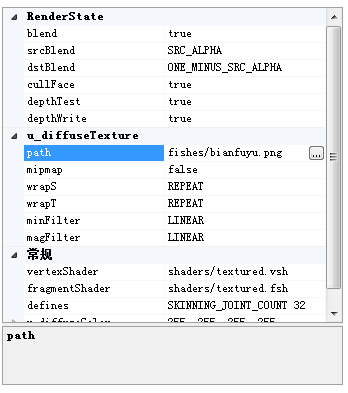


Technique属性包括：

1. Channel: 设置材质所在的渲染管线，属性包括：
2. none：无管线设置。
3. background： 背景管线。
4. opacity： 不透明管线。
5. transparency： 透明管线。
6. debug： 调试管线。

###### Pass节点：

材质属性面如下图：



对材质Pass属性的编辑主要包括以下信息：

RenderState: 渲染状态。

Blend：混合。

Srcblend：资源混合。

Desblend：目标混合。

Depthtest：深度测试。

Depthwrite：深度写入。

DiffuseTexture:采样

Path：贴图路径。

Mipmap：是否进行mipmap。

Wraps：mipmap参数。

Wrapt：mipmap参数。

Mipfileter：缩小过滤。

Magfileter：放大过滤。

Normal:常规设置

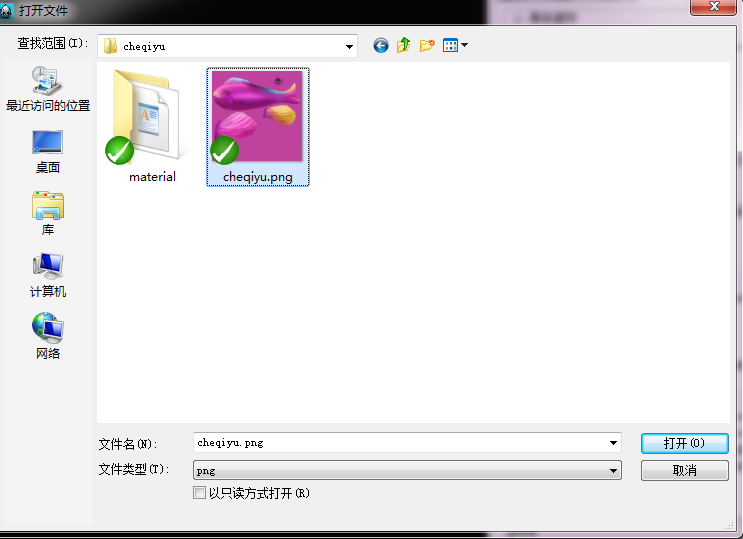
Vertexshader：顶点shader。

Fragmentshader：像素shader。

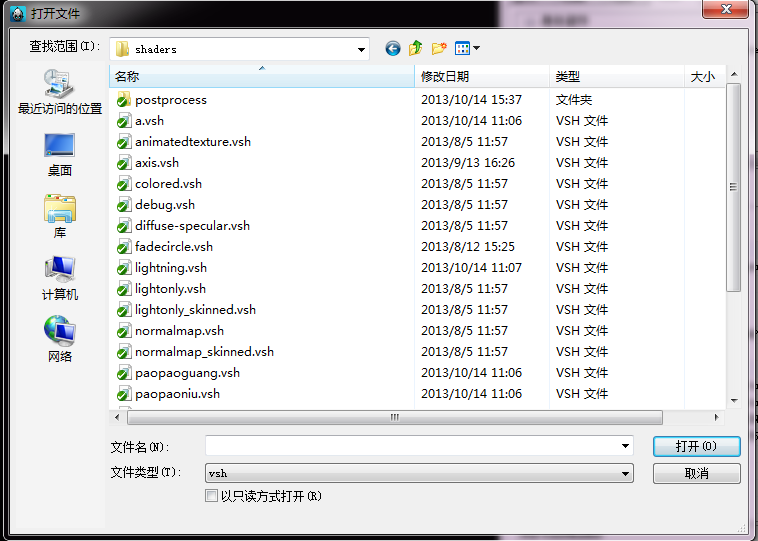
Defines：shader的宏定义。

Diffusecolor：顶点色。

点击path选项右边的按钮，会弹出贴图选择对话框，通过贴图选择对话框，来确定你要加载的贴图文件，如下图：



当点击shader，旁边的预览按钮时，会弹出shader选择对话框，通过对话框来选择相应shader文件，如图:



点击define选项旁边的按钮，会出现宏定义设置对话框，通过勾选可以确定传入shader中的宏定义，注意：编辑会自动提取shader中定义宏，但只能对shader中进行 #ifdef 或 #if defined( )处理过的宏进行提取，如果您想通过编辑器去编辑您在shader中的宏定义变量或及宏开关，请在shader中使用#ifdef 或 #if defined( )对宏进行预处理，如下图：

