- GLSL显示3D图片效果
  - 1、设置图层
  - 2、设置上下文
  - 3、清除缓冲区
  - 4、设置渲染缓冲区
  - 5、设置帧缓冲区
  - 6、渲染（需准备顶点、片元着色器）
    - 1、设置基本参数
    - 2、初始化program
      - 1、获取顶点、片元着色器文件路径
      - 2、定义顶点、片元着色器变量
      - 3、编译着色器
        - 1、根据路径获取着色器字符串内容，并转化为C字符串
        - 2、创建相应类型的着色器
        - 3、将着色器文件字符串内容与着色器变量绑定
        - 4、编译着色器
      - 4、初始化program变量
      - 5、将着色器变量附着到program
      - 6、释放着色器
    - 3、初始化坐标数组（顶点、纹理）、索引数组
    - 4、申请坐标数组缓冲区
    - 5、设置顶点着色器数据
      - 1、顶点坐标数据
      - 2、纹理坐标数据
    - 6、设置纹理
      - 1、获取纹理文件路径
      - 2、获取纹理数据
      - 3、申请纹理数据存储空间
      - 4、初始化上下文
      - 5、绘制纹理，然后释放上下文
      - 6、纹理属性设置
        - 1、绑定纹理状态
        - 2、设置纹理过滤、环绕方式（可选）
        - 3、载入纹理
        - 4、释放纹理数据
    - 7、设置投影矩阵

```
KSMatrix4 _projectionMatrix;
ksMatrixLoadIdentity(&_projectionMatrix);
float aspect = width/height;
//透视变换，视角30°
ksPerspective(&_projectionMatrix, 30.0, aspect, 5.0f, 20.0f);
glUniformMatrix4fv(projectionMatrix, 1, GL_FALSE, &_projectionMatrix.m[0][0]);
```

    - 8、设置模型视图矩阵

```
//模型视图矩阵
KSMatrix4 _modelviewMatrix;
ksMatrixLoadIdentity(&_modelviewMatrix);
ksTranslate(&_modelviewMatrix, 0.0, 0.0, -10.0);

//旋转矩阵
KSMatrix4 _rotateMatrix;
ksMatrixLoadIdentity(&_rotateMatrix);

//围绕X轴旋转
ksRotate(&_rotateMatrix, xrot, 1, 0, 0);
//围绕Y轴旋转
ksRotate(&_rotateMatrix, yrot, 0, 1, 0);
//围绕Z轴旋转
ksRotate(&_rotateMatrix, zrot, 0, 0, 1);

//模型视图矩阵与旋转矩阵相乘
ksMatrixMultiply(&_modelviewMatrix, &_rotateMatrix, &_modelviewMatrix);
glUniformMatrix4fv(modelviewMatrix, 1, GL_FALSE, &_modelviewMatrix.m[0][0]);
```

    - 9、渲染

```
glDrawElements(GL_TRIANGLES, sizeof(indexs) / sizeof(GLuint), GL_UNSIGNED_INT, indexs);
[self.context presentRenderbuffer:GL_RENDERBUFFER];
```