

OpenGL ES不容错过的实战-碰碰车



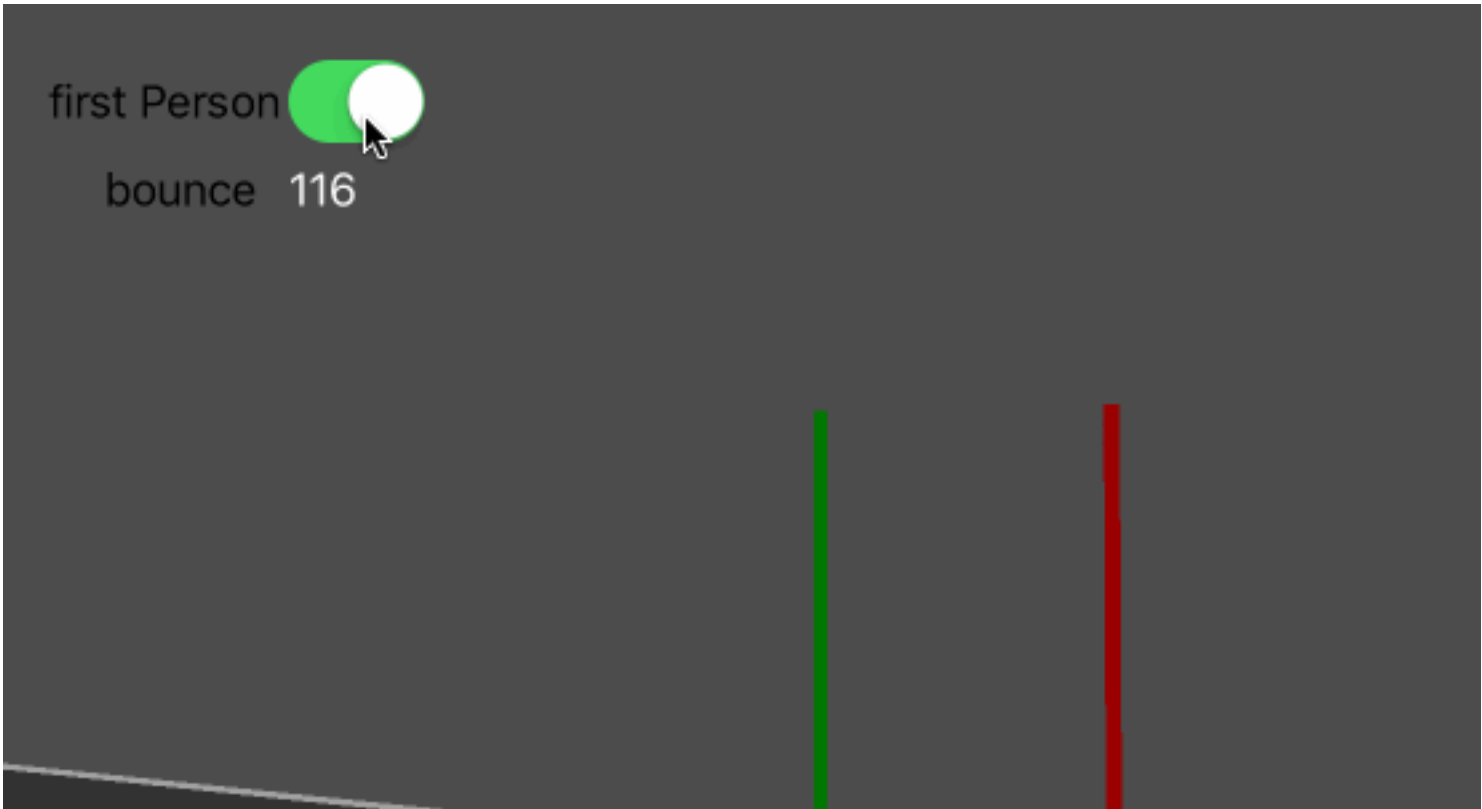
作者 落影loyinglin (/u/815d10a4bdce) [+ 关注](#)

2016.04.27 10:58 字数 1897 阅读 1921 评论 11 喜欢 24 阅读 1921 评论 11 喜欢 24 (/u/815d10a4bdce)

教程

OpenGL ES入门教程1-Tutorial01-GLKit (<http://www.jianshu.com/p/750fde1d8b6a>)
OpenGL ES入门教程2-Tutorial02-shader入门 (<http://www.jianshu.com/p/ee597b2bd399>)
OpenGL ES入门教程3-Tutorial03-三维变换 (<http://www.jianshu.com/p/87c5413c1fc7>)
OpenGL ES入门教程4-Tutorial04-GLKit进阶 (<http://www.jianshu.com/p/ed7fb9555839>)
OpenGL ES进阶教程1-Tutorial05-地球月亮 (<http://www.jianshu.com/p/a82f3f66ddddd>)
OpenGL ES进阶教程2-Tutorial06-光线 (<http://www.jianshu.com/p/4e1a28f23e75>)
OpenGL ES进阶教程3-Tutorial07-粒子效果 (<http://www.jianshu.com/p/b6d2441209f8>)
OpenGL ES进阶教程4-Tutorial08-帧缓存 (<http://www.jianshu.com/p/1193b98634a2>)
这一次是碰碰车。非常重要的一节，是前面教程的一个应用。
这一次的内容较多，包括复杂顶点模型、**第一和第三人称视角变化**、万向节锁、**物理碰撞模拟**、平滑动画与高低通滤波器、**模型封装**、材质绘制。
这一次的教程会尽可能多详细介绍。

效果展示



核心思路

通过加载头文件的顶点数据，得到复杂的顶点模型--车和场景。
先绘制场景，再单独绘制每一辆车。
第三人称视角固定为俯视角，第一人称视角选取一辆车作为视点所在，视线为车的前进方向。
当第一、第三人称视角切换的时候，通过滤波器来达到视线平滑过渡。
物理碰撞通过向量运算来模拟。

细节解析

1、模型加载

模型放在头文件，car的数据放在 bumperCar.h 包括 bumperCarVerts 顶点位置， bumperCarNormals 顶点法线，总共有 bumperCarNumVerts = 1164 个顶点。
场景的数据放在 bumperRink.h ，格式和car一致，顶点较少，只有 bumperRinkNumVerts = 180 个顶点。

2、模型封装

- SceneMesh 类是网格类，通过 AGLKVertexAttribArrayBuffer 管理顶点数据，发送顶点数据到GPU，分配顶点数据内存，绘制顶点数据。
- SceneModel 类是模型类，属性 axisAlignedBoundingBox 放置了模型的最大最小边界，属性 mesh 是模型的网格类，管理顶点数据。
- SceneCarModel 类是car的模型类，包括car的顶点数据和模型的基本属性，可以绘制 car模型。
- SceneRinkModel 类是场景的模型类，包括场景的顶点数据和边界等基本属性，可以绘制场景。
- SceneCar 类是car的逻辑类，包括car的速度、位置、偏航角、半径，还有滤波器函数、cars的碰撞处理、car与场景的碰撞处理、绘制car模型。

观察 SceneCar 类的初始化函数

```
- (id)initWithModel:(SceneModel *)aModel
    position:(GLKVector3)aPosition
    velocity:(GLKVector3)aVelocity
    color:(GLKVector4)aColor;
```

SceneCar需要SceneModel（模型类）、position（位置）、velocity（速度）、color（颜色）来初始化。
注意，这里的SceneCar并没有依赖SceneCarModel，而是依赖抽象（基类）SceneModel，实现了解耦。可以新建一个SceneOtherCar继承SceneModel，传递给SceneCar，不需要修改SceneCar的代码就可以创建出一个新的car。

3、视角变化

视角通过函数 GLKMatrix4MakeLookAt 确定，参数如下：

```
GLKMatrix4 GLKMatrix4MakeLookAt(
    float eyeX, float eyeY, float eyeZ,
    float centerX, float centerY, float centerZ,
    float upX, float upY, float upZ)
```

GLKMatrix4MakeLookAt() 函数会计算并返回一个model-view矩阵，这个矩阵会对齐从眼睛的位置到看向的位置之间的矢量与当前视域的中心线。
GLKMatrix4MakeLookAt()的详细解析会在接下来的教程--**基于视锥体（平截体）的OpenGL ES性能优化篇**详细介绍，目前可以按照参数来理解：假设有一个人，他的眼睛在前三个参数指定的位置，眼睛望向中间三个参数的位置，头的朝向为最后三个参数的方向，up（0，1，0）为标准方向。

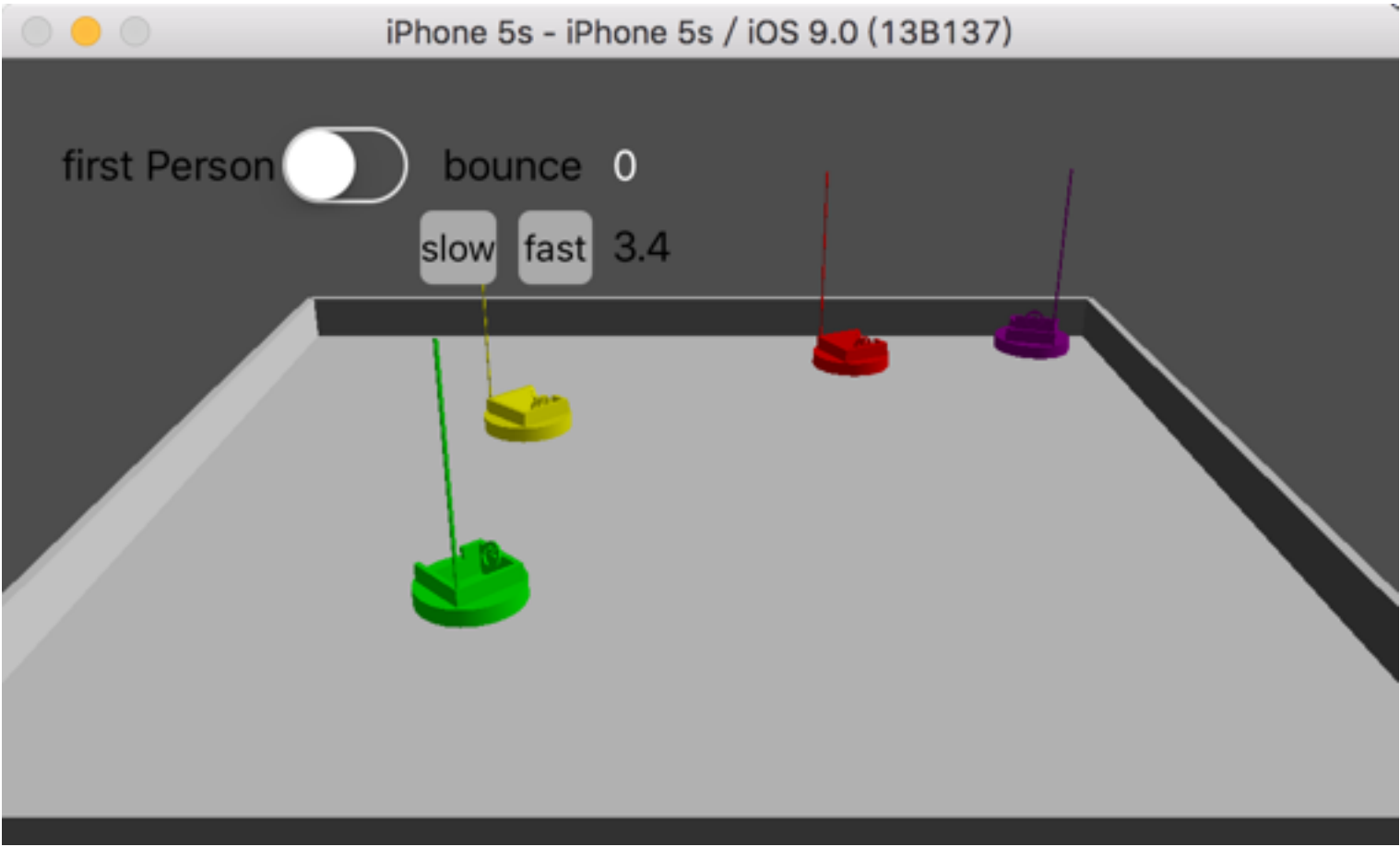
- 第一人称视角如下。

```
self.eyePosition = GLKVector3Make(10.5, 5.0, 0.0);
self.lookAtPosition = GLKVector3Make(0.0, 0.5, 0.0);
```



- 第三人称视角如下。

```
self.targetEyePosition = GLKVector3Make(viewerCar.position.x,
                                         viewerCar.position.y + 0.45f,
                                         viewerCar.position.z);
self.targetLookAtPosition = GLKVector3Add(_eyePosition, viewerCar.velocity);
```



细心的同学已经发现有 `eyePosition` 和 `targetEyePosition` 两个position。
`eyePosition`表示的是当前eye所在的位置，`targetEyePosition`表示的eye最终的目的。
之所以需要设置一个目标的位置，是为了做视角的切换，通过高通滤波器函数 `SceneVector3FastLowPassFilter` 和低通滤波器函数 `SceneVector3SlowLowPassFilter`，实现视角平滑过渡。

4、滤波器函数

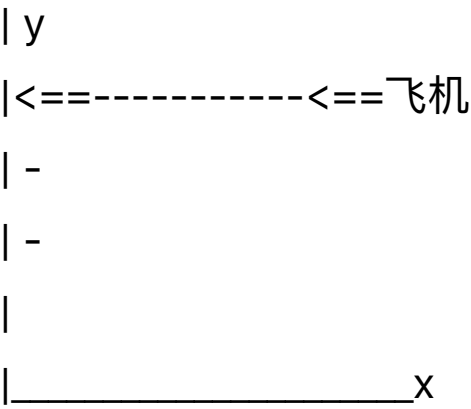
- 高通滤波器函数，50.0是一个可替换的较大的常数。可以模拟撞墙后震动的效果，因为50.0比较大，`current`值再增加后可能超过`target`。

```
GLfloat SceneScalarFastLowPassFilter(NSTimeInterval elapsed,
                                     GLfloat target,
                                     GLfloat current)
{
    return current + (50.0 * elapsed * (target - current));
}
```

- 低通滤波器函数，4.0是一个可替换的较小的常数。可以模拟视角切换过程的效果，因为4.0比较小，`current`会逐渐接近`target`。

```
GLfloat SceneScalarSlowLowPassFilter(NSTimeInterval elapsed,
                                      GLfloat target,
                                      GLfloat current)
{
    return current + (4.0 * elapsed * (target - current));
}
```

5、万向节锁



如上，假设地面的正北方向为x轴，正朝上为y轴。现在左下角的原点处有一台望远镜，它通过(a, b)来望向天空中的飞机，a为与望远镜与x轴正方向的夹角，我们用偏航角来解释；b为望远镜向上抬起后与地面的夹角，我们用高度角来解释。

对于任意一台坐标为(a, b)的飞机，我们都可以通过**先偏移a偏航角，再抬起b高度角**，从而观察到飞机。

现在假设有一台坐标为(0, 45)的飞机，朝望远镜飞来。飞机在望远镜的正北方向，偏航角为0，高度角为45。随着飞机不断接近望远镜，偏航角不变，高度角不断变大。

当飞机飞到望远镜的正上空的时候，坐标为(0, 90)。

此时，如果飞机转向朝东方向飞，偏航角由0直接变为90，但是此时由于望远镜正朝上（之前的坐标为0，90），偏航角的改变没有意义（丢失一个自由度）。

本教程还不需要用到四元数来解决这个问题，只是作为背景了解一下。Google上面有很多解释，比我讲的好很多。

6、物理碰撞模拟

- car与墙壁的碰撞

通过 (SceneAxisAlignedBoundingBox)rinkBoundingBox 可以得到墙壁的最大小边界。根据car的radius属性得到半径，通过半径radius+nextPosition与rinkBoundingBox判断是否到达边界。

如果到达边界则把对应轴的速度向量反向。

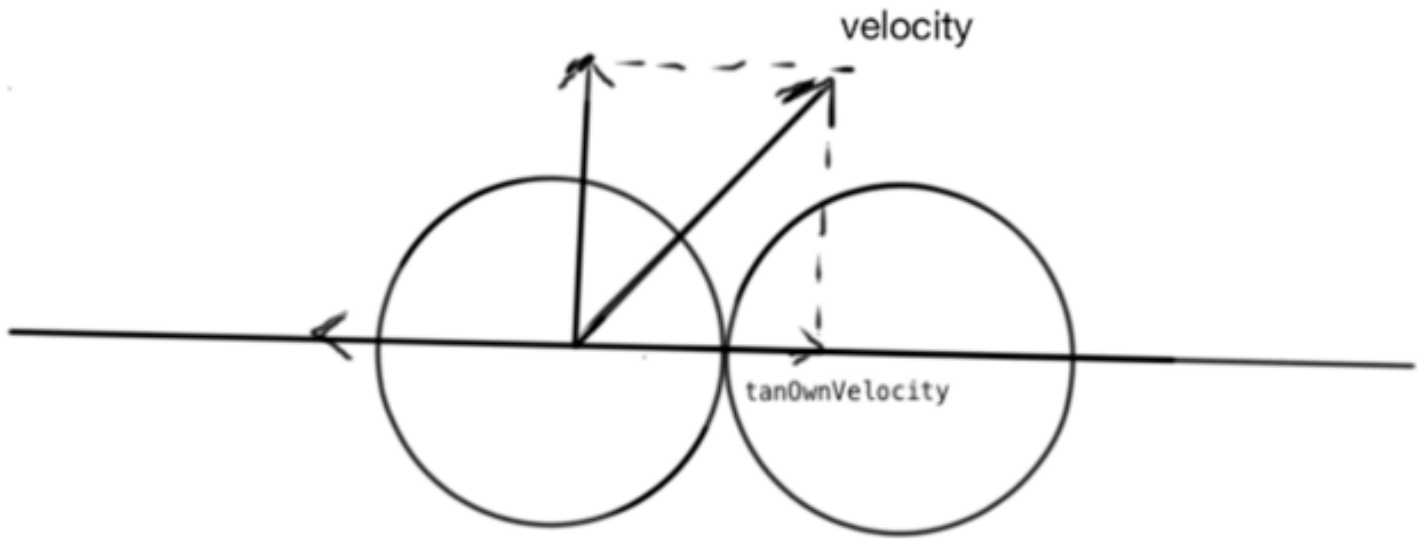
- car之间的碰撞

假设两辆车分别为self和other。

selfCar的速度为velocity、位置为position，otherCar的速度为otherVelocity、位置为otherPosition。

通过position和otherPosition，可以得到一条直线，car的碰撞就发生在这一条线上的方向上。

velocity在直线上的分量为tanOwnVelocity，otherVelocity在直线上的分量为tanOtherVelocity。



如上，碰撞完成后selfCar的速度为velocity - tanOwnVelocity，otherCar的速度为otherVelocity - tanOtherVelocity。

```
self.velocity = GLKVector3Subtract(ownVelocity, tanOwnVelocity);
currentCar.velocity = GLKVector3Subtract(otherVelocity, tanOtherVelocity);
```

7、材质绘制

GLKBaseEffect 有一个 material 属性，用于材质颜色的设置。
注意的是需要调用anEffect的prepareToDraw之后，再调用model的draw。

```
//设置材质
anEffect.material.diffuseColor = self.color;
anEffect.material.ambientColor = self.color;
[anEffect prepareToDraw];
[model draw];
```


思考

场地、车都是分别绘制的，他们的位置是如何控制的？当第一人称视角切换后，为什么车的显示会跟着变大？
这一部分作为思考题，就不给出答案了，在代码里面有注释。

总结

这一次的实现还是比较简单，模型是放在头文件，并没有用modellist文件来管理。
我在源代码的基础上，简化了不需要的代码和增加了很多注释。看完教程和代码的后，相信都能理解。

附上源码 (<https://github.com/loyinglin/LearnOpenGLES/tree/master/Tutorial09-%E7%A2%B0%E7%A2%B0%E8%BD%A6>)



落影loyinglin (/u/815d10a4bdce)

写了 171405 字，被 4763 人关注，获得了 2744 个喜欢

(/u/815d10a4bdce)写了 171405 字，被 4763 人关注，获得了 2744 个喜欢

+ 关注

工程师一枚，喜欢思考，喜欢游戏，喜欢运动。做过什么已经不重要，未来的方向以及当下的准备是生活的...

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button)

24

💬






🐼

🖼️


更多分享

(<http://cwb.assets.jianshu.io/notes/images/3712039/>)


被以下专题收入，发现更多相似内容

-  iOS Dev... (/c/3233d1a249ca?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS学习 (/c/1332c736fe39?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS移动端开发 (/c/6b7b728b037d?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS 开发 (/c/2ffaa203eb6a?utm_source=desktop&utm_medium=notes-included-collection)
-  OpenGL ... (/c/044a5240577d?)

utm_source=desktop&utm_medium=notes-included-collection)

 OpenGL ... (/c/408442c9c764?

utm_source=desktop&utm_medium=notes-included-collection)

 OpenGL+... (/c/5d2c87603bd3?

utm_source=desktop&utm_medium=notes-included-collection)

展开更多

推荐阅读

更多精彩内容

iOS开发-OpenGL ES实践教程（一） (/p/74d7ab73f0... (/p/74d7ab73f04e?

教程入门教程和进阶教程，介绍的是OpenGL ES基础，学习图形学基本概念，了解OpenGL ES的特性。实践教程是OpenGL ES在实际开发中的应用，demo的

落影loyinglin (/u/815d10a4bdce?

utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

真机的代码覆盖率测试 (/p/3a62abf85984?utm_cam... (/p/3a62abf85984?

代码覆盖率测试 以前虽然写过单元测试，但很少监测测试的完整程度，测试用例也经常存在重复的情况。这次在测试的要求下开始接入代码覆盖率测试。什么

落影loyinglin (/u/815d10a4bdce?

utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

为什么那个追了你很久的人突然放弃了？ (/p/8d0d277... (/p/8d0d277701ae?

1. 昨天，一位很好的女性朋友跟我说，之前有个男生追她，追了大半年，追得还蛮用心的，她都心动了，想着再考察一下，再过一两个星期就答应他，结果那

顾一宸 (/u/c1fed915ed12?

utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

有了框架思维，就再也不用担心碎片学习了 (/p/7ec3d... (/p/7ec3d87ef343?

1. 首先我要吐槽：韩剧《秘密森林》的男主角是我见过世上最面瘫的表演。但剧情没得说，从一个看上很简单，去快速找到嫌疑人的凶杀案，到最后——不能

彭小六 (/u/1441f4ae075d?


utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

我终于决定去做了近视眼手术..... (/p/62a5e5d67b71... (/p/62a5e5d67b71?

对近视眼手术长草也就一两个月的事，一开始是想去割个双眼皮（原谅我就是这么爱慕虚荣），去到医院才发现做了双眼皮手术，我600度的近视，还是得戴着


大千一墨 (/u/e99c77f07991?

utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

 登录/注册

为你个性化推荐内容

(/sign_in?utm_source=desktop&utm_medium=notes-included-collection)

 下载简书App

随时随地发现和创作内容

(/app/download?utm_source=desktop&utm_medium=click-note-bottom-bind)