这次的教程的目的是为了教会大家如何近似的实现一个最近比较火的游戏《天天过马路》并展示一下Cocos2d-X在3D方面的功能。

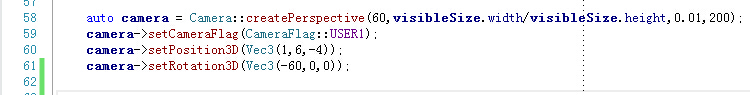
我们快速概览一下，天天过马路的游戏体验大致是怎么样的——玩家通过触摸的滑动，控制一个小人向四个方向移动，躲避车辆，尽可能朝前走足够远的距离，以期赢取高分。

车辆随机的从左右两侧跑出，你必须在他们的空隙穿过，否则你就会被压扁，当然，游戏中不可能让你一直提心吊胆——游戏会间隔的出一些人行道作为缓冲带，这些缓冲带里，不存在车辆，玩家不必担心在这里会被撞死，可是缓冲带里可能会有一些障碍物，这些障碍物，会阻挠用户的行动，如果用户并不是足够明智的话，可能在穿过马路时会把自己呆到一个很狭窄难以移动的地方。

以上就是这个游戏的大概游戏性内容，我们首先将其拆分成几个模块，我在每篇教程里只会涉及其中的一个。

首先，我们要让一个角色动起来，

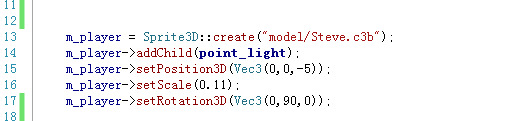
第一步，我们先创建一个相机，指定其投影的各个参数，并设置cameraFlag。



第二步，我们创建一个Sprite3D，作为模型，当然，考虑到可能会涉及到很多的关于玩家控制的角色的操作，我将其封装为了一个类Player



在其构造方法里，创建了一个Sprite3D，为了贴合原作，我使用了一个MineCraft模型，作为角色，并将其微调一下各种属性。



当然，不要忘记了，要将其设置cameraMask，使其能够正确的被指定相机进行视图-投影 变换.

C:\Users\ziwen.tang\AppData\Roaming\Tencent\Users\576419632\QQ\WinTemp\RichOle\70FHLII]6NI{C]YJ$5T3~{U.jpg。

这样，小人，就能显示出来了，但是它还不能随着我们手指(鼠标)的拖拽进行适当的移动，为了增加这样的功能。我们先创建一个listener，然后在里面监听touch的变动，并判断出其手势的样子：

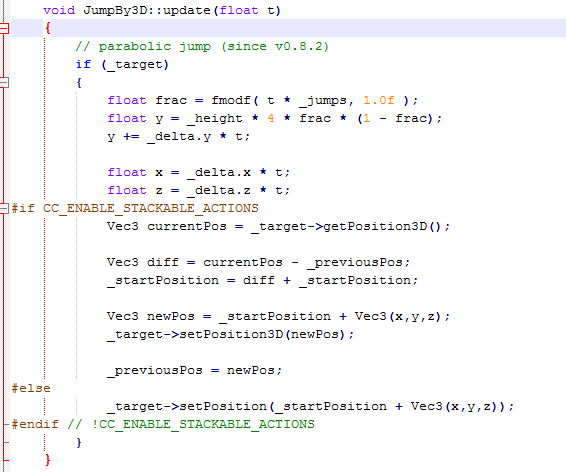


这里我们将触摸（点击）前后的坐标，转成向量，做出一个简单的判断，就可以知道用户到底是想往哪边移动，然后我们将这个方向信息传送到player对象的tryMove();方法。

在tryMove方法里，我们尝试对角色进行移动，请注意这是尝试，不一定会让角色真正意义上移动，还需要结合角色所处的位置，以及附近的障碍物，以及它是否是存活的状态——他可能此时已经被车撞死了：)

但是现在，我们先不考虑这个问题，只是单纯的考虑角色的移动问题，有两个要考虑，首先角色是跳着走的，其次相机应该会在稍微慢一些，有一个追随的感觉。为了要实现这两点，我们需要实现几个action，这里我扩展并实现了两个Action——JumpBy3D和moveTo3D。扩展其并不太复杂，大部分只需要将2d坐标扩展成3d的，在这里，我只写出update相关的部分：





有了这两个action后，实现原游戏的效果就非常的简单了。