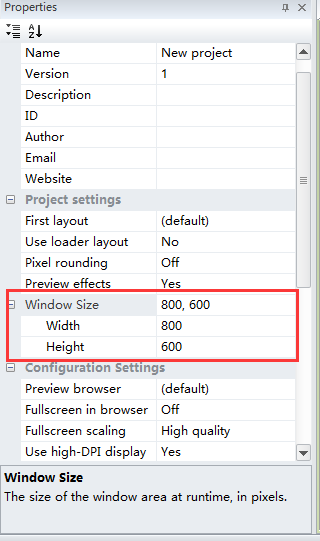
Bouncing bullets例子制作实验指导书

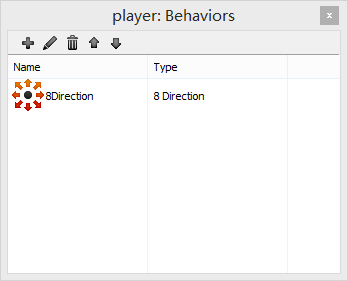
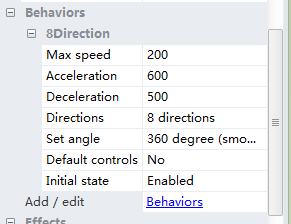
在课堂上我们讲了使用8-direction让player能够跟着我们的鼠标前进旋转，现在我们来使用键盘的输入实现player的运动。

**首先**，新建一个工程文件，window size 设为800\*600

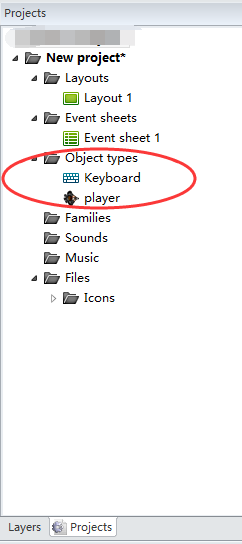


**第二步，**添加我们的主角player。（此处太简单了就不细说，课堂上说过两种方法都可以，开心就好）

**接下来，**为我们的player添加行为behavior（这一步是建立在课堂上讲解的例子的基础上的）首先添加8-direction

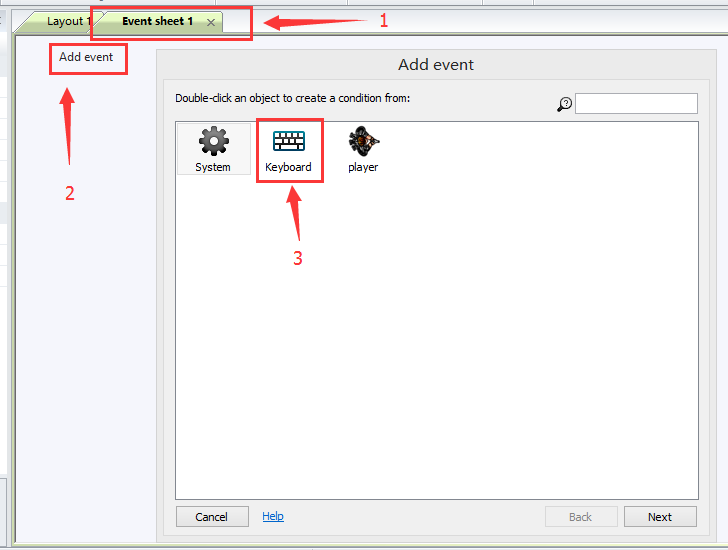


**第三步，**我们需要使用键盘键操控player，因此我们需要添加一个键盘object。点击编辑区的空白处，在跳出的窗口中找到keyboard（键盘），双击即可添加。

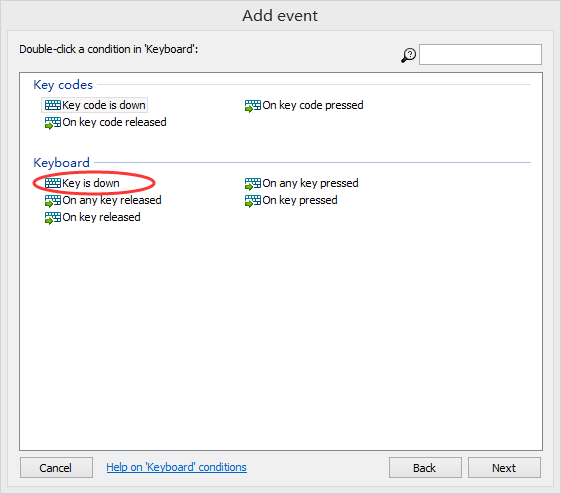


**第四步，**添加键盘事件。此处我们利用键盘的方向键来模拟player的上下左右运动。

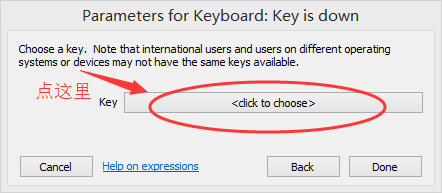
进入event sheet，点击 Add event，点击keyboard



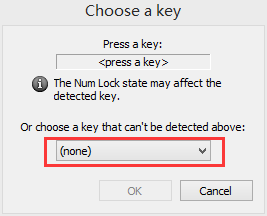
选择条件“当键盘被按下时”



然后

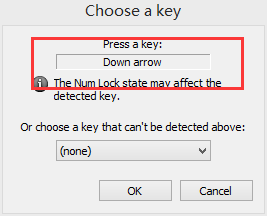


接着会跳出一个对话框，就是这样的



如果点击红色框的话，你会发现没有方向键的选项，怎么办呢？

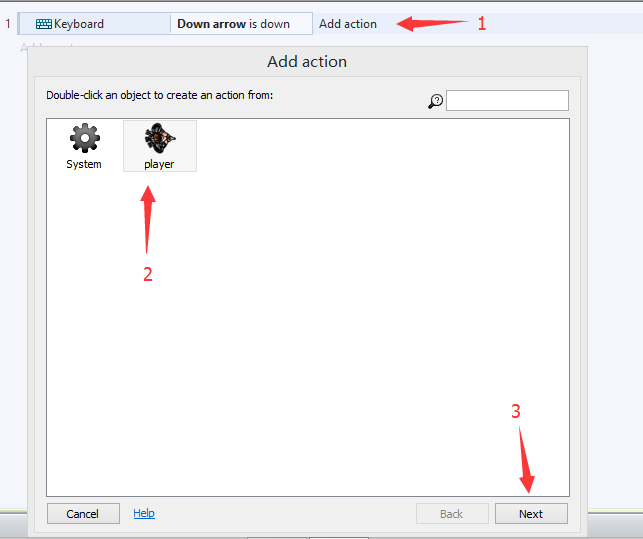
其实很简单，在这个对话框跳出后，我们只要按下键盘的向下方向键，神奇的事情就发生了



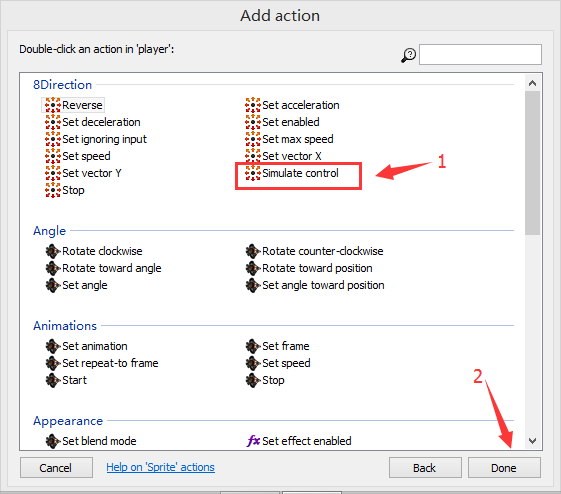
接着点击OK，再点击done，我们的条件就写好了。

F:\电脑\1332498225\FileRecv\MobileFile\Image\)4ZVN7Y)ZZGF_0FW9P9IR]4.png

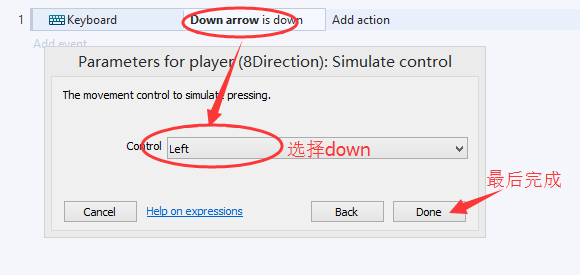
接着我们来添加动作，点击条件后面的Add action（解释一下，我们设定的规则是，当键盘向下箭头被按下，player就向下运动，此处条件是“当键盘向下箭头被按下”，因此加在键盘上，而动作是“player就向下运动”，动作的载体是player，因此我们在添加动作时选择player）



接下来，我们选择simulate control



实际上，我们利用键盘方向键操控player是模仿8-direction的运动，所以我们此处要选择8direction的simulate control（模仿控制）



F:\电脑\1332498225\FileRecv\MobileFile\Image\{B_Z4AY$EX`Q19YI1IVBLR7.png

按照相同的方法，我们将其他三个方向也添加完成（W、S、A、D键同理）



到这里我们就实现了通过按键盘方向键让player运动。我们已经完成了一半了，成功离我们不远了！

运行后，同学们会发现一个问题，当player走出窗口显示的游戏区以后就回不来了。解决思路有两种：一是让player不能走出窗口显示的游戏区，二是让player从右边出去再从左边回到游戏区。

第一种方法：添加（通过图标就可以明白，这个行为的作用就像一堵墙，使

player被挡在游戏区内）

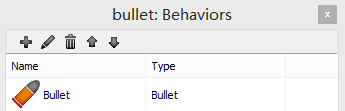
第二种方法：添加（通过图标也能清楚地明白其作用，让玩家能够右出左进）

接下来，我们来加子弹吧！

添加子弹素材，为了美观，放在游戏区外

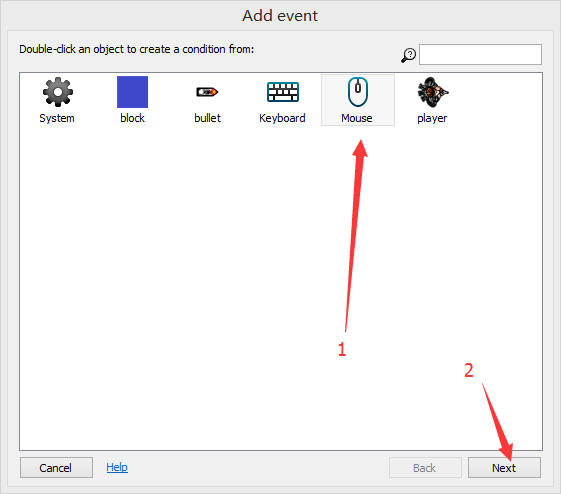


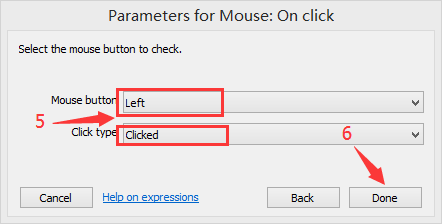
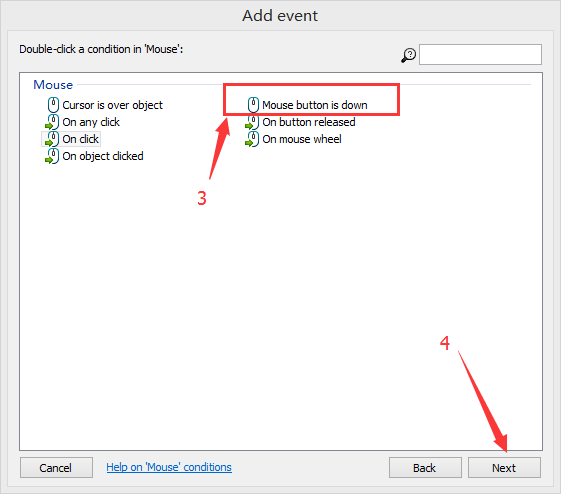
然后为子弹添加一个bullet行为，让子弹能在发射后向前运动。



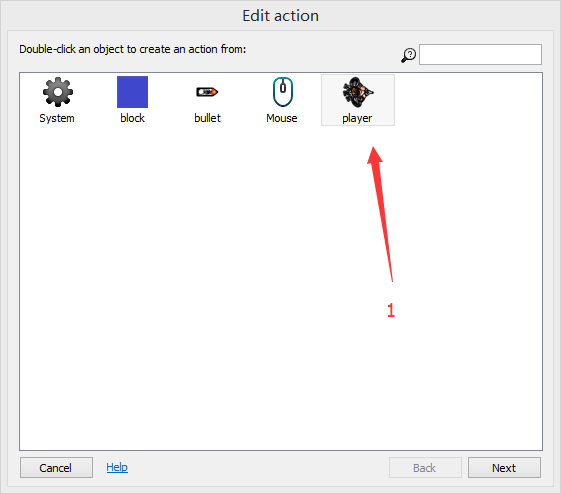
子弹会在我们点击鼠标左键时被发射，因此我们需要添加一个鼠标的object（与添加keyboard方法相同）

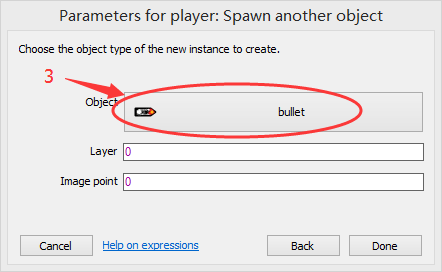
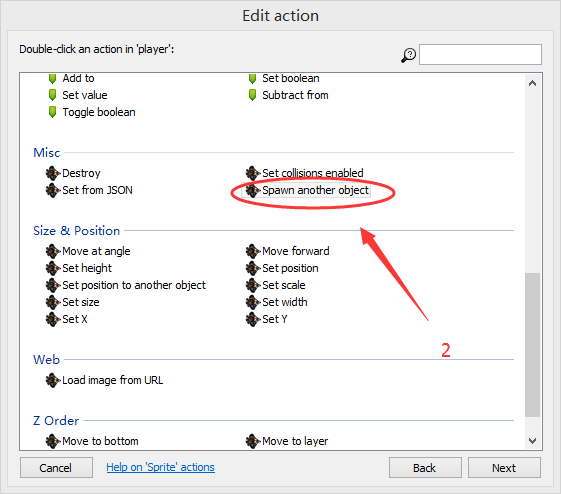
子弹发射事件的条件是“鼠标左键被按下时”，因此我们添加这样的条件





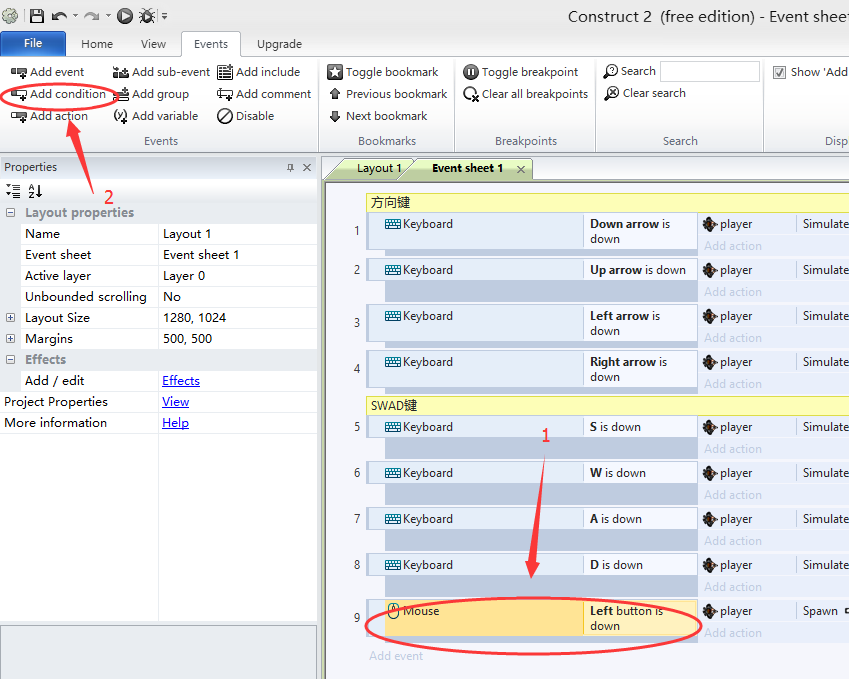
接下来添加动作，使得我们在点击鼠标左键时，在player的位置会生成子弹（swap another object生成另外一个物体）

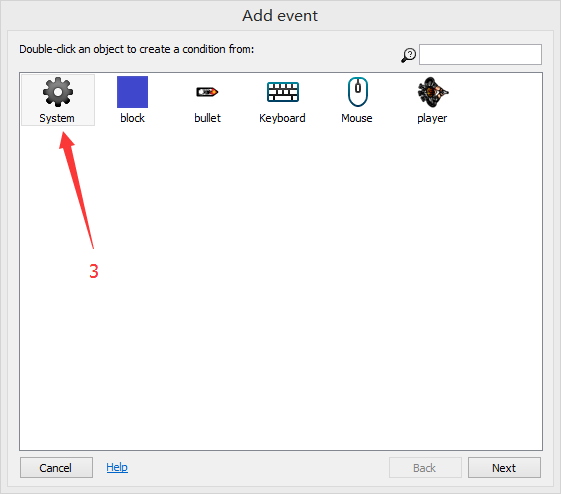


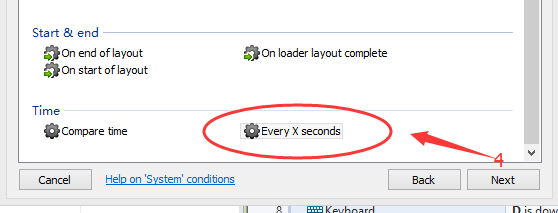


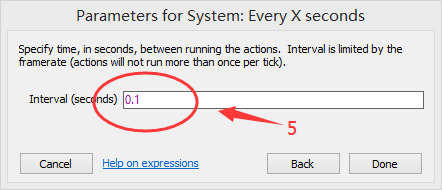
子弹添加完成，现在我们可以发射子弹了！

但是大家会一个问题，如果我连续点击鼠标，子弹会穿成一条，这样给玩家的体验就不是很好了。因此，我们的设想是，每隔0.1秒子弹发射一颗，这样无论鼠标点多快，子弹都能一颗一颗射出。





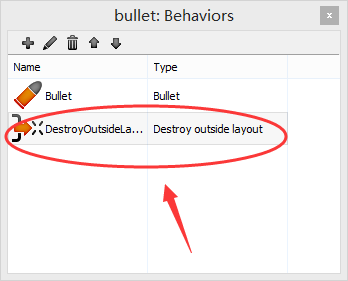




OK，这样子弹就能正常发射了。

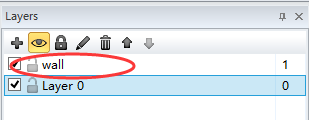
现在大家想一想，如果我们一直发射子弹，子弹会一颗颗生成，虽然飞出了游戏区，但实际上它们仍然存在，并占用内存，日积月累的话会占用很大很大的内存，因此我们需要在子弹飞出游戏区后就将其销毁。

这里我们用到一个新的behavior：该行为能使object在超出游戏范围会被销毁

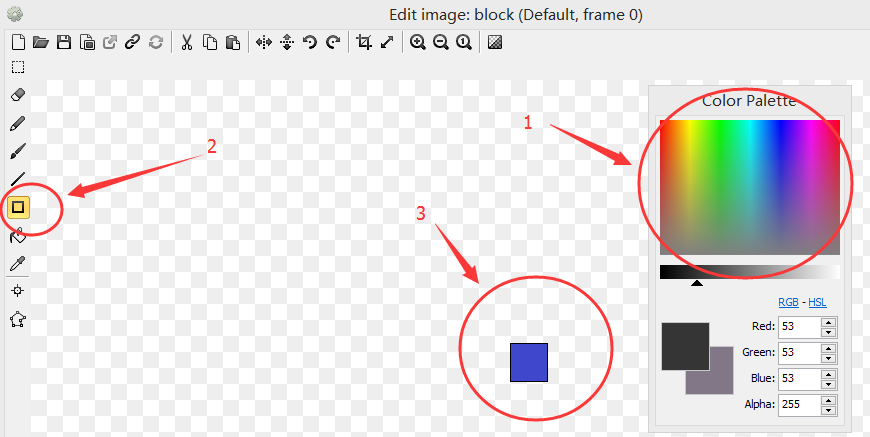


现在，为了让我们的游戏更加有趣，我们可以添加一些障碍，使得子弹打到障碍后会反弹

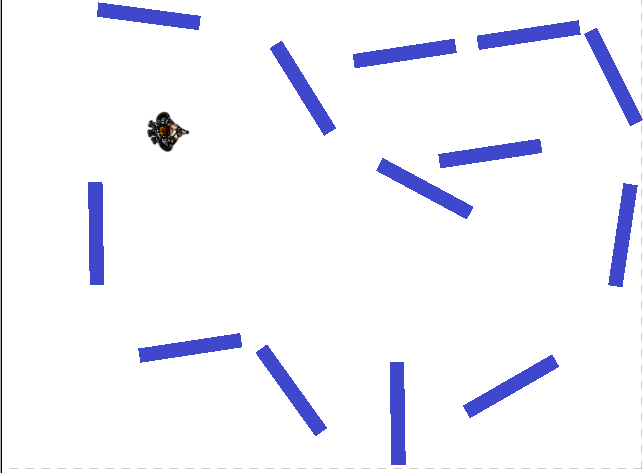
首先新建一个layer（图层），命名为wall（此处随意，也可以是障碍物等）



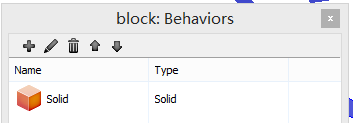
障碍物的添加：双击编辑区空白处，添加一个sprite，然后在绘制区选择喜欢的颜色，画出一个方块



然后通过复制粘贴，就能得到如下

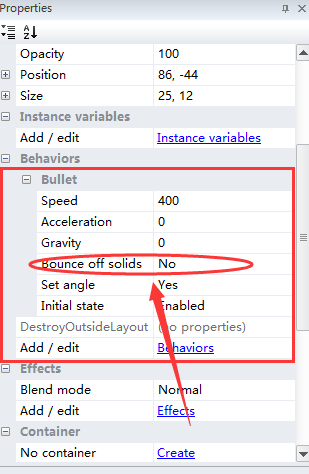


为了能让子弹反弹，子弹和墙需要发生碰撞，因此为我们的墙添加一个solid行为



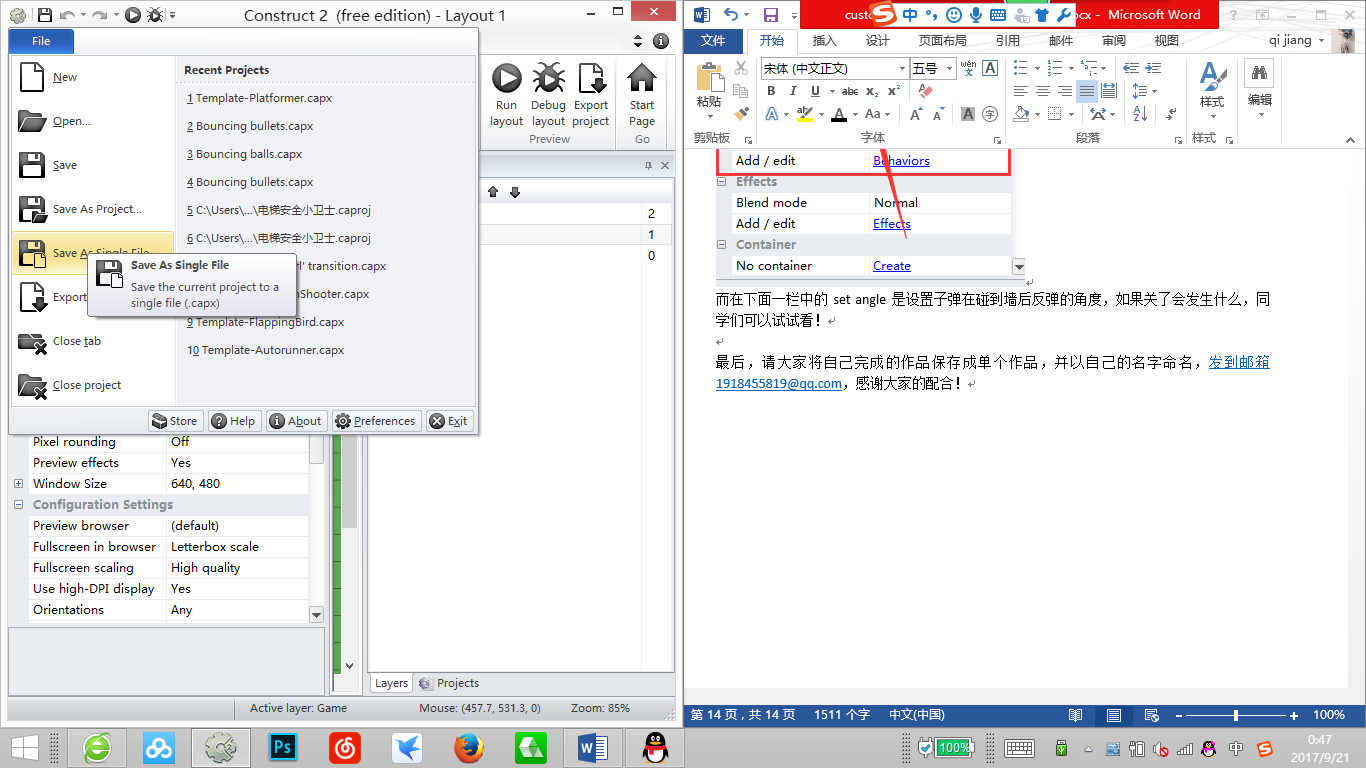
运行后同学们会发现，我们的墙似乎挡不住子弹。这是因为我们没有将子弹的碰撞检测打开

单击子弹，在属性栏里，要将bound off solid 一栏改为yes



而在下面一栏中的set angle是设置子弹在碰到墙后反弹的角度，如果关了会发生什么，同学们可以试试看！

最后，请大家将自己完成的作品保存成单个作品，



并以自己的名字命名，[发到邮箱1918455819@qq.com](mailto:发到邮箱1918455819@qq.com)，感谢大家的配合！