项目总结之触摸问题分析

DionysosLai([906391500@qq.com](mailto:906391500@qq.com)) 2014/12/24

在游戏开发领域，有几个问题，一定要提前进行真机测试，越早进行真机测试，后期修改代码越容易。其中有一个问题，就是触摸问题。

为什么触摸问题要尽早进行真机测试呢？这是由于，只有这真机运行情况下，才能实现多点触摸问题（想想多个手指在你的游戏上滑动，各种各样的bug都会一一呈现出来）。同时，有些特定的触摸响应，比如touchCanclled事件响应，也只有在真机情况下才能出现。至于如何重现touchCancle事件引起的bug，待会儿进行分析。

**触摸事件响应响应混乱问题：**

之前做过项目里面做过一个游戏，游戏内容是关于时下最热门游戏类型之一：三消类游戏。由于，做的是儿童教育产品，因此在难度上会降低。所谓的三消，有时候只需要点击一个物品，也能够消除，同时又必须可以滑动选定多个物品，同时消除。因此，比市场上三消游戏稍微复杂一点。

由于点击一个物品也能够消除，因此出现了狂点屏幕情况。后来，经过测试，发现游戏一个手指点击选中物品时，另一个手指选中另一个物品，同样能够进行消除，而原来选中的物品，状态一直处于touchMove中。这明显是没有响应了touchEnd事件了。

经过分析，问题在于touchEnd事件被其他触摸给响应了。后来，在项目中的另一个游戏，实现的是一个拖动物品操作，即物品跟着手指移动。这个响应的是简单的单点触摸。在开发过程中，发现一个有趣问题，就是一个手指拖动物品时，另一个手指去滑动屏幕，有时候物品会跟着另一个手指移动了。这些问题很普遍存在，有时候也是一个游戏乐趣。Dr.Panda游戏《Toy Cars》，就出现了这种情况，不过感觉很好玩。

要解决上面问题出现，首先要明白一点就是：每个触摸都是有ID值，这个ID值是这个触摸的身份标识，只有ID值一直，才是原来那个手指的触摸信息。理解这点，就很容易解决触摸混乱问题了。

**在单点触摸中解决触摸混乱问题：**

由于单点触摸，touchBegan只有返回true，才能将触摸信息向下传递。因此触摸ID值设定必须只有在返回true时，才能设定。

类似代码如下：

int m\_touchId; ///< 触摸ID声明

m\_touchId = -1; ///< 初始化

/// touchBegan事件处理

bool ShowBox::ccTouchBegan( cocos2d::CCTouch \*pTouch, cocos2d::CCEvent \*pEvent )

{

……

if (-1 != m\_touchId)

{

return false;

}

If( …)

{

m\_touchId = pTouch->getID();

return true;

}

else

{

return false;

}

}

/// touchMove事件处理

void ShowBox::ccTouchMoved( cocos2d::CCTouch \*pTouch, cocos2d::CCEvent \*pEvent )

{

if (pTouch->getID() != m\_touchId)

{

return;

}

…..

}

/// touchEnd事件处理

void ShowBox::ccTouchCancelled( cocos2d::CCTouch \*pTouch, cocos2d::CCEvent \*pEvent )

{

if (pTouch->getID() != m\_touchId)

{

return;

}

else

{

m\_touchId = -1;

}

….

}

**在多点触摸中解决触摸混乱问题：**

在多点触摸中，涉及到到多个触摸ID值，因此会显得比较灵活。仅介绍第一个触摸ID处理，其他类似。代码如下：

int m\_touchId; ///< 触摸ID声明

m\_touchId = -1; ///< 初始化

/// touchBegan事件处理

void GameScene::ccTouchesBegan( cocos2d::CCSet \*pTouches, cocos2d::CCEvent \*pEvent )

{

....

if(pTouches->count()==1 && -1 == m\_iTouchID) //如果触摸点为一个

{

m\_iTouchID = ((CCTouch\*)pTouches->anyObject())->getID();

.....

}

....

}

/// touchMove事件处理

void GameScene::ccTouchesMoved( cocos2d::CCSet \*pTouches, cocos2d::CCEvent \*pEvent )

{

if(pTouches->count()==1 m\_iTouchID == ((CCTouch\*)pTouches->anyObject())->getID())

{

.....

}

}

/// touchEnd事件处理

void GameScene::ccTouchesEnded( cocos2d::CCSet \*pTouches, cocos2d::CCEvent \*pEvent )

{

...

if(pTouches->count()==1 && m\_iTouchID == ((CCTouch\*)pTouches->anyObject())->getID())

{

m\_iTouchID = -1;

}

}

注：pTouches->count()==1，实际作用是将多点变成了单点触摸了

**touchCanclled事件:**

一般我们做触摸事件处理时，很少会响应touchCanclled事件，因为这一个事件不是有用户操作产生的，一般是由程序产生。但是，如果我们不做响应touchCanclled事件，轻则游戏拖动物品出现混乱，重则导致游戏无法进行了。这得看大家自己做的容错处理能力了。

如何重新啊touchCancle事件引起的bug呢？

操作如下：在拖动或者操作游戏过程中，最好是在响应touchMove事件中，这时将按下设备的开机键（例如平板的锁屏键），那么整个设备将暂时关闭 ，我们的应用也将响应了touchCancle事件了，再次按下开机键，看看自己的游戏，相信会有令你惊讶的事情发生。

如何避免这个问题发生呢？很简单，将touchEnd中的代码复制一遍到touchCancle中即可。

触摸问题，在我整个项目过程中，目前就遇到这两个比较严重、难以发现的问题。至于其他诸如触摸优先级，相信大家都会，还比如在触摸中一般不写或者少些逻辑处理问题，大家也应该懂得。

祝大家平安节快乐！！！