基于视口的地图设计

DionysosLai 2014-06-14

第三人称游戏，玩家是处在“上帝视角”掌控着整个游戏。因此，游戏设计中，我们要让游戏主角处于整个屏幕中央或者屏幕的大部分区域内，像《超级玛丽》、《天天酷跑》等一些游戏，细心点玩家就会发现主角在移动过程中，基本处于在同一个位置、或者在一定范围内，主角移动时，背景不会移动，但是当超出整个范围内，那么背景就会跟着移动了。统统这些设计，基本上就是基于视口的地图设计了。比较经典例子可以算是《愤怒的小鸟》了，注意这里的游戏主角是发射出来的那一只小鸟。

对于这种地图的设计，首先先了解视口概念。视口，即设定主角在一般情况下，可以移动的范围，超出这个范围，那么地图也要跟着移动了。如果不是一般情况，比如说地图已经移动到屏幕边缘了，这是角色超出视口时，地图是不能跟着角色移动的，不能就看到黑边了。对于视口，如下图所示：

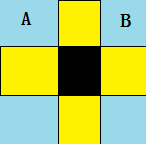


图1

黑色区域就是黑色范围了。但是，这样设计的视口存在一个问题，就是当角色由于某种原因时（也许只有上帝才知道这个原因吧），进入到了4个边角区域时，那么角色移动了，地图却不会移动（比如从A向B移动），而实际情况是地图也要跟着移动。这个问题，解决方法之一，就是判断地图是否到了屏幕边缘，否则做一些处理；解决方法之二，就是重新设定视口的概念。

第二种的视口设计如下：

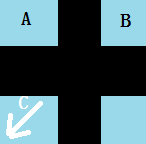


图2

视口这样设计，就可以避免上面存在的问题，但是有一个问题就是如果C朝箭头方向移动时，同样不会跨过视口，地图也不会移动了。为了解决这个问题，增加一条规定，就是如果角色在视口外了，且不向视口方向移动，那么同样地图也要跟着角色移动了。

Ok，到此，基于视口的地图设计基本就是这样了。

那么，下面就是问题，就是整体设计了。

由于，游戏中，要主角经常经常要处理一些碰撞，因此建议将主角放在地图同一层。由于这样设计，保存角色位置的不变动，在地图移动时，角色要朝着反方向移动了。

下面就是一些设计的一些重点了。

设计1 地图和角色的移动设计：

移动，归根到底，就三种移动方式：

1. 角色和屏幕一起移动；

2. 角色移动、屏幕不移动；

3. 角色和地图均不移动------比方说，角色碰到障碍物了，要停止下来。

代码如下：

enum /// 移动状态

{

MAP\_E\_MOVE\_ALL, ///< 地图和角色均移动

MAP\_E\_MOVE\_ROLE, ///< 角色移动

MAP\_E\_MOVE\_STOP, ///< 地图和角色均不能移动

};

设计2 如何设置更改移动状态：

时时检测三种情况，1. 角色是否在视口内；2. 角色在视口外了，但是否有朝着视口移动的趋势；3. 角色是否碰到障碍物了。

代码如下：

do

{

/// 首先预判断主角是否将会超出视口外并且当前在视口内

if (isRoleInView(ccp(0,0)))

{

m\_iMoveState = MAP\_E\_MOVE\_ALL; ///< 地图和主角一起移动

}

else

{

/// 已经在视口外面了，就判断是否有想视口移动趋势

if (roleToView(pointBy))

{

m\_iMoveState = MAP\_E\_MOVE\_ROLE; ///< 主角移动

}

else

{

m\_iMoveState = MAP\_E\_MOVE\_ALL; ///< 地图和主角一起移动

}

}

if (isCollision(ccpMult(pointBy, 2.f)))

{

m\_iMoveState = MAP\_E\_MOVE\_STOP;

}

} while (0);

设计3 地图调整

根据移动状态，每时每刻都要移动地图，那么需要记录每一帧移动的大小，再判断 是否超出屏幕了，根据上下左右四个方向调整位置。

设计4 如何移动

不建议使用引擎自带的moveBy函数，这是由于我们要时刻检测原因。因此，需要记录我们要移动的位置，根据角色位置（相信在平板上，玩家点击的位置，就是希望角色到达的位置），换算出相对向量。然后每一帧要移动距离乘于相对向量的单位向量了。

代码如下：

float m\_fMoveDistance; ///< 记录每次移动的距离

cocos2d::CCPoint m\_moveVector; ///< 移动向量

float m\_fMoveSpeed; ///< 移动速度

设计5 地图显示设计问题

屏幕大小就这么大，一张地图，就有一个屏幕大小，一个屏幕最多只能显示四张地 图，因此，我们可以让不在屏幕内的地图暂时消失起来，这样可以大大提高游戏效率。 ---在示例Demo中，我们设计了100\*100的地图，运行起来非常流畅。同样，如果有相

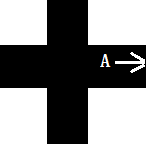
同的地图，可以使用CCSpriteBatchNode。

代码如下：在函数void setMapVisible(void);中

其实，对于一些不在地图中显示道具、物品也可以采用这个方法。目前代码中，没有这一 优化，后期要加上。

好了，基本上问题的解决了。源代码下载，请登入我的GitHup网址：<https://github.com/DionysosLai/MapLayer>。欢迎大家下载。

对于，目前的设计,经过了游戏《超级挖地机》的检测，不存在问题。但存在一个设计bug，完全无法避免，就是在如图的情况下：



A朝着箭头的发现移动，如果A比较偏向屏幕右边，那么移动感觉好差啊。