**Unity3D教程：游戏中跟踪弹的基础算法**

Posted on 2013年04月28日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 154 次

射击类游戏中经常需要跟踪算法的。其实，只需要一点数学知识就可以实现。

首先，需要三角函数sin、cos、tan(tg)、arctan(arctg)。

sin(x)—对边/斜边。在1，2项限为正，3，4项限为负

cos(x)—邻边/斜边。在1，4项限为正，2，3项限为负

tan(x)—对边/邻边。在1，3项限为正，2，4项限为负

考虑到游戏里面的坐标系如下所示：

假设敌人子弹的坐标为slug.x、slug.y、子弹的速度为slug.speed(全部是double型)。上面的三角形的斜边就代表子弹的速度，则子弹每次移动的时候座标的改变为：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | slug.x += slug.speed \* cos(theta); |
| 2 |  |
| 3 | slug.y += slug.speed \* sin(theta); |

在敌人子弹向你发射过来的时候，首先要计算子弹位置与你所在的位置所夹的角度theta。简单计算就是：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | **double** deltax = player.x - slug.x; *// 注意，是以主角位置为起点 在上图中表示就是x1-x0* |
| 2 |  |
| 3 | **double** deltay = player.y - slug.y; *// y1-y0* |
| 4 |  |

为了防止在相除的时候分母为0，做一个判断,使分母近似为0，究竟是负的近似还是正的近似呢？这就需要比较子弹和你的Y坐标谁大谁小了。

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **if**( deltax == 0 ) |
| 02 |  |
| 03 | { |
| 04 |  |
| 05 | **if**( player.y >= slug.y ) *// 子弹需要下移* |
| 06 |  |
| 07 | deltax = 0.0000001; |
| 08 |  |
| 09 | **else** *// 子弹需要上移* |
| 10 |  |
| 11 | deltax = -0.0000001; |
| 12 |  |
| 13 | } |

同理，对deltay作判断。

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **if**( deltay == 0 ) |
| 02 |  |
| 03 | { |
| 04 |  |
| 05 | **if**( player.x >= slug.x ) *// 子弹需要右移* |
| 06 |  |
| 07 | deltay = 0.0000001; |
| 08 |  |
| 09 | **else** *// 子弹需要左移* |
| 10 |  |
| 11 | deltay = -0.0000001; |
| 12 |  |
| 13 | } |

现在对角度所处的项限作判断：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **if**( deltax>0 && deltay>0 ) |
| 02 |  |
| 03 | angle = atan(fabs(deltay/deltax)); *// 第一项限* |
| 04 |  |
| 05 | **else** **if**( deltax<0 && deltay<0 ) |
| 06 |  |
| 07 | angle = π-atan(fabs(deltay/deltax)) *// 第二项限* |
| 08 |  |
| 09 | **else** **if**( deltax<0 && deltay<0 ) |
| 10 |  |
| 11 | angle = π+atan(fabs(deltay/deltax)) *// 第三项限* |
| 12 |  |
| 13 | **else** |
| 14 |  |
| 15 | angle = 2π-atan(fabs(deltay/deltax)) *// 第四项限* |

其中π取3.1415926。现在已经得到正确的方向了，可以计算子弹坐标了！

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | slug.x += slug.speed \* cos(theta); |
| 2 |  |
| 3 | slug.y += slug.speed \* sin(theta); |