**Unity3D教程：Motion Trail Script（Unity运动轨迹的脚步）**

Posted on 2013年05月14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 125 次

原理：

游戏用到的Motion Trail原理其实很简单，就是利用一个Queue，将两参考点于每个Frame的位置记录下来，这记录通常是有条件的，也许是两参考点与上一个Frame的纪录比较距离，若高过某一threadhould就push in，此外在每个Frame里也检查已经在Queue中的其它点，生命周期是不是已经到了，如果是就将这次的纪录pop out。

[](http://www.unitymanual.com/wp-content/uploads/2013/05/g47D70KAtzy4XeI5nuR8.Q-300x300.jpg)

Unity Motion Trail Script（Unity运动轨迹的脚步）

Pseudo Code如下所示：

接下来的问题就是，将这Queue里的“点”加上建构三角面的Index及计算出贴图UV，进行Rendering，即可得到两点间的移动轨迹，加上贴图就如同上图所示了。Pseudo Code如下所示：

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

|  |  |
| --- | --- |
| 01 | Update() { |
| 02 | **float** now = Time.time; |
| 03 | Vector3 P1 = m\_CurPointA; |
| 04 | Vector3 P2 = m\_CurPointB; |
| 05 | Vector3 preP1 = m\_Queue[0].PointA; |
| 06 | Vector3 preP2 = m\_Queue[0].PointB; |
| 07 |  |
| 08 | **if**( (P1 - preP1).length > 0.5f || |
| 09 | (P2 - preP2).length > 0.5f ) { |
| 10 | m\_Queue.push( p1, p2, now ); |
| 11 | } |
| 12 |  |
| 13 | **while**( m\_Queue.count > 0 && |
| 14 | now - m\_Queue[end].time > 1.0f ) { |
| 15 | m\_Queue.pop( end ); |
| 16 | } |
| 17 |  |
| 18 | *//Generate triangles indices:* |
| 19 | trailMesh.triangles = new **int**[(m\_Queue.count-1)\*2\*3]; |
| 20 | **for**( **int** i=0; i trailMesh.triangles[i \* 6 + 0] = i \* 2; i++ ) { |
| 21 | trailMesh.triangles[i \* 6 + 0] = i \* 2; |
| 22 | trailMesh.triangles[i \* 6 + 1] = i \* 2 + 1; |
| 23 | trailMesh.triangles[i \* 6 + 2] = i \* 2 + 2; |
| 24 |  |
| 25 | trailMesh.triangles[i \* 6 + 3] = i \* 2 + 2; |
| 26 | trailMesh.triangles[i \* 6 + 4] = i \* 2 + 1; |
| 27 | trailMesh.triangles[i \* 6 + 5] = i \* 2 + 3; |
| 28 | } |
| 29 | } |

进阶修饰：

如果只单纯的这样产生Motion Trail，你因该会发现，画出来的弧线会菱菱角角的不是很好看。至于要怎麽加强这个部分呢？很简单，就拿曲线演算来使用吧！在这里使用的是Catmull-Rom Spline，它有几个特点：

1. Catmull-Rom保证，曲线一定通过控制点

2. Spline C1 contunuous，这表示在tangent的方向与长度上，不会有区别。

3. 计算简单

数学表示式如下：

q(t) = 0.5 \*( (2 \* P1) + (-P0 + P2) \* t + (2\*P0 – 5\*P1 + 4\*P2 – P3) \* t2 + (-P0 + 3\*P1- 3\*P2 + P3) \* t3)

Pseudo Code如下所示：

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

|  |  |
| --- | --- |
| 01 | **public** **static** TrailSection Catmull\_Rom( |
| 02 | TrailSection p0, TrailSection p1, |
| 03 | TrailSection p2, TrailSection p3, |
| 04 | **float** t ) |
| 05 | { |
| 06 | TrailSection section = new TrailSection(); |
| 07 |  |
| 08 | **float** t2 = t \* t; |
| 09 | **float** t3 = t2 \* t; |
| 10 | **float** a0 = -t3 + 2\*t2 - t; |
| 11 | **float** a1 = 3\*t3 - 5\*t2 + 2; |
| 12 | **float** a2 = -3\*t3 + 4\*t2 + t; |
| 13 | **float** a3 = t3 - t2; |
| 14 |  |
| 15 | section.pointS = (a0\*p0.pointS + a1\*p1.pointS + a2\*p2.pointS + a3\*p3.pointS) \* 0.5f; |
| 16 | section.pointE = (a0\*p0.pointE + a1\*p1.pointE + a2\*p2.pointE + a3\*p3.pointE) \* 0.5f; |
| 17 | section.time = (a0\*p0.time + a1\*p1.time + a2\*p2.time + a3\*p3.time) \* 0.5f; |
| 18 |  |
| 19 | **return** section; |
| 20 | } |