**Unity3D体感游戏物体下落脚本**

Posted on 2013年07月10日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 26 次

这段代码是下落的位置，下落的游戏物体，还定义了ScoreHit（）得分。主要解决下落问题，此脚本具有通用性。

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

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 | **using** System.Collections.Generic; |
| 03 |  |
| 04 | **public** **class** BallCreator : MonoBehaviour |
| 05 | { |
| 06 | *// the ball prefab. @note MUST be attached* |
| 07 | **public** GameObject prefab; |
| 08 |  |
| 09 |  |
| 10 |  |
| 11 | *// the object we create the balls near @note MUST be attached.* |
| 12 | **public** Transform **where**; |
| 13 |  |
| 14 | *// holds the player mapper (used to find out if we have a target player).* |
| 15 | **public** NIPlayerManager m\_playerManager; |
| 16 |  |
| 17 | *// the time to create the next ball* |
| 18 | *//private float m\_timeToCreateNextBall;* |
| 19 | **public** **float** m\_timeToCreateNextBall; |
| 20 | *// The number of balls the user hit (used for scoring)* |
| 21 | **private** **int** m\_numBallsHit = 0; |
| 22 | *// The number of balls created (used for scoring)* |
| 23 | *// private int m\_numBallsCreated;* |
| 24 | **public** **int** m\_numBallsCreated; |
| 25 | *// Marks that the user has scored a hit (used for scoring).* |
| 26 | *//* |
| 27 |  |
| 28 |  |
| 29 | **public** **void** ScoreHit() |
| 30 | { |
| 31 | m\_numBallsHit++; |
| 32 | } |
| 33 |  |
| 34 | *// mono-behavior initialization* |
| 35 | **public** **void** Start () { |
| 36 | m\_numBallsHit = 0; |
| 37 | m\_numBallsCreated = 0; |
| 38 | m\_timeToCreateNextBall = 0; |
| 39 | **if**(m\_playerManager==**null**) |
| 40 | m\_playerManager = FindObjectOfType(typeof(NIPlayerManager)) **as** NIPlayerManager; |
| 41 | } |
| 42 | *//Unity3D教程手册：www.unitymanual.com* |
| 43 | *// mono-behavior Update is called once per frame* |
| 44 | **public** **void** Update () |
| 45 | { |
| 46 | **if** (Time.time < m\_timeToCreateNextBall) |
| 47 | **return**; *// we created a ball very recently, wait.* |
| 48 | **if** (m\_playerManager == **null**) |
| 49 | **return**; *// this means we don't even have a plyer manager.* |
| 50 | NISelectedPlayer player = m\_playerManager.GetPlayer(0); |
| 51 | **if** (player == **null** || player.Valid == **false** || player.Tracking == **false**) |
| 52 | **return**; *// this means we don't have a calibrated user* |
| 53 | **if** (SkeletonGuiControl.m\_mode == SkeletonGuiControl.SkeletonGUIModes.GUIMode) |
| 54 | **return**; *// we don't throw balls while in GUI mode.* |
| 55 |  |
| 56 | *// now we know we should throw a ball. We first figure out where (a random around the* |
| 57 | *// x axis of the "where" transform and a constant modifier on the y and z).* |
| 58 | Vector3 pos = **where**.position; |
| 59 | pos.x += Random.Range(- 2.0f, 2.0f); |
| 60 | pos.y += 8.0f; |
| 61 | pos.z += 2.1f; |
| 62 |  |
| 63 | *// create the ball* |
| 64 | Instantiate(prefab, pos, Quaternion.identity); |
| 65 | *//Unity3D教程手册：www.unitymanual.com* |
| 66 |  |
| 67 |  |
| 68 | m\_numBallsCreated++; |
| 69 | *// we set the time for the next ball. The time itself depends on how many balls were created* |
| 70 | *// (the more balls, the less time on average).* |
| 71 | **float** maxTime = 5.0f; |
| 72 | **float** minTime = 1.0f; |
| 73 | **if** (m\_numBallsCreated > 15) |
| 74 | maxTime = 4.0f; |
| 75 | **if** (m\_numBallsCreated > 30) |
| 76 | maxTime = 3.0f; |
| 77 | **if** (m\_numBallsCreated > 45) |
| 78 | minTime = 0.5f; |
| 79 | **if** (m\_numBallsCreated > 85) |
| 80 | maxTime = 2.0f; |
| 81 | m\_timeToCreateNextBall = Time.time + Random.Range(minTime,maxTime); |
| 82 | } |
| 83 |  |
| 84 |  |
| 85 | *// mono-behavior OnGUI shows the scoring* |
| 86 |  |
| 87 | **void** OnGUI() |
| 88 | { |
| 89 | **if** (SkeletonGuiControl.m\_mode == SkeletonGuiControl.SkeletonGUIModes.GUIMode) |
| 90 | **return**; *// we don't draw score while in GUI mode.* |
| 91 | GUI.Box(new Rect(Screen.width/2 -100, 10, 200, 20), "You Hit " + m\_numBallsHit + " balls of " + m\_numBallsCreated); |
| 92 |  |
| 93 |  |
| 94 | } |
| 95 |  |
| 96 | } |