**Experiment Report - 85 - test22\_BeatFlyMotion**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 25 | Logic | √ | √ | √ |
| E02 | line 38 | Semantic | √ | √ | × |
| E03 | line 50 | Semantic | √ | × | √ |
| E04 | line 67 | Logic | √ | × | √ |
| E05 | line 95 | Semantic | × | × | × |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 80%

Peer 1 Detection Rate: 40%

Peer 2 Detection Rate: 60%

1. **Source Code**
2. using UnityEngine;
3. //击飞运动效果
4. public class BeatFlyMotion : MonoBehaviour
5. {
6. //当前影响的buff的id
7. public uint mBuffId = 0;
8. //默认高度
9. private float mDefaultHeigt;
10. //向下加速度
11. private float mUpAccerate = 0;
12. //向下加速度
13. private float mDownAccerate = 0;
15. //滞空时间
16. public float mStayTime;
17. //当前高度
18. public float mCurrentHeight;
19. //上次速度
20. private float mLastSpeed;
21. //当前速度
22. private float mCurrentSpeed;
23. //激活运动
24. private bool mEnable = true;
25. //滞空持续时间
26. private float mTotalStatyTime = 0;
27. //运动方向
28. private bool bDir = true;
29. //击飞是否能被覆盖
30. private int mCanBeRecover = 1; //可以被覆盖
31. private Transform mTransform = null;
32. //立即下落
33. public void FallNow()
34. {
35. mStayTime = 0.001f;
36. }
37. void Update()
38. {
39. //击飞运动更新
40. if (mEnable)
41. {
42. //进入滞空状态
43. if (mCurrentSpeed <= 0 && mTotalStatyTime < mStayTime)
44. {
45. mLastSpeed = -mDownSpeed;
46. mCurrentSpeed = mDownSpeed;
47. mTotalStatyTime += Time.deltaTime;
48. bDir = false;
49. return;
50. }
51. //当前速度v = v0 + at
52. //向上运动
53. if (bDir)
54. {
55. mCurrentSpeed = mCurrentSpeed + mUpAccerate \* Time.deltaTime;
56. }
57. //向下运动
58. else
59. {
60. mCurrentSpeed = mCurrentSpeed + mDownAccerate \* Time.deltaTime;
61. }
63. //运动距离s
64. float dist = (mLastSpeed + mCurrentSpeed) \* Time.deltaTime \* 0.5f;
65. //当前高度
66. mCurrentHeight += dist;
67. //保存前一次速度
68. mLastSpeed = mCurrentSpeed;
69. mTransform.position = new Vector3(mTransform.position.x, mCurrentHeight, mTransform.position.z);
70. //回到原来位置或者反向速度为0
71. if (mCurrentHeight < mDefaultHeigt)
72. {
73. Reset();
74. }
76. }
77. }
78. void Reset()
79. {
80. mCurrentHeight = mDefaultHeigt;
81. //重置数据
82. mTotalStatyTime = 0;
83. mEnable = false;
84. bDir = true;
85. mCanBeRecover = 0;
86. mBuffId = 0;
87. mTransform.localPosition = new Vector3(mTransform.localPosition.x, mDefaultHeigt, mTransform.localPosition.z);
88. }
89. }