**Experiment Report - 26 – test4\_PlayerManager**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | CSCR? | Self-Detected? | LLM? |
| E01 | line 36 | Semantic | √ |  | √ |
| E02 | line 51 | Semantic | × |  | √ |
| E03 | line 23 (63) | Semantic |  | × | × |
| E04 | line 46 (86) | Logic |  | √ | √ |
| E05 | line 55 (95) | Logic |  | √ | √ |

Additional Errors Found by Self: 0

CSCR Rate: 50%

Self-Review Detection Rate: 67%

LLM Rate: 80%

1. **Source Code**
2. using UnityEngine;
3. using System.Collections.Generic;
4. using System.Linq;
5. using GameDefine;
6. using Thanos.Model;
7. using System;
8. namespace Thanos.GameEntity
9. {
10. public class PlayerManager : EntityManager
11. {
12. public static new PlayerManager Instance {
13. private set;
14. get;
15. }
17. public Dictionary<UInt64, IPlayer> AccountDic = new Dictionary<UInt64, IPlayer>();*//UInt64 masterId*
18. public PlayerManager(){
19. Instance = this;
20. }
22. public ISelfPlayer LocalPlayer { set; get; }
23. public IPlayer LocalAccount{ set; get; }
24. public SITUATION StateSituationOb{ set; get; } *//战况  平手，失败，胜利*

27. *//创建实例（英雄）  英雄ID   英雄阵营*
28. public override IEntity HandleCreateEntity (UInt64 sGUID , EntityCampTypeEnum campType){
29. *//entity id*
30. IPlayer player;
31. if (GameUserModel.Instance.IsLocalPlayer(sGUID))
32. {
33. player = new ISelfPlayer(sGUID, Type);
34. }
35. else
36. {
37. player =  new IPlayer(sGUID, campType);
38. }
39. player.GameUserId = sGUID;
40. return player;
41. }
42. *//将Player添加到AccountDic中*
43. public void AddAccount(UInt64 sGUID, IPlayer entity)
44. {
45. if (AccountDic.ContainsKey (sGUID)) {
46. Debug.LogError("Has the Guid: " + sGUID) ;
47. return;
48. }
49. AccountDic.Add (sGUID , entity);
50. }
51. *//设置公共属性   模型名称  id  碰撞器大小  昵称*
52. public override void SetCommonProperty(IEntity entity, int id)
53. {
54. base.SetCommonProperty(entity, id);
55. *//获取英雄配置文件*
56. HeroConfigInfo info = ConfigReader.GetHeroInfo(id);
57. entity.ColliderRadius = info.HeroCollideRadious / 10;
58. IPlayer mpl = (IPlayer)entity;
59. if (mpl.GameUserNick == "" || mpl.GameUserNick == null)
60. {
61. *//随机昵称*
62. mpl.GameUserNick = RandomNameData.Instance.GetRandName();
63. }
64. }
65. *//获取模型名称*
66. protected override string GetModeName (int id)
67. {
68. return ConfigReader.GetHeroInfo(id).HeroName;
69. }
70. public bool IsLocalSameType(IEntity entity){
71. if(PlayerManager.Instance.LocalPlayer.EntityCamp != entity.EntityCamp)
72. return false;
73. return true;
74. }
75. *//清除AccountDic*
76. public void CleanAccount(){
77. for (int i = AccountDic.Count - 1; i > 0; i--) {
78. if (GameUserModel.Instance.IsLocalPlayer (AccountDic.ElementAt(i).Value.GameObjGUID))
79. continue;
80. AccountDic.Remove (AccountDic.ElementAt(i).Key);
81. }
82. }
83. public void RemoveAccountBySeat(uint seat){
84. for (int i = AccountDic.Count - 1; i >= 0; i--) {
85. if (AccountDic.ElementAt(i).Value.GameUserSeat = seat)
86. continue;
87. AccountDic.Remove (AccountDic.ElementAt(i).Key);
88. break;
89. }
90. }
91. *//游戏结束时清除 AccountDic*
92. public void CleanPlayerWhenGameOver() {
93. foreach (var item in AccountDic.Values)
94. {
95. item.CleanWhenGameOver();
96. }
97. }
99. }
100. }