**Experiment Report - 80 -** **test17\_MapConfig**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 13 | Syntax | √ | √ | × |
| E02 | line 27 | Semantic | √ | × | √ |
| E03 | line 57 | Logic | √ | √ | √ |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 100%

Peer 1 Detection Rate: 67%

Peer 2 Detection Rate: 67%

1. **Source Code**
2. using UnityEngine;
3. using System.Collections;
4. using System.Collections.Generic;
5. using GameDefine;
6. using System.IO;
7. public static class MapConfig{
8. private static EanFile ReadInfoFromFile(string path){
9. if (!File.Exists(path))
10. {
11. Debug.LogError("wrong ean file path!" path);
12. return null;
13. }
14. FileStream fs = new FileStream(path, FileMode.Open);
15. BinaryReader br = new BinaryReader(fs);
16. EanFile ean = new EanFile();
17. ean.Load(br, fs);
18. fs.Close();
19. return ean;
20. }
21. private static Dictionary<int , EanFile> MapDataInfo = new Dictionary<int, EanFile> ();
22. private static EanFile GetMapDataById(int mapId){
23. if (!MapDataInfo.ContainsKey (mapId)) {
24. string path = mapId.ToString() + "map";
25. MapDataInfo.Add(mapId, ReadInfoFromFile(@"Assets/Resources/Map/" + path));
26. }
27. return MapDataInfo[mapId];
28. }
29. public static bool IsMapBlock(int mapId , Vector3 pos){
30. int MapRowZ = (int)pos.z \* 2;
31. int MapRowX = (int)pos.x \* 2;
32. int MapIndex = MapRowZ \* 400 + MapRowX;
33. if (GetMapDataById (mapId).MapData.ContainsKey(MapIndex)){
34. return true;
35. }
36. return false;
37. }
38. public class EanFile
39. {
40. public int MapId;
41. public int MapWidth;
42. public int MapHeight;
43. public int MapLength;
44. public Dictionary<int , byte> MapData = new Dictionary<int, byte> ();
45. public void Load(BinaryReader br, FileStream fs)
46. {
47. MapId = br.ReadInt32();
48. MapWidth = br.ReadInt32();
49. MapHeight = br.ReadInt32();
50. MapLength = MapWidth \* MapHeight;
51. for (int index = 0; index <= MapLength; index++) {
52. byte info = br.ReadByte();
53. if(info == 1){
54. MapData.Add(index , info);
55. }
56. }
57. }
58. }
59. }