If this game is modified, public static variables like “NoMusic” and “MaxLevel” will need to be explored the most because they are shared between multiple other scripts. If there is any audio related problems it is probably related to “NoMusic”. If the player can’t deal damage, or the scenes don’t load in order then that is related to “MaxLevel”. When files are loaded or saved these two variables are directly changed, and a lot could go wrong from there. In general, Unity will automatically detect and send a little message about errors in the console, but this should not be relied on.

If there is a specific issue is related to an enemy, then that enemy’s script needs checked, or maybe the game object has a problem. Maybe a rigidbody, animator, collider, tag, layer, or public variable in a script is not connected to the enemy. A script is important to how the enemies are controlled, but the source of an issue could begin with the game object. There are multiple times in this game when disabled menus or if the object seems fine then things like the order of nested if statements should be checked. “Debug.Log()” will be the best way to find a problem. Breakpoints cannot be set for these scripts. If there is any change to a public variable or class name in a script, then the game object will no longer be able to use it, so it needs reattached to that script and variable. Buttons will need these connections too, and if there are things that overlap it, the player will not be able to click it. There are many opportunities for modifications to cause errors, but there are some areas that are more likely to be the root of it. Sometimes a bug will only be found after playing for a little bit.