Test Documentation

Here are the problems found in the code and the approaches utilized to solve them.

# 1. Code Conventions

| **Problem** | **Solution** |
| --- | --- |
| Some variables and functions were starting with lower case characters and some were not. | The adopted code convention is:  - attributes/variables: camelCase starting with lower case.  - classes/functions: camelCase starting with upper case.  No special treatment was adopted for dealing with public/private and static variables. |

| **Problem** | **Solution** |
| --- | --- |
| There were some wrong code indentation. It’s worth mentioning that the variable newBox doesn’t even exist in this context, so this would cause a compilation error | The solution was to do the right indentation. |

| **Problem** | **Solution** |
| --- | --- |
| The structures did not have their own files. Instead, they were all together in the same folder. | Their own files were created instead of declaring them inside Types.cs. |

# 2. Optimization/Memory problems

| **Problem** | **Solution** |
| --- | --- |
| Multiples lines of code doing almost the same thing over and over again. | A more generic and concise code was created to replace. |

| **Problem** | **Solution** |
| --- | --- |
| Unnecessary loops that could possibly cause some memory and performance issues. | The approach adopted to solve this problem was exposing the characters positions and so no search needs to be done, and their positions can be retrieved in constant time. |

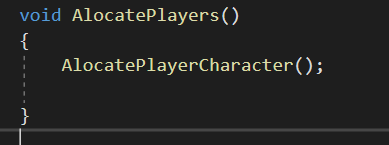
| **Problem** | **Solution** |
| --- | --- |
| Unnecessary and not efficient struct GridBox. | It was created the struct Point to store the player positions, so the struct GridBox was no longer needed. |

| **Problem** | **Solution** |
| --- | --- |
| Multiple functions holding more responsibility than they should, breaking clean code rules. In the image below, it’s clear that the function  CreateEnemyCharacter shouldn’t be the one responsible for calling the StartGame function. | The code was made so that no functions hold more responsibility than they should. |

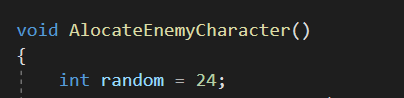
# 

# Other Problems

• There were redundant functions, which only exist to call another function. They were deleted.



• Hardcoded Values. It’s worth mentioning that there are a lot of strings being printed in order to show information about the game. Those strings are all hardcoded. For the sake of simplicity, those messages were kept hardcoded in my version, although I could turn them into variables if further needed.



# Miscellaneous

• The functions were moved out of static void Main, and the game loop logic was rebuilt.

• The class “GameInfo” was added to better organize the information, such as the grid’s size and the characters informations.

• The class “Types” was deleted, because it won’t be used anymore.

