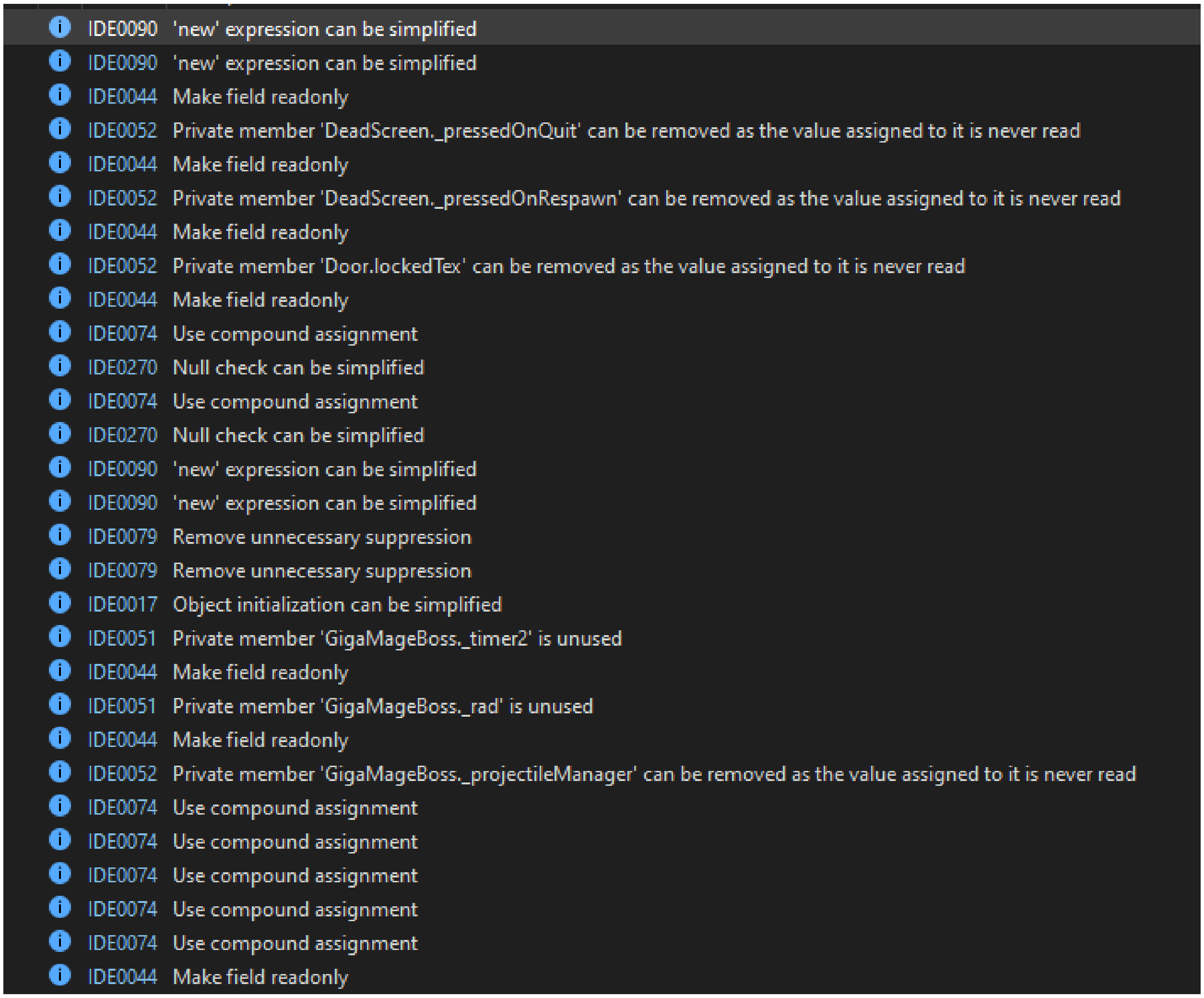
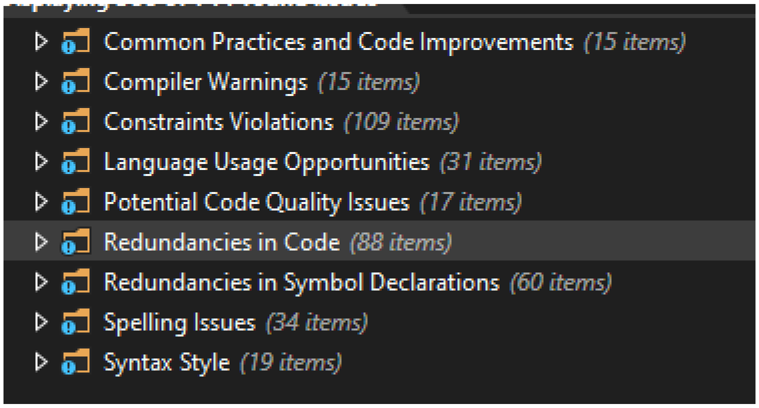
Once again, our group used the .NET built in code analyzer in combination with ReSharper to improve the code quality. Here is what the .NET analyzer gave:



```
IDE0090  'new' expression can be simplified
IDE0090  'new' expression can be simplified
IDE0044  Make field readonly
IDE0052  Private member 'DeadScreen._pressedOnQuit' can be removed as the value assigned to it is never read
IDE0044  Make field readonly
IDE0052  Private member 'DeadScreen._pressedOnRespawn' can be removed as the value assigned to it is never read
IDE0044  Make field readonly
IDE0052  Private member 'Door.lockedTex' can be removed as the value assigned to it is never read
IDE0044  Make field readonly
IDE0074  Use compound assignment
IDE0270  Null check can be simplified
IDE0074  Use compound assignment
IDE0270  Null check can be simplified
IDE0090  'new' expression can be simplified
IDE0090  'new' expression can be simplified
IDE0079  Remove unnecessary suppression
IDE0079  Remove unnecessary suppression
IDE0017  Object initialization can be simplified
IDE0051  Private member 'GigaMageBoss._timer2' is unused
IDE0044  Make field readonly
IDE0051  Private member 'GigaMageBoss._rad' is unused
IDE0044  Make field readonly
IDE0052  Private member 'GigaMageBoss._projectileManager' can be removed as the value assigned to it is never read
IDE0074  Use compound assignment
IDE0074  Use compound assignment
IDE0074  Use compound assignment
IDE0074  Use compound assignment
IDE0074  Use compound assignment
IDE0044  Make field readonly
```

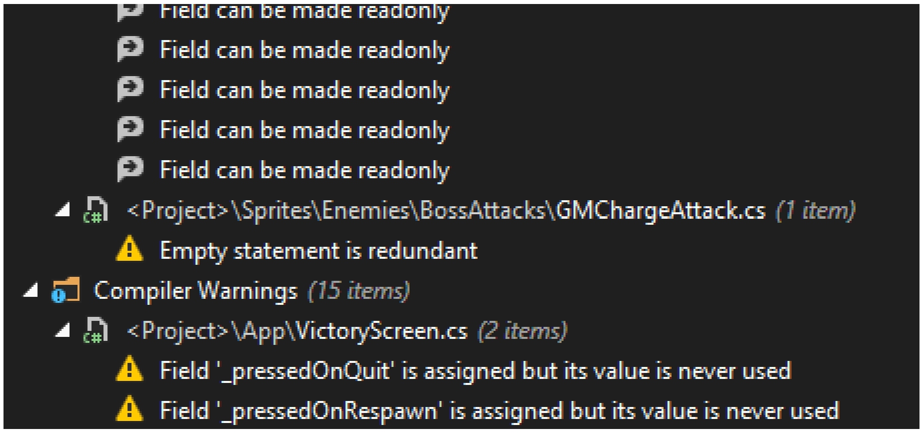
The team did not resolve any of the conflicts that required the ‘new’ expression to be simplified because the change would lead to ambiguity in the code. Similarly, the removal of unnecessary suppression was not changed as that is used to suppress an error from ReSharper. Also, the using statements were not simplified similarly to before, as that would have caused ambiguity in the code, and the team decided to leave explicit statements to better increase readability. All other issues were resolved.

From ReSharper, here are some of the types of problems that were resolved.



- Common Practices and Code Improvements *(15 items)*
- Compiler Warnings *(15 items)*
- Constraints Violations *(109 items)*
- Language Usage Opportunities *(31 items)*
- Potential Code Quality Issues *(17 items)*
- Redundancies in Code *(88 items)*
- Redundancies in Symbol Declarations *(60 items)*
- Spelling Issues *(34 items)*
- Syntax Style *(19 items)*

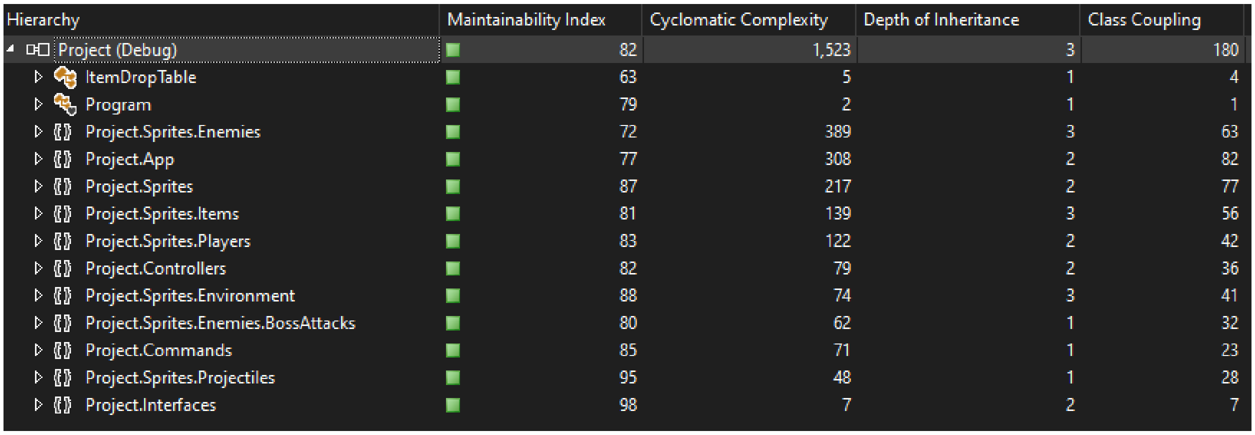
Here are some specific examples of warnings that it rose.



```
        Field can be made readonly
        Field can be made readonly
        Field can be made readonly
        Field can be made readonly
        Field can be made readonly
        Field can be made readonly
    ◢ <Project>\Sprites\Enemies\BossAttacks\GMChargeAttack.cs (1 item)
        ⚠ Empty statement is redundant
◢ Compiler Warnings (15 items)
    ◢ <Project>\App\VictoryScreen.cs (2 items)
        ⚠ Field '_pressedOnQuit' is assigned but its value is never used
        ⚠ Field '_pressedOnRespawn' is assigned but its value is never used
```

The team resolved all of the issues that ReSharper listed unless they would like to ambiguity in the code.

Additionally, this time I was able to find where the code metrics were located. Here are the results.



Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling
Project (Debug)	🟩 82	1,523	3	180
ItemDropTable	🟩 63	5	1	4
Program	🟩 79	2	1	1
Project.Sprites.Enemies	🟩 72	389	3	63
Project.App	🟩 77	308	2	82
Project.Sprites	🟩 87	217	2	77
Project.Sprites.Items	🟩 81	139	3	56
Project.Sprites.Players	🟩 83	122	2	42
Project.Controllers	🟩 82	79	2	36
Project.Sprites.Environment	🟩 88	74	3	41
Project.Sprites.Enemies.BossAttacks	🟩 80	62	1	32
Project.Commands	🟩 85	71	1	23
Project.Sprites.Projectiles	🟩 95	48	1	28
Project.Interfaces	🟩 98	7	2	7