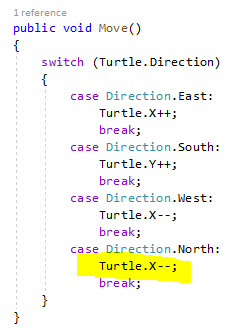
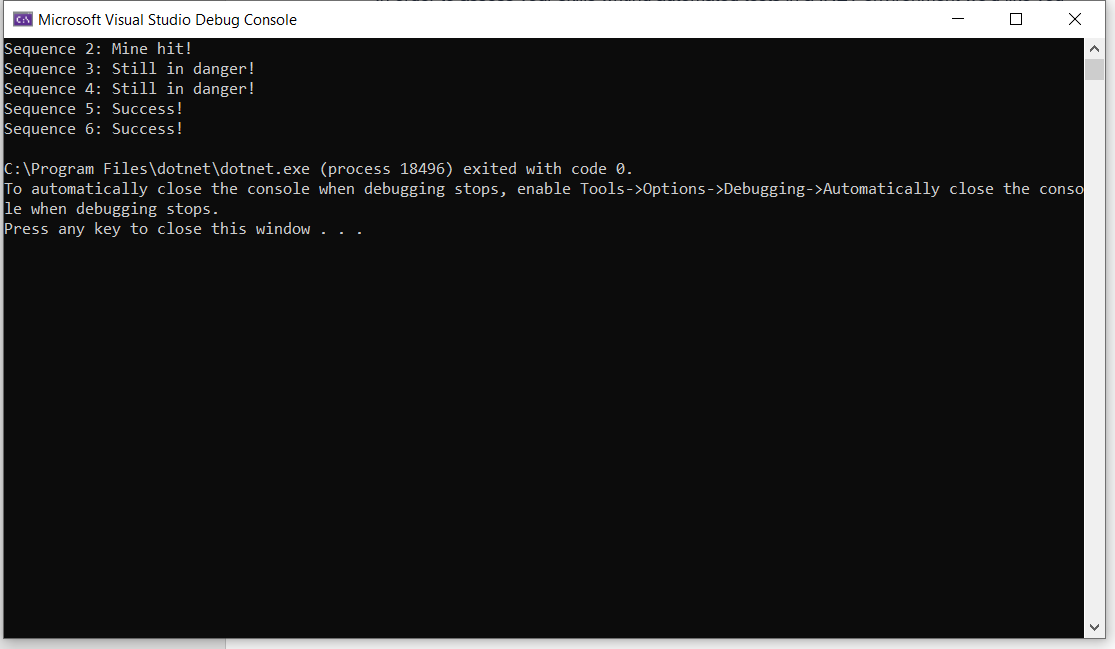
**Test My Turtle Challenge**

**Bugs**

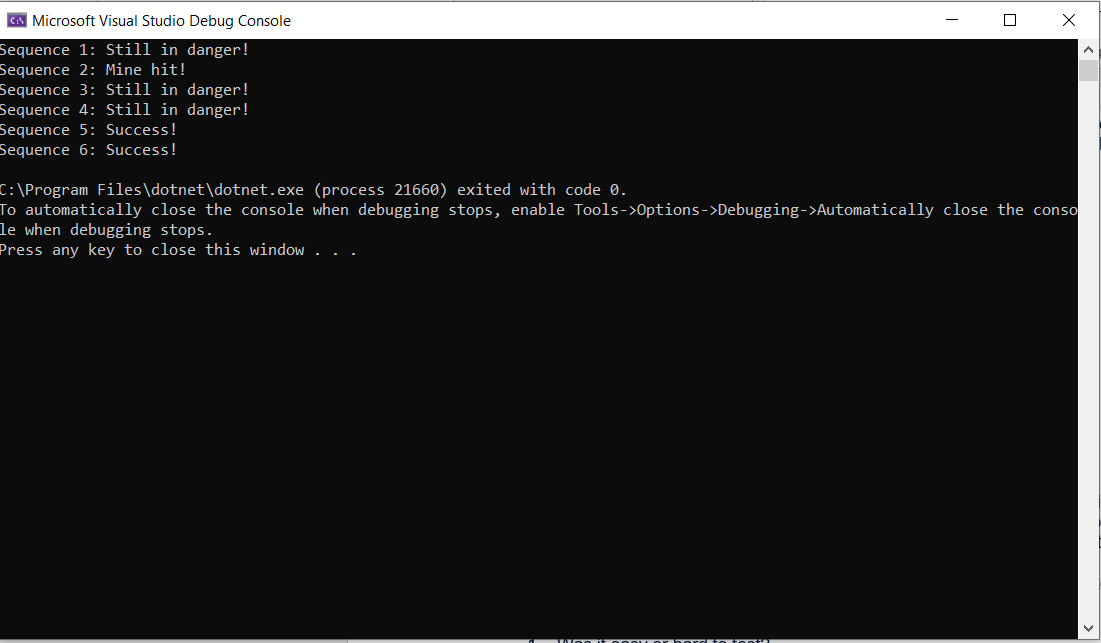
1. In the Move() method, the increment in position for direction North is incorrect. It should be ***Y- -*** not ***X- -***



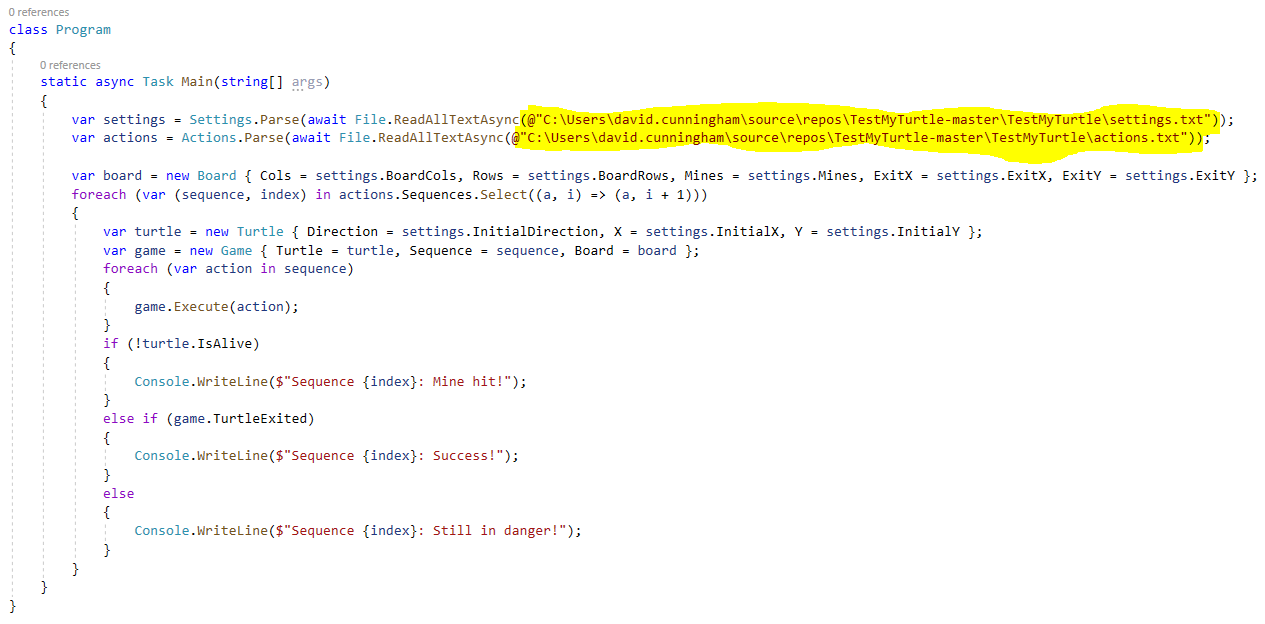
1. The 1st sequence is missing. To fix, remove the ***Skip(1)*** from the end of the main foreach loop



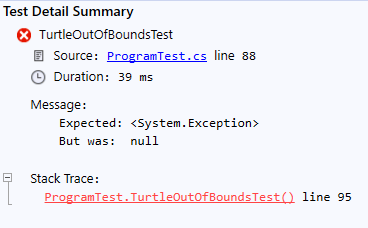


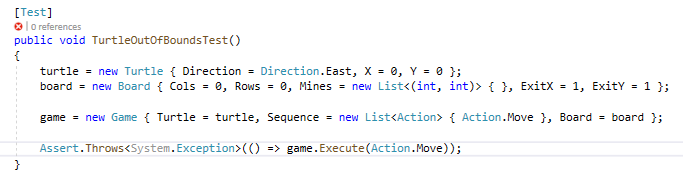


1. When I first ran the code, I got an **IndexOutOfRangeException.** I got around it by passing a file path to the **settings** and **action** files. 



1. It is possible to move the turtle out of the confines of the board, I would expect to see a message saying: “Turtle at the edge!”





* **Was it easy or hard to test?**

It was a little bit tricky to test, mainly because the scope of the challenge is a bit vague. While there were some obvious bugs to find and tests to write, Many other scenarios like whether or not a game has to have mines in it and whether or not the board can be as small as 1 square were difficult to determine as being worthwhile tests. In short there could be loads of other tests.

* **Which changes could be applied to the implementation to make it easier to test?**

A UI or visual display would be the most beneficial change to help with testing the program. It would make it easier to identify and demonstrate bugs.

* **Which tools did you use and why?**

The only tool I used was **NUnit**, as there is no UI. NUnit allows me to have parameterized test cases, so the same test can be executed multiple times with different data covering each combination. This allows me to reuse tests and avoid duplicate tests

* **Which type of tests did you perform and why?**

So, the first tests I wrote were **Unit** **tests** for the Game mechanics (Turtle Move, Rotate, Death and Escape). The purpose of these is to make sure that all core combinations that occur in the game behave as they should. **TurtleMoveTest** identified **Bug 1**

Next, I wrote a **negative test** to try and break the game. The turtle can move outside the grid, this should throw an exception (or message: “Turtle at the edge!”): **TurtleOutOfBoundsTest**

Finally, I wrote a few tests to verify that the settings that get parsed in are not zero, empty or null: **CheckBoardTest** and **CheckTurtleTest**

* **Which additional tests would you perform if you had more time?**

I would add more validation tests to make sure exceptions are thrown. Such as what happens if parsing the actions or the settings file fails. Or what happens if a game is created with no exit coordinates etc.

* **Notes**: I have fixed bugs 1,2 and 3 and I have made settings and actions file paths dynamic (Build Action = Content/Copy always).