[IRESS] Toy Robot Documentation

Created By:	Modified Date:	Notes:
Stephanie Robles	May 25, 2022	I am serious about my application :)

Table of Contents

Pre-requisites

Setting Up

Make sure you have the following installed:

- Visual Studio
- Git Bash

About

Overview

There will be a toy robot placed in the table that can turn either right or left. The toy can move one step forward wherever it is facing.

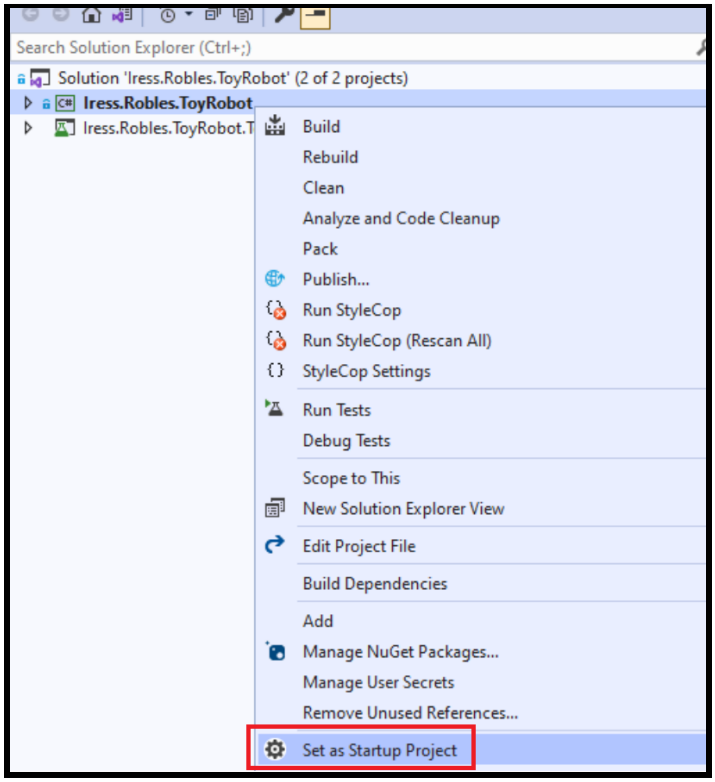
Application Specifications:

- The user can only enter commands:
    - Place
    - Move
    - Left
    - Right
    - Report
- Invalid user commands will be display error message
- User must write command **PLACE** first before being able to do anything
- The **MOVE** command will move the robot one step forward with wherever it is currently facing
- Table should be 5x5
- The **LEFT** command will make the robot turn left, and **RIGHT** command will make the robot turn right.
- The **REPORT** command will display robot coordinates and where it is currently facing
- The robot cannot fall off the table (message will be displayed)
- The starting position of the robot should start **SOUTH WEST** of the table

How to Run The Application

Debugging Code

Simply right click the Iress.Robles.ToyRobot project and set it as a start up project.
You should then be able to run debug the code:

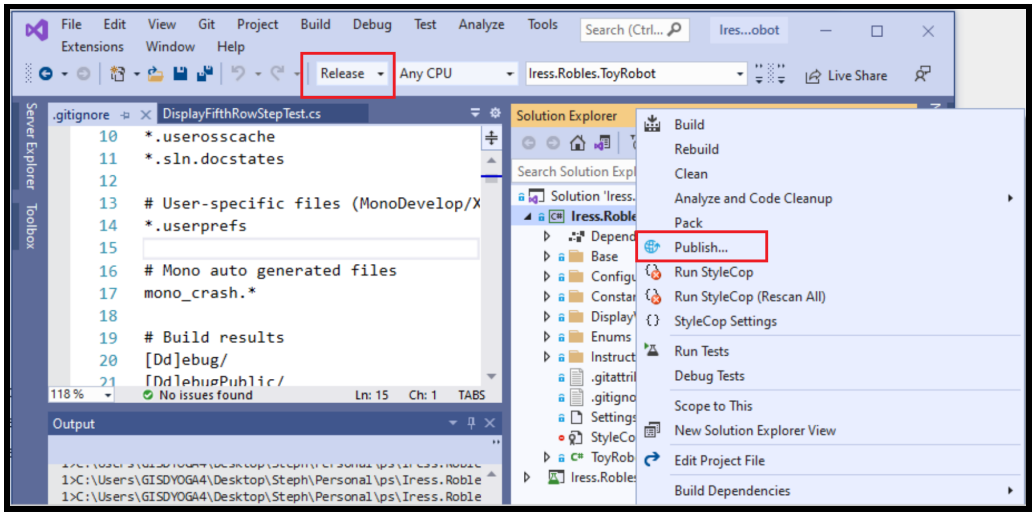


Running Unit Test

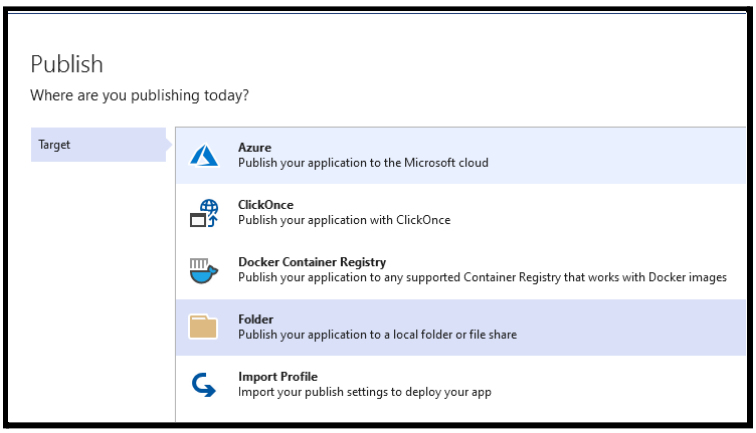
Simply go to the unit test project, place your mouse above one of the class names and click **Run test(s)**.

Deploying the App

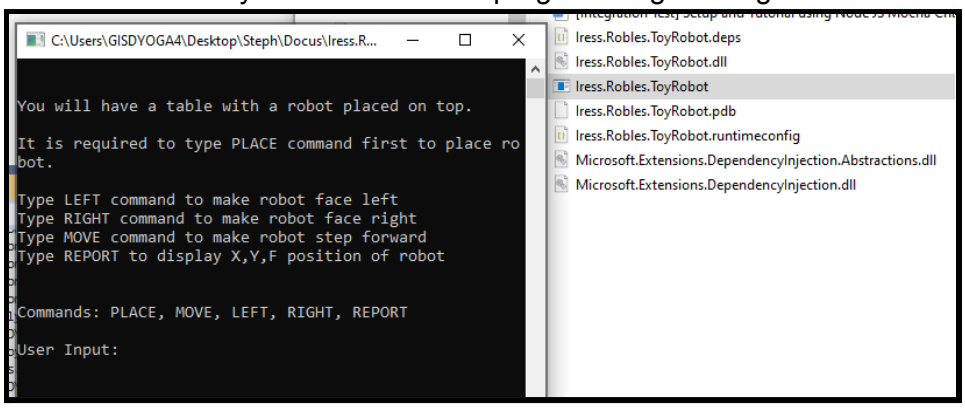
Change the build configuration from Debug to Release. Right click the project then click publish.



Select folder then folder again



After this run the exe file and you should see the programming running:

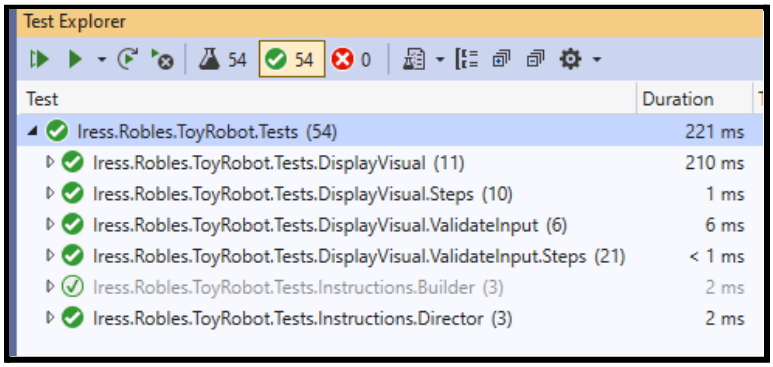


Code Walkthrough

Implementation

It was mentioned that logic was most important and not architecture, but with time I have since Monday after work, I have implemented the following:

- Inversion of Control
- GoF Design Patterns:
    - Builder
    - Chain of Responsibility
- SOLID principle
- OOP
- Stylecop Linter
    - Although did not finish linting due to lack of time
- Unit Tested all testable classes (using Moq)
    - This uses Roy Osherove's best practice for unit test (arrange-assert-act)
        - Youtube Reference: https://www.youtube.com/watch?v=96vXA4GAtD4



- Code metrics of 89: