Robot Simulator

Description

The application simulates a robot moving on a square map of dimensions X units x Y units.

This is a **full .NET console application** written in C# 7.3 for better compatibility and maintainability.

Installation

Run the following command to check that the `dotnet` version is 5.0 or higher:

```
dotnet --version
```

Run this command in the root folder of the repo to build an app.

```
dotnet build
```

The shell will output the path to a built **RobotController.dll** compiled application.

For convenience, prebuilt apps can be found in the PublishedApps folder.

Usage

Go to the published app folder. For example, PublishedApps.

If an app is run without command-line arguments, it expects the user to provide commands to the robot one by one in the console.

```
dotnet RobotController.dll
```

Alternatively, a file with commands can be passed as a command-line argument.

```
dotnet RobotController.dll commands.txt
```

Testing

Unit Testing

Unit testing project RobotTests is written with the use of xUnit

Running unit tests is as simple as

```
dotnet test
```

Tests can be also run in Visual Studio through Test Explorer. Unit tests are only applicable to a full .NET console application **RobotController**.

End-to-end Testing

E2E Testing can be done from the app root folder by running the following commands. It relies on command `test*.txt` files in the E2ETests folder

Tests for **dotnet** with extended output:

```
for t in test*.txt; do  head -1 $t && echo $t && tail -3 $t && dotnet
RobotController.dll $t && echo '\n'; done
```

Tests for **dotnet** with shorter output:

```
for t in test*.txt; do echo $t && cat $t && dotnet RobotController.dll $t
&& echo '\n'; done
```

Running tests one by one:

- `RobotController.dll`

```
set test3.txt && head -1 $1 && tail -3 $1 && dotnet RobotController.dll $1
```

Specification

Full specifications and requirements are not released and cannot be recovered. Please contact the main application developer Max for any questions.