Practical task 1.1

(Pass Task)

Proof of Concept

Submission deadline: 23:59 Sunday, last day of Week 3; **Demonstration deadline**: 23:59 Sunday, last day of Week 5

General Instructions

Your task is to implement a Proof of Concept (**POC**) of the Moon robot simulation. The surface of Moon can be presented as a square grid of the size *N*.

You need to create a console application that will simulate a Moon robot moving on a square map of dimensions *N x N*. There are no other obstacles on the map. The robot is free to wander around the surface of the Moon, but must be prevented from falling off the map. Any movement that would result in the robot falling from the table must be ignored by the robot. However, further valid direction commands must still be permitted. Robot should remain operable while on the map.

Application Specification

Supported Robot Commands

Here is a list of supported direction commands that should be implemented in the POC.

Command	Description
PLACE X,Y,D	the robot on the map at coordinate *(X,Y)* facing the cardinal direction *D* out of 4 possible directions: *NORTH*, *EAST*, *SOUTH*, *WEST*
LEFT	Turn the robot 90 degrees *LEFT* without changing its coordinates.
RIGHT	Turn the robot 90 degrees *RIGHT* without changing its coordinates.
MOVE	Move robot one coordinate in the direction that it's facing.

Constraints

1. The robot **must** ignore all commands until it is dropped on the map. E.g. The robot **must not** move or turn until it is placed on the map.
2. The robot **must not** perform any invalid moves. E.g. fall off the map.
3. The map **must** be not smaller than 2x2 and no larger than 100x100 squares.
4. The map **must** be square. E.g. 25x25, but not 10x50.

Technology Stack Requirements

The application must be cross-platfrom and be able to run from the terminal. Please use the latest version of dotnet to create a new console application and use your IDE of choice to accomplish the task.

Learning goals

While working on the POC get a better understanding of the **domain** of the task and think of the **language** that better describes all the possible interactions within this domain. A domain is the targeted subject area of a computer program. It is a term used in software engineering. Formally it represents the target subject of a specific programming project, whether narrowly or broadly defined (Bjorner, Dines (2006). "Software Engineering 3. Domains, Requirements, and Software Design")

The goal of this task is to quickly prove that the task is achievable and get a hands-on experience in the **domain** of the task.

Next step is the proper design in a different technological stack such as Backend or Server-side. As it needs to be driven by the **domain** of the task, it is beneficial to start with a simple implementation of the POC before diving into the more complex design patterns that will start from the task 1.2P.

Marking Process and Demonstration

To get your task completed, you must finish the following steps strictly on time.

- Please ensure that your program implements all the required functionality, is compilable, and has no runtime errors. Programs causing compilation or runtime errors will not be accepted as a solution. You need to test your program thoroughly before submission. Think about potential errors where your program might fail.
- Submit your program code as an answer to the task via OnTrack submission system.
- Attend an active class to demonstrate your program in one of the dedicated active class sessions. Be on time with respect to the specified demonstration deadline.
- During your demonstration, you need to show your running program and the IDE that you used to complete the task and how you can use the IDE to detect and fix potential errors. Students are allowed to record a video demonstration complemented with their verbal explanation, upload the video to one of accessible resources, and refer to it for the purpose of marking. Students must provide a working link to the video in OnTrack chat box or as part of their submission.
- Please answer any additional questions that the teaching staff might ask during the active class or in OnTrack. Answering questions is part of the assessment procedure and is critical for your understanding of the unit material.