Toy Robot Program

Coding Challenge

Project Summary and Resources

Author: Merco C'zar Salise

Definition of Done - Core Requirements:

- Working application
- Link to hosted VCS repository (preferred), or zip of source code
- Appropriate documentation such as a readme (it should be clear how-to setup and run the C++ app)
- Appropriate unit and/or integration tests included with the source code (these should be runnable)

Definition of working application:

- User can:
  - PLACE, MOVE, (rotate) LEFT, (rotate) RIGHT a Robot that is placed on a Tabletop
- The Robot:
  - Fill not fall off the tabletop
- Application can:
  - Import a file of moves, a moveset, for the Robot
  - Read user's inputs of moves, the moveset, for the Robot

Repository Location:

https://github.com/MurkTsar/toy_robot_code_challenge

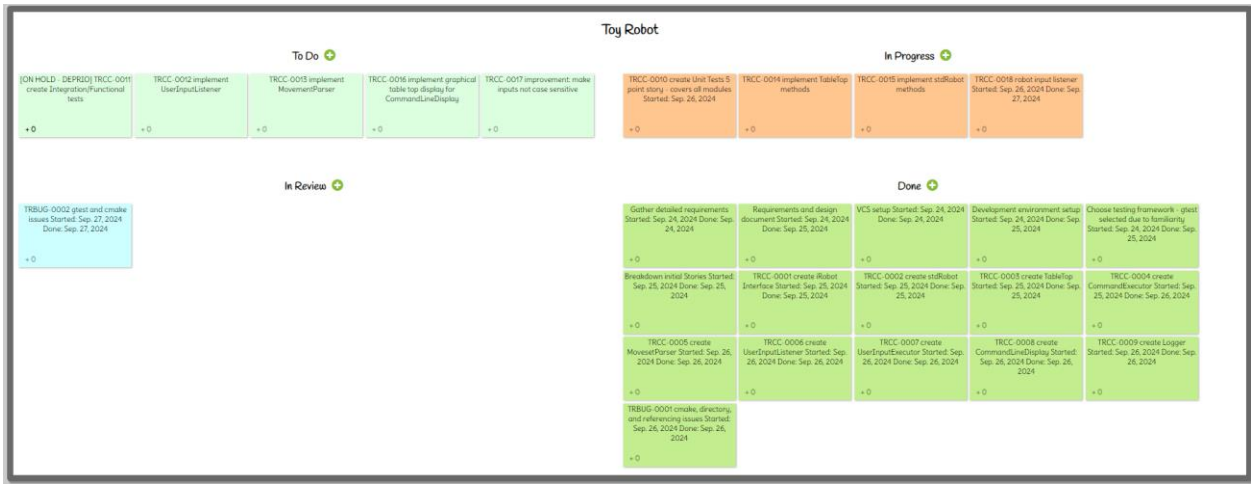
Stories and Items:

Toy Robot Code Challenge.xlsx

Task ID	Status	Title	Description/Notes		Task ID	Status	Title	Description/Notes
TRCC-0001	DONE	create iRobot Interface	skeletal form as of the moment		TRISSUE-0001	DONE	main pc damaged, setup backup laptop	
TRCC-0002	DONE	create stdRobot						
TRCC-0003	DONE	create TableTop						
TRCC-0004	DONE	create MovementInterface						
TRCC-0005	DONE	create MovesetParser						
TRCC-0006	DONE	create UserInputListener						
TRCC-0007	DONE	create UserInputExecutor						
TRCC-0008	DONE	create CommandLineDisplay						
TRCC-0009	DONE	create Logger						
TRCC-0010		create Unit Tests	main program not necessarily covered					
TRCC-0011		[ON HOLD - DEPRIO] create Integration/Functional tests	initial set of functional tests		TRBUG-0001	DONE	cmake, directory, and referencing issues	
TRCC-0012		implement UserInputListener	actual coding implementation		TRBUG-0002	DONE	gtest and cmake issues	
TRCC-0013		implement MovementParser						
TRCC-0014		implement TableTop methods						
TRCC-0015		implement stdRobot methods						
TRCC-0016		implement graphical table top display for CommandLineDisplay						
TRCC-0017		improvement: make inputs not case sensitive						
TRCC-0018	DONE	create robot input listener						
TRCC-0019		incapsulate project in a namespace						
TRCC-0020		ToyRobotProgram	The class that run() the show					
TRCC-0021		implement robot input listener						
TRCC-0022		QA: command line input feature						
TRCC-0023		QA: Upload file feature						
TRCC-0024		improve unit tests						
TRCC-0025		improve integration/functional tests						
TRCC-0026		code optimization						
TRCC-0027		code clean up						
TRCC-0028		Update Read Me						
TRCC-0029		Upload Documents in PDF						
TRCC-0030		create demo						
	33%					100%		

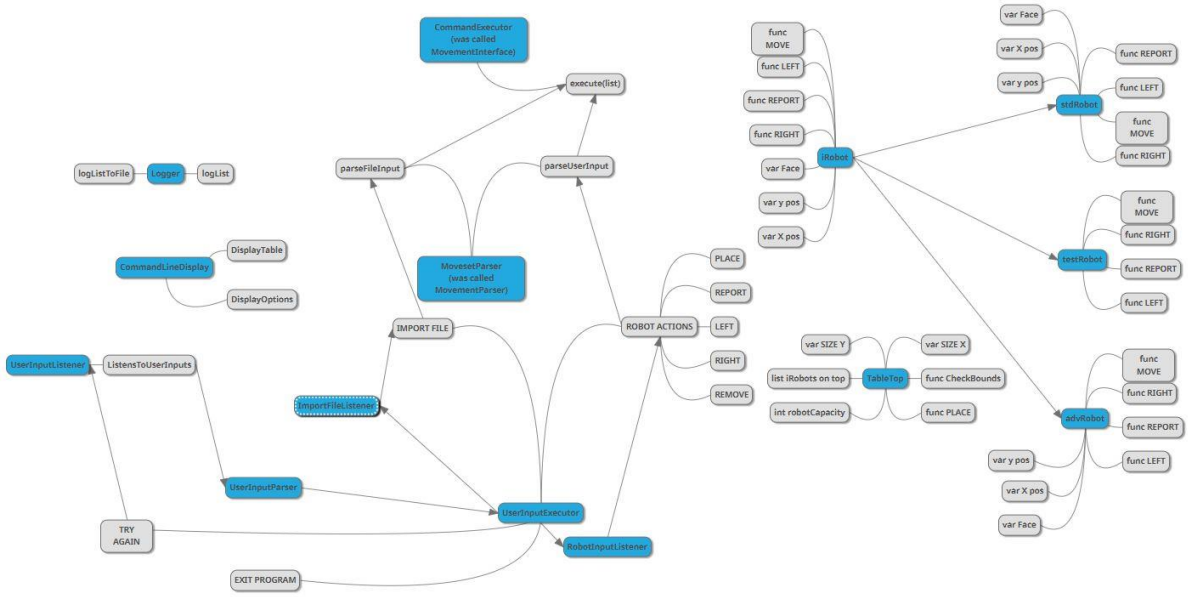
Sprint Tracker:

https://ideaboardz.com/for/Toy%20Robot/5387512



Idea Design:

https://app.mindmup.com/map/new/1727159217294



Idea Design version 4

Design Specifications:

Uploaded in the doc/ directory

Folder Structure:

- src/: This folder contains the C++ source files. For now, you have main.cpp.
- include/: This folder will hold header files (.h or .hpp) that declare functions, classes, etc., that you will define in src/.
- lib/: This is where you can place any external libraries, like GoogleTest, or the custom libraries.
- doc/: This will store the project documentation (e.g., README, design documents, etc.).
- tests/: This is for the unit tests and functional tests:
    - unit_tests/: Focus on individual components (e.g., classes or functions).
    - functional_tests/: Test complete features or system behaviors (e.g., the robot moving to the edge of the table).
- .gitignore: A Git file to ignore specific files like binaries or temporary build files.
- README.md: Documentation about the project, how to set it up, build, and run.