**Robot Test Utility**



**Scenario**

Our service department need a simple utility to test our robotics heads before they are shipped out to customers. The goals for the utility are to verify the movement quality and reliability of the robots to execute moves and reliably move to the pre-defined targets.

They will mount several robotic heads around the room, and test out targeting on several fixed targets.

The utility will allow the user to select a target location containing pan/tilt axis data and move the robotic heads to these positions by either pressing on “GO” button associated with the corresponding device or moving all the devices to the position using the “Move All” button.

A news studio with a desk and chairs

Description automatically generated

The time taken for a device to reach this location will depend on the distance the robots are required to move, it is likely that if all the devices are asked to move at the same time, they will reach the target at different times.

While the robots are moving, they will return a time-to-shot value (TimeSpan) indicating the estimated time before the devices reaches the target, this should update on the UI in the form of a progress bar while the robot is moving and the UI should remain responsive, allowing the user to select another robot to move.

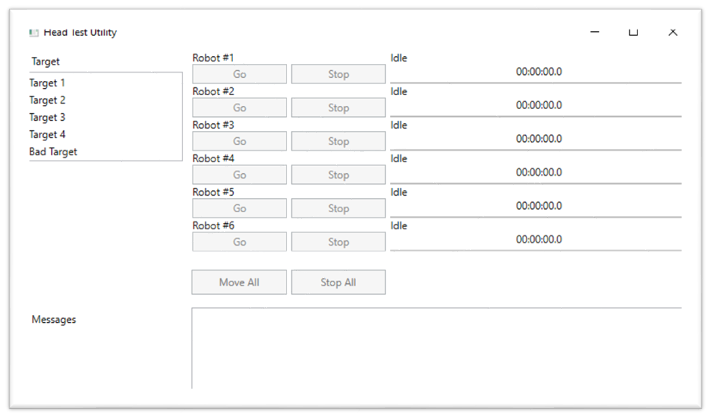
The robots will also return status information indicating whether they are moving or idle. This status information on the UI for each robot, respectively.

Whilst the robot(s) are moving it should **not** be possible to trigger the same target move a second time. But the user can select a different target and trigger the new move.

While the robot(s) are moving the user is given the option to stop the move by clicking the “Stop” button for an individual robot or by clicking on the “Stop All”.

When moving, if an error occurs on a robot, the error will be shown in the user interface to the user.

The UI looks like this:



The solution file has everything needed to run the application as well as a project demonstrating a single automated test written in MsTest and using WinAppDriver/Appium to remotely control the application.

Some modification may be needed to the code, e.g. file paths and possibly Appium/WinAppDriver Urls

As part of the development team responsible for delivering this utility within a sprint, you will be responsible for producing a test plan, writing manual as well as automated tests and identifying any additional tests needed.

Product Criteria

* Accurate Time: The user must be able to verify that the robot moves precisely according to the specified time.
* Usability: The user interface should be intuitive and easy to use.
* Movement quality: Recalling a shot from the utility should move the robot smoothly with no motion quality issues, this can be verified by viewing the live video feed from the camera while using the utility.

The main objective is to see the approach the interviewee takes to testing and a stretch goal to see some examples of automated and manual tests that would be written. Automated tests can be written as BDD tests, pseudo code or as classic MsTest/NUnit tests.