Criterion D: Transcript of Video

The following application is made for middle school students that would have a robotic arm called the uArm. The robotic arm would be controlled using this application. They would have 3 buttons to choose from, to practice, complete exercises with the arm, or learn about the overall information of the application. The user then can connect to the arm by choosing the port that the arm is connected to, if there isn't a connection, there would be an error telling the user that they don't have the arm connected.

However, let connect it to establish a connection between the user and the arm. Then the next window would be the window that would send information to the arm. There are six different directions (left, right, front, back, up, down) we can make the robotic arm move in. The distance that the arm would be moving in would be in millimeters, more specifically, 60mm per minute so that it is safe to be used by middle school students. Additionally, if you click the edit button, you will be able to change the distance and the direction of the step in the list. Then you can reset the position of the arm, making it go to its start-up position if you click the reset button.

Now, let's go and try to complete the exercises by choosing the difficulty (easy). There will be exercises at the top of the GUI and the list of steps is the same as what we have previously done. So, now let's try and complete the exercise.

We can now test the limits that I have set in the application, so that arm can read all the information that is sent to the arm. As you will hear, there will be a beep when the arm hits the limit of the arm that has been previously set. Now, testing all the limits. Additionally, all the limits are put into play as it wouldn't make any unnecessary movement like pushing the ground, which it would normally do.

Nonetheless, by looking at all the exercises by clicking next, we will see the "No Exercise Available" for the exercise. So, now let's look at another level with an exercise that we can complete. The following exercise would utilize the pump to grab things, as this is what the middle schoolers would be doing in their class if they will use the pump built into the arm. However, I have sped up the video, so that it doesn't become redundant. When done, let's look at all the exercises present in the application.

Since we don't have any more exercises, let's try and add more and look at all the exercises in this application. These exercises can be imported or exported. As you can see, I have added a new exercise and it didn't go to the end of the list (Array List) but, in the section where the level of exercise is in (easy). Moreover, you can edit the exercises and their level, which would also automatically transfer/sort them to their corresponding level/section using a queue. These exercises would all appear when completing the exercises.

Now, let's see how we can save and open the file of steps and/or exercises. So, by clicking the Save As button we will be able to save the list anywhere on our computer. Where if we accidentally delete something from the list and run it, thereby saving the steps, we would lose it. If we open the file that we have saved, then we will restore the file that we saved.

Let's say that one accidentally deletes the file that the list is stored in. So, if we look at the directory where all the files are stored, we can see that there is a log. This log would store all the steps that the arm has executed. This can

be sent to the teacher to show the process of the student. So, if the life is deleted, we could just reopen the window and then the file would be restored, but it would be an empty file. So, we can open the file that we save, and then the only way to save it into the directory would be to run the list.

Now let's see if we can email all of our data to someone else, so as we send it and go to the receiver. We can see that all the files in the directory were sent and that the receiver or teacher could use them in their application.