**Romi Exercises**

**Exercise 1: Getting Started**

**Objective:** To establish the full pipeline controlling a newly constructed Romi using the WPILIB VS Code Simulator GUI.

**Description:** Follow the instructions at <https://docs.wpilib.org/en/stable/docs/romi-robot/index.html>. Leverage the “RomiReference” example and show a video of your robot moving.

**Exercise 2: Teleop Programming**

**Objective:** To learn some of the code and where to modify to create new behaviors.

**Description:** Add a command which leverages a joystick button to toggle one of the LEDs on the Romi. Show a video of you driving your robot and flashing the LED. (Note: if you start from the Romi Example in the first exercise you will only need minor changes. If you’d like to explore more, you can start with the Romi template instead).

**Exercise 3: Teleop Programming**

**Objective:** To learn some of the code and where to modify to create new choices from the user interface.

**Description:** Building off of Exercise 2, leveraging a SendableChooser, upgrade your robot so that you can select your driving mode from the SmartDashboard during teleop between arcade and tank drive. Show a video of you driving your robot in different modes.

**Exercise 4: Autonomous Programming**

**Objective:** Expand on previous programming knowledge to move the robot on a path autonomously.

**Description:** Create a new robot program using the command-based programming framework which can autonomously perform the following tasks in sequence:

1. Drive 2 feet forward
2. Turn around
3. Drive 2 feet back

Goal is to arrive as close as you can to your starting position. There are several approaches to this ranging from directly leveraging encoder distance, using the gyro, up to using advanced controls like PID or trajectory planning (if you’d like to learn more about that see <https://docs.wpilib.org/en/stable/docs/software/examples-tutorials/trajectory-tutorial/trajectory-tutorial-overview.html> and <https://github.com/bb-frc-workshops/romi-examples/tree/main/romi-characterization>). As before, we’d love to see a video!

**Exercise 5: Navigating a Maze**

**Objective:** To take your knowledge and apply to a real competition challenge.

**Description:** Successfully navigate the maze by any means possible found here: <https://www.youtube.com/watch?v=AHXyGjqYJbU> (Recommendation is to just use tape on your floor for the gameboard). Goal is to navigate the maze in the fastest time possible without crossing any lines. If possible show a video and add tell us your time.

**Potential Modifications:**

* Pre-work
  + Can we make it simple to connect in bridge mode?
    - Setup & verification
    - IP address discovery
  + Ensure fully ready (a verification check)
  + Latest version to remove bugs?
  + Must have back up batteries and ensuring they are charged beforehand
  + Must have usb cable for imaging Romi
  + Must have an ethernet cable to debug?
  + Add file to SD Card if there, always boot in access point mode
* Basic concepts
  + Teaching of existing program (subsystems, commands, button mapping, basic wpilib DifferentialDrive, Joystick, Button, etc), build.gradle and IP determination, etc)
  + Walk through the program from a mentor?
  + Teach them Simulator and what it is and how to navigate it
  + What about teaching Java/programming concepts first?
  + What about command-based programming?
  + Can these tutorials be a recording?
* Exercises
  + Have a program that is “broken” that the students would need to debug, diagnose, and modify to fix it
  + Have a repo for starting programs and/or snippets? What about sample solutions provided as well? What about videos of it to visualize?
  + Clean up exercises. Need to have more context (for example what is tank drive and how is it used)
  + Break up exercises into smaller parts (1a, 1b, 1c) for incremental milestones
  + Leverage simple robot programs for pre-cursor exercises like this:  
    <https://github.com/wpilibsuite/allwpilib/blob/main/wpilibjExamples/src/main/java/edu/wpi/first/wpilibj/templates/educational/Robot.java>
* Future exercises
  + For gyro and encoders exposing the reasons to use each
  + For sys id
  + For PID controllers
  + Optional additional purchases and exercises? Arm, Camera, bumpers, servo, distance sensors?
* Logistical changes
  + Must have students there on-time!
  + Breakout room pairing helped focus attention
  + Mentor to mentee relationship 2:1 is great, 3:1 not so good (too many different advice viewpoints)
  + Better side-channel for mentor support that is checked frequently (especially by Joe and ZQ)
  + Duration (is 1 hour enough?)
  + Number of meetings (is 3 enough?)
  + Awards/Recognition?
* Changes to romireference
  + Better naming conventions (DriveDistance -> DriveSomeDistance)
  + Better comments around function parameters