Lesson Planning Template **Teacher: DeSIRE**

📆Date:	Topic: APP Custom Control	Time Required: 120 minutes

🎯**Learning Target/Objectives:**

- **I can** integrate previously programmed "Action Groups" into a mobile application interface to achieve a specific project outcome.
- **I can** define and assign custom parameters (names and index numbers) within a mobile app to trigger robotic hardware movements.
- **I can** manage and troubleshoot the wireless connection and file synchronization between a PC, a mobile device, and a robot controller.

📜**Vocabulary:**

- **Project Outcome**
- **Custom Function**
- **Confirmation**
- **Interface**
- **Action Group Number**
- **Dialog Box**

❓**Guiding Questions:**

- Why is it essential to confirm that the Action Group was successfully downloaded to the robot controller via the PC before trying to use it on the mobile app?
- How does using a "Custom Action Group" button improve the usability of a robot for a person who didn't program it?
- What is the difference between a "User-Friendly Name" (like "Grip Object") and a "System Index Number" (like 20), and why does the robot need both?

📚**Lesson Design Details:**

- **Activity 1: The App Developer's Checklist**
  - **Focus:** Students create a "User Manual" page for their custom app. They must list the specific steps required to add a button and explain why the Action Group Number is the most critical field.
- **Activity 2: Precision Button Mapping**
  - **Focus:** Students program three different buttons for three different tasks (e.g., Home, Grip, Release). They must verify that each button calls the correct index number from the robot's memory.
- **Activity 3: Mobile UX Audit**
  - **Focus:** Students test each other's custom buttons. They must evaluate if the names chosen (arbitrary names) are clear enough for a new user to understand what the robot will do.

Key Points (Vocabulary):

- **Project Outcome:** The specific goal or final result achieved by completing the lesson steps.
- **Custom Function:** A user-defined feature in an app that performs a specific task, such as triggering a saved action.
- **Confirmation:** The process of verifying that prerequisites, such as Bluetooth and GPS, are active before beginning work.
- **Interface:** The visual layout on the mobile screen where the user interacts with the robot.
- **Action Group Number:** The specific numerical index (0-230) that identifies a stored movement file in the robot's memory.
- **Dialog Box:** A pop-up window in the app used to input or modify data.

Key Points of Instruction

- **The Prerequisite Chain:** Students must understand that this lesson is the third "link" in a chain. Link 1 is the physical robot, Link 2 is the PC-based programming (Lesson 2), and Link 3 is the mobile interface. If any link is missing, the system fails.
- **Index Precision:** Emphasize that while the name "Grip Object" is for the human, the number "20" is for the machine. Entering the wrong number will result in either no movement or the wrong movement.
- **System Settings:** Remind students that many modern apps require "GPS/Location" services to be active to enable the Bluetooth search, a common point of frustration for young learners.
- **Interface Management:** Teach students how to "clean up" their interface by sliding or long-pressing to delete old actions, mirroring standard mobile UX (User Experience) behaviors.

Teacher's Cheat Sheet

Feature	Critical Requirement
Connection Pre-Check	Bluetooth ON and GPS ON
Required App	"Wonderbot" (Android or iOS)29.
Index Number Range	Must be between 0 and 23030.
Critical File	Action Group No. 20 (from Lesson 2)

Button Execution	Click the name in the "Custom Action Group" list to run once
Modification	Slide or Long-Press the name to Delete

Category	Standard Organization	Standard/Benchmark Code and Description
Computer Science	NCSOS	**HS-CS-02:** Design and implement strategies for troubleshooting hardware and software problems.
Technology	ITEEA	**STEL-2R:** Follow step-by-step instructions to safely use systems and troubleshoot common problems 2
Engineering	ITEEA	**STEL-7Q:** Apply a broad range of making skills to follow a design process in the construction of a prototype.
Digital Literacy	ISTE	**1.6.b:** Creative Communicator - Students create original works or responsibly repurpose or remix digital resources into new creations.
Computer Science	NCSOS	**HS-AP-10:** Create procedures with parameters to organize code and make it easier to reuse.