**AP CSP CodeBot**

|  |  |  |
| --- | --- | --- |
| **MISSION 7 Obj 6-9 Hot Pursuit** | | **Time: 45 minutes** |
| **Project Goal:** Students will use proximity sensors to program the ‘bot to track an object.  **Learning Targets**   * I can write calibration functions so the ‘bot can adapt to its environment. * I can apply previous knowledge of the motors to rotate and face an object moving in front. | **Key Concepts**   * The prox.detect(power, thresh) function lets you adapt to different environments. * Using auto calibration functions for power and thresh allows the ‘bot to adapt to a new environment. | |
| **Assessment Opportunities**   * Mission 7 Obj 6-9 Assignment * Submit the “HotPursuit” program (part 2) | **Success Criteria**   * Write a calibration function for power * Use motors to follow a detected object. | |
| **AP CSP Framework**  **DAT-2.E** Explain how programs can be used to gain insight and knowledge from data.  **AAP-1.A** Represent a value with a variable.  **AAP-2.F** Write expressions using logical operators and evaluate those expressions.  **AAP-3.C** Develop procedural abstractions to manage complexity in a program by writing procedures.  **Computational Practice 2.B** Implement and apply an algorithm.  **Computational Practice 4.C** Identify and correct errors in algorithms and programs, including error discovery through testing. | **Materials**   * An object to detect * Mission 7 Obj 6-9 Assignment / Answers * Solution code for HotPursuit\_obj9 | |
| **Teacher Notes**   * Objective 6: Students only need to read the instructions in CodeSpace. They can open their HotPursuit program from the last lesson. It already has a function for cal\_thresh(), and all goals are already met. Run the code to continue. * Objective 7: Students add a function for cal\_power(), which calls cal\_thresh(). Follow CodeTrek. Then run the code using different surfaces for CodeBot and record the results. * Objective 8: Students add code, following CodeTrek, which doesn’t give a lot of help. The goal shows the code that is expected, with students filling in the motor speeds to turn left or right, or go straight. Students should do plenty of testing, and may need to adjust their speeds to really see what is happening and make sure the turns are correct. I suggest a speed that is fairly slow. * Objective 9: This objective introduces another Boolean variable to use as a toggle for turning on/off the motors. Follow CodeTrek, except move the new if statement just after the first if statement for button 1. The code ends up there after the last objective, so might as well put it there now. | | |