

CodeX Remix Task

MISSION 11: Spirit Level	PROJECT: Spirit Level v.2	# HOURS: 1
PROJECT GOALS: Students will improve their level program to be more sensitive and accurate.	EXTRA PROJECT MATERIALS: <ul style="list-style-type: none"> • none 	VOCABULARY: <ul style="list-style-type: none"> • Comment • Accelerometer • Pixel
LEARNING TARGETS: <ul style="list-style-type: none"> • I can use comments to explain and document the purpose of each line of code. • I can use variables to calculate and convert measurements. 		
SUCCESS CRITERIA: <ul style="list-style-type: none"> <input type="checkbox"/> Display a special symbol when level is at the zero mark. <input type="checkbox"/> Adjust the sensitivity so you can measure precise levels near 0°. 		

RUBRIC:

CSTA Standard	Basic (3)	Proficient (4)	Mastered (5)
Documentation			
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.	Incomplete flowcharts.	Flowcharts provided for each process.	Flowcharts provided for each process. Evidence of revisions and improvements made.
Algorithms and Programming			
2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.	No variables; variables not named appropriately.	Variables used and named correctly in most instances.	Variables are used and named correctly in each process as needed.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	No loops or conditionals.	Loops or conditionals used correctly in most instances.	Loops or conditionals are used correctly in each process as needed.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Code is not organized or readable. No comments.	Code is sometimes organized into problems and subproblems in order to make it organized and readable. Comments used inconsistently.	Code is decomposed into problems and subproblems, making it easy to follow and read. Comments are clear and easy to understand.
Debugging			
2-CS-03 Systematically identify and fix problems with computing devices and their components.	Code bugs not identified; little or no documentation of fixes.	Code bugs mostly identified and fixed; adequate documentation of fixes.	Code bugs identified and fixed; extensive documentation of fixes.