




Final Project Planning	Name:	
Other team members:		
Remix Step 1: Review your code from the mission pack		
What programs / missions were your favorite? What did you like about them?		
What programming concepts do you feel you understand the most?		
What programming concepts do you need help with?		
Remix Step 2: Final Project Concept		
Look over the remix suggestions from your favorite projects. Discuss with your team. Then decide what you want to do for the final project that will both interest and challenge you. Describe what your final project will do:		
Remix Step 3: Plan your code. What variables will you use in the project? Fill out the charts below. Use another piece of paper to design your program with a flowchart or pseudocode.		
What variables and lists will you use in the project? Fill in the chart. You do not need to fill in every line, or you can add more.		
	Variable / List Name	What it will be used for:



What functions will you write? Describe each one.	Function name		What it will do

What buttons will you use, and what will happen when pressed?	Button		What will happen:

Remix Step 4: Write your code

Use the sandbox  when you write the code. Write just a few lines at a time and test often.

Remix Step 5: Commenting and feedback

Documentation	<ul style="list-style-type: none"> • Make sure your code is readable by adding blank lines • Add comments to explain sections of code
---------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

Peer feedback: Get feedback from two (or more) people.

Peer Review #1 Name:	
Go through the rubric. Are all requirements met? If not, list any missing criteria.	
What do you like about the program – be specific!	
Give at least one suggestion. Begin with “what if” or “maybe you could”	



Peer Review #2 Name:	
Go through the rubric. Are all requirements met? If not, list any missing criteria	
What do you like about the program – be specific!	
Give at least one suggestion. Begin with “what if” or “maybe you could”	

Review the comments. Then take time to improve or add to your project.

Post-Mission Reflection

What did you change in your project after reading the feedback?	
What do you like most about programming?	
What do you find the most challenging about programming?	
How have your attitudes or feelings about computer science changed during this mission pack?	



Final Project Rubric			
Requirement	No evidence ←-----→ Mastery		
Programming Conventions are followed	<ul style="list-style-type: none"> Variable names aren't descriptive Function names aren't descriptive Code blocks inconsistently indented Capital letters used Code is not organized into sections 		<ul style="list-style-type: none"> Variable names are descriptive Function names are descriptive Code blocks consistently indented Use of small letters (not capital) Code is organized into sections
Documentation and Readability	<ul style="list-style-type: none"> No comments are used. Code is difficult to read because no blank lines were used, or too many blank lines were included. 		<ul style="list-style-type: none"> Frequent and descriptive comments are used regularly. Blank lines are used to help with readability.
Use of Variables and constants	<ul style="list-style-type: none"> "Magic Numbers" or literal values are used in the code. Data isn't tracked or updated (no counters, states, conversions, etc.). 		<ul style="list-style-type: none"> Constants are used to eliminate "magic numbers." Variables are used for storing, keeping track of and updating data. Global and local variables are used.
Use of Functions	<ul style="list-style-type: none"> No plan or algorithm to follow. Everything in one main program. Long sections of code. Functions use all global or all local variables. Functions don't take parameters. 		<ul style="list-style-type: none"> Code is divided into smaller sections that accomplish a task. Parameters are used as needed. Local and global variables are used as needed. Functions return a value as needed.
Use of Inputs Buttons and sensors	<ul style="list-style-type: none"> Neither button is used for input. No sensors are read or used. (line sensor, proximity sensor, encoders, system temperature, battery voltage, accelerometer) 		<ul style="list-style-type: none"> At least one button is used for input and control. At least one sensor is used to give input. Conversion of raw data is performed as needed.
Algorithms and Programming	<ul style="list-style-type: none"> No algorithms identified or used. Program performs the same for every execution, without input. Lists and tuples are not utilized when they would simplify the code. Debugging practices are not used and code contains errors. 		<ul style="list-style-type: none"> Algorithms are used to manipulate data and get results. Data is used to inform decisions. Lists and tuples are used to simplify data collection and implementation. Debugging practices are used to correct errors in code and logic.
Control Structures	<ul style="list-style-type: none"> Program does not have any if or if/else or if/elif/else statements. Program does not use any while loops. Nested loops or if statements are not used, or are used incorrectly. 		<ul style="list-style-type: none"> While loops and if statements are used to control the flow of execution. Conditional and logical operators are used appropriately. Nested while and if statements are used when needed.
Use of Outputs LEDs, speaker, motors	<ul style="list-style-type: none"> No output is produced. 		<ul style="list-style-type: none"> One or more outputs are used to convey data or perform a task.
Collaboration	<ul style="list-style-type: none"> Students work independently or uncooperatively on a team. 		<ul style="list-style-type: none"> Students work collaboratively with shared tasks in their team to complete the project.
Synthesis / Purpose	<ul style="list-style-type: none"> No clear purpose for the program. Program does not incorporate learning across the mission pack. 		<ul style="list-style-type: none"> Purpose of the program is clearly stated. Program combines learning, concepts and code from several missions.
Code Completion	<ul style="list-style-type: none"> Code will not run or doesn't complete the task correctly. 		<ul style="list-style-type: none"> Code runs and accomplishes its task without any errors, including logic.