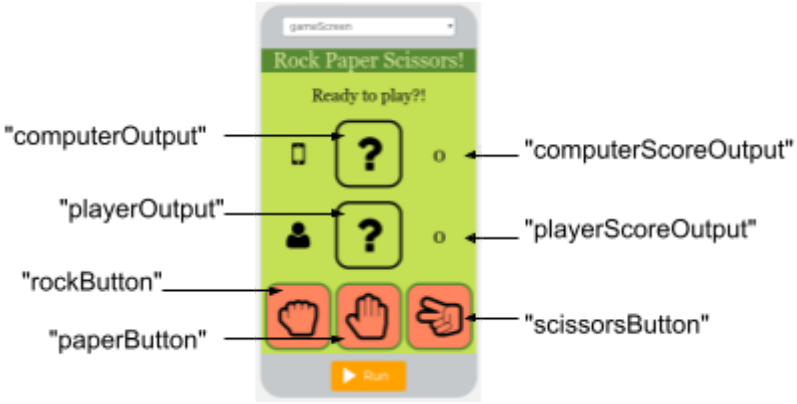
Name(s)_____ Period _____ Date _____

Activity Guide - Parameters and Return Make

Step 1 - Try the app

Try playing the game. Pay attention to:

- What each button does
- How the screen gets updated after clicking each button



Step 2 - Plan

This Make project is a little different than other ones. Instead of writing all of the code yourself, most of it has already been written for you. There are just three functions at the bottom that have been designed and have comments explaining how they should work but the code for each function is incomplete or only works for some inputs. Read the comments for these three functions carefully. You can also quickly read the rest of the program if you like.

In the table below record any notes for how you will build each function

	Notes for Building this Function
findIcon()	
randomChoose()	
decideWinner()	

Step 3 - Write Your Code

- Write the code for the app, using your plan above and the comments provided in Code Studio to help. You can also check the Help & Tips tab for Programming Patterns you can use
- Step You Can Follow
    - Write the code for findIcon() first. Notice that the list of possible icons is available on Line 2
    - Test your code. Once your program is working you should see the correct icon show up on the game board when you pick rock, paper or scissors.
    - Write the code to make the randomChoose() function work as described. You should only be working inside that function
    - Test your code. Once your program is working you should see the computer player is able to pick randomly between rock, paper, and scissors.
    - Write the code to make the decideWinner() function work as described. You should only be working inside that function
    - Test your code. Once this function is done your app should be completely working and correctly updating the score on the screen.
    - For reference: Rock beats Scissors, Paper beats Rock, Scissors beats Paper
    - Add comments to each of your functions explaining how you wrote the code.
    - Test the app to make sure it works for every combination of rock, paper, and scissors.
- Extension Ideas
    - Have the game end when one player gets to 10 wins
    - Keep track of the current streak of wins in the game

Step 4 - Submit

Before you submit, check the rubric below to make sure your program meets the requirements of the task.

Category	Extensive Evidence	Convincing Evidence	Limited Evidence	No Evidence
findIcon Function	The function returns the correct values for all input values.	The function returns the correct values for most input values.	The function returns the correct values for some input values.	The function does not return correct values for any input values.
randomChoose Function	The function returns the correct values for all input values.	The function returns the correct values for most input values.	The function returns the correct values for some input values.	The function does not return correct values for any input values.
decideWinner Function	The function returns the correct values for all combinations of inputs.	The function returns the correct values for most combinations of inputs.	The function returns the correct values for some combination of inputs.	The function does not return correct values for any combination of inputs
Code runs without errors.	No errors are present in the required code.	Some errors are present in the required code.	Many errors are present in the required code.	The code does not run.
Coding Comments	Comments are used to correctly explain the purpose and functionality of all functions.	Comments are used to explain the purpose and functionality of most functions.	Comments are present, but are not used to explain the purpose or functionality of any functions.	Comments are not present.

Step 5 - Create Performance Task Writing Practice

This question is based on the project you submitted.

Consider the procedures used in your program to perform specific tasks.

Explain how the use of these procedures helps manage the complexity of your program. How do these procedures improve the program's organization and make the code easier to maintain and modify?