| Sprint 8, Assignment 11.5 Please also update the doc name with correct numbers. | |
| --- | --- |
| Assignment type: JS Interactive | |
| Assignment name  Can remain the same as the assignment topic name, or…  can be created depending on the task in the active form, e.g. “Create your own X” | “ ‘While’ loops “ |
| BDG Description\*  What is the task and why is it important?  In this part, it’s encouraged to think about storytelling and future job-specific context e.g. “You’ve been asked to help out X with Y. They want Z on their website, yet aren’t too sure on how to achieve it”.  Drawing on practical examples and adding context can increase a student's motivation and increase long term learning according to Instructional Design principles, because this helps to relate some familiar or existing knowledge to new bits of information.  This will appear in the course as text before a button, leading to the interactive platform assignment. | An early education company wants to add a new math module to their ipad suite, and they’ve asked you to create a game that challenges children to do basic addition! |
| The Assignment  A short specific description of the assignment and tasks using bullet points that the student will need to do. | Write a ‘while’ loop, and use conditionals to confirm a user’s input |

| Steps  Step-by-step instructions on what the student should do. | | |  |
| --- | --- | --- | --- |
| Step # | Step  Write each small step of the task | At least 1x hint(s)  Write some text (not necessarily, but can also be a part of code if relevant) which would hint the student to figure out the correct step forward. | The correct output should be…  (if relevant to the task) |
| 1 | Create a variable to store the number of “tries” or attempts to solve the problem the child will recieve |  | **let lives = 3;** |
| 2 | Write a ‘while’ loop and “game over” message | To begin, have your loop   1. display the number of lives remaining to the console, and 2. reduce that number by 1   Console-log something after the loop is finished indicating that the game is over | let lives = 3;  **while (lives > 0 ) {**  **console.log(lives);**  **lives--;**  **}**    **console.log("Game Over!")** |
| 3 | Prompt the child to solve a simple addition problem (5 + 2 etc) and store their answer in a variable | * Delete the line **console.log(lives);** * Declare a variable, and store in it a user’s input using javascript’s “prompt” method * Have the prompt display a string containing a math problem * Have the string also tell the child how many chances they have left, using template literals | let lives = 3;  while (lives > 0 ) {  **let result = prompt(`You have ${lives} chances left. What is 2+2?`);**  lives--;  }  }  console.log("Game Over!") |
| 4 | Use a conditional to confirm whether or not the user’s input matches the correct answer, and deduct a life if it is not | * If the input is correct, a “congratulations!” message should be displayed and a break operation should stop the program * If it is not, make the “lives” variable decrease by 1 and tell the user that they are wrong via the console   Note: remember to only compare value, and not type (“==”) | let lives = 3;  while (lives > 0 ) {  let result = prompt(`You have ${lives} chances left. What is 2+2?`);  **if (result !=="4"){**  **lives--;**  **console.log("I'm afraid that's not right - try again");**  **} else {**  **console.log("Congratulations! That's the correct answer.");**  **break;**  **}**  }  console.log("Game Over!") |
| ... |  |  |  |