**A-Frame Components Activity**

In this activity, you will learn how to write your own A-Frame component using JavaScript syntax. First, open up the file called “log-leap.js” and let’s get the A-Frame component working so that it logs our leap hand data correctly.

1. Find the Schema property of the definition and define two data values
   1. Call the first one “id” and give it a type of “int”
   2. Call the second one “trial” and give it a type of “int”
2. Now, we move down to the init function, where we need to define and initialize two variables. IMPORTANT NOTE: variables in A-Frame components need “this.” before the name if you want the variable to be accessible from anywhere in the A-Frame component functions. So if you have a variable called “speed” and want this variable to be accessible from all functions in the component, you need to add “this.” in front of it like this: “this.speed = 0;”
   1. Add a variable called “leapcontroller” that will be given the leap controller data.
      1. We do this by setting it equal to the following:

document.querySelector('a-scene').systems.leap.controller;

* 1. Now, we want to add a variable called seconds to keep track of the seconds passing by in our tick function
     1. A good starting value is 0

1. You will notice we have lots of column headers that need to be printed to the console log
   1. Use the “console.log()” function here and pass in as an argument within the parenthesis that has all the header column names
   2. HINT: which variable are we storing all of our column names in?
2. Now we switch over to the tick function, of which is where we store data about the leap hands every frame. You will notice we have an if and else if statement in the tick function. We are trying to grab information from the left and right hand.
   1. Underneath where it says QUESTION #4 PART #1, we see code that is checking if the type of the hand is equal to something…where it says insert-hand-name-here, replace the string with ‘left’
   2. Now, go to QUESTION #4 PART #2 and do the same thing within the else if part of the statement but insert ‘right’ instead
3. Great! Now we are accessing the data from our left and right hands when they are present in the scene. Otherwise, fake filler data is added in instead. Now, we have time and timeDelta given to us from the tick function. But, they are only counting time in milliseconds and we also want the time in just seconds.
   1. So, where it says QUESTION #5, add code that will increase the value of seconds by one
   2. You can do this in multiple ways, but here the easiest way is with the “+=” operator
      1. For example, if you have a variable “a” and want to increment it’s value by 10, you type “this.a += 10;”
4. Now we need to access the data we stored and add it to the log, of which will eventually be printed out to the console. You will notice empty strings in the log assignments below where it says QUESTION #6. We are going to replace those with our data
   1. IMPORTANT NOTE: In order to access our schema data variables, we need to call ‘this.data’ and then follow that with ‘.’ and our data name. For example, if I wanted to grab the ‘id’ from the schema, I would call ‘this.data.id’
   2. For the first line, replace ‘’ with the id number
   3. For the second line, replace the ‘’ with the trial number
   4. For the fifth line, replace the ‘’ with our seconds variable (what was the proper way to access our variable from within any function in our component?)
   5. We stored our left-hand data in ‘lefthand’ and right-hand data in ‘righthand’
      1. In the sixth line, replace ‘’ with lefthand
      2. In the seventh line, replace ‘’ with righthand
5. Now, we pass in our ‘log’ variable to the console.log() function as an argument!
6. For the last part, we are going to save and close our log-leap.js file and open up the leaphand.html file where our A-Frame scene is. We want to use our javascript code to log data.
   1. To get access to our code, we add…

<script src="log-leap.js"></script>

…as a child element of the head element at the towards the top of the file

* 1. Now, we need to add our new component/attribute to an a-entity, so let’s add our ‘log-leap’ component to our a-entity that also as the camera and look-controls components added to it
  2. However, we need to add in which trial number this scene represents and what the participant’s id number is. We added id and trial to our schema, so let’s set them!
  3. Next to log-leap, add **=”id: 1; trial: 1;”**
  4. You’ll notice we set id to 1 and trial to 1, so change them both to 0 and then we are done!