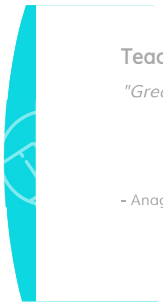
ıck

ɔngrats Kyle-Jin! This project has been arked as completed.

roject Rating



Teacher's Comment

"Great work dear and well done😍😍😍"

- Anagha Rajeev

Was this helpful?

ADV-C118: QUICK DRAW WEBAPP - 2 Completed

his Project, You Will Have To Complete The Designing Of A Web Page And Add Canvas And Timer Code To It Using Js Code.

l of the Project:

ı class 118, we completed designing a web application and drew a canvas on the web age.

UICK DRAW WEBAPP : In this project, you will have to complete the designing of a web age and add canvas and timer code to it using JS code.

*** This is a continuation of the project that we did for Class 117. Please complete that roject before attempting this project****

y:

Work is launching a drawing app for kids under the age group of 5-8, and now they want n application, which will be a doodle game, and they liked your UI. Now, they want you to dd the timer component to the web app in such a way that the user gets a time limit to erform the drawing task.

t the end of this project you should get the output something like this .

:ing Started:

1. Some predefined CSS code is provided, and this CSS code is done by you in the class. So download the **C118_predefined_code.css** file from this Link , and copy-paste the CSS code in the style.css file in which you were working on, in the last project.

2. And then start adding code for JS code in the **main.js** file.

:ific Tasks to complete the Project:

ou need to continue coding in the **main.js** file from the previous project:

1. The functionality of this project will be to create a canvas to draw the sketch - create the **draw()** function to draw the sketch, create a **check_sketch()** function to check if the drawn sketch and sketch to be drawn are equal. So, let's start by defining a function to create a canvas.
2. First, define a function **updateCanvas()**. The functionality of this function will be to clean the canvas and update with the new sketch name to be drawn after a particular time. Inside this function, add the following code:
    - Set the background of the canvas as white.
    - Add the following code in the **main.js** file:
        - Generating random numbers using Math.floor() function till the length **quick_draw_data_set** and store it in a variable **random_number**.
        - Using console.log, print the value of the array by passing this random_number as index to the array of the **quick_draw_data_set** and store it into the sketch variable.
        - Update the HTML element with the **sketch** variable inside this function.
3. Now, define a function **setup()**, and inside the function, add the following code:
    - Define a variable **canvas**, and inside it, write the code for creating the canvas with 280 pixels height and width.
    - Then, place the canvas in the center.
    - Set the background of the canvas to white.

Community Link

Publish to Community

Edit Your Project

Last Submitted

Previous Submissions

26th Dec 2021 Open Link

Start Project

Submit Your Project

Learn how to submit your project

Paste your project URL

Submit Project

Class Summary

This project is based on your last class ADV-C118

View Class Summary

Ask a doubt to your teacher

- Set the background of the canvas to white.

4. Define a **draw** function, and inside it, call the function **check_sketch()** which we will define later in the code.
    - Then, using the **if** statement, check if the **drawn_sketch** variable is equal to the **sketch** variable. And if it is equal, then update the following:
        - Update variable **answer_holder** with string **"set"**. This is to indicate that the drawn sketch is equal to the sketch.
        - Then, increment the value of the **score** variable. We are doing this for every right sketch drawn, to increase the score.
        - Now, update the text of the **span** tag which holds the string **"Score:"** and concatenate the **score** variable with it.
5. Now, define a function **check_sketch()**. The functionality of this function will be to check if the drawn sketch is equal to the sketch to be drawn and keep on incrementing time and checking if the time is up. Inside it, add the following code:
    - Increment the **timer_counter** variable value by 1.
    - Update the text of the **span** tag which holds **time** with the **"Timer"** string and concatenate the **timer_counter** variable with it.
    - Using the **console.log()** function, print the value of the **timer_counter** on the console.
    - Then, using the **if** condition, check if the **timer_counter** value is greater than 400 (We have set this as 400, but you can set any value you want). We are doing this because we are setting the time limit for drawing the sketch to 400 milliseconds.
      This is done because the name of the application is quick draw, and you have to draw the correct image quickly within the time limit.
    - So, we will check if the timer value is greater than 400:
        - Set the **timer_counter** to **0**. This is done because we are setting a time limit of 400 millisecond to draw the sketch.
        - Set the **timer_check** value as **completed** when the **time_counter** reaches the value 400.
    - Now, again create another **if** condition for checking if the **timer_check** value is equal to **"completed"** OR the **answer_holder** variable value is equal to **"set"**, and update the following variables:
        - Set the **timer_check** variable as empty, so that it gets updated for a new sketch to be drawn.
        - Set the **answer_holder** variable as empty, so that it gets updated for a new sketch to be drawn.
        - And call the **updateCanvas()**.

ı this project, you won't get the complete working web app and that's fine. We will be ɔmpleting this project in the upcoming projects.

lake the webpage using your creativity, but make sure that all the above components are ıere in the page.

nitting the Project:

1. If you have created the **quick_draw-project** folder on github then:
    - Upload **all the** files on which you have worked in the current project, in the **quick_draw-project -project** folder on GitHub. You can get the steps to do this by clicking on this link.
2. If you have **not** created the**quick_draw-project -project** folder on github then:
    - Create a folder and name it as **quick_draw-project** and upload **all the** files related to the **quick_draw-project** project on GitHub. You can get the steps to do this by clicking on this link.
3. Copy the hosted link which you will get after uploading all your files on GitHub and submit it in the Student Dashboard Projects panel against the correct class number.

s:

1. Increment the variable value by 1. For this, let's take an example:

   We define a variable counter, and assign 0 to it. Like this:
   counter=0
   Suppose, for any condition in if(), we want to increment the counter value by 1. So, we will write like this:

   if(condition){

```
if(condition){
counter++
}
```

Here, counter++ increments the value by 1.
So, similarly, you can increment the timer_count value by 1 in our application.

2. Using the OR condition in if statement:
   Suppose, we want to draw any line on the canvas with only stroke color as 'black' or 'red'. Then, we will check the condition like this:

```
if(stroke=='black' || stroke=='red'){
line(30,40,25,50)
}
```

Here, we are checking the stroke color is black or red, so for the or condition, we are using the pipe (||).
Similarly, you can apply the above concept for **checking if the timer_check is completed or answer_holder is set.**