

Project 48



DP
EDUCATION

Coding School

Letters Unscrambler

8 Points: 0

OEHRS

Submit

Built on Code Studio ▾

How It Works

See the App

Start here

Built on Code Studio ▾

Let's correct the letters of the word where the letters have been changed.

- ❖ The screen and other elements you need for this have been created. But if you want you can change them with design tools.
- ❖ Now that the design is given, let's focus on the coding part.
- ❖ First, let's create the required variables and give the initial values to them. Create one variable as number. Create another variable as Points and set its initial value to 0.
- ❖ Then let's create the corresponding code to change the images in 20 seconds and type the correct spelling that matches them and collect points. Use the code below to loop for 20 seconds.

```
timedLoop(20000, function() {
```

- ❖ So, first of all, in one round, we must get a random value for the number variable, which is the variable we created above. This can help to create random display of our images. Let's use the following block for that.

```
var number = randomNumber(1, 7);
```

- ❖ Then use the block called setImageURL to display the gif related to the word "imageForWord" according to the random number selected above. This allows different images to be displayed by changing the image url of the same image.

```
setImageURL("imageForWord", "image" + (number + ".gif"));
```

Here "id" is created as imageForWord and url is created as above. Because each image is renamed with a number at the end, the random number can be created as the number in the image url and random images can be displayed

- ❖ The second image shows an image in which the letters of the English word related to the above image have been changed. It can also be obtained by generating the url as above. Use the following block for that.

```
setImageUrl( "imageofScrambledWord", "wordImage" + (number + ".png") );
```

Give the value of imageofScrambledWord as its "id" ..

- ❖ Then let's create what should happen when the submit button is clicked. Use the onEventClick block for that. Give buttonSubmit as its id
- ❖ First, the value of the textbox should be stored in a variable. For that, create a variable as input and store the value received from the textbox in it. For that, do as in the block below.

```
var input = getText( "txtInput" );
```

Then, use the toLowerCase block as follows to simplify all the letters in the resulting value.

```
var inputLowerCase = input.toLowerCase();
```

- ❖ At the end is to check if the entered value is correct. For this you can use if else if block. If the displayed image is displayed, if it is correctly entered as "computer", let's create a point to add. For that, create blocks as follows.

```
if ( number == 1 && inputLowerCase == "computer" ) {  
    points = points + 1;
```

Otherwise, if the second image has appeared, use the else if block as follows to check whether horse has been entered correctly.

```
) else if ( number == 2 && inputLowerCase == "horse" ) {  
    points = points + 1;
```

- ❖ In the same way, check whether the correct letters are typed and entered for all the images. For that use the above else if statement.

The numbers and names of the figures are given below.

- 1 – computer
- 2 – horse
- 3 – night
- 4 – sunrise
- 5 – swimming
- 6 – teacher
- 7 – badminton

- ❖ After checking all the values in that way, use the following code to display the score.

```
setText(▼ "points", points);
```

- ❖ Use the following block to hide the submit button to avoid increasing the number of points by entering the correct word repeatedly in one round.

```
hideElement(▼ "buttonSubmit");
```

- ❖ Add the following block immediately outside the onEventClick block to show the submit button again in the next round.

```
 );  
showElement(▼ "buttonSubmit");
```

- ❖ Use the following block below the above show element block to delete the value of the input textbox after one round at the start of the next round.

```
setText(▼ "txtInput", "");
```