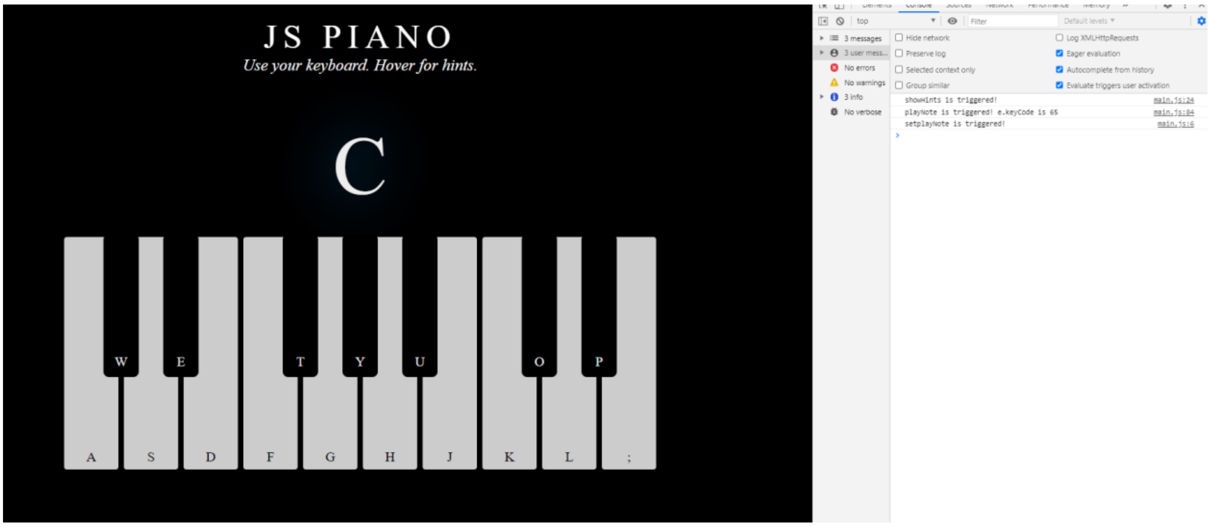
NM2207

Session 08

JSPiano Challenge Brief

Overview of what we will do today:

- Practice accessing a list of HTML elements and setting their opacity
- Practice using keyboard events
- Practice playing audio



☐ Hide network
☐ Preserve log
☐ Selected context only
☐ Group similar
☐ Log XMLHttpRequests
☑ Eager evaluation
☑ Autocomplete from history
☑ Evaluate triggers user activation

```
showHints is triggered!                           main.js:24
playNote is triggered! e.keyCode is 65            main.js:84
setplayNote is triggered!                          main.js:6
>
```

What’s in the box

0.1 Look at the image included in this document. Examine the letters you see, and examine the console output. This is the final output we want today, but your keyboard will also play music!

0.2 Try to figure out which elements might be the one where the big “C” is printed, and which correspond to the smaller letter on the keys. Are they visible in the template you have?

- We will be setting the innerHTML of the element containing the big “C” when a note is played (playNote function), and setting the opacities of the other letters when the spacebar is pressed (showHints function).

Part 1 (20 minutes)

Summary of learnings

- **Creating an event listener on the window, listening for a keydown event. Depending on which key is down, choose which function to run.**

Warm up (20 minutes)

The Overall Goal of this task set is to create an event listener for the window. If the spacebar is pressed, the showHints() function should be called. Otherwise, the playNote() function should be called and it should take the event variable as argument. Neither function returns any value.

The specific tasks for you to accomplish starting from the template code are:

1.1 Create an eventListener so that the window can listen for a keydown event.

1.2 The keyCode of the spacebar key is 32. If the spacebar is pressed, then the showHints function should be triggered

1.3 Otherwise, the playNote function should be triggered. Pass it the event variable.

1.4 We want to track the on/off state of showing hints. So create a global variable called hintsOn to track the status, and set it to 0 when the page is loaded.

Part 2 (20 min)

Summary of learnings

- Fill up the showHints function to set the opacity of hints to 1 or 0

The Overall Goal of this task set is to show the hints depending on whether hintsOn is 0 or 1. Specifically:

2.1 Create a list by getting all the elements from the DOM that are called hints.

2.2 How many elements do you expect there to be, and how can you check this? Print the size of the list to the console.

2.3 Now, if hintsOn is 0, then we will switch on the hints now. For each element, set its opacity to be 1. You can do this in two ways

(a) By accessing each element in turn, e.g., allHints[0].style……

(b) By using a loop, e.g.

```
for ()
{
        //code for setting opacity goes here
}
```

2.4 If hintsOn is 1, then we will switch off the hints now. the opacity of each element should be set to 0 instead. Again, you can do this in either of the two days provided above.

2.5 After setting opacities, remember to set the new state of hintsOn.

Part 3

Summary of learnings

- **Fill up the playNote function to play audio corresponding to the pressed key.**

3.1 You will create a new audio object by using new Audio() that takes in a file location, and then you will play that audio.

- First you need to get the file corresponding to the key that was pressed. You can do this by

  (a) use document.getElementById to get a pianoKey object (but you could have called this anything you wanted), and then

  (b) do a getAttribute("src") on that object to get its sound file property value.

  (c) use this inside your Audio() method to create an audio object, following the syntax in Lecture 5's slides.
- Set the current time of this audio to be 0 and then play this audio, following the syntax in Lecture 5's slides.

3.2 If there are no audio files corresponding to the keycode, then it will generate errors. You can handle this by putting the code you wrote in 2.1b and onward into an if condition.

Something like this should work:

```
If (pianoKey != null)
{
        //code to play audio goes here.
}
```

- **As seen in the image provided, show the note currently being played in the innerHTML of nowplayingNote**

More practice

Return to W04 and W05 assignments for practice in passing objects as arguments, for e.g.:

1. Return to the Session05.assignments.
2. For both tickTock and msDraw, store offSetX and offSetY of the event variable as the x and y properties of an object called clickLocation. This would be a local variable.
3. Also store the starting position of the needle or msDraw in an object called originalLocation. This would be a global variable.
4. Pass the clickLocation object into the functions which update the pathStrings.