

Debugging Log

Logged a total of **0.9 hours** of debugging.

Entry 1

17 minutes spent debugging

Failure

When interring the filter, the first character gives correct filtering, but with the second character typed in, the selections in move to disappeared.

Experiments

Experiment 1.1

Question: Is the new state "filter" in the EditorState correctly defined and updated?

Steps Taken: console log this.state.filter

Result: the filter is correctly defined as undefined at the begin and changed while typing, also if delete all the characters then it becomes to undefined again.

Lesson: It is not because the filter state not being correctly set up and update

Experiment 1.2

Question: It actually doing the right thing if we type the correct upper and lower case

Steps Taken: add lower() to convert both BUILDINGS[i].longName and filter to lower case

Result: It Works

Lesson: Do upper case and lower case converting when comparing strings

Defect

```
if (BUILDINGS[i].longName.includes(this.state.filter))
```

Did not convert two strings to lower case so did not perform the right behavior.

Mutation

This failure **was not** due to mutating something that was not supposed to be mutated. Does not change the state

Entry 2

15 minutes spent debugging

Failure

When typing in filter, the console complains about: A component is changing a controlled input to be uncontrolled.

Experiments

Experiment 2.1

Question: Exactly what is the controlled input and why is it uncontrolled

Steps Taken: I tested if it's because I initialized the this.state{filter = undefined} by changing it to this.state{filter = ""}

Result: Bug is still there

Lesson: not because the constructor

Experiment 2.2

Question: When it says uncontrolled input, does it mean the filter state just cannot be undefined, but has to be a certain value?

Steps Taken: I deleted the '?' in the EditorState = {filter?: string}, and modified the related code, for example the filtering on the building

Result: Bug is gone, console no complain

Lesson: Simply do not use undefined in State declaration.

Defect

```
filter?: string;
```

When declaring a state, make it possible to be undefined, even though handled every case, it's still a bad practice to make it possibly undefined.

Mutation

This failure **was not** due to mutating something that was not supposed to be mutated. I guess it's close to a bug on Mutation, but it is not because the filter field should not have been mutated.

Entry 3

22 minutes spent debugging

Failure

When check off the move to new location check box, the mark still move to the new location

Experiments

Experiment 3.1

Question: Why is it still move to the new location, could it because I did not change the doMove(boolean) state?

Steps Taken: Console log the doMove state do see if it changes with the check box changes

Result: Does not change at all

Lesson: There is no function that control the doMove field's state

Experiment 3.2

Question: Does add an handleChangeCheckboxChange and set the state solve this problem?

Steps Taken: Add a function handles setState() on doMove

Result: Still when the check box is checked off, the marker still move after click save button, but the doMove field is update to false

Lesson: The logic of the handleSaveButton is not correct

Experiment 3.3

Question: Does add another condition in the if statement of handleSaveClick solve this bug?

Steps Taken: Make another condition check if the check box is checked or not to decide what is the parameters that should be passed into the onSaveClick callback function

Result: Now everything is correct, when uncheck the checkbox, the color still changes but the location of the marker does not change

Lesson: It is hard to remember all the thing I have to do for each UI, so might need create a todo list for that.

Defect

```
if (this.props.moveTo !== undefined) {
```

forget to update the state and Condition does not sufficient

Mutation

This failure **was not** due to mutating something that was not supposed to be mutated. Even forget to mutate the state but not should not mutate the state

Edit Log