

Software Engineering

T34 – Software Development in Practice

3MIEIC04_B

(9h December 2017)

Bárbara Silva	up201505628@fe.up.pt
Julieta Frade	up201506530@fe.up.pt
Ventura Pereira	up201404690@fe.up.pt

Índice

Issue #11800: The shortcuts key (Ctrl++, Ctrl+.) would not work if input a large number as the font size on Themes Settings dialog 2

Issue #11634: Brackets window has too thick (7px width) border on Windows 10. Windows 10 uses 1px border around windows 6

Issue #11800

Issue #11800: The shortcuts key (Ctrl++, Ctrl+-) would not work if input a large number as the font size on Themes Settings dialog.

Steps:

1. Launch the Brackets
2. Click View > Themes...
3. Input a large number as the font size, such as "100px, 1000px..."
4. Click Done
5. Try to Press Ctrl++ and Ctrl+- to increase or decrease the font size

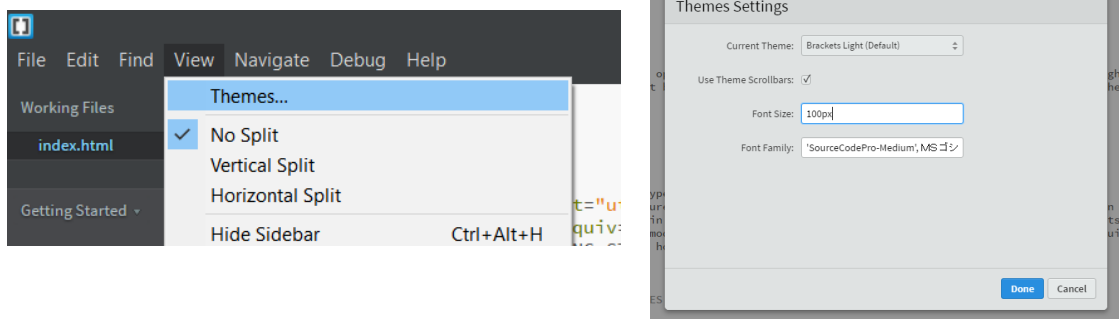
Result: The shortcuts key doesn't work

Expect: The shortcuts key could work normally

(<https://github.com/adobe/brackets/issues/11800>)

Requirements

The expected was the shortcuts keys Ctrl++ and Ctrl+- do zoom in and zoom out even though the font size on themes settings dialog is a large number, but they don't.



The problem was that when you try to increase or decrease the font size, Brackets doesn't let you if the *new font size* is more than the *maximum font size* (72px) or less than the *minimum font size* (1px). Therefore, if the font size in themes settings is 100px and you try to decrease it pressing Ctrl+- nothing happens because the new font size will be 99px which is still more than 72px.

Source Code Files

File involved with the issue: /src/view/ViewCommandHandlers.js

Code fragments involved the issue:

- Declarations of the minimum and maximum font size in pixels:

```

/**
 * @const
 * @private
 * The smallest font size in pixels
 * @type {number}
 */
var MIN_FONT_SIZE = 1;

/**
 * @const
 * @private
 * The largest font size in pixels
 * @type {number}
 */
var MAX_FONT_SIZE = 72;

```

- Function that decides whether or not to update the font size:

```

/**
 * @private
 * Increases or decreases the editor's font size.
 * @param {number} adjustment Negative number to make the font smaller; positive
 number to make it bigger
 * @return {boolean} true if adjustment occurred, false if it did not occur
 */
function _adjustFontSize(adjustment) {
    var fsStyle = prefs.get("fontSize");
    var validFontSizeRegExp = new RegExp(validFontSizeRegExpStr);

    // Make sure that the font size is expressed in terms we can
    // handle (px or em). If not, simply bail.

    if (fsStyle.search(validFontSizeRegExp) === -1) {
        return false;
    }

    // Guaranteed to work by validation above.
    var fsUnits = fsStyle.substring(fsStyle.length - 2, fsStyle.length),
        delta = fsUnits === "px" ? 1 : 0.1,
        fsOld = parseFloat(fsStyle.substring(0, fsStyle.length - 2)),
        fsNew = fsOld + (delta * adjustment),
        fsStr = fsNew + fsUnits;

    // Don't let the font size get too small or too large. The minimum font size
    // is 1px or 0.1em
    // and the maximum font size is 72px or 7.2em depending on the unit used
    if (fsNew < MIN_FONT_SIZE * delta || fsNew > MAX_FONT_SIZE * delta) {
        return false;
    }
}

```

```

    }

    setFontSize(fsStr);
    return true;
}

```

Design of the Fix

Because we think it makes sense to be able to decrease the font size even though it is still more than the allowed we tried to solve the problem. So now if the font size is more than the maximum and you try to decrease it, it will decrease to the maximum font size, and if the font size is less than the minimum and you increase it, it will increase to the minimum font size.

Fix Source Code

Change this if clause:

```

// Don't let the font size get too small or too large.
// The minimum font size is 1px or 0.1em and the maximum font size is 72px or 7.2em
// depending on the unit used
    if (fsNew < MIN_FONT_SIZE * delta || fsNew > MAX_FONT_SIZE * delta) {
        return false;
    }

```

To this if else clause:

```

//The minimum font size is 1px or 0.1em and the maximum font size is 72px or 7.2em
//depending on the unit used

    //If the goal is to decrement but the current font size is more than the
    maximum font size
    if(adjustment < 0 && fsNew > MAX_FONT_SIZE * delta){
        fsNew = MAX_FONT_SIZE * delta; //The new font size is set to the maximum
font size
    }
    //If the goal is to increment but the current font size is less than the
    minimum font size
    else if(adjustment > 0 && fsNew < MIN_FONT_SIZE * delta){
        fsNew = MIN_FONT_SIZE * delta; //The new font size is set to the minimum
font size
    }
    else if (fsNew < MIN_FONT_SIZE * delta || fsNew > MAX_FONT_SIZE * delta) {
//Don't let the font size get too small or too large.
        return false;
    }

```

And because we are changing the value of *fsNew*, we must define *fsStr* after that:

```
var fsStr = fsNew + fsUnits;
```

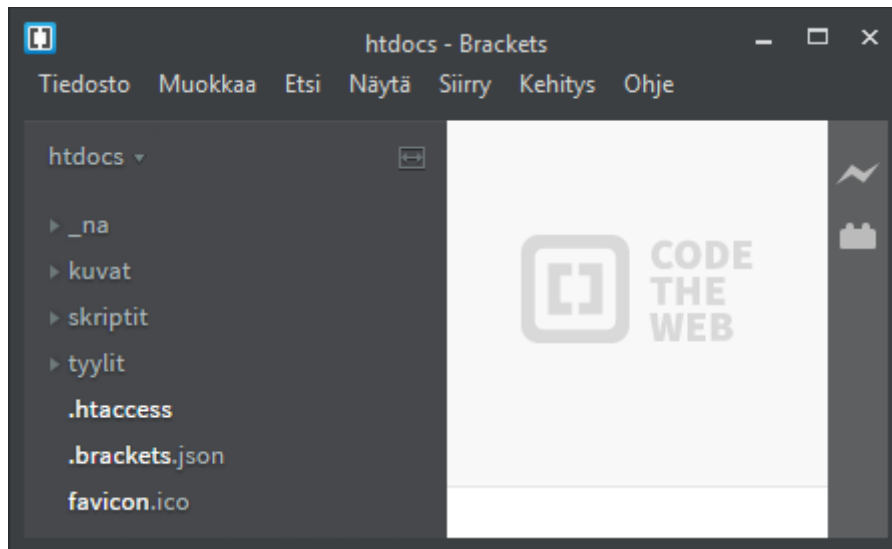
Submit the Fix

We tried to submit our changes doing a pull request but it was denied by a bot test. Because we didn't know what had failed, we asked for explanations and the person responsible for the repository said they didn't had time to review it yet.

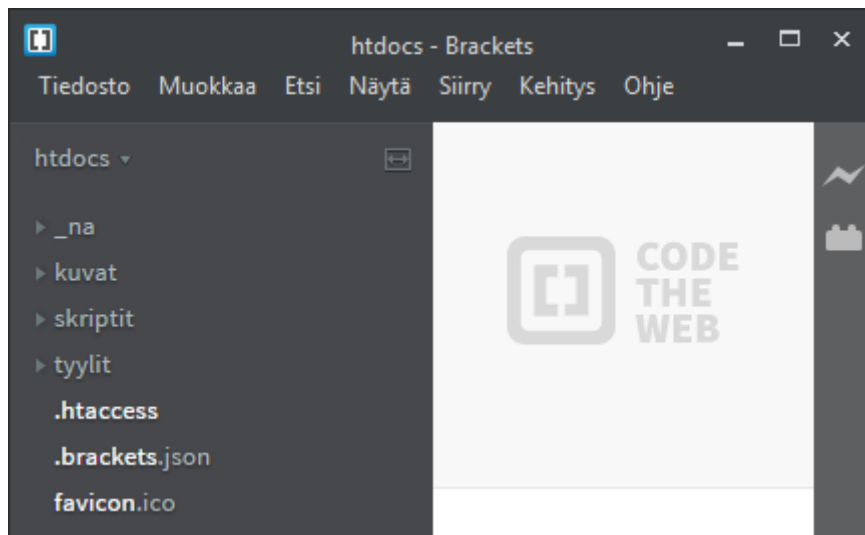
Issue #11634

Issue #11634: Brackets window has too thick (7px width) border on Windows 10. Windows 10 uses 1px border around windows.

Currently, the window looks like this:



It could look more like this:



<https://github.com/adobe/brackets/issues/11634>

The issue, identified by the number #11634, which title is “too thick border on Windows 10”, is described as an aesthetic problem - a layout that doesn’t please the user. It only verifies itself on Windows, where the border lines of the display window are too thick.

The Requirements

The requirement is that the window's border lines are thinner.

Source code files examined

The files that we had examined were the *.less* that took care of the layout and the *.js* files that took care of the panel:

- layout.less;
- pane.js;
- brackets.js;
- panelManager.js

Design of the fix

In order to handle the proposed problem, we had to diminish the width/height of the border bars, while not ruining the appearance of the window's content and the way it resizes.

Fix source code

The method to solve this was by hardcoding the desired values, instead of the defined variables they had.