## Changes introduced

To find the lines introduced by the buggy commit, you can use git diff. You'll need the ID of the buggy commit, which you just found to be 547f4171a82ec6429d002c1acef357aec41d3f17. Then you'll need the ID of the previous commit, which will be the commit below it in git log. (That's because git log lists the most recent commit first.) That turns out to be 71d52709ddc4066e7a79a1d0a412e43429a0cdeb.

Thus, by running git diff 71d52709ddc4066e7a79a1d0a412e43429a0cdeb 547f4171a82ec6429d002c1acef357aec41d3f17, you can find out that the lines changed by the buggy commit were:

- return !(

- bounds1.end\_x < bounds2.start\_x ||

- bounds2.end\_x < bounds1.start\_x ||

- bounds1.end\_y < bounds2.start\_y ||

- bounds2.end\_y < bounds1.start\_y

- );

-

+ if (bounds1.end\_x < bounds2.start\_x) {

+ return true;

+ }

+ if (bounds2.end\_x < bounds1.start\_x) {

+ return true;

+ }

+ if (bounds1.end\_y < bounds2.start\_y) {

+ return true;

+ }

+ if (bounds2.end\_y < bounds1.start\_y) {

+ return true;

+ }

+ return false;

This change represents an "or" expression being separated out into several "if" statements. The number of functions did not change, and no variables were renamed.

## File changed

Near the top of the git diff output, you can see the lines

--- a/js/utils.js

+++ b/js/utils.js

This indicates that the file changed was js/utils.js, that is, the file utils.js within the js directory.

## What caused the bug

Based on the change that was made, a reasonable guess is that the bug is some sort of logic error - maybe the new version does not return true and false at the correct times.

It turns out that this is correct. The new code has true and false reversed! Even if you weren't sure exactly why the bug was there, congratulations! You tracked down exactly where the bug was introduced, and knew which lines introduced it, without knowing the code base. All you had to know was how to use Git.