

# Crushing Bugs Dev Notes

## Before Solving

### Locating the Bugs

If I find I had the incorrect location of the bugs, it will be logged later in this document.

1. Bug 1 is located **line 30** and below (under this function) in the **main.js file**. This is a mandatory fix.

```
function allowDrop(e) {  
  e.preventDefault();
```

The issue is the puzzle pieces can stack on top of each other, and once they do you are unable to receive the pieces stuck below. The only way to get them out as of now is moving the pieces around in hopes to create a free space to place that piece, even then they should not stack whatsoever. If they are allowed to stack on top of each other, then there must be an issue with “allowDrop” since there is no command that allowDrop is not activated when another puzzle piece is in its way.

2. Bug 2 is located **line 11** and below (under this function) in the **main.js file**. This is a mandatory fix.

```
function changelImageSet() {  
  gameBoard.style.backgroundImage = `url(images/backGround${this.dataset.bgregf}.jpg)`;  
}
```

The issue is when you change puzzles, the pieces on the board stay on. They need to go back into place off to the side. This bug does not include the fact the pieces do not change (that is the bonus bug). As the board changes, so does the position of the pieces, that is why I think the bug is around in this line of code.

### Considerations Before Solutions

- Bug 1: I first want to try the idea of adding a command that states do not allow allowDrop function to activate when one of the pzlPieces is already in a drop zone. So, the following must be considered when creating the solution:

- dropZones needs to know when one of the pzlPieces is dropped in one out of four spots (which it already does as we can see we are able to drag and drop)
- pzlPieces needs to know that it cannot stack on top of each other (owning its own drop zone until moved elsewhere)
- dropZones should know that the side area where the pzlPieces are first shown can also be a drop zone (or the returning area) OR maybe have a way to reset the puzzle (just make sure the pzlPieces are not stuck on the board)
- Bug 2 needs to literally move the pieces back into the container, so consider
  - pzlPieces needs to know where to move to/what container
  - the pieces need to know when puzzles are changing

## During Solving

### Obstacles, Research, & Solution

As I work on the project, this is logging situations I encounter, problem solving, and how difficult it was to complete.

#### Bug 1

I couldn't seem to find a website on my own that was talking about my issue. I remembered you had links on FOL to reliable sources that maybe I could search around in. Was trying to find something under functions since that is where the bug was located. I started reading the function section and the examples had the terms "if" and "return". I looked further into it, and it seems like a pretty useful tool. So, I wrote "if" and needed to add the situation in (). I thought to myself, "Well, what am I doing right now, I am trying to find if there is a piece in that drop zone... I must look in dropZone, and to search we have to use querySelector (not All because it's that particular drop zone, not all at once) since that is the action that searches for us. What am I searching for? I tried doing 'puzzle-pieces img', but that didn't work. I'm not looking for all pieces, I had to go lower to the actual item, which is just the image itself, 'img'. That seemed okay and it ended up working out. I logged it so it mentioned there was already something there, then used return to... well... return the piece.

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/return>

#### Bug 2

I first just tried a simple " return pzlPieces; " under function changelImageSet area thinking it would just reset everything when the function happens... not at all. I was struggling. I knew I

wanted to deal with pzlPieces since I needed to deal with each piece. I asked a classmate what their thoughts were, and they so happened to know what to do. They told me I had to look at each piece and move the pieces into a different container. So, appending the child over to a new area. The original container was called 'puzzle-pieces' in html/css, so I had resetPieces be that class and not was there before '.reset-pieces' then had each piece go into that container/class using appendChild.

- Classmate

## Final Thoughts

Overall, I did learn a lot, especially "if" and "return" those were good tools I came across. It was hard to find tools on my own, especially since it was working with what I just started this term, but I am very happy I got what I needed and did it right.