

Bugs

The first bug is that you can stack more than 1 piece in a drop zone, losing a piece until refresh. The other is the fact that when the background is swapped to a different image – a different level, the pieces from the previous puzzle remain on the board.

Here's an idea that might help solve the issue:

First of all, the pieces are actually placed on the board when a drag and drop function appends the images as children of the drop zone divs. This is essentially placing the images inside the divs. If I was able to reverse this designation, then when I use the changeBGImage function, it will “unappend” the children, taking them out of the drop zones, causing them to disappear.

How should I start?

I can start by creating a function that states when an object has a child. Something like `.hasChildNodes` might be able to help me by creating a `console.log` if the drop zones have children. If I can get the console log to work, I can then use an if statement to block certain functions from happening if there is a children element already occupying the drop zone.

Getting Stuck

I tried many different strategies, my best being to add an if statement, where if the parent div had child elements, it would log “has child nodes” in the console. When trying to implement this, I was always told that what I wrote was not a function. In order to work on solving this, I created a new file, where I made a div with a child and one without. I started from scratch, implementing the JavaScript and I managed to get it to work. However I was having trouble using `forEach` to select containers under 1 class, but when I put individual id names in I was able to get the function to work. Clicking on the container with the `<p>` child displayed the text in the chat log, where as clicking on the container without any children would display a different message with the use of an else statement. To work with multiple, simply setting the event `forEach` and the query selector to “all” would not work. After many attempts, I found that using “this”, instead of the variable name within the child length property would allow the console to recognize the child lengths of both divs. I then added a similar thing to the actual game. I started with a function that tells me via console the target and if it has children when you hover over it with the mouse.

Solving The First Bug

To solve the first bug, I had added an if statement that made it so if `children.length = 0`, then it appends the child. This means that when there are any children in a drop zone, the append child does not work for that drop zone.

The Second Bug

For the second bug, I know I have to remove child elements when the `changeBGImage` function runs, and then also have them reparented within the puzzle-pieces container.

Strategy

My strategy was to first unparent the images within the drop zones by removing them, and then reparenting them to the puzzle-pieces section.

Getting Closer

I used `draggedPiece.parentElement.removeChild(draggedPiece.);` However, this only works for the most recently placed piece that has been designated “draggedPiece.” Unfortunately, it seems that when a new piece gains that designation, the old ones lose it. Additionally, by inspecting the console, each drop zone has a child, but that cannot be targeted simply by using `dropZone.puzzlePieces` or `dropZone.childelement`.

Solving The Second Bug

When solving the second bug, I found that when trying to remove children of dropZones, it would have trouble. `dropZones` does in fact have child elements as seen through the sources, however, I couldn’t figure out how to call upon them without calling each individual drop zone. Through this I was able to remove the children and reparent them into the div that originally contained the puzzle pieces – now given the variable name `pieceZone`.

Conclusion

There are still bugs present, like the pieces staying the same in appearance even when a different puzzle is selected. Additionally, after pieces are reparented, they react to `dragStart`, but no functions afterwards. I could tell by looking at the console log.