Crushing programming bugs written plan

First issue:

When dragging puzzle pieces from the left to the right of the puzzle board, the pictures cannot overlap.

Step 1
1. Set the variable let puzzlePieces = document.querySelectorAll(".puzzle-pieces img") to grab all the pictures
2. Set the variable let draggedPiece; (for the last step appendChild sub-element use)

Step 2
set 3 events
start drag > move drag > drop pictures
1. add the drag event handling to the puzzle pieces
puzzlePieces.forEach(piece => piece.addEventListener("dragstart", handleStartDrag));

2. add the dragover AND the drop event handling to the drop zones
dropZones.forEach(zone => zone.addEventListener("dragover", handleDragOver));

3. add the drop event handling
dropZones.forEach(zone => zone.addEventListener("drop", handleDrop));

Step 3
Set different functions according to the above three events. In the last function, appendChild is required to paste the picture of the child element to the puzzle board.
Set the function of handleDrop, give him a (e) start as a function to avoid default, e.preventDefault();
Then, to solve the problem that the pictures cannot overlap, use the if, return function.

1. Use variables to assume draggedPiece = this

2. if (this.children.length >= 1) If the number of child elements exceeds one, the child elements will be returned.

3. Add this.appendChild(draggedPiece); to start the drop picture function.

Second issue:
The second bug was a partial issue with the drop zone when resetting/selecting a new puzzle piece. Those should also be removed/reset back to the drag zone so the player has a new board to drop on.

step 1
I put all the functionality in the function changeBGImage(), because when we click on the four buttons below the puzzle board, it switches to a different background, however, the four backgrounds need to switch to the correct image, the left One of

the puzzle pieces also needs to switch different pictures according to different backgrounds

1. Set the button variable
Select all four buttons below
theButtons = document. querySelectorAll("#buttonHolder img"),

2. Set the array
The top, bottom, left, and right sides of the four groups of pictures have different codes. Use the Array array to select the file names of the four groups of pictures, so that there is no need to do the function of the four groups.
imageNames = ['topLeft', 'topRight', 'bottomLeft', 'bottomRight'],

3. Set the variables of the for loop, when grabbing the name puzzle, all related pictures will be selected
puzzlePiece = document. querySelector(".puzzle-pieces"),

4. Set the variable dropZones to control the top, bottom, left, and right positions of the image in the dropZone
dropZones = document. querySelectorAll('. drop-zone'),

step 2
Set the button event activation function
theButtons.forEach(button => button.addEventListener("click", changeBGImage));

step 3
1. When the button below is clicked, the picture of the puzzle pieces on the left is also changed. Use the forEach loop to start the function, give (x, y) two values in the brackets, and finally use the variable puzzlePieces to grab four pictures
imageNames.forEach((pieces,index) => {
puzzlePieces[index].src = `images/${pieces+this.id}.jpg`
});

2. Enable the function of deleting pictures in the puzzle board
Use the forEach loop and use the node.firstChild interface to return, so when the following four buttons are clicked, the puzzlePieces inside will return to the left, and finally use the variable puzzlePiece to paste the child element appendChild, and the puzzle piece in the picture will be reset every time
dropZones. forEach((zones) => {
while(zones. firstChild) {
// let returnPieces = zones. removeChild(zones. firstChild);
puzzlePiece.appendChild((zones.firstChild));
}
})
console.log('Reset all puzzlePieces from puzzle board to left side');
}