**Plan for Crushing Bugs**

***Problem is***

*The image does not go back to the box-zone and when you change the background the puzzle is not following*.

**First problem is the background does**

1. The changeBGImage function handles altering the background image of the puzzle board and adjusting the images of puzzle pieces to match.
2. To resolve the problem, we modify the src attribute of each puzzle piece image to correspond with the new background image ID..

function changeBGImage() {

// the `` is a JavaScript template string. It tells the JS enging to evaluate the expression

// inside the braces - run that little bit of code. In this case it's just pulling the ID of the

// button we clicked on and putting it at the end of the image name (0, 1, 2, 3)

// and updating the background-image style of the puzzle board element.

const backgroundBox = this.id;

// bug fix #2 should go here. it's at most 3 lines of JS code.

puzzleBoard.style.backgroundImage = `url(images/backGround${backgroundBox}.jpg)`;

for (let i = 0; i < dropZones.length; i++) {

const zone = dropZones[i];

if (zone.firstChild) {

const piece = zone.firstChild;

puzzlePiecesContainer.appendChild(piece);

piece.classList.remove("dropped");

}

}

for (let i = 0; i < puzzlePieces.length; i++) {

const piece = puzzlePieces[i];

const originalPieceImage = piece.getAttribute('src');

const newPieceImage = originalPieceImage.replace(/\d/g, this.id);

piece.src = newPieceImage;}

}

**Second problem is that image should not be in same drop zone**

1. The handleDrop function manages the drop event when a puzzle piece is placed onto a drop zone.
2. To avoid dropping a piece onto a drop zone where it already exists, we verify whether the drop zone already has child elements. If it does, we take no action; otherwise, we add the dragged piece to the drop zone.

function handleStartDrag() {

console.log('started dragging this piece:', this);

// store a reference to the puzzle piece image that we're dragging

// so we can use it later and move it to a drop zone

draggedPiece = this;

}

function handleDragOver(e) {

e.preventDefault(); // e is shorthand for event

// this overrides the default dragover behaviour

console.log('dragged over me');

}

function handleDrop(e) {

e.preventDefault();

console.log('dropped something on me');

// bug fix #1 should go here, and it's at most 3 lines of JS code

//code for when you place the images, it is won't crysh eachother

if (this.children.length >= 1) {

}

else {

this.appendChild(draggedPiece);

}

// 1 to many event handling

// add event handling to each button in the collection of buttons, one at a time

theButtons.forEach(button => button.addEventListener("click", changeBGImage));

// add the drag event handling to the puzzle pieces

puzzlePieces.forEach(piece => piece.addEventListener("dragstart", handleStartDrag));

// add the dragover AND the drop event handling to the drop zones

dropZones.forEach(zone => zone.addEventListener("dragover", handleDragOver));

// add the drop event handling

dropZones.forEach(zone => zone.addEventListener("drop", handleDrop));

**Third problem reset button not defined and doesn't have function**

Making reset button work:

1. The resetPuzzle function is tasked with resetting the puzzle by removing all dropped pieces from the board and appending them back to the puzzle piece container.
2. To enable the reset button, we iterate through all puzzle pieces, remove their "dropped" class, detach them from their current position on the board, and reattach them to the puzzle piece container.

//reset

function resetPuzzle() {

for (let i = 0; i < puzzlePieces.length; i++) {

const piece = puzzlePieces[i];

piece.classList.remove("dropped");

piece.parentNode.removeChild(piece);

puzzlePiecesContainer.appendChild(piece);

}

}

const resetAll = document.getElementById("resetBut");

resetAll.addEventListener("click", resetPuzzle);

**Apapat Juntarattanakamol**