

Drag and DROP HUMAN ANATOMY

(Crushing Bugs - Explanation)

Bug A: Prevent Multiple Labels in One Drop Zone

Branch: des.ng.BugA

In the original file, players can drag more than one label into the same drop zone. This caused labels to stack on top of each other, which shouldn't happen. The goal was to make sure each drop zone only holds one label at a time.

To fix this, I used ***this.children.length > 0***

This checks if the drop zone already has something inside. If it does, the drop just stops. I learned this from MDN's explanation of the drop event, it lets you control what happens when something is dragged and released.

I used the formula below to fix Bug A and placed them on its designated section.

```
function drop(event) {
  event.preventDefault();

  this.classList.remove("highlight");

  if (this.children.length > 0) {
    return;
  }

  if (!currentDraggedElement) {
    return;
  }

  this.appendChild(currentDraggedElement);
  this.classList.add("dropped");
  currentDraggedElement = null;
}
```

** Reference:

children property – to check if a drop zone already has a label:

🔗 <https://developer.mozilla.org/en-US/docs/Web/API/Element/children>

drop event – defines behavior when something is dropped:

🔗 https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API

Bug B – Reset Button

Branch: des.ng.BugB

As we discussed in class, I learned from **MDN** that `appendChild()` doesn't just copy an element, it also **moves** it. That made it more clear for sending each label back to the starting

area when resetting the game. I also used `forEach()` to loop through each label and drop zone. I also had to clear any **leftover styles** like `.highlight` and `.dropped`. These are styles added during the game (like when hovering or after a drop). If they aren't removed, the game might look like something is still in a drop zone even when it's empty, which can confuse the user. I learned how to remove them using `classList.remove()`.

This fix makes sure that when the user clicks **Reset**:

- All labels move back to the original box
- Any leftover visual styles are cleared
- The game looks clean and is ready to play again — no need to refresh

I used the formula below to fix Bug B and placed them on its designated section.

```
function resetGame() {
  const labelBox = document.getElementById("label-box");

  labels.forEach(label => {
    labelBox.appendChild(label);
  });

  targetZones.forEach(zone => {
    zone.classList.remove("highlight");
    zone.classList.remove("dropped");
  });

  currentDraggedElement = null;
}
```

Reference:

[MDN – Node.appendChild\(\)](#)

[MDN – Array.prototype.forEach\(\)](#)

[MDN – Element.classList](#)

Bonus : Effects on Drag and Drop Zone

Branch: des.ng.bonus

I used drag-enter and drag-leave events to highlight the drop zone when a label is being dragged over it. Which gives the player feedback about where they can drop the label. I also used `classList.add()` and `classList.remove()` to change the styling by adding/removing CSS classes. These classes make sure there is color changes and an animation using CSS.

This effect improves the **user experience** by giving a good visual feedback when hovering or dropping. It makes the game feel more interactive especially to the end user.