DEBUGGING HOMEWORK – RESEARCH PLAN

Written and Created by Bernardo Macapagal

Here is a document outlining the requirements for the drag and drop puzzle game, listing out the bugs of the current files, and methods to fix the problems on the code.

REQUIREMENTS

User

- Picks any of the 4 puzzle pieces and drag to drop zone.
- Only drag and drop one piece into any given drop zone.
- Selects any of the 4 background images to change puzzle.
- Puzzle pieces return when new puzzle is selected.

System

- Knowledge of which piece the user is dragging.
- Which drop zone was the piece dropped to.
- Preventing duplication of dropped pieces in a zone.
- What puzzle has been clicked and it changing accordingly.

DEVELOPMENT

- Attach event listeners to BG images to know which image the user has clicked on.
- Create let variable with querySelectorAll and add event listeners for each loop.
- Call a function or event handler to execute the functionality above properly.
- Change the background image property with CSS and JS using a URL.
- Knowledge of which image is clicked using ‘this’ and grabbing the ID.
- Name images and ID’s to match up and also allow ID to swap image path.

FOUND BUGS

- Multiple puzzle pieces can be placed on one zone causing overlapping images.
- Puzzle pieces cannot be dragged back to the original zone or placement holder.
- Background changes but the puzzle pieces don’t change or reset accordingly.
- Reset button and the manual changing of puzzle pieces are not functional.
- Puzzle piece doesn’t replace and returns other piece when you add it to a zone.

SOLUTIONS

First Bug

- Description: Only one puzzle pieces needs to be dropped into one drop zone.
- Bug Fix: Added “if (!this.hasChildNodes())” and “else {console.log(‘Slot Occupied’);”

- Definition: hasChildNodes() – is a node interface that returns a boolean value indicates whether the given node has child nodes or not. This will return true is 'this' has children, otherwise it will return false.
- Usage: This method is to determine if an element is already placed in the drop zone, leading to it having a child as the 'puzzle piece' has been attached and it is a way to return true or false whether another puzzle piece can be placed.
- References:
  https://developer.mozilla.org/en-US/docs/Web/API/Node/hasChildNodes
  https://dom.spec.whatwg.org/#ref-for-dom-node-haschildnodes
  https://www.w3schools.com/jsref/met_node_haschildnodes.asp

Second Bug

- Description: The puzzle pieces don't reset when another puzzle board or background is selected and when the reset button is clicked. All puzzle pieces stay on the same position from where it was initially placed.
- Bug Fix: First, I added a variable declaration on the puzzle pieces placed on the board and another for the reset button. Second, I targeted the drop zones and the puzzle pieces, adding the innerHTML property and making them reset with '' in the changeBGImage function.

  Third, instead of using a connected function for the reset button, I just created a separate function that resets the entire page for an easier approach as the reset page has the same method view.

  Lastly, I added a slight eventListener on click to check if the reset button has been clicked to run the function of resetting the entire webpage.
- Definition: .getElementById – declares the variable with the ID directly instead of the finding it via the querySelector document method since it is also one element.

  .forEach – this is an array which targets each element in a group. An example of this is targeting all the puzzle pieces in the board.

  innerHTML – this is a property that gets the element inside the HTML directly.
  location.reload() – this is a method to refresh and reset the webpage efficiently.
  .onclick – this is a summarized eventListener that directly checks if the user clicks on the button or HTML element stated.
- Usage: This method targets the elements and adds functionality when the background puzzle images and the reset button is triggered or clicked to run the reset protocol for the drop zones and individual puzzle pieces.
- References:
  https://developer.mozilla.org/en-US/docs/Web/API/Element/innerHTML
  https://www.w3schools.com/jsref/prop_html_innerhtml.asp

https://www.javascripttutorial.net/javascript-dom/javascript-innerhtml/

https://developer.mozilla.org/en-US/docs/Web/API/Location/reload
https://www.freecodecamp.org/news/javascript-refresh-page-how-to-reload-a-page-in-js
https://www.w3schools.com/jsref/met_loc_reload.asp

https://developer.mozilla.org/en-US/docs/Web/API/Element/click_event
https://www.freecodecamp.org/news/html-button-onclick-javascript-click-event-tutorial/
https://www.w3schools.com/jsref/event_onclick.asp

Additional References

- https://developer.mozilla.org/en-US/docs/Web/API/Node/appendChild
- https://www.w3schools.com/jsref/met_node_appendchild.asp
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach
- https://www.w3schools.com/jsref/jsref_foreach.asp
- https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById
- https://www.w3schools.com/jsref/met_document_getelementbyid.asp