Interface Design

Audience:

- Each game is designed for all ages.

Interactive Features on each Game's Interface Page:

- Game logo cards when clicked, go to game page.
- Footer with link to readme file that contains directions, copyright info and other details that are specific to each game.

Interactive Features on each Game Page:

- The games themselves are interactive.
- Each game has either clicking or clicking and dragging aspects.
- Canvas game has background music and plays different sound on gameOver.
- Footer with link to readme file that contains directions, copyright info and other details that are specific to each game.
- Hover animations on each page for buttons, footer and other elements.
- Game info area in Tic-Tac-Toe game that updates as you play as well as score area in Memory game.

Interface Design

Layout Selection & Color Scheme:

Used similar layout as well as the same fonts, colors, background images, borders and other styling across each game's interface page and game pages as much as possible to keep things cohesive.

Interface Design

User Experience

The cohesive layout and design will enhance the user’s experience as well as the game content, because it’s clear that the games are all part of the same main page and I believe it will make people want to play all of them.

How to Play

Burger Stacking Game:

Drag and drop each burger element in the correct stacking order to make the perfect burger.

Canvas Game:

Use the arrows to navigate the burger through the maze of fries without bumping into them.

Tic-Tac-Toe Game:

Like classic tic-tac-toe but use fry Xs and onion-ring Os to get three in a row horizontally, vertically or diagonally.

Memory Game:

Click on a card and then click on another to find its mate. Try to find all the matching cards.

Coding Issues:

My most recent issues involved the Tic-Tac-Toe game. I initially thought this would be the easiest game to code but it ended up taking the most time and being the most difficult.

Some of the most time-consuming issues with this game include:

* I couldn’t get the X or O to show up when a square was clicked on.
I fixed this by adding classes for the X and O images in the css file and added the images to the classes as background images. In the js file I then added an event listener for whenever one of the game squares was clicked on, and if it was X’s turn, I added the X class and if it was O’s turn, I added O’s class.

* I also had trouble getting the game info to update in the different button areas in the html.
I fixed this by adding a counter variable in the js for each of the game info elements and then updated each of the corresponding id’s at the end of each turn.

* To prevent more plays from happening after a play has already been made, I added an “occupied” class with cursor: not-allowed; and pointer-events: none; to prevent any additional plays from being made.
Then to prevent more plays being made once X or O wins (even if there are still empty squares), I added the occupied class to all the squares whenever X or O wins the game.