Planning & Analysis Sheet

Design

For the home page, I chose to create a simple and minimal page to ensure an easy understanding of the site's structure.

To make the page responsive, I used Bootstrap, a framework that allows for visually appealing websites with built-in responsiveness.

I also opted for a black-and-white layout to enhance contrast and readability and kept the default font for easy reading.

Development Process

I began by creating the homepage, which features two cards to help users navigate the site, along with a navbar.

Next, I worked on the video games library using the RAWG API to fetch all the games and display them on the webpage.

I then implemented filtering features, including filtering by name, rating in ascending and descending order, Metacritic value, and release date in ascending and descending order.

I also added a spinner animation that is triggered when a filter is applied or a fetch request is made, along with the functionality to hide and show games.

Lastly, I implemented a feedback form that allows users to provide their first name, last name, email address, application rating using a star-based scale, experience feedback using an emoji scale, and additional comments.

Challenges Faced

I encountered challenges when filtering the data fetched from the API, as I initially focused on modifying the data instead of the URL.

Another challenge was implementing the satisfaction scale to show the right color and restrict checking only one radio input using a different approach from the default.

Javascript Interactivity

I used Javascript to enable the FetchGames function to create the nextPage and previousPage conditionally, based on the data results.

I also created an interaction to handle API fetch errors due to CORS policy, displaying an error message and prompting the user to refresh the page to resolve the issue.

The ShowGames() function populates the page with cards containing values from the fetched data. This function includes a conditional card population based on the presence of real values, accompanied by a message for unavailable values.

The page also features buttons for navigating to the next or previous page, and dropdown menus for filtering by rating, Metacritic score, and release date, all of which make requests to the updated URL using fetch functionality.

Additionally, I implemented a search functionality allowing users to search for games by name, using a button or the Enter key.

For the form page, I implemented validation for each input field without using the built-in 'required' keyword in HTML.

Api Key

api key = b259f29c517940719a7779c9084f878c