Design and Development Process Report

- Raghavendra Niteesh Ganugapati, Rahul Chanumolu, Vamsi Krishna Pasam

Introduction

Creating a web-based game is a complex task that involves a variety of skills, from programming languages to teamwork . Our team embarked on a five-week journey to develop a word guessing game. Along the way, we faced numerous challenges, overcame obstacles, and learned valuable lessons.

Week 1: Foundations

During the first week, we focused on building a solid foundation. Recognizing our limited experience in website development, we dove into the basics of JavaScript, HTML, and CSS. This week was crucial in ensuring that every team member had a good grasp of web development fundamentals.

Week 2: Setting the Blueprint

In Week 2, we started with a template from Coding Nepal and began enhancing its features. We divided responsibilities among team members, with each member responsible for curating questions, hints, and images for four different categories. We also started using GitHub, which came with its challenges, especially with Git Bash. However, our determination and collaboration helped us get comfortable with essential git commands.

Week 3: Building the Structure

Week 3 marked the transition from planning to actual development. We designed the main interface of our game, introducing two crucial HTML pages: home.html, displaying categories, and movie.html, showing questions related to the chosen category.

Week 4: Adding Logic and Style

The core functionality of the game came into play during Week 4 through two JavaScript files: word.js and script.js. The former contained questions, answers, and hints, while the latter handled game logic, including score calculation and question randomization. We also improved the user experience by integrating the Intro Js library and started refining the game's appearance with CSS.

Week 5: Testing and Deployment

We initiated an extensive testing phase to identify and fix any bugs in our codebase. Issues were found in both word.js and script.js and were promptly addressed. We implemented cookies to

ensure that Introjs loaded only for first-time players. After thorough testing, we were confident in the stability and functionality of our game, leading us to host it on GitHub Pages.

Reflections

What Worked? We successfully identified and resolved the UI issues in the initial template. We replaced intrusive alert messages with user-friendly notifications and ensured that the game provided a score even if the user answered incorrectly.

What Didn't Work? Our ambitious plans to incorporate confetti.js animations and potentially switch to advanced frameworks like ReactJS or Angular were hindered by time constraints and our learning curve.

Lessons Learned

1. User-Centric Design: Prioritizing a user-friendly and enjoyable experience is crucial.

2. Flexibility: Being adaptable in the face of challenges, especially when dealing with new technologies or unexpected issues, is essential.

3. Testing: Regularly testing the product is vital for delivering a polished and bug-free game.

4. Teamwork: Effective collaboration and clear communication are the foundations of a successful project.

5. Documentation: Maintaining clear documentation is helpful for troubleshooting and future improvements.

6. Collaboration: Tools like GitHub facilitate a collaborative environment, allowing for efficient version control and collective contributions.

7. Technical Knowledge: Understanding technical aspects, from programming languages like JavaScript to version control systems like Git, is of utmost importance.

In conclusion, our journey was characterized by continuous learning, teamwork, and problem-solving. We not only created a functional web-based game but also gained valuable skills and insights that will undoubtedly benefit us in future projects.