Final Report

Periodic Table Web Game

Version 1.0

Paul Detloff, Camden Smith, Noah Lee, Saif Alam

UAB Department of Computer Science CS/CSA499

11/30/2023

Abstract

The Periodic Table Memorization Game is a web-based application designed to assist students in memorizing the elements of the periodic table through an engaging and interactive experience. The game employs JavaScript for dynamic table generation, MongoDB for user authentication and score storage, and various frontend technologies for a seamless user interface.

The game features multiple modes, such as “Symbol to Name”, where users match symbols to element names, and a study page for a focused learning experience. The project emphasizes the integration of technology to enhance educational outcomes.

This report provides an overview of the development process, challenges faced, and the current state of the application. It further discusses the software design, tools utilized, dependencies, and assumptions made during the project. Evaluation includes a test plan, testing procedures, and results. Lastly, the report outlines potential future enhancements for the Periodic Table Memorization Game, offering insights into the project's evolution and suggesting areas for further development.

Table of Contents

1. Introduction

1.1 Purpose

Our project aims to provide STEM students with an engaging and educational interactive game focused on helping memorize the periodic table, and each element's characteristics. Recognizing the challenges students face in memorizing this essential scientific information, the game aims to turn education into an engaging experience.

1.2 Planned End Product

The planned product was a periodic table of elements that would change depending on if the answer was correct on the first try or if it was wrong. A leader board would be present to the left of the periodic table, with a zoomed in view of the element in question to the right of the periodic table. Under the zoomed in element, there would be a game mode selector and above the zoomed in element, there would be the question and controls. A study page would be linked in the menu bar at the top of the page, in addition to a login page, and user support. A csv file was going to be used to gather all the data for the periodic table of elements. On the study page, there would be a section that would suggest areas for improvement for the user, based on data collected of their game results. The web-based game would be hosted so that anyone with the dedicated domain name could access it.

1.3 Current End Product

The current product is a periodic table of elements that do change color depending on whether the user answered the question correctly on the first try. Solid green if correct or outlined red if incorrect. This is done using JavaScript. A leader board is present on the left and the question is displayed on the right along with the game mode selector. This is done using user data stored in a MongoDB database. A study page is accessible on the menu bar at the top of the page and the user can log-in to keep their high score stored in the database. A study page is available, with all the information that will be tested on in the game for the user to study. The periodic table data is hard coded into the program, as it has been years since the values have been changed. The web game must be locally hosted.

1.4 Differences

The reason that we did not include the adaptive study suggestions and the viewed in version of the element was the time constraint of the project and subpar planning for the final product. The group learned that all data that should be stored and used be written down and made clear before starting to code the final product. This would allow the team to better write the code to meet those requirements, rather than try to make it work later. Creating a more structured timeline for the development of future projects will greatly aid in these areas. We also learned that the free services, such as GitHub website hosting, cannot handle python code, and can only handle HTML/CSS/JS.

2. Technical Documentation

2.1 Software Design

The software design of our periodic table memorization game is centered around HTML, JavaScript, CSS, MongoDB, and Python.

2.2 Tools used

The project primarily utilizes the following technologies:

- Frontend
  - **HTML**: Used as the base for creating the game interface.
  - **CSS**: Employed for styling the HTML elements and enhancing the visual appeal.
  - **JavaScript** (JS): Utilized to create dynamic tables, implement game functions, and handle user interactions.
- Backend:
  - **Python**: The main backend language used to handle server-side logic.
  - **Flask**: A Python web framework used for server-side scripting and routing.
- Database:
  - **MongoDB**: Employed as the database system to store and retrieve game-related data.

2.2.1 Code Overview

The HTML file **index.html** serves as the foundation for the game interface. It contains a dynamic table representing the periodic table. The table cells are populated with element data, and the game logic is implemented using JavaScript. The interface includes input fields for user answers and displays the score. Key HTML Components:

- <div id="periodic_table">: Container for the dynamic periodic table.
- <div id="question">: Container for displaying the current question to the user.
- <input id="answer" />: Input field for the user to input their answer.
- <div id="score_counter">: Display area for the user's current score.

-JavaScript (script block in index.html)

The JavaScript code is responsible for creating the dynamic periodic table, implementing game logic, and handling user interactions. It dynamically generates HTML elements based on the provided element data and controls the flow of the game.
Key JavaScript Functions:

- shuffleArray(array): Shuffles the elements of an array to randomize the order.
- isElement(element): A filter function to remove blank elements from the element list.
- displayQuestion(): Displays questions based on the selected game mode.
- checkAnswer(giveUp): Checks the user's answer against the correct answer and updates the score.
- set_score(): Updates and displays the user's score.

-Python (main.py)

The main backend script written in Python using the Flask framework. It handles routing, user sessions, and interactions with the MongoDB database.
Key Python Functions:

- account_create: Creates a new user account in the database.
- user_checker: Checks the validity of a user's credentials.
- get_user: Retrieves user data from the database.
- username_checker: Checks the availability of a username during account creation.
- leaderboard: Retrieves and returns the leaderboard data from the database.
- check_score: Updates and checks the user's score in the database.

2.3 Dependencies/Assumptions

•The project assumes a modern web browser with JavaScript enabled for optimal performance.
•MongoDB must be installed and configured for the database functionality.
•Python with Flask is required for server-side scripting.

3. Evaluation

3.1 Test plan

The following test plan was implemented to ensure the functionality and reliability of the game:

- **User Authentication Tests**: Ensure that user login and registration processes work correctly.
- **Game Logic Tests**: Validate that the game responds accurately to user input and updates the dynamic periodic table.
- **Database Interaction Tests:** Confirm that user scores are properly stored and retrieved from the MongoDB database.

3.2 Testing conducted

We conducted testing using different browsers and to ensure responsiveness of the log in page. Additionally, unit tests were employed to check the functionality of individual components step by step, with each gamemode addition and with each component addition.

3.3 Result of Testing

All components of the periodic table memorization game passed the test plan, demonstrating functionality and reliability.

4. Future Work

4.1 Outlook

- **Multiplayer Mode** - Implement a multiplayer mode, allowing users to compete against each other in real time
- **Enhanced Study Features:** Expand the study page with more interactive and educational content about each element.
- **Mobile Application**: Develop a mobile application version for on-the-go learning

4.2 Next Steps

If we were to continue working on this project, the next steps would include:

- Enhancing the visual appeal of the overall web application
- Enhance user profiles with personalized statistics and progress tracking
- Introduce badges, achievements, and other gamification elements to motivate users
- Implement a community forum or discussion board for users to share tips and challenges

5. Conclusion

This technical documentation provides an overview of the project's history, methods, technologies used, and analysis. Throughout the project, many lessons were learned. The Importance of teamwork was the most prevalent factor in this project. An area of improvement that could have been implemented was clear planning of the project, and set sections to work on. Most of the project was done by whoever can manage which task at a given time, leading some group members to be the workhorse. Challenges in technology integration were addressed; Debugging and constant testing was an absolute must. The methodology used was agile programming; partly due to the nature of the groups schedule, and also because of an unclear plan of the final product. In the future, if we were to do this project again, we would begin by laying out a clear plan for the final product, and then implement set tasks for group members, as well as set time frames for each part, and consistent times for weekly meetings.