S2308-CPSC-8710

Foundation of Software Engineering

MEMORUNES

CHASE GUYNUP

PEDRAM MOHAJERANSARI

SIDDHARTH KANDLAKUNTA

ABSTRACT:

MemoRune is an engaging memory game that invites players to match pairs of ancient runic symbols. Explore four exciting game modes, including Freeplay, VS, Countdown, and Sprint. Customize the game by choosing grid sizes of 4, 6, or 8, and set timers to challenge your memory skills.

INTRODUCTION:

MemoRune is an intuitive memory game where players are presented with a grid of runic symbols, and the objective is to match pairs of identical runes.

There are four different modes to choose from, they are -

Free Play Mode

Unleash your memory skills in our Freeplay Mode, where the fun never ends. You can choose your grid size 4, 6, or 8 to match your skill level. Experience a continuous stream of random runic boards, with a timer that counts up to mark your journey's duration. Complete one board, and another instantly appears, ensuring non-stop entertainment.

VS Mode

In the fast-paced VS Mode, players compete head-to-head within a set time limit. Take turns choosing cards and scoring points for matches. Grid sizes of 4, 6, or 8 are yours to choose. You can set the timer to your preferred minutes and seconds and even give yourself and your opponents custom player names. The player with the most points when the clock runs out claims victory. It's a high-energy memory challenge like no other!

Countdown Mode

In Countdown Mode, the clock is your adversary. Your mission: match the runic cards and complete the board before time runs out. Choose your grid size of 4, 6, or 8 and set the timer in minutes and seconds to fit your challenge level. It's a thrilling test of speed and memory!

Sprint Mode

In Sprint Mode, it's a race against the clock. Your mission: clear the runic board as swiftly as possible. Choose your grid size of 4, 6, or 8 and prove your speed and memory skills in this adrenaline-pumping challenge!

Technology Stack Used:

HTML:

HTML (Hypertext Markup Language) is the backbone of the project, providing the structure and content of web pages.

CSS:

CSS (Cascading Style Sheets) is used for styling and layout, ensuring a visually appealing and responsive user interface.

JavaScript:

JavaScript is the scripting language that adds interactivity and dynamic behavior to the game. It's responsible for game logic and user interactions.

Setup and Deployment Instructions:

Clone this repository: git clone https://github.com/SiddharthKandlakunta/memory-game.git.

Go to the memory-game folder.

Open index.html file to open the game in your preferred browser.

You can also directly open the game at: https://8710-memory-game.pages.dev/.

Reflections:

Reflection on the Design and Development of MemoRune Memory Game

The creation of the MemoRune Memory Game was an exciting journey that involved various design and development challenges, as well as moments of success. The objective of the game was to present players with a fun and educational memory challenge using runic symbols. In this reflection, We will discuss the design and development process, challenges encountered, what worked well, what didn't, and the valuable lessons learned along the way.

Game Design

The initial design phase was crucial for outlining the game's objectives, features, and overall user experience. We wanted MemoRune to be intuitive, engaging, and visually appealing. The decision to use runic symbols as the game's theme was an interesting choice as we wanted to do a viking themed game, adding cultural and historical depth to the experience. We wanted to add more depth to the classic Memory game so we decided to add modes to the game. The four different game modes—Freeplay, VS, Countdown, and Sprint—offered a variety of

challenges to cater to different player preferences, making the game more versatile and appealing.

Development Challenges

One of the primary challenges during development was ensuring that the game functioned flawlessly in both design and functionality. Matching runic symbols required careful implementation to ensure a seamless and enjoyable user experience. At first we thought about addiding different jpegs as images for our runes, this turned out to be hard because we decided to do multiple grid sizes. So, we went for svg images, where we could change the color of the runes in code, which made it easier for us as we did not have to search for different jpegs for each grid size. We encountered difficulties in managing the various game modes, especially in synchronizing the timer across the different modes. Debugging and testing became a crucial part of the development process, as it was essential to catch and fix any issues that might disrupt the gameplay.

What Worked Well

The choice of using HTML, CSS, and JavaScript as the technology stack proved to be effective. We were deciding between react or this, but in the end we chose HTML, CSS, and JavaScript as all of us are well versed on them. HTML provided a solid structure for the game's web page, while CSS allowed us to create an attractive and responsive design. JavaScript was the engine that drove the game, bringing it to life with interactivity and dynamic behavior. The development team's collaboration and dedication were also essential to the project's success. Regular team meetings, code reviews, and clear communication helped keep the development process on track. We had a team meet every monday where we discussed the things we needed to do that week and assigned works based on our strengths, we also discussed about our previous week's work.

The diversity of game modes provided a great deal of replayability and catered to a wide range of players. Freeplay Mode allowed players to explore and learn at their own pace, while VS Mode introduced a competitive element. Countdown Mode and Sprint Mode challenged players to test their speed and memory skills. Customization features, such as grid size and timer settings, allowed players to tailor the game to their preferences.

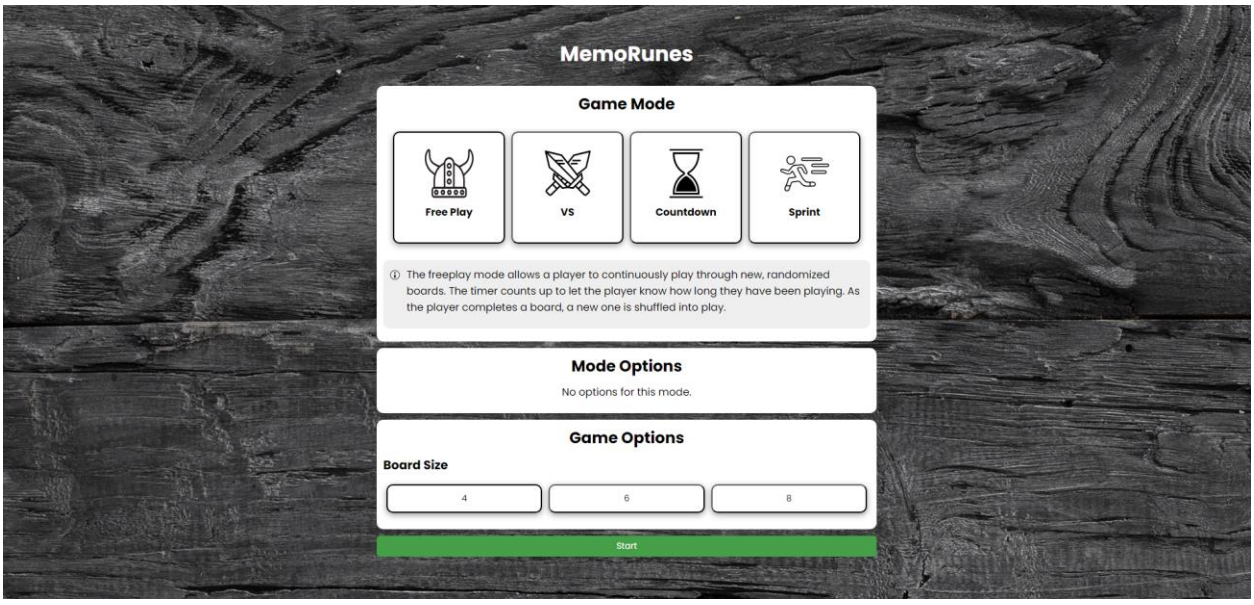
What Didn't Work

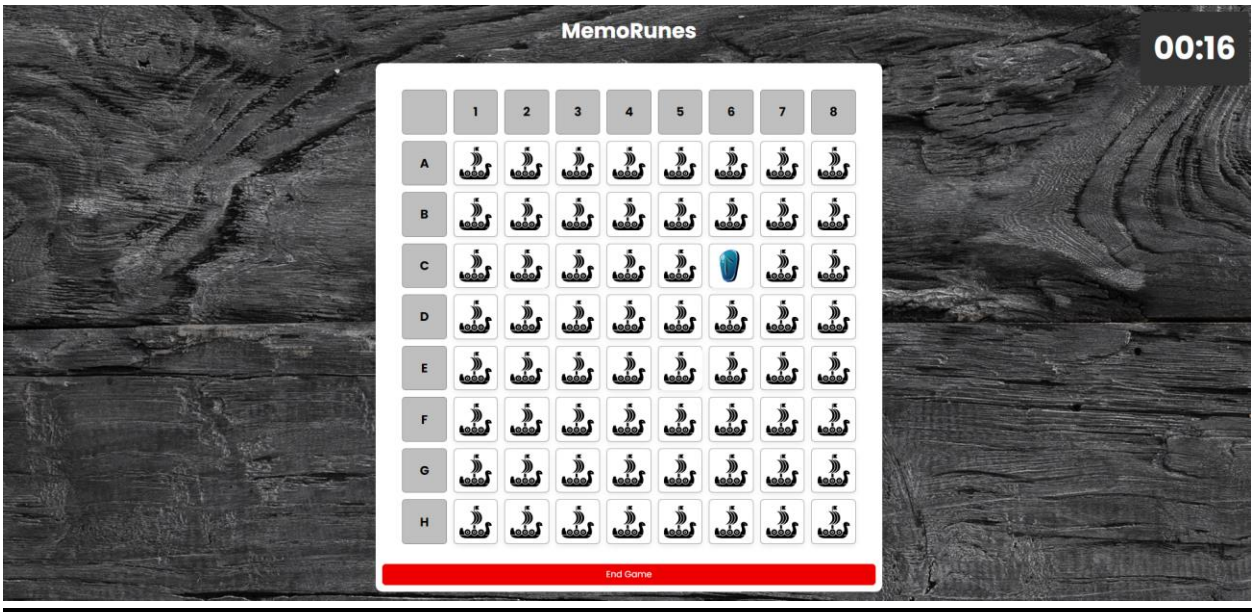
While the overall development process was successful, there were moments where we faced technical issues. The synchronization of timers across different game modes proved to be a complex task, and we encountered some unexpected bugs. Additionally, ensuring cross-browser compatibility required extra attention, as certain features behaved differently on various web browsers.

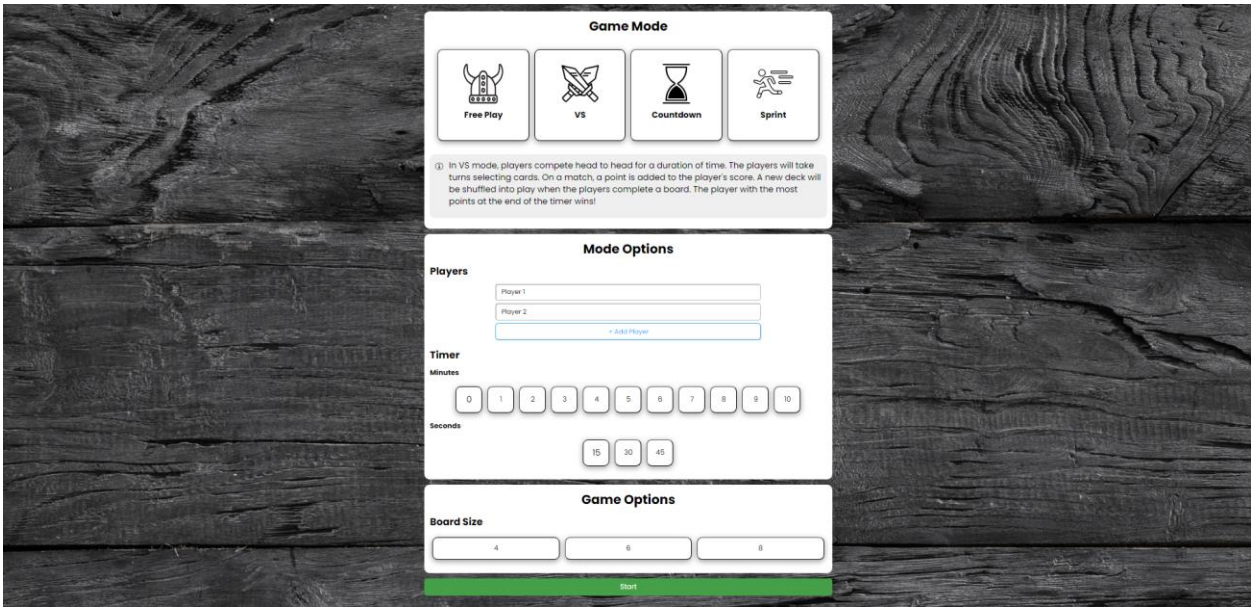
Lessons Learned

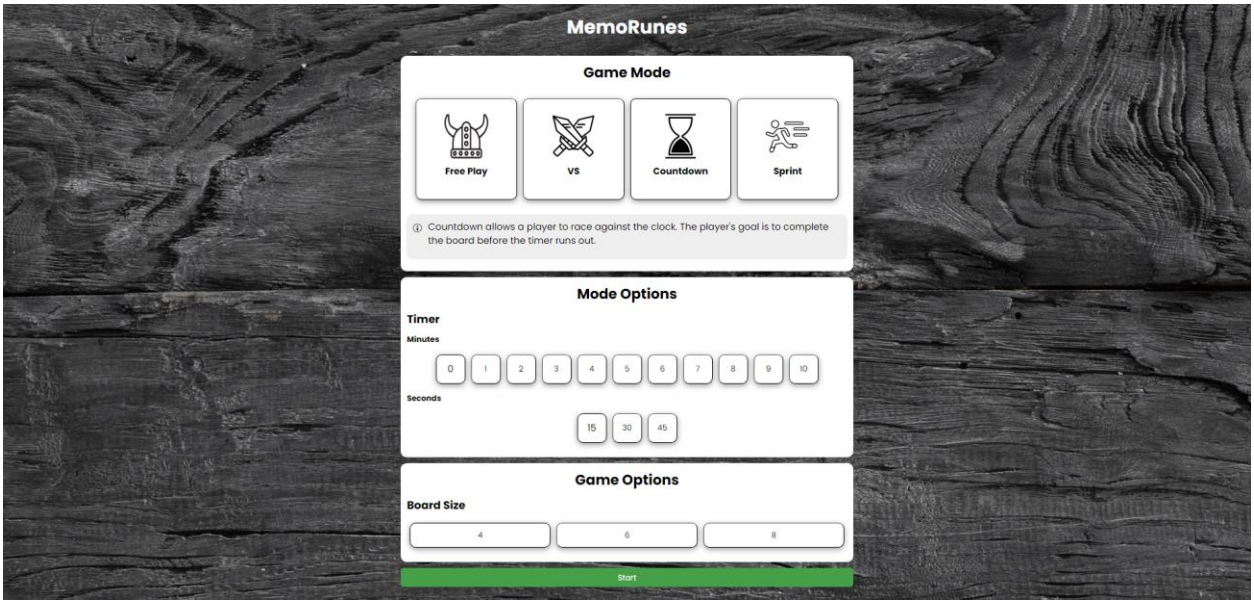
Developing MemoRune Memory Game taught us several valuable lessons. Firstly, thorough testing and debugging are essential to identify and resolve issues promptly. Additionally, it's important to plan for compatibility across different web browsers from the early stages of development. Continuous communication within the development team is crucial for maintaining a cohesive and efficient workflow.

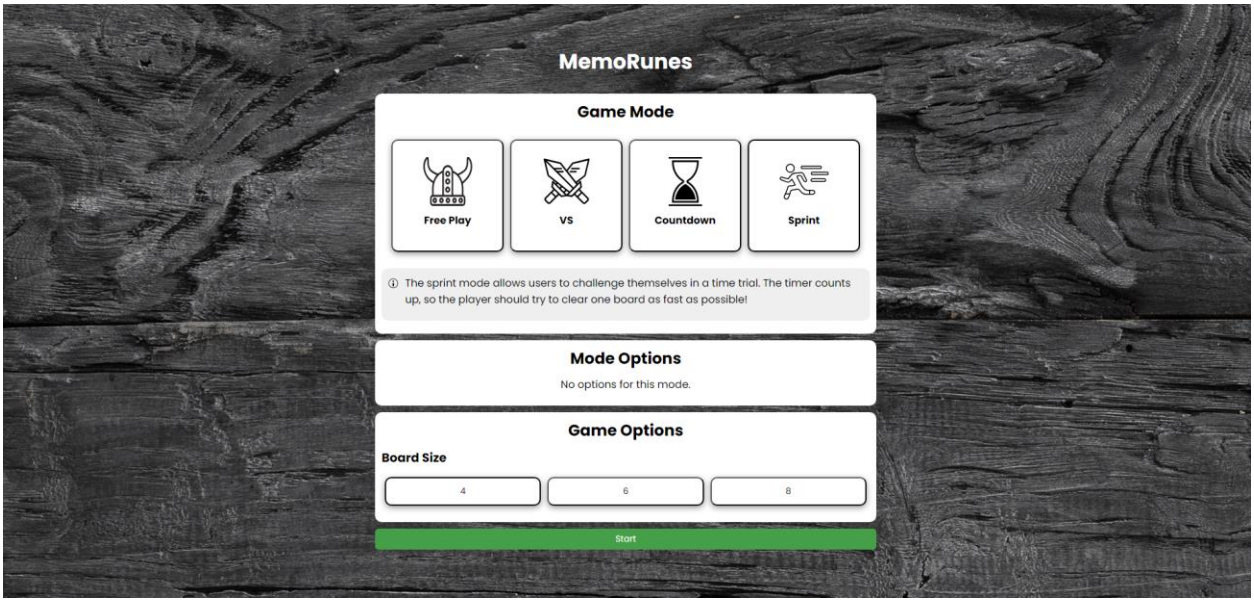
OUTPUT:






Game Mode
Free Play
VS
Countdown
Sprint
In VS mode, players compete head to head for a duration of time. The players will take turns selecting cards. On a match, a point is added to the player's score. A new deck will be shuffled into play when the players complete a board. The player with the most points at the end of the timer wins!
Mode Options
Players
Player 1
Player 2
+ Add Player
Timer
Minutes
0 1 2 3 4 5 6 7 8 9 10
Seconds
15 30 45
Game Options
Board Size
4 6 8
Start


MemoRunes
Game Mode
Free Play
VS
Countdown
Sprint
Countdown allows a player to race against the clock. The player's goal is to complete the board before the timer runs out.
Mode Options
Timer
Minutes
0 1 2 3 4 5 6 7 8 9 10
Seconds
15 30 45
Game Options
Board Size
4 6 8
Start



CONCLUSION:

In conclusion, the design and development of MemoRune Memory Game was a rewarding experience. The blend of technology, creative design, and thoughtful game modes resulted in a captivating and educational memory challenge. The challenges we faced during development served as valuable learning experiences, ultimately leading to the successful creation of a game that we are proud to share with the world.