Proiect IBM Summer School

WebDev

Team – Bytes

1. **Team Details**

|  |  |  |  |
| --- | --- | --- | --- |
| Team – Bytes | | | |
| Lead | Dev | Dev | Ops |
| Lavric Cosmin | Chelaru Alexandru | Bobeică Teodor | Ciubuc Vlad |

LEAD: Lavric Cosmin

Lavric Cosmin was chosen as the LEAD due to his professionalism and strong organizational skills. He has a comprehensive understanding of the project requirements and is adept at coordinating efforts among team members. Cosmin is capable of presenting complex information clearly and confidently, making him an ideal candidate to lead the presentation of the homework. His ability to defer technical questions to the appropriate team members ensures that the presentation remains accurate and thorough.

DEV: Chelaru Alexandru

Chelaru Alexandru was selected as the DEV because of his experience and proficiency in software development. He has a talent of writing clean, optimized, and efficient code. Alexandru is skilled in problem-solving and implementing requests accurately. His technical expertise and ability to collaborate effectively with other developers make him a valuable asset to the team.

DEV: Bobeică Teodor

Bobeică Teodor is another DEV chosen for his strong programming skills and attention to detail. Teodor excels in developing robust code and is known for his innovative solutions to complex problems. His ability to work closely with Alexandru ensures that the development process is smooth and that any challenges are promptly addressed. Teodor’s contribution to the project is critical in maintaining high standards of code quality and functionality.

OPS: Ciubuc Vlad

Ciubuc Vlad is an OPS known for his creativity and passion for design. He excels in managing and optimizing operational processes while bringing a unique, designed-focused perspective to problem-solving. His innovative approach ensures efficient workflows and enhances the overall user experience. Vlad is friendly and sociable, always striving to ensure the well-being of the entire team.

1. **Changelog**

**The Ops Manual was written by the entire team, and revised by lead, Lavric Cosmin.**

|  |  |  |
| --- | --- | --- |
| CHANGELOG | | |
| DATE – [DATA CURSULUI] | LIST OF CHANGES | AUTHOR |
| [10/07/24] | Implementation for MapBase based in a 10x10 table with the implementation of PlayerController for the red square placed in the middle of the table built in .js and .css file.  I contributed to the creation of the documentation. | Bobeică Teodor |
| [10/07/24] | Added a new font in order to improve the design of the website.  Import of the font in the index.css file  Design changes to the buttons so that they resemble an actual keypad  Added animations to the text and the moving cube so that it looks more like an vintage arcade game.  I contributed to the creation of the documentation. | Lavric Cosmin |
| [10/07/24] | Change the position of the red cell based on the player controller actions.Set an upper and lower bound for how much the player X and Y coordinates can change(never less than 0, never more than 9. I contributed to the creation of the documentation | Chelaru Alexandru |
| [12/07/24] |  | Ciubuc Vlad |

1. **Deep Dive**

For the first course on [10/07/24], our team focused on several enhancements to our React project, which are detailed below:

Implementation of MapBase Component

Author: Bobeică Teodor

We began by introducing a new component named MapBase, which features a 10x10 table where each cell maintains equal width and height, ensuring they are perfect squares. This component replaces the previously used spinning React element. In this implementation, the middle cell of the table is initially highlighted in red. The component was built using both JavaScript and CSS files to ensure that the structure and styling were well-defined. This foundational change was crucial as it set the stage for further interactive elements in our project.

Player Controller and Boundary Implementation

Author: Chelaru Alexandru

The functionality of the MapBase component was extended to include player controller actions, allowing the red cell to be moved around the table based on user inputs. This dynamic interaction was complemented by setting strict upper and lower bounds for the player's coordinates, ensuring they stay within the grid's limits (from 0 to 9). This implementation required careful handling of state changes and event listeners in React to maintain a smooth and responsive control system. Despite the complexity, this feature was successfully integrated, providing a foundational interactive element for future enhancements.

Design Enhancements

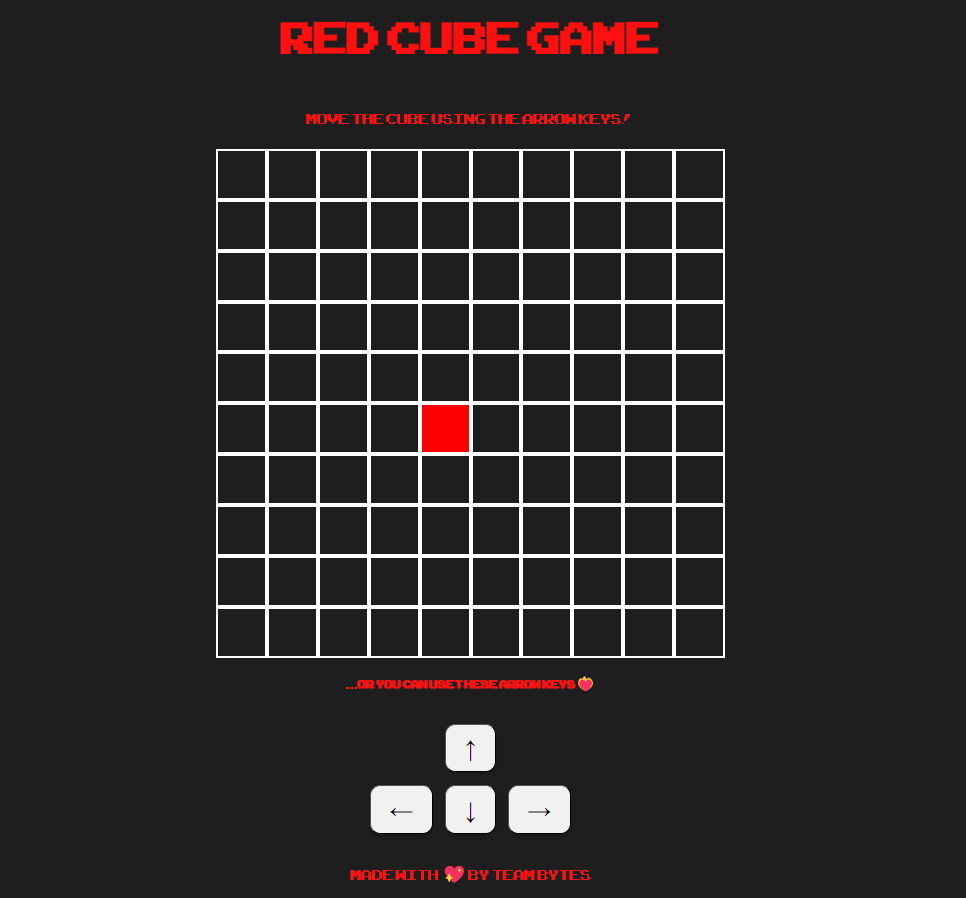
Author: Lavric Cosmin

To improve the overall design aesthetics of the website, a new font was added and imported into the index.css file. Additionally, significant design changes were made to the buttons, giving them the appearance of a classic keypad, enhancing the user interface to be more intuitive and visually appealing. Furthermore, animations were added to the text and the moving cube, aiming to evoke a vintage arcade game feel. These animations not only improved the visual appeal but also contributed to a more engaging user experience.

Dependencies and Challenges

During this session, no additional dependencies were introduced beyond the standard React library and CSS. One major challenge we faced was ensuring the cells remained perfectly square across various screen sizes, which required precise CSS adjustments and testing. Additionally, implementing smooth and responsive player controls in React posed its own set of difficulties, particularly in managing state changes and event listeners efficiently. However, through collaborative effort and iterative testing, these challenges were overcome.

Here is a screenshot showcasing the current state of the project, highlighting the MapBase component with the movable red cell in the middle of the grid:



After that we made the border of the 10x10 grid by adding spikes with collider on the edges so that the player could not pass through them, also we set the spikes to be pointed in an appropriate direction (toward the center of the square) for a more pleasant appearance.

In addition, we added the green square that represents the NPC which moves randomly on the X and Y axis every second through the moveNPC() function.

A screenshot of a video game

Description automatically generated