Blockbuzz

Blockbuzz is a stress-buster web game for everyone! Match the blocks in a column and earn points in this booming puzzle game.

Match the blocks with minimum clicks to win the maximum number of points. There are a total of 25 levels in the game.

Rules

1. Match three colors to clear them and earn points. Only sets of three will clear, even if more touch.
2. Click on a row and the entire row will move to the top of the column.
3. After clearing blocks, remaining blocks in the row shift left.
4. Fewer clicks earn more points.

Built using simple HTML, CSS and JavaScript with Confetti library!

> To play, visit https://rishav-karanjit.github.io/CPSC8710/ or download repo and run index.html

Design and Development process:

We thought of a simple yet fun game for this project. This game is somewhat inspired by candy crush as it is based in mathcing entities and clearing them for earning points. We decided to make it aesthetic just by using HTML, CSS and JavaScript. The main process we followed was brainstorming ideas and selecting the best one, assigning work amongnst the developers and refining the app to make it simple yet beautiful.

Challenges We Faced:

- Idea Generation:
  - Conceptualizing a game that was both engaging for players and feasible for our team to develop within the given timeframe.
  - We faced challenge to create a unique gaming experience which would stand out, yet remain simple to understand and play.
  - We were able to overcome these both of the challenges after completion of the project
- Technical Limitations:
  - We faced the challenge of devising an algorithm for dynamic level generation. The learning curve for creating such an algorithm was steep, and given our tight deadline of 1.5 months, it became unfeasible.
  - We did have animation in the game but this was a repeated animation and it would be better if we had various kind of animation in the game. If there was more time we would work in this feature
- Team Dynamics:
  - We did try doing Agile and scrum way of software development however coordinating among team members and ensuring that everyone was on the same page was a challenge in an academic setting.
  - Given the time constraints we have difficulty in managing the development process, and testing the game
- Iterative Refinement:
  - We thought to do code review as well however, this was difficult as every team mate had there own experience
  - We were not able to continous test and refine the game. We would like to have few groups of users play the game and give reviews so that we could do iterative refinement.

What worked

- The game was straight forward yet was challenging.

- Using just HTML, CSS, and JavaScript, we were able to create a visually pleasing game interface.
- Incorporating the Confetti library added a fun and festive element to the game.
- By setting a clear objective for players (matching colors and earning points), we created a focused gaming experience.
- The choice to allow players to click on a row and have it move to the top was intuitive and added a strategic element to the game.

What didn't work

We were not able to dynamically create levels as we might had to create a level generation algorithm which had a high learning curve and was difficult to complete in 1.5 months.

The lessons we learned are

- Teamwork. Distributing work amongst team members and completing parts of the project in time was learning experience.
- Design. Working on ideas, finding the best design for an application and improving it throughout the development cycle.
- The Value of Simplicity. Simple, clear design can often be more effective than a complex one.
- While integrating third-party libraries like the Confetti library, we learned the importance of understanding the capabilities and limitations of external tools.
- User-Centric Design: Designing with the user in mind is paramount.
- Time Management: We learned the importance of setting achievable goals within a given timeframe.

Third party library used

- jQuery
- tsparticles-confetti