

Phase III: Web Game Devs

Charles Bostwick, Jade Pearl, Ada Truong, Robyn Cohen, and Holland Brawner

Department of Computer Science, University of Maryland Global Campus

Professor Shanna Nevarez

April 26, 2023

## Introduction

In Phase III of our project, we transitioned from developing a basic functional prototype to refining and enhancing the product to align more closely with our initial design and vision. We also added a game (Minesweeper), as we are on track for meeting our deadlines and had established that additional games/features could be incorporated in the last few weeks. This phase involved fine-tuning the core aspects of our project to ensure that the final deliverable met and exceeded our expectations in terms of functionality, aesthetics, and user experience.

Our primary objectives during this phase were:

- **Integrating Design with Functionality:** We adapted our core game mechanics to match our Figma designs, focusing on optimizing user experience and interface.
- **Enhancing Technical Framework:** We evolved our HTML framework to support advanced web operations and more accurately reflect our visual style.
- **Streamlining Deployment:** We refined our use of GitHub to facilitate smoother live deployments, setting the stage for efficient, ongoing updates and maintenance.
- **Incorporating Minesweeper:** We created a fully functional version of Minesweeper that matches the website's visual/aesthetic theming and is accessible on desktop screens.

The following sections detail our milestones, their statuses, the problems we've encountered along the way, and our strategies for overcoming them.

# Milestones

## Key Milestones / Timeline of Events

Task	Member(s)	Deadline
Finalize and review documentation: <ul style="list-style-type: none"><li>- Software Requirements Specification (SRS)</li><li>- Game Design Document (GDD)</li><li>- MVP (Minimum Viable Product) Strategy</li></ul>	Charles Bostwick	March 27th
Generate Figma frames for different screen sizes (desktop, tablet, and mobile).	Ada Truong, Robyn LaMontagne	End of week 4
Create the website's HTML skeleton with a focus on accessibility.	Holland Brawner, Ada Truong, Robyn LaMontagne	End of week 5
Apply CSS styling based on Figma frames, focusing on reusability and clean styles.	Holland Brawner, Robyn LaMontagne, Ada Truong	End of week 6
Organize all programming-related terms and definitions in a Godot-friendly file.	Jade Pearl	End of week 5
Develop core mechanics/prototype for the Hangman game on Godot. Add sprites and other visual assets.	Jade Pearl, Charles Bostwick	End of week 5
Style the hangman game using the Figma frames as a guide. Add sound design.	Jade Pearl	End of week 6
Corrections/bug fixes and finishing touches on the Hangman Game	Jade Pearl	End of week 7
Generate tests to ensure the functionality of the Hangman game.	Team	End of week 7
Export the Godot game and embed it within the website.	Charles Bostwick	End of week 8
Deploy the website to GitHub Pages.	Charles Bostwick	End of week 8

- **Phase III Milestone(s):**

- Integrated Figma stylizations and fleshed out CSS content for the Meet the Team page
- Tested functionality of Hangman game/started refactoring code to work with Figma stylization
- Further refactored code for GitHub deployment
- Initial export of base game to test web deployment

## Status

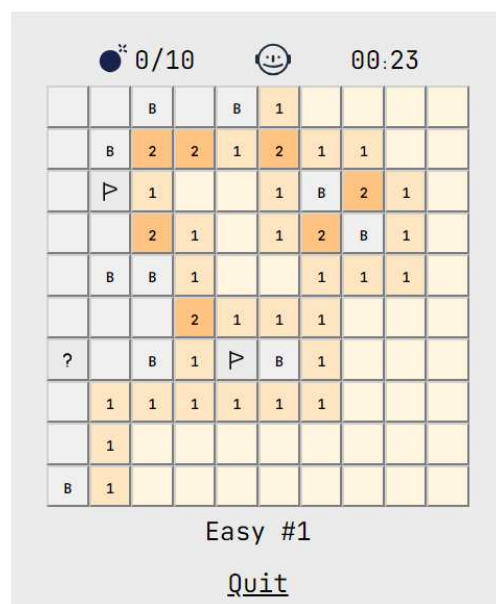
### Hangman Game:

- **Figma Adaptation:** We took the Figma layout and started incorporating key design features.
- **Audio Design:** We started working on sound integration.

### Minesweeper Game:

- **Functional game mechanics:** The mine placement on the grid is randomly generated upon the first click. The user can click uncleared cells to reveal the number of adjacent mines or a mine. The game ends automatically once all cells are cleared or when a mine is discovered.
  - Mid-gameplay in classic mode ('B' signifies the presence of a mine--this feature will be removed before the final release, and only aids in testing until that point):

## Minesweeper



- **Multiple modes:** The user may choose to play the game in classic or customized mode. Classic mode progresses the user through a set of seven increasingly difficult Minesweeper grids (determined by grid size and mine density) with each successful round. Once the user completes the final grid, the user may return to the main menu. In customized mode, a grid and its mines are generated according to user input.

# Minesweeper

## Classic

Starts with a beginner grid and mine density. Increases in difficulty with each successful round.

Start game

## Customized

Choose your own grid dimensions and mine density.

Length \_\_\_\_\_  
Width \_\_\_\_\_  
Mines \_\_\_\_\_

Start game

- Main menu:
- Classic mode difficulty progression:

```
const classicDifficultyProgression = [  
  {width: 10, length: 10, mines: 10, level: 'Easy #1'},  
  {width: 9, length: 9, mines: 9, level: 'Easy #2'},  
  {width: 8, length: 8, mines: 10, level: 'Easy #3'},  
  {width: 16, length: 16, mines: 40, level: 'Intermediate #1'},  
  {width: 15, length: 15, mines: 40, level: 'Intermediate #2'},  
  {width: 14, length: 14, mines: 40, level: 'Intermediate #3'},  
  {width: 16, length: 30, mines: 99, level: 'Expert'},  
]
```

- **Input validation/user feedback:** In order to play in customized mode, the user must enter the grid dimensions and the number of mines. Validation logic prevents the user from entering negative grid dimensions or a mine quantity that meets or exceeds the number of cells available. Appropriate error messages are displayed to alert the user to required changes.
  - Following attempt to enter customized mode without providing dimensions and mine quantity:

## Customized

Choose your own grid dimensions and mine density.

Width must be between 1 and 30 cells.      Length must be between 1 and 30 cells.      Number of mines must be greater than 0 and less than the total number of cells.

Length \_\_\_\_\_  
Width \_\_\_\_\_  
Mines \_\_\_\_\_

Start game

- Following entry of incompatible mine quantity and dimensions:

## Customized

Choose your own grid dimensions and mine density.

Number of mines must be greater than 0 and less than the total number of cells.

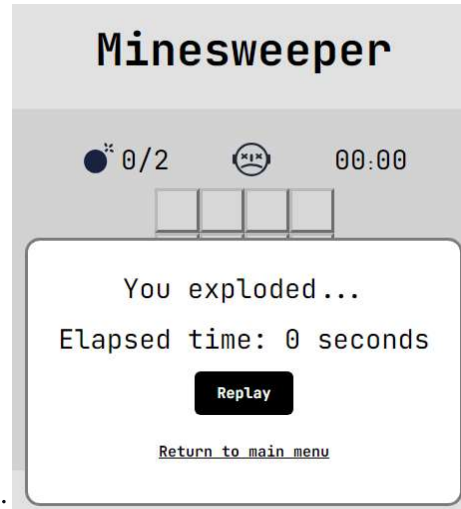
Length	<input type="text" value="4"/>
Width	<input type="text" value="2"/>
Mines	<input type="text" value="8"/>

Start game

- **Successive games:** The user may reset the grid (undoing all clicks and generating new mine positions) at any time, regardless of the game mode. In customized mode, the user may replay the same grid configuration as many times as they desire.



- After successfully clearing a customized grid:



- After triggering a mine:
- **Instruction:** Instructions are displayed to teach users how to play the game. A minesweeper guide is linked.
  - Instruction screen, which appears after the main menu and before each round of games:

### Instructions

You will be presented with a grid of squares. Most of the squares will be safe, but some will conceal deadly mines.

If you click on a safe square, the number of adjacent mines will be revealed (anywhere from 1 to 8). When there are no adjacent mines, surrounding squares will automatically be revealed until mines are encountered. But if you click on a square containing a hidden mine--boom!

The game ends when you have either accidentally discovered a mine, or cleared all safe squares.

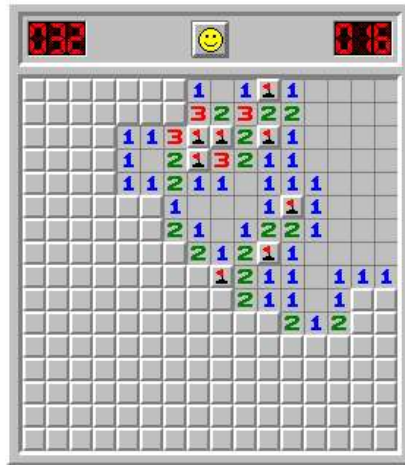
### Tips + Other Info

- The first click is always safe. There will never be a mine hiding under the first cell you choose.
- Not loving the current grid? Click on the happy emoji above the board to generate new mine placements and start over.
- Right-click to toggle between three states: uncleared, flagged (highly suspicious for mine activity), and question mark (somewhat suspicious). Although these states have no bearing on the final outcome, they can help you deduce which cells have mines.
- If you use the flag feature, keep an eye on the total number of flagged mines. This can help you assess risk and track your progress.
- Watch the time! The time only starts after the first click. The faster you can clear a grid, the better!
- For more information and to view basic patterns, visit this [minesweeper guide](#).

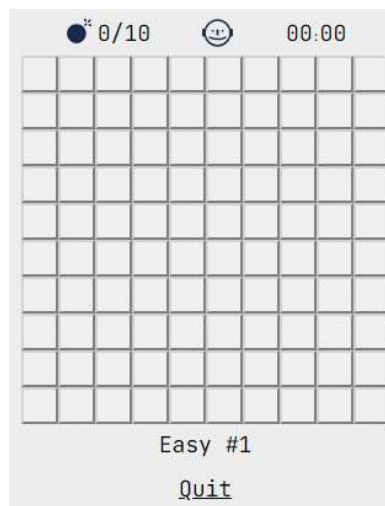
Start

- **Visuals:** Placement and appearance of various features (e.g., flagged mines, board regeneration button, timer) are loosely adapted from the original Windows minesweeper game.





- Original game layout:



- Our interpretation:
- **Accessibility:** On desktop, all buttons and text are an appropriate size. Although the game can still be played on tablet and mobile screens, the size of the larger grids (combined with the minimum required click area) makes scrolling and zooming almost unavoidable.

**Website:** We consolidated team efforts via GitHub to update and finalize layout per our Figma. We merged team inputs and populated changes into the Meet the Team page and CSS stylesheets to display accurately (see example below).



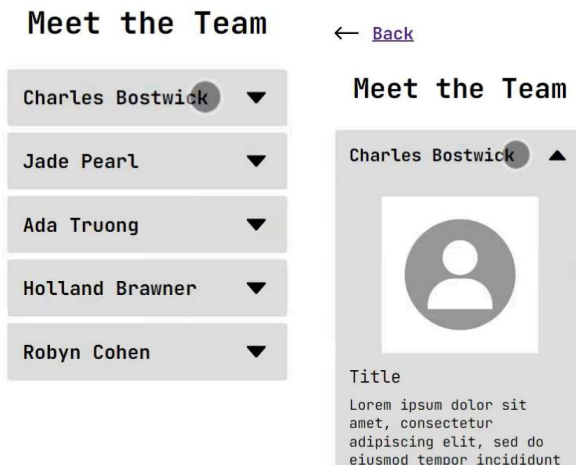
- Before styling:





- After styling (desktop):

← [Back](#)



- After styling (mobile):

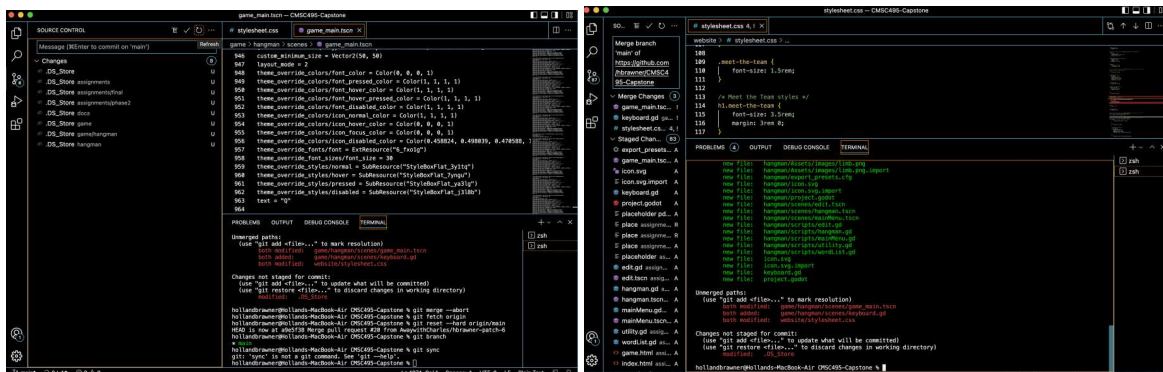
# Problems Encountered

## GitHub issues/deployment Challenges:

During this phase, our team faced significant new GitHub challenges primarily due to synchronization and branching issues that complicated our pull request (PR) process. We inadvertently created multiple branches that needed to be synchronized with each other, leading to discrepancies in the codebase and obstructing our workflow. This problem was exacerbated by the differing operating systems used by team members, which added a layer of complexity to our version control process. Luckily, between Mac, Linux, and Windows, all of the commanders for the terminal were the same, so it also eased the process.

We tackled these challenges head-on through a troubleshooting session on Discord. We utilized Git commands such as `git status` to assess the state of our repositories and `git rebase` to realign our branches with the main project timeline, remove the other branches, and set the main branch.

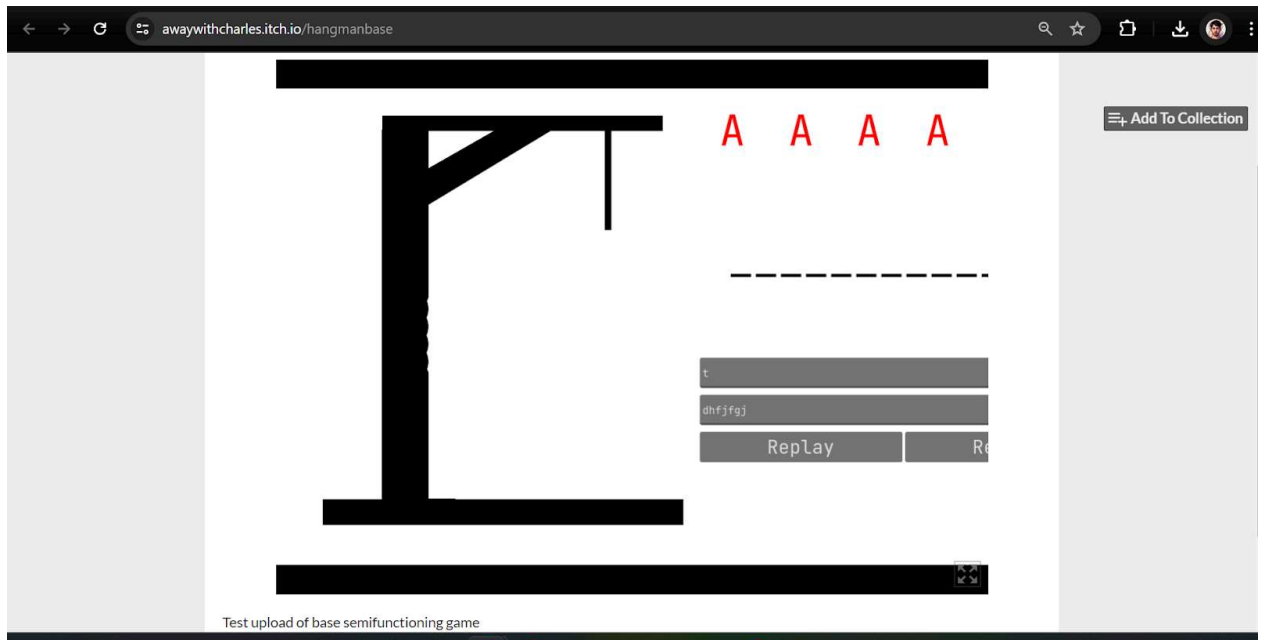
Furthermore, we documented our session with screenshots and detailed notes to aid in future troubleshooting and to provide a reference that could help mitigate similar issues. These documents were mainly focused on illustrating the differences in managing Git repositories on Mac versus Windows platforms, thereby tailoring our guidance to accommodate various systems used by the team (we found that the sync button is located in different areas).



## Game Dictionary Integration/Changes:

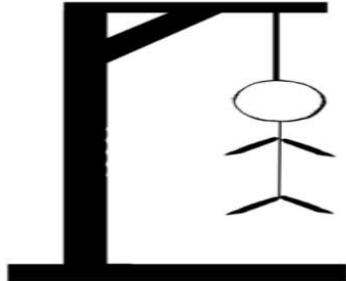
We were running into issues with the dictionary populating correctly with a single word on a new user's PC. After some code changes it was working. Additionally, we exported it to web formatting, and it displayed the scenes from the editor and not the actual game that would be displayed in the editor. We are working to troubleshoot and integrate/export it correctly and will continue the testing process.

## Game testing/Bugs:



Above is the uploaded draft web version and the game stopped working once exported and showed screens from the engine that should not be visible during gameplay. We kept trying to further implement it but it just was not exporting correctly. We decided to recycle the code from Godot and use the same methods and functionality to develop it into a Javascript game instead and directly embed it into the browser since Godot was making it more difficult than it should be.

Hangman



\_ \_ \_ \_ u r r e \_ \_ \_



Guess the full word

Guess Word

Restart Game

## Progress

- **Hangman game:** The updated Javascript version is now functioning mostly correctly. Additional error handling needs to be implemented to align it with the initial goals of the project.
- **Minesweeper game:** It is fully functional (see Status section for additional details).
- Added updated license for Creative commons/noncommercial to MD

```
CMSC495-Capstone > license.md > # License > ## Creative Commons Attribution-NonCommercial (CC BY-NC) > ### Notices:
1  # License
2
3  ## Creative Commons Attribution-NonCommercial (CC BY-NC)
4
5  This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit [http://
6  creativecommons.org/licenses/by-nc/4.0/](http://creativecommons.org/licenses/by-nc/4.0/) or send a letter to Creative Commons, PO Box 1866, Mountain View,
7  CA 94042, USA.
8
9  ### You are free to:
10
11  - **Share** – copy and redistribute the material in any medium or format
12  - **Adapt** – remix, transform, and build upon the material
13
14  The licensor cannot revoke these freedoms as long as you follow the license terms.
15
16  ### Under the following terms:
17
18  - **Attribution** – You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable
19  manner, but not in any way that suggests the licensor endorses you or your use.
20  - **NonCommercial** – You may not use the material for commercial purposes.
21  - **No additional restrictions** – You may not apply legal terms or technological measures that legally restrict others from doing anything the license
22  permits.
23
24  ### Notices:
25
26  You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or
27  limitation.
28
29  No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity,
30  privacy, or moral rights may limit how you use the material.
```

- Added navigation bar, footer, and Documents page to the website

## Document Revisions

- Add additional items to the plans/design phase documents
- Configured documents for final submission
- Added table of contents to CSS stylesheet

```
Table of Contents
-----
1. Import Fonts
2. Base Styles
  2.1 Root Variables
  2.2 Reset Styles
  2.3 Typography
  2.4 Header Styles
3. Layout Styles
  3.1 General Layout
  3.2 Back Button Styling
4. Interactive Elements
  4.1 Links and Buttons
  4.2 CTA Buttons
5. Homepage Highlights
6. Responsive Design Adjustments
7. Footer Styles
*/

/* ----- */
/* ----- 1. Import Fonts ----- */
/* ----- */

@import url('https://fonts.googleapis.com/css2?family=JetBrains+Mono:ital,wght@0,100..800;1,100..800&dis

/* ----- */
/* ----- 2. Base Styles ----- */
/* ----- */

/* 2.1 Root Variables */
:root {
  --gray-background: #DDD; /* sets a light grey background color for elements like toggle sections */
  --border-radius: 0.4rem; /* standard border radius for rounded corners on elements */
}

MS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
MAL
\\Data\\Capstone\\CMSC495-Capstone>
```

## Conclusion

We switched from Godot to Javascript for Hangman game development, as learning the entire Godot game engine to deploy, embed, and cross-site host the game was no longer feasible for the timeline set. The game was rapidly developed in Javascript and met all the requirements set forth in the initial planning phase. We found several bugs in the new version during testing, but overall, the development process has been streamlined and the game's functionality improved as a direct result of the Godot-Javascript change. We also developed a Minesweeper game in Javascript, which is primarily adapted to desktop but remains functional in tablet and mobile views. We plan to further develop and run tests during the last week to ensure its reliability, as well as refine the styling. As we continued to use GitHub for version control, we ran into some issues but have learned and gotten past the issues which is natural in any team environment, especially when using unfamiliar and new technologies. Overall, we are on track in spite of minor adjustments required to deliver a completed project within the established scope.