Phase II: 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 19, 2023

Introduction

Phase II of our project marks a pivotal development stage where we transition from conceptual design to tangible, functional deliverables. This phase is particularly significant as we have developed foundational working components, including the core structure and mechanics/functionality.

The primary goal of this phase was to build and refine the basic architecture of our game and website infrastructure. This involves several key objectives:

- Developing the core game mechanics that will drive user engagement.
- Establishing an HTML framework to support our web operations.
- Pushing our initial designs to GitHub for live deployment.

These critical milestones lay the groundwork for our project's subsequent enhancements and expansions.

Our iterative approach focuses on achieving a solid and scalable foundation to build more complex features and functionalities. We aim to ensure that each component performs its intended function and interacts flawlessly with other system elements to create a cohesive and enjoyable user experience. This phase is crucial for setting a precedent for the quality and reliability of our final product.

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

Milestones

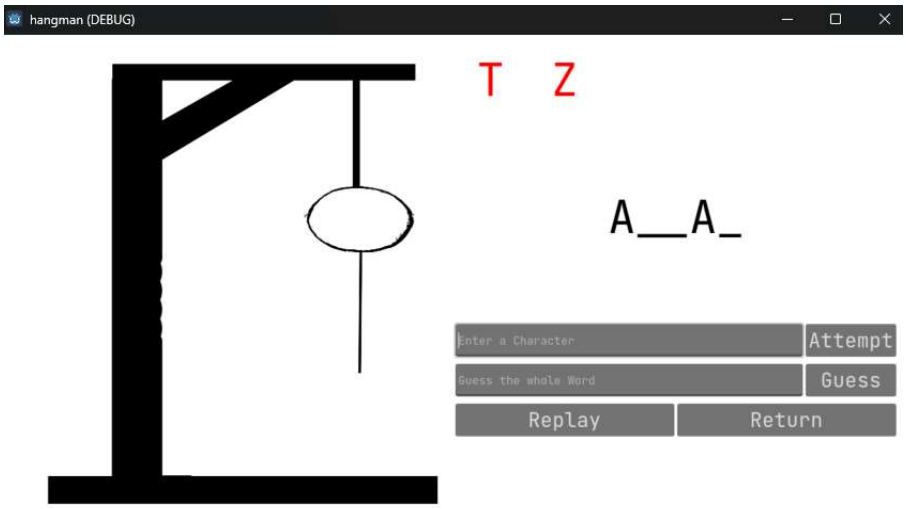
Key Milestones / Timeline of Events

Task	Member(s)	Deadline
Finalize and review documentation: - Software Requirements Specification (SRS) - Game Design Document (GDD) - MVP (Minimum Viable Product) Strategy	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

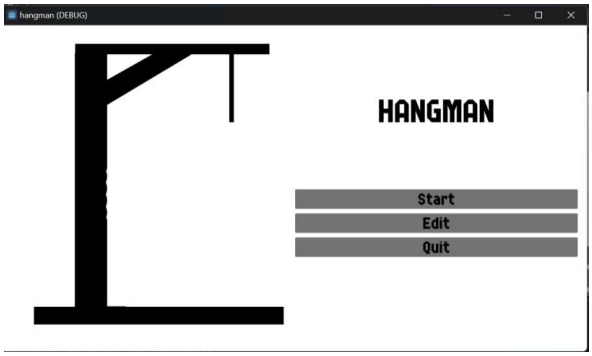
- **Phase II Milestone(s):**
    - Develop core mechanics
    - Create website HTML skeleton for deployment
    - Organize all words to the definitions.txt file for godot
    - Push skeleton site to GitHub to ensure it deploys

- **Status:**
    - **Core Mechanics:** Successfully developed and integrated the core mechanics for the game, utilizing Godot's scripting language. This included implementing essential game functions and the interaction framework within *.tscn and *.gd files.

Working skeleton of Hangman game



Game start screen/main menu

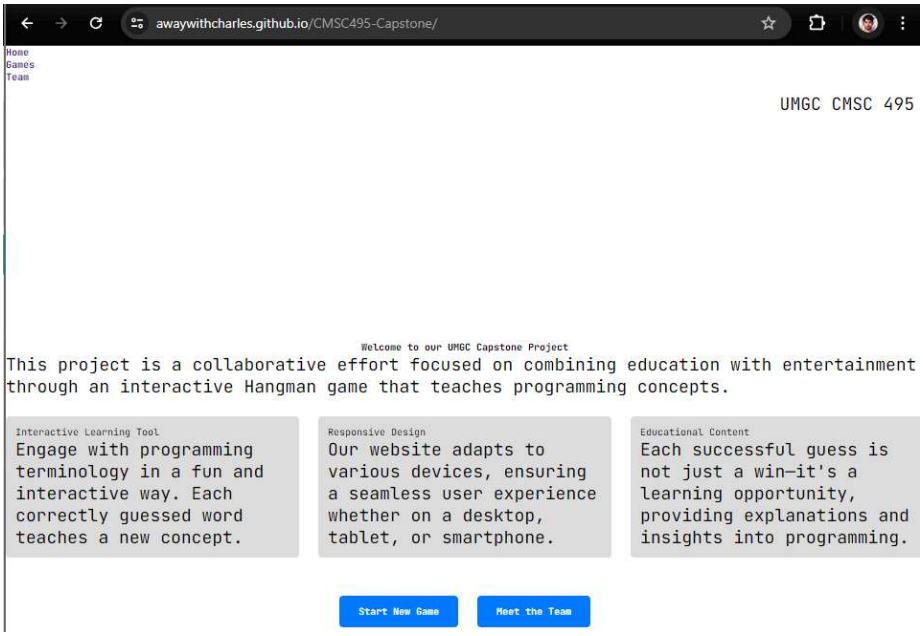


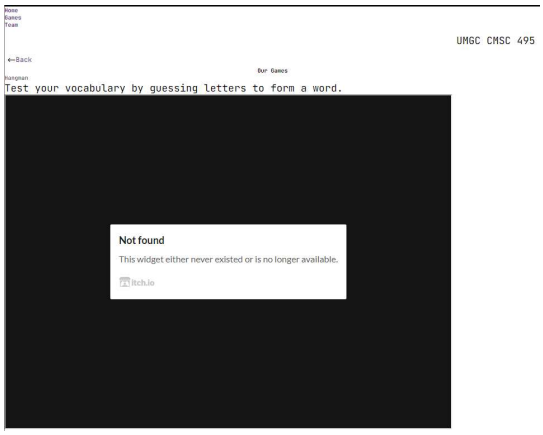
Game edit screen to add/remove words for the game

- **Status cont:**
    - **Website Skeleton:**

      Constructed the foundational HTML structure and accompanying backend functionalities/integration for GitHub pages. The deployment included setting up a main-live branch specifically for GitHub Pages to manage and streamline production updates.

Homepage refactored to index.html and deployed to github.





    - **Documentation:** Enhanced code comprehensibility through comprehensive documentation and comments, facilitating more manageable maintenance and scalability.

Problems Encountered

Deployment Challenges:

Initial attempts to deploy the site encountered failures, requiring multiple refactoring sessions. Eventually, a dedicated 'main-live' branch was established for the live deployment on GitHub Pages.

Dictionary Integration:

The automated generation of the dictionary.txt file for the Hangman game faced bugs. The script wordList.gd was supposed to automate this but failed in execution. Recognizing the urgency of this issue, we temporarily resolved it by manually creating the file. Rest assured, a permanent fix is our top priority for the next phase.

```gdscript
func _process(delta: float) -> void:
    var histogram: Dictionary = {}
    for word in words:
        if histogram.has(word):
            histogram[word] += 1
        else:
            histogram[word] = 1
    words = histogram.keys()

# Function to randomly select a word from the 'words' array.
func set_chosen_word_randomly():
    var words_count := words.size()
    if words_count == 0:
        chosen_word = ""
    var random_word_index := randi() % words_count
    chosen_word = words[random_word_index]

# Function to populate the 'words' array from a dictionary file.
func populate_words():
    var dictionary: FileAccess = Utility.open_dictionary()
    while dictionary.get_position() < dictionary.get_length():
        words.append(dictionary.get_line().strip_edges())

# Function to set 'chosen_word' randomly and convert it to uppercase.
func set_chosen_word() -> void:
    var random_index = randi() % words.size()
    chosen_word = words[random_index].to_upper()
```

```gdscript
        chosen_word = ""
    var random_word_index := randi() % words_count
    chosen_word = words[random_word_index]

func populate_words():
    var dictionary: FileAccess = Utility.open_dictionary()
    while dictionary.get_position() < dictionary.get_length():
        words.append(dictionary.get_line().strip_edges())

func set_chosen_word() -> void:
    var random_index = randi() % words.size()
    chosen_word = words[random_index].to_upper()
```

Progress

Significant milestones were achieved in this phase:

- **Game Skeleton:**
  Successfully developed interactive game modules for a functioning hangman game.

- **Nav Bar Integration / GitHub refactoring:**
  Integrated a basic nav bar into the webpage layout, enhancing user experience and interface fluidity. Refactored code for GitHub page deployment rather than local deployment.

Document Revisions

During this phase, we updated our project documentation to reflect significant changes in the codebase and strategic adjustments. These revisions were necessary to maintain the accuracy and relevance of our documentation as the project evolved. Key updates included:

1. **Enhanced Source Code Comments:**

   Each file within our repository now includes detailed comments explaining the purpose and functionality of the code.

2. **Deployment Strategy Updates:**

   As we encountered challenges with GitHub Pages deployment, we took a proactive approach and documented our troubleshooting process and solutions. This includes detailed descriptions of the refactoring sessions and the establishment of a stable deployment pipeline through the 'main-live' branch.

3. **Error Handling Documentation:**

   We added a new section focused on error handling strategies, particularly around the issues encountered with the wordList.gd script. This section outlines the steps taken to diagnose and temporarily resolve the problem, setting the stage for a permanent fix.

4. **Technical Specifications:**

   The technical specifications have been updated to include the latest architectural changes, such as the integration of the navigation bar and the modifications made to the game's core mechanics. These updates provide a clear and current overview of the system's structure and functionalities.

Conclusion

Phase II successfully laid a foundation for us to iterate on further and allow us to enhance during the subsequent phases and final product. The core mechanics are not only in place but are also primed for expansion, and the initial deployment obstacles have been effectively navigated and documented for future reference.

Future Directions:

- **Gameplay Refinement:** Our next steps will involve refining the gameplay mechanics based on user feedback and initial testing results, as well as integrating our figma designs to enhance user experience and interaction.
- **User Interaction Enhancement:** As mentioned above, aligning with our Figma designs, we will enhance our games' visual and interactive aspects, improving accessibility and user engagement. This includes optimizing the UI for various devices and screen sizes to ensure a consistent user experience. This isn't only for the game but also for our web development portion too.
- **Permanent Solutions to Existing Issues:** We will prioritize addressing the temporary fixes implemented in this phase, such as manually creating the dictionary.txt file. We can update the code to initially provide a prefilled dictionary.txt file. We are committed to developing a solution that automates this process effectively.

By advancing these areas, we aim to meet and exceed the expectations outlined in our project's initial scope, ensuring that we deliver a user-friendly product. The insights gained from this phase will move us toward successfully implementing Phase III, where our focus will shift to enhanced UI/UX and testing.