Plans and Specifications

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

Department of Computer Science, University of Maryland Global Campus

Professor Shanna Nevarez

March 26, 2024

Introduction

This document presents the Web Game Devs project by Charles Bostwick, Jade Pearl, Ada Truong, Robyn Cohen, and Holland Brawner. Our project aims to link theoretical knowledge with practical application through a responsive educational website featuring "Programming Hangman," an interactive game developed with Godot 4.2 to teach programming terms. By integrating HTML, CSS, and JavaScript with Godot, we strive to offer an informative and accessible experience. Our planning, clear project scope, and collaboration using agile methodologies and tools like Trello and Discord lay the foundation for success. This details our software requirements, development process, milestones, and assumptions, reflecting our commitment to academic achievement and contributing to the computer science community.

Scope

The project scope encompasses a responsive website in HTML, CSS, and Javascript, deployed and hosted on GitHub. The scope also includes creating an interactive web embedded game using Godot 4.2. The game will be similar to Hangman while focusing on programming concepts, and implementing a user-friendly interface for visitors. Moreover, the scope of the project will not include a database, user authentication, API calls, or server/cloud networking and implementation. Finally, depending on the rate at which the team can accomplish their tasks, they have leeway to develop additional games/features to later deploy on the site.

Requirements

- The product shall be a website featuring an interactive Hangman game centered on programming concepts/terms.
- The website shall be responsive and accessible, across different screen sizes, user abilities, etc.
- The website shall have a page that features information about the team members.
- This project's documentation shall include a Software Requirement Specifications (SRS) document for the web design portion (following the IEEE template).
- This project's documentation shall include a Game Development Document (GDD) for the game development.
- This project's documentation shall include a Minimum Viable Product (MVP) sheet.

Process

- This project shall utilize Software Development Life Cycle processes.
- This project shall use Discord to facilitate communication between team members.
- This project shall leverage Trello as a kanban board to employ agile methodologies.
- The project will use GitHub for version and source control.

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

Assumptions

- The user will have access to modern web browsers, such as Chrome, Firefox, Safari, and Edge, hosted on GitHub Pages.
- The user will have access to mobile, tablets or web devices to utilize aforementioned modern web browsers.
- The user will not need access to any specialized hardware.
- The user can read.
- The user can understand english.
- The user will have a basic level of knowledge relevant to programming related words.

Software Requirements Specification (SRS) for the Web Game Devs Project

1. Introduction

1.1 Purpose

This document specifies the software requirements for the Web Game Devs project. The project aims to develop a responsive website hosted on GitHub Pages. It will feature an interactive Hangman game that incorporates programming-related words. The project emphasizes collaboration and agile methodologies, utilizing Trello for project management and Discord for team communication.

1.2 Document Conventions

This document adheres to the IEEE SRS standard format to ensure clarity and consistency.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for project team members and stakeholders. It guides the development of the website and the game that will be embedded in the site. It emphasizes the tools and methodologies employed for effective collaboration.

1.4 Project Scope

The project scope encompasses developing a responsive website, creating an interactive Hangman game using Godot 4.2, focusing on programming concepts, and implementing a user-friendly interface for visitors.

2. Overall Description

2.1 Project Perspective

This website is a standalone project designed to be hosted on GitHub Pages. It is intended to function across various web browsers and devices, providing a responsive and interactive user experience. The project showcases the team's development skills, integrating an educational Hangman game. It leverages Discord for communication and Trello as a Kanban board to adopt agile practices.

2.2 Project Functions

- A navigation bar for seamless access to the website's sections.
- An interactive Hangman game centered around programming terminology.
- A "Meet the Team" section with member profiles and information.
- Utilization of Discord for continuous team communication.
- Adoption of Trello for agile project management and task tracking.
- There is a possibility of adding additional pages for additional games, time permitting.

2.3 User Classes and Characteristics

- Casual web visitors interested in playing the Hangman game.

- Team members and stakeholders are reviewing the progress of the project.

2.4 Operating Environment

The website is accessible through modern web browsers like Chrome, Firefox, Safari, and Edge, hosted on GitHub Pages. The development and project management tools include Godot 4.2, Discord, and Trello.

2.5 Design and Implementation Constraints

- The project must be browser-compatible and responsive.
- Development for the Game will use Godot 4.2 and GDScript, with HTML5 for web export/embed.
- The team will use Discord for communication
- The team will use Trello for project management, adhering to agile methodologies.
- The team will use GitHub for version and source control.

2.6 User Documentation

The site will provide instructions for gameplay and navigation, along with detailed documentation on GitHub for development insights.

3. System Features

3.1 Website Layout and Design

3.1.1 Description and Priority

High priority. The website's layout and design are crucial for user engagement and accessibility.

3.1.2 Functional Requirements

- FR1: The website must feature a responsive design, ensuring usability across devices and screen sizes.
- FR2: The navigation buttons at the top of the page must allow users to smoothly scroll to the desired section and/or page.
- FR3: The "Meet the Team" section must display team members' names, roles, and a short bio (a photo is optional).
- FR4: The game section should embed the Hangman game, allowing users to interact with it directly on the website. Initially, a placeholder image could be used.

3.2 Website Content

3.2.1 Description and Priority

Medium priority. Content should be engaging and informative, reflecting the team's objectives and personalities.

3.2.2 Functional Requirements

- FR5: Content must be clearly written and understandable, catering to a broad audience.
- FR6: The website should include a section describing the project's goals and the technologies used in development.

4. External Interface Requirements

4.1 User Interfaces

- The website should have a visually appealing interface, color scheme, and layout that enhances readability and user experience.

4.2 Hardware Interfaces

- No hardware interfaces are required as the project is web-based.

4.3 Software Interfaces

- Web browsers: Chrome, Firefox, Safari, Edge
- GitHub Pages for hosting
- Development Tools: Godot Engine 4.2, VS Code
- Languages: GDScript, HTML5, CSS3, and JavaScript.
- Communication: Discord for team discussions and updates.
- Project Management: Trello for tracking progress, tasks, and agile methodologies.
- Version Control: Git via GitHub.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The website should load within 3 seconds on standard broadband connections. (GitHub hosting permitting (could be slower if their end is under heavy load))

5.2 Security Requirements

- Basic security measures to protect against common web vulnerabilities.

5.3 Software Quality Attributes

- **Maintainability:** The code should be well-documented and structured for easy maintenance and future updates.
- **Scalability:** The design should accommodate potential expansion, such as adding more games or sections.
- **Reliability:** The website should have a high uptime, with minimal user downtime.

Game Design Document (GDD) for Programming Hangman

1. Game Overview

1.1 Concept

"Programming Hangman" is an educational web game designed to teach and test players on programming terminology. Developed using Godot 4.2 and exported for the web via HTML5, the game challenges players to guess programming-related words with hints related to their usage in the programming world.

1.2 Game Objectives

- Engage players with interactive gameplay focused on programming concepts.
- Educate players about different programming terms and their meanings.

1.3 Target Audience

Our primary audience includes students learning programming, developers interested in testing their knowledge, and anyone curious about programming.

2. Gameplay

2.1 Mechanics

- Players are presented with a series of underscores representing the letters of a programming-related word.
- Players guess letters to reveal the word. Incorrect guesses contribute to the drawing of a hangman.
- The game provides hints related to the word's usage in programming to aid the player.
- Successfully guessing the word before the hangman is completed rewards the player with a brief explanation of the term.

2.2 Levels

The game progresses through increasingly complex programming terms, starting from basic concepts and moving to more advanced terminology. The words can be randomly generated based on the "difficulty" level or value assigned to the word/concept/terminology.

2.3 Ending

The game ends when players complete a set number of words or fail a certain number of times. Players are encouraged to learn more about programming through external resources provided at the end.

3. Development

3.1 Tools and Technologies

- **Game Engine:** Godot 4.2
- **Programming Language:** GDScript

- **Graphics:** 2D sprites and text, designed within Godot or external graphic design software.
- **Sound:** Background music and sound effects for correct or incorrect guesses.
- **Export:** HTML5 for web integration.

3.2 Collaboration Tools

- **Communication:** Discord will be used for daily communication, updates, and discussions among team members.
- **Project Management:** Trello will be our Kanban board for tracking tasks, progress, and agile development milestones.

4. Art Style

4.1 Graphics

The game will feature a simple, clean, modern 2D art style that is appealing and non-distracting. It will focus on its educational aspect.

4.2 UI/UX

- The user interface will be intuitive, with a clear display of the word to guess, the hangman, and an on-screen keyboard for letter input.
- Pop-ups for hints and word explanations will be designed to be informative and engaging.

5. Sound

5.1 Music

Background music will be subtle and conducive to concentration and learning.

5.2 Sound Effects

Sound effects for letter guesses, correct or incorrect answers, and game progression cues will enhance the gameplay experience.

6. Marketing and Monetization (Optional)

6.1 Release Platform

The game will be hosted on the team's GitHub Pages website and freely accessible to all users.

6.2 Promotion

Promotion will occur through social media, programming forums, and educational platforms to attract our target audience.

7. Project Management

7.1 Milestones

- **Prototype:** A basic playable version demonstrating the core mechanics.
- **First Playable:** Incorporation of all planned game mechanics and a set number of words.
- **Alpha:** Integration of all programming words, hints, and explanations with essential UI/UX.

- **Beta:** Complete the game with finalized art and sound; it is ready for testing.
- **Launch:** Deployment on the website, with marketing efforts to follow.

7.2 Task Allocation

Tasks will be divided among team members based on expertise and interest, with clear assignments and deadlines managed through Trello.

7.3 Communication

Regular check-ins on Discord will ensure that all team members are aligned, and Trello will visually represent the project's progress.

MVP Document for Web Game Devs Project

1. Purpose

This MVP document outlines the essential features and functionalities that our website and Hangman game must include for the initial launch. It will guide our development process and ensure we focus on the core aspects that deliver value to our users.

2. MVP Definition

MVP (Minimum Viable Product) is the version of a new product that allows the team to collect the most validated learning about customers with the least effort.

3. Scope of MVP

Website:

- Responsive one-page design that adapts to desktop and mobile devices.
- Navigation bar to jump between sections of the site.
- The “Meet the Team” section has brief profiles of each member.
- The interactive Hangman game is focused on programming concepts.

Hangman Game:

- Basic gameplay mechanics allow users to guess letters of programming-related words.
- Hints related to the programming concept of the word.
- Simple feedback system to show correct/incorrect guesses, remaining attempts, and the solution upon completion.
- Popup or modal window explaining the programming concept of the guessed word upon game completion.

4. Development Priorities

- Priority 1: Core functionality of the Hangman game (game logic, basic UI).
- Priority 2: Basic website layout and responsive design.
- Priority 3: Content for the “Meet the Team” section.
- Priority 4: Integration of the Hangman game into the website.
- Priority 5: Additional features like game score tracking, leaderboards, and more detailed user feedback are considered beyond MVP and will be developed in subsequent iterations.

5. Tools and Technologies

- Game Development: Godot 4.2 uses GDScript and is exported to the web via HTML5.
- Website Development: HTML, CSS, JavaScript for front-end development.

- Project Management: Trello for task management and agile development.
- Communication: Discord for team discussions and updates.

6. Milestones

- Milestone 1: Game Concept and Logic Development
- Milestone 2: Basic Website Structure and Design
- Milestone 3: Hangman Game Development and Testing
- Milestone 4: Website and Game Integration
- Milestone 5: MVP Launch

7. Success Criteria

The MVP will be considered successful if it:

- Is fully functional on significant web browsers.
- Provides an engaging user experience.
- Educates users on programming concepts through gameplay.
- Encourages users to learn more about the team and project.

References

GitHub. (n.d.). *GitHub Learning Lab*. Retrieved from https://github.com/apps/github-learning-lab

GitHub. (n.d.). *GitHub Pages*. Retrieved from https://pages.github.com/

GitHub. (n.d.). *GitHub Pages Documentation*. Retrieved from https://docs.github.com/en/pages

GitHub. (n.d.). *Introduction to GitHub*. Retrieved from https://skills.github.com/

Godot Engine. (n.d.). *Documentation*. Retrieved from https://docs.godotengine.org/en/stable/index.html

Institute of Electrical and Electronics Engineers. (n.d.). *278253*. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/278253

Game Development Document link (GDD): https://github.com/AwaywithCharles/CMSC495-Capstone/blob/main/documentation/GDD.txt

Minimum Viable Product (MVP): https://github.com/AwaywithCharles/CMSC495-Capstone/blob/main/documentation/MVP.txt

Software Requirements Specifications (SRS): https://github.com/AwaywithCharles/CMSC495-Capstone/blob/main/documentation/SRS.txt

Project GitHub Repository: https://github.com/AwaywithCharles/CMSC495-Capstone