

# Week 2 Assignment: Capstone Project

## Week 2 Project Goals and Milestones for The Web Game Devs

This week, our team focused on laying the groundwork for our project. We aim to create a responsive, one-page website featuring an interactive Hangman game centered on programming concepts. Below I broke it down following similar SDLC processes that I have used previously for various projects in the industry, I included an Software Requirement Specifications (SRS) document for the web design portion (following IEEE's template), and Game Development Document (GDD) for the game development, as well as a Minimum Viable Product (MVP) sheet.

### Key Milestones:

#### 1. Requirements and Documentation:

- Finalize and review the Software Requirements Specification (SRS) and Game Design Document (GDD), incorporating our MVP (Minimum Viable Product) strategy. (Charles leads, with contributions from all members for feedback.)

#### 1. Development Environment Setup:

- Establish the development environment with Godot 4.2 and configure the tools for HTML5 web export. (Charles and Jade)
- Create a GitHub repository for version control and collaboration on the project. (Charles will initialize it, and everyone will clone and set it up in VSCode.)
- Use Trello board for agile project management, defining our workflow and initial tasks. (Start breaking down the weeks for the project to set deadlines and goals.)

### 1. Initial Game and Website Development:

- Start developing the Hangman game in GDScript, emphasizing core mechanics and the programming-related word database. (Jade/ Charles)
- Begin designing and coding the website's layout, ensuring responsiveness and including placeholders for the game, team information, and other relevant sections. (Robyn, Ada, and Holland focus on website layout design.)

### 1. Design Considerations:

- Decide on a cohesive color palette for the website and game to ensure a unified look and feel. (Ada leads the design choice, with input from Jade and Robyn.)

### 1. Team Collaboration and Communication:

- Regularly communicate through Discord to stay updated on progress and hurdles. (Everyone)
- Update the Trello board consistently to reflect the current status of tasks and upcoming priorities. (Everyone)

### 1. Technical Setup Guide:

- Compile a guide for team members using GitHub with Visual Studio Code for efficient version control and collaboration. (Charles will draft.)

## **Specific Requirements and Scenarios:**

- The game will actively engage users by challenging them to guess programming-related terms, with hints about their functions in the programming world. Successfully guessing a word triggers a popup explaining the term's significance, adding an educational layer to the game.

## **System Specifications:**

- Development focuses on Godot 4.2 for game programming and standard web technologies (HTML, CSS, JavaScript) for the website. We advise users to access our product using up-to-date web browsers on devices capable of efficiently running HTML5 content for an optimal experience

## # Week 2 Project Goals and Milestones for The Web Game Devs

This week, our team focused on laying the groundwork for our project. We aim to create a responsive, one-page website featuring an interactive Hangman game centered on programming concepts. Below I broke it down following similar SDLC processes that I have used previously for various projects in the industry, I included an Software Requirement Specifications (SRS) document for the web design portion (following IEEE's template), and Game Development Document (GDD) for the game development, as well as a Minimum Viable Product (MVP) sheet.

### ## Key Milestones:

#### 1. Requirements and Documentation:

- Finalize and review the Software Requirements Specification (SRS) and Game Design Document (GDD), incorporating our MVP (Minimum Viable Product) strategy. (Charles leads, with contributions from all members for feedback.)

#### 1. Development Environment Setup:

- Establish the development environment with Godot 4.2 and configure the tools for HTML5 web export. (Charles and Jade)
- Create a GitHub repository for version control and collaboration on the project. (Charles will initialize it, and everyone will clone and set it up in VSCode.)
- Use Trello board for agile project management, defining our workflow and initial tasks. (Start breaking down the weeks for the project to set deadlines and goals.)

#### 1. Initial Game and Website Development:

- Start developing the Hangman game in GDScript, emphasizing core mechanics and the programming-related word database. (Jade/ Charles)
- Begin designing and coding the website's layout, ensuring responsiveness and including placeholders for the game, team information, and other relevant sections. (Robyn, Ada, and Hol

land focus on website layout design.)

#### 1. Design Considerations:

- Decide on a cohesive color palette for the website and game to ensure a unified look and feel. (Ada leads the design choice, with input from Jade and Robyn.)

#### 1. Team Collaboration and Communication:

- Regularly communicate through Discord to stay updated on progress and hurdles. (Everyone)
- Update the Trello board consistently to reflect the current status of tasks and upcoming priorities. (Everyone)

#### 1. Technical Setup Guide:

- Compile a guide for team members using GitHub with Visual Studio Code for efficient version control and collaboration. (Charles will draft.)

### ## Specific Requirements and Scenarios:

- The game will actively engage users by challenging them to guess programming-related terms, with hints about their functions in the programming world. Successfully guessing a word triggers a popup explaining the term's significance, adding an educational layer to the game.

### ## System Specifications:

- Development focuses on Godot 4.2 for game programming and standard web technologies (HTML, CSS, JavaScript) for the website. We advise users to access our product using up-to-date web browsers on devices capable of efficiently running HTML5 content for an optimal experience

SRS:

# Software Requirements Specification (SRS) for the Web Game Dev

## ## 1. Introduction

### ### 1.1 Purpose

This document specifies the software requirements for the Web G

### ### 1.2 Document Conventions

This document adheres to the IEEE SRS standard format to ensure

### ### 1.3 Intended Audience and Reading Suggestions

This SRS is intended for project team members, stakeholders, and

### ### 1.4 Project Scope

The project scope encompasses developing a responsive website, (

## ## 2. Overall Description

### ### 2.1 Project Perspective

This one-page website is a standalone project designed to be ho

### ### 2.2 Project Functions

- A navigation bar for seamless access to the website's sections.
- An interactive Hangman game centered around programming termin
- A "Meet the Team" section with team member profiles.
- Utilization of Discord for continuous team communication.
- Adoption of Trello for agile project management and task track

### ### 2.3 User Classes and Characteristics

- Casual web visitors interested in playing the Hangman game.
- Team members and stakeholders are reviewing the progress of tl

### ### 2.4 Operating Environment

The website is accessible through modern web browsers like Chro

### ### 2.5 Design and Implementation Constraints

- The project must be browser-compatible and responsive.
- Development will use Godot 4.2 and GDScript, with HTML5 for web.
- The team will use Discord for communication and Trello for project management.

### ### 2.6 User Documentation

The site will provide instructions for gameplay and navigation, ensuring users can easily find and understand the game mechanics.

## ## 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 ease of navigation.

#### #### 3.1.2 Functional Requirements

- FR1: The website must feature a responsive design, ensuring usability across all devices.
- FR2: Navigation buttons at the top of the page must allow users to easily access different sections.
- FR3: The "Meet the Team" section must display team members' names and roles.
- FR4: The game section should embed the Hangman game, allowing users to play directly on the website.

### ### 3.2 Website Content

#### #### 3.2.1 Description and Priority

Medium priority. Content should be engaging and informative, reflecting the game's theme and providing clear instructions.

#### #### 3.2.2 Functional Requirements

- FR5: Content must be clearly written and understandable, catering to a wide audience.
- FR6: The website should include a section describing the project's goals and the team's vision.

## ## 4. External Interface Requirements

### ### 4.1 User Interfaces

- The website should have a visually appealing interface, color-coded to match the game's theme.

### ### 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 issues
- Version Control: Git via GitHub.

## ## 5. Other Nonfunctional Requirements

### ### 5.1 Performance Requirements

- The website should load within 3 seconds on standard broadband

### ### 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 readability.
- **Scalability:** The design should accommodate potential expansion of features.
- **Reliability:** The website should have a high uptime, with minimal downtime.

GDD:

## # Game Design Document (GDD) for Programming Hangman

### ## 1. Game Overview

#### ### 1.1 Concept

"Programming Hangman" is an educational web game designed to teach

### ### 1.2 Game Objectives

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

### ### 1.3 Target Audience

Our primary audience includes students learning programming, developers, and game enthusiasts.

## ## 2. Gameplay

### ### 2.1 Mechanics

- Players are presented with a series of underscores representing a hidden word.
- Players guess letters to reveal the word. Incorrect guesses cost lives.
- The game provides hints related to the word's usage in programming.
- Successfully guessing the word before the hangman is completed wins the round.

### ### 2.2 Levels

The game progresses through increasingly complex programming terms and concepts.

### ### 2.3 Ending

The game ends when players complete a set number of words or fail to guess the word correctly.

## ## 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 tools.
- **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.
- **Project Management:** Trello will be our Kanban board for task management.

## ## 4. Art Style



### ### 4.1 Graphics

The game will feature a simple, clean, and modern 2D art style

### ### 4.2 UI/UX

- The user interface will be intuitive, with a clear display of
- Pop-ups for hints and word explanations will be designed to be

## ## 5. Sound

### ### 5.1 Music

Background music will be subtle and conducive to concentration

### ### 5.2 Sound Effects

Sound effects for letter guesses, correct or incorrect answers,

## ## 6. Marketing and Monetization (Optional)

### ### 6.1 Release Platform

The game will be hosted on the team's GitHub Pages website and

### ### 6.2 Promotion

Promotion will occur through social media, programming forums,

## ## 7. Project Management

### ### 7.1 Milestones

- **Prototype:** A basic playable version demonstrating the core
- **First Playable:** Incorporation of all planned game mechanic
- **Alpha:** Integration of all programming words, hints, and ex
- **Beta:** Complete the game with finalized art and sound; it
- **Launch:** Deployment on the website, with marketing efforts

### ### 7.2 Task Allocation

Tasks will be divided among team members based on expertise and

### ### 7.3 Communication

Regular check-ins on Discord will ensure that all team members are

'MVP:

## # MVP Document for Web Game Devs Project

### ## 1. Purpose

This MVP document outlines the essential features and functional

### ## 2. MVP Definition

**\*\*MVP (Minimum Viable Product)\*\*** is the version of a new product

### ## 3. Scope of MVP

#### ### Website:

- Responsive 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
- Hints related to the programming concept of the word.
- Simple feedback system to show correct/incorrect guesses, remaining
- Popup or modal window explaining the programming concept of the

### ## 4. Development Priorities

- **Priority 1:** Core functionality of the Hangman game (game logic).
- **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.

## ## 5. Tools and Technologies

- **Game Development:** Godot 4.2 uses GDScript and is exported to HTML5.
- **Website Development:** HTML, CSS, JavaScript for front-end and PHP for back-end.
- **Project Management:** Trello for task management and agile workflow.
- **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.

## Resources

- **IEEE SRS Template/Reference:** A comprehensive guide and template for creating a Software Requirements Specification document, provided by the IEEE.

Institute of Electrical and Electronics Engineers. This resource is essential for understanding the structure and content needed for a professional SRS.

<https://ieeexplore.ieee.org/document/278253>

- **GitHub Pages:** The official site for GitHub Pages, which offers the ability to host websites directly from your GitHub repository. A fantastic solution for our project's web hosting needs. <https://pages.github.com/>
- **GitHub Pages Documentation:** Detailed documentation on how to set up and manage a GitHub Pages site. This will be invaluable for our team as we deploy our project website. <https://docs.github.com/en/pages>
- **GitHub Learning Lab:** An interactive learning platform offered by GitHub. It provides hands-on courses to help our team members get comfortable with GitHub's workflow and features. <https://github.com/apps/github-learning-lab>
- **Introduction to GitHub:** A beginner-friendly introduction to GitHub. This resource is perfect for team members new to version control and looking to understand the basics of GitHub operations. <https://skills.github.com/>
- **Godot Documentation:** The official documentation for the Godot Engine. It covers everything from basic setup to advanced features, making it a must-have for our game development efforts. <https://skills.github.com/>

## References

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

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.). *GitHub Learning Lab*. Retrieved from <https://github.com/apps/github-learning-lab>

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>

