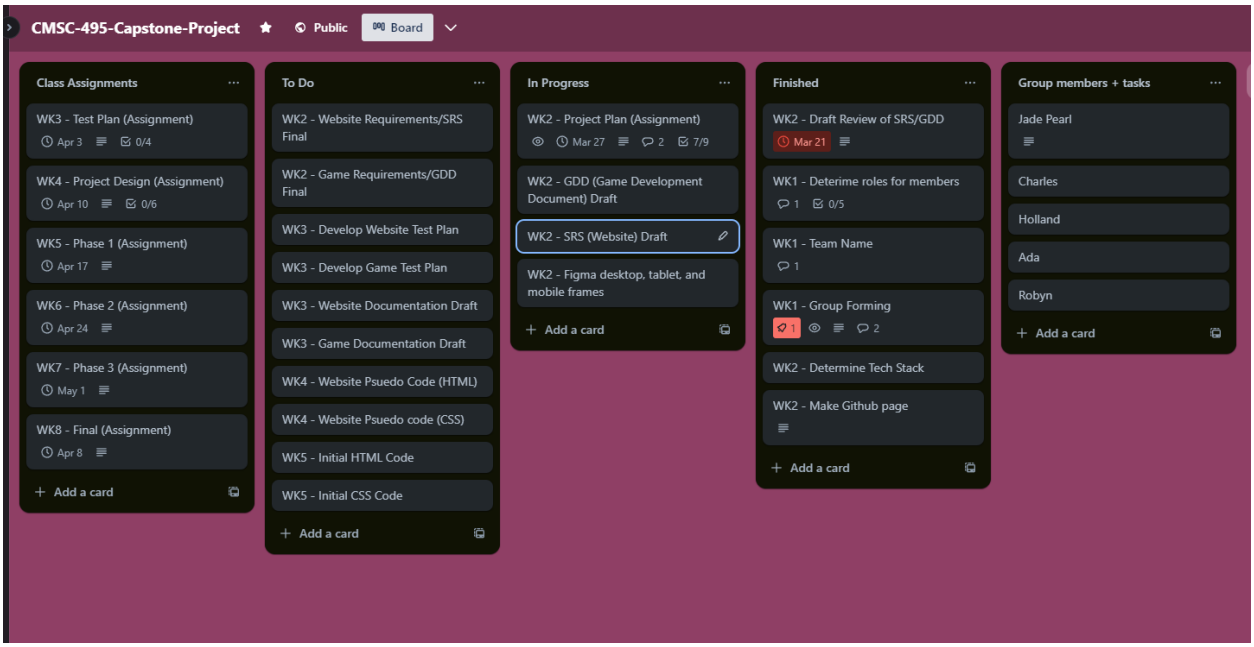
Week 2 Assignment: Capstone Project

Week 2 Project Goals and Milestones for The Web Game Devs

Team Members: Charles Bostwick, Jade Pearl, Ada Truong, Robyn LaMontagne, Holland Brawner

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 a Software Requirement Specifications (SRS) document for the web design portion (following IEEE's template) and a Game Development Document (GDD) for the game development, as well as a Minimum Viable Product (MVP) sheet.

I provided the initial layout to our team in Discord and via GitHub so that we could collaborate and review it. After discussing with the team on Discord and using Trello, we collectively created a roadmap for this project. Below is an image of our Trello board



We then took the team's feedback in our Discord channels and discussions to implement the changes and recommendations to finalize the initial draft based on an IEEE template for the SRS. Below, we have broken it down to align with the assignment rubric and the post requirements for the class. We have our Key Milestones, the discussed SRS, GDD, and MVP documents, and some of the references and documentation we will be using at the end in APA formatting.

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.)

2. 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.)

- We use the 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.)
- Determine if additional milestones, such as writing tests and deploying the website on GitHub Pages, are necessary.

3. 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.)
- Discuss the potential inclusion of multiple pages in the website structure and the necessity of a navigation bar.

4. 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.)

5. 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)
- Consider integrating Google Drive or Microsoft 365 for collaborative editing of documents, ensuring adherence to APA formatting guidelines.

6. Technical Setup Guide:

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

Software Requirements Specification (SRS) (SRS.md on our Github):

```
# 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 b
```

```
rowsers and devices, providing a responsive and interactive u
ser 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 ado
pt agile practices.

### 2.2 Project Functions
- A navigation bar for seamless access to the website's secti
ons.
- An interactive Hangman game centered around programming ter
minology.
- A "Meet the Team" section with member profiles and informat
ion.
- Utilization of Discord for continuous team communication.
- Adoption of Trello for agile project management and task tr
acking.
- There is a possibility of adding additional pages for addit
ional 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 Ch
rome, Firefox, Safari, and Edge, hosted on GitHub Pages. The
development and project management tools include Godot 4.2, D
iscord, 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, w
ith HTML5 for web export/embed.
- The team will use Discord for communication
- The team will use Trello for project management, adhering t
```

```
o agile methodologies.
- The team will use GitHub for version and source control.

### 2.6 User Documentation
The site will provide instructions for gameplay and navigatio
n, along with detailed documentation on GitHub for developmen
t 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 fo
r 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 all
ow users to smoothly scroll to the desired section and/or pag
e.
- 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, allowi
ng users to interact with it directly on the website. Initial
ly, 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, ca
```

```
tering to a broad audience.
- FR6: The website should include a section describing the pr
oject's goals and the technologies used in development.

## 4. External Interface Requirements

### 4.1 User Interfaces
- The website should have a visually appealing interface, col
or scheme, and layout that enhances readability and user expe
rience.

### 4.2 Hardware Interfaces
- No hardware interfaces are required as the project is web-b
ased.

### 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, an
d agile methodologies.
- Version Control: Git via GitHub.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements
- The website should load within 3 seconds on standard broadb
and 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 vulne
rabilities.
```

```
### 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:

```
# 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 represen
ting 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 prog
ramming to aid the player.
- Successfully guessing the word before the hangman is comple
ted 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 advanc
ed 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 lea
rn 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 o
r incorrect guesses.
- **Export:** HTML5 for web integration.
```

```
### 3.2 Collaboration Tools
- **Communication:** Discord will be used for daily communica
tion, 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 th
at is appealing and non-distracting. It will focus on its edu
cational 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 concentratio
n and learning.

### 5.2 Sound Effects
Sound effects for letter guesses, correct or incorrect answer
s, and game progression cues will enhance the gameplay experi
ence.

## 6. Marketing and Monetization (Optional)

### 6.1 Release Platform
The game will be hosted on the team's GitHub Pages website an
d freely accessible to all users.
```

```
### 6.2 Promotion
Promotion will occur through social media, programming forum
s, and educational platforms to attract our target audience.

## 7. Project Management

### 7.1 Milestones
- **Prototype:** A basic playable version demonstrating the c
ore mechanics.
- **First Playable:** Incorporation of all planned game mecha
nics 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; i
t is ready for testing.
- **Launch:** Deployment on the website, with marketing effor
ts to follow.

### 7.2 Task Allocation
Tasks will be divided among team members based on expertise a
nd interest, with clear assignments and deadlines managed thr
ough Trello.

### 7.3 Communication
Regular check-ins on Discord will ensure that all team member
s are aligned, and Trello will visually represent the projec
t's progress.
```

MVP Document for Web Game Devs Project

```
# MVP Document for Web Game Devs Project
```

```
## 1. Purpose

This MVP document outlines the essential features and functio
nalities that our website and Hangman game must include for t
he 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 prod
uct that allows the team to collect the most validated learni
ng about customers with the least effort.

## 3. Scope of MVP

### Website:

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

### Hangman Game:

- Basic gameplay mechanics allow users to guess letters of pr
ogramming-related words.
- Hints related to the programming concept of the word.
- Simple feedback system to show correct/incorrect guesses, r
emaining 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 (gam
e 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 we
bsite.
- **Priority 5:** Additional features like game score trackin
g, leaderboards, and more detailed user feedback are consider
ed beyond MVP and will be developed in subsequent iterations.

## 5. Tools and Technologies

- **Game Development:** Godot 4.2 uses GDScript and is export
ed to the web via HTML5.
- **Website Development:** HTML, CSS, JavaScript for front-en
d development.
- **Project Management:** Trello for task management and agil
e development.
- **Communication:** Discord for team discussions and update
s.

## 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

The main repo for the project: https://github.com/AwaywithCharles/CMSC495-Capstone