Test Plan and User Guide

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

Department of Computer Science, University of Maryland Global Campus

Professor Shanna Nevarez

April 2, 2023

Test Plan Document for Web Game Devs Project

Team Members

- **Charles Bostwick:** Project Lead, Technical Writing
- **Jade Pearl:** Game Development
- **Ada Truong:** UI/UX, Front-End Development
- **Robyn Cohen:** UI/UX, Front-End Development
- **Holland Brawner:** Front-End Development, Game Development

Introduction

The Web Game Devs project combines theoretical knowledge with practical application to create a responsive educational website for our capstone project at University of Maryland Global Campus that is hosted on GitHub. The centerpiece of the site or key aspect will be that it embeds a "Programming Hangman" game that is developed using Godot 4.2 and GDScript. The website will demonstrate the team's effectiveness in developing with HTML, CSS, and JavaScript and ability to set milestones and goals as well as meet their deadlines for the required tasks to accomplish this project. This document outlines our comprehensive Test Plan.

Purpose

We want to ensure that the website and "Programming Hangman" game are fully operational, user-friendly, and educational. Our goal is to provide an informative, accessible experience for users while maintaining high performance and responsiveness standards.

Scope

The scope encompasses developing and testing the website and web-embedded game, focusing on programming concepts. This includes testing for responsiveness, game mechanics, and team member information accuracy without delving into database integration, user authentication, API calls, or extensive server/cloud networking.

1. **Test Items**
    - **Responsive Website Interfaces:** Ensure compatibility across devices and browsers.
    - **"Programming Hangman" Game Mechanics and Integration:** Test for functional gameplay and seamless integration within the site.
    - **Team Member Information Page:** Verify accuracy and responsiveness.
    - **Cross-browser and Cross-device Compatibility:** Ensure uniformity of experience across platforms.
1. **Features To Be Tested**

- **Navigation and Screen Adaptability:** Evaluate ease of navigation and adaptability to various screen sizes.
- **Game Functionality:** Test game logic, user interaction, and feedback mechanisms.
- **Accurate Team Information Display:** Ensure all team member details are correct and up to date.
- **UI/UX Evaluation:** Assess the aesthetic and user experience across the website.

1. **Features Not To Be Tested**
    - **Backend Server Functionalities:** Given that the project's scope does not include server-side processing.
    - **Third-Party APIs:** No external APIs are involved in this project.
    - **External Links Functionality:** Links leading outside the project domain will not be tested.

Test Plan

Approach

Our approach combines manual and automated testing methods, focusing on specific project components to ensure thorough coverage.

Testing Methods

- **Manual Testing:** Conducted by team members for subjective assessment of UI/UX and navigation. Involves exploratory testing to identify unexpected behavior.
- **Automated Testing:** Uses tools like Selenium WebDriver for repetitive tasks, ensuring accuracy and efficiency. GitHub Actions automate CI/CD pipelines, facilitating continuous testing.
- **Performance Testing:** Utilizes Google PageSpeed Insights to benchmark website load times and responsiveness, ensuring optimization.

Test Deliverables

- **Test Cases Document:** A detailed enumeration of all test scenarios and expected outcomes.
- **Issue Reports:** Document any issues encountered, including severity, impact, and reproduction steps.
- **Final Testing Summary:** A report summarizing the testing phase, outcomes, and unresolved issues.

Testing Tools

- **Development and Testing:** Visual Studio Code, Godot Engine, Jest (for JavaScript testing), Selenium WebDriver (for automated web testing), GitHub Actions (for CI/CD).
- **Project Management:** Trello (task management), Discord (team communication), GitHub (version control and issue tracking).

Dependencies and Prerequisites

Software Requirements

Requirement	Details	Version(s)	Required For
Operating System	Windows, macOS, Linux	Windows 10+, macOS Catalina+, Ubuntu 20.04+	Development & Testing
Browser	Chrome, Firefox, Safari, Edge	Latest two versions	Cross-browser Testing
Code Editor	Visual Studio Code	Latest version	Development
Web Server	GitHub Pages	N/A	Hosting & Deployment
Game Engine	Godot Engine	4.2	Game Development
Testing Framework	Jest, Selenium WebDriver	Latest versions	Automated Testing

Hardware Requirements

Requirement	Minimum Specifications	Recommended Specifications	Required For
Processor	Intel i3 or equivalent	Intel i7 or equivalent	Development & Testing
RAM	8GB	16GB+	Development & Testing
Storage	SSD with 20GB free space	SSD with 50GB+ free space	Development & Testing
Internet Connection	Broadband (10Mbps)	High-speed Broadband (50Mbps+)	Testing & Deployment
Display	1080p	1440p or higher	UI/UX Testing

Pass/Fail Criteria

Success criteria include functional navigation, consistent aesthetics across devices, swift load times, and accurate game mechanics. Bug tracking identifies failures, which are addressed iteratively.

Test Cases

Test Case ID	Description	Expected Outcome	Pass/Fail	Note
TC001	Test website responsiveness on Chrome	Website adapts to screen size without horizontal scrolling		
TC002	Test "Programming Hangman" game load time on Firefox	Game loads within 3 seconds		
TC003	Verify game functionality for a basic programming question	Correct answer proceeds to the next level		
TC004	Test website navigation on mobile device	Users can navigate to all pages without zooming		
W001	User loads the webpage for the first time.	The homepage should be displayed, with buttons/links to start the hangman game or view team member information.		
W002	User clicks on the “Meet the Team” link.	The “Meet the Team” page should be displayed, with information about and/or a profile picture of each team member.		
W003	User clicks on the “Back” button on any page.	The homepage should be displayed.		

GF001	User indicates that they would like to start the game by clicking a “Start Game” button.	The game should load a random programming word, and an empty hangman board should appear. The game should also display an on-screen keyboard with an expandable hint.		
GF002	User enters a correct letter, either through their own keyboard or through the on-screen keyboard.	The game should display the matching letter(s) on the word.		
GF003	User enters an incorrect letter, either through their own keyboard or through the on-screen keyboard.	The game should register incorrect answers and display feedback (e.g., by drawing another part of the figure and/or ending the level if the figure is fully drawn)		
GF004	User clicks on the hint symbol/button.	The game should either expand or collapse a hint about the term, depending on its current state.		
GF005	User completes a level with the correct answer (i.e., the word is fully filled out)	The game should display a success message and the definition of the term. The game should also provide the user with the option to move onto the next level.		Ensures game progression is logical
GF006	User indicates that they would like to move onto the next level by clicking the “New Game” button.	The game should load a new random programming word, and the hangman board should be wiped/reset.		

GF007	User indicates that they would like to exit the game by clicking the “Back” button.	The game should end and the user should be rerouted to the home page.		
GF008	Keyboard users enter edge case input (e.g., symbols, numbers).	Game should not crash and provide appropriate feedback		Checks for robust input handling
RS001	Resize browser window on desktop (Chrome)	Website layout adapts smoothly to varying window sizes		
RS002	Access website on a smartphone (Safari)	Website is usable without horizontal scrolling and minimal zooming		Checks mobile responsiveness
RS003	Access website on a tablet in landscape mode (Firefox)	Elements adjust appropriately for the tablet screen size		

Test Schedule

Phase	Tasks	Start Date/End Date
Preparation	Setup environments, create test cases	
Execution	Manual and automated testing	
Evaluation	Analyze test results, report issues	
Finalization	Final testing, documentation review	

Control Procedures

- **Bug Tracking:** Using GitHub Issues for detailed defect management, categorizing by severity, and assigning for resolution.
- **Change Management:** Code changes are managed through GitHub Pull Requests and reviewed by at least two team members before integration.

Risks and Mitigations

- **Resource Limitations:** Mitigated by prioritizing test cases based on impact and likelihood.
- **Technical Challenges:** Addressed through regular team training sessions and leveraging community support.
- **Tight Schedules:** Flexibility in task assignments and overtime planning to ensure deadlines are met.

Approval

Requires sign-off by all project team members and Professor Shanna Nevarez. The digital approval process can/will be managed through GitHub, with all members reviewing and endorsing the final document.

References

"IEEE Standard for Software and System Test Documentation," in IEEE Std 829-2008 , vol., no., pp.1-150, 18 July 2008, doi: 10.1109/IEEESTD.2008.4578383. keywords: {IEEE Standards;Software;Documentation;Testing;Task analysis;829-2008;integrity level;life cycle;test documentation;testing},

User Guide for Web Game Devs Project

Welcome to the Web Game Devs project, an interactive platform that enhances your programming knowledge through the fun and engaging "Programming Hangman" game. This guide will walk you through navigating our website and making the most of the educational content available.

Getting Started

Accessing the Website

- The Web Game Devs project is accessible at (enter the thingy from GitHub later)
- Compatible with modern web browsers like Chrome, Firefox, Safari, and Edge.
- Designed to be responsive on mobile, tablet, and desktop devices.

Navigation/Pages

- **Home:** The welcome page, from which you can play the Programming Hangman game, learn more about the contributors, and get in contact with the team.
- **Meet the Team:** Discover more about the creators behind the project.
- **Play Game:** Direct access to the "Programming Hangman" game.
- **Contact Us:** Get in touch with us for feedback or queries.

Playing 'Programming Hangman'

Starting the Game

- Click on "Play Game" on the Home page to begin.
- No user registration is required, allowing immediate access to the game.

Gameplay Instructions

- **Objective:** Guess programming-related words or terms with a limited number of incorrect guesses.
    a. **How to Play:** Click or tap on the letters you guess might be in the word. You may also enter guesses through your keyboard.
    b. Try to guess the word correctly before the hangman drawing is completed.
    c. Use the provided hint if you're stuck.

Features

- **Hints:** Click on the "Hint" button if you need a clue to the current word.
- **Restart Game:** You can start over anytime by selecting "Restart."
- **Continue Playing:** Once a round ends, you can choose to start a new round with a new programming term by clicking “Play Again”, or return to the Home page by clicking “Back”.

Understanding 'Meet the Team'

- This section showcases the team behind the Web Game Devs project.
- Learn about each team member's role, background, and contributions to the project.

Frequently Asked Questions (FAQs)

Q: Do I need to install anything to play the game?
A: The web-based game can be played directly in your browser.
Q: Is the game free to play?
A: Yes, "Programming Hangman" is entirely free to play.
Q: Can I play the game on my phone?
A: Absolutely! The website and game are optimized for mobile devices.
Q: Where can I provide feedback or report a bug?
A: Please use the "Contact Us" section of the website to reach out to us.

Support and Contact Information

If you encounter any issues or have questions, please don't hesitate to contact us through the website's "Contact Us" section. We are here to help and improve your experience.
Thank you for visiting the Web Game Devs project. Enjoy learning and playing!
This guide is part of the Web Game Devs project documentation and is subject to updates and enhancements. For the latest version, please visit our website at