WASHINGTON STATE UNIVERSITY
VANCOUVER

CS 320 Course Project Final Report

for

Hangman

Prepared by

Group Name: CupaRum Team 11

David Barko	**11684680**	**David.barko@wsu.edu**
Daniel Yarmolenko	**11688943**	**Daniel.yarmolenko@wsu.edu**

Date: 12/13/2019

Contents

CONTENTS ..... **II**

1 INTRODUCTION **1**

1.1 PROJECT OVERVIEW 1
1.2 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 1
1.3 REFERENCES AND ACKNOWLEDGMENTS 1

2 DESIGN **1**

2.1 SYSTEM MODELING 1
2.2 INTERFACE DESIGN 2

3 IMPLEMENTATION **3**

3.1 DEVELOPMENT ENVIRONMENT 3
3.2 TASK DISTRIBUTION 3
3.3 CHALLENGES 3

4 TESTING **4**

4.1 TESTING PLAN 4
4.2 TESTS FOR FUNCTIONAL REQUIREMENTS 4
4.3 TESTS FOR NON-FUNCTIONAL REQUIREMENTS ..... **ERROR! BOOKMARK NOT DEFINED.**
4.4 HARDWARE AND SOFTWARE REQUIREMENTS 5

5 ANALYSIS **5**

6 CONCLUSION **6**

APPENDIX A - GROUP LOG ..... ERROR! BOOKMARK NOT DEFINED.

1 Introduction

1.1 Project Overview

This project is a the simple to play and fun game of Hangman. The whole project is web based and has multiple functionalities. This game can be played multiplayer or single player. It also has a few options where users can report problems with the software by contacting the developers. It also has an option which allows a player to add a word to the database which will then be used in single player.

1.2 Definitions, Acronyms and Abbreviations

Challenge –A Challenge includes a hint and the answer to the hint. The hint is a sentence that gives the user a clue as to what the answer iswhere the answer is a word or a phrase.
Hint –A hint is a short sentence or phrase that gives a user a clue of what the word is

1.3 References and Acknowledgments

"MySQL Improved Extension," php. [Online]. Available: https://www.php.net/manual/en/book.mysqli.php. [Accessed: 14-Dec-2019].

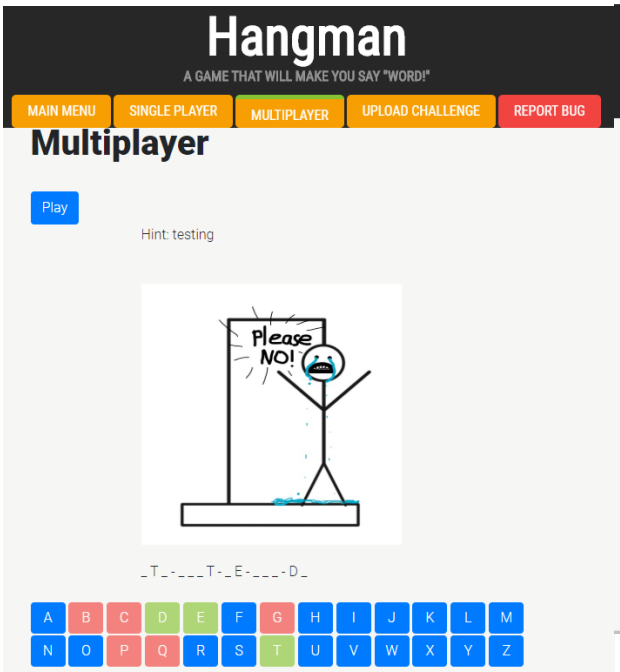
M. Otto and J. Thornton, "Introduction," · Bootstrap. [Online]. Available: https://getbootstrap.com/docs/4.4/getting-started/introduction/. [Accessed: 14-Dec-2019].

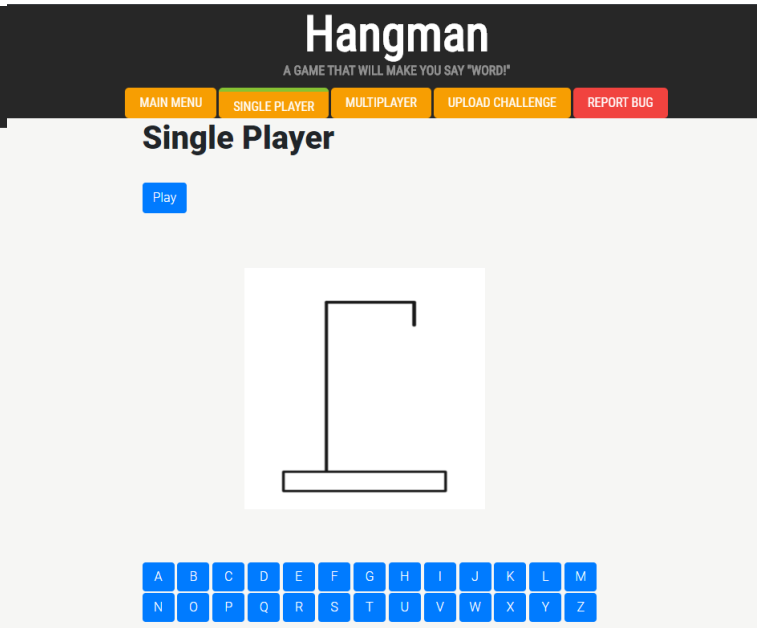
2 Design

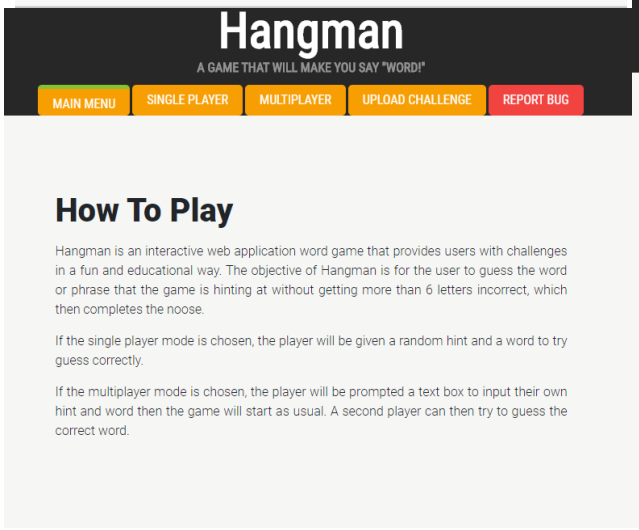
2.1 System Modeling

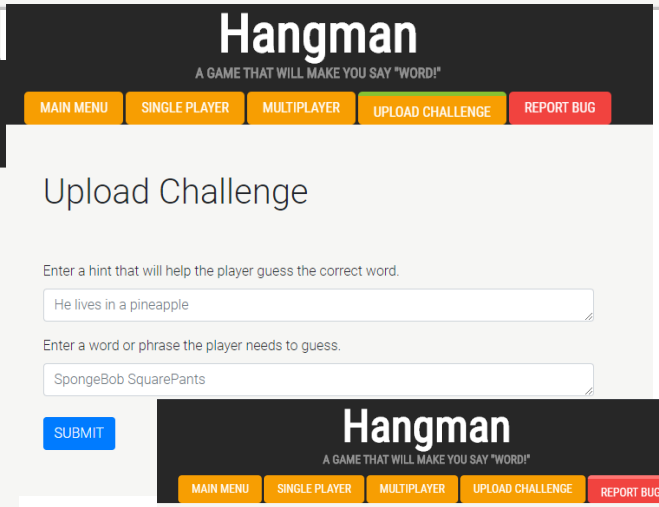
Our implementation strictly follows the design document (milestone 2)

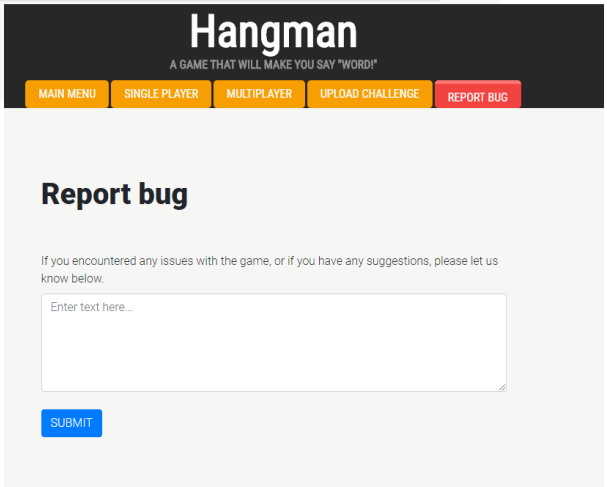
2.2 Interface Design











3 Implementation

3.1 Development Environment

Our main development envoirment was intellij, however, we did use some visual studio code. We also use Wamp localhost server to host our web based project.

3.2 Task Distribution

Daniel Yarmolenko:

Created the main UI design of the project
Put together all the functionalitys
Tied the front end with the back end
Created the report a problem

David Barko:

Created the main game functions
Made main gameplay
Managed the sql database along with all the data handling
Mostly worked with backend programming

3.3 Challenges

4 Testing

4.1 Testing Plan

Most of our testing methods consisted of white box testing where we would take a section of code and test its ouput based on the input it received. For example, we would input strings into the test areas in multiplayer such as numbers and symbols which would break our code. Then we would review the code and check the input and output of each function. If the output of a function was not correct, we would modify the code until it works.

Test 1: Input non alphabetic strings into the hint and word text boxes in multiplayer to check if it will break the web application.

Test 2: Try to play the game again mid-game and check if any errors occur.

Test 3: Click on letters that have already been guessed and check if strike counter increases (it should not)

Test 4: Add a word to MySQL that is already in there and check if a duplicate is added when it shouldn't.

4.2 Tests for Functional Requirements

Test 1:

When symbols or numbers were entered into the text box the game would not function properly until he web application was refreshed. The function that needed to be fixed was prepWord(), which takes in a string and outputs true if the string is valid.

Test 2:

When the play button was pressed again in the middle of a game, the game would not function properly until the web app was refreshed. We forgot to make sure that the play button deactivates during the game. So in the function gameLoop() and initWord() we added several flags that represent if the game is playing, or if the game has been played.

Test 3:

The alphabet buttons would increase the strike counter and the correct counter multiple times. To fix that we added a unique id to each letter button and onclick the gameLoop() function would be called with the corresponding id of the letter in the parameter. The gameLoop() function would then deactivate the corresponding letters that were clicked, and change the color depending on whether the letter was correct or not.

Test 4:

Adding a word into the MySQL database that already exists would add a duplicate word and hint. This problem was fixed with a few lines of code that compares the word to the word in the database, and it would not upload the word if one already exits.

4.3 Tests for Non-functional Requirements

Test 1:

Tested the sql database by entering a string with more than 100 characters to make sure it would not insert. This worked because the sql code limits the character array to 100.

Test 2:

Tested the web application on other browsers to check if there were any major differences. No differences that interferred with the usage of the web application were found.

Test 3:

Decreased the browser width size to the minimum and tested the usability of the web application. The web application was easy to use and there were minimal compromises because the bootstrap grid system was utilized to ensure that the design of the web application did not break on smaller screens. The only issue that arose was that the letter buttons layout were not symmetrical, but they were still usable.

4.4 Hardware and Software Requirements

The Hardware requirments are to have a working computer. The software requirements was to have a localhost setup with a fully functional mySQL database hosted by myphpAdmin. This allowed us to use the SQL database.

5 Analysis

Daniel Yarmolenko:
Milestone 1: 9 hours
Milestone 2: 7 hours
Milestone 3: 35 hours

David Barko:
Milestone 1: 9 hours
Milestone 2: 6 hours
Milestone 3: 35 hours

Milestone 3 took the most amount of hours from both of us, because we both had to work together to make sure what we program would be compatible with the others code.

6 Conclusion

Throughout this project we have learned a lot about php, html, mySQL and Javascript. We also learned how to all of those languages together to create a fully functional website with a database. We also learned how to implement a UI framework such as bootstrap to make our website design more modern and clean. This project has been a great learing experince for the both of us and we have taken away a lot of lessons for our future endeavors.

Appendix A - Group Log

We had daily meetings after classes. Most of our work was done over github and discord. The communication was very effective and it was the only way that we would be able to put all of our work together.