Lab 3 – War of Brawns Test Plan

War of Brawns (Team Bronze)

Old Dominion University

CS 411W

Professor James Brunelle

30 April 2020

Final Version 2

Table of Contents

1.   **Objectives**

The purpose of this Test Plan is to analyze the performance and operation of the War of Brawns prototype, which will then validate the development and capabilities of the War of Brawns product. The test cases defined in the document below will be utilized in order to prove the functionality of the War of Brawns prototype. The prototype will also implement simulated electrocardiogram (ECG) data and a mock nutritional database to accentuate the feasibility of the application.

[This space left intentionally blank]

2.   **References**

Team Bronze. (2020, April 4). Lab 1 - War of Brawns Product Description. Retrieved from

https://www.cs.odu.edu/~411bronze/Labs.php.

Team Bronze. (2020, April 4). Lab 2 - War of Brawns Product Specification. Retrieved from

https://www.cs.odu.edu/~411bronze/Labs.php.

[This space left intentionally blank]

3.   **Test Procedures**

Section 4 will define the necessary testing the War of Brawn prototype will undergo to ensure functionality. The test cases below include the following fields: test category, description, test case, case name, version, author, requirements fulfilled, purpose, setup conditions, test case activity, pass/fail condition, comments, and the expected result.

[This space left intentionally blank]

4.1.   Login page

Test Category: Login	Description: Authentication of users through the login page in the application		
Test Case: 4.1.1	Case Name: Authentication	Version: 1.0	Written By: Tim Novak
Requirements Fulfilled: 3.1.1.2.1, 3.1.1.3.1, 3.1.1.4.1, 3.1.5.1.1,	Purpose: To verify that users are only accessing their data and no one else's		
Setup Conditions:			
1 Create a test user in the Firebase Console with an email and password			
2 Launch the War of Brawns prototype application			
3 Enter the email address for the test user in the Email input field on the login screen			
Test Case Activity	Pass/Fail	Comments	Expected Result
1 Enter the correct password and press the Login button		Firebase automatically checks passwords and handles exceptions in the backend	User will be logged into the application and brought to the homepage which displays the user's avatar, stats, and a navigation bar..

4.2.   Progress page

Test Category: Progress page	Description: Current weight entry and the display of weight progression graph		
Test Case: 4.2.1	Case Name: Current Weight Entry	Version: 1.0	Written By: Roshan Ghimire
Requirements Fulfilled: 3.1.1.5.1, 3.1.1.5.3, 3.1.1.7.1, 3.1.5.1.2, 3.1.5.1.4,	Purpose: To demonstrate the process of how a user would input their current weight and a display of the progression graph including the updated weight.		
Setup Conditions:			
1. Obtain the requirements document.			
2. The tester must have Unity set up with the WoB GitLab repository.			
3. The tester must be logged in to the application.			

Test Case Activity		Pass/ Fail	Comments	Expected Result
1	Click on the progress page on the navigation bar.			The progress page displays a line graph with a white dot representing the current weight.
2	Click on the settings page on the navigation bar.			The settings page displays the details of the user's profile information and goals as follows: Current weight, goal weight, height, age, sex, activity level, goal per week.
3	Click on the "Update Weight" button.			It displays a pop-up box where the user is able to update their current weight.
4	Enter the current weight of the user and click on "set".			The current weight of the user shall be updated to the weight entered.
5	Click on the progress page on the navigation bar.			The progress page shall display a line graph of the

				user weights since they started playing this game including the recently updated weight.

4.3.   Diet Tracking

Test Category: Diet Tracking	Description: This test will verify that the gamer can query to the database in order to calculate their daily recommended calorie intake. This will also certify that the algorithm determining this value is properly functioning.		
Test Case: 4.3.1	Case Name: Daily Suggested Caloric Intake	Version: 2.0	Written By: Mohammad Bashghareh
Requirements Fulfilled: 3.1.3.1 - 3.1.3.1.1	Purpose: To provide the gamer with the suggested amount of calories they should consume in order to maintain their weight, lose weight, or lose weight rapidly.		
Setup Conditions:			
1. Populate the War of Browns database with sample user account data.			
2. Tester must launch the War of Brawns prototype application.			
3. Tester must login to the prototype using the sample username and password.			
4. Tester must navigate to and select the 'Diet Journal' from the navigation bar.			

Test Case Activity		Pass/Fail	Comments	Expected Result
1	The tester may view the display of the suggested caloric intake upon opening the 'Diet Journal' pane.		The algorithm to calculate this value is based on the Mifflin-St. Jeor equation, which has shown to be an	The caloric intake in the display should be the same as the one calculated in setup.

			accurate estimate of calorie needs.	
2	The tester may navigate to the 'Settings' pane and edit the profile such that a required value becomes NULL, then return to the 'Diet Journal'		The required values include the gamer's gender, current weight, height, age, activity level, and goal weight.	An error message should display notifying the gamer that one or more required inputs are missing, and to return to their 'Settings' to submit this information.
3	The tester may disable Internet access and return to the 'Diet Journal'			An error message should display indicating that connection to the database has failed and informing the gamer to try again.

Test Category: Diet Journal	Description: Logging meals thorough search and custom entry functions.		
Test Case: 4.3.2	Case Name: Diet Journal Entry Features	Version: 1.0	Written By: April McIntire
Requirements Fulfilled: 3.1.1.8.1, 3.1.1.8.2, 3.1.3.1.1, 3.1.3.2.1, 3.1.3.3.1, 3.1.3.4.1, 3.1.5.1.5	Purpose: To ensure all features of the diet journal are functioning correctly.		

Setup Conditions:
1 Log into War of Brawns prototype account
2 Navigate to the Diet Journal Page

Test Case Activity		Pass/ Fail	Comments	Expected Result
1	Add a food item to the breakfast section through the search function.			The search page should allow the user to search the database for a specific food. The results should be displayed and then allow the user to add a search result. That search result should be displayed in the breakfast section.
2	Add a food item to the breakfast section through the custom entry function.			The custom entry page should allow for user input and the user input should be displayed under the breakfast section.
3	Add a food item to the lunch section through the search function.			The search page should allow the user to search the database for a specific food. The results should be displayed and then allow the user to add a search result. That search result should be displayed in the

				lunch section.
4	Add a food item to the lunch section through the custom entry function.			The custom entry page should allow for user input and the user input should be displayed under the lunch section.
5	Add a food item to the dinner section through the search function.			The search page should allow the user to search the database for a specific food. The results should be displayed and then allow the user to add a search result. That search result should be displayed in the dinner section.
6	Add a food item to the dinner section through the custom function.			The custom entry page should allow for user input and the user input should be displayed under the dinner section.
7	Add a food item to the snack section through the search function.			The search page should allow the user to search the database for a specific food. The results should be displayed and then allow the user to add a search result. That search result should be displayed in the snack section.
8	Add a food item to the snack section through the custom entry function.			The custom entry page should allow for user input and the user input should be displayed under the snack section.

4.4.   Gameplay

Test Category: Gameplay	Description: Testing in-game combat mechanics		
Test Case: 4.4.1	Case Name: In-game mechanics	Version: 1.0	Written By: Josten Asercion
Requirements Fulfilled: 3.1.2-3.1.2.4.2, 3.1.1.9.4	Purpose: To ensure that in-game combat mechanics are functional		
Setup Conditions:			
1 Launch the War of Brawns prototype application			
2 Navigate to the gameplay page and enter the castle			

Test Case Activity		Pass/ Fail	Comments	Expected Result
1	Observe the opening display			Player health bar and energy bar will display in the top left of the screen, player avatar and dungeon environment will display on screen
2	Attempt to move, roll, jump, and attack			Player avatar will be able to move horizontally, jump, roll, and attack. Each action consumes some energy
3	Observe enemy AI			Enemies will follow the player and start an attack when player is within attack range
4	Pause the game			Game will pause, pause menu will display
5	Resume the game			Game will unpause, pause menu will disappear
5	Allow Enemy AI to deplete all of the player avatar's health			A defeat screen will display indicating that the dungeon was failed
6	Attack enemies until all enemies are defeated			A victory screen will display indicating that the dungeon was completed successfully

4.5.   Workout Tracking

	Description: This test case will go through the process of a user completing exercises within a workout challenge using the simulated smartwatch ECG data. It will also show the display of post-exercise experience point distribution and improvement in gameplay after the stat increase.		
Test Category: Workout Tracking			
Test Case: 4.5.2	**Case Name:** Workout Challenges Process and Result	**Version:** 1.0	**Written By:** Kevin Fang & Rodolfo Monterosso
Requirements Fulfilled: 3.1.1.6.2, 3.1.1.6.3, 3.1.4.1.1, 3.1.4.1.2, 3.1.4.2.1, and 3.1.5.1.3.	**Purpose:** To demonstrate the process of how a user would complete a workout challenge and how it would affect gameplay.		

Setup Conditions:
1. Obtain the requirements document.
2. The tester must have Unity setup with a clone of WoB GitLab repository.
3. WoB database must be populated with user data.
4. ECG simulated data file must be populated.
5. The tester must be logged in to the application.
6. The tester must have already navigated to the workout challenges page

	Test Case Activity	Pass/ Fail	Comments	Expected Result
1	The tester must activate the verification interpretation algorithm by pressing the "Start Challenge" button.			A new display with a timer of and multiple buttons will be shown. The timer is how long the tester wants to spend on that exercise. The multiple buttons consist of pause, next challenge, cancel, and finish.
2	The tester must press "Finish" when they do not want to spend more time on the exercise.			The previous display will be shown with how long they did the specific exercise.

3	The tester must repeat steps 1-2 to complete the whole challenge.			The display will show the exercises in the challenge with time of how long the tester did those exercises.
4	The tester should finish all exercises and press "Finish Challenge".			There will be a new display that displays how many experience points they received and what their new stat level will be.
5	The tester must navigate to the gameplay page and activate the dungeon to show the result of improvement in stat level.			Dungeon fighting shall display improvement as a result of stat level increase.

5.   **Traceability to Requirements**

Requirement ID	Test Case ID						
	4.1.1	4.2.1	4.3.1	4.3.2	4.4.1	4.5.1	4.5.2
3.1.1.1.1	Not Implemented						
3.1.1.2.1	X						
3.1.1.3.1	X						
3.1.1.4.1	X						
3.1.1.5.1		X					
3.1.1.5.2	Not Implemented						
3.1.1.5.3		X					
3.1.1.5.4	Not Implemented						
3.1.1.6.1						X	
3.1.1.6.2							X
3.1.1.6.3						X	X
3.1.1.7.1		X					
3.1.1.8.1				X			
3.1.1.8.2				X			
3.1.1.9.1	Not Implemented						
3.1.1.9.2	Not Implemented						
3.1.1.9.3	Not Implemented						
3.1.1.9.4					X		
3.1.2.1.1					X		
3.1.2.1.2					X		
3.1.2.1.3					X		
3.1.2.2.1					X		
3.1.2.2.2					X		
3.1.2.3.1					X		
3.1.2.3.2					X		
3.1.2.3.3					X		
3.1.2.4.1					X		
3.1.2.4.2					X		

3.1.3.1.1			X	X			
3.1.3.2.1				X			
3.1.3.3.1				X			
3.1.3.4.1				X			
3.1.4.1.1							X
3.1.4.1.2							X
3.1.4.2.1							X
3.1.5.1.1	X						
3.1.5.1.2		X					
3.1.5.1.3							X
3.1.5.1.4		X					
3.1.5.1.5		X					