Dankest Dungeon
Testing and Inspection Report

Prepared by: ***Aashish Agrawal, Karan Ahuja, Richard Miramontes, and Zohar Sajith for*** **CS 440**
at the
University of Illinois Chicago

November 2019

Table of Contents

List of Figures 3
List of Tables 4

I Project Description 5

1 Project Overview 5

2 Project Domain 5

3 Relationship to Other Documents 5

4 Naming Conventions and Definitions 5

4a Definitions of Key Terms 6
4b UML and Other Notation Used in This Document 6
4c Data Dictionary for Any Included Models 6

II Testing 7

5 Items to be Tested 7

6 Test Results 26

7 Regression Testing 34

III Inspection 35

8 Items to be Inspected 35

9 Inspection Procedures 35

10 Inspection Results 35

IV Recommendations and Conclusions 37

V Project Issues 37

11 Open Issues 37

12 Waiting Room 37

13 Ideas for Solutions 38

14 Project Retrospective 38

VI Glossary 38

VII References / Bibliography 39

List of Figures

Figure 1 - Character, Shopkeeper, and Monster UML 8

Figure 2 - Level UML..... 9

List of Tables

Table 1 - Tutorial Functionality 13

Table 2 - Level One Functionality 17

Table 3 - Level Two Functionality 18

Table 4 - Level Three Functionality 20

Table 5 - Level Four Functionality 22

Table 6 - Tutorial Functionality Result 27

Table 7 - Level One Functionality Result 29

Table 8 - Level Two Functionality Result 30

Table 9 - Level Three Functionality Result 31

Table 10 - Level Four Functionality Result 32

Table 11 - Waiting Room Table 37

I Project Description

1 Project Overview

Dankest Dungeon is a web-based dungeon-crawler game. The player must traverse through each level of the dungeon, solving puzzles using various potions. They must also avoid monsters that can hurt them. The player is scored based on how fast they clear the levels, and they can win the game by clearing all the levels in the game.

2 Project Domain

The domain of our project consists of two main components: the server, and the client. The server is a simple web server, serving our game files to the client. The client consists of the client/user as well as the web browser they use to access the game.

The testing will primarily be done through the client side, with the web browser, testing various in-game actions and verifying that the behavior of the game matches desired output. There will also be some testing scripts, that verify that the game code on the server side updates internal data structures correctly when relevant functions are called.

3 Relationship to Other Documents

There are some relevant test scripts we have written for this report, located on our GitHub repository under the directory Develop/Test/.

4 Naming Conventions and Definitions

Ice Potion/Wall - Often interchangeable with Water Potion/Wall or Blue Potion/Wall, it refers to the blue colored potions or walls.

Acid Potion/Wall - Interchangeable with Green Potion/Wall, it refers to the green colored potions or walls.

Fire Potion/Wall - Interchangeable with Red Potion/Wall, it refers to the red colored potions or walls.

Explosion Potion/Wall - Interchangeable with Yellow Potion/Wall, it refers to the yellow colored potions or walls.

Healing Spring- Interchangeable with Health Fountain, it is the place where the player can attain more health.

Scene – The game is designed with scenes. A scene is a graphical layout, that can be swapped out. For example, Main Menu is a scene, each level is a scene, and the inventory is a scene.

4a Definitions of Key Terms

Score- A value determined by factors such as potions used, and time elapsed, to represent how well a player cleared the game.

Money- The money that the player, or shopkeeper, has throughout the game

Inventory – An array representing the potions that the player, or shopkeeper, has throughout the game

Sprite – The visual representation in game, of certain game objects. These game objects include, the character, shopkeeper, monsters, potions, walls, spikes, ladders, stairs, and the floor. Some can be overlapped over each other, some cannot.

Health Bar vs. Health - The 'health bar' consists of a set of 'heart' images signifying current health. 'Health' is the integer value of the health stored in code.

4b UML and Other Notation Used in This Document

This document generally follows the Version 2.5.1 OMG UML Standard

4c Data Dictionary for Any Included Models

Score (an integer, calculated as:) =

*(if(Time > 60) then Time*2)*

else if (Time > 0 then Time)

else (-Time)) + Walls Destroyed + Potions Picked-Up

Player Money (an integer, calculated as:)=

100 - (Money lost from buying Potions) +

(Money gained from selling Potions)

Shopkeeper Money (an integer, calculated as:) =

0 + (Money gained from selling Potions)

- (Money lost from buying Potions)

Health Bar (an integer, ranging from 0 to 100, beginning at 50)

Inventory (an array, consisting of Potion Objects)

Potion (an object, consisting of:) - Name(String), Img(Image)

II Testing

5 Items to be Tested

- WS: Web Server
- MM: Main Menu
- L: Levels
    - L-T: Tutorial
    - L-1: Level One
    - L-2: Level Two
    - L-3: Level Three
    - L-4: Level Four
    - L-S: Spikes
    - L-P: Potions
    - L-W: Walls
    - L-ST: Level State After Scene Changes
- G: Game Over Screen
- W: You Win Screen
- P: Player Character
    - P-H: Health
    - P-S: Score
    - P-M: Money
    - P-I: Inventory
    - P-MM: Menu
- S: Shopkeeper
    - S-M: Money
    - S-I: Inventory

- S-B: Buy
- S-S: Sell

- M: Monster

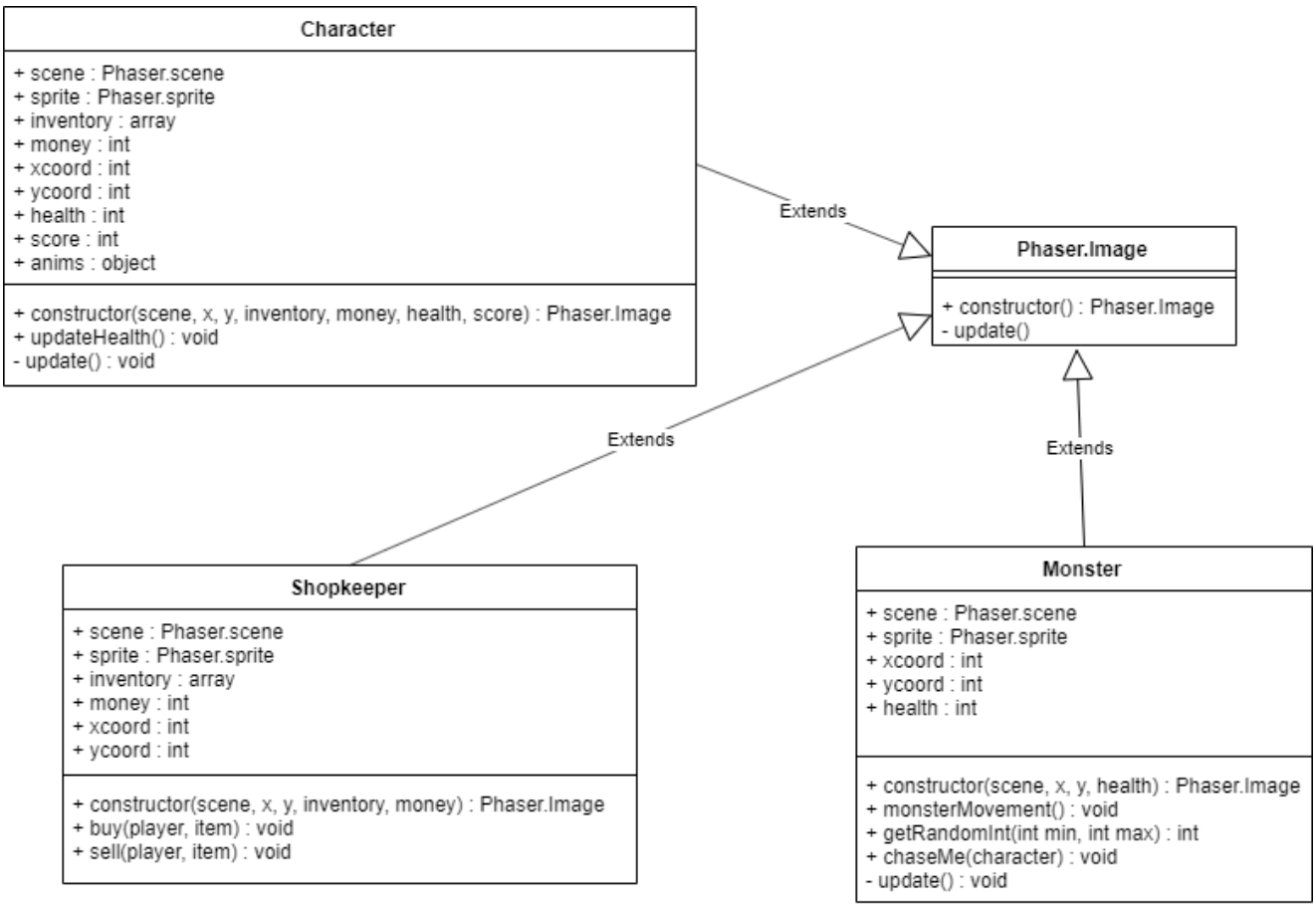

Figure 1 - Character, Shopkeeper, and Monster UML

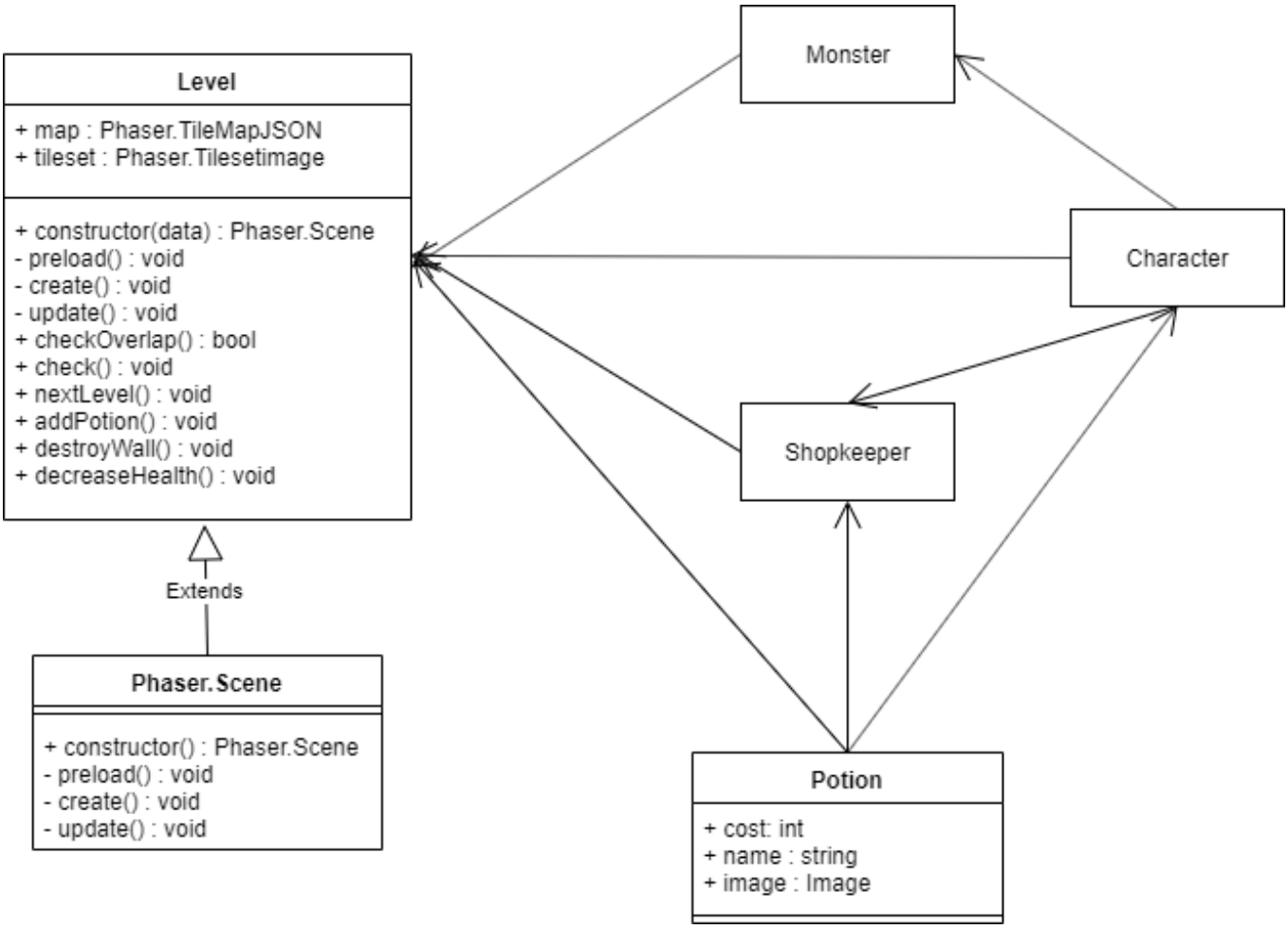

Figure 2 - Level UML

Test Specifications:

WS - 1: Web Server Startup

Description: Test to see if the server begins running without issues.

Items covered by this test: WS

Requirements addressed by this test: NA

Environmental needs: Server must be run on a Linux Machine, with NodeJS and SimpleHTTP Server installed. The server must have at least 1GB of RAM, a 2GHz CPU, and 1GB of Hard Drive Space.

Intercase Dependencies: NA

Test Procedures:

1. Open up a Terminal to the “Dankest Dungeon” Directory where the server files are located

2. Run “http-server”

3. Verify that the HTTP-Server has started via Terminal Output

Input Specification: Typing **“**http-server” in Terminal, in Game Directory

Output Specifications: Terminal Output, server running.

Pass/Fail Criteria:

Test Passes if Terminal Output begins with:

“Starting up http-server serving./”

“Available on:”

“http://127.0.0.1:8080”

MM – 1: Main Menu Loads

Description: Test if Main Menu Loads Properly

Items covered by this test: MM

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1

Test Procedures:

1. In Web Browser, type in URL”localhost:8080/test” and hit ‘Enter’

2. Observe to see if Main Menu Screen Loads Up

Input Specification: URL ”localhost:8080/test” in web browser

Output Specifications: Main Menu Scene displayed in web browser

Pass/Fail Criteria: If the Main Menu Screen loads up on the web browser screen with the brick background, dragon logo, purple scene background, and the words “Start Game”, then the test passes.

L-T - 1: Tutorial Loading

Description: Test if the tutorial level loads properly

Items covered by this test: L-T

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1

Test Procedures:

1. Run Test Case MM-1. Click "Start Game" on Main Menu Screen

2. Observe if the map loads, and if the tutorial start message is displayed.

Input Specification: One mouse click, on 'Start Game', in the web browser.

Output Specifications: Tutorial level displayed in web browser.

Pass/Fail Criteria: The test passes if the tutorial screen loads, the level displayed matches the tutorial map, and the message displayed at the top of the screen is:

"Welcome to the tutorial for Dankest Dungeon."

"Press X to continue"

L-T – 2: Tutorial Functionality

Description: Test if the functionality in the tutorial level works properly.

Items covered by this test: L-T, L-P, L-W, L-ST, G, P-H, P-I, S, M

Requirements addressed by this test: N/A

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1

Test Procedures:

1. Run Test Case L-T-1
2. Press X on the keyboard
3. Press arrow keys to move around
4. Press X

5. Move downward (using arrow keys) to next room
6. Press X
7. Move to overlap player sprite with shopkeeper. Press ‘A’
8. Click Exit in Shopkeeper Scene
9. Press X
10. Move downward to next room
11. Press X
12. Move to overlap player sprite with monster
13. Press X
14. Move downward to next room
15. Press X
16. Move to overlap player sprite with healing spring
17. Press X
18. Move downward to next room
19. Press X
20. Press I
21. Click Exit in inventory scene
22. Press X
23. Move downward to next room
24. Move so that player sprite is touching potion sprite
25. Press P
26. Press X
27. Move so that the player sprite is touching the green wall sprite
28. Press B
29. Press X

30. Move so that the player sprite touches the ladder

Input/Output Specification:

Table 1 - Tutorial Functionality

Input	Output
First ‘X’ Press	Display “User Arrow Keys to Move Around” message.
Arrow Key Input	Player character should move around the level.
Second ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Third ‘X’ Press	Display “Interact with Shopkeeper” message
Press ‘A’ near Shopkeeper Sprite	Switch Scene to Shopkeeper Scene
Click ‘Exit’	Return to tutorial screen
Fourth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Fifth ‘X’ Press	Display “Run away from monster” message
Move player sprite (using arrow keys) to overlap with monster sprite	Player health decreases

Sixth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Seventh ‘X’ Press	Display “This is a healing spring Message”
Move player sprite (using arrow keys) to overlap with healing spring	Player health increases
Eighth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Ninth ‘X’ Press	Display “Press I to open Inventory” message
Press ‘I’	Switch scene to inventory scene
Click ‘Exit’	Return to tutorial screen
Tenth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Eleventh ‘X’ Press	Display “Press P to pick up a potion” message
Press ‘P’ while player sprite is touching the potion sprite	Potion is removed from map. Potion appears below player

	health bar, signaling it is in inventory
Twelfth ‘X’ Press	Display “Press B to destroy wall” message
Press ‘B’ while player is touching wall sprite	Wall is destroyed on map. Potion disappears from below player health bar, signaling it is gone from the inventory
Thirteenth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Fourteenth ‘X’ Press	Display “Walk to ladder to go to level 1” message
Move player to touch ladder sprite	Game scene should shift to level 1

Pass/Fail Criteria: If all the steps of the procedure are followed, and the input/output specification match without any noticeable error, the test passes.

L-1-1 - Level One Loading

Description: Test if level one loads properly.

Items covered by this test: L-1

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2

Test Procedures:

1. Run Test Case L-T-2 and reach the end of the level.

2. Observe if the map loads, check if the inventory, money, score, and timer are displaying correctly.

Input Specification: Triggered by going through the ladder which is at the end of the Tutorial level

Output Specifications: Level one is displayed on the web browser.

Pass/Fail Criteria: The test passes if the Level One screen loads, and the level displayed matches the Level One map. The test fails if the game freezes or an asset does not load correctly.

L-1-2 - Level One Functionality

Description: Check to make sure Level One is fully playable.

Items covered by this test: L-1

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1

Test Procedures:

1. Setup an environment for Tape testing framework in game directory
2. Run test script 'LevelOne.test.js'
3. Run test case L-1-1
4. Move player character (using arrow keys) to bottom right of screen to border the Acid Potion
5. Press P
6. Move player character to top right of screen to border Acid Wall
7. Press B
8. Move player character to overlap staircase

Input/Output Specification:

Table 2 - Level One Functionality

Input	Output
Level One Object, sent to testing framework	List of how many tests in our script pass or failed
Arrow Key Input to move player to border Acid Potion	Player sprite borders Acid Potion
'P' Press	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Acid Wall	Player sprite borders Acid Wall
'B' Press	Acid Wall disappears from level map
Arrow Key Input to move player to staircase	Scene Change to Level Two

Pass/Fail Criteria: All of the tests from the test script must pass. Then if all of the remaining steps of the procedure are followed, and the input/output specification match without any noticeable error, the test passes.

L-2-1 - Level Two Loading

Description: Test if level two loads properly.

Items covered by this test: L-2

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2

Test Procedures:

1. Run Test Case L-1-2 and reach the end of the level.

2. Observe if the map loads, check if the inventory, money, score, and timer are displaying correctly.

Input Specification: Triggered by going through the staircase which is at the end of level one.

Output Specifications: Level Two is displayed on the web browser.

Pass/Fail Criteria: The test passes if the Level Two screen loads, and the level displayed matches the Level Two map. The test fails if the game freezes or an asset does not load correctly.

L-2-2 - Level Two Functionality

Description: Check to make sure Level Two is fully playable.

Items covered by this test: L-2

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2, L-2-1

Test Procedures:

1. Setup an environment for Tape testing framework in game directory
2. Run test script 'LevelTwo.test.js'
3. Run test case L-2-1
4. Move player character (using arrow keys) all the way to the right end of the level, to overlap the staircase.

Input/Output Specification:

Table 3 - Level Two Functionality

Input	Output
Level Two Object, sent to testing framework	List of how many tests in our script pass or failed
Arrow Key Input to move player to staircase	Scene Change to Level Three

Pass/Fail Criteria: All of the tests from the test script must pass. Then if all of the remaining steps of the procedure are followed, and the input/output specification match without any noticeable error, the test passes.

L-3-1 - Level Three Loading

Description: Test if level three loads properly.

Items covered by this test: L-3

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2, L-2-1, L-2-2

Test Procedures:

1. Run Test Case L-2-2 and reach the end of the level.

2. Observe if the map loads, check if the inventory, money, score, and timer are displaying correctly.

Input Specification: Triggered by going through the staircase which is at the end of level two.

Output Specifications: Level Three is displayed on the web browser.

Pass/Fail Criteria: The test passes if the Level Three screen loads, and the level displayed matches the Level Three map. The test fails if the game freezes or an asset does not load correctly.

L-3-2 - Level Three Functionality

Description: Check to make sure Level Three is fully playable.

Items covered by this test: L-3

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2, L-2-1, L-2-2, L-3-1

Test Procedures:

1. Setup an environment for Tape testing framework in game directory

2. Run test script ‘LevelThree.test.js’

3. Run test case L-3-1

4. Move player character (using arrow keys) to bottom left of screen to border the Fire Potion

5. Press P
6. Move player character to the middle right of the screen, to border Fire Wall
7. Press B
8. Move player character to overlap staircase

Input/Output Specification:

Table 4 - Level Three Functionality

Input	Output
Level Three Object, sent to testing framework	List of how many tests in our script pass or failed
Arrow Key Input to move player to border Fire Potion	Player sprite borders Fire Potion
'P' Press	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Fire Wall	Player sprite borders Fire Wall
'B' Press	Fire Wall disappears from level map
Arrow Key Input to move player to staircase	Scene Change to Level Four

Pass/Fail Criteria: All of the tests from the test script must pass. Then if all of the remaining steps of the procedure are followed, and the input/output specification match without any noticeable error, the test passes.

L-4-1 - Level Four Loading

Description: Test if level four loads properly.

Items covered by this test: L-4

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2, L-2-1, L-2-2, L-3-1, L-3-2

Test Procedures:

1. Run Test Case L-3-2 and reach the end of the level.

2. Observe if the map loads, check if the inventory, money, score, and timer are displaying correctly.

Input Specification: Triggered by going through the staircase which is at the end of level three.

Output Specifications: Level Four is displayed on the web browser.

Pass/Fail Criteria: The test passes if the Level Four screen loads, and the level displayed matches the Level Four map. The test fails if the game freezes or an asset does not load correctly.

L-4-2 - Level Four Functionality

Description: Check to make sure Level Four is fully playable.

Items covered by this test: L-4

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2, L-2-1, L-2-2, L-3-1, L-3-2, L-4-1

Test Procedures:

1. Setup an environment for Tape testing framework in game directory
2. Run test script ‘LevelFour.test.js’
3. Run test case L-4-1
4. Move player character (using arrow keys) to bottom left of screen to border the Ice Potion
5. Press P
6. Move player character to the top right of the screen, to border Ice Wall
7. Press B

8. Move player character (using arrow keys) to top right of screen to border the Fire Potion

9. Press P

10. Move player character to the bottom right of the screen, to border Fire Wall

11. Press B

12. Move player character to overlap staircase

Input/Output Specification:

Table 5 - Level Four Functionality

Input	Output
Level Four Object, sent to testing framework	List of how many tests in our script pass or failed
Arrow Key Input to move player to border Ice Potion	Player sprite borders Ice Potion
‘P’ Press	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Ice Wall	Player sprite borders Ice Wall
‘B’ Press	Ice Wall disappears from level map
Arrow Key Input to move player to border Fire Potion	Player sprite borders Fire Potion
‘P’ Press	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Fire Wall	Player sprite borders Fire Wall
‘B’ Press	Fire Wall disappears from level map
Arrow Key Input to move player to staircase	Scene Change to You Win screen

Pass/Fail Criteria: All of the tests from the test script must pass. Then if all of the remaining steps of the procedure are followed, and the input/output specification match without any noticeable error, the test passes.

G-1: Game Over Screen Loading

Description: Test if our game over screen loads properly, and is triggered when player health reaches 0

Items covered by this test: G, PH

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2

Test Procedures:

1. Run test case LT-2 up to step 'Fifth X Press'

2. Move the player to continuously overlap the monster sprite, until the player loses all their health

Input Specification: Arrow Keys, to move player.

Output Specifications: Player health decreases to 0. Game switches to game over scene.

Pass/Fail Criteria: If the game switches to the game over scene upon the player reaching 0 health, and a message stating 'Game Over' is displayed on the screen, then this test passes.

W-1: You Win Screen Loading

Description: Test if our you win screen loads properly, and that it is triggered when the player reaches the end of the last level.

Items covered by this test: W, L-4-2

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1, L-T-2, L-1-1, L-1-2, L-2-1, L-2-2, L-3-1, L-3-2, L-4-1, L-4-2

Test Procedures:

1. Run Test Case L-4-2

2. Observe scene change to you win screen

Input Specification: Arrow Keys, to move player.

Output Specifications: Game switches to you win scene.

Pass/Fail Criteria: The game should switch scenes from level four to the you win scene. The scene should display the message 'You Win', and the player's score should be displayed on the screen. If all these conditions are met, the test passes.

S-1: Shopkeeper Functionality

Description: Test if all of the shopkeeper functionality works properly

Items covered by this test: S, S-M, S-I, S-B, S-S

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: NA

Test Procedures:

1. Setup an environment for Tape testing framework in game directory

2. Run test script 'shopkeeper.test.js'

Input Specification: Shopkeeper object, sent to testing framework

Output Specifications: List of how many tests in our script pass or failed

Pass/Fail Criteria: All of the tests in our script should pass

P-1: Player Functionality

Description: Test if all of the player functionality works properly

Items covered by this test: P, P-H, P-S, P-M, P-I

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: NA

Test Procedures:

1. Setup an environment for Tape testing framework in game directory
2. Run test script 'Character.test.js'

Input Specification: Character object, sent to testing framework

Output Specifications: List of how many tests in our script pass or failed

Pass/Fail Criteria: All of the tests in our script should pass

P-MM-1: Player Menu Functionality

Description: Check to make sure the player menu loads correctly and displays the right content

Items covered by this test: P-MM

Requirements addressed by this test: NA

Environmental needs: Webserver should be running, need a modern web browser (such as Chrome, Edge, Firefox, or Safari).

Intercase Dependencies: WS-1, MM-1, L-T-1

Test Procedures:

1. In the tutorial, at any point before the end, press 'M'
2. In the Main Menu Scene, click 'Exit'

Input Specification: Keyboard press 'M', and mouse click on 'Exit' button.

Output Specifications: Game changes scene from tutorial level to player menu scene on 'M' press. Game changes back to tutorial level on 'Exit' click.

Pass/Fail Criteria: The game must switch scenes to the player menu scene upon pressing 'M'. The player's health, sprite, money, and inventory should be displayed on the screen. Clicking 'Exit' must return the game back to the tutorial. If all these conditions are met, the test passes.

6 Test Results

WS-1: Web Server Startup

Date(s) of Execution: 11/26

Staff conducting tests: Aashish Agrawal

Expected Results: Terminal output should begin with:

"Starting up http-server serving./"

"Available on:"

"http://127.0.0.1:8080"

Actual Results: Terminal output began with:

"Starting up http-server serving./"

"Available on:"

"http://127.0.0.1:8080"

Test Status: Pass

MM-1: Main Menu Loads

Date(s) of Execution: 11/26

Staff conducting tests: Zohar Sajith

Expected Results: Main Menu Scene displayed in web browser

Actual Results: Main Menu Scene displayed in web browser

Test Status: Pass

L-T-1: Tutorial Loading

Date(s) of Execution: 11/26

Staff conducting tests: Zohar Sajith

Expected Results: Tutorial level displayed in web browser.

Actual Results: Tutorial level displayed in web browser.

Test Status: Pass

L-T-2: Tutorial Functionality

Date(s) of Execution: 11/26

Staff conducting tests: Aashish Agrawal

Results:

Table 6 - Tutorial Functionality Result

Input	Expected Results	Actual Results
First ‘X’ Press	Display “User Arrow Keys to Move Around” message.	Display “User Arrow Keys to Move Around” message.
Arrow Key Input	Player character should move around the level.	Player character should move around the level.
Second ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Third ‘X’ Press	Display “Interact with Shopkeeper” message	Display “Interact with Shopkeeper” message
Press ‘A’ near Shopkeeper Sprite	Switch Scene to Shopkeeper Scene	Switch Scene to Shopkeeper Scene
Click ‘Exit’	Return to tutorial screen	Return to tutorial screen
Fourth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Fifth ‘X’ Press	Display “Run away from monster” message	Display “Run away from monster” message
Move player sprite (using arrow keys) to overlap with monster sprite	Player health decreases	Player health decreases

Sixth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Seventh ‘X’ Press	Display “This is a healing spring Message”	Display “This is a healing spring Message”
Move player sprite (using arrow keys) to overlap with healing spring	Player health increases	Player health increases
Eighth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Ninth ‘X’ Press	Display “Press I to open Inventory” message	Display “Press I to open Inventory” message
Press ‘I’	Switch scene to inventory scene	Switch scene to inventory scene
Click ‘Exit’	Return to tutorial screen	Return to tutorial screen
Tenth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Eleventh ‘X’ Press	Display “Press P to pick up a potion” message	Display “Press P to pick up a potion” message
Press ‘P’ while player sprite is touching the potion sprite	Potion is removed from map. Potion appears below player health bar, signaling it is in inventory	Potion is removed from map. Potion appears below player health bar, signaling it is in inventory
Twelfth ‘X’ Press	Display “Press B to destroy wall” message	Display “Press B to destroy wall” message
Press ‘B’ while player is touching wall sprite	Wall is destroyed on map. Potion disappears from	Wall is destroyed on map. Potion disappears from

	below player health bar, signaling it is gone from the inventory	below player health bar, signaling it is gone from the inventory
Thirteenth ‘X’ Press	The wall blocking the player from moving downward should disappear. Display “Move Down” message.	The wall blocking the player from moving downward should disappear. Display “Move Down” message.
Fourteenth ‘X’ Press	Display “Walk to ladder to go to level 1” message	Display “Walk to ladder to go to level 1” message
Move player to touch ladder sprite	Game scene should shift to level 1	Game scene should shift to level 1

Test Status: Pass

L-1-1: Level One Loading

Date(s) of Execution: 11/27

Staff conducting tests: Karan Ahuja

Expected Results: Level one is displayed on the web browser.

Actual Results: Level one is displayed on the web browser.

Test Status: Pass

L-1-2: Level One Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Aashish Agrawal

Expected/Actual Results:

Table 7 - Level One Functionality Result

Input	Expected Results	Actual Results
Level One Object, sent to testing framework	List of how many tests in our script pass or failed. All tests should pass	List of how many tests in our script pass or failed All tests passed
Arrow Key Input to move player to border Acid Potion	Player sprite borders Acid Potion	Player sprite borders Acid Potion

‘P’ Press	Potion disappears from map. Potion is added to player inventory	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Acid Wall	Player sprite borders Acid Wall	Player sprite borders Acid Wall
‘B’ Press	Acid Wall disappears from level map	Acid Wall disappears from level map
Arrow Key Input to move player to staircase	Scene Change to Level Two	Scene Change to Level Two

Test Status: Pass

L-2-1: Level Two Loading

Date(s) of Execution: 11/27

Staff conducting tests: Karan Ahuja

Expected Results: Level Two is displayed on the web browser.

Actual Results: Level Two is displayed on the web browser.

Test Status: Pass

L-2-2: Level Two Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Aashish Agrawal

Expected/Actual Results:

Table 8 - Level Two Functionality Result

Input	Expected Results	Actual Results
Level Two Object, sent to testing framework	List of how many tests in our script pass or failed All tests should pass	List of how many tests in our script pass or failed All tests passed

Arrow Key Input to move player to staircase	Scene Change to Level Three	Scene Change to Level Three

Test Status: Pass

L-3-1: Level Three Loading

Date(s) of Execution: 11/27

Staff conducting tests: Karan Ahuja

Expected Results: Level three is displayed on the web browser.

Actual Results: Level three is displayed on the web browser.

Test Status: Pass

L-3-2: Level Three Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Aashish Agrawal

Expected/Actual Results:

Table 9 - Level Three Functionality Result

Input	Expected Results	Actual Results
Level Three Object, sent to testing framework	List of how many tests in our script pass or failed All tests should pass	List of how many tests in our script pass or failed All tests passed
Arrow Key Input to move player to border Fire Potion	Player sprite borders Fire Potion	Player sprite borders Fire Potion
'P' Press	Potion disappears from map. Potion is added to player inventory	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Fire Wall	Player sprite borders Fire Wall	Player sprite borders Fire Wall

‘B’ Press	Fire Wall disappears from level map	Fire Wall disappears from level map
Arrow Key Input to move player to staircase	Scene Change to Level Four	Scene Change to Level Four

Test Status: Pass

L-4-1: Level Four Loading

Date(s) of Execution: 11/27

Staff conducting tests: Karan Ahuja

Expected Results: Level four is displayed on the web browser.

Actual Results: Level four is displayed on the web browser.

Test Status: Pass

L-4-2: Level Four Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Aashish Agrawal

Expected/Actual Results:

Table 10 - Level Four Functionality Result

Input	Expected Results	Actual Results
Level Four Object, sent to testing framework	List of how many tests in our script pass or failed All tests should pass	List of how many tests in our script pass or failed All tests pass
Arrow Key Input to move player to border Ice Potion	Player sprite borders Ice Potion	Player sprite borders Ice Potion
‘P’ Press	Potion disappears from map. Potion is added to player inventory	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Ice Wall	Player sprite borders Ice Wall	Player sprite borders Ice Wall

‘B’ Press	Ice Wall disappears from level map	Ice Wall disappears from level map
Arrow Key Input to move player to border Fire Potion	Player sprite borders Fire Potion	Player sprite borders Fire Potion
‘P’ Press	Potion disappears from map. Potion is added to player inventory	Potion disappears from map. Potion is added to player inventory
Arrow Key Input to move player to border Fire Wall	Player sprite borders Fire Wall	Player sprite borders Fire Wall
‘B’ Press	Fire Wall disappears from level map	Fire Wall disappears from level map
Arrow Key Input to move player to staircase	Scene Change to You Win screen	Scene Change to You Win screen

Test Status: Pass

G-1: Game Over Screen Loading

Date(s) of Execution: 11/27

Staff conducting tests: Karan Ahuja

Expected Results: When player health decreases to 0, game switches to game over scene.

Actual Results: When player health decreases to 0, game switched to game over scene.

Test Status: Pass

W-1: You Win Screen Loading

Date(s) of Execution: 11/27

Staff conducting tests: Karan Ahuja

Expected Results: When player clears level four, game should switch to the you win scene.

Actual Results: When player cleared level four, game switched to the you win scene.

Test Status: Pass

S-1: Shopkeeper Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Richard Miramontes

Expected Results: All of the tests in our script should pass

Actual Results: All of the tests in our script passed

Test Status: Pass

P-1: Player Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Richard Miramontes

Expected Results: All of the tests in our script should pass

Actual Results: All of the tests in our script passed

Test Status: Pass

P-MM-1: Player Menu Functionality

Date(s) of Execution: 11/27

Staff conducting tests: Aashish Agrawal

Expected Results: Game should switch to player menu screen, and the player's health, sprite, money, and inventory should be displayed on the screen.

Actual Results: Game switched to player menu screen, and the player's health, sprite, money, and inventory were displayed on the screen.

Test Status: Pass

7 Regression Testing

NA

III Inspection

8 Items to be Inspected

Tutorial Class, Level One Class, Monster Class, Level Four Class

9 Inspection Procedures

The first step each group member did was find the largest section of code that was primarily their own contribution. After identifying and isolating each chunk of code, we would bring our code to four meetings that we held, taking turns analyzing each section of code. The person who wrote the code would present their code on a TV screen, and highlight the goals of the code, and why it is organized the way it is. After that, the reviewers would split up and review the code, and regroup and give feedback in the next meeting.

When inspecting our code, we followed a general guideline where we looked at the following aspects: Was the code written following standard coding conventions? Is the code written where someone who has never looked at it could follow it? Are there any obvious bugs that we can find? Does the code that was written follow the style of our other members? Does the code reference the proper libraries? Then, each inspector would take turns stating their findings to the code presenter, and then the next code presenter would start.

10 Inspection Results

Inspection 1: November 22nd at 11:30 A.M.

- Inspectors: Aashish Agrawal, Karan Ahuja, Zohar Sajith
- Code Inspected: Level One Class, by Richard Miramontes
- Results: When analyzing the Level One class, we concluded that the level was written cleanly, with proper naming conventions for our classes that we had come up with in the beginning of our development process. We were all able to easily follow the code's control flow and understand how each function we defined operated in our game. The code was consistent with all our other level classes, making it even easier to understand. There was only one library we referenced, Phaser, and all our levels reference the library, so that condition was met as well. The only issue we found was that there were some minor inconsistencies in what data was being sent to different scenes. We noted that the information being sent to the shopkeeper scene or player menu scene, was not the same as the information being sent to the player inventory scene. In theory, the same data should be sent. After working on this, we found that this was a bug and we resolved it after the meeting.

Inspection 2: November 25th at 11:30 A.M.

- Inspectors: Karan Ahuja, Richard Miramontes, Zohar Sajith
- Code Inspected: Tutorial Class, by Aashish Agrawal
- Results: When analyzing the Tutorial class, we concluded that it was written cleanly, but it did not follow the same naming conventions as the other levels. We had to rename some of our variables and functions to match the rest of the levels. We believed this happened because the tutorial level was developed based on an older version of our initial Level One class. It was still easy to follow and understand, however. There was only one library we referenced, Phaser, and all our levels reference the library, so that condition was met as well. An additional flaw that was found was that, in his inspections, Karan found that monster movement did not seem as if it was mapped to the animations correctly. Upon further inspection, we found that this was true. The monster sprite was moving the opposite direction than what was intended in some cases. The team came together to rectify the situation and we were able to map the animations with the monster properly.

Inspection 3: November 26th at 11:30 A.M.

- Inspectors: Aashish Agrawal, Richard Miramontes, Zohar Sajith
- Code Inspected: Monster class, by Karan Ahuja
- Results: The code seemed to follow all our standard coding conventions and was easy to follow. There was an obvious bug we found in our inspection, however. An open issue that was already present in our game was that when the server was kept open for more than 5 minutes, we found out the game would go through unbearable lag. Moving the character was near impossible as the screen would take a couple of seconds before recognizing input. While we had no idea what was causing the issue, our open issues were kept in mind during our inspections. While inspecting, Aashish found out that the Monster's AI movement was updating way too frequently, and this could potentially be an issue with game lag. After the inspection, the team worked on resolving that issue. Surprisingly, our open issue was resolved after this, our game continued to perform as intended even after 5 minutes passed.

Inspection 4: November 27th at 11:30 A.M.

- Inspector: Aashish Agrawal, Richard Miramontes, Karan Ahuja
- Code Inspected: Level Four class, by Zohar Sajith
- Results: The code seemed to follow all our standard coding conventions and was easy to follow. One obvious bug that Richard found during inspections was that the games currency and score were not updating correctly for this specific level. In order to figure out what was happening, we had to go over

the currency methods that kept track and updated when an event happened and found out that in our level design, the events were not being registered to the methods. We as a team had to rewrite the methods that Zohar wrote which fixed our issue. Just in case, after this we checked the other levels to make sure that the issue was not repeated anywhere else.

IV Recommendations and Conclusions

All our testing procedures passed, as we intended for them to. Despite this however, we found new bugs during our inspection process. We may have resolved these bugs, but this tells us that our testing procedures may not have been extensive enough. For the future, we will have to come up with more extensive testing procedures and test more boundary cases. Inspections seem to do a good job of finding errors our testing procedures could not, so we hope to schedule more inspections in the future as well.

V Project Issues

11 Open Issues

There are some issues we encountered during testing with data not being passed properly between scenes. We believe there are few more issues like that left.

There is also another issue which affects the flat inventory display. Sometimes, the potions a player picks up or uses will not update the graphical inventory display, even though internally the correct changes are made.

12 Waiting Room

Table 11 - Waiting Room Table

Requirement	Priority	Intended Version Number
More Monster Variety	Very High	4
More Levels	Very High	4
Health Potion	Medium	6
Ghost removing potions from player when they collide	High	5
Combat, wherein player can kill monsters	Medium	6

Add more information to the player menu	Low	5
Improve the player inventory menu	Low	5

13 Ideas for Solutions

Some ideas that could be included later include implementing a meme mode. This would just be for fun, and it would be unlocked by the player if they can find a secret passage in a random level. This feature would change the visuals of the game, making it so that there are memes littered throughout the whole game. Another potential idea would be to move our game to a proper web server, instead of running it only on a local machine. We could also work on using the Mocha or JEST testing framework, as those are much more popular than the one we used (Tape).

14 Project Retrospective

One thing that worked well were that throughout the coding process, informal inspections and testing were done at regular intervals. By doing this, it helped in finding the bugs in our game a lot sooner. Also, another thing that went well was the usage of publicly available royalty-free sprites as it helped in creating new levels easier. None of us had any artistic talent.

Something that did not work well was our usage of the Phaser gaming library itself. For many of the features we wanted to implement in our game, it was hard to troubleshoot or find documentation that we needed. In the end, much of it we had to write functions ourselves. In the future, we could work on trying to find a better gaming library or developing the game in a different language altogether. Phaser also made it difficult to test our game, as we had many issues making Tape collaborate properly with Phaser. In the end, we had to download a local copy of the Phaser codebase to make the test cases work, but this would in practice greatly slow down the game.

VI Glossary

Acid Potion/Wall: *Interchangeable with Green Potion/Wall, it refers to the green colored potions or walls.*

Explosion Potion/Wall: *Interchangeable with Yellow Potion/Wall, it refers to the yellow colored potions or walls.*

Fire Potion/Wall: *Interchangeable with Red Potion/Wall, it refers to the red colored potions or walls.*

Healing Spring: *Interchangeable with Health Fountain, it is the place where the player can attain more health.*

Ice Potion/Wall: *Often interchangeable with Water Potion/Wall or Blue Potion/Wall, it refers to the blue colored potions or walls.*

Monster: *Refers to the ghost that attacks the player.*

Player: *Refers to the user that plays the game.*

Scene: *The game is designed with scenes. A scene is a graphical layout, that can be swapped out. For example, Main Menu is a scene, each level is a scene, and the inventory is a scene.*

References / Bibliography

[1] "16x16 DungeonTileset II." *Itch.io*, https://0x72.itch.io/dungeontileset-ii.

The tilemap we used for the levels

[2] Blade, Night. "Animated Roguelike/RPG Monsters from Eman Quest." *OpenGameArt.org*, OpenGameArt, 6 Sept. 2019, https://opengameart.org/content/animated-roguelikerpg-monsters-from-eman-quest.

The tilemaps we used for the player, shopkeeper, and monster

[3] West, Parker. "Fantasy Setting Background." *Pixabay*, Shutterstock, https://pixabay.com/illustrations/fantasy-setting-background-backdrop-3334834/.

The background image used for our main menu

[4] "Stones Wall Quarry Stone." *Pixabay*, Shutterstock, https://pixabay.com/photos/stones-wall-quarry-stone-texture-770264/.

The background image used for our overarching game

[5] Mezheritskiy, Arthur, et al. *Dungeons and Dank Technical Report*. Spring 2018, *Dungeons and Dank Technical Report*.

The development report made by Group 14 in Spring 2018