***Escape Room Game***

***Testing and Inspection Report***

**CS 440 - Group 4**

**Luqmaan Baiyat**

**Joshua Herman**

**Hamza Shahid**

**Patrick GundryTable of Contents**

[List of Figures](#_3znysh7)  3

[List of Tables](#_3znysh7) 4

[Project Description](#_tyjcwt) **5**

[Project Overview.](#_3dy6vkm) 5

[Project Domain](#_1t3h5sf)  5

[Relationship to Other Documents](#_4d34og8) 5

[Naming Conventions and Definitions](#_2s8eyo1) 5

[Testing](#_lnxbz9) **5**

[Items to be Tested](#_35nkun2) 5

[Test Specifications](#_1ksv4uv) 6

[Regression Testing](#_2jxsxqh) 22

[Inspection](#_z337ya) **2**2

[Items to be Inspected](#_1y810tw) 23

[Inspection Procedures](#_4i7ojhp) 24

[Inspection Results](#_2xcytpi) 24

[Recommendations and Conclusions](#_1ci93xb) **28**

[Project Issues](#_3whwml4) **2**7

[Open Issues](#_1pxezwc) 27

[Ideas for Solutions](#_dnydbbd7p05r) 28

[Project Retrospective](#_ykj4cnsqqilr) 28

[Glossary](#_2p2csry) **29**

[References / Bibliography](#_147n2zr) **29**

### **List of Figure**s**List of Tables**

Table 1-1 7

Table 1-2 8

Table 1-3 10

Table 2-1 24

Table 2-2 26

Table 2-3 27

# Project Description

## Project Overview. The project is an escape room game in which the player(s) must solve puzzles and find clues in order to progress within the level. The player is able to choose different levels that that they would like to solve. Some levels will be locked until the player completes the prerequisite level(s).

## Project Domain The domain of the project is digital entertainment. For this project, the required features must be properly operational before the the project is released in order to ensure the best experience of the player(s).

## Relationship to Other Documents This document is related to *Escape Room Project Report* by Patel, Fan, and Douailly-Backman.

## Naming Conventions and Definitions Refer to *Escape Room Project Report* by Patel, Fan, and Douailly-Backman for all naming conventions, definitions, and diagrams.

# Testing

## Items to be Tested

1. Main Menu Buttons
2. Level Selection
3. Player Movement
4. Wall Collision
5. Level Progress
6. Game Timer
7. Points System
8. Hints System
9. Lock Puzzle Level 1
10. Floor Tile Puzzle Level 1
11. Lever Puzzle Level 1
12. Anagram Puzzle Level 2
13. Sqeuence Puzzle Level 2
14. Floor Tile Puzzle Level 2
15. Riddle Puzzle Level 2
16. Lever Puzzle Level 2

## Test Specifications

**ID #1 - Main Menu Buttons**

**Description:** Ensuring main menu buttons work properly.

**Items covered by this test:** Item 1

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Not applicable.

**Test Procedures:**

1. Launch application.
2. Click on button labeled *Start Game*.
3. Close the application and relaunch.
4. Click on the button labeled *Settings*.
5. Click on the button labeled *Exit Game*.

**Input and Output Specifications:**

Table 1-1

|  |  |
| --- | --- |
| **Input** | **Output** |
| Clicking *Start Game* Button | Being redirected to level selection |
| Clicking *Settings* Button | Nothing should happen (settings unimplemented) |
| Clicking *Exit Game* Button | Application closes |

**Pass/Fail Criteria:** The pass criteria is completing all of the procedures in order with the specified inputs resulting in the specified outputs. The fail criteria is any action of a specified input that results in the wrong output.

**ID #2 - Level Selection**

**Description:** Ensure only level 1 is unlocked in the beginning. As levels are completed only then the next level of the game is unlocked.

**Items covered by this test:** Item 2

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 1.

**Test Procedures:**

1. Navigate to level selection screen.
2. Click on the button labeled *Level 1*.
3. Navigate to level selection screen.
4. Click on the button labeled *Level 2*.
5. Navigate to level selection screen.
6. Click on the button labeled *Level 3*.
7. Navigate to level selection screen.
8. Click on the button labeled *Level 4.*

**Input Specification:** Clicking the mouse on each of the appropriate buttons

**Output Specification:** The corresponding level beginning as a response to its button click.

**Pass/Fail Criteria:** The test passes if every button click is mapped to the correct level beginning.

**ID #3 - Player Movement**

**Description:** Ensure player is able to move around the rooms in the game.

**Items covered by this test:** Item 3

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 1 and Test 2

**Test Procedures:**

1. From level selection screen, select Level 1.
2. Press W key.
3. Press A key.
4. Press S key.
5. Press D key.

**Input and Output Specifications:**

Table 1-2

|  |  |
| --- | --- |
| **Input** | **Output** |
| W key | Player icon moves up |
| A key | Player icon moves left |
| S key | Player icon moves down |
| D key | Player icon moves right |

**Pass/Fail Criteria:** The pass criteria is that all of the specified inputs map to the specified outputs. If any of the inputs lead to incorrect outputs, the test has failed.

**ID #4 - Wall Collision**

**Description:** Ensure player is not able to go through any of the walls in the rooms. Also make sure that the player can only go to the next room if and only if the door is unlocked.

**Items covered by this test:** Item 4

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Passing of Test 1, Test 2 and Test 3

**Test Procedures:**

1. From level selection screen, select Level 1.
2. Using the player movement key “A”, go near walls to the left of the room.
3. Using the player movement key “D”, go near walls to the right of the room.
4. Using the player movement key “S”, go near walls to the down of the room.
5. Using the player movement key “W”, go near walls to the up of the room.

**Input Specification:** Getting the player next to the walls in the room.

**Output Specifications:** Ensure player cannot go through any of the walls in the room.

**Pass/Fail Criteria:** The pass criteria is that the player cannot go through the walls in any of the rooms. Player must solve a puzzle for a particular room which will then open a door through which player can go through to the next room.

**Pass/Fail Criteria:** The pass criteria is that the player must solve puzzle for room 1 in order to go to the next room. The player has to solve all puzzles for a Level in order to go to play the next Level.

**ID # 5 - Level Progress**

**Description:** Make sure player goes through the different levels of the game in ascending order until all levels are unlocked.

**Items covered by this test:** Item 4

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 3, Test 4, and Test 5

**Test Procedures:**

1. Begin Level 1 from the level selection screen.
2. While inside of Level 1, complete the stages of the room.
3. Begin Level 2 from the level selection screen.
4. While inside of Level 2, complete the stages of the room.

**Input Specification:** Make the player solve all the puzzles in one Level.

**Output Specifications:** Player solves puzzles in one Level, then goes to the next Level.

**Pass/Fail Criteria:** The pass criteria is that the player must complete the first Level of the game in order to go play the next Level of the game.

**ID # 6 - Game Timer**

**Description:** Make sure the game time works properly. When player pauses the game, make sure game timer is also paused, and when player resumes, make sure game timer continues from where it left of.

**Items covered by this test:** Item 6

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 2

**Test Procedures:**

1. Begin Level 1.
2. Allow the game timer of Level 1 to run down to 0:00.
3. Begin Level 2.
4. Allow the game timer of Level 2 to run down to 0:00.

**Event and Consequence Specifications:**

Table 1-3

|  |  |
| --- | --- |
| **Event** | **Consequence** |
| Level 1 timer running out | Game displaying Game Over screen |
| Level 2 timer running out | Game displaying Game Over screen |

**Pass/Fail Criteria:** The test is considered passing if both instances of the level’s timer running out results in the Game Over screen being displayed. If either level’s game timer runs any further than allowed, the test has failed.

**ID #7 - Points System**

**Description:** Ensure there is a point system for each of the level 2 of the game and it gives the correct number of points.

**Items covered by this test:** Item 7

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Completion of Level 2, test 2 and 5.

**Test Procedures:**

1. Begin Level 2
2. Win Level 2 at 2, 4, 6, and 8 minutes in
3. Level Completed screen pops up
4. Check that points are displayed
5. Compare points to actual number of points they should have received
6. Begin Level 2
7. Let time run out
8. Level Failed Screen Pops up
9. 0 points completed

**Input Specification:** Make the player win the level at 2, 4, 6, and 8 minutes. Make player let time run out

**Output Specifications:** Player sees points displayed on the Level Completed Screen. Player sees 0 points displayed at Level Failed Screen

**Pass/Fail Criteria:** The pass criteria is that the player must be able to see their points for the level in the Level Completed screen and they player must be given the correct amount of points for completing the level at that time based on our equation. The pass criteria is also the player sees 0 points and a level failed screen when they run out of time.

**ID #8 Hints System**

**Description:** Ensure there are hints for each of the playable levels of the game, and only displaying when once the ‘show hints’ button is enabled.

**Items covered by this test:** Item 8

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 2 and 5

**Test Procedures:**

1. Begin Level 1
2. Click toggle hints
3. Check hint is being displayed at this progression
4. Check that the correct hint is being displayed at the correct time
5. Click toggle hints and check that hints are not being shown
6. Continue to next progression and do 2-6 until level is completed
7. Begin Level 2
8. Click toggle hints
9. Check hint is being displayed at this progression
10. Check that the correct hint is being displayed at the correct time
11. Click toggle hints and check that hints are not being shown
12. Continue to next progression and do 8-12 until level is completed

**Input Specification:** Make the player begin level 1, then toggle hints, then toggle hints again, then continue to the next progression starting with the first toggle hints. Then make the player go through level 2, and turn on hints.

**Output Specifications:** When users presses toggle hints for the first time player sees hints displayed on the right panel, and the correct hints being displayed for each progression and puzzle. When the user presses toggle hints the second time hints are hidden. Move onto the next progression and check again

**Pass/Fail Criteria:** The pass criteria is that the player must be able to see the hints on the right panel of the game when they toggle hints on and must not be able to see hints when they toggle hints off. They also must be able to see the correct hints at the correct progressions, for both levels.

**ID #9 - Lock Puzzle Level 1**

**Description:** Ensure the lock puzzle asks for different random numbers when played more than once, and functions correctly when the right number is guessed.

**Items covered by this test:** Item 9

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 2 and 5

**Test Procedures:**

1. Begin Level 1
2. Start Room 1
3. Walk to the drawer in Room 1 of this level
4. Guess the correct number from 0-9
5. Grab the key once correct with space
6. Open door to Room 2
7. Restart Level
8. Repeat steps 1-5 10 times ensure at least 3 different numbers are shown

**Input Specification:** Make the player begin level 1, walk to the drawer of room 1 of this level, guess the correct number from 0-9, grab the key, and open the door to room 2.

**Output Specifications:** When the user guesses the correct number they will receive a key.

**Pass/Fail Criteria:** The pass criteria is that the player must receive a key after guessing the right number. The player also must be able to grab this key, and open the door to room 2. The numbers also are randomly generated so the test expects to see at least 3 different numbers through 10 iterations to pass.

**ID #10 - Floor Tile Puzzle Level 1**

**Description:** Make sure that the sequence of the floor tile puzzle is random if played more than once. Also, make sure that once solving, the whole player goes through the floor tile inorder for it to register as being stepped on.

**Items covered by this test:** Item 10

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 2, 5, and 9

**Test Procedures:**

1. Begin Level 1
2. Pass Room 1
3. Start Room 2
4. Walk to floor tile puzzle
5. Solve the puzzle
6. Receive key
7. Open door to Room 3
8. Restart Level
9. Repeat steps 1-8 5 times ensure different patterns each time

**Input Specification:** Make the player begin level 1, pass room 1, and solve the floor tile puzzle in room 2, and open the door for room 3

**Output Specifications:** When the user guesses the correct pattern they will receive a key.

**Pass/Fail Criteria:** The pass criteria is that the player must receive a key after guessing the correct pattern of the floor tile puzzle. Once they receive this key they must be able to open the door for Room 3. Lastly, the pattern must be different through the 5 iterations of tests as the pattern is randomly generated.

**ID #11 - Lever Puzzle Level 1**

**Description:** Ensure all three levers are random if played again, and make sure the correct sequence will open the door

**Items covered by this test:** Item 11

**Environmental needs:** Not applicable.

**Intercase Dependencies:**  Passing of Test 2, 5, 9, 10

**Test Procedures:**

1. Begin Level 1
2. Pass Room 1
3. Pass Room 2
4. Walk to lever puzzle
5. Solve the puzzle
6. Door opens exit level
7. Restart Level
8. Repeat steps 1-7 10 times ensure at least 3 different combinations are seen

**Input Specification:** Make the player begin level 1, pass room 1, pass room 2, solve the lever puzzle and exit the level.

**Output Specifications:** When the user guesses the correct pattern the door will open and when they exit the lever the are shown the Level Completed Screen

**Pass/Fail Criteria:** The pass criteria is that the player must see the door open when they solve the puzzle. Once the door opens they must be able to exit the level and see the exit screen. Lastly, the pattern must be different at least 3 times through the 10 iterations of tests as the pattern is randomly generated.

**ID #12 - Anagram Puzzle Level 2**

**Description:** Ensure the anagram puzzle is different when player plays more than once, and make sure the puzzle can be solved with proper input

**Items covered by this test:** Item 12

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Passing of test 2, 5, 9, 10, 11

**Test Procedures:**

1. Begin Level 2.
2. Start Room 1.
3. Walk towards the green tile that appears on the top right of the room.
4. Guess the correct word from the given letters.
5. Open door to Room 2.
6. Restart Level.
7. Repeat steps 1-6 5 times to ensure the anagram puzzle is different atleast 2 times.

**Input Specification:** Make the player begin level 2, assuming that the player has completed level 1. Make the player walk to the green tile in the top right and put in the correct answer.

**Output Specifications:** When the player guesses the correct word, they receive a key.

**Pass/Fail Criteria:** The pass criteria is that the player must be able to solve the anagram puzzle by guessing the correct word from the given letters inorder to open the door to room 2. They need to be able to put in any case insensitive text with the correct answer and must be able to pass. The anagrams also are randomly generated so the test expects to see at least 2 different words through 5 iterations to pass.

**ID #13 - Sequence Puzzle Level 2**

**Description:** Ensure the sequence puzzle is different when the player plays more than once, ensure when the play puts the correct answer the door opens and they can leave the room.

**Items covered by this test:** Item 13

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Passing of test 2, and 5

**Test Procedures:**

1. Begin Level 2.
2. Pass Room 1.
3. Start Room 2.
4. Walk towards the red tile that appears on the top right of the room.
5. Guess the next number in the sequence.
6. Open Room 3.
7. Restart Level.
8. Repeat steps 1 - 7 to ensure sequence puzzle is different atleast 2 times for 5 iterations

**Input Specification:** Make the player begin level 2, assuming that the player has completed level 1. Pass room 1, pass room 2, walk to the red tile, solve the sequence puzzle, open the door to room 3, and restart level.

**Output Specifications:** When the player guessed the next number in the sequence correctly, they receive a key.

**Pass/Fail Criteria:** The pass criteria is that the player must be able to solve the sequence puzzle by guessing the next number in the sequence correctly and receive the key to advance. The player then must be able to open to door to room 3. The numbers also are randomly generated so the test expects to see at least 2 different sequences through 5 iterations to pass.

**ID #14 - Floor Tile Puzzle Level 2**

**Description:** Ensure that the sequence of the floor tile puzzle is different if the player plays more than once. In addition, once playing, make sure sure that the whole player goes through the floor tile inorder for it to register as being stepped on.

**Items covered by this test:** Item 14

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Passing of test 2, 5, 9, and 13

**Test Procedures:**

1. Begin Level 2.
2. Pass Room 1.
3. Pass Room 2.
4. Start Room 3.
5. Walk to the floor tile puzzle.
6. Solve the floor tile puzzle.
7. Open Room 4.
8. Restart Level.
9. Repeat steps 1 - 8 to ensure floor tile puzzle is different atleast 4 times through 10 iterations

**Input Specification:** Make the player begin level 2, assuming that the player has completed level 1. Pass room 1, pass room 2, solve the floor tile puzzle in room 3, walk into room 4.

**Output Specifications:** When the player guesses the correct pattern, the next door will open automatically.

**Pass/Fail Criteria:** The pass criteria is that the player must see the door open when they solve the puzzle which will allow them to enter room 4. Also the tile should light up when it is walked on and it should reset when the user guesses incorrectly. The sequence of going through the tiles is randomly generated so the test expects to see at least 4 different sequences through 10 iterations to pass.

**ID #15 - Riddle Puzzle Level 2**

**Description:** Make sure the riddle puzzle is different when the player plays more than once. Ensure that when the user puts the correct answer they can advance to the next room.

**Items covered by this test:** Item 15

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Passing of test 2, 5, 9, 13, and 14

**Test Procedures:**

1. Begin Level 2.
2. Pass Room 1.
3. Pass Room 2.
4. Pass Room 3.
5. Start Room 4.
6. Walk towards the orange tile that appears in the room.
7. Solve the riddle puzzle.
8. Open Room 5.
9. Restart Level.
10. Repeat steps 1 - 9 to ensure floor tile puzzle is different atleast 5 times.

**Input Specification:** Make the player begin level 2, assuming that the player has completed level 1. Pass room 1, pass room 2, pass room 3, solve the riddle puzzle, walk into room 5.

**Output Specifications:** When the player guessed the riddle correctly, the next door will open automatically.

**Pass/Fail Criteria:** The pass criteria is that the player must be able to solve the riddle puzzle by guessing the riddle correctly and when guessed correctly the door will automatically open. The user must be able to guess the correct answer case insensitive and pass. The riddles are randomly generated so the test expects to see at least 2 different riddles through 5 iterations to pass.

**ID #16 - Lever Puzzle Level 2**

**Description:** Ensure all three levers are random if played again. Ensure player can complete the puzzle and exit the level.

**Items covered by this test:** Item 16

**Environmental needs:** Not applicable.

**Intercase Dependencies:** Passing of test 2, 5, 13, 14, and 15

**Test Procedures:**

1. Begin Level 2.
2. Pass Room 1.
3. Pass Room 2.
4. Pass Room 3.
5. Pass Room 4.
6. Start Room 5.
7. Walk towards the lever puzzle.
8. Solve the lever puzzle.
9. Door opens exit level.
10. Restart Level.
11. Repeat steps 1 - 10 to ensure floor tile puzzle is different atleast 3 times for 5 iterations.

**Input Specification:** Player should begin level 2, assuming 1 has been completed. Then pass room 1, 2, 3, and 4. Start room 5, walk to the level puzzle and solve it, then the user should exit the level.

**Output Specifications:** When the player guesses the correct pattern the door will open, and when they exit the level, they are shown the Level Completed Screen along with their points for that level.

**Pass/Fail Criteria:** The pass criteria is that the player must see the door open when they solve the puzzle. Once the door opens they must be able to exit the level and see the exit screen. Lastly, the pattern must be different at least 3 times through the 5 iterations of tests as the pattern is randomly generated.

**ID #1 - Main Menu Buttons**

**Date(s) of Execution:** 4/17/19

**Staff conducting tests:** Joshua Herman

**Expected Results:** Pass

**Actual Results:** Pass

**Test Status:** Pass, button clicks result in appropriate results.

**ID #2 - Level Selection**

**Date(s) of Execution:** 4/17/19

**Staff conducting tests:** Hamza Shahid

**Expected Results:** Pass.

**Actual Results:** Pass.

**Test Status:** Pass, the Level selection buttons work appropriately. Assuming the previous Levels are completed, player is able to select the next Level and play it.

**ID #3 - Player Movement**

**Date(s) of Execution:** 4/18/19

**Staff conducting tests:** Luqmaan Baiyat

**Expected Results:** Pass

**Actual Results:** Pass

**Test Status:** Pass, player’s movement of left, right, down up position is precise. When “A” is pressed, player moves to the left. When “D” is pressed, player moves to the right. When “S” is pressed, player moves down. When “W” is pressed, player moves up.

**ID #4 - Wall Collision**

**Date(s) of Execution:** 4/17/19

**Staff conducting tests:** Joshua Herman

**Expected Results:** Pass

**Actual Results:** Fail

**Test Status:** Fail, there is no proper wall collision when the player is in between doors.

**ID #5- Level Progress**

**Date(s) of Execution:** 4/17/19

**Staff conducting tests:** Luqmaan Baiyat

**Expected Results:** Pass.

**Actual Results:** Pass.

**Test Status:** Pass, the Level selection buttons work appropriately. Initially only one Level is available for selection. As player successfully completes a Level, the next Level is available for selection

**ID #6 - Game Timer**

**Date(s) of Execution:** 4/17/19

**Staff conducting tests:** Hamza Shahid

**Expected Results:** Expecting test to have a pass result

**Actual Results:** Test has passed

**Test Status:** Pass, Game Over screen appeared when both Level 1 and 2 timers reached 0:00.

**ID #7 - Points System**

**Date(s) of Execution:** 4/18/19

**Staff conducting tests:** Joshua Herman

**Expected Results:** Expecting test to have a pass result

**Actual Results:** Test has passed

**Test Status:** Pass, the player sees their points for the level in the Level Completed screen and the player was given the correct amount of points for completing the level at that time based on our equation.

Pass, the player sees the level failed screen and 0 points for when the time ran out.

**ID #8 - Hints System**

**Date(s) of Execution:** 4/18/19

**Staff conducting tests:** Joshua Herman

**Expected Results:** Expecting test to have a fail result

**Actual Results:** Test has failed

**Test Status:** Pass, in level 1 the player sees the hints on the right panel of the game when they toggle hints on and does not see hints when they toggle hints off. The player is able to see the correct hints at the correct progressions.

Fail, in level 2 the player does not see hints on the right panel when toggling. Hints have not been implemented for this level.

**ID #9 - Lock Puzzle**

**Date(s) of Execution:** 4/18/19

**Staff conducting tests:** Joshua Herman

**Expected Results:** Expecting test to have a pass result

**Actual Results:** Test has passed

**Test Status:** Test has passed, the player receives a key after guessing the right number. The player also is able to grab this key, and open the door for Room 2. The numbers were randomly generated as there were 7 different numbers to guess throughout the 10 iterations, which passed the threshold of 3.

**ID #10 - Floor Tile Puzzle Level 1**

**Date(s) of Execution:** 4/18/19

**Staff conducting tests:** Luqmaan Baiyat

**Expected Results:** Expecting test to have a pass result

**Actual Results:** Test has passed

**Test Status:** The test has passed, the player receives a key after guessing the correct pattern of the floor tile puzzle. Once they received this key they were able to open the door for Room 3. Lastly, the pattern was different through each of the 5 iterations.

**ID #11 - Lever Puzzle Level 1**

**Date(s) of Execution:** 4/19/19

**Staff conducting tests:** Joshua Herman

**Expected Results:** Expecting test to have a pass result

**Actual Results:** Test has passed

**Test Status:** The test has passed, the player sees the door open when they solve the puzzle. Once the door opened they were able to exit the level and see the exit screen. Lastly, the pattern was different at least 3 times through the 10 iterations of tests.

**ID #12 - Anagram Puzzle Level 2**

**Date(s) of Execution:** 4/22/2019

**Staff conducting tests:** Hamza Shahid

**Expected Results:** Pass the result.

**Actual Results:** Test passed successfully.

**Test Status:** This test passed successfully. No matter what the case sensitivity is of the input, the player always sees the door open when he/she solves the anagram puzzle successfully. If the player guesses it incorrectly, he/she can guess the word again as many times as he/she likes. Once door opens, the next room is no longer hidden for the player. The anagram was randomly generated from an ArrayList and was different atleast twice during the 5 iterations.

**ID #13 - Sequence Puzzle Level 2**

**Date(s) of Execution:** 4/22/2019

**Staff conducting tests:** Hamza Shahid

**Expected Results:** Pass the result.

**Actual Results:** Test passed successfully.

**Test Status:** This test passed successfully. The next door opens once the player guesses the sequence puzzle correctly. The player is able to guess the number as many times until he/she gets it right. They then receive the key and can open the next door. When the door opens, the next room is no longer hidden for the player, and the previous rooms also remain visible for the player. If played again, the player would be asked a different sequence puzzle. This was the case atleast 2 times during the 5 iterations.

**ID #14 - Floor Tile Puzzle Level 2**

**Date(s) of Execution:** 4/22/2019

**Staff conducting tests:** Joshua Herman

**Expected Results:** Pass the result.

**Actual Results:** Test passed successfully.

**Test Status:** This test passed successfully. The third door opens once the player guesses the floor tile puzzle correctly. The player has unlimited attempts to guess the pattern of the floor tile puzzle. The tiles were reset each time a wrong path was taken. Once the player gets the sequence correct, the door automatically opens, and the next room is no longer hidden for the player along with previous rooms being visible. The sequence of the floor tile puzzle was different 4 times out of the 10 iterations.

**ID #15 - Riddle Puzzle Level 2**

**Date(s) of Execution:** 4/22/2019

**Staff conducting tests:** Luqmaan Baiyat

**Expected Results:** Pass the result.

**Actual Results:** Test passed successfully.

**Test Status:** This test passed successfully. The fourth door opens once the player guesses the riddle puzzle correctly. As soon as the riddle is guessed all rooms of the level are visible for the player. We were also able to do case insensitive answer and it still worked. The player can then go to the final room of the level. This riddle puzzle was different atleast 2 times out of the 5 iterations of the level.

**ID #16 - Lever Puzzle Level 2**

**Date(s) of Execution:** 4/22/2019

**Staff conducting tests:** Joshua Herman

**Expected Results:** Pass the result.

**Actual Results:** Test passed successfully.

**Test Status:** This test passed successfully. Once the player solves the lever puzzle, the final door opens. The player goes through the final door which ends level 2, and is taken to the exit screen. Player’s score is displayed for that level as well. This lever puzzle was different atleast 3 times out of the 5 iterations.

## Regression Testing

Not applicable.

# Inspection

## Items to be Inspected

Luqmaan Baiyat - Abstract Level Class

Joshua Herman - Progression System and Hints

Hamza Shahid - Level Puzzles

## Inspection Procedures

Ensure good readability of code

Check for meaningful comments

Check for meaningful variable and function names

Ensure proper usage of data structures

## Inspection Results

**Luqmaan Baiyat:**

Table 2-1

|  |  |
| --- | --- |
| **Contributor - Item(s)** | **Results of Procedures** |
| Joshua Herman - Game Progression and Hints  Inspection performed on 4/17/19 | 1. Readability of code was very easy to follow. Proper indenting was used as line breaks. 2. Code was heavily commented, which was beneficial to understanding what each segment of code was used for. 3. Meaningful variable names and function names utilized. 4. Appropriate usage of Array Lists and necessary functions to add, delete, and access elements within those Array Lists. |
| Hamza Shahid - Level Puzzles  Inspection performed on 4/17/19 | 1. Good readability of code. Appropriate spacing and proper indenting. 2. Comments were easy to follow and relevant. 3. Function and variable names were meaningful and relevant to the context in which they were used. 4. Usage of a Hash Table for Riddle Puzzle seemed unnecessary. Data structure usage for other puzzles comprised of Array Lists which had practical implementation for the puzzle applications. |

**Joshua Herman:**

table 2-2

|  |  |
| --- | --- |
| **Contributor - Item(s)** | **Results** |
| Luqmaan Baiyat - Abstract Level Class | 1. The code is extremely readable as there are great variable and function names throughout. The code itself has good comments in most parts which aides in the readability, but it could use more. For the functions that weren’t abstract the detectCollision was written extremely clean as the switch statements for directions made it simple to understand how it worked. Likewise all of the functions for drawing things were simple to read as they were written clean and concise. 2. The first thing I notice missing for comments is a description of what this class does at the top. This would be a good addition to get an understanding of the class’s purpose. There is a good amount of consistent comments throughout. I really like how the abstract methods had an explanation of why they would be overridden. The Level constructor needs comments as it will define the order that the room gets made, but it is not completely necessary due to good variable names. Missing comments for several functions like resetLevel and drawDoors and drawRoomCovers, but mostly self explanatory with proper variable names. 3. Every function and variable name is descriptive and very good. I can understand what everything does by just looking at the variable name and function. With names like addHints, addWalls, gameProgession etc. the code becomes a lot easier to read and clean. 4. For data structures he used ArrayLists for the levels. the walls, the rooms, and the doors. This was a perfect usage for this implementation as all we needed was a way to group these elements together to hold these objects. |
| Hamza Shahid - Level Puzzles | 1. The code is readable most parts as it is split into easy to understand functions and classes with specific purposes for each. To make it more readable it could use more comments in some parts. At times you used inheritance for some puzzles when you should have used composition as the child class was not using anything from the parents, or rather take advantage of the code reuse. The data structures for the most part are simple to understand and the classes represent the real world objects so it makes it more readable. 2. Lacking comments on about half of the puzzles especially on some of the word puzzles. They all have similar functionality however so it is not completely necessary as you can understand them all by understanding one. Also there are good variable and function names On the ones with comments I like how there is a comment on the top of the classes explaining the purpose of the classes. Also like how there are descriptions for the functions and in-line comments explaining the difficult parts of the code. Makes these simple to understand. There are functions scattered throughout that also require comments. 3. Overall you had very good function and variable names as it allowed me to read through the code and not have to go through extra steps to figure out what variables were doing. They were descriptive of their purpose and function with their names alone and made them self-explanatory. Functions and variables like randomlyDisplayRiddle, addHints, and checkIfCorrect, word, anagram etc. are great naming conventions. 4. Used ArrayLists for every puzzle when applicable and it was a good use of the data structure as it allowed random selections of puzzles and good organization. The one issue is for the RiddlePuzzle you decided to use a HashMap, and it works but I feel it is an unnecessary use of a data structure when you could have used an ArrayList. The issue here is in the Level class it is called differently than the other ArrayList based puzzles which makes the code harder to read. Either all of these text-based puzzles should have been ArrayLists or HashMaps. The one benefit to the HashMap is that it can be done in all classes rather than making an ArrayList of Riddle objects we can store it directly in the data structure, but on the downside it is harder to work with and it is the one implementation that differs from the rest. |

**Hamza Shahid:**

Table 2-3

|  |  |
| --- | --- |
| **Contributor - Item(s)** | **Results** |
| Joshua Herman - Game Progression and Hints  Inspection Performed on April 24, 2019 | 1. The readability of the code was quite good and easy to follow. There was consistent indentation throughout the code as well as consistent name scheme. 2. Comments were placed in appropriate places throughout the code. There was not an overuse of comments where understanding the code was quite obvious. 3. The function and variable names were meaningful and showed consistency with the naming. 4. The data structure used for this was an ArrayList which was appropriate in this situation given that it made accessing elements of the progression and hints simple. |
| Luqmaan Baiyat - Abstract Level Class  Inspection Performed on April 24, 2019 | 1. There was good readability of code. The code was well organized with consistency in spacing and indentation. Similar variable names were placed together. 2. The use of comments was consistent throughout the code, and made it easier in understanding what was going on. No overuse of comments where understanding the code was simple. 3. Function and variable names were meaningful and showed consistency. 4. The use of ArrayList data structure was appropriate as it was used for walls, rooms, doors, and list of levels. It also made accessing elements in the ArrayList quick and simple without writing to much code. |

# Recommendations and Conclusions

*SV: Indicate whether or not the items covered have passed their testing and inspection process or not, and what actions should be taken next. ( E.g. further testing & inspection, or implementation. )*

# Project Issues

## Open Issues

When implementing levels for 2 players we have not implemented collision between the 2 players, at the current moment they are able to walk over each other. In addition, we have not resolved the issue that will allow them to race as they both cannot use the spacebar to interact with objects, so player 2 will need a new key binding. They also cannot both use the keyboard at the same time in order to put in inputs for word puzzles so this is another issue that will need to be resolved. In 2 player co-op there are all of these same issues.

The hints system is not working for the second level so it is sitting in code ready to be used, it just has to be implemented properly in the abstract level class or the hints invocation needs to be moved elsewhere. We also have several puzzles ready to be used in the code like a substitution cipher and a binary word puzzle that have the infrastructure, but just need to be included in the GUI. We also do not have a points system for all levels so this can be easily implemented for the other levels by looking its implementation in level 2. Another fix required is the collision when walking through doors as players can go through the walls when going through the door and the collision is no completely accurate with the GUI. Likewise, the floor tile puzzles do not have completely accurate bounds when the user interacts with it as they can cut through part of and still light it up for some, but not for others.

## Ideas for Solutions

For implementing 2 player collision, the players need to be aware of each other’s location and cannot walk into their location within their square bounds. For implementing 2 player game functionality, bind the P key for player 2 to interact with objects like the spacebar works for player 1. For implementing 2 players racing to type, you can allow the players to switch off taking turns with who can type. So if player 1 gets there first they have ability to type in their answer, and after they guess it switches to player 2 and vice versa. For co-op typing it does not have to switch and the player that is in the proper section can type since they are working together.

For fixing the hints system for the second level and so on the addHints function should be taken out of the abstract Level class and only be added to the respective levels. The error was being caused by the hints being asked for before the word objects were being created so it was asking for a hint that did not exist. Level 1 depends on the calling of this in the parent class, but all that needs to be done is pull them down to their respective classes as the hints are added at separate times for each so the invocation must be different.

For adding the puzzles all that needs to be done is create the objects in the GUI, in this case both of them need a text box that can take in user input from the keyboard. Have the player walk to a certain square on the map where the text box will open up and ask them for their answer. Then the program must call the proper functions that are already implemented to do a text comparison to check if the user has input the correct solution. If so give them a key or have the door automatically open. For adding points system from level 2 for all levels, all that needs to be done is to copy the function from level 2 into the level that is scaled off of the time left and put it in its respective class. Then it needs to be called at the end of the level and displayed to the user.

For collision through the doors the collision system needs to be implemented for the walls in between the doors. All that needs to be done is that walls in between doorways need to prevent players from walking through them and the players cannot be allowed to walk through these as well. How this works can be seen in the code for normal collision that works on every other wall. The last fix is for the floor tile puzzle where the bounds are not completely accurate for the player walking through them. All that needs to be done is redo the bounds so that it will not light up unless a player completely walks into it and it will be accurate for all of them.

## Project Retrospective

Throughout the course of the project, we made sure to have meetings where each subject of the meeting was discussed only as long as it was needed, be it short or long. Additionally, we held group calls online where we collaborated on the coding project. Combining in person meetings and online meetings helped us combine strategy and implementation for the coding project.

Our biggest challenge during this project was consistently integrating everyone’s changes flawlessly using the Git repository. There were many instances of merge issues, as well as having one member’s changes being overwritten by another member’s changes. We feel this is a result of lack of communication on who should be working on which parts of the code, as well as not using the branching feature of Git. For example, our Level 1 and Level 2 classes required everyone to work on them to test their own code’s functionality, but changes were committed that overwrote other people’s changes. We then had to checkout from previous commits and correct our mistakes. More frequent integration, usages of Git branches, and communication about who was allowed to edit which pieces of code at a certain would fix this issue.

For consideration, we were assigned as a group of 4 students for this coding project, but only 3 people contributed any actual work. We are tremendously disappointed with Patrick, as he has literally no work to show for the project. As as group, we have made it clear to each other if anyone needed help, they are always welcome to ask anyone else in the group, if the person they were asking was able to provide it. With that being said, Patrick has stated during meetings he had issues with completing his assigned parts, but never asked for anyone’s help. Moving on, while integrating our code, Patrick said he had issues with pushing his code to our group’s Git repository. To address this issue, he was told he can copy and paste his code and someone else in the group would commit the changes on his behalf. Even with that opportunity, Patrick never sent his code to the group.

# Glossary

## Refer to *Escape Room Project Report* by Patel, Fan, and Douailly-Backman for all glossary materials

# References / Bibliography

|  |  |
| --- | --- |
|  | Patel, Fan, and Douailly-Backman, Escape Room Project Report |
|  |  |
|  |  |
|  |  |