Product	Design Inspection		
Date	9/11/2013		
Author	Xinpi Du, Art Malinin		
Moderator	Jun Xiang Tee		
Inspectors	Lirong Yuan, Matt Walters, Jun Xiang Tee, Xinpi Du, Wei Haow Tan		
Recorder	Lirong Yuan		
Defect Number	Description	Severity	How corrected
1	The character model can move past enemy units at the cost of one life. The user can use number of remaining lives of the character model to complete a level (kind of cheating).	1	Design such that the character model is teleported back to the start port. This will serve as a penalty that motivates the user to think of a better way to complete the level.
2	The enemy units can move past the doors. This undermines importance of the doors, as they serve as a shortcut to the exit port that avoids the enemy units.	2	Design such that only the character model can move past the doors. This will make the doors serve a more important role, which makes the marks deduction for opening them worthy.
3	Users can sign up for an account by typing a username consisting of numbers only. This is unconventional. Normally, a username consists of a combination of letters, numbers, and special characters. Besides, the username is normally begun with a letter.	2	Added a check to ensure that username starts with letters. Add a requirement in the database tablesetup that requires usernames be letters, numbers, and special characters.
4	Scores are ordered based on the time it was inserted to the database(chronologically). Further discussion yields a better idea: sort the scores.	2	Edited score queries to select scores ordered from the highest to the lowest. The top 5 scores will be shown on the score list on the main menu.
5	A user account can be opened more than once simultaneously.	1	In the database, set a flag for each user account that verifies whether the user account is currently opened or not. The flag is turned on when the user signs into the account, and turned off when the user logs out from the account.
6	Current game does not contain many enemies. Users can pass all the levels without making any efforts.	2	Added more enemies to all levels. The enemy units are scattered evenly on the map, covering the different paths.
7	Some levels are not well designed so that half of the screen is not used.	3	Redesigned the levels to make use of at least 60% of the whole screen. The size of the first few levels is enlarged accordingly to meet this requirement.
8	The start port does not have a unique picture. Users may not know where they enter the level.	1	Added the image of the start port and display it in the program. The image is different from that of the exit port.

9	The exit port does not have a unique picture. Users may not know where the character model should go to.	2	Added the image of exit port and display it in the program. The image is different from that of the start port.
10	The character model does not have different images depending on the direction it faces.	3	Added four images of character and changed its image based on the direction. This allows the user to notice the direction of the character model is facing, facilitating game playing.
11	The enemy does not have different images depending on the direction it faces.	3	Added four images of enemy and change its image based on the direction. This allows the user to notice the direction of the enemy units, facilitating game playing.
12	There is no path way to the exit port/treasure on some maps.	1	Make sure there is at least one path to the exit port/treasure when design each map. Double check to ensure that the exit port / treasure is not blocked completely by the walls.
13	There is no safe path that reaches the exit port and treasure on some levels as the path is blocked by moving enemy units.	1	Make sure there is at least a safe path that reaches the exit port and treasure on each level when assign locations of the enemy units of the levels.
Product	**Code Inspection**		
Date	9/11/2013		
Author	Xinpi Du		
Moderator	Jun Xiang Tee		
Inspectors	Lirong Yuan, Art Malinin		
Recorder	Lirong Yuan		
Defect Number	Description	Severity	How corrected
1	Enemy Paths/positions should be set in level build methods.	1	There are existing methods where each level is layed out. Enemy paths and positions should be hard coded in the same method.
2	Locations of walls should be stored in arrays per level to make checking for collisions easier.	2	In the methods where the walls are created, the locations of the walls should also be added to an array.
3	Enemy Paths should be set for each level, so that you do not need to check for enemy collision with walls.	2	The paths of each enemy should be hard coded in the same method where the level is created. This will insure a custom path for each enemy in each level
4	Enemy Locations should always be stored and updated, so it can easily checked against character model.	2	Create an array for in each level with enough space for each enemy in that level. The enemy locations will be updated in the array, and the engine will check the character location against this array of enemy locations
5	Setting walls manually can lead poor code readibility as well as potential problems down the line.	3	One way is to use a configuration file. Have the program read the levels from a text file and convert it into graphics.

6	Everything is in one class. Character, map, and enemy are all in the same class. Hard to read code, and hard to distinguish what function belongs to which part.	3	Separate into three classes: map, character, and enemy. This follows the rules of object-oriented programming paradigm, making the coding tasks easier.
7	The update score function is unnecessary. top scores will be loaded every time the menu screen is opened.	2	Delete the update score function, and make changes to other affected functions accordingly. Make sure that the top scores are loaded successfully when the menu screen is opened.
8	Initialized the array to all 0 in the constructor. Global variables in java are initilized to 0 by default.	3	Remove the initialization. Unnecessary initialization makes the code harder to read.
9	The password is not being encrypted. This leads to potential security flaws.	1	Hash the passwords. Use the Java build in function to hash the passwords. This will also require us to un-hash the passwords.
10	User and password variables are package private. This makes the get methods useless as well as imposes a security risk.	1	Make the variables private. By this way, not everyone will have access to these variables. This will reduce the security risk too.
11	setPassword and setUsername functions seem useless as the constructor requires parameters of the username and password. However, the parameters are set directly without calling the functions.	3	Remove the two functions or set those variables in the constructor. Then, have the constructor to initialize the variables.
12	The GUI window will not disappear after the user successfully login.	1	Added setVisible(false) to the code after the execution of the successful login.
13	The score will not be saved if the program is closed during a game.	1	Save game score to database after each level is passed. Any game termination before completing a level will result in lost of the score.
14	User is allowed to log in if password is incorrect.	1	Added password check. Based on the database, compare the account username with the inputted password.
15	The user is able to sign up for a username even when the username is already in the database.	1	Before adding the username and password to the database, checked if the username already exists in the database first. If the username exists, the sign up process fail. The user is showed with a message indicating the issue, and prompted to enter a different username.
16	Top 5 scores are currently hardcoded.	1	Added a SQL query to store all scores in an array. Then, sort the array based on scores, and display the top 5.
17	The password isn't being encrypted. This leads to potential security flaws.	1	Hash the passwords before storing them to database. The passwords are unhashed later on for future retrieval.

18	Did not check for password length when signing in. The program is vulnerable to SQL injection attack.	1	Check for password validity before executing the SQL query. Set up constant variables indicating the minimum and maximum password lengths for verifying the password length.
19	Did not check for special characters in username.	2	Check for special characters in username. Ensure the username does not begin with a special character.
Product	**Unit Testing**		
Date	9/11/2013		
Author	Xinpi Du		
Moderator	Matt Walters		
Inspectors	Lirong Yuan, Art Malinin, Wei Haow Tan, Matt Walters, Jun Xiang Tee		
Recorder	Lirong Yuan		
Defect #	Description	Severity	How Corrected
1	After clicking on play button, program will not jump to game screen.	1	Add an action listener for the play button.
2	Buttons on menu bar does not have correct responses.	1	Add action listeners for the menu bar buttons.
3	After clicking on play button, the menu window will not disappear.	3	Add setVisible(false) after the execution of the play button.
4	The score board did not update immediately when the user uploaded a new high score.	1	Update the score board if the scoreboard shows. Create a refresh function that updates the score board everytime the main menu is accessed (The score board is shown at the main menu).
5	Login and create user buttons do not work.	1	Added listeners to access database to get or set information, and check it against user input. Ensure a particular user account exists for login purpose. Ensure a particular user account has not yet created when signing up.
6	The exit button does not work.	3	Added listeners to close the program.
7	The switch user button does not work.	3	Added listeners to logout and jump to login page again.
8	The enemy model moves past walls.	1	Added checks for walls whenever location of the character model changes. If the positions of the walls and the enemy model are the same, set the wall to be impermeable for the enemy model.
9	The character model moves past the right map boundary and down boundary.	1	Add boundary check so the character model cannot move through it.
10	The character model throw an execption when moves out of the up and left map boundary	1	Keep checking coordination of character model to make sure it is positvie
11	The character model is not affected by hazards.	1	Added tests for hazard collision to the engine.

12	When the user completes the last level, nothing happens.	1	Checked whether the current level completed is the last level. If so, link it to ending screen.
13	The character model cannot move past the doors.	1	Set permeability of the doors to be true for the character model.
14	The enemy units move past the doors.	2	Set permeability of the doors to be false for the enemy units.
15	The doors do not disappear when the character model walks past them.	2	Checked whether current coordinates of the character model is the same as those of the doors. If they are the same, set visibility of the doors to be invisible.
16	Game does not end when a character collide with an enemy when the character has only one life left.	1	Check against enemy positions whenever character position changes. End the game and jump to main menu if two are at the same position.
17	Game does not move to the next level when the charater model steps on the exit.	1	Change game level if character moves to the exit.
18	The character model stuck on the walls.	1	Added checks for character model location to make sure it does not embed in any walls
19	The treasure will not disappear after the character model walks through it.	2	Checked whether current location of the character model is the same as the treasure. If they are the same, set the treasure to be invisible.
20	After clicking the radio buttons for the "easy", "medium", and "hard" levels, all the radio buttons can be selected and cannot be deselected later on.	1	Check the conditions of the enability of the radio button using if-else statement. For example, if "easy" level radio button is selected, the "medium" and "hard button" should be in setSelected(false).
21	Level difficulty radio buttons are not mutually exclusive. Users can select more than one button.	2	Select one radio button should deselect other buttons.
22	The user can sign in by merely entering username.	1	Added a check to ensure password correct.
23	Users can sign up for an account using password with invalid length.	1	Checked to make sure that length of the password is between 4 and 15 before proceeding. If the length of the password is out of the range, display a warning message and prompt the user to reenter his or her username and password.
24	Nothing happens when the switch user button is pressed. The user cannot log out from the game.	2	Correct listener for the switch user button to correctly bring up the login screen.
25	Scores of other users are also displayed wrongly in current user score section.	1	Queried the scores with the correct foreign key originated from the current user account.
26	The character model moves too slowly when difficulty level is high. It's too hard for users to pass the level.	1	Increased the speed of model character while keeping differences between different difficulty levels.

27	The enemy moves too fast when difficulty level is high. It's too hard for users to pass the level.	1	Decreased the speed of enemy in general while keeping differences between different difficulty levels.
28	When the arrow keys are pressed, the character model is not able to move.	1	Passed new location information to the game engine.
29	The character model does not turn smoothly at corners.	1	Refined character model size to allow it to turn at corners smoothly.
30	Users cannot sign up for accounts.	1	Added the sign up page to the GUI.
31	The GUI window of the menu will not disappear after the user clicked the "Play" button.	1	Added setVisible(false) to the code after the execution of the play button.
32	Nothing happens when the exit button is pressed.	3	Correct listener for the exit button so that the program can be closed successfully. Besides, set the default close operation of JFrame to be JFrame. EXIT_ON_CLOSE.
33	The doors do not disappear after the character model walk through them.	2	Need to add code to replace the door model with a floor model after the character model walks through it.
34	The display refreshes in a slow manner.	3	Set the timer to be shorter in duration. A shorter timer increases the display speed.