VORMUND

Black Box Test Plan

Team 6

Seth Cook
Margarita Lee Li
Samuel Marshall
Adam Rice
Mitchell Stendal
John Vall

Test Plan Requirements

Requirement: The user will input their chosen username and password to the GUI in order to create a new account.

- Functionality Test Cases
    - Login Test 001 - Account Creation

      Severity: 2
      Instructions: Enter a username and password that are both within the [5-25] character length for text fields within the system.
      Expected Result: Account creation will proceed normally, and the login window will close, revealing the game play window.
    - Login Test 002 - Account Persistence

      Severity: 2
      Instructions: Once a new account has been created, exit the system and then reopen it, this time navigating to the “Sign in” tab at the top of the login window to enter the previously used credentials.
      Expected Result: The user should be successfully logged into the system and the login window will close, revealing the game play window.
    - Login Test 003 - Incorrect Username

      Severity: 3
      Instructions: Attempt to login to the system using a username that does not yet exist
      Expected Result: The user should be prompted to input another username
    - Login Test 004 - Incorrect Password

      Severity: 3
      Instructions: Attempt to login to the system using an incorrect password
      Expected Result: The user should be prompted to input another password
- Equivalence Test Cases
    - Login Test 005 - Proper username input

      Severity: 2
      Instructions: Input a username that is within [5-25] characters in length.
      Expected Result: Account creation will proceed as normal, and the login window will close, revealing the game play window.
    - Login Test 006 - Proper password input

      Severity: 2
      Instructions: Input a password that is within [5-25] characters in length.
      Expected Result: Account creation will proceed as normal, and the login window will close, revealing the game play window.

  - Login Test 007 - Bad username under 5 characters
    Severity: 2
    Instructions: Enter a username that is less than 5 characters in length
    Expected Result: User is alerted to input a username that is between 5 and 25 characters in length
  - Login Test 008 - Bad username over 25 characters
    Severity: 2
    Instructions: Enter a username that is greater than 25 characters in length
    Expected Result: User is alerted to input a username that is between 5 and 25 characters in length.
  - Login Test 009 - Bad password under 5 characters
    Severity: 2
    Instructions: Enter a password that is less than 5 characters in length
    Expected Result: User is alerted to input a password that is between 5 and 25 characters in length
  - Login Test 010 - Bad password over 25 characters
    Severity: 2
    Instructions: Enter a password that is greater than 25 characters in length
    Expected Result: User is alerted to input a password that is between 5 and 25 characters in length
- Boundary Value Test Cases
  - Login Test 011 - Blank entry account creation
    Severity: 2
    Instructions: Attempt to create an account without inputting any values for the username and password fields.
    Expected Result: The system will halt account creation and prompt the user to enter username and password values that are within the [5-25] character length interval allowed by the program.
  - Login Test 012 - Bad password length of 4
    Severity: 2
    Instructions: Attempt to create an account with a password that only consists of 4 characters--a value on the boundary of the equivalence class.
    Expected Result: The system will not allow account creation to continue, and the user will be prompted to enter a password that is at least 5 characters and at most 25 characters in length.
  - Login Test 013 - Bad username length of 4
    Severity: 2
    Instructions: Attempt to create an account with a username that only consists of 4 characters--a value on the boundary of the equivalence class.

    Expected Result: The system will not allow account creation to continue, and the user will be prompted to enter a username that is at least 5 characters and at most 25 characters in length.

- Login Test 014 - Max username length

  Severity: 2

  Instructions: Attempt to create an account with a username that consists of 25 characters--a valid equivalence class boundary value.

  Expected Result: The system will correctly validate the username field data, and account creation will proceed normally.

- Login Test 015 - Max password length

  Severity: 2

  Instructions: Attempt to create an account with a password that consists of 25 characters--a valid equivalence class boundary value.

  Expected Result: The system will correctly validate the password field data, and account creation will proceed normally.

Requirement: The user can also choose game difficulties and can access old scores

- Functionality Test Cases
  - Difficulty 001 - Radio Button Selection

    Severity: 2

    Instructions: Select any of the three difficulty options (Easy, Medium, or Hard) that are displayed as radio buttons at the top of the main screen for Dungeon Explorer

    Expected Result: Once a new difficulty has been selected, all other difficulties except for the once newly selected by the user should be unselected, such that only one difficulty is ever selected at one given time

  - Difficulty 002 - Manifestation of difficulty level

    Severity: 2

    Instructions: Run the game with easy mode selected, once more with medium selected, a finally with hard selected as the difficulty.

    Expected Result: The user should be able to observe a discernable difference in the speed of the both the character model and of the enemies on the map. With each level of increasing difficulty, the speed of the character should decrease, whereas those of the enemies should increase.

  - High Scores 001 - Global Bests

    Severity: 3

Instructions: Play several games under multiple test user accounts, keeping track of the scores that were achieved during each play through

Expected Result: When logged into the system from any user account, the "Global Bests" section should display the cross-account user high scores in descending order.

- High Scores 002 - Personal Bests

  Severity: 3

  Instructions: Play multiple games after having created a new user account.

  Expected Result: The "Personal Bests" section of the Dungeon Explorer screen should display the top scores achieved by the actively logging in player in descending order.

Requirement: The game display will pass data about the location of character model and the engine will determine how the character model interacts with the environment.

- Functionality Test Cases
  - Collision Test 001 - Walls in Map

    Severity: 2

    Instructions: Run the character into the walls in the interior of the map.

    Expected Result: The character should stop and not be able to advance any further.
  - Collision Test 002 - Walls on perimeter

    Severity: 2

    Instructions: Run the character into the walls on the perimeter of the map.

    Expected Result: The character should stop and not be able to advance any further.
  - Collision Test 003 - Treasure Functionality

    Severity: 3

    Instructions: Run the character into a piece of treasure.

    Expected Result: The treasure should disappear and a 20% bonus should be given to the score.
  - Collision Test 004 - Enemy Functionality

    Severity: 3

    Instructions: Run the character into an enemy.

    Expected Result: The character should be unable to move and the fight dialogue should be presented.

    - Collision Test 005 - End of level collision

        Severity: 2

        Instructions: Take you character to the end of the level.

        Expected Result: The character should be warped to the beginning of the next level or the game should end if the player is on the last level. The players points should also be appropriately updated.
- Equivalence Test Case
    - *Not applicable for this requirement*
- Boundary Value Test Cases
    - *Not applicable for this requirement*

Requirement: The GUI will retrieve the user's data from the database.

- Functionality Test Cases
    - Database Test 001 - User login information being retrieved from Database

        Severity: 1

        Instructions: Enter a previously created username and password correctly within the [5-25] character length for text fields within the system.

        Expected Result: The username and password will be found in the database, login will proceed as normal, game play window will open as the login window closes. Previous saved scores will be showing if the user has played a game before.
- Equivalence Test Cases
    - Database Test 002 -Check for correct existing Username with Database

        Severity: 1

        Instructions: Input a username that is within [5-25] characters in length and has been created already.

        Expected Result: The username will be found in the database, login will proceed as normal as long as the password matches too, game play window will open as the login window closes.
    - Database Test 002 -Check for correct existing password with corresponding username in the Database

        Severity: 1

        Instructions: Input a password that is within [5-25] characters in length that has been created already.

        Expected Result: The password will be found in the database, login will proceed as normal as long as the username matches it too, game play window will open as the login window closes.
    - Database Test 003 -Check if username exist in the database

Severity: 1
Instructions: When login, input a username that is within [5-25] characters in length but has yet to be added in the database.
Expected Result: A window will pop-up telling the user that the username doesn't exist in the database and needs to create an account.

    - Database Test 004 -Check for duplication of username
      Severity: 2
      Instructions: When login, input a username that is within [5-25] characters in length but is already in the system.
      Expected Result: A window will pop-up telling the user that the username does exist in the database and needs to change it.
    - Database Test 005 -Check for incorrect password with correct username in database
      Severity: 2
      Instructions: Attempt to login with a password that does not match the username that was typed.
      Expected Result: The system will not allow account login to continue, and the user will be prompted to enter the correct password that matches the username typed.
- Boundary Value Test Cases
    - Database Test 007 -Comparing blank entries with the Database entries
      Severity: 1
      Instructions: Attempt to login without inputting any values for the username and password fields.
      Expected Result: The system will halt account login and prompt the user to enter username and password values that are within the [5-25] character length interval allowed by the program.

Requirement: The game engine will transfer the high score to the database.

- Functionality Test Cases
    - Database Test 001 -New high score entry in easy/medium/difficult difficulty for personal score
      Severity: 2
      Instructions: Completed the 10 levels on easy/medium/difficult difficulty with a new high score for personal score.
      Expected Result: The new high score will be added in the database for personal score and the main window will be up again, the game window will close.
    - Database Test 002 -New high score entry in easy/medium/difficult difficulty for

overall score list

Severity: 2

Instructions: Completed the 10 levels on easy/medium/difficult difficulty with a new high score.

Expected Result: The new high score will be added in the database for overall score and the main window will be up again, the game window will close.

  - Database Test 003 - New high score entry in easy/medium/difficult difficulty for both overall and personal list

    Severity: 2

    Instructions: Completed the 10 levels on easy/medium/difficult difficulty with a new high score for both overall and personal list of scores.

    Expected Result: The new high score will be added in the database for both list and the main window will be up again, the game window will close.

  - Database Test 004 - New score entry in easy/medium/difficult difficulty for both overall and personal list when all scores are 0

    Severity: 2

    Instructions: Completed the 10 levels on easy/medium/difficult difficulty with a new high score for overall or personal list of scores.

    Expected Result: The new score will be added in the database for personal and overall scores, and the main window will be up again, the game window will close.

- Equivalence Test Cases
  - Database Test 005 - New same high or lower score in easy/medium/difficult difficulty for personal scores.

    Severity: 2

    Instructions: Completed the 10 levels on easy/medium/difficult difficulty with a same high score or lower for personal scores.

    Expected Result: The new score will not be added in the database and the main window will be up again, the game window will close.

  - Database Test 006 - New same high or lower score in easy/medium/difficult difficulty for overall scores.

    Severity: 2

    Instructions: Completed the 10 levels on easy/medium/difficult difficulty with a same high score or lower for personal scores.

    Expected Result: The new score will not be added in the database and the main window will be up again, the game window will close.

- Boundary Value Test Cases
  - Database Test 007 - New negative score in easy/medium/difficult difficulty

Severity: 2

Instructions: Completed the 10 levels on easy difficulty with a negative score when all slots for scores are filled up with previous scores.

Expected Result: The new score will not be added in the database and the main window will be up again, the game window will close.

Requirement: The response time should be quick. DELETE the score and other relevant information should display quickly, and the graphics on the screen should refresh instantly.

- Functionality Test Cases
  - Scores test 001 - A score increase should update quickly
    - **Severity:** 3
    - **Instructions:** Complete a level by advancing through the green arrows
    - **Expected Result:** the score at the top title bar should increment the correct number of points without any latency
  - Scores test 002 - A score decrease should update quickly
    - **Severity:** 3
    - **Instructions:** Run into an enemy and lose a battle.
    - **Expected result:** the loss of the fight should be reflected by an immediate negative score decrease in the title bar
  - User account storage 001 - Upon registering, a user’s account info should be processed and stored immediately
    - **Severity:** 3
    - **Instructions:** Fill out valid user account information as prescribed under the “Sign Up” tab and click the button below.
    - **Expected result:** the home screen should immediately appear with the username and options to play the game as well as a score view
  - User account storage 002 - User should be logged in immediately
    - **Severity:** 3
    - **Instructions:** Enter a valid, existing username/password pair under the “Sign In” tab and click the button below.
    - **Expected result:** the home screen should immediately appear with the username and options to play the game as well as a score view
  - Character requirement 001 - User’s character should move fluidly throughout the game
    - **Severity:** 3
    - **Instructions:** Launch a game level. Move the character around with the left, right, up, and down arrow keys.
    - **Expected result:** character should move fluidly and not “jump” i.e. it

should not disappear from one position and move to the next instantly; instead, it should move pixel-by-pixel (or close) to whatever direction is prompted by the keyboard

  - Character requirement 002 - Action prompt
    - **Severity:** 3
    - **Instructions:** Encounter an enemy on the playing screen
    - **Expected result:** Popup box should immediately display with any actions (fight, run away) that a user can take
- Equivalence Test Cases
  - *Not applicable for this requirement*
- Boundary Value Test Cases
  - Character requirement 003 - User's character should move fluidly throughout the game even when a user "button mashes"
    - **Severity:** 3
    - **Instructions:** Press the up, down, right, and left arrow keys on the playing screen in random order at high velocity.
    - **Expected result:** The character and the entire game should not freeze, instead it will continue to run in correspondence with the user's keystrokes

Requirement: To ensure security, nobody is allowed to see other users' information in the database. The user cannot modify his/her own score.

- Functionality Test Cases
  - User account 001 - Not gain access to other user's information
    - **Severity:** 1
    - **Instructions:** Ensure that multiple accounts are created on the system. Attempt to log in with one users' credentials multiple times.
    - **Expected result:** Only the user whose credentials have been supplied correctly should have their info displayed on the home screen. Not once should ANY of other users' data be displayed
  - User score 001 - User cannot manipulate score
    - **Severity:** 2
    - **Instructions:** Attempt to repeat certain actions (such as fights) or access max-health through title bar.
    - **Expected result:** Is not able to systematically manipulate score to cheat into attaining a high schools
- Equivalence Test Case
  - *Not applicable for this requirement*
- Boundary Value Test Cases
  - *Not applicable for this requirement*

Requirement: Any computer with Java 1.7 installed should run Dungeon.

- Functionality test cases
    - Program 001 - Running on any machine with Java
        - **Severity:** 1
        - **Instructions:** Ensure that the Java Runtime Environment is downloaded; save to ordinary location (e.g., the desktop, C:\ drive, home directory, etc.). Double-click the jar file
        - **Expected result:** the program should launch without any additional configurations
- Equivalence Test Cases
    - Program 002 - Running on any machine with Java v. < 1.7
        - **Severity:** 1
        - **Instructions:** Download and install prior stable releases of the Java JRE that are not the most recent (i.e. Java 1.7). Run Dungeon
        - **Expected result:** the program should display an error message informing the user of the need to update the out-of-date version of JRE
    - Program 003 - Running on any machine with Java 1.7
        - **Severity:** 1
        - **Instructions:** Download and install (if not already done) the latest version of Java JRE (1.7). Run Dungeon
        - **Expected result:** the program should run as described in the product backlog
- Boundary Value Test Cases
    - Program 004 - Placing executable in directory without sufficient permissions
        - **Severity:** 1
        - **Instructions:** Attempt to run Dungeon in a folder/directory without sufficient executable permissions
        - **Expected result:** no part of the program should run; either an error message from the program or the OS should display

Requirement: Dungeon should be as reliable as possible

- Functionality Test Cases
    - Program crash 001 - When a crash happens, user's score should be saved
        - **Severity:** 2
        - **Instructions:** Observe your score in a live game. Forcefully kill the Dungeon program through the terminal or task manager. Relaunch the program
        - **Expected result:** the user's score should be restored from the previous session.

    - Recover password 001 - A user’s password should be emailed to them if they request it as forgotten and then should be reset
        - **Severity:** 2
        - **Instructions:** Enter in an email address of an existing account under the “Forgot your password?” space
        - **Expected result:** A link should be emailed to the user with a link to reset their password
    - Recover password 002 - An email address for a non-existing account is entered
        - **Severity:** 2
        - **Instructions:** Enter in an email address of a non-existent account under the “Forgot your password?” space
        - **Expected result:** an error message should be displayed explaining that the account doesn’t exist
- Equivalence Class Cases
    - *Not applicable for this requirement*
- Boundary Value Test Cases
    - Recover password 003 - An invalid email is entered for Forgot Password
        - **Severity:** 2
        - **Instructions:** Enter in a string that does not resemble an email address
        - **Expected result:** an error message should be displayed explaining that the email address is invalid

Requirement: The game should be very usable

- Functionality Test Cases
    - Finish Game 001 - The game should be capable of being completed on easy mode.
        - **Severity:** 3
        - **Instructions:** Start Dungeon. Login with your credentials, create an account if not yet created. Select difficulty ‘easy’. Play the game to the end.
        - **Expected result:** You should finish the game without getting ‘stuck’ on a level (getting lost and being uncertain how to finish the level) for more than a couple of minutes.
    - Finish Game 002 - The game should be capable of being completed on medium mode.
        - **Severity:** 3
        - **Instructions:** Start Dungeon. Login with your credentials, create an account if not yet created. Select difficulty ‘medium’. Play the game to

        the end.
        - **Expected result:** You should finish the game without getting ‘stuck’ on a level (getting lost and being uncertain how to finish the level) for more than a couple of minutes.
    - Finish Game 003 - The game should be capable of being completed on hard mode.
        - **Severity:** 3
        - **Instructions:** Start Dungeon. Login with your credentials, create an account if not yet created. Select difficulty ‘hard’. Play the game to the end.
        - **Expected result:** You should finish the game without getting ‘stuck’ on a level (getting lost and being uncertain how to finish the level) for more than a couple of minutes.
    - Modify Personal Information 001 - A user should be capable of easily modifying their information.
        - **Severity:** 2
        - **Instructions:** Start Dungeon. Login with your credentials, create an account if not yet created. Look for a button or menu item that implies you can modify your information. Activate that element.
        - **Expected result:** A form shows up allowing the user to modify their information.
    - Equivalence Class Cases
        - *Not applicable for this requirement*
    - Boundary Value Test Cases
        - *Not applicable for this requirement*

Requirement: When the user is playing the game, the game display will record input and transfer it to the game engine.

- Functionality Test Cases
    - Movement Test 001 - Right Movement
        - **Severity:** 2
        - **Instructions:** After the game is started, the user presses the right arrow key.
        - **Expected result:** The character moves to the right one pixel
    - Movement Test 002 - Left Movement
        - **Severity:** 2
        - **Instructions:** After the game is started, the user presses the left arrow key.
        - **Expected result:** The character moves to the left one pixel

- Movement Test 003 - Upward Movement
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the up arrow key.
    - **Expected result:** The character moves up one pixel.
- Movement Test 004 - Downward Movement
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the down arrow key.
    - **Expected result:** The character moves down one pixel.
- Movement Test 005 - Left Movement Sustained
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the left arrow key and holds it.
    - **Expected result:** The character moves left one pixel, then continually moves to the left.
- Movement Test 006 - Right Movement Sustained
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the right arrow key and holds it.
    - **Expected result:** The character moves right one pixel, then continually moves to the right.
- Movement Test 007 - Upward Movement Sustained
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the up arrow key and holds it.
    - **Expected result:** The character moves up one pixel, then continually moves upward.
- Movement Test 008 - Downward Movement Sustained
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the down arrow key and holds it.
    - **Expected result:** The character moves down one pixel, then continually moves downward.
- Movement Test 005 - Left Movement Sustained
    - **Severity:** 2
    - **Instructions:** After the game is started, the user presses the left arrow key and holds it.
    - **Expected result:** The character moves left one pixel, then continually

moves to the left.

    - Fight Test 001 - Fight Dialogue
        - **Severity:** 2
        - **instructions:** When the player runs into an enemy, the fight dialogue will pop up and the player chooses any random option.
        - **Expected result:** The option should be clickable and

- Equivalence Test Case
    - *Not applicable for this requirement*
- Boundary Value Test Cases
    - *Not applicable for this requirement*