System and Unit Test Report

Heading:

System and Unit Test Report
D&D DM Toolset
The Fighting Mongooses
5/28/2021

Testing Approach:

We focused primarily on manual testing for this project for two main reasons. First, Unity is a platform that lends itself to rapid live testing of a project as changes are made. The second, and perhaps more pertinent reason was that few members of our team were familiar with Unity as a development environment, and so we were not familiar with any potential automated testing frameworks that are available for Unity.

As the project progressed and we became more familiar with Unity, we learned of, and began to introduce, automated testing. Unfortunately this occurred late in the overall development cycle, so it was only used to test a few components.

System Test Scenarios:

Sprint 1:

User story 1: "As a DM, I want to not have to manage large initiative queues myself so I can focus on other things"

User story 2: "As a DM, I want to have pertinent player info available as a ready reference, because large groups of players I don't know well can be hard to keep track of when running a session"

Scenario:

1. Start app and begin new session.
    a. Should see empty player list and empty initiative tracker as well as dice roller
2. Add players to player list
3. Add players and monsters to initiative queue
4. Update initiatives
    a. App will auto sort all combatants
5. Press “Complete Turn” to end each combatant’s turn
6. Press “New Round” to repopulate initiative queue and begin a new round of combat.
7. All pertinent information on monsters and players is available on screen at any point throughout the game.

Sprint 2:

User story 1: “As a DM I would like to be able to easily access monster stat blocks, because searching through a book for it is tedious and disruptive to the flow of the session.”

User story 2: “As a DM, I want to see all player status effects on the initiative queue, because tracking them manually can get out of hand”

User story 3: “As a DM, I want the players I create to automatically populate the initiative queue, so that they can be used during encounters.”

Scenario:

1. Start the app and begin a new session.
2. Click the monsters button to open the monster profiles window.
3. Click the search bar and type in the name of a specific monster.
4. Click the monster profile in the results.
5. Click “add selected monsters” button
6. Click the monsters button again to close the monster profiles window.
7. When the monster profile appears in the initiative queue, hover over the profile and observe the stat block visual pop-up which appears at the mouse position.
8. On the monster profile, click the status effect button and observe the status effects panel which appears.
9. Click the “X” button at the top right of the status effects panel to close it.
10. Click the “add player” and observe the form which is opened.
11. Fill out the appropriate fields in the form and click the save button.
12. Click the player name which will appear in the players panel.
13. The player’s character is then added to the initiative queue.

Sprint 3:

User story 1: “As a DM, I would like to save all my information very quickly so that I don’t have to redo this every time the app starts.”

User story 2: “As a DM, I would like to have a cleaner, less overwhelming means of recording info about players and encounters so that my mind can be freed up to improve the narration and atmosphere for the players.”

Scenario:

1. start the DM Toolset; select “New Game”; type
    a. Game Session Name = <my D&D session>
2. click “Add Player”; type
    a. Player Name = <Gary Gygax>
    b. Character Name = <Greyhawk>
    c. Character Class = <Ranger>
    d. Armor Class = <20>
    e. Max HP = <60>
3. click on “Gary Gygax” under the “Players:” text; move the cursor over the newly added character in the queue on the left side of the screen
4. click on “Status: NONE” at the bottom of the character in the queue; click “Add a Status”; select “Petrify”
5. User is now familiar with setting status conditions for a character
6. click “Save Game” at the bottom of the application; then choose Quit
7. select “Load Game” and choose the correct session by clicking the text “my D&D session” as entered before; click “Load Game”
8. User should see the character/initiative queue created prior to pressing the “Save Game” button

Sprint 4:

User story 1: “As a player I want to be able to make dice rolls that are visible to my fellow players and DM, because I don’t own any D&D dice.”

User story 2: “As a DM, I would like to see the documentation for this program so that I know how to use it.”

User story 3: “As a DM, I would like to have a cleaner, less overwhelming means of visualizing info about players and encounters so that my mind can be freed up to improve the narration and atmosphere for the players.”

Scenario:

1. start the DM Toolset; select “New Game”; type
    a. Game Session Name = <my D&D session>
2. click “Dice Roller”; click one of the “Roll” buttons to roll the die; click “X” to close the page
3. click “Add Player”; type
    a. Player Name = <Gary Gygax>
    b. Character Name = <Greyhawk>
    c. Character Class = <Ranger>
    d. Armor Class = <20>
    e. Max HP = <60>
4. click on “Gary Gygax” under the “Players:” text; see the left side of the screen for all relevant player information, and move the cursor over the character to see the stat block on that character
5. open a preferred web browser; navigate to “dnd-dmts-toolset.readthedocs.io”; click any of the initial links to see helpful information about the program

Modules:

- Combatant Initiative Panel
    - Class: Monsters
        - Test Case:
            - If the Combatant is a monster, then it should load in and display values for that monster, including the associated image.
    - Class: Players
        - Test Case:
            - If the Combatant is a player, then it should load in and display values for that player, including an image of the player's class if available
    - Class: Change of Combatant Information
        - Test Case:
            - If the Combatant's initiative or status values change, then the Combatant Initiative Panel should update to display the new values.
- Initiative Queue
    - Class: Empty
        - Test Case:
            - If the Initiative Queue is empty, then end turn, start new round, and delete mode should not change anything in regards to the queue.
    - Class: Populated
        - Test Case:
            - If there are some Combatants in the Initiative Queue, then end turn should remove the top combatant from the list, start new round should return all combatants removed using the end turn button. Clicking delete mode allows the permanent removal of combatants from the Initiative Queue, these removed combatants should not be restored on start new round.
    - Class: Change of Combatant Initiative
        - Test Case:
            - If a Combatant's initiative is changed then the Combatant's position in the Initiative Queue should be changed to reflect that change. Greater initiatives are higher in the queue, if there is a tie, then no preference is given.

- Global Player List
    - Class: Empty List
        - Test Case:
            - If no player has been added with player creation, then none should be listed in menu, and regardless of clicks within the player list section none should be added to Initiative Queue
    - Class: Populated List
        - Test Case:
            - If there are some players that have been created with the player creation module, then the players should be listed visually. If a player name is clicked, then it should be added to the Initiative Queue.
    - Class: Duplicate Entries
        - Test Case:
            - If there are some players that have been created with the player creation module, then the players should be listed visually. If a player name that has already been added to the Initiative Queue is clicked, then it should not be added to the Initiative Queue.
- Player Creation
    - Test Case:
        - When any fields are left empty, the player is not added. The program remembers the filled in fields when the add player button is pressed again.
    - Test Case:
        - When any field has an invalid input (integer fields contain non-integer input) the player is not saved. The program remembers the filled in fields when the add player button is pressed again.
- Dice Roller
    - Class: Pressing any of the dice buttons
        - Test cases:
            - Pressing any of the dice buttons should result in a randomly generated number ranging from [1, X] where X = the maximum roll number (e.g., a D6 die will have a roll result range of [1, 6]).
- Searchable Monster List
    - Class: No search query provided
        - Test case:
            - When no search query is provided, it should list all of the monsters
    - Class: Search query provided which has results
        - Test case:
            - When a search query is provided, it should list the monsters that have its name match the query

        - Class: Search query provided which has no results
            - Test case:
                - When a search query is provided, it should list the monsters that have its name match the query, or no monsters at all.
- <u>Selected Monsters List</u>
    - Class: No selected monsters
        - Test Case:
            - No selected monsters should result in the selected monsters list being hidden
    - Class: One selected monster (1 copy)
        - Test cases:
            - When any number of monsters are selected, it should show up in the list.
            - Pressing the “Add selected monsters” list should remove all selected monsters from the list.
    - Class: Two selected monsters (1 copy each)
        - Test cases:
            - When any number of monsters are selected, it should show up in the list.
            - Pressing the “Add selected monsters” list should remove all selected monsters from the list.
    - Class: One selected monster (2 copies)
        - Test cases:
            - When any number of monsters are selected, it should show up in the list.
            - Pressing the “Add selected monsters” list should remove all selected monsters from the list.