

Sprint 2: Design Inspection, Code Inspection, Unit Testing

Jacob Riggs, Julian Ng, Christian McKain, Cole Compton, Adrian Taubner

Design Inspection

Product: The Endless Abyss

Date: 10/11/2019

Authors: Development Team - Team 21

Defect #	Description	Severity	How Corrected
1	Database needs to account for arrays within objects (Ex. Scenarios, inventories, player, recent rooms, user trophies)	Critical	We created multiple tables that are queried and added to during games. Once a game has ended, the table entries will be cleared
2	The vuex store which holds global data for the webpage is currently struggling to handle time sensitive server request	Important	We currently worked around this issue by moving request to related page for no reactive requests like logging in.
3	Some frontend page method calls are rather large (bloated)	workaround	Currently we are keeping it this way to keep working on new features, time will be taken later to make the method calls smaller and more module

4	When passing the next room from the server, the server would respond with 3 objects	workaround	We edited the existing room object and added the description values to room as well, this lead to cleaner and easier code to work with
5	When passing information between the Vuex store and other vue pages we found data was being lost	workaround	We removed our dependence on Vuex an added the ability to pass props using vue-router
6	More database tables necessary to account for loading game states.	Critical	Added 3 different tables in order to store crucial information for continuing games.
7	If a room is entered more than once, it is possible for a player to get an item that already exists in their inventory.	Important	We decided to check for the item already existing, and if so, give gold to the user instead of the duplicate item.

Code Inspection

Product: The Endless Abyss

Date: 10/11/2019

Authors: Development Team - Team 21

Defect #	Description	Severity	How Corrected
1	Backend organization of user objects not storing or sending enough information to the frontend	Important	Instead of sending a Room ID, we created a Room object in the backend that stores the Room ID and Room Description
2	Login requests were completing the redirect before server response	Critical	Moved Login request from the vuex store to the vue page then check for server responses.
3	Similar to Login Create user and recover password failed to prevent redirection when failed create	Critical	Moved the request code outside of the vuex store to prevent these issues.
4	Refactoring of objects to keep overall project consistent and readable.	Workaround	Renamed the user object to the game object so as to not confuse usernames with games.

Unit Testing

Module: Leaderboard

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
N/A	-	-	-	-	-

Module: Game

Date: 10/11/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
1	Inventory box appears as a line due to empty inventory	User starts a new game or loses their last item	Inventory box appears incorrectly	Workaround	Added v-if to display inventory section only when user has at least one item
2	Game screen would not load correctly	User starts or continues game	Game screen displays as blank	Critical	Added methods within MainMenu to modify game object before opening game page

Module: Server

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
1	Incorrect formatting of a JSON object.	A user object that was contained within another dictionary.	Access to the value on the frontend was not as intended.	Workaround	Combined two dictionaries into one in order to avoid access abnormalities.
2	Access to the database was done incorrectly as duplicates could occur when they were not supposed to.	Rooms that we were keeping track of for deciding the next room.	Getting multiple of the same room in the database.	Important	Made sure to delete and add rooms as necessary as to prevent duplicates.
3	Inventory is able to have duplicate items which is not allowed.	When a player hits the same room twice they can choose an item that is already in their inventory.	Item appears twice in inventory.	Important	Make sure to check inventory for the item that already exists and respond accordingly.
4	No description of the room nor a name for the room existed.	When a room was entered there was no information initially for the player.	No output would occur due to lack of a description.	Critical	Added another table in the database that describes each room upon entering.