

|  |
| --- |
| DAT602 Assessment 2 |
| RE:Battlespire |
| Name: Rhylei Tremlett  ID: 13513865  Date: 25/06/2023 |

Table of Contents

[Known Issues 2](#_Toc137542074)

[Player Login 3](#_Toc137542075)

[Player Registration 3](#_Toc137542076)

[Creating a Gameboard 3](#_Toc137542077)

[Spawning Entities 3](#_Toc137542078)

[Player Movement 3](#_Toc137542079)

[Player Scoring 3](#_Toc137542080)

[Player Acquiring Inventory 3](#_Toc137542081)

[Monster NPC Movement 3](#_Toc137542082)

[Reset Game 4](#_Toc137542083)

[Re-Generate Gameboard 4](#_Toc137542084)

[Update Player 4](#_Toc137542085)

[Delete Player 4](#_Toc137542086)

[References 1](#_Toc137542087)

[Appendix 1](#_Toc137542088)

# Known Issues

1. If a player attacks and kills a player, the player who died does not know they died and the game breaks but does not crash or show an error.
2. The game crashes if the inventory fills up and the player takes another item.
3. Monsters can move onto players, chests, exits and other monsters.
4. Players can attack all players on the home tile at once.
5. Monsters don’t attack players unless they are attacked.
6. Tiles aren’t generated when a player moves into an area with missing tiles, instead existing tiles are ‘smeared’ across. There is effectivley a wall that you cant go past, except its hard to tell where it is.
7. Items can’t be moved inside a chest’s inventory, and the game crashes.
8. Messages can’t be sent in the chat system.
9. The game does not have a logout feature, so the game needs to be restarted.
10. Players do not acquire kill score points when killing an entity.
11. Account creation doesn’t properly work; the account entry is missing data.
12. Deleting a player account doesn’t close the GUI.
13. Admins can’t delete their own accounts.
14. Monsters move at N speed, where N is the number of players logged in.
15. Users can log in multiple times.

# Implementing Exception Handlers and Parameter Validation

For each of my procedures I need to implement the following:

1. A standard exception handler.
2. Validation for each parameter.
3. Any specific error handlers the procedure needs.
4. Foreign key validation for any parameters that are foreign keys.

NOTE: All following code snippets are Embedded word documents, double click them and you can copy the contents.

NOTE2: The snippets don’t fit well in word due to its width, I suggest pasting them into an editor if you have trouble reading them.

Below is the standard handler that is used in every single procedure. Using this allows every error produced by the procedure to be formatted in the same way.

It catches the custom errors that my procedure throws, and re throws them upto the application. Any other exceptions it reformats and throws them up as well.



The above code is originally based on the code found [here](https://medium.com/startupward/how-to-try-and-catch-errors-in-mysql-stored-procedures-56604ffa17df).

Next is my validation, for each procedure the conditions and messages will be different, but the standard structure looks like this. For most of the variables I just need to check if they are missing, some of them are numeric and need to be positive so I can check for that as well.



Inside the if block is the standard implementation for my user-defined exceptions. When ever I need to throw an error based on a condition the above is also what I use.

# Player Login

Providing a non-existent login or an incorrect password results in the following:

A screenshot of a computer error message

Description automatically generated with medium confidence

Not providing one of the fields results in the following:

A screenshot of a computer

Description automatically generated with medium confidence

Failing the password 5 times results in the following:

A screenshot of a computer error message

Description automatically generated with low confidence

Once this is received even the correct password will not let the user login.

Any other errors are displayed as such:

A screenshot of a computer error

Description automatically generated with medium confidence

To avoid repetition I wont show the above error message for every requirement.

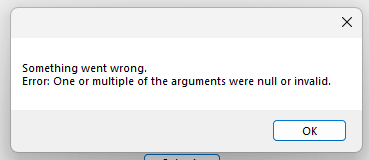
# Player Registration

Attempting to use a username that already exists reults in:

A screenshot of a computer error message

Description automatically generated with medium confidence

Not providing one of the fields results in the following:



Attempting to use an email that already exists reults in:

A screenshot of a computer error message

Description automatically generated with medium confidence

# Other Procedures

For the other procedures, none of them have any specific error messages since they are typically unfailable. These procedures don’t cause any issues unless the paramters given to them are incorrect, therfore I found no reason to add in specific exception handling. This also makes it extremley difficult to provide example testing as they are background procedures that don’t have any visible results. The move player function updates the tile the players is standing on but doesn’t return anything to the GUI, the same goes for most of the procedures. Instead the GUI refreshes itself regularly by manually getting the information it needs, these are mere select statements that don’t throw errors as its just “get everything”.

I apologize for not being able to provide proper testing documentation. I have learned that going beyond the scope was not a good idea, and in fact reduced the overall quality of my code. Due to the excessive amount of code I have it was difficulty to implement proper OO design that I had learned from DAT602 and SDV601 into this project.