

SWEN1: Monster Trading Cards Game-Protocol

About:

The purpose of the recently completed game called “Monster Trading Cards Game” is to create a simple multiplayer card game that can be played by many users. The project/game allows users to register and log in to their accounts. Once logged in, they can collect cards with coins and trade them with other players online. They also have the option to create their own decks, which can be used in battle mode. In battle mode, players with more versatile decks have an advantage, but luck also plays a role, since cards are randomly drawn from the deck. In addition, there is also a scoreboard that shows the best players, and users can strive to move up in the rankings by being victorious in battles.

Lessons learned:

Developing a card game in C# showed me that developing a game requires a lot of precision and coding skills. In game development, overlooking even a single case can have significant consequences and potentially ruin the entire game. Therefore, it is crucial to consider all possible scenarios and test the game to ensure that the functionality and user experience are error free, because even a small bug can quickly turn into a bigger problem.

To manage the project effectively, I learned that it was important to break it down into smaller tasks and approach each one systematically. Although I had divided the project into smaller tasks, I had a major setback when I realized that I had put most of my code in the main/Program.cs file in C#. This was because I was too lazy at the time and had not divided the code into individual classes and methods. Unfortunately, this decision caused the code to look messy and confusing and although the code worked, it wasn't structured properly. That is why it took a lot of time to revise it and organize everything properly. This experience taught me the importance of writing well-structured code from the beginning to avoid problems in the future.

And it still took me about 80 hours to finish the card-based game even though it is still not 100% complete, but overall, developing the card game in C# taught me valuable lessons about project management, user experience, and software testing that I can apply to future projects.

Unit test design:

There are four main functions that I am using for my unit tests. The first two are the methods of the MySQL Database, which confirm the connection status. The third method does not perform a full calculation of damage and health points, but instead uses predetermined rules to determine the outcome of the battle so called instant wins (Dragon > Goblin, Kraken > Spell). The fourth method I use for unit test is the actual damage calculation method. If there is no instant win, the cards are sent to this method to determine the damage and the winner. This also takes into account the special elements of certain monsters and spells that can double or halve the damage when facing each other. There are

26 unit test in total and many of them focus on testing the accuracy of the damage calculation and the instant-win method, as these are critical aspects of the combat logic.