MCTG - SWEN1 - SEM3

How to use the server

- git clone https://github.com/SchaqO/MCTG.git
- Start the programe
  - Run Main.java
  - Sever accepts up to 5 clients at the same time
- Database required (postgres only since only those drivers are bundled)

Additional resources

- CURL file
- DB ERM Diagrame
- JUnit library
- GSON library
- Postgre JDBC library
- Lombok library
- Mockito library
- Opentest4j library
- Apigarudian library

Functions

```
- User:
    - a user is a registered player with credentials (unique username, password).
    - a user can manage his cards.
    - a card consists of: a name and multiple attributes (damage, element type).
    - a card is either a spell card or a monster card.
    - the damage of a card is constant and does not change
    - a user has multiple cards in his stack.
    - a stack is the collection of all his current cards (hint: cards can be removed by trading).
    - a user can buy cards by acquiring packages.
    - a package consists of 5 cards and can be acquired from the server by paying 5 virtual coins
    - every user has 20 coins to buy (4) packages.
    - the best 4 cards are selected by the user to be used in the deck
    - the deck is used in the battles against other players.
```

Game logic

```
 Your cards are split into 2 categories:
• monster cards
    cards with active attacks and damage based on an element type (fire, water, normal).
    The element type does not effect pure monster fights.

• spell cards
    a spell card can attack with an element based spell (again fire, water, normal) which is:
      – effective (eg: water is effective against fire, so damage is doubled)
      – not effective (eg: fire is not effective against water, so damage is halved)
      – no effect (eg: normal monster vs normal spell, no change of damage, direct
comparison between damages) Effectiveness:
• water -> fire
• fire -> normal
• normal -> water

  Cards are chosen randomly each round from the deck to compete (this means 1 round is a
  battle of 2 cards = 1 of each player). There is no attacker or defender. All parties are equal in
  each round. Pure monster fights are not affected by the element type. As soon as 1 spell
  cards is played the element type has an effect on the damage calculation of this single
  round. Each round the card with higher calculated damage wins. Defeated monsters/spells
  of the competitor are removed from the competitor’s deck and are taken over in the deck of
  the current player (vice versa). In case of a draw of a round no action takes place (no cards
  are moved). Because endless loops are possible we limit the count of rounds to 100 (ELO
  stays unchanged in case of a draw of the full game).
```

Unit tests

```
 The heaviest testing was done regarding Users and Packages, I tested the initialization
of the Objects and tested if they return the correct values.
For the unit tests I kept it simple and mostly just tested the objects and the parameters they were constructed with.
Meaning I checked the a class object after construction if the correct paramters were passed during creation.
Aswell as some tests to check the correct CardType and Elements
That includes classes such as the User, Profile, CardElement, CardType and Holder.
Having several tests within each testing class.
```

Lesson learned

```
 It was definitely a massive challenge when creating the REST API without any framework,
especially since learning material provided from the FH was very short and dry and by far not enough.
Without any help it was a huge challenge, taking me several days and hours to grasp some conecpts.
However from this challenge I believe my coding skills have improved significantly.
Especially in the sense of object-oriented programming. With this big project, understanding the object oriented approach
while still keeping an overview was for me personally quite difficult, the object relation and nesting was very hard to grasp someti
Trying to understand the concepts of HTTP requests, methods and responses has been very crucial
and informative to general knowledge in the field. Continuing with DB management, the project was
almost a whole package that included many aspects of programming and would require
a lot of time invested to complete properly. I massively underestimated the project,
this ended up being a massive mistake, I kept pushing the deadlines until it was too late.
This has resulted in me being more careful with deadlines and trying to stay on track
properly from now, to not fall behind and get into the same tedious situation as this project.
```

Time tracked

```
 It was hard to track time as I had many breaks inbetween, however I can say that since I started
this project late into the semester, i got too close to the deadline, as close as a few days.
Therefore this has led to me working most of the day across several days on this project to get it done.
I would probably say close to 4-8 hours a day for several days if not weeks. And I am yet to fully understand
every single concept used in this projet.
```

GitHub Link

https://github.com/SchaqO/MCTG

Git Log

```
 $ git log
```

commit 15d7cfe7d4dbbd39779b37f8583d0e322c82f3a4 (HEAD -> master) Author: SchaqO shako-1998@hotmail.com Date: Thu Apr 7 05:40:52 2022 +0200

```
 Project finito :)
```

commit 2b4742cda14e9c3c2a479dba375f769dc2df41c8 (origin/master) Author: SchaqO shako-1998@hotmail.com Date: Wed Apr 6 23:48:23 2022 +0200

```
 battle done. Project done (maybe some refining later).Hallelujah
```

commit 6b4e74738a770af537fd89f7807614d2447ab2cd Author: SchaqO shako-1998@hotmail.com Date: Wed Apr 6 18:50:00 2022 +0200

```
 edit profile done & started working on battle
```

commit eb137250ccdc573c31247b79019fc34046d7b757 Author: SchaqO shako-1998@hotmail.com Date: Tue Apr 5 19:13:22 2022 +0200

```
 started coding on the battle
```

commit 9208f383989e83ef9a9cd2c184817a2c38a110e2 Author: SchaqO shako-1998@hotmail.com Date: Tue Apr 5 18:49:11 2022 +0200

```
Deck done
```

commit 83e0cb75a5f55d610378187489bc7687bbc16d12 Author: SchaqO shako-1998@hotmail.com Date: Tue Apr 5 16:09:47 2022 +0200

```
package sell , stack, test
```

commit ddb6437de52ab2a141d33bdd01e116797140e13d Author: SchaqO shako-1998@hotmail.com Date: Mon Apr 4 03:21:06 2022 +0200

```
hashID implemented into DB
```

commit 7cc0869e26a98eb4fdbc7a7ccc6c07443cee2ccc Author: SchaqO shako-1998@hotmail.com Date: Mon Apr 4 02:47:33 2022 +0200

```
Simple hash method for generating IDs done
```

commit ca6ecaf15d4fbe936c1da0dfaca0e0a89c450e43 Author: SchaqO shako-1998@hotmail.com Date: Mon Apr 4 00:28:22 2022 +0200

```
m
```

commit 22b8453608c634552f8deadfa3a04d5ea3cdb322 Author: SchaqO shako-1998@hotmail.com Date: Mon Apr 4 00:06:53 2022 +0200

```
Package done
```

commit 630be21eb5c3058f28e95257e09ed7a63ed8889f Author: SchaqO shako-1998@hotmail.com Date: Wed Mar 30 20:14:15 2022 +0200

```
Further work on the controller
```

commit 757f231dca0c1f048edc30023bcd915851f0a015 Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 14:40:55 2022 +0100

```
fixed a mistake in the README
```

commit 1db987748485cea1a0b4fd60d0bb489848f1ebc3 Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 14:39:56 2022 +0100

```
final README
```

commit 42996331a98ebe3a1521dc5d72dcabcad0f381cf Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 14:34:44 2022 +0100

```
Worked on controls for CURL
```

commit a650363e055f575667b84381c265c923ef8d8765 Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 06:11:52 2022 +0100

```
added some files
```

commit 06c2a1d2b0c21ccb8857a50e3fe6a52714836589 Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 06:09:22 2022 +0100

```
created a readme
```

commit 6ff1764d0a1f43ac96e1ced4130bd83ad93da14f Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 05:21:30 2022 +0100

```
Unit Tests added
```

commit bbd6f9e9f5b6a51ab560858da17a139706b5773a Author: SchaqO shako-1998@hotmail.com Date: Fri Feb 25 04:42:30 2022 +0100

```
Some more progress
```

commit 590a7527856edf8b10d5032079463e9102ad6364 (db) Author: SchaqO shako-1998@hotmail.com Date: Thu Feb 24 22:00:17 2022 +0100

```
V1.1
```

commit 9bb54cd811784281d5e208fb58efa7c16492a351 (model) Author: SchaqO shako-1998@hotmail.com Date: Thu Feb 24 18:42:25 2022 +0100

```
DB progress
```

commit 2eb7bab78fb9a2fcd9eb633251d266a1177b77f8 Author: SchaqO shako-1998@hotmail.com Date: Thu Feb 24 17:35:55 2022 +0100

```
User model completed
```

commit 3bb7af416d1a751c2758f6ec0af83a1dca9e44b5 Author: SchaqO shako-1998@hotmail.com Date: Thu Feb 24 17:11:45 2022 +0100

```
V1.0
```

commit 43e475b5832d33f4b2eac6cd79a4171efb8e969d Author: SchaqO shako-1998@hotmail.com Date: Wed Feb 23 23:12:39 2022 +0100

```
server finally done
```

commit cb0e1a2f6450a7d0fe27f7d3381c0108f0dc9cb5 Author: SchaqO shako-1998@hotmail.com Date: Wed Feb 23 14:57:27 2022 +0100

```
some rest api implementation progress
```

commit 04549b30630aebb08fda3145e4c0cb25209e63d9 Author: SchaqO shako-1998@hotmail.com Date: Tue Feb 22 20:17:30 2022 +0100

```
trying to build the project structure and test git
```

commit 5df5e4fbd4f59309fa18a60e1ad44dff14f4784e Author: SchaqO shako-1998@hotmail.com Date: Mon Feb 21 21:02:22 2022 +0100

```
Test to add to Github
```

commit 7bce3ebd5a03d6267e224010d82e4d7f32bf18ed Author: SchaqO shako-1998@hotmail.com Date: Sat Jan 22 03:43:51 2022 +0100

```
Another test commit
```

commit cc156e1f927ec0a399cbddcfcd1b36127e01f575 Author: SchaqO shako-1998@hotmail.com Date: Sat Jan 22 03:36:55 2022 +0100

```
test commit
```