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

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

```
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.
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. 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 42996331a98ebe3a1521dc5d72dcabcad0f381cf (HEAD -> master, origin/master)
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
:...skipping...
commit 42996331a98ebe3a1521dc5d72dcabcad0f381cf (HEAD -> master, origin/master)
```

```
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

:...skipping...
commit 42996331a98ebe3a1521dc5d72dcabcad0f381cf (HEAD -> master, origin/master)
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

:
commit 42996331a98ebe3a1521dc5d72dcabcad0f381cf (HEAD -> master, origin/master)
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>
```

```
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
```