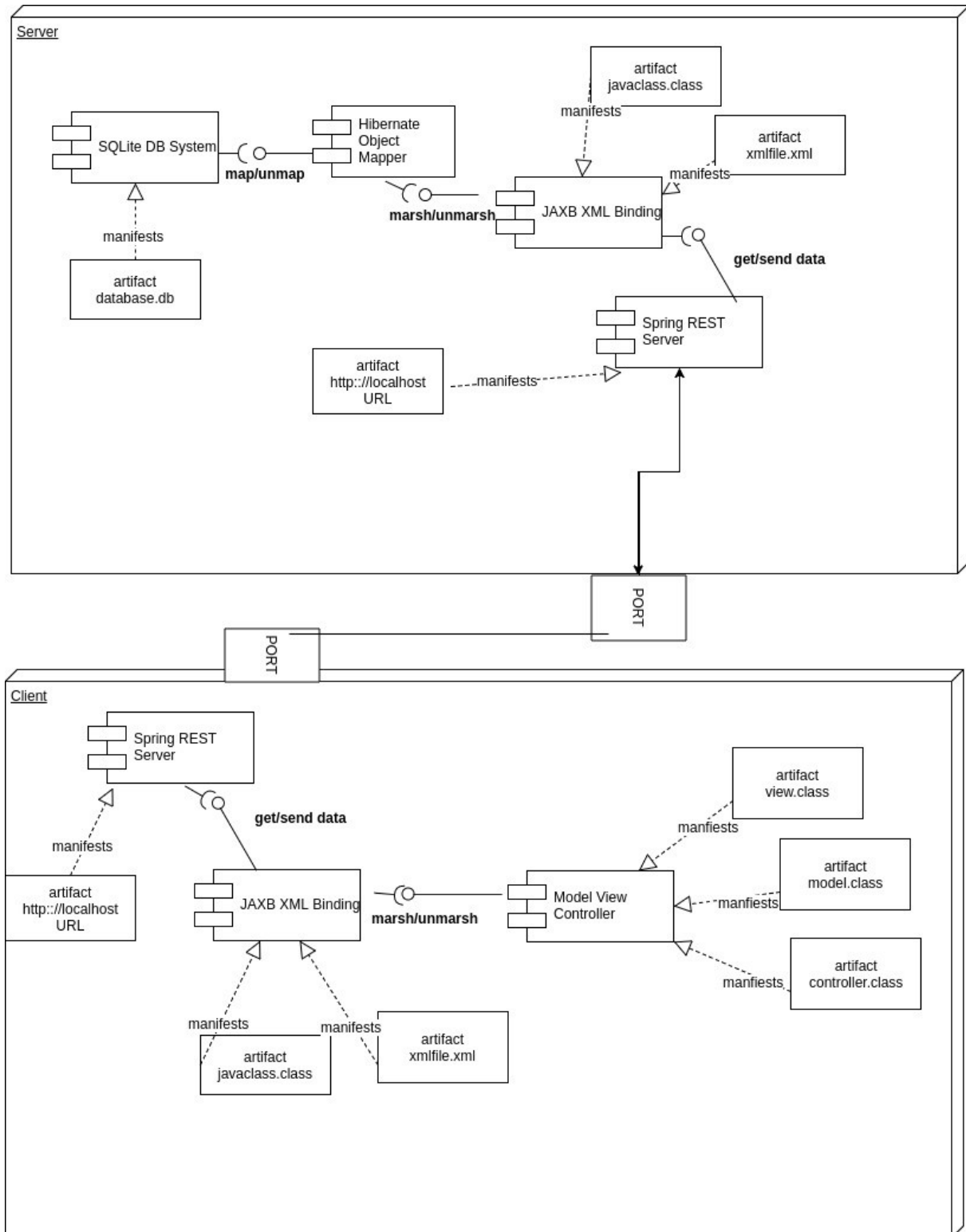


VU Software Engineering 1**Abgabedokument****Teilaufgabe 1 +2****(Anforderungsanalyse und Planungsphase +
Serverimplementierung)**

Persönliche Daten, vollständig ausfüllen:

Nachname, Vorname:	Hartmann Benjamin
Matrikelnummer:	0915812
E-Mail Adresse:	a0915812@univie.ac.at
Datum:	06.12.2017

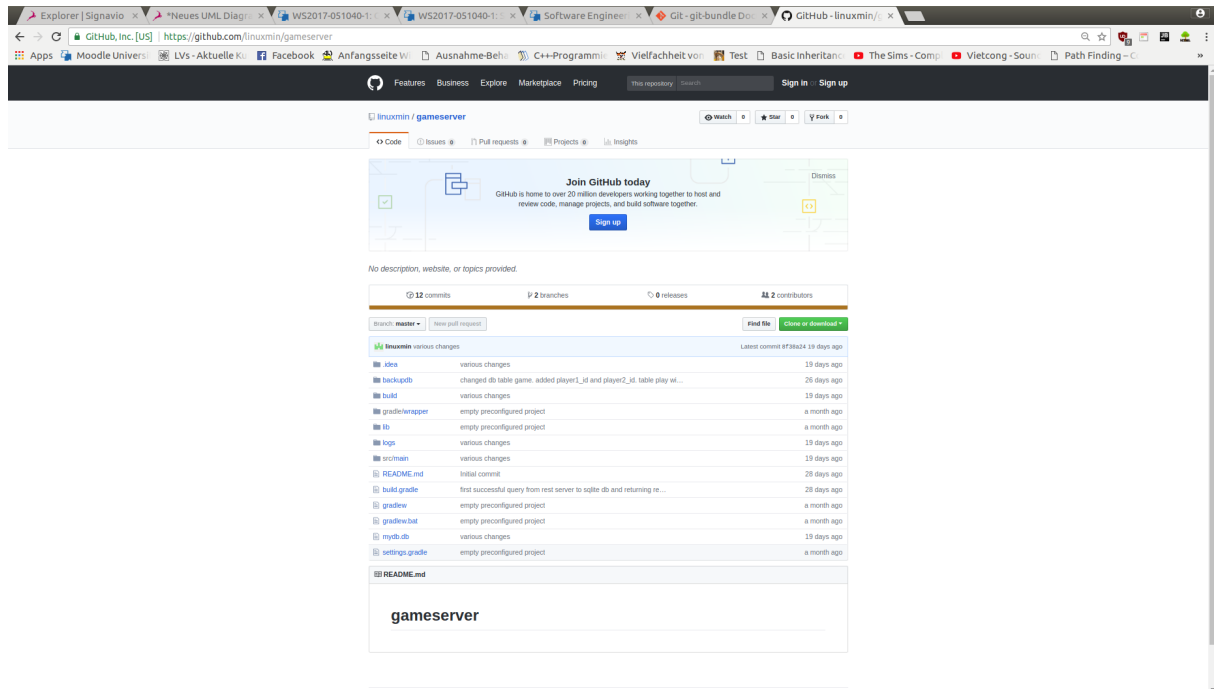
Grobarchitektur



Konfigurationen 2:

Git Bundle:

refs/heads/master oder refs/remotes/origin/dao (commits wurden alle auf dem remot rep gemacht, da ich auf 2 Rechnern (Arbeit und Zuhause) arbeite.



Persistenzkonfigurationsdatei für JPA:

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd" version="2.1">
  <persistence-unit name="gameserver" transaction-
type="RESOURCE_LOCAL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</
provider>
    <class>server.Player</class>
    <class>server.Game</class>
    <class>server.Map</class>
    <class>server.Tile</class>
    <class>server.ActionMove</class>
    <exclude-unlisted-classes>true</exclude-unlisted-classes>
    <properties>
      <property name="javax.persistence.jdbc.driver"
value="org.sqlite.JDBC"></property>
      <property name="javax.persistence.jdbc.user"
value=""></property>
      <property name="javax.persistence.jdbc.password"
value=""></property>
      <property name="javax.persistence.jdbc.url"
value="jdbc:sqlite:mydb1.db"></property>
      <property name="hibernate.dialect"
value="org.hibernate.dialect.SQLiteDialect"></property>
```

```
        <property    name="hibernate.show_sql"
value="true"></property>

        <property    name="hibernate.format_sql"
value="true"></property>

        <property    name="hibernate.hbm2ddl.auto"
value="create"></property>

    </properties>
</persistence-unit>
</persistence>
```

Relativer Pfad zum Programm und zur mydb1.db

/gameserver/Executables/

Ausführung

java -jar Server_Teilaufgabe_2_Benjamin_Hartmann_0915812.jar

In der Entwicklungsumgebung:

Klassen Pfad /gameserver/src/main/java/server/

MainKlasse: StandAloneRESTServer

Persistence.xml von JPA und log4j2.properties Logger:

/gameserver/src/main/java/resources/META-INF

User-Stories

1. Als Spieler möchte ich die Möglichkeit haben ortsungebunden gegen einen anderen Spieler antreten zu können.
2. Als Spieler möchte ich die Möglichkeit haben, meine Spielzüge in Echtzeit zu verfolgen.
3. Als Spieler möchte ich die Möglichkeit haben, die Spielzüge meines Gegners in Echtzeit zu verfolgen.
4. Als Spieler möchte ich die Möglichkeit haben, den Spielverlauf im Nachhinein nachvollziehen zu können.
5. Als Spieler möchte ich die Möglichkeit haben, meiner Spielfigur einen Namen zu geben.

Business Rules

1. Ein Spiel darf nicht länger als 200 Spielaktionen dauern, ansonsten verliert die KI die gerade an der Reihe ist.
2. Eine Spielaktion darf max. 3 Sekunden lang dauern, sonst verliert die KI die gerade an der Reihe ist.
3. Sollte keine KI nach 200 Aktionen auf normalem Wege gewinnen, verlieren beide KI's.
4. Jede KI generiert ihre eigene Kartenhälfte mit Ihrer Burg. Die Kartenhälften sind für beide KI's sichtbar (außer der Burg und den Schätzen).
5. Ein Schatz kann nur von der zugeordneten KI aufgenommen und verwendet werden um zu siegen.
6. Eine KI muss in jeder Runde eine Aktion setzen.
7. Eine KI darf maximal eine Aktion pro Runde setzen.
8. Die den KI's zugeordneten Schätzen und Burgen werden vom Server auf den von Ihnen generierten Kartenhälften versteckt und sind nicht sichtbar.

Netzwerkcommunication

Allgemeine Beschreibung

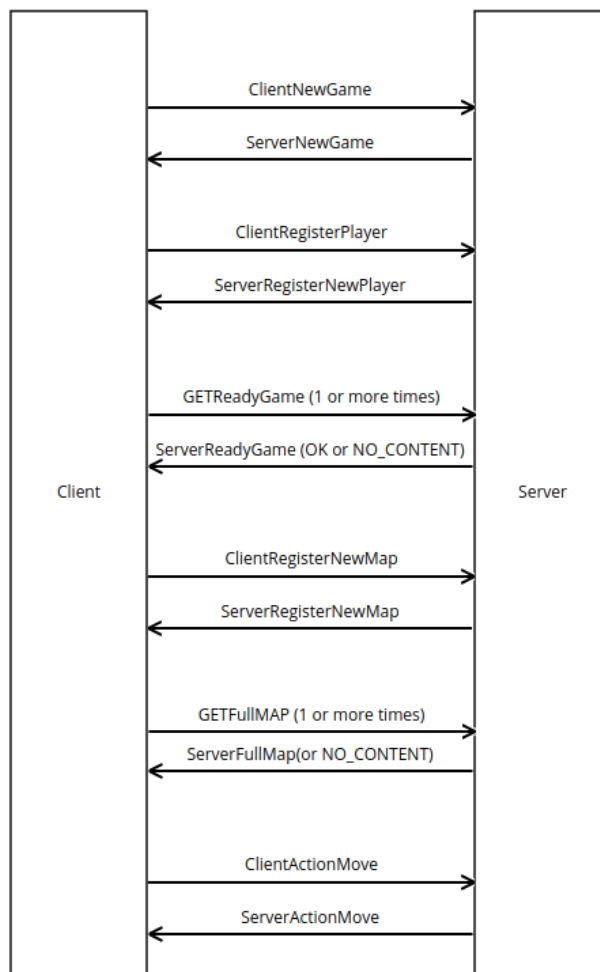
Das Protokoll soll es ermöglichen mittels REST Services (Jersey) und XML Nachrichten den Spielablauf der Clients über einen zentralen Server zu koordinieren und den Spielverlauf aufzuzeichnen.

Veränderungen zu Teilaufgabe 1:

Bei den relevanten Nachrichten wurden lt. Angabe zu Aufgabe 2 Zeitmarken (Start, Ende) hinzugefügt. Zu Game wurde eine `loser_id` sowie `players_no` hinzugefügt. `Loser_id` aufgrund einer einfacheren Implementierung. Ebenso `players_no` um zu überprüfen ob das Spiel schon bereit ist. Bei Player wurden `map_id` und `game_id` hinzugefügt, auch dies dient einer leichteren Implementierung, da ansonsten überflüssige Querys anfallen. Wenn in Nachrichten und Objekten fremde ID's auftauchen, bzw. mehrfach auftauchen dient dies ebenso diesem Grund, da Datenbankabfragen langsam sind. Zudem wurde berücksichtigt, dass der Server nicht selbstständig Nachrichten oder Überprüfungen durchführen sollte und somit Anfragen und Antworten paarweise auftreten im Gegensatz zu Teilaufgabe 1. Aufgrund dieser Änderungen und insbesondere dem Hinzufügen von fremden ID's in anderen Nachrichten/Objekten entfällt die Entity `play`.

Gegenüberstellung Anfragen und Antworten

Client	Server
<i>ClientNewGame</i>	ServerNewGame
ClientRegisterPlayer	ServerRegisterPlayer
GETReadyGame	ServerReadyGame
ClientRegisterNewMap	ServerRegisterNewMap
GETFullMap	ServerFullMap
ClientActionMove	ServerActionMove

SEQUENZDIAGRAMM

Definition der Nachrichten

Nachricht 1: ClientNewGame <http://localhost:7777/rest/startgame/id> (id == 1 || 2)

Diese Nachricht ist eine einfache GET Anfrage und enthält eine ID: 1- für Player 1 und 2 für Player 2. Wer Player 1 oder Player 2 ist wird vorher unter den Mitspielern ausgemacht (offline).

Nachricht2: ServerNewGame

Diese Nachricht ist die Antwort auf Nachricht 1 und enthält player_id, game_id und map_id welche dem jeweiligen Spieler 1 oder Spieler 2 zugeordnet durch den Server zugeordnet wurden.

Beispielnachricht für Nachricht 2 (ServerNewGame)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<registernewplayer>
  <player_id>716</player_id>
  <game_id>103</game_id>
  <map_id>165</map_id>
</registernewplayer>
```

Beispielschema Nachricht 2(ServerNewGame)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="registernewplayer"
type="registernewplayerType"/>
  <xs:complexType name="registernewplayerType">
    <xs:sequence>
      <xs:element type="xs:integer" name="player_id"/>
      <xs:element type="xs:integer" name="game_id"/>
      <xs:element type="xs:integer" name="map_id"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
</xs:complexType>
</xs:schema>
```

Nachricht 3: ClientRegisterPlayer

http://localhost:7777/rest/registernewplayer

Diese Nachricht dient der Registrierung eines Spielers. Die Nachricht enthält die `player_id`, den Vornamen, Nachnamen, das Alter und den Nicknamen des Spielers.

Beispielnachricht für Nachricht 3 (ClientRegisterPlayer)

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<registernewplayer>
  <player_id>716</player_id>
  <first_name>Benjamin</first_name>
  <last_name>Hartmann</last_name>
  <age>29</age>
  <nickname>ElMuldo</nickname>
  <game_id>103</game_id>
</registernewplayer>
```

Beispielschema für Nachricht 3 (ClientRegisterPlayer)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="registernewplayer"
    type="registernewplayerType"/>
  <xs:complexType name="registernewplayerType">
    <xs:sequence>
      <xs:element type="xs:integer" name="player_id"/>
      <xs:element type="xs:string" name="first_name"/>
      <xs:element type="xs:string" name="last_name"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
<xs:element type="xs:integer" name="age"/>
<xs:element type="xs:string" name="nickname"/>
<xs:element type="xs:integer" name="game_id"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

Nachricht 4: ServerRegisterNewPlayer

Der Server antwortet mit OK und dem Player als Nachricht mit all seinen Daten. Wenn die Nachricht nicht vollständig ist, antwortet der Server mit EXPECTATION_FAILED und einer Error Nachricht mit dem Inhalt „Missing attributes for Player!“. Bei anderen Fehlern wird der StackTrace in die Nachricht geschrieben. Eine Beispiellerrnachricht inklusive Schema ist in Nachricht 6 abgebildet.

Nachricht 5: GETReadyGame

http://localhost:7777/rest/readygame/player_id

Diese GET Nachricht enthält die player_id des Spielers um zu überprüfen, ob das Spiel „ready to start“ ist.

Nachricht 6: ServerReadyGame

Der Server antwort mit einem einfachen OK wenn das Spiel „ready to start“ ist, mit NO_CONTENT wenn das Spiel noch nicht fertig ist und mit EXPECTATION_FAILED und folgender Errornachricht:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Error>
  <message>Player not registered in a Game</message>
  <timestamp>2017-12-06 22:14:03.672</timestamp>
</Error>
```

Beispielschema Errornachricht

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="Error" type="ErrorType"/>

  <xs:complexType name="ErrorType">

    <xs:sequence>

      <xs:element type="xs:string" name="message"/>

      <xs:element type="xs:string" name="timestamp"/>

    </xs:sequence>

  </xs:complexType>

</xs:schema>
```

Nachricht 7: ClientRegisterNewMap

<http://localhost:7777/rest/registernewmap>

Nachdem sich beide Spieler beim Server gemeldet haben und zu einem Spiel zugeteilt wurden und das Spiel somit ready ist, schickt der jeweilige Client seine generierte Map dem Server.

Beispielnachricht Nachricht 7 (ClientRegisterNewMap)

```
<?xml version="1.0" encoding="UTF-8" ?>
<tiles>
  <time_end_generation>1512347937003</time_end_generation>
  <player_id>717</player_id>
  <tile>
    <tile_id></tile_id>
    <map_id>166</map_id>
    <type>2</type>
    <x>1</x>
```

```
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>2</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>2</type>
  <x>3</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>2</type>
  <x>4</x>
```

```
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>5</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>6</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>7</x>
```

```
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>1</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>2</x>
```

```
<y>2</y>

<castle>0</castle>

<treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>3</x>

  <y>2</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>4</x>

  <y>2</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>5</x>
```



```
<y>2</y>

<castle>0</castle>

<treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>6</x>

  <y>2</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>7</x>

  <y>2</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>8</x>
```

```
<y>2</y>

<castle>0</castle>

<treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>1</x>

  <y>3</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>2</x>

  <y>3</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>3</x>
```

```
<y>3</y>

<castle>0</castle>

<treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>4</x>

  <y>3</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>5</x>

  <y>3</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>6</x>
```

```
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>7</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>1</x>
```

```
<y>4</y>

<castle>0</castle>

<treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>2</x>

  <y>4</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>3</x>

  <y>4</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>4</x>
```

```
<y>4</y>

<castle>0</castle>

<treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>5</x>

  <y>4</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>6</x>

  <y>4</y>

  <castle>0</castle>

  <treasure>0</treasure>

</tile>

<tile>

  <tile_id></tile_id>

  <map_id>166</map_id>

  <type>1</type>

  <x>7</x>
```

```
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
</tiles>
```

Beispielschema Nachricht 7 (ClientNewMap)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="tiles" type="tilesType"/>
  <xs:complexType name="tileType">
    <xs:sequence>
      <xs:element type="xs:integer" name="tile_id"/>
      <xs:element type="xs:integer" name="map_id"/>
      <xs:element name="type">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="2"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
<xs:enumeration value="3"/>
<xs:enumeration value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="x">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="6"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="y">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```



```
</xs:simpleType>
</xs:element>
<xs:element type="xs:integer" name="castle"/>
<xs:element type="xs:integer" name="treasure"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tilesType">
  <xs:sequence>
    <xs:element type="xs:integer" name="time_end_generation"/>
    <xs:element type="xs:integer" name="player_id"/>
    <xs:element type="tileType" name="tile" maxOccurs="unbounded"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Nachricht 8 ServerNewMap

Wenn die Map, welche der Server vom Client erhalten hat, alle BusinessRules erfüllt und korrekt ist, sendet der Map ein OK inklusive der bereits enthaltenen Map im gleichen Format wie Nachricht 5 zurück. Allerdings hat der Server nun die Burg und den Schatz in die Nachricht eingetragen (vom Player der die Map generiert hat). Das Schema und die Nachricht bleiben diesselbe, weshalb hier auf die Beispielabbildungen verzichtet wird.

Nachricht 9 GETFullMap http://localhost:7777/rest/map/map_id

Der Client frag mit seiner Map_id an, ob der Server bereits die ander Hälfte der zum Spiel gehörenden Map erhalten hat.(Fortsetzung Nachricht 10)

Nachricht 10 ServerFullMap

Wenn ja, schickt der Server die Map des anderen Spielers zurück mit OK, wobei bei Treasure und Castle jeweils eine 0 statt eine eventuell

vorhandene 1 eingetragen wird um die Business Rules nicht zu verletzen. Auch hier wird auf die Abbildungen verzichtet, da diese denen in Nachricht 7 exakt gleichen. Sollte die Map des anderen noch nicht ready sein oder nicht vorhanden, antwortet der Server mit NO_CONTENT. Da die Überprüfung der BusinessRules und etwaiger Fehler bei der MapGeneration in der vorherigen Anfrage vom Server gemacht wurde, wird der Client bei einer gewissen Anzahl von NO_CONTENT Antworten abbrechen und weitere Schritte setzen.

Nachricht 11: ClientActionMove <http://localhost:7777/rest/map/action>

Der Client sendet nun seine gewünschte Action an den Server.

Beispielnachricht 11 (ClientActionMove)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<sendaction>
  <action_id>13</action_id>
  <player_id>716</player_id>
  <game_id>103</game_id>
  <map_id>165</map_id>
  <time_start>1512564993107</time_start>
  <time_end>1512564993108</time_end>
  <start_tile>259</start_tile>
  <target_tile>260</target_tile>
  <actions_left>1</actions_left>
  <inventory>0</inventory>
  <won>0</won>
</sendaction>
```

Beispielschema ClientActionMove

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="sendaction" type="sendactionType"/>
</xs:schema>
```

```
<xs:complexType name="sendactionType">
  <xs:sequence>
    <xs:element type="xs:integer" name="action_id"/>
    <xs:element type="xs:integer" name="player_id"/>
    <xs:element type="xs:integer" name="game_id"/>
    <xs:element type="xs:integer" name="map_id"/>
    <xs:element type="xs:string" name="time_start"/>
    <xs:element type="xs:string" name="time_end"/>
    <xs:element type="xs:integer" name="start_tile"/>
    <xs:element type="xs:integer" name="target_tile"/>
    <xs:element type="xs:integer" name="actions_left"/>
    <xs:element type="xs:integer" name="inventory"/>
    <xs:element type="xs:integer" name="won"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Nachricht 12: ServerActionMove

Der Server überprüft die enthaltene Nachricht 10 auf die Einhaltung der Businessrules. Wenn der Spieler nach einem ActionMove gleich noch einen abschickt und der andere Spieler noch nicht dran war, schickt der Server NO_CONTENT zurück. Wird eine BusinessRule nicht erfüllt oder der Spieler verloren hat, sendet der Server ein EXPECTATION_FAILED mit einer entsprechenden Error Nachricht welche bereits beispielhaft in Nachricht 5 dargestellt wurde. Bei Überschreiten der 3 Sekunden für eine Aktion ist der Text „Too slow, you lost!“. Bei Betreten von Wasser „Water, you lost!“, wenn auf regulärem Weg verloren wurde „You lost!“. Bei erfüllen aller BusinessRules sendet der Server die ActionMove Nachricht zurück, jeweils verändert um Actions_left und Inventory sowie evtl Won. Die Nachricht entspricht dem Inhalt und dem Schema ansonsten exakt Nachricht 9, weshalb hier wieder auf die Abbildung verzichtet wird.

Beispielhafte Beschreibung der Implementierung von Nachricht 1-3:

Nachricht 1: GETStartNewGame

Umsetzung als GET Service mit der URL

<http://localhost:7777/rest/startgame/id>

Relevante Klassen: Game, GameDAO, Map, MapDAO, Player, PlayerDAO, Entity Manager Factory, Entity Manager.

Der Spieler welcher im Client festgelegt hat, dass er Spieler 1 sein möchte, sendet 1 als ID, Spieler 2 sendet 2 als ID.

Ablauf ID = 1:

Das GameDAO erstellt ein neues game Objekt welches somit eine ID aus der Datenbank zugewiesen bekommt. Mit diesem Game Objekt wird ein Map Objekt durch das MapDAO erzeugt, welches somit mit diesem assoziiert wird. Mit diesen beiden Objekten wird durch das PlayerDAO ein neuer Player erzeugt, der nun eine Assoziation mit diesen beiden aufweist. Zudem wird die Player_id von diesem Player Objekt durch den setter der Map und den getter von Player zugewiesen und das Map Objekt in der DB aktualisiert. Danach wird ein Response Objekt mit Status OK und dem Player Objekt als XML erzeugt.

Ablauf ID = 2:

Das GameDAO erstellt ein neues game Objekt indem es mittels der Methode findOpenGame() das Spiel von Spieler 1 findet. Mit diesem Game Objekt wird ein Map Objekt durch das MapDAO erzeugt, welches somit mit diesem assoziiert wird. Mit diesen beiden Objekten wird durch das PlayerDAO ein neuer Player erzeugt, der nun eine Assoziation mit diesen beiden aufweist. Zudem wird die Player_id von diesem Player Objekt durch den setter der Map und den getter von Player zugewiesen und das Map Objekt in der DB aktualisiert. Danach wird ein Response Objekt mit Status OK und dem Player Objekt als XML erzeugt.

Ablauf andere ID:

Es wird der Status NOT_ACCEPTABLE als Response geschickt.

Bei Fehlern wird eine Error XML Nachricht mit „No ID specified or open game found“ sowie Timestamp geschickt.

Nachricht 2: PostRegisterNewPlayer

Umsetzung als POST Service mit der URL

<http://localhost:7777/rest/registernewplayer>

Relevante Klassen: Game, GameDAO, Player, PlayerDAO, Entity Manager Factory, Entity Manager.

Ablauf:

Es wird ein Player Objekt als XML an den Server geschickt. Dieser ruft die Methode insertPlayerData(player) von PlayerDAO auf um die Daten des Spielers (First Name, Last Name ...) zur dazugehörigen Player_ID einzutragen. Danach wird von GameDAO die Methode iterateGamePlayerNo(game_id) mit der game_id des Players aufgerufen, um den Spielerzähler des Spiels um 1 zu erhöhen. Dieser gibt die bereits vollständig registrierten Spieler zu einem Game an. Wenn dieser auf 2 steht ist das Game „ready“. Dies wird hier aber noch nicht überprüft. Dann wird der player mit den eingetragenen Daten zurückgeschickt mit Status OK. Bei fehlenden Daten wird ein Error XML Response mit Timestamp und „Missing attributes for Player“ geschickt. Bei sonstigen Fehlern der Stack Trace. Beides mit Status „EXPECTATION FAILED“.

Nachricht 3: GETReadyGame

Umsetzung als GET Service mit der URL

http://localhost:7777/rest/readygame/player_id

Relevante Klassen: Game, GameDAO, Map, MapDAO, Player, PlayerDAO, Entity Manager Factory, Entity Manager.

Ablauf:

Das PlayerDAO sucht den player mit der player_id in der DB und weist diesem dem Objekt zu. Mit der game_id des Players findet das GameDAO das zugehörige Spiel und weist dieses dem Game Objekt zu. Zudem wird ein Integer player_no mit der player_no des Game Objekts initialisiert, welches angibt, wieviel Spieler bereits vollständig im Spiel registriert sind. Wenn 2 Spieler registriert sind und das Spiel „ready“ ist, wird ein OK als Response zurückgeschickt. Ansonsten ein NO_CONTENT. Bei einer unbekannten player_id wird ein EXPECTATION_FAILED mit einer Error XML Message „Player not registered in a Game“ und einem Timestamp geschickt.

Beispielhafter Ablauf von Spielen unter der Verwendung der definierten Nachrichten

<http://localhost:7777/rest/startgame/1>

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<registernewplayer>  
  <player_id>716</player_id>  
  <game_id>103</game_id>  
  <map_id>165</map_id>  
</registernewplayer>
```

<http://localhost:7777/rest/startgame/2>

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<registernewplayer>  
  <player_id>717</player_id>  
  <game_id>103</game_id>  
  <map_id>166</map_id>  
</registernewplayer>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<registernewplayer>  
  <player_id>716</player_id>  
  <first_name>Benjamin</first_name>  
  <last_name>Hartmann</last_name>  
  <age>29</age>  
  <nickname>ElMuldo</nickname>  
  <game_id>103</game_id>  
</registernewplayer>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<registernewplayer>
```

```
<player_id>717</player_id>
<first_name>Tobias</first_name>
<last_name>Hartmann</last_name>
<age>29</age>
<nickname>Maeuerle</nickname>
<game_id>103</game_id>
</registernewplayer>
```

<http://localhost:7777/rest/readygame/716>

OK

<http://localhost:7777/rest/readygame/717>

OK

<?xml version="1.0" encoding="UTF-8" ?>

```
<tiles>
  <time_end_generation>1512347937003</time_end_generation>
  <player_id>717</player_id>
  <tile>
    <tile_id></tile_id>
    <map_id>166</map_id>
    <type>2</type>
    <x>1</x>
    <y>1</y>
    <castle>0</castle>
    <treasure>0</treasure>
  </tile>
  <tile>
    <tile_id></tile_id>
    <map_id>166</map_id>
```



```
<type>3</type>
<x>2</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>2</type>
  <x>3</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>2</type>
  <x>4</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>3</type>
<x>5</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>6</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>7</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>8</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>1</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>2</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>3</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>4</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>5</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>6</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>7</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>1</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>2</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>3</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>4</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>5</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>6</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>7</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>1</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```



```
<type>1</type>
<x>2</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>3</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>4</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>5</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>6</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>7</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
```

```
<type>1</type>
<x>8</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
</tiles>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<tiles>
  <time_end_generation>1512347937003</time_end_generation>
  <player_id>717</player_id>
  <tile>
    <tile_id></tile_id>
    <map_id>165</map_id>
    <type>2</type>
    <x>1</x>
    <y>1</y>
    <castle>0</castle>
    <treasure>0</treasure>
  </tile>
  <tile>
    <tile_id></tile_id>
    <map_id>165</map_id>
    <type>3</type>
    <x>2</x>
    <y>1</y>
    <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>2</type>
  <x>3</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>2</type>
  <x>4</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>3</type>
  <x>5</x>
  <y>1</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>3</type>
  <x>6</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>3</type>
  <x>7</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>1</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>1</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>2</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>3</x>
  <y>2</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>4</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>5</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>6</x>
  <y>2</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>7</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>1</x>
  <y>3</y>
  <castle>0</castle>
```



```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>2</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>3</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>4</x>
  <y>3</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>5</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>6</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>7</x>
  <y>3</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>1</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>2</x>
  <y>4</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>3</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>4</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>5</x>
  <y>4</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>6</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>7</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>4</y>
  <castle>0</castle>
```

```
<treasure>0</treasure>
</tile>
</tiles>
```

<http://localhost:7777/rest/map/165>

<?xml version="1.0" encoding="UTF-8" ?>

```
<tiles>
  <time_end_generation>1512347937003</time_end_generation>
  <player_id>717</player_id>
  <tile>
    <tile_id></tile_id>
    <map_id>166</map_id>
    <type>2</type>
    <x>1</x>
    <y>1</y>
    <castle>0</castle>
    <treasure>0</treasure>
  </tile>
  <tile>
    <tile_id></tile_id>
    <map_id>166</map_id>
    <type>3</type>
    <x>2</x>
    <y>1</y>
    <castle>0</castle>
    <treasure>0</treasure>
  </tile>
  <tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>2</type>
<x>3</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>2</type>
  <x>4</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>5</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>3</type>
<x>6</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>3</type>
  <x>7</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```



```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>1</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>2</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>3</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>4</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>5</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>6</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>7</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>1</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>2</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>3</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>4</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>5</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>6</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>7</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>8</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>1</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>2</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>3</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>4</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>5</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
```

```
<tile_id></tile_id>
<map_id>166</map_id>
<type>1</type>
<x>6</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>7</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>166</map_id>
  <type>1</type>
  <x>8</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
</tiles>
```


<http://localhost:7777/rest/map/166>

<?xml version="1.0" encoding="UTF-8" ?>

```
<tiles>
  <time_end_generation>1512347937003</time_end_generation>
  <player_id>716</player_id>
  <tile>
    <tile_id></tile_id>
    <map_id>165</map_id>
    <type>2</type>
    <x>1</x>
    <y>1</y>
    <castle>0</castle>
    <treasure>0</treasure>
  </tile>
  <tile>
    <tile_id></tile_id>
    <map_id>165</map_id>
    <type>3</type>
    <x>2</x>
    <y>1</y>
    <castle>0</castle>
    <treasure>0</treasure>
  </tile>
  <tile>
    <tile_id></tile_id>
    <map_id>165</map_id>
```

```
<type>2</type>
<x>3</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>2</type>
  <x>4</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>3</type>
  <x>5</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>3</type>
<x>6</x>
<y>1</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>3</type>
  <x>7</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>1</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>1</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>2</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>3</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>4</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>5</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>6</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>7</x>
<y>2</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>2</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>1</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>2</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>3</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>4</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>5</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>6</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>7</x>
  <y>3</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```



```
<type>1</type>
<x>8</x>
<y>3</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>1</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>2</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>3</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>4</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>5</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
```

```
<type>1</type>
<x>6</x>
<y>4</y>
<castle>0</castle>
<treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>7</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
<tile>
  <tile_id></tile_id>
  <map_id>165</map_id>
  <type>1</type>
  <x>8</x>
  <y>4</y>
  <castle>0</castle>
  <treasure>0</treasure>
</tile>
</tiles>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<sendaction>
```

```
<action_id></action_id>
<player_id>716</player_id>
<game_id>103</game_id>
<map_id>165</map_id>
<time_start>1512564993107</time_start>
<time_end>1512564993108</time_end>
<start_tile>259</start_tile>
<target_tile>260</target_tile>
<actions_left>2</actions_left>
<inventory></inventory>
<won></won>
</sendaction>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<sendaction>
  <action_id>13</action_id>
  <player_id>716</player_id>
  <game_id>103</game_id>
  <map_id>165</map_id>
  <time_start>1512564993107</time_start>
  <time_end>1512564993108</time_end>
  <start_tile>259</start_tile>
  <target_tile>260</target_tile>
  <actions_left>1</actions_left>
  <inventory>0</inventory>
  <won>0</won>
</sendaction>
```

....

```
<?xml version="1.0" encoding="UTF-8" ?>
```

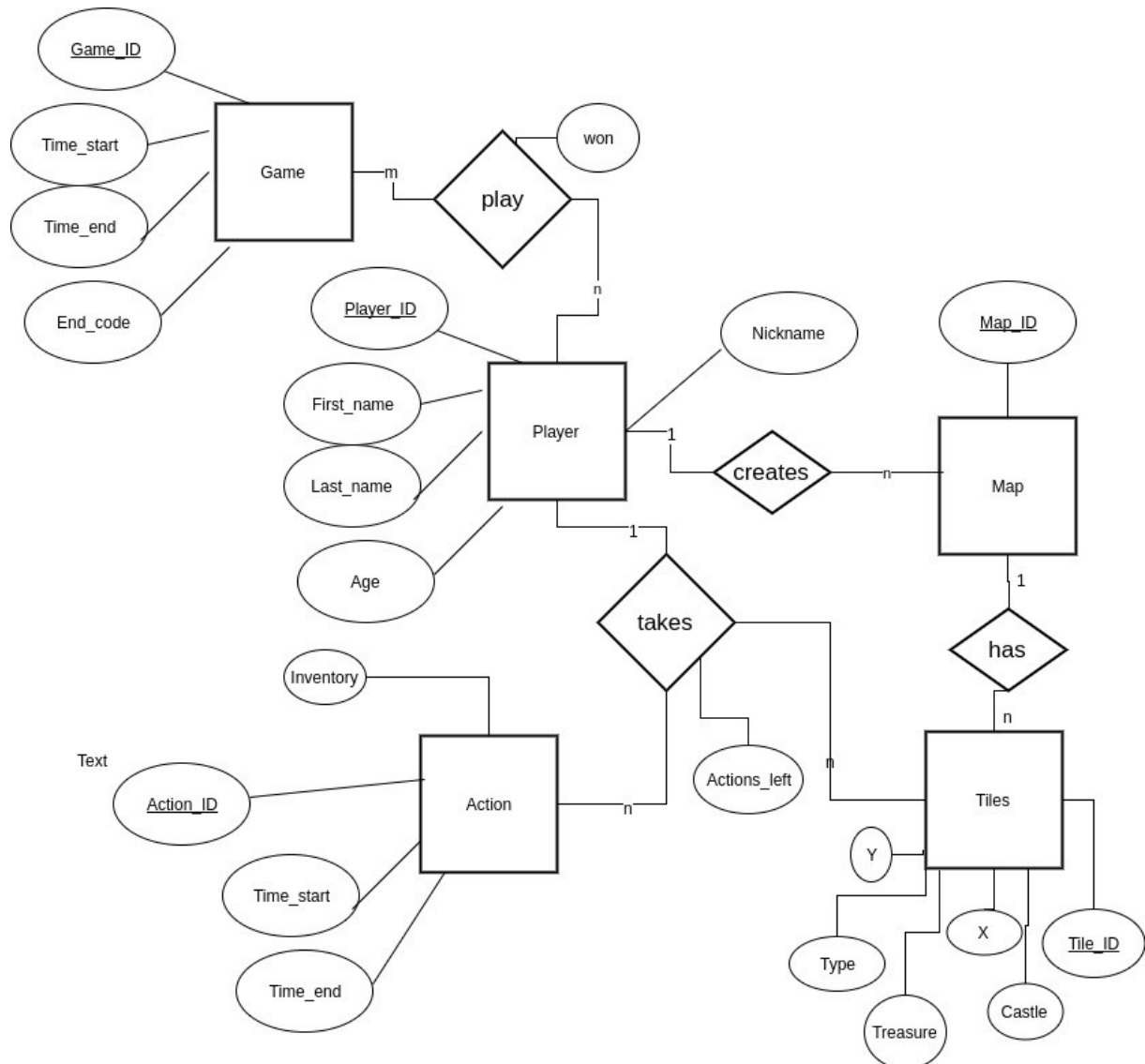
```
<sendaction>
  <action_id></action_id>
  <player_id>717</player_id>
  <game_id>103</game_id>
  <map_id>165</map_id>
  <time_start>1512564993107</time_start>
  <time_end>1512564993108</time_end>
  <start_tile>256</start_tile>
  <target_tile>257</target_tile>
  <actions_left>1</actions_left>
  <inventory>1</inventory>
  <won></won>
</sendaction>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<sendaction>
  <action_id>13</action_id>
  <player_id>716</player_id>
  <game_id>103</game_id>
  <map_id>165</map_id>
  <time_start>1512564993107</time_start>
  <time_end>1512564993108</time_end>
  <start_tile>259</start_tile>
  <target_tile>260</target_tile>
  <actions_left>0</actions_left>
  <inventory>1</inventory>
  <won>1</won>
```

</sendaction>

Persistierung



Die Datenbank zur Persistierung bildet in der Tabelle Player die jeweiligen Spieler ab, die am Spiel teilgenommen haben. Die Tabelle Game bildet alle bereits gespielten Spiele ab. End Code bezeichnet hierbei den Grund für das Spielende ab. (bspw. 0 für regulärer Sieg, 1 für Regelverletzung bei der Dauer der Spielaktionen etc.). Die Tabelle Map enthält die Map_ID, Player_ID sowie Game_ID um sie zuordnen zu können. Die Tabelle Tiles bildet die einzelnen Spielfelder ab mit Ihren XY Koordinaten, ob eine Burg drauf steht, ob der Schatz drauf steht, welcher Typ der Tile hat (Wasser, Gras etc). Die Tabelle Action zeigt schließlich die jeweiligen Spielzüge und die Start sowie Endzeitpunkt der Aktion an. Um darzustellen, dass immer ein Start und ein Ziel Tile sowie ein Spieler Teil einer Action ist, wird die Tabelle Action um die Player_ID, Start_tile, Target_tile, actions_left (wieviel actions noch um das target zu erreichen?) sowie inventory (ist der Schatz in Besitz?) erweitert.

SQL Statements:

```
CREATE TABLE `player` (  
    `player_id` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,  
    `first_name` TEXT,  
    `last_name` TEXT,  
    `age` INTEGER,  
    `nickname` TEXT,  
    `game_id` INTEGER,  
    `map_id` INTEGER,  
    `time_start` INTEGER,  
    FOREIGN KEY(`game_id`) REFERENCES `game`(`game_id`),  
    FOREIGN KEY(`map_id`) REFERENCES `map`(`map_id`)  
);
```

```
CREATE TABLE `game` (  
    `game_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    `loser_id` INTEGER,  
    `winner_id` INTEGER,  
    `time_start` TEXT,  
    `time_end` TEXT,  
    `end_code` INTEGER,  
    `players_no` INTEGER,  
    FOREIGN KEY(`loser_id`) REFERENCES `player`(`player_id`),  
    FOREIGN KEY(`winner_id`) REFERENCES `player`(`player_id`)  
);
```

```
CREATE TABLE `map` (  
    `map_id` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,
```



```
`game_id` INTEGER,  
`player_id` INTEGER,  
`time_start_generation` TEXT,  
`time_end_generation` TEXT,  
FOREIGN KEY(`game_id`) REFERENCES `game`(`game_id`),  
FOREIGN KEY(`player_id`) REFERENCES `player`(`player_id`)  
);
```

```
CREATE TABLE `tile` (  
  `tile_id` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,  
  `map_id` INTEGER,  
  `type` REAL,  
  `x` INTEGER,  
  `y` INTEGER,  
  `castle` INTEGER,  
  `treasure` INTEGER,  
  FOREIGN KEY(`map_id`) REFERENCES `map`(`map_id`)  
);
```

```
CREATE TABLE `action` (  
  `action_id` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,  
  `player_id` INTEGER,  
  `game_id` NUMERIC,  
  `map_id` REAL,  
  `time_start` TEXT,  
  `time_end` TEXT,  
  `start_tile` INTEGER,
```

```
`target_tile`    INTEGER,  
`actions_left`   INTEGER,  
`inventory`      INTEGER,  
`won`            INTEGER,  
FOREIGN KEY(`player_id`) REFERENCES `player`(`player_id`),  
FOREIGN KEY(`game_id`) REFERENCES `game`(`game_id`),  
FOREIGN KEY(`map_id`) REFERENCES `map`(`map_id`),  
FOREIGN KEY(`start_tile`) REFERENCES `tile`(`tile_id`),  
FOREIGN KEY(`target_tile`) REFERENCES `tile`(`tile_id`)  
);
```

```
INSERT into player values(NULL, Arnebald,Berne,18,baldi,1,1,'2017-11-01  
19:02:15');
```

```
INSERT into player values(NULL, Anna,Bader,10,annshecan1,2,'2017-11-01  
19:02:15');
```

```
INSERT into player values(NULL, Arne,Brot,18,chef,2,3,'2017-11-01  
19:02:16');
```

```
INSERT into player values(NULL, Affe,Bart,11,affi,2,4,'2017-11-01  
19:02:16');
```

```
INSERT into player values(NULL, Angelika,Bog,22,angi,3,5,'2017-11-01  
19:02:17');
```

```
INSERT into player values(NULL, Bettina,Sax,24,betty,3,6,'2017-11-01  
19:02:17');
```

```
INSERT into player values(NULL, Arnold,Bader,18,arni,4,7,'2017-11-01  
19:02:18');
```

```
INSERT into player values(NULL, Arnold,Basten,18,arnoldius,4,8,'2017-11-  
01 19:02:18');
```

```
INSERT into player values(NULL, Caesar,Augustus,22,romi,5,9,'2017-11-01  
19:02:18');
```

```
INSERT into player values(NULL, Julia,Sprenger,29,sprengi,5,10,'2017-11-01 19:02:18');
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-01 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-02 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-03 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-04 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-05 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-06 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-07 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-08 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-09 19:00:05',NULL,NULL,0);
```

```
INSERT into game values(NULL,NULL,NULL,'2017-11-10 19:00:05',NULL,NULL,0);
```

```
INSERT into action values(NULL,1,1,1,'2017-11-01 18:00:05','2017-11-01 18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,2,1,2,'2017-11-01 18:00:05','2017-11-01 18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,3,2,3,'2017-11-01 18:00:05','2017-11-01 18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,4,2,4,'2017-11-01 18:00:05','2017-11-01 18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,5,3,5,'2017-11-01 18:00:05','2017-11-01  
18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,6,3,6,'2017-11-01 18:00:05','2017-11-01  
18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,7,4,7,'2017-11-01 18:00:05','2017-11-01  
18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,8,4,8,'2017-11-01 18:00:05','2017-11-01  
18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,9,5,9,'2017-11-01 18:00:05','2017-11-01  
18:00:06',11,12, ,1,0,1);
```

```
INSERT into action values(NULL,10,5,10,'2017-11-01 18:00:05','2017-11-01  
18:00:06',11,12, ,1,0,1);
```

```
INSERT into map_id values(NULL,1,1,1,'2017-11-01 18:00:06','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,1,2,'2017-11-01 18:00:07','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,2,3,'2017-11-01 18:00:08','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,2,4,'2017-11-01 18:00:09','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,3,5,'2017-11-01 18:00:10','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,3,6,'2017-11-01 18:00:11','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,4,7,'2017-11-01 18:00:12','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,4,8,'2017-11-01 18:00:13','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,5,9,'2017-11-01 18:00:14','2017-11-01  
18:00:17');
```

```
INSERT into map_id values(NULL,1,5,10,'2017-11-01 18:00:15','2017-11-01  
18:00:17');
```

```
INSERT into tile values(NULL,1,1,2,1,0,0);
INSERT into tile values(NULL,1,1,3,1,0,0);
INSERT into tile values(NULL,1,1,4,1,0,0);
INSERT into tile values(NULL,1,1,1,2,0,0);
INSERT into tile values(NULL,1,1,1,3,0,0);
INSERT into tile values(NULL,1,1,1,4,0,0);
INSERT into tile values(NULL,1,1,1,5,0,0);
INSERT into tile values(NULL,1,1,1,6,0,0);
INSERT into tile values(NULL,1,1,1,7,0,0);
INSERT into tile values(NULL,1,1,1,8,0,0);
```

Beispielhafte Umsetzung von 4 SQL Prepared Statements Abfragen

Alle Abfragen wurden mittels Hibernate/JPA und NamedQueries umgesetzt.
Die Parameter wurden dann an anderer Stelle im Programmablauf gesetzt.

Map:

@NamedQueries({

```
@NamedQuery(
    name = "get_map_by_player_id",
    query = "select map_id from Map where player_id = :id"
),
@NamedQuery(
    name = "get_other_map",
    query = "from Map where game_id = :game_id and player_id <>
:player_id"
)
})
```

Die Query dient dazu, eine Map_id zurückzugeben welche zu einer bestimmten Player_id gehört.

Implementierung:

MapDAO:

In der Methode findOtherMapByGame_ID wird mit der übergebenen map_id eine neue Map initialisiert sowie die game_id und die player_id. Danach wrd mit dem entityManager die Named Query mit diesen Parametern ausgeführt bzw. erzeugt und anschließend returniert.

```
public Map findOtherMapByGame_id(Integer map_id) throws Exception{
    Integer player_id = 0;
    Map map = new Map();
    Map other_map = new Map();
    Integer game_id = 0;
    map = findMapById(map_id);
    game_id = map.getGame_id();
    player_id=map.getPlayer_id();
    System.out.println(player_id);

    other_map = entityManager.createNamedQuery("get_other_map",
Map.class).setParameter("game_id",game_id).setParameter("player_id",pla
yer_id).getSingleResult();

    return other_map;
}
```

GETFullIMAP:

In dieser Methode wird eine Map mit dieser Query initialisiert um dem Spieler die andere Hälfte der Map zu einem Spiel zurückzuliefern.

```
map=mapDAO.findOtherMapByGame_id(map_id);
```

GAME:

```
@NamedQueries({
```

```
    @NamedQuery(
        name = "get_open_game",
```

```
query = "select max (game_id) from Game"
),
```

Mit dieser Query wird vom letzten erzeugten Spiel aus der DB die game_id retourniert.

GameDAO

Die Methode findOpenGame() erzeugt aus der erhaltenen game_id ein neues Game objekt, indem er mit der game_id das Game in der DB sucht und anschließend retourniert.

```
public Game findOpenGame() throws Exception{
```

```
    Integer game_id = 0;

    game_id= entityManager.createNamedQuery("get_open_game",
Integer.class).getSingleResult();

    Game game = findGameByID(game_id);

    return game;
}
```

GETStartNewGame

Wenn GETStartNewGame mit der ID 2 als Spieler 2 aufgerufen wird, wird das offene Spiel von Spieler 1 gesucht (das letzte eröffnete Spiel) und im weiteren Programmverlauf dem Spieler zugewiesen.

```
else if (id == 2) {
```

```
    Game game2 = new Game(gameDAO.findOpenGame());
```

Action Move (2 Queries)

```
@NamedQueries({
```

```
    @NamedQuery(
        name = "get_last_move",
        query = "select max (action_id) from ActionMove"
    ),

    @NamedQuery(
        name = "count_actions",
```

```
        query = "SELECT count (action_id) from ActionMove where  
game_id = :game_id"  
    ),
```

Die 1. Query gibt die Action_ID der letzten ausgeführten Action zurück.

Die 2. Query zählt die Actions eines bestimmten Games mit game_id.

ActionMoveDAO

Die Methode findLastAction findet die letzte Action eines bestimmten Spiels indem es die von Query 1 gefundene Action_id benutzt um die den zugehörigen ActionMove in der DB zu finden und zu retournieren.

Die Methode countActions retourniert die Anzahl der bereits vergangenen Transaktionen in einem Spiel indem es Query 2 mit der game_id des übergebenen ActionMove aufruft und das Ergebnis als long retourniert.

```
public ActionMove findLastAction() throws Exception{
```

```
    Integer action_id;  
  
    action_id= entityManager.createNamedQuery("get_last_move",  
Integer.class).getSingleResult();  
  
    ActionMove actionMove = new ActionMove();  
  
    actionMove = findActionMoveByID(action_id);  
  
    return actionMove;  
}
```

```
public Long countActions(ActionMove actionMove) throws Exception{
```

```
    Long count;  
  
    count= entityManager.createNamedQuery("count_actions",  
Long.class).setParameter("game_id",actionMove.getGame_id()).getSingleResult();  
  
    return count;  
}
```

```
//UPDATE
```

POSTActionMove

Die beiden Queries werden hier zB verwendet um herauszufinden ob es schon eine Aktion in diesem Spiel gegeben hat. Wenn ja, dann wird der

letzte ActionMove gesucht und überprüft ob dieser von einem anderen Spieler durchgeführt wurde um die Einhaltung eines abwechselnden Spielablaufs zu garantieren.

```
if(actionMoveDAO.countActions(actionMove) != 0){
```

```
    otherplayermove = actionMoveDAO.findLastAction();
```

```
if(otherplayermove.getPlayer_id().equals(actionMove.getPlayer_id())) {  
    //other player's turn
```

Beispiel um den Gewinner eines Spiels zu ermitteln in der Klasse Game:

```
@NamedQuery(  
    name = "get_game_id_winner",  
    query = "select winner_id from Game where game_id = :game_id"
```

```
)
```

Beispiel um die 10. und 20. Action eines Spieles zu finden:

```
@NamedQuery(  
    name = "get_actions_by_game_id",  
    query = "from ActionMove where game_id = :game_id"
```

```
),
```

Umsetzung Query 2:

```
public ActionMove findActions(ActionMove actionMove, Integer number)  
throws Exception{
```

```
    ActionMove actionMove1 = new ActionMove();  
    Query query =  
entityManager.createNamedQuery("get_actions_by_game_id",  
ActionMove.class);  
    List<ActionMove> list = query.getResultList();  
    actionMove = list.get(number);  
    return actionMove;  
}
```

Es werden alle Aktionen eines bestimmten Spiels aus der Datenbank in eine Liste gelesen und dann wird aus dieser Liste je nach übergebener Number die Aktion Nummer X (10 oder 20 oder beliebig) retourniert.

Umsetzung Query 1:

```
public Integer findWinner(Game game) throws Exception{
```

```
    Integer winner_id = 0;

    winner_id= entityManager.createNamedQuery("get_game_id_winner",
Integer.class).setParameter("game_id",game.getGame_id()).getSingleResult();

    return winner_id;
}
```

Es wird die Query in der Methode findWinner mit der Gameld des übergebenen Game Objekt ausgeführt und die winner_id retourniert.

Exportierung der Umsetzung:

Klasse XMLFinishedGame zur Zuweisung der Felder zu XML Elementen:

```
package server;
```

```
import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "XMLFinishedGame")
@XmlAccessorType(XmlAccessType.FIELD)
public class XMLFinishedGame {

    @XmlElement
    public ActionMove actionMove10;

    @XmlElement
    public ActionMove actionMove20;

    @XmlElement
```

```
public Integer winner_id;

public ActionMove getActionMove10() {
    return actionMove10;
}

public void setActionMove10(ActionMove actionMove10) {
    this.actionMove10 = actionMove10;
}

public ActionMove getActionMove20() {
    return actionMove20;
}

public void setActionMove20(ActionMove actionMove20) {
    this.actionMove20 = actionMove20;
}

public Integer getWinner_id() {
    return winner_id;
}

public void setWinner_id(Integer winner_id) {
    this.winner_id = winner_id;
}
}
```

POSTAction Move

Wenn das Spiel regulär verloren wird, speichert der Server die 10., die 20. und den Gewinner in ein XML-File ab.

```
}else if(otherplayermove.getWon() == 1) {
```

```
    Error error = new Error();
    error.setMessage("You lost!");
    Integer winner_id = gameDAO.findWinner(game);
```

```
    ActionMove actionMove10 =  
actionMoveDAO.findActions(actionMovetoDB,10);  
  
    ActionMove actionMove20 =  
actionMoveDAO.findActions(actionMovetoDB,20);  
  
    XMLFinishedGame xmlFinishedGame = new XMLFinishedGame();  
  
    xmlFinishedGame.setActionMove10(actionMove10);  
  
    xmlFinishedGame.setActionMove20(actionMove20);  
  
    xmlFinishedGame.setWinner_id(winner_id);  
  
    JAXBContext contextObj =  
JAXBContext.newInstance(XMLFinishedGame.class);  
  
    Marshaller marshallerObj = contextObj.createMarshaller();  
  
    marshallerObj.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);  
  
    marshallerObj.marshal(xmlFinishedGame, new  
FileOutputStream("XMLFinished.xml"));  
  
    return  
Response.status(Response.Status.EXPECTATION_FAILED).entity(error).build  
();  
}
```

Beispiel:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<XMLFinishedGame>  
  <actionMove10>  
    <action_id>28</action_id>  
    <player_id>717</player_id>  
    <game_id>103</game_id>  
    <map_id>165</map_id>  
    <time_start>1512564993107</time_start>  
    <time_end>1512564993108</time_end>  
    <start_tile>259</start_tile>  
    <target_tile>260</target_tile>
```

```

    <actions_left>0</actions_left>
    <inventory>0</inventory>
    <won>0</won>
  </actionMove10>
  <actionMove20>
    <action_id>29</action_id>
    <player_id>716</player_id>
    <game_id>103</game_id>
    <map_id>165</map_id>
    <time_start>1512564993107</time_start>
    <time_end>1512564993108</time_end>
    <start_tile>259</start_tile>
    <target_tile>260</target_tile>
    <actions_left>0</actions_left>
    <inventory>0</inventory>
    <won>0</won>
  </actionMove20>
  <winner_id>716</winner_id>
</XMLFinishedGame>

```

Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="XMLFinishedGame"
    type="XMLFinishedGameType"/>

  <xs:complexType name="actionMove10Type">

    <xs:sequence>

```

```
<xs:element type="xs:integer" name="action_id"/>
<xs:element type="xs:integer" name="player_id"/>
<xs:element type="xs:integer" name="game_id"/>
<xs:element type="xs:integer" name="map_id"/>
<xs:element type="xs:integer" name="time_start"/>
<xs:element type="xs:integer" name="time_end"/>
<xs:element type="xs:integer" name="start_tile"/>
<xs:element type="xs:integer" name="target_tile"/>
<xs:element type="xs:integer" name="actions_left"/>
<xs:element type="xs:integer" name="inventory"/>
<xs:element type="xs:integer" name="won"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="actionMove20Type">
  <xs:sequence>
    <xs:element type="xs:integer" name="action_id"/>
    <xs:element type="xs:integer" name="player_id"/>
    <xs:element type="xs:integer" name="game_id"/>
    <xs:element type="xs:integer" name="map_id"/>
    <xs:element type="xs:integer" name="time_start"/>
    <xs:element type="xs:integer" name="time_end"/>
    <xs:element type="xs:integer" name="start_tile"/>
    <xs:element type="xs:integer" name="target_tile"/>
    <xs:element type="xs:integer" name="actions_left"/>
    <xs:element type="xs:integer" name="inventory"/>
    <xs:element type="xs:integer" name="won"/>
  </xs:sequence>
```

```
</xs:complexType>
<xs:complexType name="XMLFinishedGameType">
  <xs:sequence>
    <xs:element type="actionMove10Type" name="actionMove10"/>
    <xs:element type="actionMove20Type" name="actionMove20"/>
    <xs:element type="xs:integer" name="winner_id"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Unit Tests:

Screenshot (Test Coverage 51%, 11 Unit Tests)

75 classes, 51% lines covered in package 'server'

Element	Class, %	Method, %	Line, %
ActionMove	100% (1/1)	88% (2/2)	58% (15/26)
ActionMoveDAO	0% (0/1)	0% (0/0)	0% (0/1)
Error	100% (1/1)	100% (8/8)	100% (15/15)
Game	100% (1/1)	100% (17/17)	100% (34/34)
GameDAO	100% (1/1)	57% (4/7)	46% (13/28)
GETFullMap	100% (1/1)	100% (1/1)	83% (15/18)
GETReadyGame	100% (1/1)	100% (1/1)	95% (12/12)
GETStartNewGame	100% (1/1)	100% (1/1)	100% (34/34)
Map	100% (1/1)	100% (17/17)	94% (36/38)
MapDAO	100% (1/1)	66% (4/6)	60% (20/33)
Player	100% (1/1)	100% (18/18)	100% (38/38)
PlayerDAO	100% (1/1)	60% (3/5)	42% (9/21)
POSTActionMove	0% (0/1)	0% (0/1)	0% (0/0)
POSTMapTiles	0% (0/1)	0% (0/1)	0% (0/0)
POSTRegisterNewPlayer	0% (0/1)	0% (0/1)	0% (0/0)
StandaloneRESTServer	0% (0/1)	0% (0/2)	0% (0/9)
Tile	100% (1/1)	100% (16/16)	100% (30/30)
TileDAO	100% (1/1)	25% (1/4)	18% (3/16)
TileList	100% (1/1)	100% (7/7)	100% (10/10)
XMLFinishedGame	100% (1/1)	100% (6/6)	100% (10/10)

Ich habe von allen Klassen mit getter und setter Methoden diese getestet. Der naheliegendste Grund war die Test Coverage. Ich habe in jeder dieser Testmethoden einen Integer oder String value initialisiert und diesen mit den set methoden zugewiesen. Danach habe ich verglichen ob diese Übereinstimmen mittels der get Methoden und `assertEquals(expected, object.get())`.

Zudem habe ich folgende GET Services bzw. Klassen getestet:

```
public class GETFullMapTest {
```

```
@Test
public void getMsg() throws Exception {
    GETFullMap getFullMap = new GETFullMap();
    assertEquals(204, getFullMap.getMsg(715).getStatus());
    assertEquals(200, getFullMap.getMsg(165).getStatus());
}
}
```

Hier rufe ich den GET Service einmal mit einer player_id auf, die eine Map zurückliefern kann, also den Status OK zurückliefert und einmal mit eine

player_id, bei der ein Fehler auftreten sollte der den Status 204 zurückliefert.

Dieselbe Logik wurde bei folgenden GET Service Tests ebenso angewandt:

```
public class GETReadyGameTest {
```

```
    @Test
    public void getMsg() throws Exception {
        GETReadyGame getReadyGame = new GETReadyGame();
        assertEquals(200, getReadyGame.getMsg(716).getStatus());
        assertEquals(417, getReadyGame.getMsg(7160).getStatus());
    }
}
```

```
public class GETStartNewGameTest {
```

```
    @Test
    public void getMsg() throws Exception {
        GETStartNewGame getStartNewGame = new GETStartNewGame();
        Player player = new Player();
        assertEquals(200, getStartNewGame.getMsg(1).getStatus());
        GETStartNewGame getStartNewGame1 = new GETStartNewGame();
        assertEquals(200, getStartNewGame1.getMsg(2).getStatus());
        GETStartNewGame getStartNewGame2 = new GETStartNewGame();
        assertEquals(406, getStartNewGame2.getMsg(3).getStatus());
        GETStartNewGame getStartNewGame3 = new GETStartNewGame();
        assertEquals(417, getStartNewGame3.getMsg(null).getStatus());
    }
}
```

Logging:

Ich habe den Server als Solches geloggt um etwaige Fehlermeldungen, aber auch die prinzipielle Ausführung der einzelnen Klassen im Server zu loggen. Zudem habe ich in der Klasse GETStartNewGame folgende 5 Fälle geloggt: Die Initialisierung der Klassen Player, Map und Game, sowie die player_id von Game und Map um sicherzugehen, dass beide gleich sind.

Ich habe mich dagegen entschieden, speziellere Server Parameter zu testen, da dies den Umfang der LOG Files gesprengt hat und der Überblick dadurch verloren geht.

```
2017-12-07 03:03:42 WARN [jdk-http-server-0]
org.hibernate.ejb.HibernatePersistence: HHH015016: Encountered a
deprecated javax.persistence.spi.PersistenceProvider
[org.hibernate.ejb.HibernatePersistence]; use
[org.hibernate.jpa.HibernatePersistenceProvider] instead.

2017-12-07 03:03:42 WARN [jdk-http-server-0]
org.hibernate.ejb.HibernatePersistence: HHH015016: Encountered a
deprecated javax.persistence.spi.PersistenceProvider
[org.hibernate.ejb.HibernatePersistence]; use
[org.hibernate.jpa.HibernatePersistenceProvider] instead.

2017-12-07 03:03:42 INFO [jdk-http-server-0]
org.hibernate.jpa.internal.util.LogHelper: HHH000204: Processing
PersistenceUnitInfo [
    name: gameserver
    ...]

2017-12-07 03:03:42 INFO [jdk-http-server-0] org.hibernate.Version:
HHH000412: Hibernate Core {5.2.11.Final}

2017-12-07 03:03:42 INFO [jdk-http-server-0]
org.hibernate.cfg.Environment: HHH000206: hibernate.properties not
found

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.annotations.common.Version: HCANN000001: Hibernate
Commons Annotations {5.0.1.Final}

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.dialect.Dialect: HHH000400: Using dialect:
org.hibernate.dialect.SQLiteDialect
```

```
2017-12-07 03:03:43 WARN [jdk-http-server-0]
org.hibernate.orm.connections.pooling: HHH10001002: Using Hibernate
built-in connection pool (not for production use!)

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.orm.connections.pooling: HHH10001005: using driver
[org.sqlite.JDBC] at URL [jdbc:sqlite:mydb.db]

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.orm.connections.pooling: HHH10001001: Connection
properties: {}

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.orm.connections.pooling: HHH10001003: Autocommit mode:
false

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionP
roviderImpl: HHH000115: Hibernate connection pool size: 20 (min=1)

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.dialect.Dialect: HHH000400: Using dialect:
org.hibernate.dialect.SQLiteDialect

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.engine.jdbc.env.internal.LobCreatorBuilderImpl: HHH000423:
Disabling contextual LOB creation as JDBC driver reported JDBC version [2]
less than 4

2017-12-07 03:03:43 INFO [jdk-http-server-0]
org.hibernate.validator.internal.util.Version: HV000001: Hibernate Validator
5.3.5.Final

2017-12-07 03:03:44 INFO [jdk-http-server-0]
org.hibernate.hql.internal.QueryTranslatorFactoryInitiator: HHH000397:
Using ASTQueryTranslatorFactory

2017-12-07 03:03:44 INFO [jdk-http-server-0] server.Game:
server.Game@15f4701d

2017-12-07 03:03:44 INFO [jdk-http-server-0] server.Game:
server.Map@617fcce6

2017-12-07 03:03:44 INFO [jdk-http-server-0] server.Game:
server.Player@59ee740

2017-12-07 03:03:44 INFO [jdk-http-server-0] server.Game: 107

2017-12-07 03:03:44 INFO [jdk-http-server-0] server.Game: 107
```

```
2017-12-07 03:03:44 INFO [jdk-http-server-0]
org.hibernate.orm.connections.pooling: HHH10001008: Cleaning up
connection pool [jdbc:sqlite:mydb.db]
```

Überprüfung der BusinessRules:

1. Ein Spiel darf nicht länger als 200 Spielaktionen dauern, ansonsten verliert die KI die gerade an der Reihe ist.

```
Long count = actionMoveDAO.countActions(actionMove);
if(count > 198 && actionMove.getWon() != 1){
    Error error = new Error();
    error.setMessage("Too many actions, both lost");
    actionMovetoDB.setProperties(actionMove);
    actionMoveDAO.createActionMove(actionMovetoDB);
    return
Response.status(Response.Status.EXPECTATION_FAILED).entity(error).build
();
}
```

In POSTActionMove werden mittels der ActionMoveDAO Methode countActions, die eine SQL Query aufruft (bereits beschreiben) welche die Aktionen zählt, die zu einem Spiel gehörigen Aktionen gezählt und wenn der Spieler bei der 200. Aktion nicht zufällig gewonnen hat, das Spiel abgebrochen und beide Spieler verlieren.

2. Eine Spielaktion darf max. 3 Sekunden lang dauern, sonst verliert die KI die gerade an der Reihe ist.

Beispielhaft an Bewegungsaktionen in POSTActionMove:

ActionMove Methode checkSeconds:

```
public boolean checkSeconds(){
    Long diff;
    Timestamp start = new Timestamp(this.getTime_start());
    Timestamp end = new Timestamp(this.getTime_end());
    if(this.getTime_start() != null && this.getTime_end() != null){
        diff = (end.getTime()- start.getTime())/1000;
    }
}
```

```
    if(diff > 3){  
        return false;  
    }  
    return true;  
}  
return false;  
}
```

Wenn zwischen Start und Ende der Aktion mehr als 3 Sekunden liegen, wird false retourniert. Dies geschieht auch wenn keine Einträge vorhanden sind.

POSTActionMove Überprüfung:

```
if(!actionMove.checkSeconds()){  
    Error error = new Error();  
    error.setMessage("Too slow, you lost");  
    game.setloser_id(actionMove.getPlayer_id());  
    gameDAO.createGame(game);  
    return  
Response.status(Response.Status.EXPECTATION_FAILED).entity(error).build  
();  
}
```

Wenn die routinemäßige Überprüfung bei Übermittlung einer Aktion false liefert, wird der Player als loser im Game gespeichert und ein error retourniert. Das Spiel ist vorbei.

3. Die den KI's zugeordneten Schätze werden vom Server auf den von Ihnen generierten Kartenhälften versteckt und sind nicht sichtbar.

Dazu werden die Tiles von der Map aus der DB ausgelesen und in allen Tiles der Map in einer neuen TilesList auf 0 gesetzt und diese dann retourniert.

```
Map map = new Map();
```

```
TileList tileList2 = new TileList();
```

```
try {  
    map = mapDAO.findOtherMapByGame_id(map_id);  
    List<Tile> tileUtilList = map.getTiles();  
    tileList.setTiles(tileUtilList);  
    List<Tile> tileList1 = new ArrayList<>();  
    for(Tile tile : tileList.getTiles()){  
        Tile tile1 = new Tile(tile);  
        tile1.setTreasure(0);  
        tile1.setCastle(0);  
        tileList1.add(tile);  
    }  
    tileList2.setTiles(tileList1);  
} catch (NullPointerException e) { // map not ready or not found  
    e.printStackTrace();  
    return Response.status(Response.Status.NO_CONTENT).build();  
} catch (Exception e) {  
    e.printStackTrace();  
    return Response.status(Response.Status.NO_CONTENT).build();  
}  
  
return  
Response.status(200).type(MediaType.APPLICATION_XML).entity(tileList2).  
build();  
}
```

4. Es dürfen sich keine Inseln bilden.
5. Fixe Kartenausdehnung von 8*8 Feldern (4*8 pro Hälfte)
6. Es darf nur weniger als die Hälfte der Kartengrenze mit Wasser bedeckt sein.
7. Pro Karte muss es 3 Berg, 5 Wiesen und 4 Wasserfelder geben.

All diese Regeln werden unten überprüft in der Methode PostMapTiles. Zur Sicherstellung, dass keine Inseln entstehen, lasse ich Wasser nur in der Ersten Reihe und in der Grenzreihe (lange Seite, also x-reihe) zu. Wenn eine der Regeln verletzt wird, wird ein error mit den jeweiligen werten retourniert. Der Schatz wird zufällig auf dem 2. gefunden Gras Feld platziert.

```
switch(tile.getType()){
```

```
case 1:
```

```
    grass = grass+1;
```

```
    treasure = treasure +1;
```

```
    if(treasure == 2){           //place treasure on 2nd grass field found
```

```
        tile.setTreasure(1);
```

```
        hastreasure = true;
```

```
    }
```

```
    break;
```

```
case 2:
```

```
    mountain = mountain+1;
```

```
    if(tile.getCastle() == 1)
```

```
        castlewater=1;
```

```
    break;
```

```
case 3:
```

```
    water = water+1;
```

```
    if(tile.getY() == 4){
```

```
        borderwater = borderwater+1;
```

```
    }
```

```
    if(tile.getY() == 2 || tile.getY() == 3){ //to avoid islands
```

```
        islandwater = islandwater + 1;
```

```
    }
```

```
        if(tile.getCastle() == 1){
            castlewater = 1;
        }
        break;
    }
}

if(mapsize != 32 || mountain < 3 || grass < 5 || water < 4 || borderwater >
3 || islandwater > 0 || castlewater != 0 || !hastreasure){ //auf 32
zurücksetzen

    Error error = new Error();

    error.setMessage("Violated Map Rules! mapsize: " + mapsize + "
Mountains: " + mountain + " Grass: " + grass + " Water: " + water + "
Borderwater: " + borderwater + "Possible island: " + islandwater + "castle
not on grass" + castlewater);

                                                                    return
Response.status(Response.Status.EXPECTATION_FAILED).entity(error).build
();
}
```

8. Eine KI darf maximal eine Aktion pro Runde setzen.

In der Methode POSTActionMove wird die letzte Aktion gesucht und verglichen ob diese vom selben Player ausgeführt wurde und wenn ja, NO_CONTENT retourniert.

```
otherplayermov= actionMoveDAO.findLastAction();

if(otherplayermove.getPlayer_id().equals(actionMove.getPlayer_id()) ){
//other player's turn

return Response.status(Response.Status.NO_CONTENT).build();
```

9. Wasser darf nicht betreten werden.

In der Methode POSTActionMove wird überprüft ob das Zielfeld ein Wasserfeld ist, wenn ja wird ein Error retourniert und das Spiel ist vorbei.

```
tile = tileDAO.findTileByID(actionMove.getTarget_tile());

if(tile.getType() == 3){
```



```
Error error = new Error();  
error.setMessage("Water, you lost!");  
  
return  
Response.status(Response.Status.EXPECTATION_FAILED).type(MediaType.A  
PLICATION_XML).entity(error).build();  
}
```