

# Software Engineering 1

## Abgabedokument

### Teilaufgabe 1

#### (Anforderungsanalyse und Planungsphase)

Persönliche Daten, bitte vollständig ausfüllen:

<b>Nachname, Vorname:</b>	Adem Mehremic
<b>Matrikelnummer:</b>	01650669
<b>E-Mail-Adresse:</b>	a01650669@unet.univie.ac.at
<b>Datum:</b>	21.10.2020

## Aufgabe 1: Anforderungsanalyse (2 Punkte)

Analysieren der Spielidee und des Netzwerkprotokolls um 8 Anforderungen (bestehend zumindest aus 3 funktionalen, 3 nichtfunktionalen und einer Designbedingung) nach den folgenden Kriterien zu dokumentieren. Achten Sie darauf den im Skriptum und der Vorlesung behandelten Qualitätsaspekten Genüge zu tun.

### Typ der Anforderung: funktional

#### Anforderung 1

- **Beschreibung:** Karten Generierung - Jeder Client generiert jeweils eine Kartenhälfte, die er dann dem Server schickt
- **Bezugsquelle:** Netzwerkprotokoll - „Im Anschluss an die Registrierung eines Spielers muss der Client eine Kartenhälfte generieren und dem Server übertragen.“

#### Anforderung 2

- **Beschreibung:** Alle Felder erreichbar - Die Karte darf kein Feld (Wiese oder Berg) haben das man nicht erreichen kann
- **Bezugsquelle:** Spielidee – „Weiters dürfen **keine Inseln** generiert werden, daher dürfen Berge oder Wiesen niemals vollständig von Wasser oder Kartengrenzen (oder einer Kombination aus beidem) umschlossen werden. Abschließend gilt, dass **jede Kartengrenze** einer zu generierenden Kartenhälfte nur zu **weniger als der Hälfte aus Wasserfeldern** (maximal 3 Wasserfelder an den langen und 1 Wasserfeld an den kurzen Seiten) bestehen darf, um den Wechsel zwischen beiden Spielkartenhälften nicht zu verhindern. Es gilt außerdem, dass zu jedem Feld einer Kartenhälfte (das nicht aus Wasser besteht) von jedem anderen Feld der Kartenhälfte (welches nicht aus Wasser besteht) ein Weg aus einer Reihe von validen Bewegungsbefehlen generiert werden können muss, um dieses Feld zu betreten.“

#### Anforderung 3

- **Beschreibung:** Rundenbasiertes Spiel – Die KIs machen abwechselnd eine Spielaktion bzw. eine KI wartet, während die andere an der Reihe ist
- **Bezugsquelle:** Spielidee - Die Spielaktionen selbst werden rundenbasiert durchgeführt. Daher kann jede KI immer nur eine Aktion setzen (z.B. einen Bewegungsbefehl oder die Übertragung einer Kartenhälfte) und muss danach warten, bis die andere KI ihre Aktion gesetzt hat.

#### Anforderung 4

- **Beschreibung:** Visualisierung des Spiels – Das Spiel wird in der CLI visualisiert nach jedem Zug einer KI visuell dargestellt (Position der Spieler, die Zugehörigkeit Burg der KI, Resultat hat die KI gewonnen oder verloren, etc.)
- **Bezugsquelle:** Spielidee - Während des Spiels müssen die Karte und deren bekannte Eigenschaften auf den Rechnern (den Clients) der beiden menschlichen Spielern visualisiert werden, z.B., gegnerische Spielfiguren (beide Spielfiguren sind immer für beide KIs sichtbar), Burgen, Felder (und deren jeweiliges Terrain) etc. Dies dient dazu, dass menschliche Spieler mit dem Verhalten ihrer künstlichen Intelligenzen mitfeiern können.

### Anforderung 5

- **Beschreibung:** Startpunkt der Spielfigur – Die Spielfiguren des jeweiligen Spielers starten von Ihren Burgen, auf ihrer Seite der Karte.
- **Bezugsquelle:** Spielidee – „Die Spielfigur der KI startet anschließend jeweils an einer von ihr selbst definierten Position (ihrer Burg) auf dem von ihr erstellten Teil der Karte.“

### Anforderung 6

- **Beschreibung:** Bewegungskosten – Jeder Feldtyp hat eine gewisse Anzahl von Spielaktionen die die KI machen muss um die Spielfigur auf das Feld oder von dem Feld runter zu bewegen. Auf ein Grass Feld braucht die KI eine Spielaktion und eine runter, auf/von einem Berg zwei solcher Aktionen, z.B. von Grass zu Grass sind zwei solcher Aktionen nötig, Grass zu Berg drei und von Berg zu Berg vier.
- **Bezugsquelle:** Spielidee – „Wasser und Wiesen können jeweils mit einer Bewegungsaktion betreten und im Fall von Wiesen auch wieder verlassen werden.“ „Im Gegensatz zu Wiesen benötigen Berge zwei Bewegungsaktionen, um diese (das Bergfeld) zu betreten und zwei zusätzliche, um den Bergwieder zu verlassen.“

## Typ der Anforderung: nicht funktional

### Anforderung 1

- **Beschreibung:** Zuschauer auf Trab halten – Das Spiel endet nach 200 Spielaktionen und jede Spielaktion muss in unter 3 Sekunden ausgeführt werden
- **Bezugsquelle:** Spielidee - Um die Spiele für die Zuschauer spannend zu gestalten, wurde festgelegt, dass ein Spiel insgesamt nicht länger als **200 Spielaktionen** dauern darf und eine KI für **jede dieser rundenbasierten Spielaktion nicht mehr als 3 Sekunden** Bedenkzeit zur Berechnung erhält

### Anforderung 2

- **Beschreibung:** Statusabfrage – Die KI fragt den Server, nach mindestens 0,4 Sekunden, nach dem Spielstatus ab
- **Bezugsquelle:** Netzwerkprotokoll – „Um zu verhindern, dass der Server überlastet wird sollte zwischen zwei vom gleichen Clientdurchgeführten Abfragen zum Spielstatus **mindestens eine Zeitspanne von 0,4 Sekunden** vergehen.“

### Anforderung 3

- **Beschreibung:** Validierung der Kartenhälfte – Die generierte Kartenhälfte muss bestimmte Regeln erfüllen
- **Bezugsquelle:** Spielidee – „Die erstellten Karten sollten dabei zufällig generiert und nicht statisch vorgegeben werden. Außerdem muss jeder Terraintyp vorkommen, das heißt jede Kartenhälfte muss mindestens 3 Bergfelder, 15 Wiesenfelder und 4 Wasserfelder beinhalten. Weiters dürfen keine Inseln generiert werden, daher dürfen Berge oder Wiesen niemals vollständig von Wasser oder Kartengrenzen (oder einer Kombination aus beidem) umschlossen werden. Abschließend gilt, dass jede Kartengrenze einer zu generierenden Kartenhälfte nur zu weniger als der Hälfte aus Wasserfeldern (maximal 3 Wasserfelder an den langen und 1 Wasserfeld an den kurzen Seiten) bestehen darf, um den Wechsel zwischen beiden Spielkartenhälften nicht zu verhindern. Es gilt außerdem, dass zu jedem Feld einer Kartenhälfte (das nicht aus Wasser besteht) von jedem anderen Feld der Kartenhälfte (welches nicht aus Wasser besteht) ein Weg aus einer Reihe von validen Bewegungsbefehlen generiert werden können muss, um dieses Feld zu betreten.“

## Typ der Anforderung: Designbedingung

### Anforderung 1

**Beschreibung:** Nachrichtenformat - Die Nachrichten zwischen Client und Server sind in XML-Format.

- **Bezugsquelle:** Netzwerkprotokoll – „Die ausgetauschten Daten bzw. Nachrichten werden im XML-Format definiert bzw. erwartet.“

### Anforderung 2

**Beschreibung:** Nachrichtenaustausch - Der Austausch von Nachrichten zwischen Client und Server geht über eine Restschnittstelle, mittels HTTP und GET/POST Methoden.

**Bezugsquelle:** Netzwerkprotokoll – „Die technologische Basis des Nachrichtenaustauschs stellt eine Restschnittstelle dar, daher es wird das HTTP Protokoll verwendet sowie die zugehörigen Operationen GET und POST.“

## Aufgabe 2: Anforderungsdokumentation (2 Punkte)

Dokumentation von *einer* in Aufgabe 1 identifizierten Anforderungen nach dem vorgegebenen Schema. Ziehen Sie eine Anforderung heran, für die alle Bestandteile der Vorlage mit relevantem Inhalt befüllt werden können. Wir empfehlen hierzu eine **funktionale** Anforderung auszuwählen.

### Dokumentation Anforderung

**Name:** Die KIs erstellen die Mappe, je eine Hälfte.

**Beschreibung und Priorität:** Jede KI soll eine Hälfte der Karte erstellen. Die erstellte Karte muss einige Bedingungen erfüllen. Falls die Karte alle Bedingungen erfüllt wird sie zum Server gesendet.

Priorität der Anforderung: Hoch.

**Relevante Anforderungen:** Funktionelle: 1, 2, 5 Nichtfunktionelle: 4

### Relevante Business Rules:

- Die von den KIs bespielte Karte wird bei Beginn des Spiels von beiden beteiligten künstlichen Intelligenzen kooperativ erstellt.
- Jede der beiden KIs erstellt zufällig eine Hälfte der finalen Spielkarte (mit je 4 x 8 Feldern für eine endgültige Kartengröße von 8 x 8 oder 4 x 16).
- Die Karten sind in Felder aufgeteilt, zwischen denen sich die Spielfiguren bewegen können.
- Jedes Feld repräsentiert genau eine von drei möglichen Terrainarten, nämlich Wasser, Wiese und Berg.
- Ein, und nur ein Feld hat eine Burg und die kann nur auf einem Wiesenfeld sein
- Außerdem muss jeder Terraintyp vorkommen, das heißt jede Karten-hälfte muss mindestens 3 Bergfelder, 15 Wiesenfelder und 4 Wasserfelder beinhalten.
- Weiters dürfen keine Inseln generiert werden, daher dürfen Berge oder Wiesen niemals vollständig von Wasser oder Kartengrenzen (oder einer Kombination aus beidem) umschlossen werden
- Jede Kartengrenze einer zu generierenden Kartenhälfte kann maximal 3 Wasserfeldern den langen und 1 Wasserfeld an den kurzen Seiten.

**Impuls/Ergebnis - Typisches Szenario:** Die generierte Karte der KI, hat alle Business Rules erfüllt, so wurde die ganze Karte von Server generiert und das Spiel kann beginnen.

### Vorbedingungen:

- Eis Spieler hat ein Spiel auf dem Server erstellt
- Beide KIs haben sich auf dem Server erfolgreich registriert

**Hauptsächlicher Ablauf:**

- Die KI generiert eine Kartenhälfte
- Karte wird überprüft nach dem Business Rules des Klienten
- Die Karte hat alle Business Rules erfüllt
- Der Client sendet die Karte zum Server
- Server antwortet das die Kartenhälfte valide ist

**Nachbedingungen:**

- Die Kartenhälften wurden von Server verbunden, und so die ganze Karte generiert.
- Die KI fragt den Server nach Ihren Spielstatus ab.

**Impuls/Ergebnis - Alternativszenario:** Es wird keine Valide Karte erstellt innerhalb des Zeitraums von 3 Sekunden erstellt

**Vorbedingungen:**

- Eis Spieler hat ein Spiel auf dem Server erstellt
- Beide KIs haben sich auf dem Server erfolgreich registriert

**Hauptsächlicher Ablauf:**

- Die KI generiert eine Kartenhälfte
- Karte wird überprüft nach dem Business Rules
- Die Karte hat nicht alle Business Rules des Klienten erfüllt
- Neue Karte wird generiert
- Neue Karte wird auf Business Rules des Klienten überprüft

**Nachbedingungen - Zwei Fälle:**

- Keine Valide Karte wird innerhalb 3 Sekunden erstellt und die KI verliert das Spiel

**Impuls/Ergebnis - Fehlerfall:** Die Erstellte Karte erfüllt nicht alle Business Rules auf dem Server

**Vorbedingungen:**

- Eis Spieler hat ein Spiel auf dem Server erstellt
- Beide KIs haben sich auf dem Server erfolgreich registriert

**Hauptsächlicher Ablauf:**

- Die KI generiert eine Kartenhälfte
- Karte wird überprüft nach dem Business Rules des Klienten
- Die Karte hat alle Business Rules erfüllt
- Der Client sendet die Karte zum Server
- Server antwortet das die Kartenhälfte nicht valide ist

**Nachbedingungen:**

- Die KI verliert das Spiel

• **Benutzergeschichten:** Als Anwender möchte ich das meine KI die Karte generiert, um zu erreichen das ich mich nicht jedes mahl mit der Generierung beschäftige und die Karte korrekt ist.

• **Benutzerschnittstelle:**

In der CLI wird ganze Ablauf wie folgt ausgegeben:

<Nachname, Vorname>

<Matrikelnummer>

1. November 2020

Seite 7 von 11

- Karte wird generiert
- Karte würde generiert
- Karte wird überprüft
- Karte wird gesendet
- Karte gesendet

Server Antwort:

- Kartenhälfte ist valide! THE GAME CAN START!
- Kartenhälfte ist NICHT valide! YOU LOST THE GAME

- **Externe Schnittstellen:** [Beschreiben Sie hier kurz die von dieser Applikation zur Realisierung verwendeten Schnittstellen zu externen Systemen. 1-2 Sätze je Schnittstelle mit Informationen zu den ausgetauschten Informationen, Netzwerkverbindungen, Protokollen, Synchronisationsmechanismen, Beschränkungen der Datenrate etc. die EntwicklerInnen erkennbar machen sollen wie mit den Schnittstellen interagiert werden muss.]

ist die REST Schnittstelle zwischen Client und Server. Der Client sendet einen HTTP POST Request mit einer XML-Nachricht im Body an dem Endpunkt:

<http://<domain>:<port>/games/<SpielID>/halfmaps>

Im body werden der SpielerID und die HA und Server antwortet generisch mit einem `responseEnvelope`.



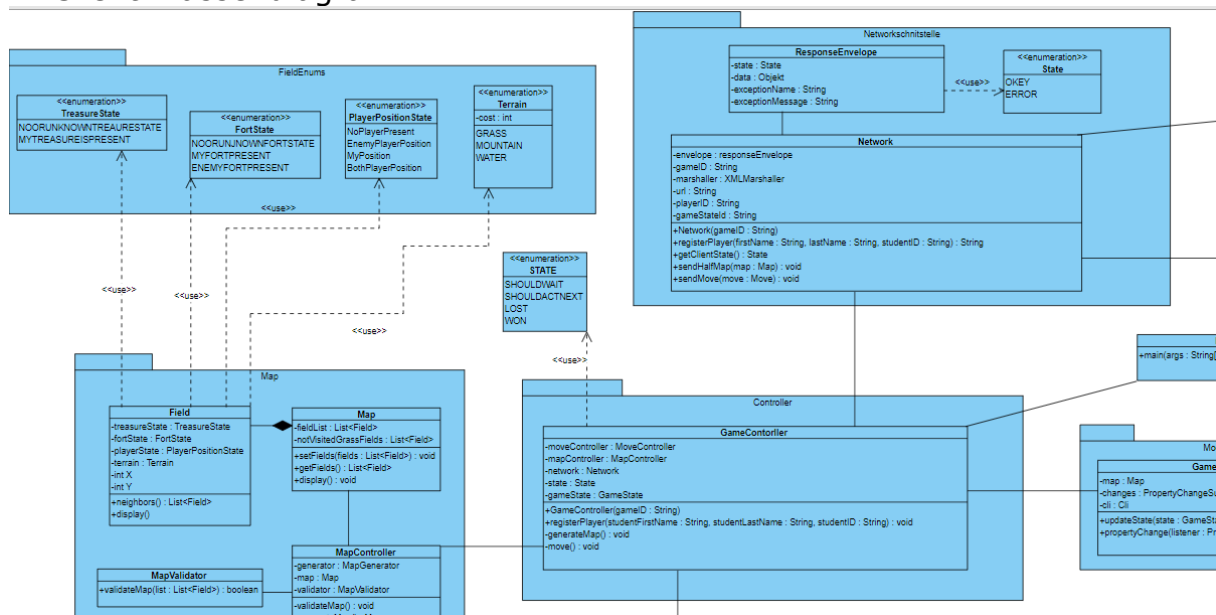
## Aufgabe 3: Architektur entwerfen, modellieren und validieren (10 Punkte)

Modellieren Sie händisch alle notwendigen Packages, Klassen und deren Methoden (samt Beziehungen) als zwei UML Klassendiagramme. Achten Sie darauf, dass die Modelle sinnvoll benannte Packages, Klassen, Methoden und Felder beinhalten und die Vorgaben der Spielidee bzw. des Netzwerkprotokolls vollständig in sinnvoller Granularität abgedeckt werden.

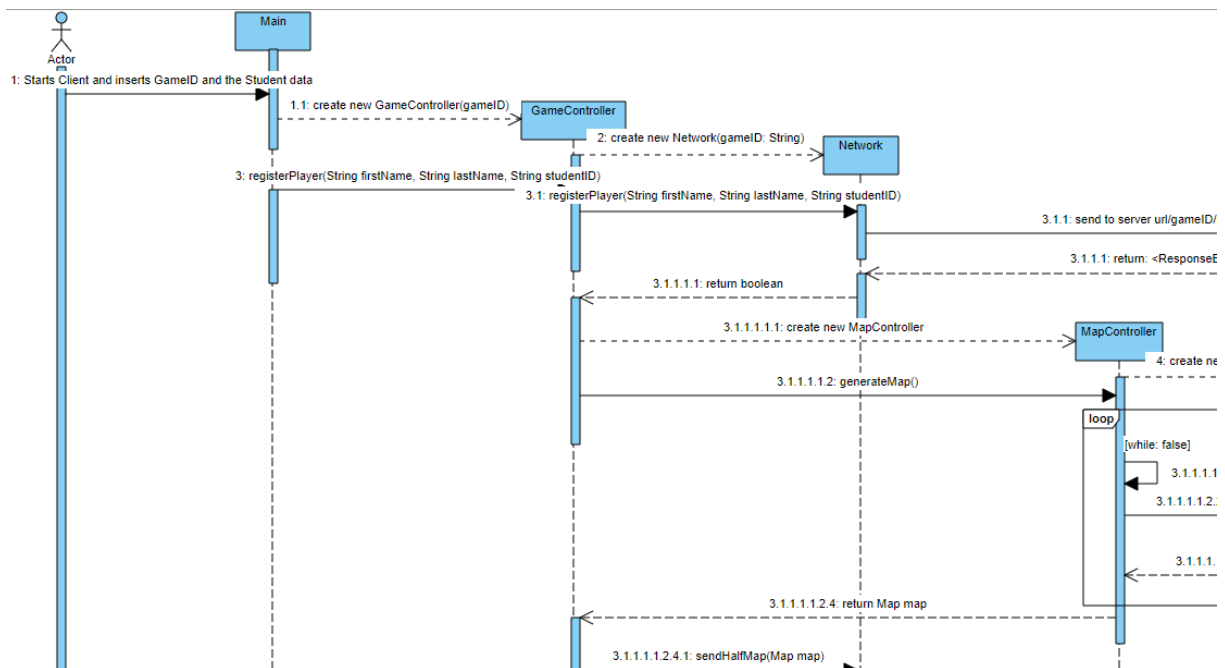
**Basierend auf dem Klassendiagramm:** Erstellen Sie zwei Sequenzdiagramme zu den beiden in der Übungsangabe vorgegebenen Aspekten. Alle erstellten Diagramme sollten semantisch und syntaktisch korrekt sowie untereinander konsistent sein.

Die ganzen Diagramme kann man als separate SVG Dateien finden!

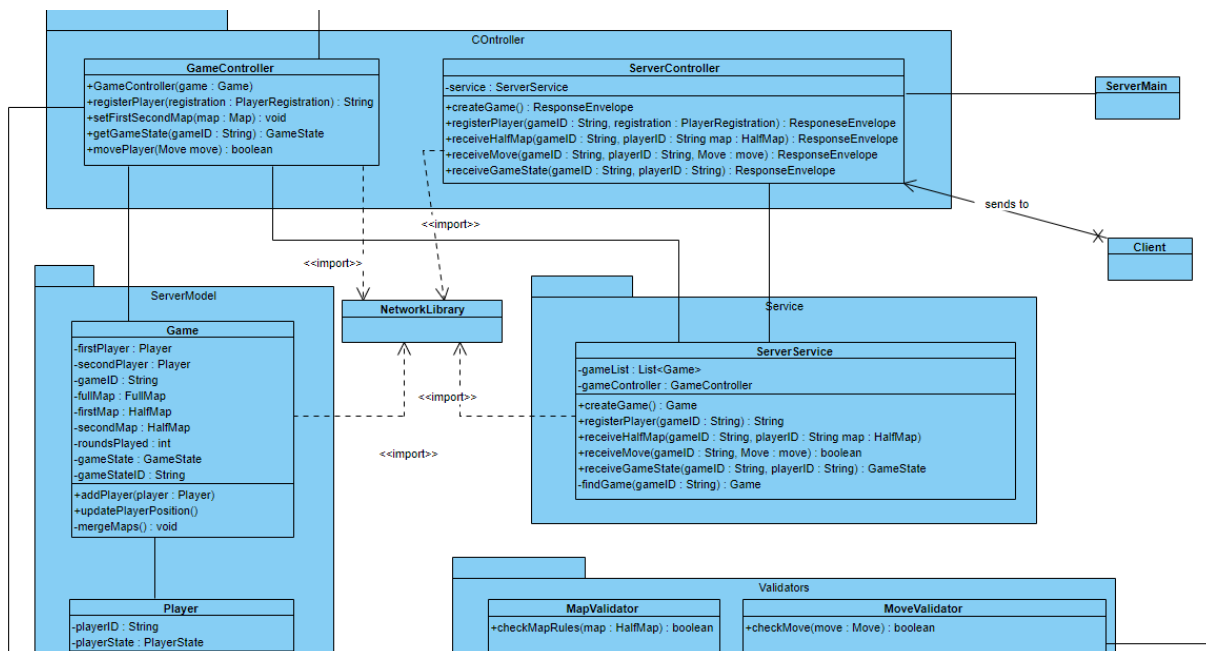
### 1. Client Klassendiagramm



## 2. Client Sequenzdiagramm



## 3. Server Klassen Diagramm



#### 4. Server Sequenzdiagramm

