

Software Engineering 1

Abgabedokument

Teilaufgabe 1

(Anforderungsanalyse und Planungsphase)

Persönliche Daten, bitte vollständig ausfüllen:

Nachname, Vorname:	Bryan Yi Jue, Tan
Matrikelnummer:	11930138
E-Mail-Adresse:	a11930138@unet.univie.ac.at
Datum:	04.04.2024

Aufgabe 1: Anforderungsanalyse

1. Funktionale Anforderung:

- **Beschreibung:** Clients/KIs müssen das Spiel autonom ohne menschliches Eingreifen abschließen, damit das Spiel reibungslos ablaufen kann.
- **Bezugsquelle:** Spielidee, Zitat *“Die grundlegende Spielidee ist, dass zwei Clients bzw. deren KIs, ohne menschliches Eingreifen, auf der gleichen Spielkarte eine vergleichbare Aufgabe erfüllen müssen.”* (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 1).

2. Funktionale Anforderung:

- **Beschreibung:** Die Karte wird vom Server aus zufälligen Kartenhälften der Clients auf beiden Seiten zusammengestellt, damit die KIs das Spiel auf der erstellten Karte spielen kann.
- **Bezugsquelle:** Spielidee, Zitat *“...sondern entsteht durch die Kombination dieser zufällig erzeugten Kartenhälften durch den Server.”* (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 2).

3. Funktionale Anforderung:

- **Beschreibung:** Eine Spielfigur kann sich nur horizontal und vertikal zu direkt benachbarten Feldern bewegen, das Überspringen von Feldern ist nicht möglich.
- **Bezugsquelle:** Spielidee, Zitat *“Eine Spielfigur kann sich nur horizontal und vertikal zu direkt benachbarten Feldern bewegen, das Überspringen von Feldern ist nicht möglich.”* (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 14).

4. Funktionale Anforderung:

- **Beschreibung:** Der Client soll in der Lage sein, den Spielstatus abzufragen.
- **Bezugsquelle:** Spielidee, Zitat *“Der Server unterstützt dies, können Clients doch abfragen ob diese gerade an der Reihe sind.”* (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 9).

5. Nicht funktionale Anforderung:

- **Beschreibung:** Die vom Server gemeldeten Gegnerpositionen müssen bis zum Abschluss der ersten ~8 eigenen Clientaktionen zufällig ausgewählt werden.
- **Bezugsquelle:** Spielidee, Zitat *“Bis zum Abschluss der ersten ~8 eigenen Clientaktionen hat die vom Server gemeldete Position des Gegners zufällig zu sein.”* (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 6).

6. Nicht funktionale Anforderung:

- **Beschreibung:** Die KI hat pro Spielzug nur maximal 5 Sekunden Zeit, um über ihren nächsten Spielaktion nachzudenken.
- **Bezugsquelle:** Spielidee, Zitat "Für jede dieser rundenbasierten Spielaktion hat die KI maximal 5 Sekunden Bedenkzeit." (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 10).

7. Nicht funktionale Anforderung:

- **Beschreibung:** Die Kartenhälfte muss eine Größe von 5 x 10 Feldern haben.
- **Bezugsquelle:** Spielidee, Zitat "*Hierzu erstellt jede der beiden KIs zufällig eine Hälfte der finalen Spielkarte (mit je 5 x 10 Feldern).*" (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 11).

8. Nicht funktionale Anforderung:

- **Beschreibung:** Jede Kartenhälfte muss mindestens 10 % Bergfelder beinhalten.
- **Bezugsquelle:** Spielidee, Zitat "*Jede Kartenhälfte muss mindestens 10% Bergfelder, 48% Wiesenfelder, 14% Wasserfelder und eine Burg beinhalten.*" (siehe Abschnitt 1. Beschreibung der Spielidee, Absatz 15).

9. Designbedingung:

- **Beschreibung:** Die ausgetauschten Daten oder Nachrichten müssen im XML-Format definiert sein.
- **Bezugsquelle:** Netzwerkprotokoll, Zitat "*Die ausgetauschten Daten bzw. Nachrichten werden im XML Format definiert bzw. erwartet.*" (siehe Abschnitt 1. Einleitung zum Netzwerkprotokoll, Absatz 2).

Aufgabe 2: Anforderungsdokumentation

Dokumentation Anforderung

- **Name:** Spielstatus abfragen
- **Beschreibung und Priorität:** Der Client sollte in der Lage sein, den Spielstatus abzufragen. Dies stellt sicher, dass der Client seinen Spielstatus abfragen kann, um festzustellen, ob er als nächstes handeln darf.

Priorität: Hoch

- **Relevante Anforderungen:**

- o 1. Funktionale Anforderung: Clients/KIs müssen das Spiel autonom ohne menschliches Eingreifen abschließen. Clients müssen seinen Spielstatus autonom beim Server erfragen.
- o 2. funktionale Anforderung: Die Karte wird vom Server aus zufälligen Kartenhälften der Clients auf beiden Seiten zusammengestellt, damit der Client das Spiel spielen zu können.
- o 6. nicht funktionale Anforderung: Die KI hat pro Spielzug nur maximal 5 Sekunden Zeit, um über ihren nächsten Spielaktion nachzudenken, auch wenn sie den Spielstatus abfragen möchte.
- o 9. Designbedingung: Die ausgetauschten Daten oder Nachrichten müssen im XML-Format definiert sein.

- **Relevante Business Rules:**

- o Die Spielaktionen selbst werden rundenbasiert durchgeführt.
- o Die Größe der Spielkarte ist entweder 10x10 oder 5x20.
- o Jede KI kann immer nur eine Aktion pro Spielzug setzen.
- o Für jede dieser rundenbasierten Spielaktion hat die KI maximal 5 Sekunden Bedenkzeit.

1. Impuls/Ergebnis - Typisches Szenario:

Vorbedingungen:

- o Der Client hat das Spiel auf dem Server registriert.
- o Der Client hat Kartenhälfte generiert.

Hauptsächlicher Ablauf:

- o Impuls: Der Client fragt den Spielstatus er.
- o Ergebnis: Der Server übermittelt den Spielstatus.
- o Impuls: Der Client liest den Spielstatus aus.
- o Ergebnis: Der Client muss eine Aktion durchführen.

- o Impuls: Der Client übermittelt Kartenhälfte.
- o Ergebnis: Der Server bestätigt Empfang und meldet Fehler.

Nachbedingungen:

- o Der Client hat erfolgreich nach seinem Spielstatus abgefragt und in seinem Zug die Kartenhälfte an den Server übermittelt.

2. Impuls/Ergebnis - Alternativszenario:

Vorbedingungen:

- o Der Client hat das Spiel auf dem Server registriert.
- o Der Client hat Kartenhälfte generiert.
- o Der Client hat den Schatz gefunden und aufgenommen.
- o Der Client hat die gegnerische Burg gefunden und sich zu dieser bewegt.

Hauptsächlicher Ablauf:

- o Impuls: Der Client fragt den Spielstatus er.
- o Ergebnis: Der Server übermittelt den Spielstatus.
- o Impuls: Der Client liest den Spielstatus aus.
- o Ergebnis: Der Client muss warten.
- o Impuls: Der Client fragt erneut den Spielstatus er.
- o Ergebnis: Der Client hat das Spiel gewonnen.

Nachbedingungen:

- o Der Client hat erfolgreich nach seinem Spielstatus abgefragt, aber in diese Fall muss er keine mehr weitere Aktion durchführen.
- o Nachdem der Client das Spiel gewonnen/verloren hat, endet das Spiel.

3. Impuls/Ergebnis - Fehlerfall:

Vorbedingungen:

- o Der Client hat das Spiel auf dem Server registriert.
- o Der Client hat Kartenhälfte generiert.

Hauptsächlicher Ablauf:

- o Impuls: Der Client fragt den Spielstatus er.
- o Ergebnis: Der Server bereitet die Übermittlung des Spielstatus vor.
- o Impuls: Der Client fragt innerhalb von 0,4 Sekunden erneut den Spielstatus er.

- o Ergebnis: Der Server bearbeitet den Anfrage mit Verzögerung.
- o Impuls: Der Client fragt innerhalb von 0,4 Sekunden erneut den Spielstatus er.
- o Ergebnis: Der Server überlastet.

Nachbedingungen:

- o Der Client hat seinem Spielstatus nicht bekommen und der Server überlastet.

• **Benutzergeschichten:**

- o Als Anwender möchte ich eine intelligenteste Client entwerfen, um das Spiel erfolgreich zu gewinnen.
- o Als Client möchte ich die schnellste Route berechnen, um das Spiel schnell zu gewinnen.
- o Als Server möchte ich alle empfangenen Daten genau überprüfen, um ein faires Spiel zu gewährleisten.

• **Benutzerschnittstelle:**

Entsprechend den Anforderungen sollte die Position der KI auf der CLI angezeigt werden, und nach jedem Zug sollte die neue Position der KI aktualisiert werden. Hier ist eine mögliche CLI-Oberfläche:

m = Berg, f = Wiesenfelder, w = Wasser, o = Burg, 1 = Erste AI, 2 = Zweite AI

m	f	w	o	m	f	w	w	m	w
w	m	w	f	w	m	f	w	w	w
w	w	m	w	w	m	w	w	w	w
w	w	w	w	m	w	w	w	w	2
1	m	f	w	m	f	m	f	w	w

AI 1 vor dem Bewegen

m	f	w	o	m	f	w	w	m	w
w	m	w	f	w	m	f	w	w	w
w	w	m	w	w	m	w	w	w	w
1	w	w	w	m	w	w	w	w	2
w	m	f	w	m	f	m	f	w	w

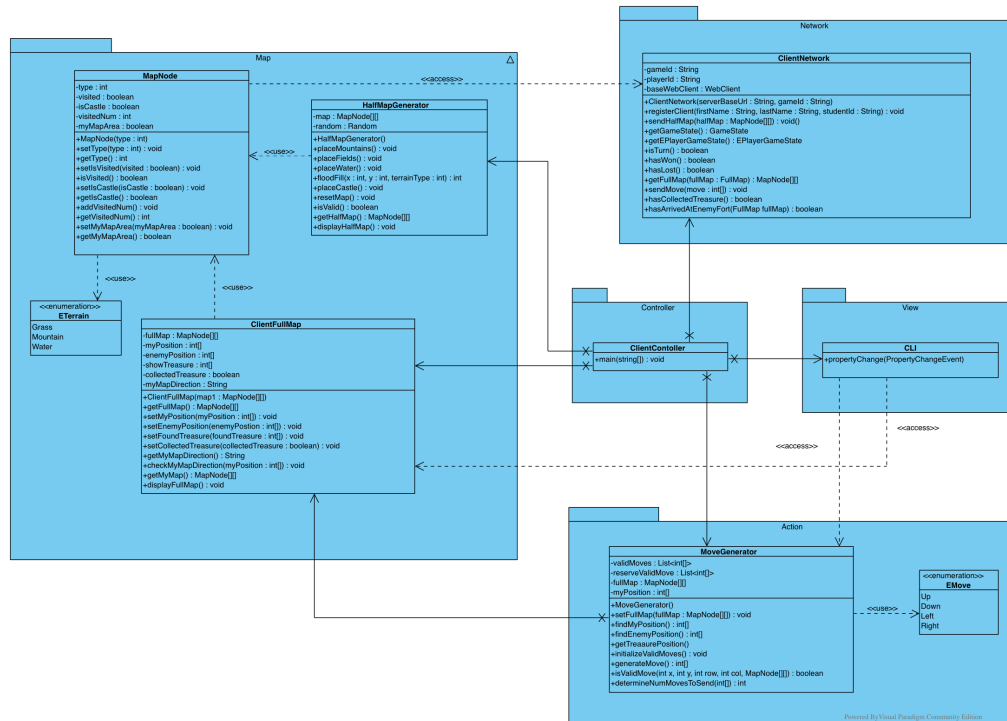
AI 1 nach dem Bewegen (nach oben)

• **Externe Schnittstellen:**

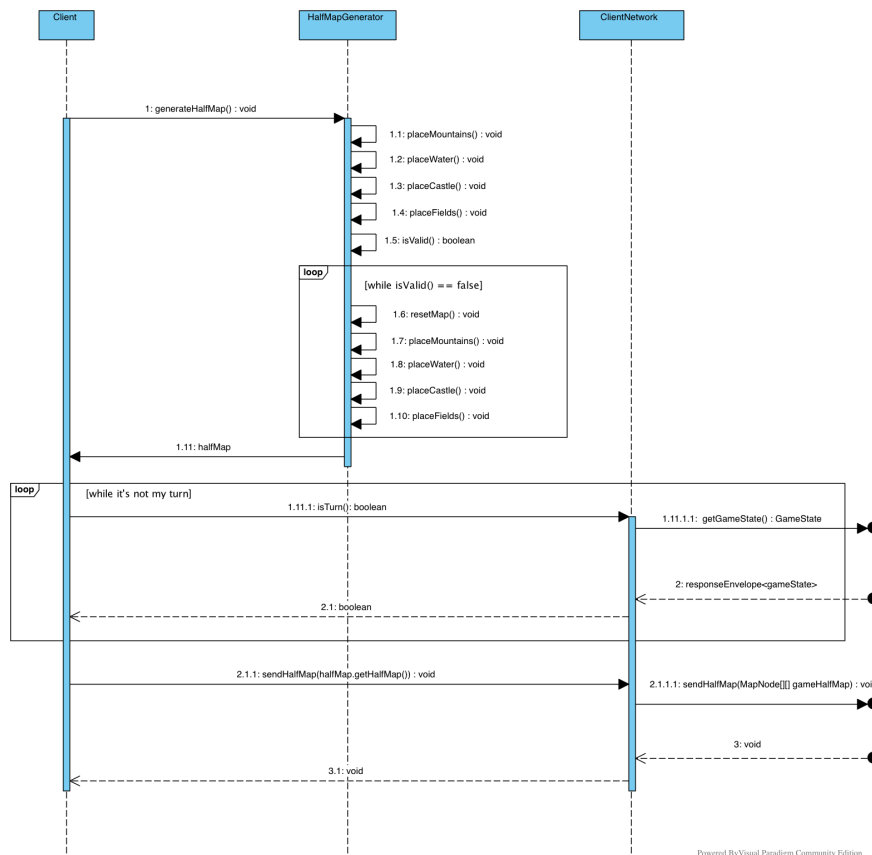
Schnittstelle Server: Die Nutzung der verbleibenden Schnittstellen ist notwendig, weil der Client muss Informationen an den Server schicken, und Informationen vom Server empfangen können. Die Daten wird im XML Format auszutauschen.

Aufgabe 3: Architektur entwerfen, modellieren und validieren

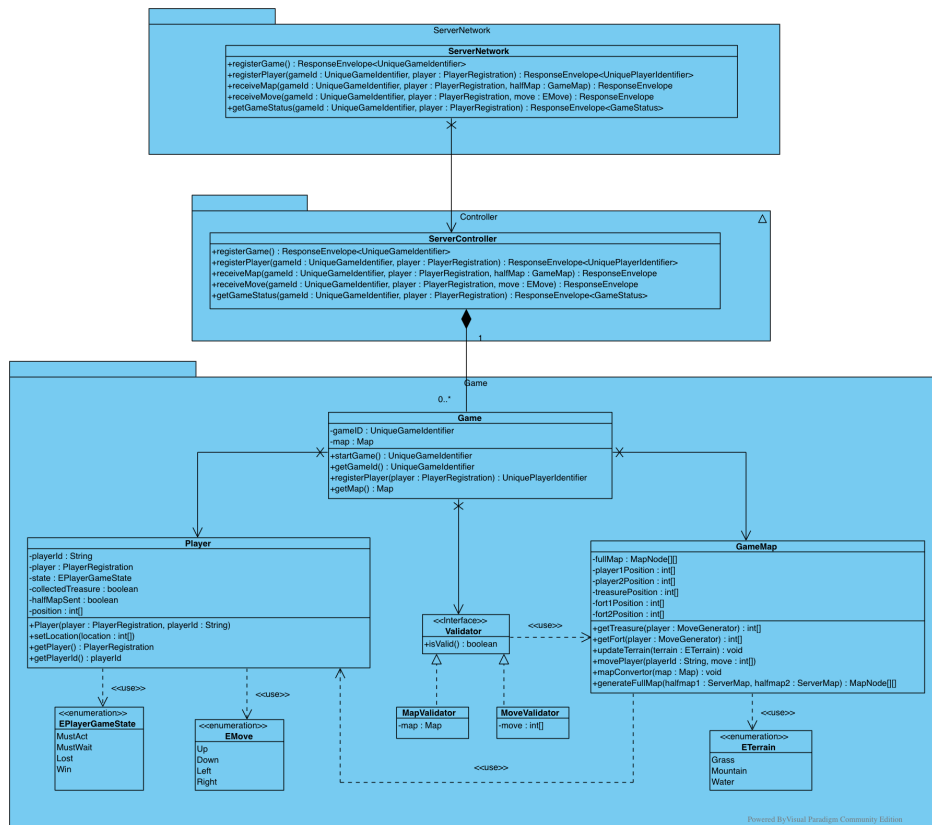
Client-Klassendiagramm:



Client-Sequenzdiagramm:



Server-Klassendiagramm:



Server-Sequenzdiagramm:

