

Software Engineering 1

Abgabedokument

Teilaufgabe 1

(Anforderungsanalyse und Planungsphase)

Persönliche Daten, bitte vollständig ausfüllen:

Nachname, Vorname:	Jarda, Miruna-Diana
Matrikelnummer:	11921801
E-Mail-Adresse:	a11921801@unet.univie.ac.at
Datum:	12.03.2022

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:** Generating the map – The AI has to generate one half of the game's map at the beginning of the game.
- **Bezugsquelle:** Spielidee, *Die Karte, auf welcher gespielt wird, ist hierbei nicht fest vorgegeben, sondern wird von beiden Kis selbstständig und automatisch beim Start des Spiels erstellt (jeweils eine Hälfte der Karte).*

Anforderung 2

- **Beschreibung:** Player movement – The player has to be able to move in horizontal and vertical direction to the fields around him.
- **Bezugsquelle:** Spielidee, *Eine Spielfigur kann sich nur horizontal und vertikal zu direkt benachbarten Feldern bewegen, das Überspringen von Feldern ist daher nicht möglich.*

Anforderung 3

- **Beschreibung:** Client registration – The client can register for a new game using the unique GameID generated by one of the two clients on the server.
- **Bezugsquelle:** Netzwerkprotokoll, *Sobald ein Spiel erstellt wurde und beiden Clients die eindeutige SpielID (siehe oben) bekannt ist können sich die Clients für das neue Spiel registrieren. Wichtig dabei ist, dass kein Mechanismus vorgesehen ist um beiden Clients direkt vom Server ausgehend die gleiche SpielID mitzuteilen. Daher es ist vorgesehen, dass die SpielID einmalig durch einen Menschen erzeugt wird (z.B. indem über einem Webbrowser auf den Endpoint zu Erstellung eines neuen Spiels zugegriffen wird) und dann beispielsweise die Clients mit der SpielID als Startparameter gestartet werden. Hierdurch können beide Clients ungefähr zur gleichen Zeit beginnen mit dem Server zu interagieren.*

Typ der Anforderung: nicht funktional

Anforderung 4

- **Beschreibung:** Maximum number of actions – The AI has maximum 200 game actions to complete the game.
- **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 (relevant sind hierbei Spielerbewegung und Kartengenerierung).*

Anforderung 5

- **Beschreibung:** Maximum time for a round – The AI has maximum 3 seconds to calculate a new action in each round.
- **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 (relevant sind hierbei Spielerbewegung und Kartengenerierung).*

Anforderung 6

- **Beschreibung:** Waiting time between requests regarding the status – The client should wait at least 0.4 seconds between two requests regarding the status of the game.
- **Bezugsquelle:** Netzwerkprotokoll, *Um zu verhindern, dass der Server überlastet wird sollte zwischen zwei vom gleichen Client durchgeführten Abfragen zum Spielstatus mindestens eine Zeitspanne von 0,4 Sekunden vergehen.*

Typ der Anforderung: Designbedingung

Anforderung 7

- **Beschreibung:** Supporting HTTP – The game has to support the HTTP GET and POST methods.
- **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.*

Anforderung 8

- **Beschreibung:** XML format – The game has to be able to exchange and use XML format data and messages.
- **Bezugsquelle:** Netzwerkprotokoll, *Die ausgetauschten Daten bzw. Nachrichten werden im XML Format definiert bzw. erwartet.*

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:** Client registration

• **Beschreibung und Priorität:** At the beginning of each game, each one of the two clients needs to register themselves using a unique GameID generated during the game creation. During the registration, each client gets an unique PlayerID assigned to it that is then used throughout the entire game.

Priority: High

• **Relevante Anforderungen:**

- *Anforderung 1 - Generating the map* – The client should only send its generated half map after both players are registered.
- *Anforderung 2 - Player movement:* In order for the player to send moves to the server, it needs to be identified by the unique PlayerID assigned to it during registration.
- *Anforderung 7 - Supporting HTTP:* During the registration process, HTTP is needed to send a POST request to the server.
- *Anforderung 8 - XML format:* During the registration process, the XML format is used for the message in the body of the POST request sent to the server.

• **Relevante Business Rules:**

- A player is not allowed to send its map unless both players are registered.

• **Impuls/Ergebnis - Typisches Szenario:**

Vorbedingungen:

- A game has been generated.
- The GameID is known to both clients.

Hauptsächlicher Ablauf:

- Impulse: Client1 registers itself with the details of the student and the GameID.
- Result: Client1 is registered and it receives the PlayerID unique to this client.
- Impulse: Client2 registers itself with the details of the student and the GameID
- Result: Client2 is registered and it receives the PlayerID unique to this client.
- Impulse: Client1 checks game status to see if Client2 is registered.
- Result: Client1 is informed that Client2 is registered.
- Impulse: Client2 checks game status to see if Client1 is registered.
- Result: Client2 is informed that Client1 is registered.

Nachbedingungen:

- Two clients are registered to the game.

- o It is one of the clients' turn to send its half map.

- **Impuls/Ergebnis - Alternativszenario:**

Vorbedingungen:

- o A game has been generated.
- o The GameID is known to both clients.

Hauptsächlicher Ablauf:

- o Impulse: Client1 registers itself with the details of the student and the GameID.
- o Result: Client1 is registered and it receives the PlayerID unique to this client.
- o Impulse: Client1 checks the game status to see if Client2 is registered.
- o Result: Client1 is informed that Client2 is not registered.
- o Impulse: Client1 checks the game status again to see if Client2 is now registered.
- o Result: Client1 is informed that Client2 is not registered.
- o Impulse: Client2 registers itself with the details of the student and the GameID.
- o Result: Client2 is registered and it receives the PlayerID unique to this client.
- o Impulse: Client1 checks the game status again to see if Client2 has registered yet.
- o Result: Client2 has finally registered and Client1 is informed about it.
- o Impulse: Client2 checks game status to see if Client1 is registered.
- o Result: Client2 is informed that Client1 is registered.

Nachbedingungen:

- o Two clients are registered to the game.
- o It is one of the clients' turn to send its half map.

- **Impuls/Ergebnis - Fehlerfall:** [Erwarteten Fehlerfall beschreiben und wie dieser gehandhabt wird. Dies ist besonders relevant, da ansonsten nur Gutfälle berücksichtigt werden was im Produktiveinsatz schnell nicht mehr ausreichend ist.]

Vorbedingungen:

- o A game has been generated.
- o The GameID is known to both clients.

Hauptsächlicher Ablauf:

- o Impulse: Client1 registers itself with the details of the student and the GameID.
- o Result: Client1 is registered and it receives the PlayerID unique to this client.
- o Impulse: Client1 generates and sends its half map.
- o Result: Client1 breaks a business rule.

Nachbedingungen:

- o Game has ended.
- o Client1 gets an error for insufficient amount of clients.
- o Client2 gets an error if they still try to register.

- **Benutzergeschichten:**

- o As human player I want to be informed if the registration process succeeded so that I can remediate the problem in case it goes wrong.
- o As client I want to be able to send a HTTP Post Request to the server containing the XML message with the GameID and student's details in order to register myself.
- o As server I want both clients to be registered before either of them sends their half map so the sufficient number of clients criteria is met when the game starts.
- o As client I want to know what PlayerID I received during the registration process so that I can link it to all the actions I send and be identified based on it by the server.

- **Benutzerschnittstelle:**

The registration confirmation will be shown as a CLI output to each client. As soon as both clients are registered, this will also be shown as a CLI output. In case of an error during the registration process, an explanation will be shown as output.

- **Externe Schnittstellen:**

- o Server interface: The client interacts with the server by sending a HTTP Post Request containing a XML message to the following endpoint: `http(s)://<domain>:<port>/games/<SpielID>/players`, whereas `<SpielID>` is a placeholder for the unique GameID obtained during the game creation. The server sends back a registration confirmation, as well as an unique PlayerID.

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

I uploaded 4 different diagrams in .svg format