

Software Engineering 1

Abgabedokument

Teilaufgabe 1

(Anforderungsanalyse und Planungsphase)

Persönliche Daten, bitte vollständig ausfüllen:

Nachname, Vorname:	Ura, Rigers
Matrikelnummer:	11718263
E-Mail-Adresse:	a11718263@unet.univie.ac.at
Datum:	30.10.2020

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:** Pathfinding algorithm – the client must have an AI that is able to traverse the map to find the treasure and then the opponent's fort as fast as possible without doing any illegal moves
- **Bezugsquelle:** Spielidee: Beschreibung der Spielidee – „Um den Schatz zu finden, bewegen beide KIs ihre Spielfigur über die Karte und decken jeweils mit ihrer Spielfigur den versteckten eigenen Schatz und die gegnerische Burg auf.“

Anforderung 2

- **Beschreibung:** Map-generating algorithm – the client must have an AI that is able to randomly generate the own half-map of the client/player which follows to predefined rules before the match begins.
- **Bezugsquelle:** Spielidee: Beschreibung der 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 3

- **Beschreibung:** Transparency of the map: during its turn, a player knows all the map elements, where it and its opponent is, where its fort is, and if the treasure of the opponent's fort is present in a field (if discovered).
- **Bezugsquelle:** Spielidee: Beschreibung der Spielidee – „Während des Spiels müssen die Karte und deren bekannte Eigenschaften auf den Rechnern (den Clients) der beiden menschlichen Spieler visualisiert werden, z.B., gegnerische Spielfiguren (beide Spielfiguren sind immer für beide KIs sichtbar), Burgen, Felder (und deren jeweiliges Terrain) etc.“

Anforderung 4

- **Beschreibung:** Turn-based game: The game is turn-based; therefore, each player can only perform an action if it is their turn.
- **Bezugsquelle:** Spielidee: Beschreibung der Spielidee – „Die KI, die diese schneller erfüllt, gewinnt das Spiel; welches rundenbasiert abläuft (daher setzt jede KI abwechselnd eine Spielaktion).“

Typ der Anforderung: nicht functional

Anforderung 5

- **Beschreibung:** Maximum number of games: The maximum number of games running in parallel in the server is 999.
- **Bezugsquelle:** Netzwerkprotokoll: Anfrage des Clients – „Um die Ressourcen des Servers zu schonen gilt die Einschränkung, dass maximal 999 Spiele parallel ausgeführt werden dürfen.“

Anforderung 6

- **Beschreibung:** Action-Time limit: The game may not last longer than 200 game actions and an AI may not take longer than 3 seconds to commit an action of any sorts including player movement and map creation with failure to do so resulting in an automatic loss of the player who's turn it currently is.
- **Bezugsquelle:** Spielidee: Beschreibung der Spielidee – „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).“

Sollten diese Bedingungen nicht erfüllt werden, verliert die KI, welche gerade an der Reihe ist, das Spiel automatisch und der zugehörige menschliche Spieler bekommt dies mitgeteilt.

Anforderung 7

- **Beschreibung:** Automatic termination: A game that exceeds 10 minutes of runtime is automatically terminated and removed.
- **Bezugsquelle:** Netzwerkprotokoll: Anfrage des Clients – „Weiters wird jedes Spiel 10 Minuten nach der Spielerzeugung automatisch entfernt/beendet, sodass keine weiteren Nachrichten an dieses gesandt werden können.“

Anforderung 8

- **Beschreibung:** Request time interval: There time difference between two consecutive game status queries by the same client must be at least 0.4 seconds.
- **Bezugsquelle:** Netzwerkprotokoll: Abfrage des Spielstatus – „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 9

- **Beschreibung:** Network Protocol: HTTP protocol is to be used for the communication between the server and the respective clients of the players.
- **Bezugsquelle:** Netzwerkprotokoll: Details zum Netzwerkprotokoll - „Die technologische Basis des Nachrichtenaustauschs stellt eine Restschnittstelle dar, daher es wird das HTTP Protokoll verwendet...“

Anforderung 10

- **Beschreibung:** Message-parsing protocol: to parse messages to and from the server, the XML protocol is used
- **Bezugsquelle:** Netzwerkprotokoll: Erstellung eines neuen Spiels - „Da die Antwort mit Daten im XML Format definiert wird, ...“

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:** Map Generation
- **Beschreibung und Priorität:** After the game is initiated and the players registered, the AI of the client must be able to creates its half-map according to the game rules and send it to the server using XML
 - o Priority - High
- **Relevante Anforderungen:**
 - o It must be the clients turn in order for it to be able to send its half of the map (requirement 4)
 - o As with any action, there is only a 3 seconds time window to create and send the half-map (requirement 6)
- **Relevante Business Rules:**
 - o 4x8 grid of fields
 - o At least 3 mountain fields
 - o At least 15 grass fields
 - o At least 4 water fields
 - o For each boarder, less than half of the fields may be water.
 - o No islands are allowed (grass or mountain fields completely surrounded by water fields)
- **Impuls/Ergebnis - Typisches Szenario:** [Welche Schritte werden durchlaufen, um das der Anforderung zugehörige Verhalten auszulösen bzw. zu nützen und was ist das unmittelbare Ergebnis jedes Schrittes.]
 - **Vorbedingungen:**
 - o The user receives a unique *gameID* from the server
 - o The user runs client-1 and client-2 with the unique gameID.
 - **Hauptsächlicher Ablauf:**
 - o Impuls: Client-1 registers the player by providing the server with the gameID and additional player information through XML.
 - o Ergebnis: The server sends back an XML response envelope with the unique playerId for the player of client-1.
 - o Impuls: Client-2 registers the player by providing the server with the gameID and additional player information through XML.
 - o Ergebnis: The server sends back an XML response envelope with the unique playerId for the player of client-2.
 - o Impul: Client-1 sends to the server game-state XML request
 - o Ergebnis: The server sends back an XML response envelope with player state: *ShouldActNext*
 - o Impul: The AI of client-1 creates its half-map in accordance to the business rules and the client sends it to the server through XML.
 - o Ergebnis: The server sends back an XML response envelope with state: *Okay*.
 - **Nachbedingungen:**
 - o It is the turn of client-2 and it runs through the same steps

- o The match begins

• **Impuls/Ergebnis - Alternativszenario:**

Vorbedingungen:

- o The user receives a unique *gameID* from the server
- o The user runs client-1 and client-2 with the unique *gameID*.

Hauptsächlicher Ablauf:

- o Impuls: Client-1 registers the player by providing the server with the *gameID* and additional player information through XML.
- o Ergebnis: The server sends back an XML response envelope with the unique *playerID* for the player of client-1.
- o Impuls: Client-2 registers the player by providing the server with the *gameID* and additional player information through XML.
- o Ergebnis: The server sends back an XML response envelope with the unique *playerID* for the player of client-2.
- o Impul: Client-1 sends to the server game-state XML request
- o Ergebnis: The server sends back an XML response envelope with player state: *ShouldWait*.
- o Impul: Client-2 sends to the server game-state XML request
- o Ergebnis: The server sends back an XML response envelope with player state: *ShouldActNext*.
- o Impul: The AI of client-2 creates its half-map in accordance to the business rules and the client sends it to the server through XML.
- o Ergebnis: The server sends back an XML response envelope with state: *Okay*.

Nachbedingungen:

- o It is the turn of client-1 and it runs through the same steps
- o The match begins

• **Impuls/Ergebnis - Fehlerfall:**

Vorbedingungen:

- o The user receives a unique *gameID* from the server
- o The user runs client-1 and client-2 with the unique *gameID*.

Hauptsächlicher Ablauf:

- o Impuls: Client-1 registers the player by providing the server with the *gameID* and additional player information through XML.
- o Ergebnis: The server sends back an XML response envelope with the unique *playerID* for the player of client-1.
- o Impuls: Client-2 registers the player by providing the server with the *gameID* and additional player information through XML.
- o Ergebnis: The server sends back an XML response envelope with the unique *playerID* for the player of client-2.
- o Impul: Client-1 sends to the server game-state XML request
- o Ergebnis: The server sends back an XML response envelope with player state: *ShouldActNext*
- o Impul: The AI of client-1 creates its half-map with an island in it and the client sends it to the server through XML.
- o Ergebnis: The server sends back an XML response envelope with state: *Error* and with an error message.

Nachbedingungen:

- o Client-1 loses automatically, and client-2 is declared winner

- **Benutzergeschichten:**

- o As a client, I need to have an AI that is able to create the map according to the rule so that the client doesn't lose automatically.
- o As a server, I need to receive an XML with the half-map according to the schema so that I can process it correctly.

- **Externe Schnittstellen:**

- o HTTP: the client uses HTTP to send the generated map to the server
- o XML: XML is used as the message-parsing protocol to contain the information about the map generated, i.e. the coordinates, terrain type and the presence of the fort

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.

TIPP: Beachten Sie zur Ausarbeitung das auf Moodle zu Verfügung gestellte auszugsweise abgebildete 4+1 Diagrammbeispiel. Dieses sollte als Vorlage dienen, und verdeutlicht welche Erwartungen an das Klassendiagramm beziehungsweise die dazugehörigen Sequenzdiagramme gestellt werden (Darstellung, Inhalte, Detailgrad, etc.) und wie diese zusammenhängen.

[Alle vier Diagramme mit einer kurzen Beschreibung hier einfügen. Sollten die Diagramme zu viel Platz benötigen können Sie diese auch als separate SVG Dateien (empfohlenen) oder PNG Dateien im Dokumentationsordner (siehe GitLab Repo) einfügen. Achten Sie bei der Verwendung von SVG Dateien darauf, dass sich diese fehlerfrei darstellen lassen.]