

# **Software Engineering 1**

## **Abgabedokument**

### **Teilaufgabe 1**

#### **(Anforderungsanalyse und Planungsphase)**

Persönliche Daten, bitte vollständig ausfüllen:

<b>Nachname, Vorname:</b>	Farajzadeh Pegah
<b>Matrikelnummer:</b>	a01549668
<b>E-Mail-Adresse:</b>	<a href="mailto:farajzadep54@unet.univie.ac.at">farajzadep54@unet.univie.ac.at</a>
<b>Datum:</b>	23.10.2023

## Task 1: Requirement Analysis

Analyze the “Game-idea” and the network protocol to document 8 requirements (consisting of at least 3 functional, 3 non-functional and one design condition) according to the following criteria. Make sure you comply with the quality aspects covered in the script and the lecture.

### Type of Requirement: Functional

#### Requirement 1

- **Description:** *Generate Half-Maps*: Each Client must provide half of the Map and send it to the server after starting the game.
- **Source:** [Spielidee page 1 paragraph 2, “**Nach Start des Clients registrieren sich die KIs für das Spiel am Server und erstellen/tauschen danach mit dem Server Kartenhälften aus.**”]

#### Requirement 2

- **Description:** *Generation Of Castle*: For each Half-Map a Castle position is generated by the client on a Grass field.
- **Source:** [Spielidee page 1 paragraph 3, “**„,starten die Spielfiguren jeder KI jeweils auf einer selbst ausgewählten Position (ihrer Burg) auf der von ihr erstellten Kartenhälfte.“**”]

#### Requirement 3

- **Description:** *Generation Of Treasure*: For each Half-Map a Treasure position is chosen by the server on a Grass field.
- **Source:** [Spielidee page 1 paragraph 3, “**Hierzu wird je ein Schatz vom Server auf jeder der beiden Kartenhälften versteckt.**”]

#### Requirement 4

- **Description:** *Half-Map Configuration*: The Map should be divided into individually defined fields each half-map has 5x10 field.

- **Source:** [Spielidee page 1 paragraph 11, “**Die Karten sind in Felder aufgeteilt...**”]

## Requirement 5

- **Description:** *Placement Of Half-Maps:* The server will place each of the Half-Maps together either by their length or width.
- **Source:** [Spielidee page 1 paragraph 11, “**Die Kartenhälften werden vom Server zufällig entweder an den kurzen (Kartenabmessungen 5 x 20) oder den langen Seiten (Kartenabmessungen 10 x 10) zusammengefügt.**”]

## Requirement 6

- **Description:** *Field Types:* Each field must represent only one of the possible types of fields: Water, Mountain or Grass.
- **Source:** [Spielidee page 1 paragraph 11, “**Jedes Feld repräsentiert genau eine von drei möglichen Terrainarten: Wasser, Wiese oder Berg.**”]

## Requirement 7

- **Description:** *Player Movement:* The player of each client must be able to move through the map fields.
- **Source:** [Spielidee page 1 paragraph 4, “**Um den Schatz zu finden, bewegen beide KIs ihre Spielfigur über die Karte und decken dabei mit ihrer Spielfigur schrittweise Kartenfelder auf.**”]

## Requirement 8

- **Description:** *Water Field Movement:* Game will end if any player steps towards a Water field.
- **Source:** [Spielidee page 1 paragraph 11, “**Außerdem darf Wasser unter keinen Umständen betreten werden.**”]

## Type of Requirement: Non-Functional

### Requirement 9

- **Description:** *Game Actions Limit: A game may not last longer than 320 actions last.*
- **Source:** [Spielidee page 1 paragraph 9, “**„ein Spiel insgesamt nicht länger als 320 Spielaktionen (und damit 320 Runden) dauern darf.“**]

### Requirement 10

- **Description:** *Time limit per action: An action may not exceed the time limit of 5 seconds.*
- **Source:** [Spielidee page 1 paragraph 9, “**Für jede dieser rundenbasierten Spielaktion hat die KI maximal 5 Sekunden Bedenkzeit.**”]

### Requirement 11

- **Description:** *Direction of movement: The players are only allowed to move vertically and horizontally, i.e. up/down/left/right.*
- **Source:** [Spielidee page 1 paragraph 14, “**Eine Spielfigur kann sich nur horizontal und vertikal zu direkt benachbarten Feldern bewegen.**”]

### Requirement 12

- **Description:** *Action: The player must always place an action:move/half-map.*
- **Source:** [Spielidee page 1 paragraph 9, “**Eine KI kann hierbei nicht auf das Setzen einer Aktion verzichten, sondern muss immer eine Aktion durchführen.**”]

## Type of Requirement: Design

### Requirement 13

- **Description:** *Client-server*: This game must be played between two clients on a server. The server is responsible for communication between the two clients and keeping the regulations of the game upheld.
- **Source:** [Spielidee page 1 paragraph 1, “**Dabei werden die KIs von einem Server unterstützt der beide Clients koordiniert, Datenaustausch ermöglicht...**”]

## Task 2: Requirement Documentation

Documentation of a requirement that fits the relevant area according to the specified scheme. Consider a requirement for which all parts of the template can be populated with relevant content. We recommend selecting a functional requirement.

### Documentation Of Requirement

- **Name:** Half-Map Generation
- **Description:** A half-map must be generated from each clients and sent to the server in the beginning of the game, the server will then put them together and prepare it for access for the clients again.
- **Priority:** High
- **Relevante Anforderungen:**
  - o Requirement 2: *Generation Of Castle*
  - o Requirement 3: *Generation Of Treasure*
  - o Requirement 4: *Half-Map Configuration*
  - o Requirement 6: *Field Types*
- **Relevant Business Rules:**
  - o The half map should be 5x10 and overall with 50 nodes.
  - o Each half of the map must contain at least 10% mountain field, 48% grass field, 14% water field.
  - o Fort and Treasure must be on a grass field.

- o Fort and Treasure must not be on the same field.
- o Islands (grass/mountain surrounded with water) must not be created.

- **Impulse and Event - Typical Flow:**

- **Precondition:**

- o The game is registered.
- o Both clients are registered.

- **Main Flow:**

- o **Impulse** : The first client sends their half-map.
- o **Event** : The server validates the half-map.
- o **Impulse** : The second client is notified of their turn.
- o **Event** : The client sends a call through the controller to the model to generate the half-map.
- o **Impulse** : The model calls relevant methods to generate and validate a half-map.
- o **Event** : the model sends the map back to the controller.
- o **Impulse** : The controller sends the created half-map through the network class to the server.
- o **Event** : The controller gets notified of half-map being accepted by server.

- **Postconditions:**

- o The server creates and gives access to the full-map.

- **Impulse and Event - Alternative Flow:**

- **Precondition:**

- o The game is registered.
- o Two clients are registered.

- **Main Flow:**

- o **Impulse** : The first client sends their half-map.
- o **Event** : The server validates the half-map.

- o **Impulse** : The second client is notified of their turn.
- o **Event** : The client sends a call through the controller to the model to generate the half-map.
- o **Impulse** : The model calls relevant methods to generate and validate a half-map.
- o **Event** : Validation recognises an island and a new map must be generated.
- o **Event** : The model gets a new validated map and sends the map back to the controller.
- o **Impulse** : The controller sends the created half-map through the network class to the server.
- o **Event** : The controller gets notified of half-map being accepted by server.

• **Postconditions:**

- o The server creates and gives access to the full-map.

• **Impulse and Event - Failed Flow:**

• **Precondition:**

- o The game is registered.
- o Two clients are registered.

• **Main Flow:**

- o **Impulse** : The first client sends their half-map.
- o **Event** : The server validates the half-map.
- o **Impulse** : The second client is notified of their turn.
- o **Event** : The client sends a call through the controller to the model to generate the half-map.
- o **Impulse** : The model calls relevant methods to generate and validate a half-map but validation overlooks a created island in the map.
- o **Event** : the model sends the map back to the controller.
- o **Impulse** : The controller sends the created half-map through the network class to the server.
- o **Event** : The server rejects the half-map as it's not valid.

• ***Postconditions:***

- o The second client loses and the game is over.

• **User Story:** As the Client I'd like to have entities and functions which gradually will help me create a map that encompasses all the requirements set by the server and helps me win the game.

• **Benutzerschnittstelle:**

- o Turn: MUST WAIT
- o Player ID xxxx -> sent Half-Map
- o Turn: MUST ACT
- o Half-Map created
- o Half-Map sent
- o Player ID xxxx -> sent Half-Map
- o Full-Map available

• **Externe Schnittstellen:**

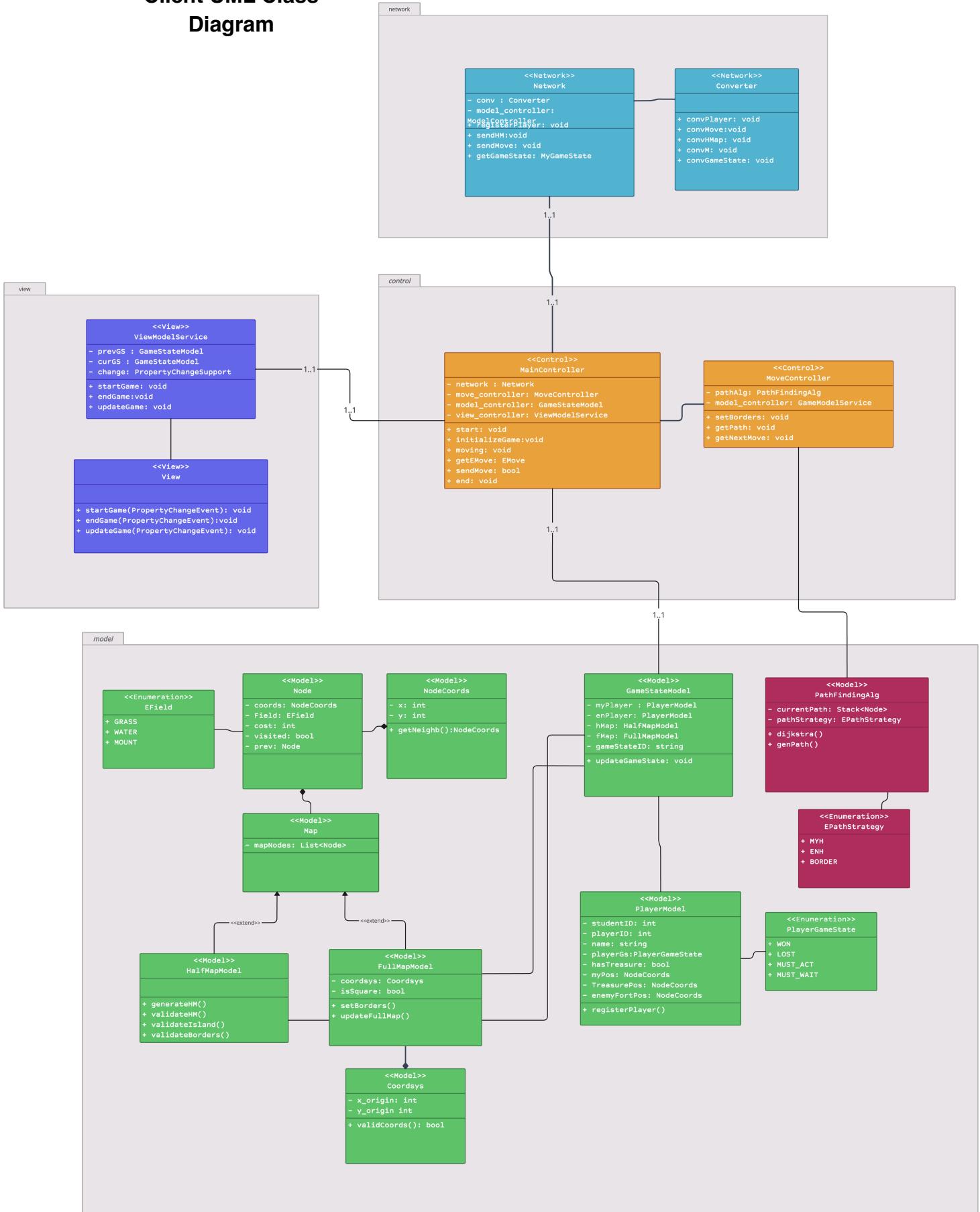
- o The client makes an HTTP POST request to the server's half-map specific endpoint (/gameID/halfmaps) with the half-map in the body of an XML message
- o The client then receives a ResponseEnvelope from server in which ERequestState indicates whether the request was accepted or not (Okay/Error).

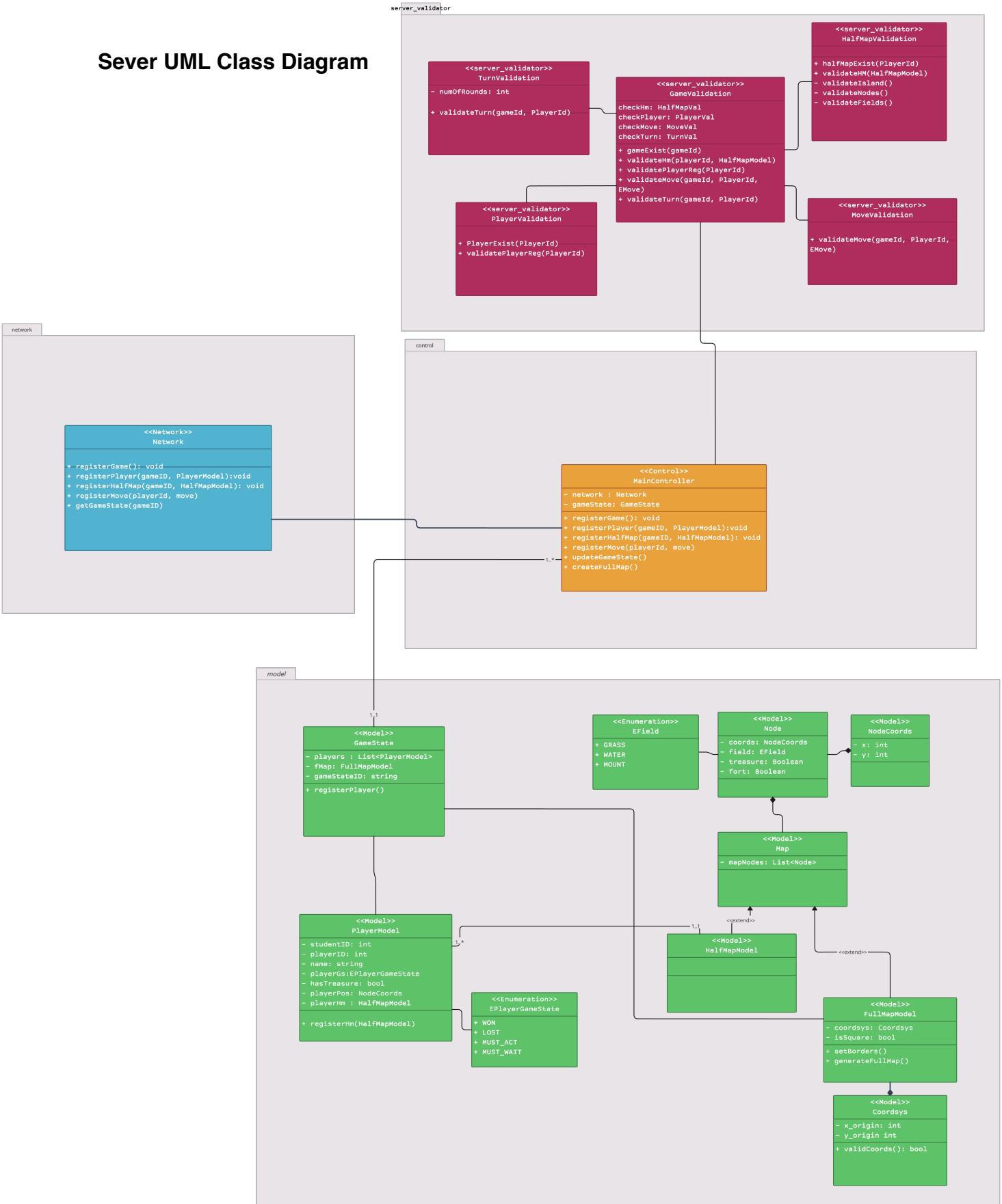
## Task 3: Design, Model and Validate Architecture

Manually model all necessary packages, classes and their methods (including relationships) as two UML class diagrams. Make sure that the models contain meaningfully named packages, classes, methods (including parameters and returns) and fields and that the specifications of the game-idea or the network protocol are completely covered in a meaningful granularity.

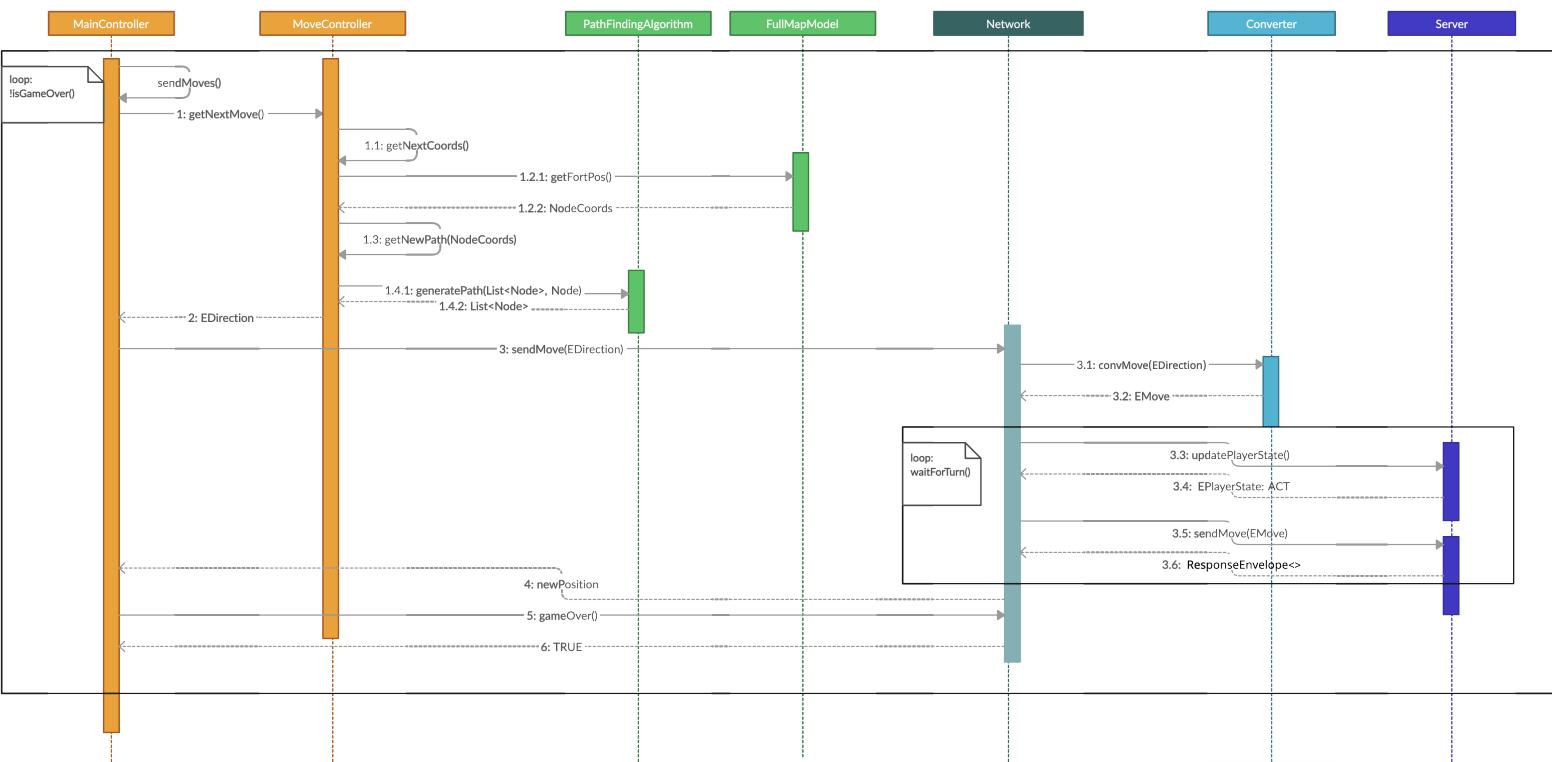
**Based on the Class Diagrams:** Create two sequence diagrams for the two aspects given in the exercise instructions. All diagrams created should be semantically and syntactically correct and consistent with each other.

## Client UML Class Diagram



**Sever UML Class Diagram**


### Client Sequence Diagram



### Server Sequence Diagram

