Communication Protocol

AM44

June 30, 2022

Summary

1 Introduction

This document illustrates the sequence diagrams about the exchange of messages between client and server. In particular, those diagrams show the interaction sequences without negative responses. If the current player's client sends incorrect or unexpected messages, the server will reply with an error message. Otherwise if the other players clients send messages the server will ignore them. For the messages format we use com.google.gson.Gson library to convert special objects into JSON strings and vice-versa. At the end of every phase, there is a "GamePhase" message together with a "CurrentPlayer" message which inform clients about the game status and the available choices.

2 Sequence Diagrams

2.1 Creation Phase

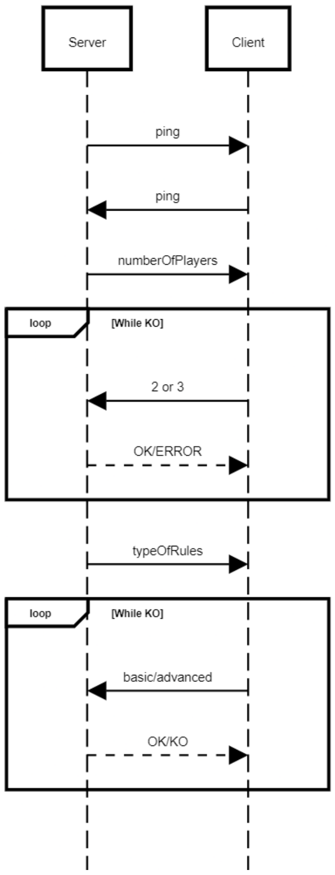

Figure 1: Creation Phase sequence diagram

2.2 Login Phase

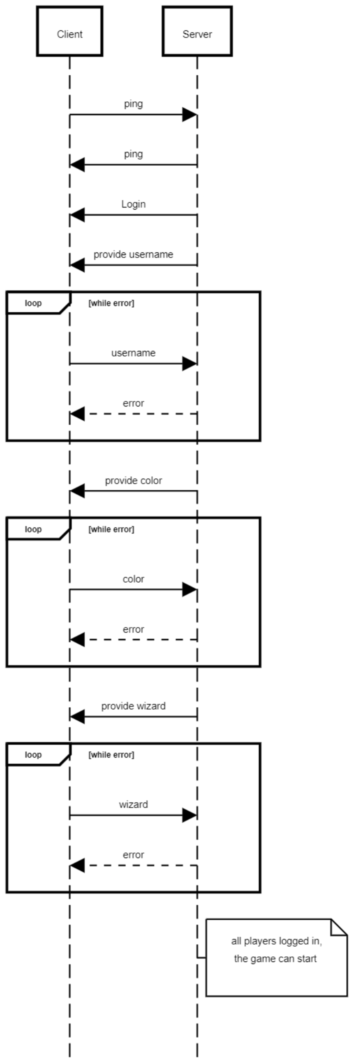

Figure 2: Login Phase sequence diagram

2.3 Planning Phase

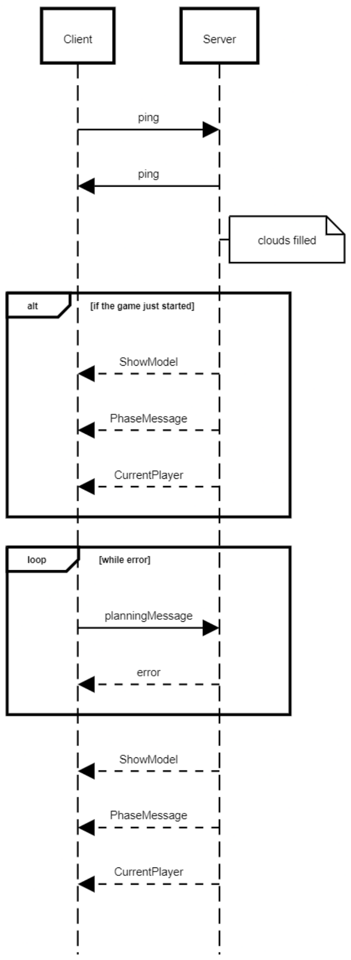

Figure 3: Planning Phase sequence diagram

2.4 Action Phase

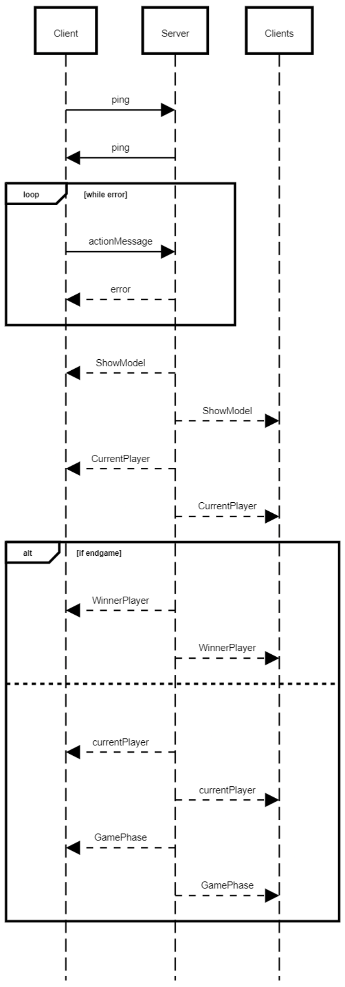

Figure 4: Action Phase sequence diagram

2.5 Client Disconnection

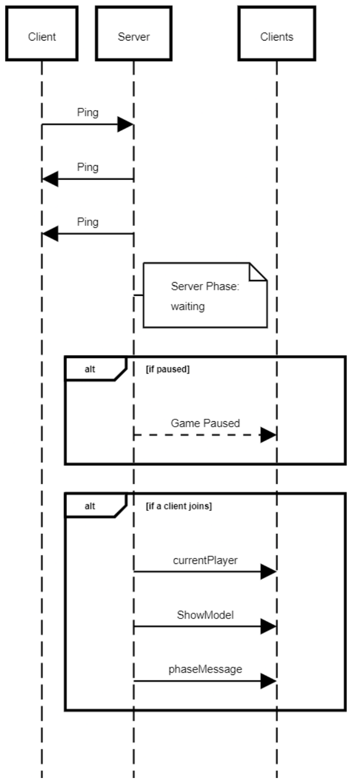

Figure 5: Client disconnection sequence diagram

2.6 Client Reconnection

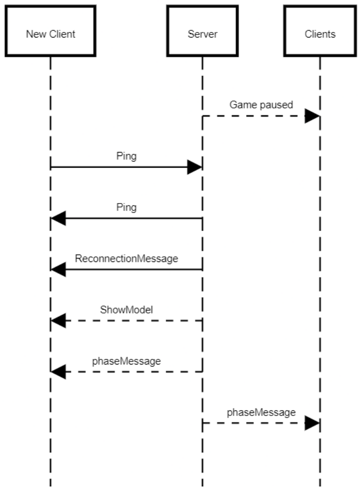

Figure 6: Client reconnection sequence diagram

2.7 Server Disconnection

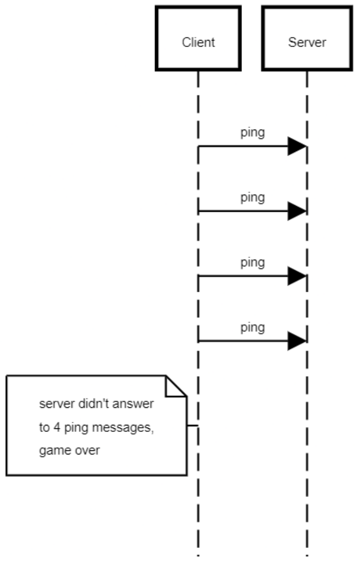

Figure 7: Server disconnection sequence diagram

3 JSON Messages

Our messages are composed of an Header and a Payload, where the first one defines the kind of message exchanged, and the second actually contains the information. The Message class is used to maintain an high level of abstraction. When a message is received the server reads the Header section, determining the type of Payload in order to convert the JSON string to the correct Java object.

3.1 Client and Server messages:

3.1.1 ping Message:

```
{"Header":"ping",
"StringPayload":{
    "String": "string",
    }
}
This message is used to send ping and pong
```

3.2 Server messages:

3.2.1 Phase Message:

```
{"Header":"planning",
"StringPayload":{
    "currentPlayer": "string",
    }
}

{"Header":"action",
"ActionPayload":{
    "moveStudents": "boolean",
    "moveMotherNature": "boolean",
    "selectCloud": "boolean",
    "playCharacter": "boolean"
    "currentPlayer":"string"
    }
}
```

3.2.2 Winner Player Message:

```
{"Header":"winnerPlayer",
"StringPayload":{
    "winner" : "String",
    }
}
```

3.2.3 Error Message:

```
{"Header":"errorMessage",
"StringPayload":{
    "error": "String"
    }
}
```

3.2.4 Creation requirement Message:

```
{"Header":"creationRequirementMessage",
"StringPayload":{
    "type": "String"
    }
}

This message is used to ask for username, color, wizard
and numberOfPlayers during the login phase
```

3.2.5 Player re-connection Message:

```
{"Header":"reconnection",
"ReconnectionPayload":{,
    "username": "String",
    }
}
```

This message delivers the information about the reconnection

3.2.6 Show Model Message:

```
{"Header":"showModelMessage",
"ShowModelPayload":{
    "currentPlayerUsername": "String",
    "PlayersList": "List<Player>",
    "islands": "List<Island>",
    "clouds": "List<Cloud>",
    "mothernature": "int",
    "deactivators": "int",
    "coinReserve": "int",
    "characters":"List<CharacterInformation>,
    "monkCreatures":"List<Creature>",
    "princessCreatures":"List<Creature>",
    "jokerCreatures":"List<Creature>",
    "updatePlayersAssistant":"boolean",
    "updatePlayersEntrance":"boolean",
    "updatePlayersDiningRoom":"boolean",
    "updateIslands":"boolean",
    "updateAll":"boolean",
    "updateClouds":"boolean",
    "updateMotherNature":"boolean",
    "updateCoinReserve":"boolean",
    "updatePlayedCharacter":"boolean",
    "reconnection":"boolean",
    }
}
```

3.3 Client Messages:

3.3.1 Planning Message:

```
{"Header":"planning",
"PlanningAnswerPayload":{
    "indexOfAssistant" : "int",
    }
}
```

3.3.2 Action Message:

```
{"Header":"actionMessage",
"ActionAnswerPayload":{
    "moveStudents" : "boolean",
    "moveMotherNature" : "boolean",
    "selectCloud" : "boolean",
    "playCharacter" : "boolean",
    "isDestinationDiningRoom" : "boolean",
    "studentCreatureToMove": "Creature",
    "clientInt": "int"
    }
}
```

```
This message is used to communicate to the server the action
that the player wants to perform, every boolean parameter is
"false" by default, except the chosen one.
```

3.3.3 Characters parameters Message:

```
{"Header": "charactersParametersMessage",
"CharactersParametersPayload":{
    "providedSourceCreatures": "List<Creature>",
    "providedIslandIndex": "int",
    "providedMnMovements": "int",
    "providedDestinationCreatures": "List<Creature>",
    }
}
```

This message is used to communicate to the server the values that the character effect needs.

3.3.4 Quit Message:

```
{"Header":"quitMessage",
"StringPayload":{
    "username" : "String"
    }
}
```

This message is used to communicate the willingness to quit the game.