



**University of  
Zurich**<sup>UZH</sup>

# IT Software Praktikum

**University of Zurich**

**Department of Informatics**

**FS 2017**

***Supervisors: Prof. Dr. Harald C. Gall, Sali Zumeri***

## Assignment 4 - Group 13

David Wyss	14-734-453
Kenny Kunzmann	15-707-961
Remo Hertig	13-738-323
Roland Benz	97-923-163
Spasen Trendafilov	15-712-805
Vincent Rüegge	15-700-966

Zurich, 12.05.2017

# List of Contents

<b>1 What Was It About</b>	<b>4</b>
1.1 Our Goals	4
1.2 Goal Of Milestone 4	4
1.3 Lessons Learned	4
<b>2 Domain Model/Class Diagram</b>	<b>5</b>
2.1 Domain Model	5
2.2 Class Diagram	6
<b>3 User Stories</b>	<b>13</b>
3.1 Final Version of User Stories	13
<b>4 REST Interface/WebSocket</b>	<b>21</b>
4.1 Specification of the REST Interface	21
4.2 Specification of the WebSocket	23
4.2.1 Interaction Messages	23
4.2.2 Interaction Diagram	24
<b>5 Risk Analysis</b>	<b>26</b>
<b>6 User Interface</b>	<b>28</b>
6.1 User Interface from Milestone 1	28
6.1.1 Login Screen	28
6.1.2 Lobby Screen	28
6.1.3 New Game Screen	29
6.1.4 In Game Screen	30
6.1.5 End Game Screen	30
6.2 User Interface from Milestone 2	31
6.2.1 Login Screen	31
6.2.2 Lobby Screen	32
6.2.3 New Game Screen	33
6.2.4 In Game Screen	34
6.2.5 End Game Screen	36
6.3 User Interface from Milestone 3	37
6.3.1 Login Screen	37
6.3.2 Lobby Screen	38
6.3.3 New Game Screen	38
6.3.4 In Game Screen	39
6.3.5 End Game Screen	39
6.3.6 Click Sequence Diagrams	40
6.4 User Interface from Milestone 4	41

6.4.1 Login Screen	41
6.4.2 Lobby Screen	41
6.4.3 New Game Screen	42
6.4.4 In Game Screen	42
6.4.5 End Game Screen	43
<b>7 Time Planning</b>	<b>44</b>
7.1 Time Planning of Milestone 1 and Milestone 2	44
7.1.1 Task List until Milestone 2	44
7.1.2 Gantt Chart until Milestone 2	46
7.2 Time Planning from Milestone 2 to Milestone 4	46
7.2.1 Task List from Milestone 2 to Milestone 4	46
7.2.2 Gantt Chart from Milestone 2 to Milestone 4	51
7.3 Redefined Time Planning for Milestone 4	53
7.3.1 Redefined Task List for Milestone 4	54
7.3.2 Redefined Gantt Chart for Milestone 4	55
7.3.3 Issue Tracker Client	55
7.3.4 Issue Tracker Server	61
<b>8 Testing</b>	<b>65</b>
8.1 Example Unit Test	65
8.2 Final Test Coverage	66
8.3 Fast Forward Function	68
<b>9 Development Infrastructure</b>	<b>69</b>
9.1 Published Game on Firebase	69
9.2 Backend on Heroku	69
9.3 Software Code on GitHub	69
<b>10 Glossary</b>	<b>70</b>
<b>11 List Of Tables</b>	<b>72</b>
<b>12 List Of Figures</b>	<b>73</b>
<b>13 Auto Generated Class Diagram Final Version</b>	<b>74</b>

# 1 What Was It About

## 1.1 Our Goals

Whenever you are able to create something and see an output, your intrinsic motivation starts to fire up. For this course, this state of mind was crucial.

In the beginning we set ourselves the goal to create an intuitive and fun to play web application. Obviously, with no previous experience, it is easy to state such claims but in the end we, as a group, had to deliver. In order to achieve all of our goals and visions, we made sure to develop a solid first mockup of our user interface and continuously develop it. We thought about different functionalities and where to use them. Drag and Drop versus Click, dynamic versus static. Additionally, we had to make sure that our game doesn't only look nice but that its logic works flawlessly. Striving for a high test coverage and a solid structure was embedded in our work ethic.

## 1.2 Goal Of Milestone 4

The goal of Milestone 4 is to give the game its final touch. This means, implementing missing functionalities and adding styling additions where necessary, in order to hand in a rock solid, funny and intuitive game.

This report can be seen as a guideline through our project development process. For some categories (e.g. User Interface, Time Planning etc.) the contents of the last three reports are listed in chronological order. Thus providing overview and the possibility to take a deeper look at the development of one category.

## 1.3 Lessons Learned

This paragraph describes our project development process in the form of a summary. It does not only focus on issues and tasks leading to problems, but also provides overview and insight into our work for the past ten weeks and what we can say about SoPra.

“Tell me and I forget. Teach me and I remember. Involve me and I learn.”

This quote from Benjamin Franklin pretty much sums up what SoPra is about. At least, as we saw it. It would be quite an understatement to say that SoPra is easy and does not take up a lot of time. Frankly, none of us were able to do anything for other courses during the semester.

But instead of just trying to remember slides and hoping for a decent grade in the final exam, we were able to compose a project, guide ourselves through it and close it during the final stages. For most of us, it was the first time putting so much planning and effort into a single project. Because of the intense work ethic and the time constraints, there was no way around it. We just had to deal with it. Thus, we had to really look into the material and cope with all the necessary tools.

Of course, a project consists not only of smooth times. Struggles of all kinds and several sleepless nights were part of it too. Especially after Milestone 2, where the actual coding started, it took us quite some time to get things going. Almost none of us had previous

experience and the requirements for the course are heavy. No previous courses teaching us, HTML, CSS, Angular2, Spring, TypeScript or whatever was necessary to create the web application.

Because of that lack of knowledge, we invested time and effort into the implementation of third party code, such as dragula, which in the end, had to be completely restructured. Also the communication between the frontend and the backend team didn't work fluently from the beginning. Maybe, for the future, some previous courses and introductions would be helpful and most certainly much appreciated.

Nevertheless, we are very happy with the outcome of our project. Our team ethic and spirit pushed us through the development process and helped us delivering a fully functional first web application. If we were to redo such a project, we would definitely stick to the same granularity and precision of our time planning. Despite all the struggles and time losses, we were able to fulfill each deadline in time and deliver everything required.

In the end it was fascinating to see how easily an IT project becomes complex. So far we have only heard such claims in theory and never actually experienced it ourselves. Therefore SoPra was a very good introduction into the world of IT projects, their conceptualization and development.

## 2 Domain Model/Class Diagram

The following chapter includes the domain model from milestone 1 as well as the class diagram from milestone 2. The class diagram is illustrated with an overview graphic first, followed by snippets of the different sections of the diagram.

A from the codebase generated UML diagram can be found found in the appendix (13. Auto Generated Class Diagram Final Version), which shows certain packages from the main game logic in the final form. It closely resembles the planned architecture and has been only slightly extended.

### 2.1 Domain Model

Figure 2.1.1 shows the domain model of the backend. The 'action' **classes** at the top are responsible to calculate the allowed actions at each state of the game. The Game **class** manages all parts of the game, represented by all the other classes shown, and communicates with the frontend via representational state transfer (REST) interface. Maybe not every class shown in this domain model may be implemented as a class in a future class model but end up as a base type attribute of another class.

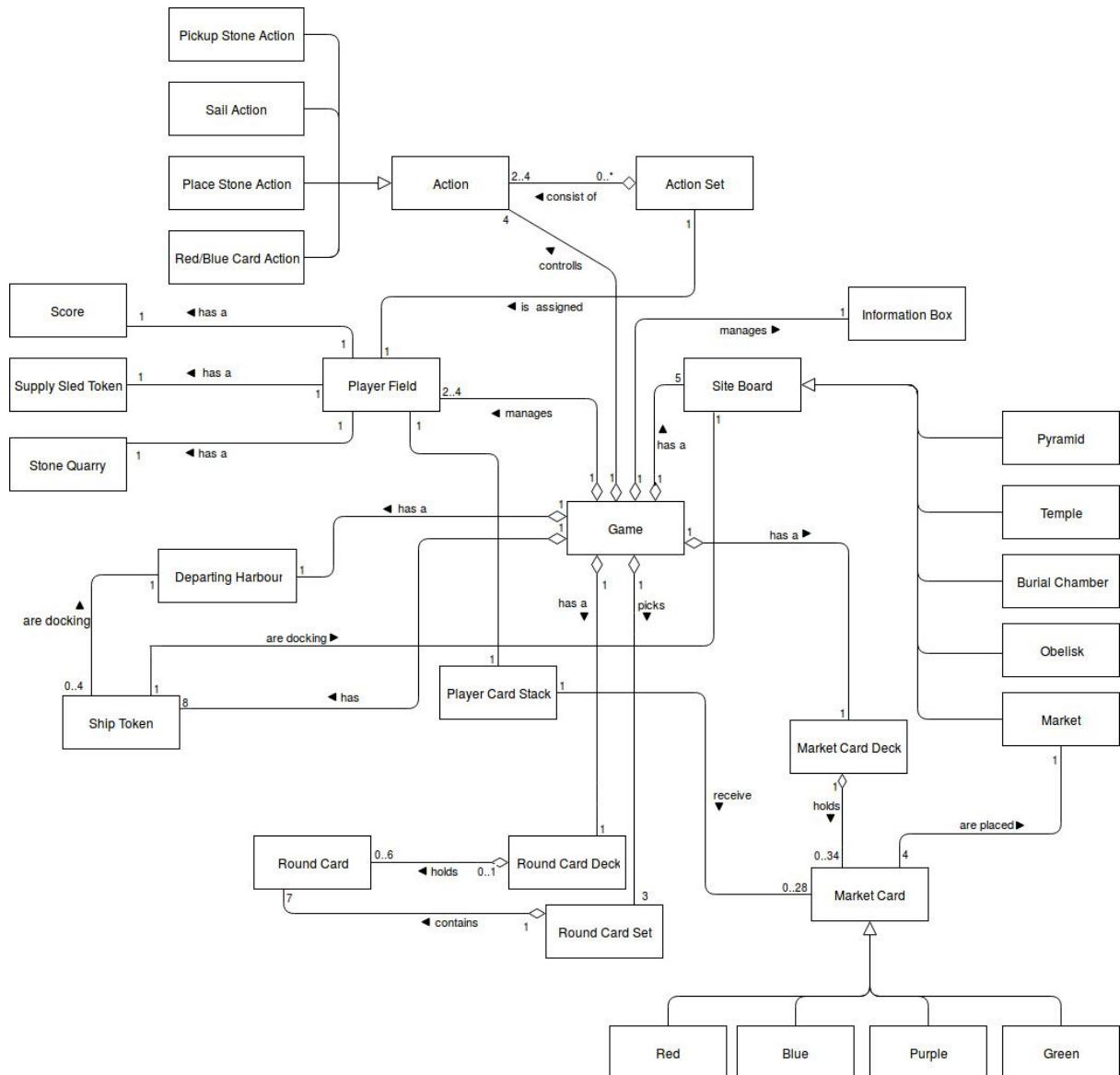


Figure 2.1.1

## 2.2 Class Diagram

Figure 2.2.1 shows a compact overview of the class model. It serves as the main architectural representation of the code we are going to implement at the backend side of our client- server web-application.

Figure 2.2.2 to 2.2.4 show sections of the class model in more detail. For important classes or groups of classes a short description is given. Not shown here are the REST interfaces and the web-sockets, which we use to send messages between the server and its clients. These are discussed in chapter 4.1/4.2.

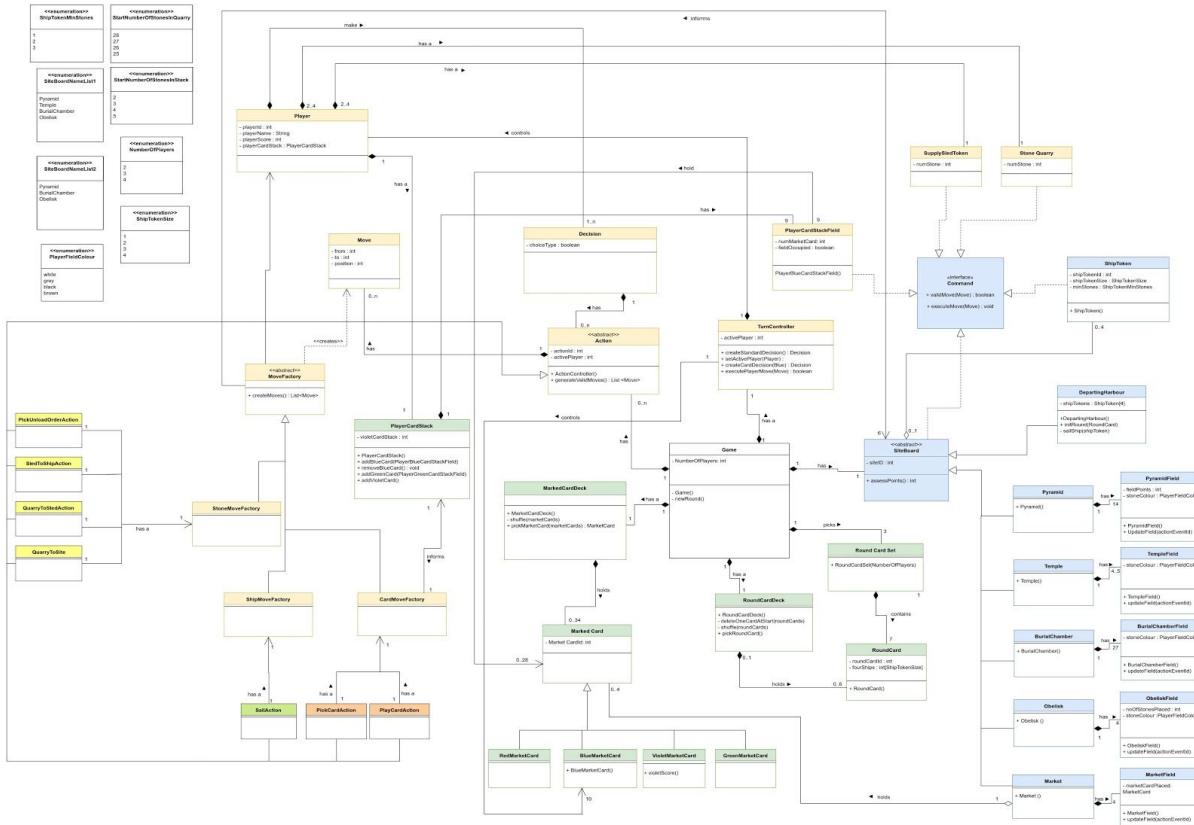


Figure 2.2.1

Figure 2.2.2 shows the **Game** class and its attributes. The Game class is responsible for starting a new game, starting and ending the six rounds and eventually ending the game. At first it instantiates its main entities, **MarketCardDeck**, **RoundCardDeck**, **RoundCardSet**, **SiteBoard**, **Action**, **TurnController** to which the Game class delegates concrete tasks. In a nutshell, those entities are used to store the last state of the game, calculate and inform the active player about his or her allowed moves, validate the move the active player performs, and eventually execute or reject it and broadcasts the current state to all players.

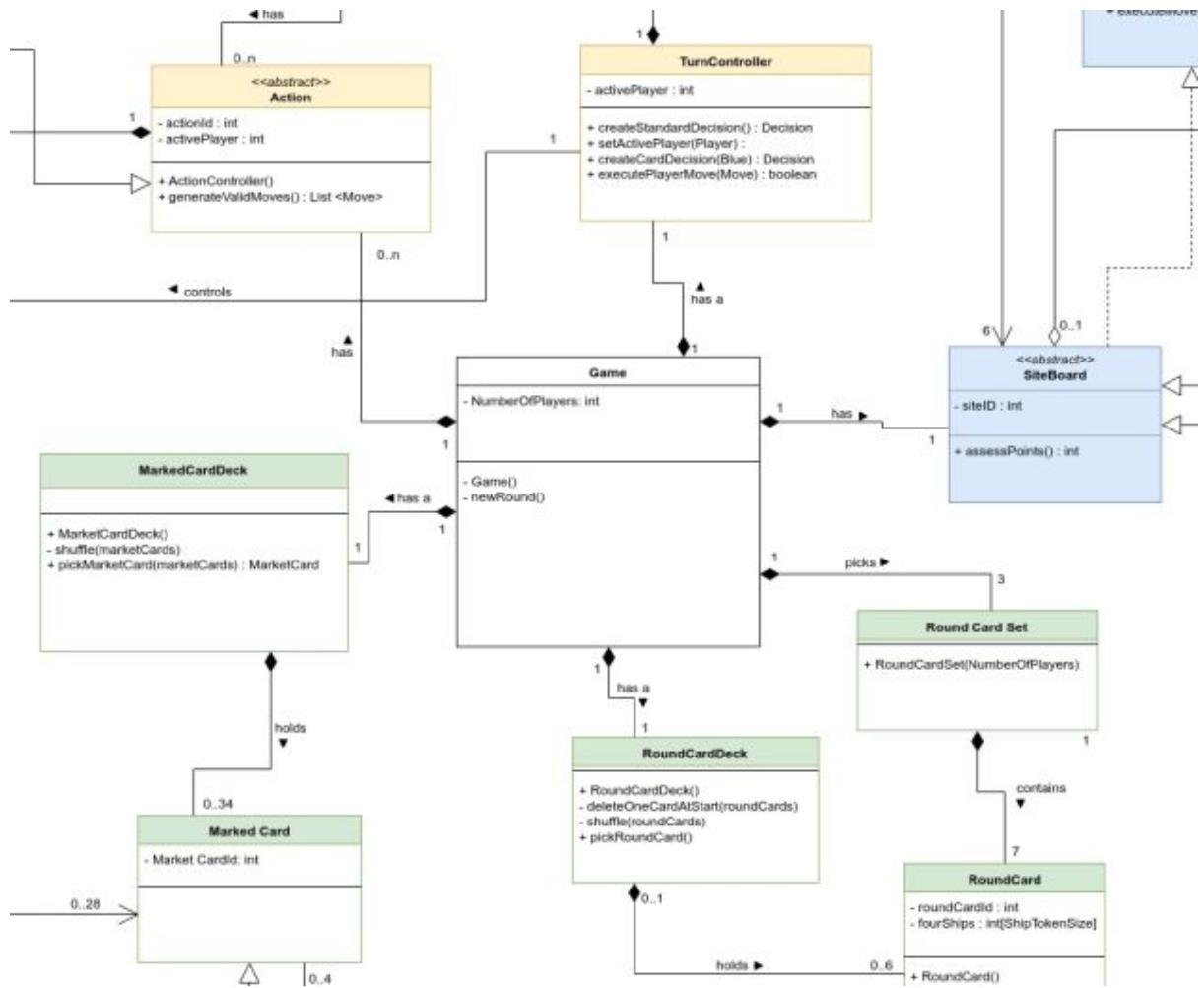


Figure 2.2.2

Figure 2.2.3 shows the **TurnController** which manages the **Player** class and decides, who the currently active player is. The TurnController gets information about player interactions, modelled in the **Move** class, and executes a move carried out by the active player by delegating the information to the involved entities, **PlayerCardStackField**, **ShipToken**, **SupplySledToken**, **StoneQuarry**, **SiteBoard**.

When a player, modelled by the **Player** class, is informed by the TurnController that (s)he is the active player (s)he can make a decision, modelled by the **Decision** class, which move (s)he wants to carry out.

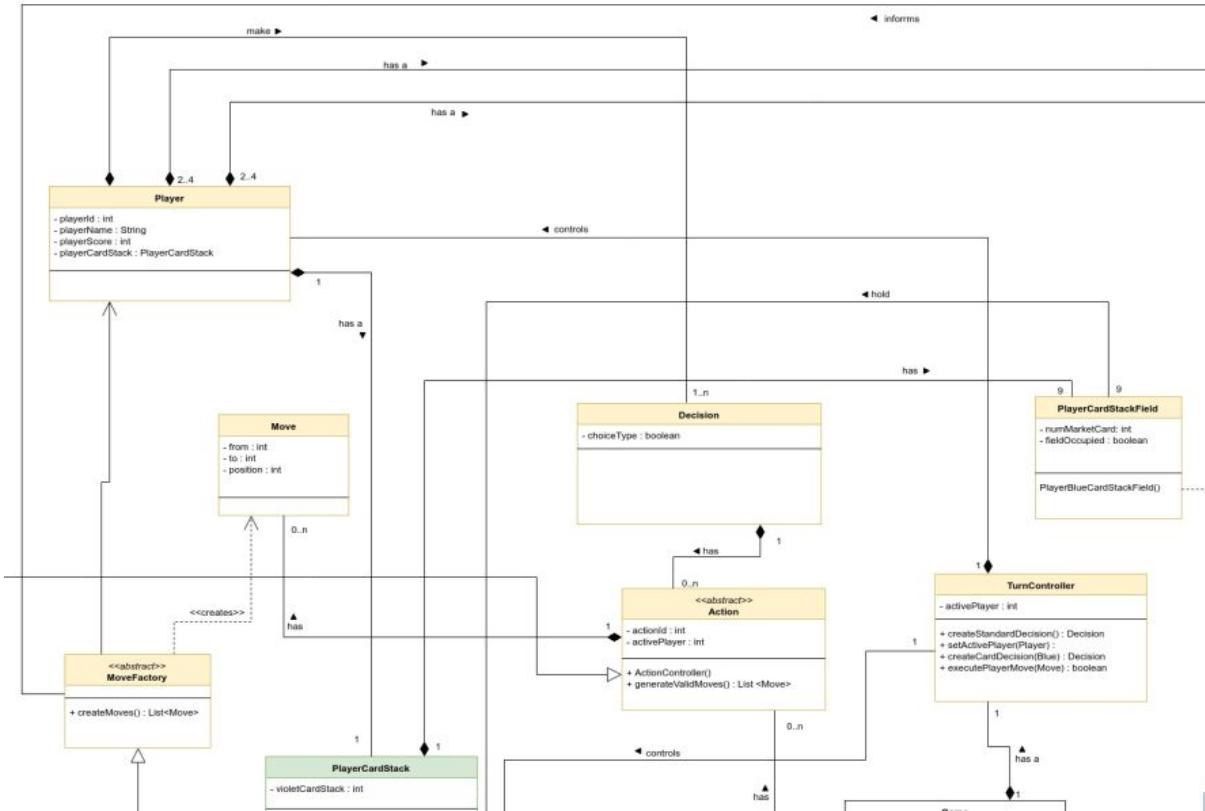


Figure 2.2.3

Figure 2.2.5 shows the **MoveFactory**. Because there are so many possible move variants, we decided to apply a variant of the factory design pattern. There are three concrete move factories, **StoneMoveFactory**, **ShipMoveFactory** and **CardMoveFactory** which calculate all possible moves in their respective scope.

And in order to further package the moves instantiated by the **Move** class, we model another layer of user interactions, called actions, in the **Actions** class. The actions represent the seven basic user interactions which are modelled in the seven concrete action classes:

**PlayCardAction**,      **PickCardAction**,      **SailAction**,      **PickUnloadOrderAction**,  
**SledToShipAction**, **QuarryToSledAction**, **QuarryToSiteAction**.

Each concrete action class has a list of its own valid moves generated by its corresponding move factory. This is done by calling the **createvalidmoves()** method, which on its part calls **createmoves()** method of the corresponding MoveFactory. This list of valid actions and its moves is sent to the frontend and used by each UI component to decide what is a valid user interaction and what not.

Figure 2.2.4 summarizes the information given in the figures 2.2.3 and 2.2.5.

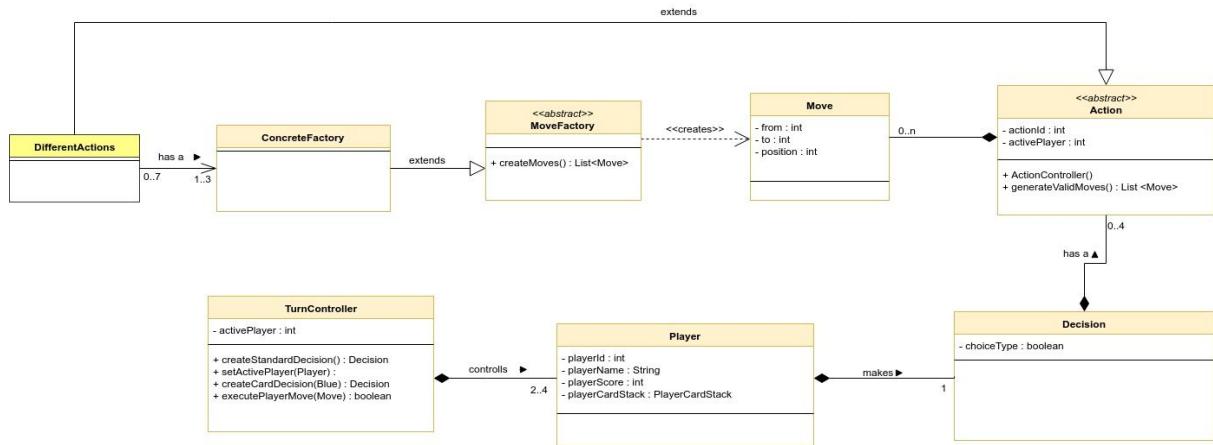


Figure 2.2.4

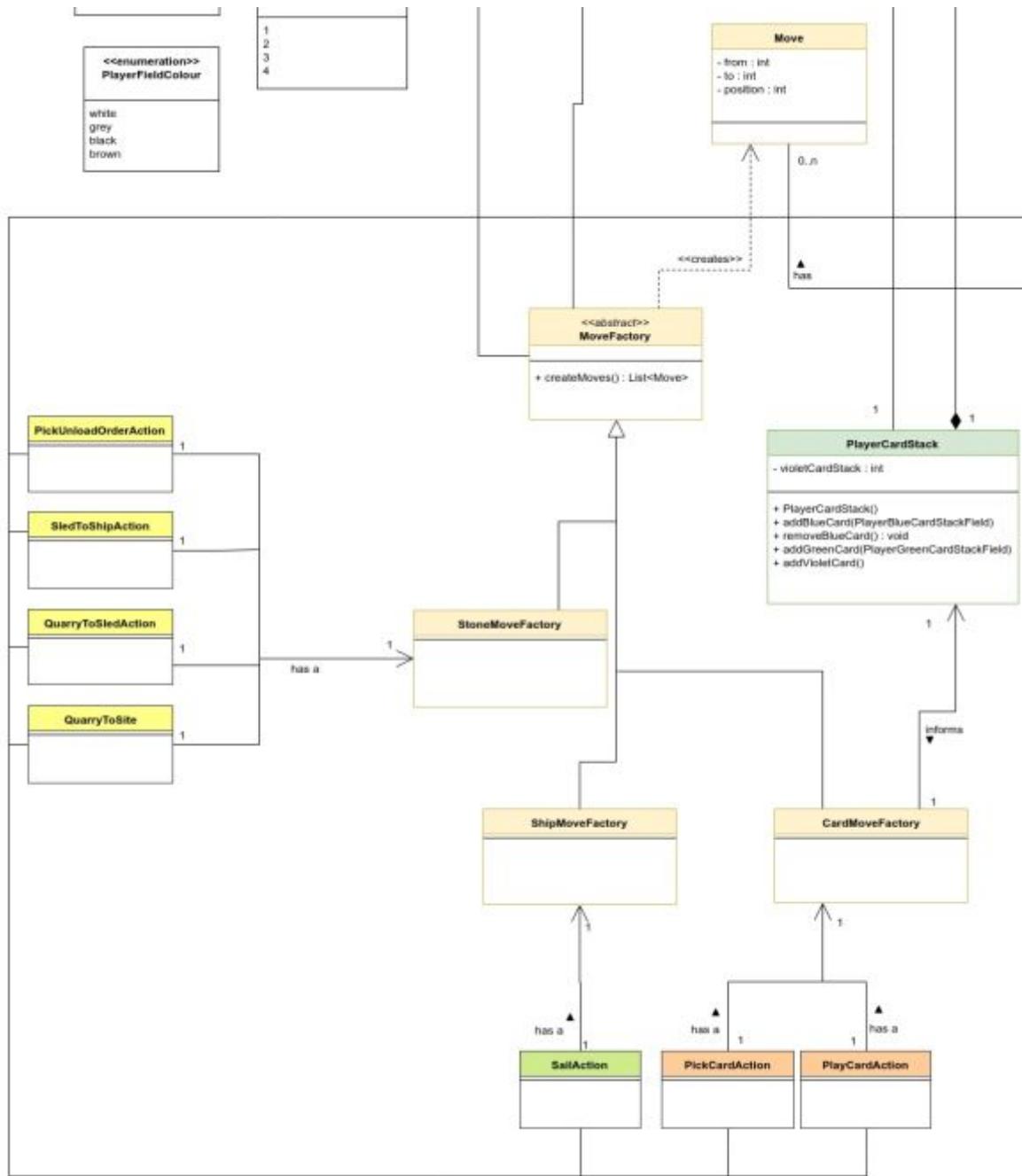


Figure 2.2.5

Figure 2.2.6 shows the **Command** interface and the classes which implement it. In order to make sure that only valid moves are executed by the backend, we applied the command design pattern. This makes sure invalid user interactions, which may result from bugs in the frontend software code or from cheating by a user, are rejected. No invalid user interaction can be accepted as a valid new state and processed further. The **PlayerCardStackField**, **ShipToken**, **SupplySledToken**, **StoneQuarry**, **SiteBoard** classes need to implement the Command interface in order to be able to assess, whether a move made by the active player is valid and then execute or reject it.

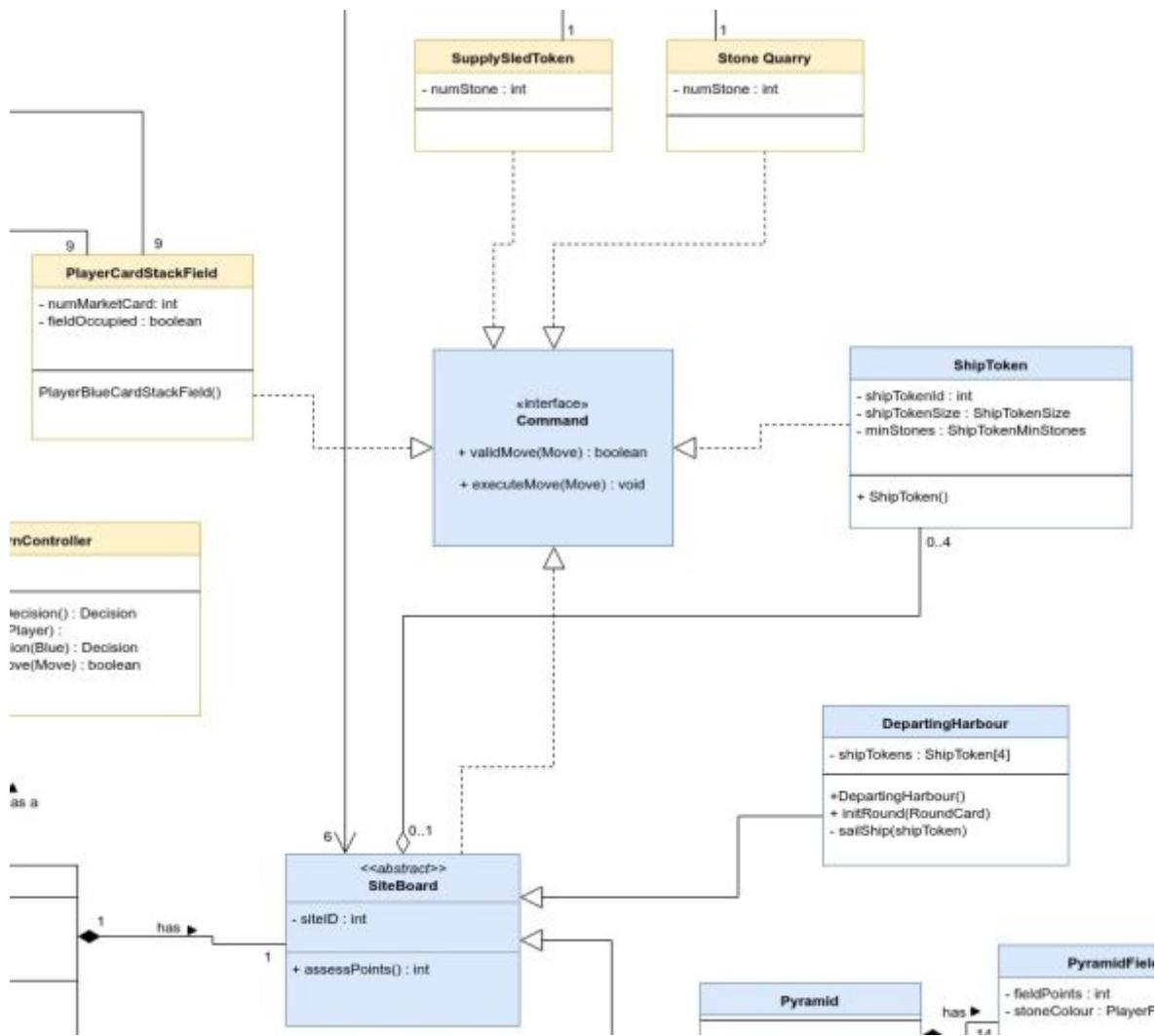


Figure 2.2.6

# 3 User Stories

This sections contains our final version of our user stories. We extended and redefined them during the whole project.

## 3.1 Final Version of User Stories

The following user stories shown in Table 3.1.1 and 3.1.2 are arranged in chronological order. The green range **S-1a.x** indicates the beginning of the game in which the player has to log into the game in order to join the lobby and select a game. After the user has successfully registered, the lobby screen is shown next, where the corresponding number range is indicated in blue **S-2ax**. From there on every user has the possibility to either create a new game or join an existing one. The user stories indicated in red **S-3** show the users' needs and wishes on the in-game screen while playing the game from the beginning to the end of each game. Finally, the group of user stories indicated in orange **S-4a.x** describe the rules and actions.

Each user story has a matching **ID** within the corresponding **number range** and a **priority**. “Must” priorities have to be implemented in order to match the game’s rules or to make the game playable online. “Should” priorities are not needed for the game to work although the user sees them as a sign of quality. Lastly, there are “could” priorities that are nice to have but not required to fulfill the minimum expectations.

Number Range	Web-Site	Specific Place	Actor
S-1a.x	Login Screen	login form	as a new arrival to the website
S-2a.x	Lobby Screen	game table	as a guest in the lobby
S-3a.x	In-Game Screen	player field	as a player
S-3b.x	In-Game Screen	departing harbour, unmoved ship tokens	as a player
S-3c.x	In-Game Screen	site boards without market site board	as a player
S-3d.x	In-Game Screen	market site board	as a player, as the active player
S-3e.x	In-Game Screen	timer	as a player, as the active player
S-3f.x	In-Game Screen	card deck	as a player
S-3g.x	In-Game Screen	menu	as a player
S-4a.x	In-Game Screen	game actions / rules	as a player, as the active player

Table 3.1.1

ID	User Story	Priority
S-1a.1	<p>As a new arrival to the website, I want to login in order to get to the lobby screen.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Save chosen user name</li> <li>• Routing to lobby screen</li> </ul>	Must
S-2a.1	<p>As a guest in the lobby screen, I want to see a list of all open games in the lobby screen with their corresponding number of players joined and required.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Game list in lobby screen displays created games</li> <li>• Game list in lobby screen displays number of players joined</li> <li>• Game list in lobby screen displays number of players required</li> </ul>	Must
S-2a.2	<p>As a guest in the lobby screen, I want to be able to add a new game to the list in the lobby screen and automatically join this game.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• User can add a new game with two, three or four player</li> <li>• User joins automatically to the game he created</li> </ul>	Must
S-2a.3	<p>As a guest in the lobby screen, I want to be able to remove a game I added to the list in the lobby screen.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Delete functionality deletes created game in frontend and backend</li> </ul>	Must
S-2a.4	<p>As a guest in the lobby screen, I want to be able to join a game shown in the list in the lobby.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Join button for users to join a game</li> <li>• Update and show list with open games</li> </ul>	Must

S-2a.5	<p>As a guest in the lobby screen, I want to leave a game shown in the list in the lobby screen.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>Leave button deletes user entry</li> </ul>	Must
S-2a.6	<p>As a guest in the lobby screen, I want the game to be started automatically when the number of players joined have reached the number of players required.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>Implement autostart functionality</li> <li>Routing to in-game screen</li> </ul>	Should
S-2a.7	<p>As a guest in the lobby screen, I want to be able to read the game rules.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>Read manual button for opening game manual</li> <li>Open new tab or modal that shows the game manual</li> </ul>	Could
S-2a.8	<p>As a guest in the lobby screen, I want to have links to youtube videos explaining the game.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>Button that leads to youtube videos</li> <li>Open new tab to chosen videos</li> </ul>	Could
S-2a.9	<p>As a guest in the lobby screen, I want to see the player statistics.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>Player statistics are saved</li> <li>Player statistics are displayed in lobby screen</li> </ul>	Could
S-3a.1	<p>As a player in the in-game screen, I want to see the statistics of each player. (stones on the sled, stones on the quarry, playername, individual cards, score, color)</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>Visualize stones on the sled for every player</li> <li>Visualize stones on the quarry for every player</li> </ul>	Must

	<ul style="list-style-type: none"> <li>• Visualize score for every player</li> <li>• Visualize individuals cards for every player</li> <li>• Visualize color for every player</li> <li>• Visualize playername for every player</li> </ul>	
S-3a.2	<p>As a player in the in-game screen, I want to see who the active player is.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Client gets player status from server</li> <li>• Active player status results in change of player UI appearance</li> </ul>	Must
S-3b.1	<p>As a player in the in-game screen, I want to see the unmoved ship tokens (with the free and occupied spots to place stones) in the departing harbour.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Client gets player status from server</li> <li>• Active player status results in change of player UI appearance</li> </ul>	Must
S-3b.2	<p>As a player in the in-game screen, I want to see the minimum required number of stones for each unmoved ship token in the departing harbour in order to be able to move the ship token to a site board.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Display minimum required number of stones in UI</li> <li>• or use of 'original' ship pictures, displaying the minimum required number of stones on each ship</li> </ul>	Must
S-3b.3	<p>As a player in the in-game screen, I want to see to which site boards the unmoved ship tokens in the departing harbour can still sail to.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Fully covered with the implementation of a drag-and-drop functionality</li> </ul>	Must
S-3b.4	<p>As a player in the in-game screen, I want to see in which order the stones on any unmoved ship token in the departing harbour will be unloaded in the current state of the game.</p> <p>Acceptance criteria:</p>	Must

	<ul style="list-style-type: none"> <li>• Design UI so that unload logic is intuitive and resembles design of board game (ships are placed horizontally, on the left side of the sites, facing the sites)</li> </ul>	
S-3c.1	<p>As a player in the in-game screen, I want to see for each site board where the stones have been placed.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Unloading of stones from ship to site is implemented</li> <li>• Stones are visible pictures displayed in the UI</li> </ul>	Must
S-3c.2	<p>As a player in the in-game screen, I want to see for each site board how many points each player would earn at the current state of the game</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Save points depending on game state</li> <li>• Display saved points in user interface</li> </ul>	Could
S-3d.1	<p>As a player in the in-game screen, I want to see the four market cards on the market site board that are available in the current round.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Market card logics are implemented</li> <li>• Each round gets four randomly generated market cards</li> <li>• Market card container on market site is structured</li> </ul>	Must
S-3d.2	<p>As the active player in the in-game screen, I want to be made aware, which market card I am able to choose, from the market site board.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Stone colour indicates player id</li> <li>• Sequence of player choosing is mapped to sequence of stones on a ship</li> <li>• Turn logic corresponding to stone sequence works</li> <li>• UI informs player when it is corresponding stone's (colour) turn</li> </ul>	Must
S-3e.1	<p>As a player in the in-game screen, I want to see the timer of the active player showing how much time he has left for his move.</p> <p>Acceptance criteria:</p>	Could

	<ul style="list-style-type: none"> <li>• Time limit is set</li> <li>• Countdown starts when player starts turn</li> <li>• Timer is player independent and displayed for all players</li> </ul>	
S-3e.2	<p>As a player in the in-game screen, I want the active player to be punished for overstepping the time limit (either being kicked out, points deduction or random move).</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Time limit is set</li> <li>• Countdown starts when player starts turn</li> <li>• Time-out leads to user being returned to lobby, points being reduced or random move</li> <li>• In case of kick-out, user has to be deleted from user array for further game logic</li> </ul>	Could
S-3e.3	<p>As a player in the in-game screen, I want repeatedly inactive users to be disqualified after a certain time.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Term inactive defined</li> <li>• Maximum number of being inactive is set</li> <li>• Number reach leads to player kick-out</li> </ul>	Could
S-3e.4	<p>As the active player in the in-game screen, I want to be warned early enough before overstepping the time limit.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Time limit is set</li> <li>• Overstepping time limit is set</li> <li>• Countdown starts when player starts turn</li> <li>• UI displays information to corresponding player when overstepping time limit is reached</li> </ul>	Could
S-3e.5	<p>As a player in the in-game screen, I want the stones of a disqualified user that has been disconnected to remain on the board in order to keep the game fair.</p> <p>Acceptance criteria:</p>	Could

	<ul style="list-style-type: none"> <li>• Stones on site are round dependent not player status/connection dependent</li> </ul>	
S-3f.1	<p>As a player in the in-game screen, I want to see the current round number at any time.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Game is initialised</li> <li>• Round information (start) received from server</li> <li>• Round number displayed in UI</li> </ul>	Must
S-3g.1	<p>As a player in the in-game screen, I want to be able to leave the game.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Leave button displayed in UI</li> <li>• Routing to lobby screen</li> </ul>	Should
S-3g.2	<p>As a player in the in-game screen, I want to be able to see the game instructions.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Same as for user story S-2a.7</li> </ul>	Could
S-4a.1	<p>As the active player in the in-game screen, I want to see clearly, what moves I can choose from.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Client gets player status from server</li> <li>• Active player status results in change of player UI appearance</li> <li>• Move possibilities are saved as a list and each turn gets updated version of the list</li> <li>• UI appearance changes depending on the received list</li> </ul>	Could
S-4a.2	<p>As the active player in the in-game screen, I want to receive a warning message, when I choose to take stones from the quarry although I only have room for one or two.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Current number of stones in sled must be saved per user</li> <li>• Maximum number of five stones must be set</li> <li>• Remaining slots for stones must be calculated</li> </ul>	Could

	<ul style="list-style-type: none"> <li>• Function for slot check when player clicks on quarry</li> <li>• UI displays message when remaining slot number is below 3 and user clicks on quarry</li> </ul>	
S-4a.3	<p>As a player in the in-game screen, I want to see the scored points of every player after the game.</p> <p>Acceptance criteria:</p> <ul style="list-style-type: none"> <li>• Points are saved for each player</li> <li>• End screen displays each player's score</li> </ul>	Must

Table 3.1.2

# 4 REST Interface/WebSocket

This chapter provides the specification of our REST interface and the WebSocket messages. Additionally, an interaction diagram of the WebSocket service is illustrated.

## 4.1 Specification of the REST Interface

Table 4.1.1 shows information about the **REST interface**. The **ID** column clearly identifies each interface, the **Mapping** column shows the URL with which the service can be accessed by the frontend, the **Method** column shows which HTTP method has to be used, the **Parameter** column shows which parameters can be sent as part of the URL, the **Returned Value** column describes the data structure sent back to the frontend, the **Ret. Links** column shows the returned links, i.e. which URIs can be accessed from that application states, and the **Description** column explains the usage in plain English. We identified 17 interfaces the backend offers to the frontend to start a communication via HTTP requests methods. The classes to implement the communication channels are offered by the Spring boot framework.

Not all game states are served by the REST endpoint, in particular those requiring frequent polling. Instead game states are directly broadcast via WebSockets to the clients. The WebSocket is specified in chapter 4.2.1.

ID	Mapping	Method	Parameter	Returned Values	Ret. Links	Description
1	/games	GET	-	List<Game>	4*	Retrieve list of games
2	/games	POST	numPlayers:int	-	-	Create a new game
3	/games/{gameId}	GET	gameId: string	Game: gameId<string>, players<User>	4	Display game info, joined players etc.
4	/games/{gameId}/join	POST	gameId: string	-	5	Join a game
5	/games/{gameId}/leave		gameId: string	-	1	Leave a game
6	/users	POST	username: string, pwd:string	userToken:string	-	Register an user
7	/users/{userId}/login	POST	userId:string	userToken:string	8	Login an user
8	/users/{userId}/logout	POST	userId:string	-	-	Logout an user

9	/users/{userId}	PUT	userId:string	-	-	Update user data i.e name
10	/users/{userId}	DELETE	userId:string	-	-	Delete an user
11	/games/{gameId}/actions	GET	gameId: string	AvailableActions: List<string>	12*	List of currently available actions for a player
12	/games/{gameId}/move	POST	gameId: string, move: Move	-	12*	Executes a move

Table 4.1.1

## 4.2 Specification of the WebSocket

Using the WebSocket we can get rid of the additional request that would be needed to get the information from the server side. The WebSocket automatically informs the client about the current status. This is visualized by the figure 4.2.1 below.

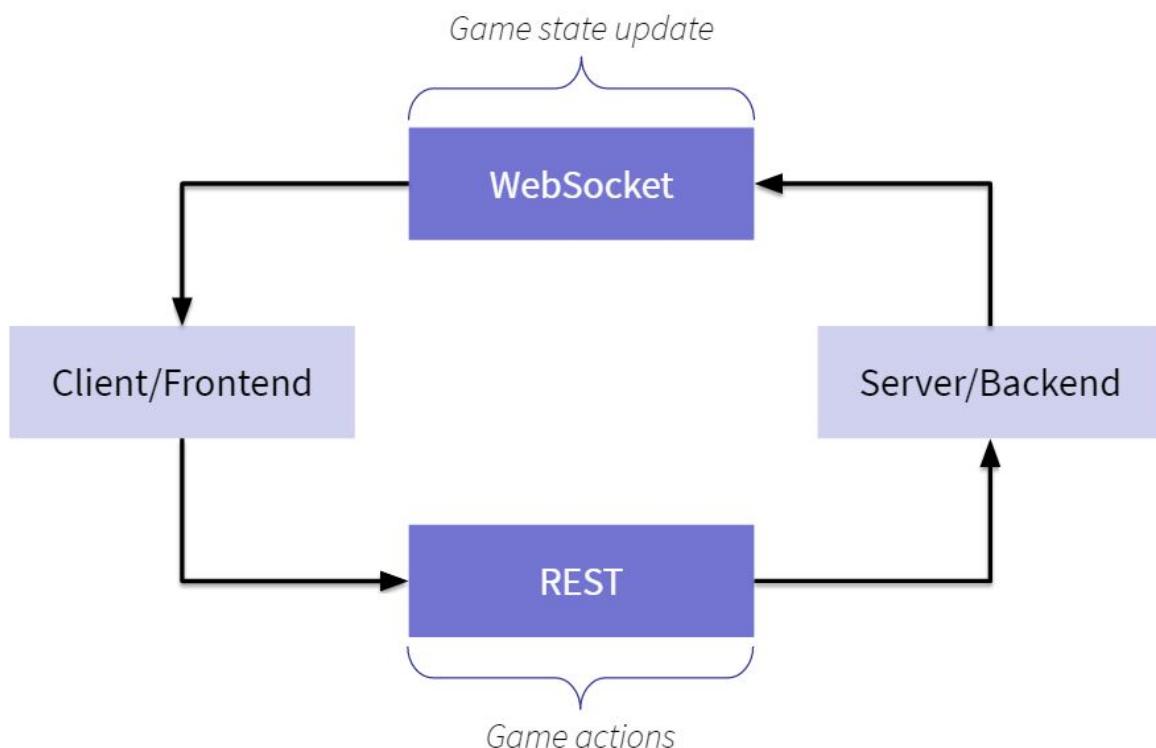


Figure 4.2.1

### 4.2.1 Interaction Messages

Table 4.2.1.1 shows information about the **web-sockets**. The **ID** columns clearly identifies each event, the **EventName** column shows the name used in the program code, the **Origin** column shows whether the frontend (P) or the backend (S) is the initiator of the event, the **Parameter** column shows what kind of data is sent to the frontends, and the **Intent** column explains the usage in plain English. We identified 8 events the backend broadcasts to the frontend.

ID	EventName	Origin	Parameters	Intent
1	CurrentTurn	S	user	Shows which user's turn it is
2	EndGame	S	finalscores	Signals the end of the game
3	ConnectToGame	S	gameId, user	Shows which user connected to the game
4	NewRound	S	roundNumber, marketCards, ships, scores, quarries, sleds	Signals that a new round begins
4	PlayedMove	P	player: userId, move: Move, scores, quarries, sleds	Signals who played what move and the resulting new scores/sled states
5	PickCard	S	user: userId	Signals who should be picking a card from the market

Table 4.2.1.1

*S: System, P: Player*

#### 4.2.2 Interaction Diagram

Figure 4.2.2.1 illustrates the interaction between a user (Client), the WebSocket and the REST interface. The diagram is structured as if the Client is a player who joined an already existing game.

## WebSocket – Interaction Diagram

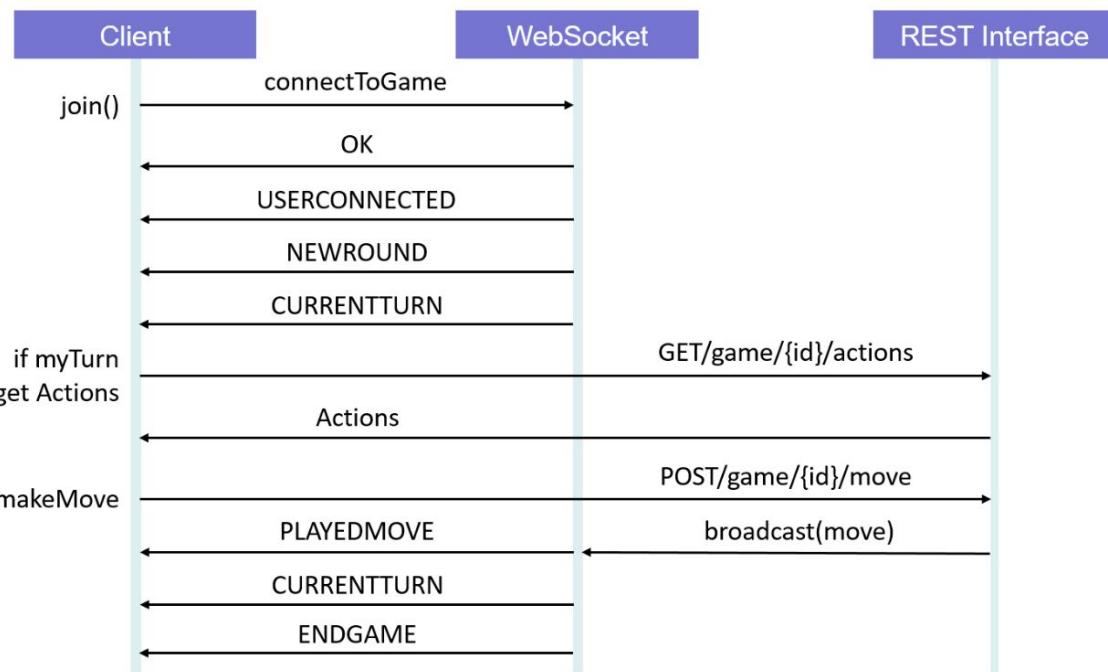


Figure 4.2.2.1

# 5 Risk Analysis

Table 5.1 shows the risk analysis. The first column lists 10 **Identified Risks** which are relevant to our project. The column **Proactive Measures** lists the measures we take to prevent an identified risk event from happening. The green background colour in the table header indicates that these measures are linked to the column **Likelihood of Occurrence**. In case we didn't take any proactive measures the likelihood that a risk event occurs would actually be quite high. But after implementing our proactive measures the likelihood of occurrence can be reduced significantly. The column **Reactive Measures** lists the measures we take to mitigate an event that actually happened. The blue background colour in the table header indicates that these measures are linked to the column **Severity of Hazard**. In real life projects the severity of hazard is an indicator of the financial damage caused by a risk event. In case we didn't take any rational reactive measures the damage of such a risk event to our project would be higher. The assumption behind this statement is that the reactive measures are rational and actually make a net contribution in reducing the severity of hazard. The column **Overall Risk Indicator** shows what would actually be presented to an executive board member. It is the product of likelihood of occurrence and severity of hazard.

Identified Risk	Proactive Measures (prevention)	Reactive Measures (mitigation)	Likelihood of Occurrence (0-100%)	Severity of Hazard (1-10)	Overall Risk Indicator
Lack of knowledge	Reserve time to acquire relevant skills	Increase planned resources	40 %	5	200
Consequences of sickness or accident <i>(if it's the tech savvy team member)</i>	Make everyone replaceable by spreading the knowledge among team members	Reallocate resources and work more	20 %	2 (9)	40 (180)
Consequences of other duties/commitments	Apart from making everyone replaceable also have a good time management	Reallocate resources and work more	30 %	5	150
Consequences hardware defect / theft	Hard drive clones and backup / spreading of data	Acquire and set up a new computer	3 %	3	9
Overestimation of skills (of one team)	Feedback loop, consideration of strengths when	Inform and ask for support	20 %	2	40

member)	allocating task				
Underestimation of tasks	Sensible / knowledgeable and detailed planning	Execute emergency plan	10 %	8	80
Unrealistic goals	Iterative process and project / tech manager who know what's what, good priority allocation	Inform stakeholders and adapt plans	50 %	4	200
Not meeting the deadlines	Good planning, communication and efficient work.	Is it noticed and important, inform the stakeholders otherwise don't mention it	10 %	7	70
Language barrier during presentation	Train the text and be prepared for obvious questions	Keep calm, take time to think or ask a team member for help	10 %	6	60
Technical problems during presentation	Test beamer connection with several computers	Connect another untested computer	0.01 %	10	0.1

Table 5.1

The overall risk indicator shows that even after implementation of proactive and reactive risk management measures, we assume that three identified risks namely: lack of knowledge (200), unrealistic goals (200), and consequences of other duties / commitments (150), remain a significant and damaging risk.

Apart from those three identified risks it is worth mentioning two other problematic risk events: Firstly, the consequences of sickness or accident to our project in case it is Remo, our tech savvy team member. Hardly noticed in case it is anyone else, if he remains absent for a longer period of time, we would have major problems reaching our goals.

Secondly, the risk of technical problems during the presentation and especially connection problems with the beamer, which occur surprisingly often. We assess the likelihood that all our six tested computers fail to connect as very low, but nevertheless, in case every measure fails, the severity of hazard remains devastating. This scenario is what economists call a black swan, something that should never happen, but if it does it has enormous consequences. If we deliver badly documented code or tests that make no sense, it probably goes unnoticed. This is certainly not the case with the presentation and the result would be a failure of the milestone or even the project.

# 6 User Interface

Chapter 6 includes all the user interface mockups. Thus, providing insight into the development process of our UI. Starting with a handmade mockup from milestone 1, ending with screenshots from our finished web application.

## 6.1 User Interface from Milestone 1

### 6.1.1 Login Screen

Figure 6.1.1.1 shows the login screen. This screen/layout is already provided for us. We include it simply for the purpose of completion as well as to illustrate the starting point of our interface tour. A new arrival to the website has to enter his/her name and username and click the 'Login'-button in order to enter the lobby screen of figure 6.1.2.1. The 'Clear' - button clears the name and username fields.

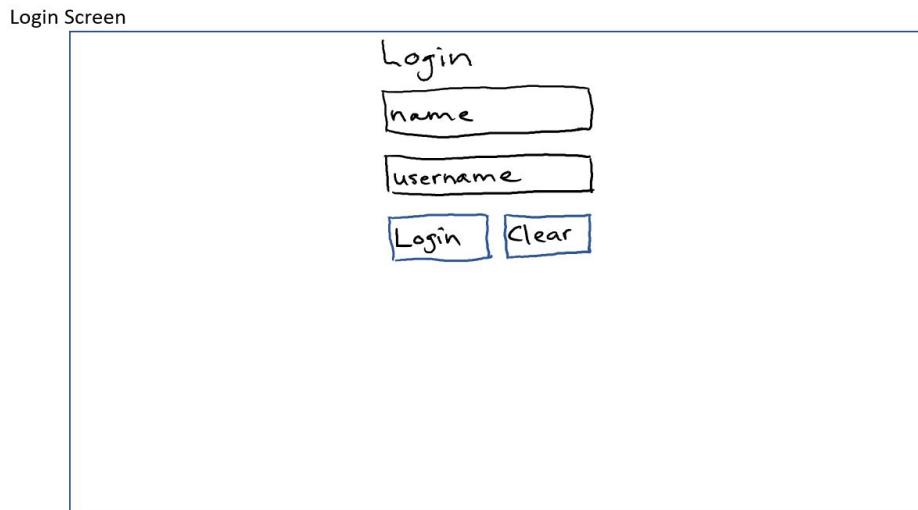


Figure 6.1.1.1

### 6.1.2 Lobby Screen

The next step is the lobby screen shown in figure 6.1.2.1. It shows a list of all the games that are awaiting the required number of players to start the round. As a guest in the lobby you have the option to join a game by clicking on the 'Join' - button or leave the game you already joined by clicking on the 'Leave' - button. The colored game indicates the game you have joined. You also have the option to look at the game instructions by clicking on the 'Manual'-button or create a new game by clicking on the 'New Game'-button. As soon as the required number of players for a game is reached, the game starts and the players can see the In-Game screen of figure 6.1.3.1.

Lobby Screen

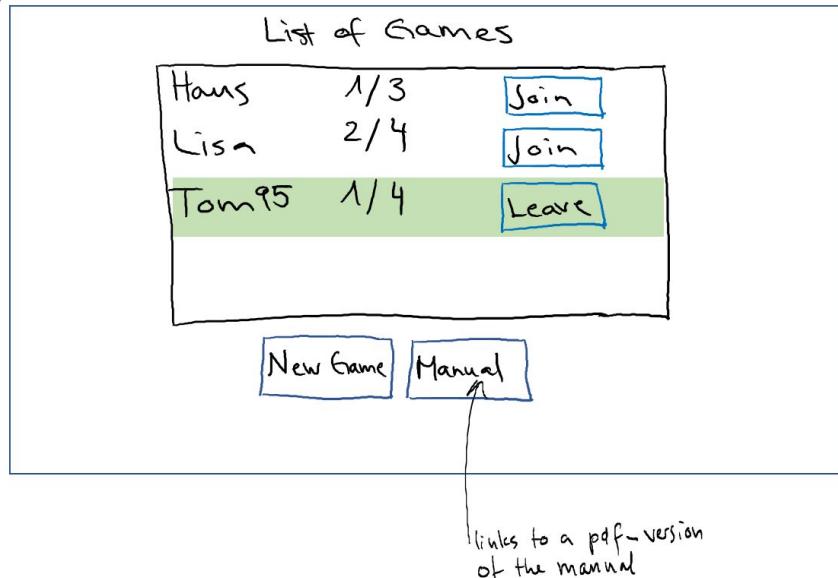


Figure 6.1.2.1

### 6.1.3 New Game Screen

The screen in figure 6.1.3.1 is shown after the guest in the lobby has chosen to create a new game and clicked on the 'New Game'-button on the lobby screen of figure 6.1.2.1. He can then choose the desired number of players for the game and click on the 'Create' - button. This will generate a new game and take the player to the lobby screen of figure 6.1.2.1. Clicking on the 'Back'-button takes the player back to the lobby screen of figure 6.1.2.1 without creating a new game entry.

New Game Screen

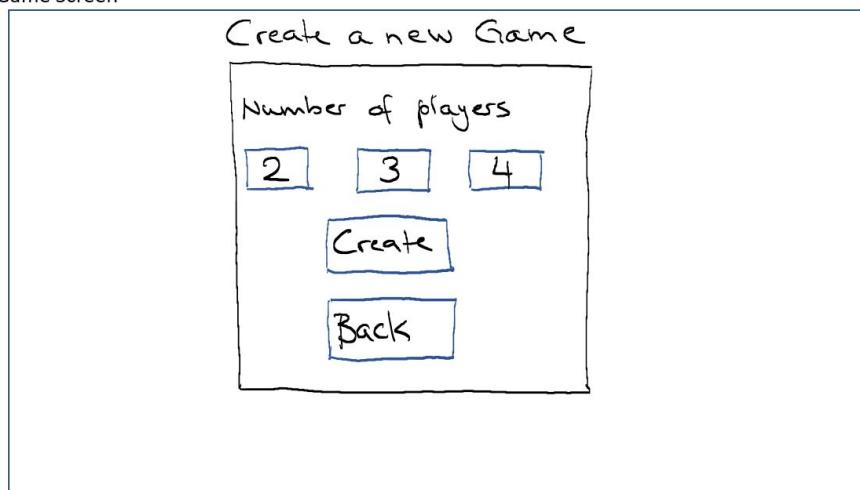


Figure 6.1.3.1

#### 6.1.4 In Game Screen

Now that the game has started, every player can see the in-game screen shown in figure 6.1.4.1. It provides all the necessary game information at a glance: Players can see all their data including username, score, amount of stones in their supply sled and amount of stones left in the stone quarry, as well as the information of all their opponents within the player fields in each corner of the in-game screen are the five site boards, pyramids, temple, burial chamber, obelisk and market. On the left of the site boards is the departing harbor with the ship tokens. On the right of the site boards is the information box used by the system for announcements and offering navigation to game instructions or to leave the game and go back to the lobby seen in figure 6.1.2.1.

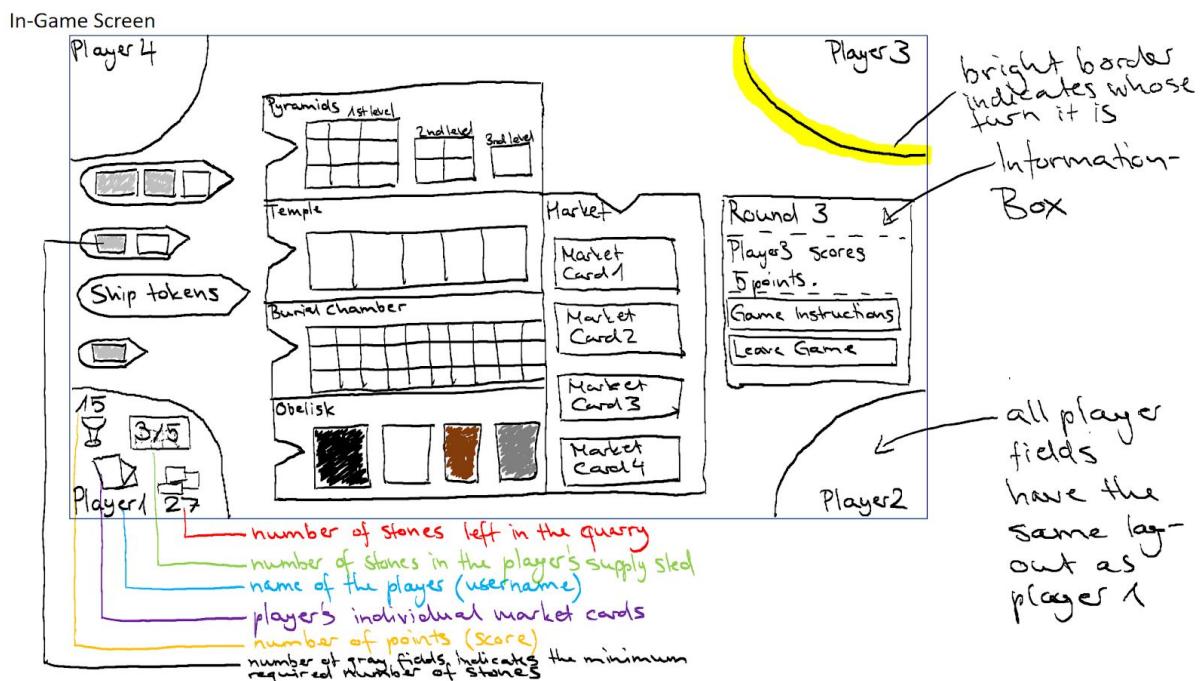


Figure 6.1.4.1

#### 6.1.5 End Game Screen

As soon as the game has ended, an overlay screen, similar to the one shown in figure 6.1.5.1 appears. On this overlay, all the players can see the winner as well as the individual scores for each player. By clicking on the 'Leave'-button, the player is taken back to the lobby screen shown in figure 6.1.2.1.

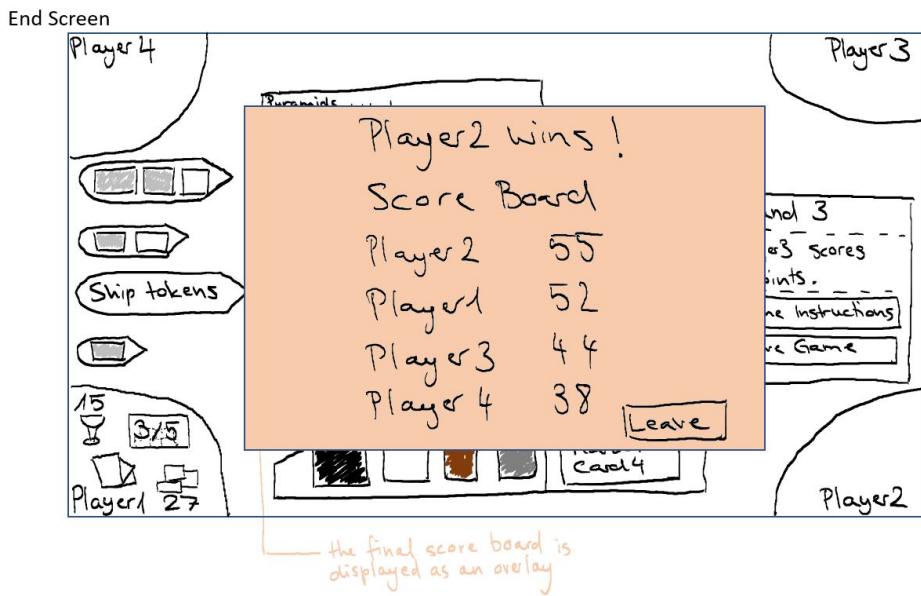


Figure 6.1.5.1

## 6.2 User Interface from Milestone 2

The next five subsections show the mockup for the whole user interface of the future web-application. They are made with Photoshop and show the five different screens a user will come across during the visit of the web-application. First, he will see a **Login Screen**, which gives him access to the **Lobby Screen**, where he is able to initiate a new game in the **New-Game Screen** or join one initiated by another guest in the lobby. From there he is going to access the **In-Game Screen** as soon as the game starts. In case the player stays until the end of the game (s)he is going to see the **End Game Screen**.

### 6.2.1 Login Screen

Figure 6.2.1.1 shows the login screen which is the first thing a visitor to the web-application is going to see. The new arrival to the website has to enter his or her name and click the '**LOGIN**' - button in order to enter the Lobby Screen of figure 6.2.2.1. The '**CLEAR**' - button clears the name field. The web-application is not going to have registration and subsequent authentication by a backend server, i.e. entering a password will not be necessary.

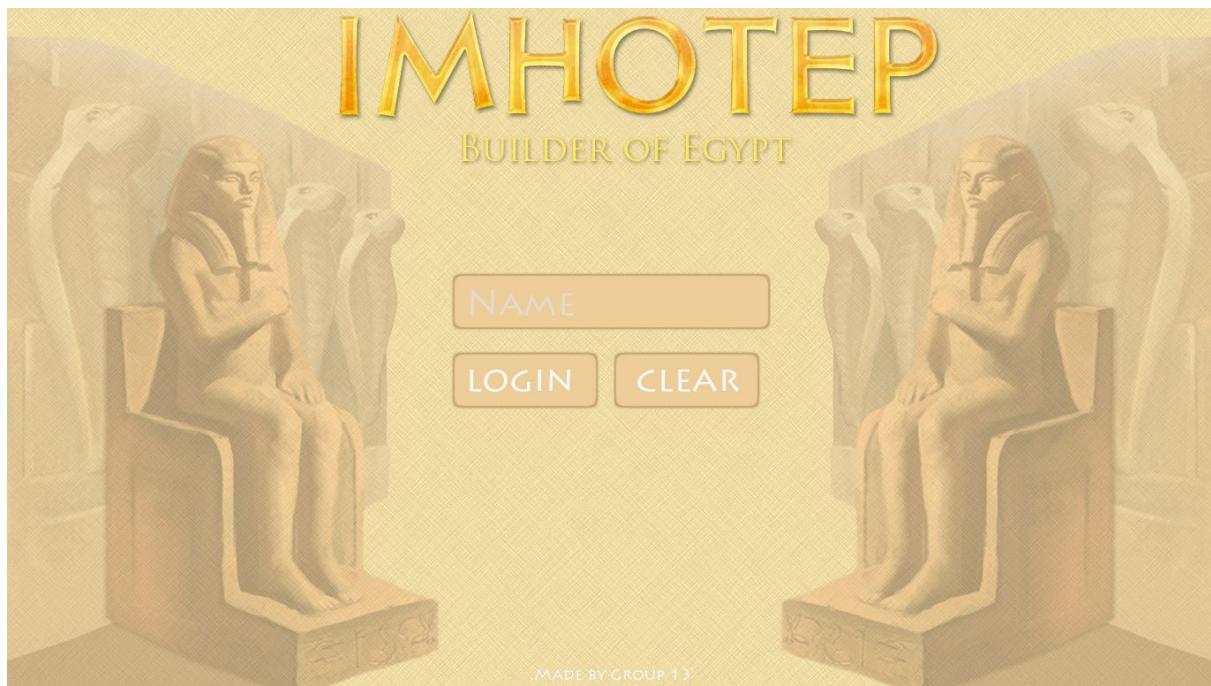


Figure 6.2.1.1

### 6.2.2 Lobby Screen

As soon as the new arrival to the website has passed the Login Screen of figure 6.2.1.1 (s)he gains access to the Lobby Screen shown in figure 6.2.2.1. It shows a list of all the games that are waiting for guests in the lobby to join, as well as those games, which have already started and are no longer open for anybody to join. Each guest in the lobby has the option to either join a game by clicking on the '**JOIN**' - button or leave the game (s)he already joined by clicking on the '**LEAVE**' - button. The highlighted line in the list indicates the game, the guest in the lobby has joined. Each guest also has the option to look at the game instructions by clicking on the '**INSTRUCTIONS**' - Button or to create a new game by clicking on the '**NEW GAME**'-button. By clicking the '**NEW GAME**' - button the guest in the lobby will be routed to the New-Game screen of figure 6.2.3.1. As soon as the required number of players for a game is reached, the game starts and the players can see the In-Game screen of figure 6.2.4.1.

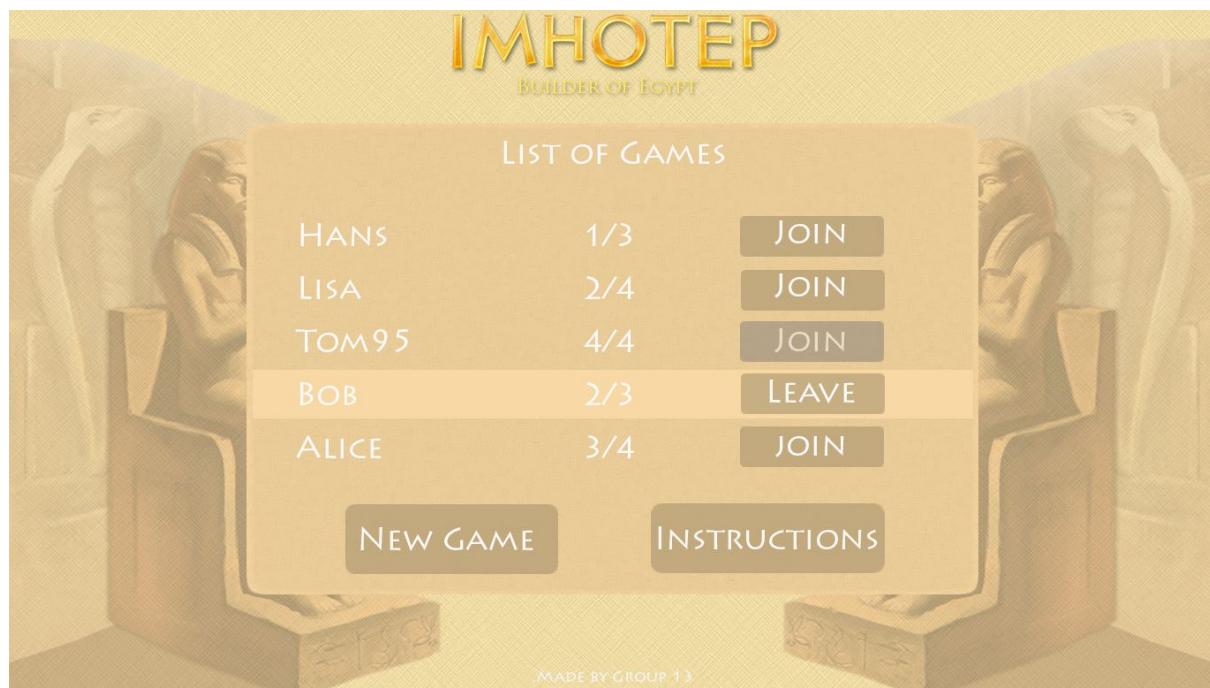


Figure 6.2.2.1

### 6.2.3 New Game Screen

The screen in figure 6.2.3.1 is shown after the guest in the lobby has chosen to create a new game by clicking on the '**NEW GAME**' - **button** in the Lobby Screen of figure 6.2.2.1. S(he) can then choose the desired number of players for the game by clicking on the desired '**2**'- **button**, '**3**'- **button** or '**4**'- **button** respectively and eventually click on the '**CREATE**' - **button**. This will generate a new game entry in the list of the Lobby Screen in figure 6.2.3.1 waiting for other guests to join and is going to take the player back to the Lobby Screen. Clicking on the '**BACK TO LOBBY**' - **button** takes the player back to the Lobby Screen without creating a new game entry.

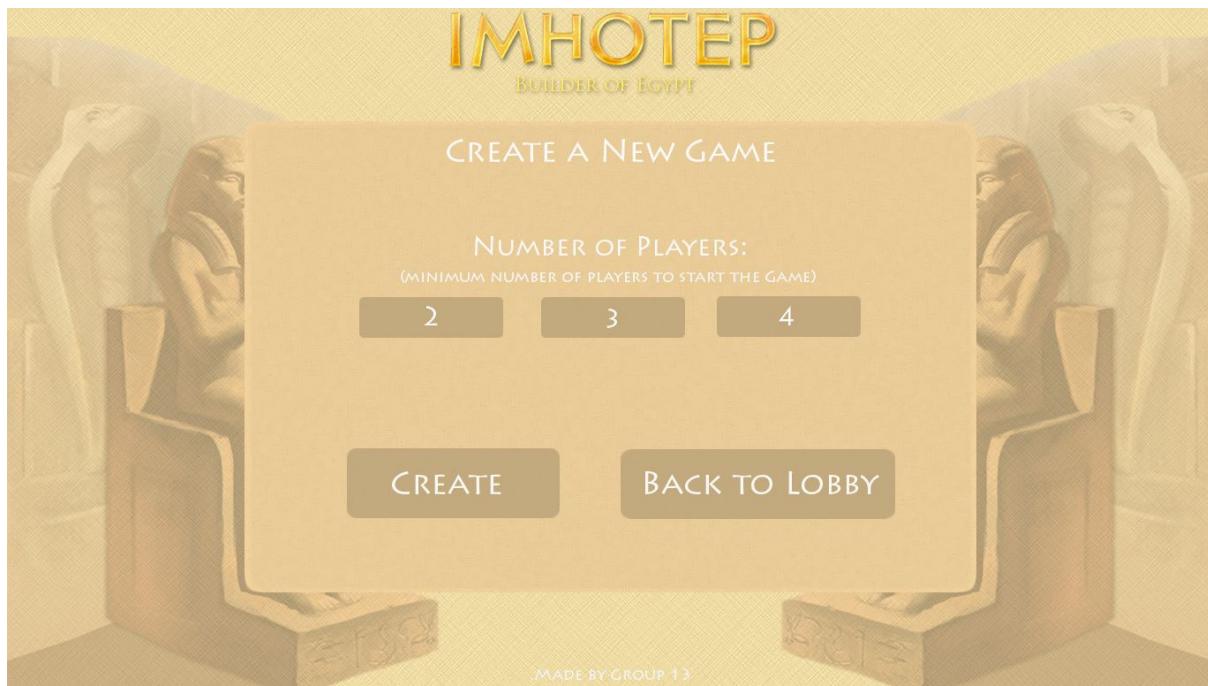


Figure 6.2.3.1

#### 6.2.4 In Game Screen

Now that the game has started, each player can see the In-Game Screen shown in figure 6.2.4.1. At each corner is a **player field**, on the left side in the middle is the **departing harbour** with the ships waiting to be loaded with stones and sailed to the island in the middle of the In-Game screen. The island contains the four site boards, **obelisk**, **burial chamber**, **temple**, **pyramids**, and **market**. On the right side in the middle is the **information box** showing the current round played and two buttons to either get instructions or to leave the game and go back to the Lobby Screen in figure 6.2.2.1.

Each player sees his **player field** on the bottom left corner. And each player field provides all the necessary game information at a glance: username, score, amount of stones in the supply sled, amount of stones left in the stone quarry, as well as information about the fetched green -, blue -, and violet - market cards. As the mouse pointer indicates, hovering over a market card icon opens a tooltip containing the corresponding market card.

The way the player interacts with the game and chooses his actions are described subsequently:

- In case he wants to **move a ship token** from the departing harbour to a free site

board harbour on the island, he has to start a drag-and-drop event on the ship. This causes all free site board harbours on the island to lighten up, as shown in the figure. He can then choose one of these harbours by dragging the ship to it.

- In case he wants to **get new stones** from his stone quarry to his supply sled token, he can click on the stone quarry icon in his player field.
- In case he wants to **place a stone** on a ship token, he can click on the supply sled icon in his player field and then chooses a free spot on a ship token by clicking on it.
- In case he wants to **play a fetched blue market card**, he can click on the corresponding icon in the player field. In case he is unsure about the actions of the blue market card he can hover over the icon and gets a tooltip as shown in the figure.
- In case he has to **choose in which order the stones are unloaded** at the free site board harbour, after he played the blue lever market card, he can click on the stone he want to be unloaded next.
- In case he wants to **fetch a market card** right after a ship token sailed to the market, he can click on the corresponding market card. The market card disappears from the market board and either add it to his player field or in case it is a red market card it starts a background event, which automatically places a stone from his quarry to the market card indicated by the red market card.

Further functionality shown in figure 6.2.4.1 includes the **information on when points are assessed** for each site. This is shown when hovering the mouse over the site board name.

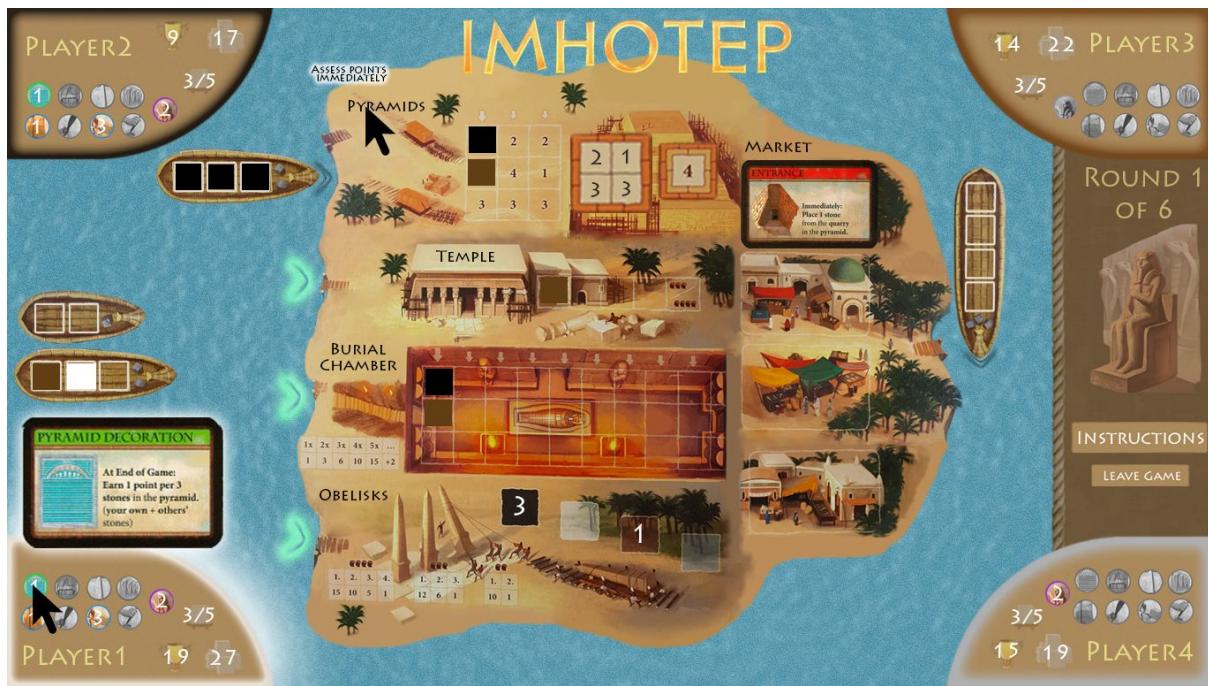


Figure 6.2.4.1

## 6.2.5 End Game Screen

As soon as the game has ended, the overlay screen shown in figure 6.2.5.1 appears. On this overlay, the players are informed about who is the winner as well as the individual scores for each player. By clicking on the '**BACK TO LOBBY**-button, the player is taken back to the Lobby Screen shown in figure 6.2.2.1. In order to leave the game at any time the player has of course also the option to simply close his or her web-browser tab.

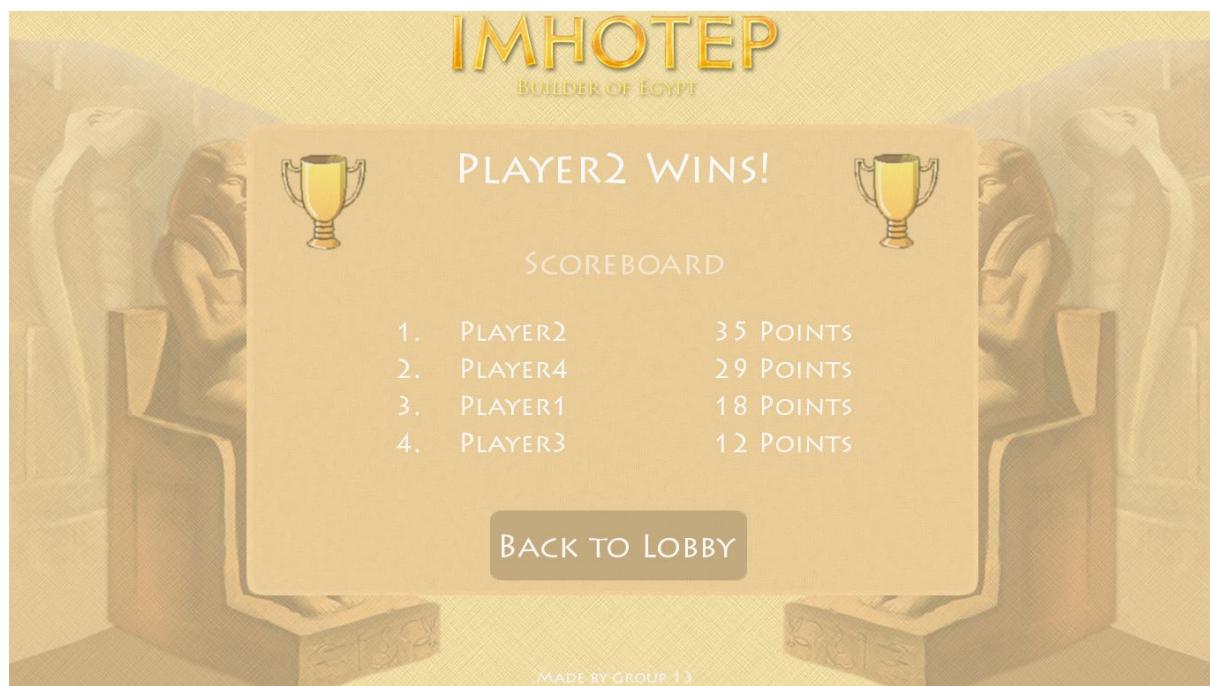


Figure 6.2.5.1

## 6.3 User Interface from Milestone 3

### 6.3.1 Login Screen

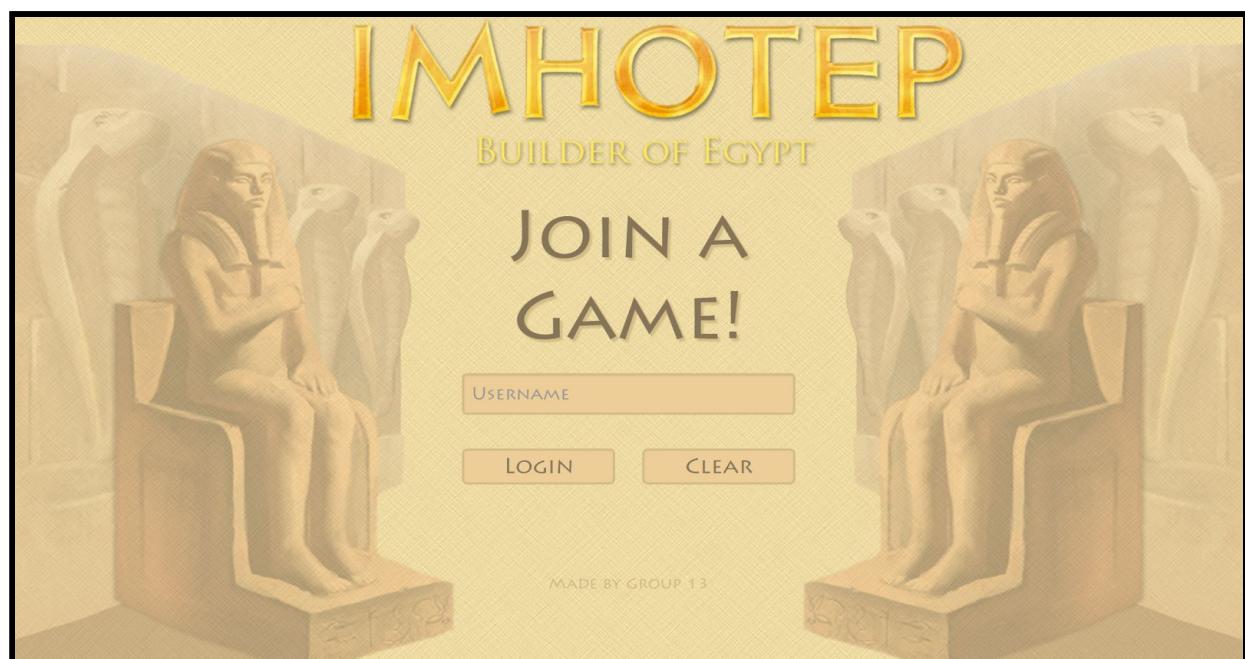


Figure 6.3.1.1

### 6.3.2 Lobby Screen

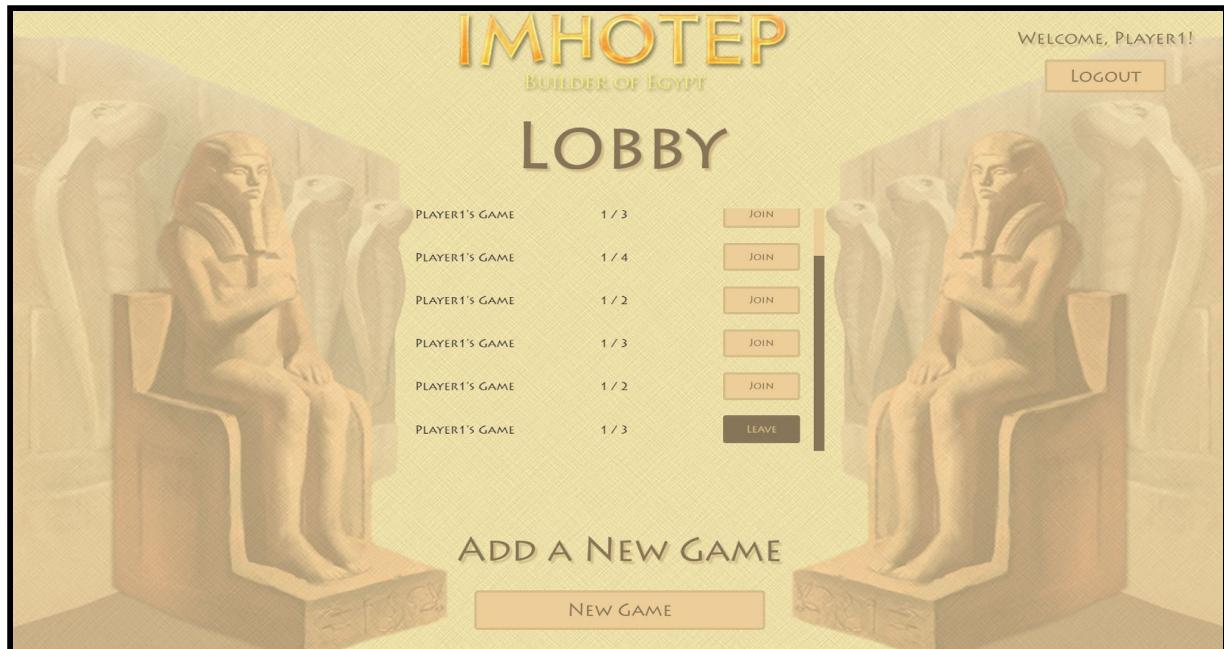


Figure 6.3.2.1

### 6.3.3 New Game Screen



Figure 6.3.3.1

### 6.3.4 In Game Screen

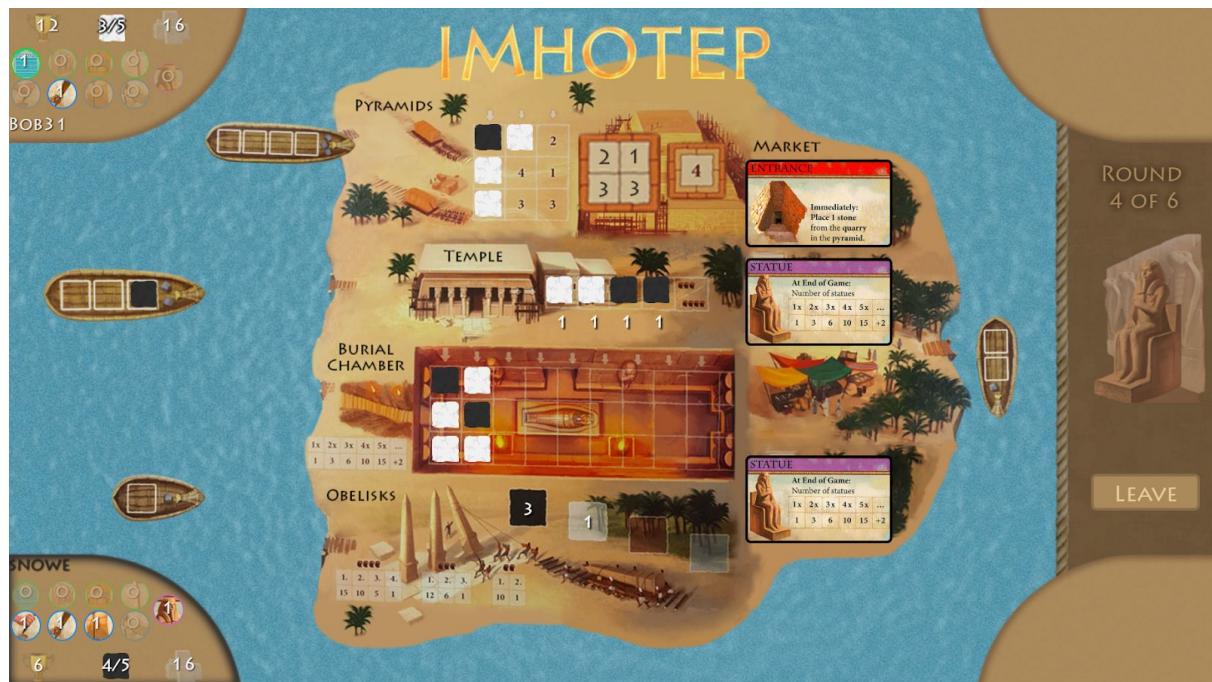


Figure 6.3.4.1

### 6.3.5 End Game Screen

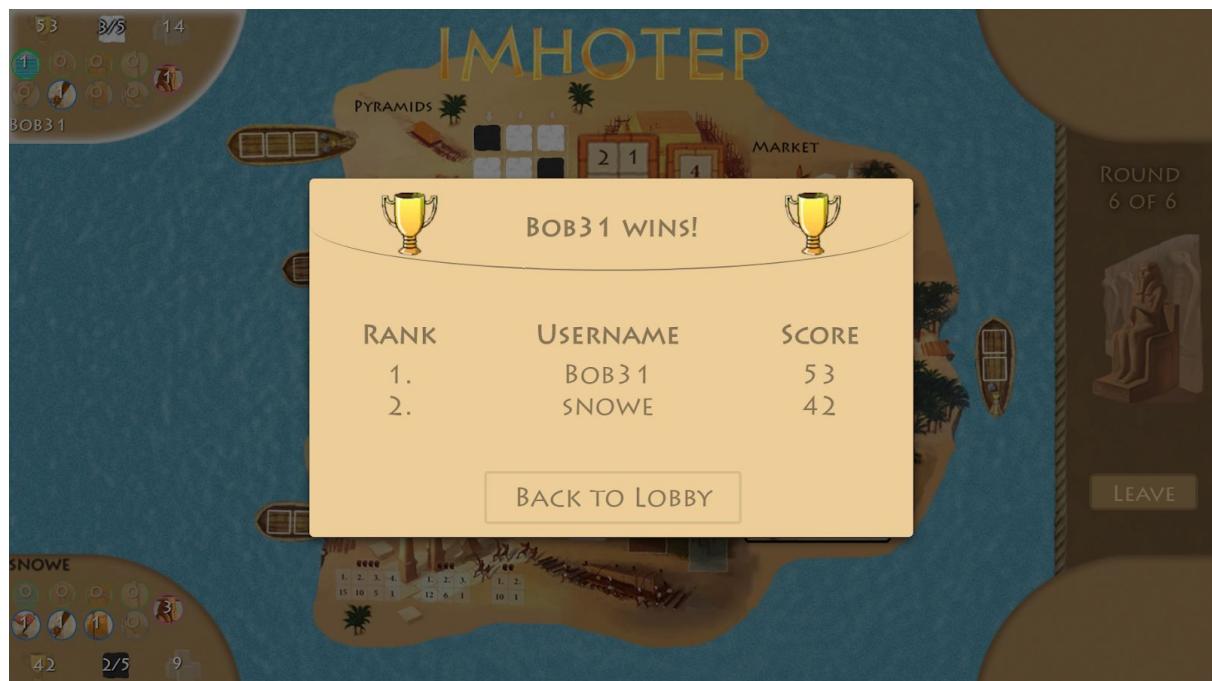


Figure 6.3.5.1

### 6.3.6 Click Sequence Diagrams

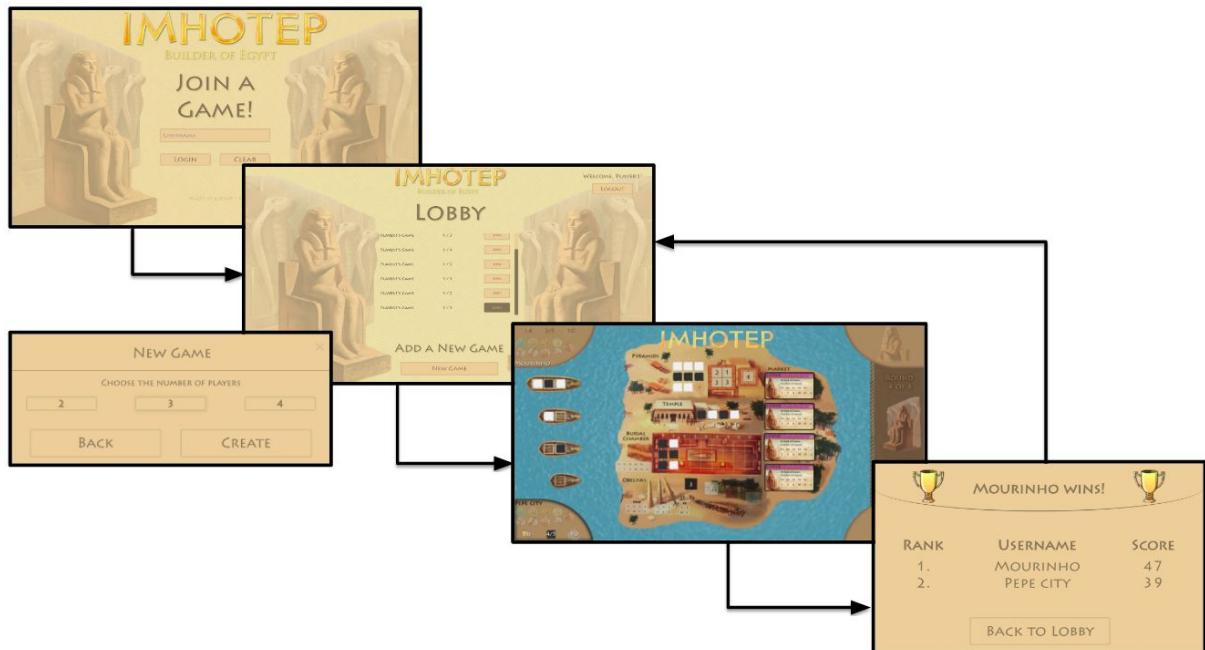
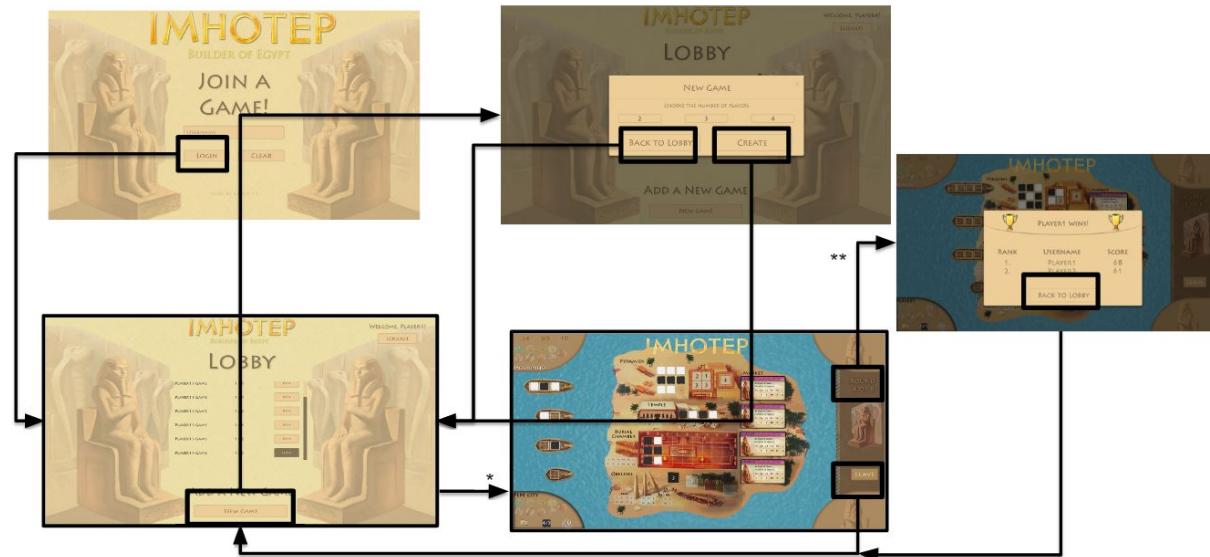


Figure 6.3.6.1



\*Autostart if all players have joined

\*\*Appears when round 6 is finished

Figure 6.3.6.2

## 6.4 User Interface from Milestone 4

### 6.4.1 Login Screen



Figure 6.4.1.1

### 6.4.2 Lobby Screen

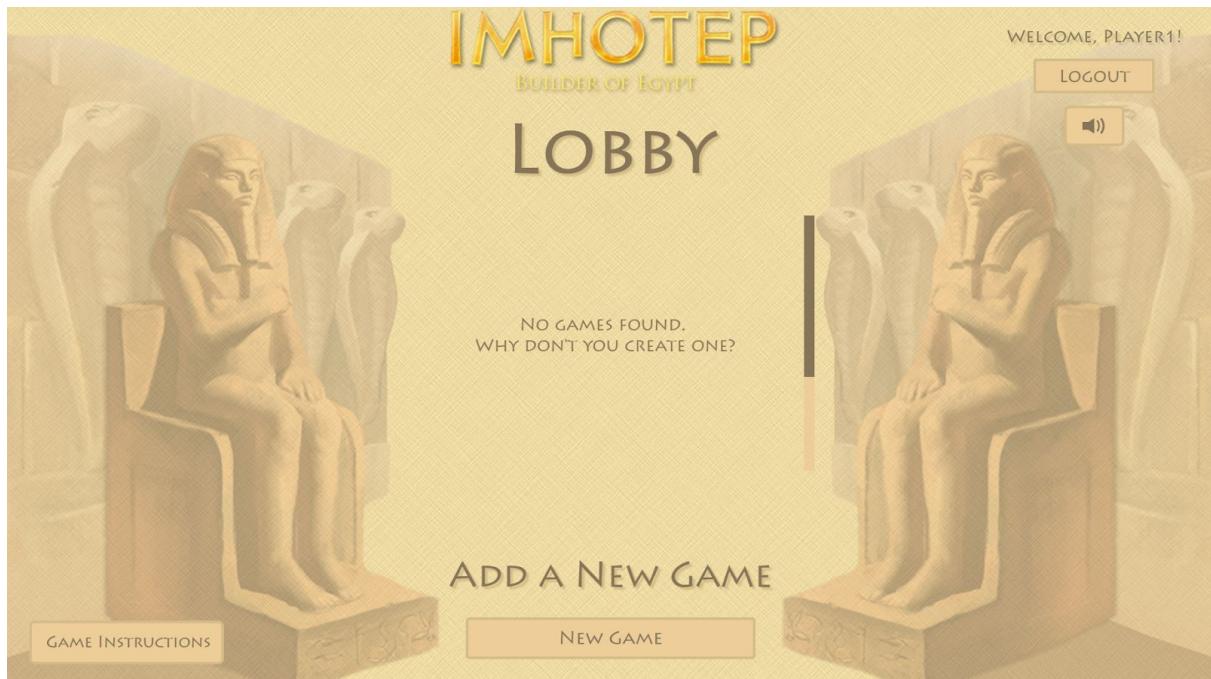


Figure 6.4.2.1

### 6.4.3 New Game Screen

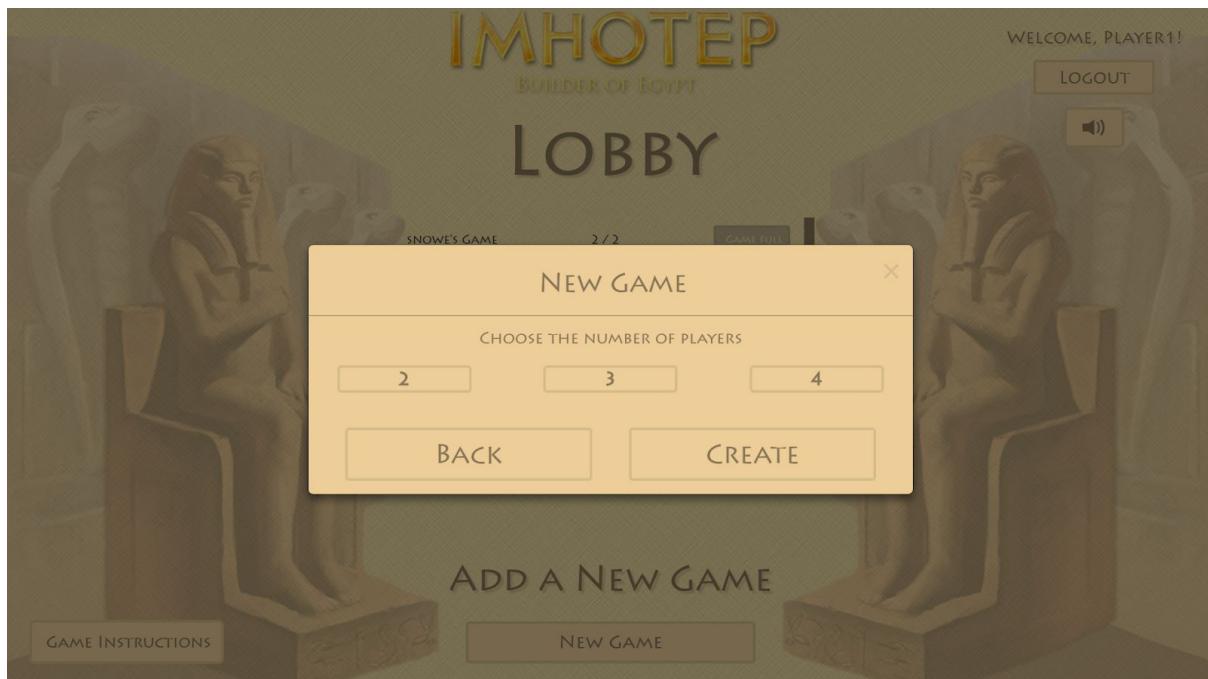


Figure 6.4.3.1

### 6.4.4 In Game Screen

Figure 6.4.4.1 shows the game tutorial. It is being displayed when a game starts to give a player useful information on how to play the game.



Figure 6.4.4.1

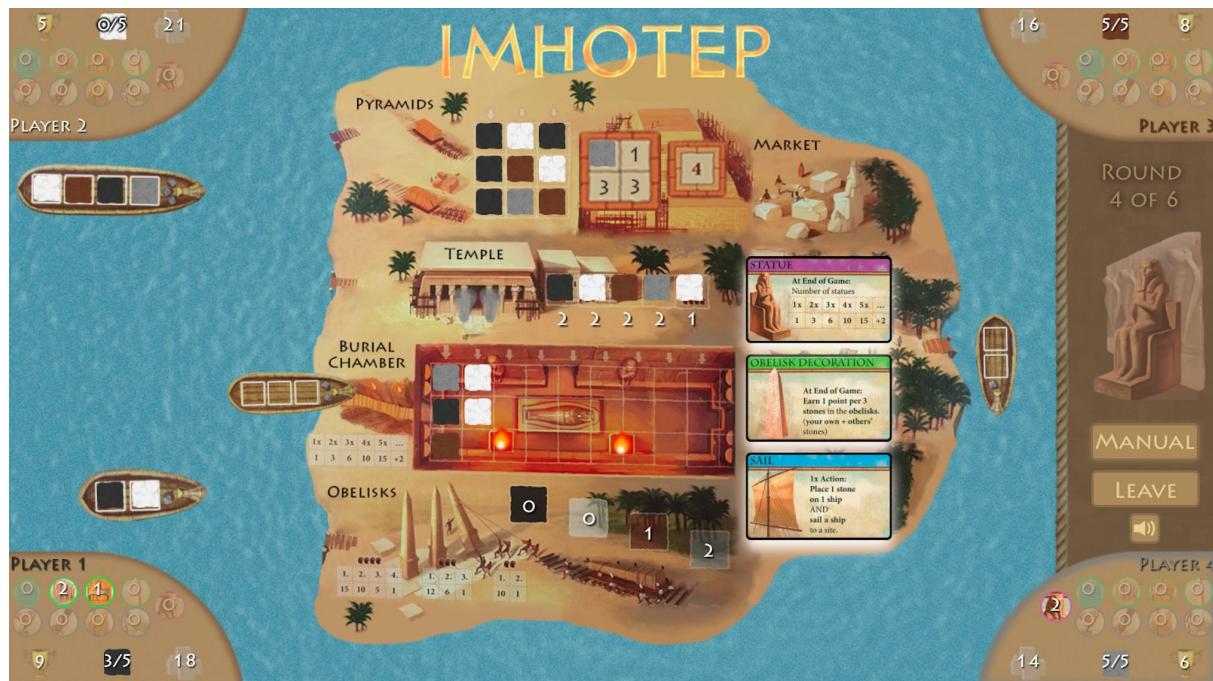


Figure 6.4.4.2

#### 6.4.5 End Game Screen

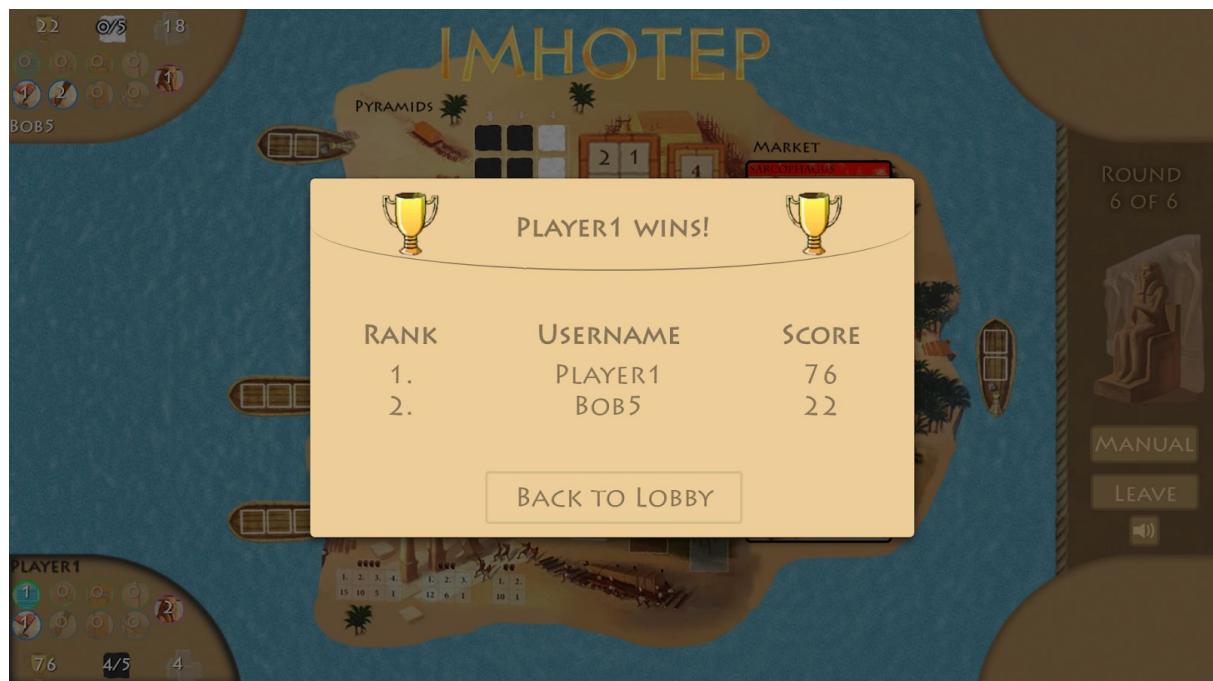


Figure 6.4.5.1

# 7 Time Planning

In this section all the task lists and their corresponding gantt chart(s) are listed. For milestone three there are screenshots of our issue trackers from the client as well as the server side included.

## 7.1 Time Planning of Milestone 1 and Milestone 2

In the first three weeks we had to start the project and set it up for the coming development. Therefore we started to get familiar with the necessary tools, create a risk analysis, establish the first user stories, create the domain model and a first mockup for our web application.

### 7.1.1 Task List until Milestone 2

Table 7.1.1.1 shows a list of tasks up to the official milestone M2. The list starts with some general tasks and milestones, goes on with tasks and milestones for assignment 1 and ends with those for assignment 2. The columns **From** and **To** show the dates between which we plan to work on each task. This information is translated into time bars in the Gantt-Chart of figure 7.1.2.1. The column **Duration** holds the number of man hours we plan to work for each task. Official milestones and sub-milestones have a duration of zero and are marked with MS. The column **Priority** shows for each task whether it is a must, should or could requirement.

Task / Milestone Description	Part of Milestone	From	To	Duration	Priority
Weekly TA Meetings	all	27.2	13.5	MS	must
Milestone meetings and presentations	all	27.2	13.5	MS	must
Installation of software: Node, NPM, IntelliJ, Webstorm, Angular CLI, Git, Dia, Linux-Planner	all	27.2	5.3	20h	must
Make group accounts on GitHub, Heroku, Firebase, Jenkins, LucidCharts	all	11.3	15.3	6h	must
Read tech instructions and tutorials for: Angular 2, HTML, CSS, Spring Boot, HATEOAS principle	all	27.2	17.3	42h	should
Reserve and play the game or watch youtube tutorials, read game instructions	all	27.2	7.3	12h	could
Reserve room at Strickhof for weekly group works	all	3.3	3.3	0.5h	should
Assignment 1					

Write report (A1, 1.1)	M1	4.3	9.3	15h	must
Upload report on OLAT (A1, 1.1)	M1	10.3	10.3	MS	
Make a domain model in UML (A1, 2.1)	M1	2.3	9.3	30h	must
Define user stories (A1, 2.1)	M1	1.3	7.3	15h	must
Make first draft of user interface (mockup on paper) (A1, 2.2)	M1	28.2	7.3	15h	must
Let user interface mockup and other tasks be reviewed by TA (A1, 2.2)	M1	6.3	6.3	6h	should
Make a risk analysis (A1, 2.3)	M1	3.3	5.3	10h	must
Create a project plan until M2 and generate a Gantt-Chart (A1, 2.4)	M1	27.2	5.3	11h	must
Assignment 2					
Write report (A2, 1.1)	M2	11.3	16.3	15h	must
Upload report on OLAT (A2, 1.1)	M2	17.3	17.3	MS	must
Write and train presentation (A2, 1.2)	M1, M2	12.3	16.3	10h	must
Upload presentation on OLAT (A2, 1.2)	M1, M2	17.3	17.3	MS	must
Upload source code on GitHub group account (A2, 1.2)	M2	17.3	17.3	MS	must
Make a project plan until M4 and generate a Gantt-Chart (A2, 2.1)	M2	13.3	15.3	18h	must
Refine and extend user stories (A2, 2.1)	M2	7.3	13.3	20h	must
Create a class diagram (A2, 2.2)	M2	8.3	13.3	25h	must
Create activity diagrams (A2, 2.2)	M2	12.3	14.3	10h	should
Create sequence diagrams (A2, 2.2)	M2	12.3	14.3	10h	should
Specify the REST interface (A2, 2.3)	M2	13.3	16.3	20h	must
Make user interface mockup for whole application on LucidCharts (A2, 2.4)	M2	9.3	13.3	8h	must
Let user interface mockup and other tasks be reviewed by TA (A2, 2.4)	M2	13.3	13.3	6h	should
Clone server and client project templates and learn how to start (A2, 2.5)	M2	11.3	12.3	12h	must

Familiarise with the code of the server and client templates (A2, 2.5)	M2	10.3	16.3	40h	should
Change server and client project templates and test them (A2, 2.5)	M2	13.3	16.3	23h	must

Table 7.1.1.1

## 7.1.2 Gantt Chart until Milestone 2

The Gantt-Chart of figure 7.1.2.1 shows a timeline, starting on February 27th 2017, time bars for each task and an 'x' for each milestone and sub-milestone up to to milestone M2.

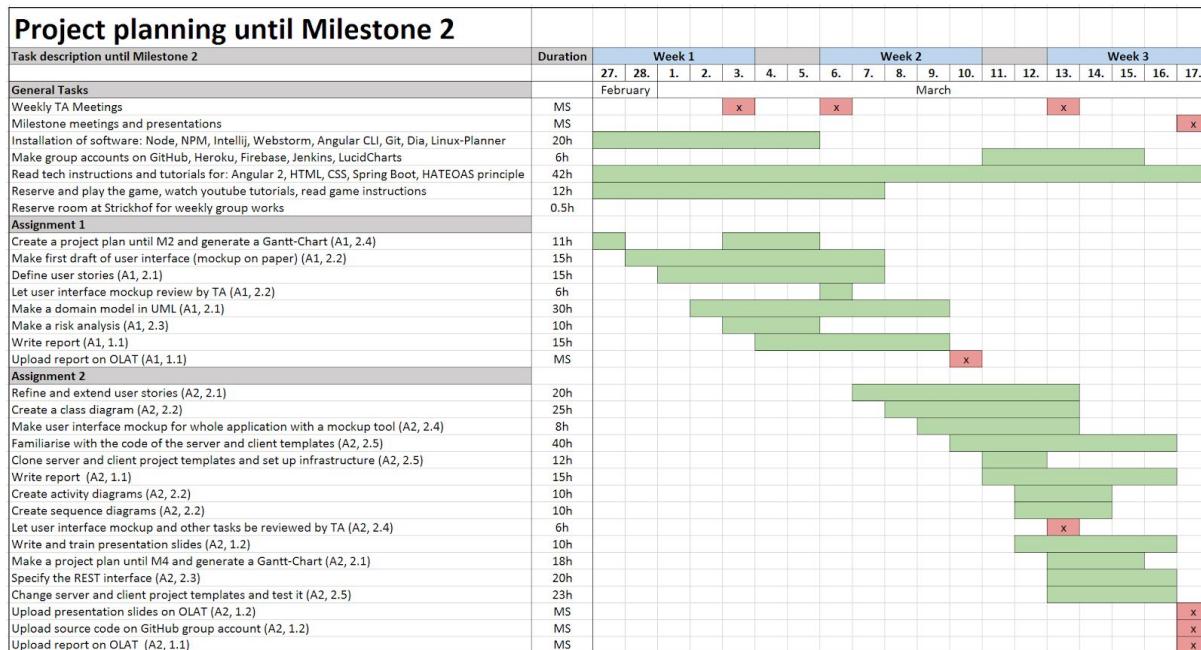


Figure 7.1.2.1

## 7.2 Time Planning from Milestone 2 to Milestone 4

This chapter contains the task list and gantt chart for the implementation part. We wrote down all the tasks necessary until the end of the project. This was mainly for the purpose of overview. We tried to hold on to the created planning as precisely as possible, but we wanted to make sure to be flexible for upcoming changes and restructures.

### 7.2.1 Task List from Milestone 2 to Milestone 4

Table 7.2.1.1 shows a list of tasks from official milestone M2 to M4. It is divided into our eight sprints following the official milestone M2.

The columns **From** and **To** show the dates between which we plan to work on each task. This information is translated into **time bars** in the Gantt-Chart of figure 7.2.2.1. The column

**Duration** holds the number of effective hours we plan to work for each task. Official milestones and sub-milestones have a duration of zero and are marked with MS. The column **Priority** shows for each task whether it is a must, should or could requirement.

Task / Milestone Description	Team	From	To	Duration	Priority
<b>Sprint 1</b>					
TA meeting, discussion about Milestone 3	MS	20.3	20.3	MS	Must
Implement login UI components incl. actions/events	Client	20.3	21.3	10h	Must
Implement login server logic	Server	20.3	21.3	2h	Must
Visualize lobby UI	Client	21.3	22.3	5h	Must
Implement lobby UI components incl. actions/events	Client	21.3	23.3	15h	Must
Implement lobby server logic	Server	21.3	23.3	15h	Must
Implement in-game turn logic	Server	22.3	26.3	5h	Must
Define and visualize game board	Client	22.3	26.3	20h	Must
Integration testing (login functionality, lobby)	both	24.3	26.3	10h	Must
TA meeting, review sprint 1, preparation sprint 2	MS	27.3	27.3	MS	Must
<b>Sprint 2</b>					
Visualize ship placement	Client	27.3	28.3	3h	Must
Display Obelisks site	Client	27.3	28.3	3h	Must
Implement ship server logic	Server	27.3	31.3	15h	Must
Visualize ship movement in UI	Client	28.3	30.3	5h	Must

Display player dashboard (without cards)	Client	28.3	30.3	10h	Must
Implement stone to ship placement	Client	28.3	30.3	15h	Must
Implement stone from ship to site placement	Client	29.3	1.4	10h	Must
Implement Obelisks logic	Server	30.3	2.4	15h	Must
Display stones on Obelisks	Client	31.3	2.4	3h	Must
Implement scoring for Obelisks site	Server	1.4	2.4	7h	Must
Integration testing (Stone to Ship, Ship to Obelisks)	both	31.3	2.4	10h	Must
TA meeting, review sprint 2, sprint 3 preparation	MS	3.4	3.4	MS	Must
<b>Sprint 3</b>					
Display Market site	Client	3.4	4.4	3h	Must
Implement Market logic	Server	3.4	5.4	15h	Must
Display cards on Market site	Client	4.4	6.4	3h	Must
Implement handing out of market cards	Server	4.4	7.4	15h	Must
Implement red/blue card actions/events	Client	5.4	8.4	20h	Must
Server logic for playing cards	Server	5.4	9.4	20h	Must
Visualize market cards on player dashboard	Client	6.4	9.4	7h	Must
Integration testing (pick-up card, play card)	both	7.4	9.4	10h	Must
TA meeting, review sprint 3, sprint 4 preparation	MS	10.4	10.4	MS	Must

Sprint 4					
Display Temple site	Client	10.4	11.4	3h	Must
Implement Temple logic	Server	10.4	12.4	15h	Must
Display stones on Temple site	Client	11.4	12.4	3h	Must
Implement scoring for Temple site	Server	12.4	13.4	7h	Must
Display Pyramids site	Client	12.4	13.4	3h	Must
Implement Pyramids logic	Server	13.4	14.4	15h	Must
Display stones on Pyramids site	Client	13.4	14.4	5h	Must
Implement scoring for Pyramids site	Server	13.4	14.4	10h	Must
Visualize game menu	Client	14.4	15.4	3h	Must
Implement game menu actions/events	Client	14.4	16.4	10h	Must
Leave game logic	Server	14.4	16.4	15h	Must
Integration testing (placing stones, game menu)	both	14.4	16.4	10h	Must
TA meeting, review sprint 4, sprint 5 preparation	MS	17.4	17.4	MS	Must
Sprint 5					
Display Burial Chamber site	Client	17.4	18.4	3h	Must
Implement Burial Chamber logic	Server	17.4	19.4	15h	Must
Display stones on Burial Chamber site	Client	18.4	20.4	4h	Must
Implement scoring for Burial Chamber site	Server	18.4	20.4	7h	Must
Implement active player effects and available actions	Client	20.4	23.4	30h	Must

Implement active move effects	Client	20.4	23.4	30h	Must
Integration testing (placing stones, effects)	both	21.4	23.4	10h	Must
TA meeting, review sprint 5, sprint 6 preparation	MS	24.4	24.4	MS	Must
<b>Sprint 6</b>					
Write report for Milestone 3	both	24.4	27.4	20h	Must
Integration testing (game components, game flow)	Client	24.4	27.4	25h	Must
Test coverage of at least 50%	Server	24.4	27.4	15h	Must
Write presentation slides for Milestone 3	both	25.4	27.4	20h	Must
Prepare presentation for Milestone 3	both	26.4	27.4	8h	Must
Presentation Milestone 3	MS	28.4	28.4	MS	Must
<b>Sprint 7</b>					
TA meeting, review Milestone 3, Milestone 4 preparation	MS	1.5	1.5	MS	Must
Visualize final scoreboard	Client	29.4	30.4	5h	Must
Implement final scoreboard	Server	29.4	1.5	20h	Must
Buffer for unfinished tasks during Milestone 3	both	1.5	7.5	50h	Must
Optional: Implement hand-picked features (if enough time)	both	1.5	7.5	20h	Should/ Could
Optional: Stress testing UI with random clicking	Client	5.5	7.5	5h	Could
Integration testing (scoreboard)	both	5.5	7.5	10h	Must

TA meeting, review sprint 7, sprint 8 preparation	MS	8.5	8.5	MS	Must
<b>Sprint 8</b>					
Write report for Milestone 4	both	8.5	11.5	20h	Must
Integration testing (game components, game flow)	Client	8.5	11.5	15h	Must
Test coverage of at least 75%	Server	8.5	11.5	15h	Must
Write presentation slides for Milestone 4	both	9.5	11.5	20h	Must
Prepare presentation for Milestone 4	both	10.5	11.5	8h	Must
Presentation Milestone 4	MS	12.5	12.5	MS	Must

Table 7.2.1.1

## 7.2.2 Gantt Chart from Milestone 2 to Milestone 4

The Gantt-Chart of figures 7.2.2.1 and 7.2.2.2 show a timeline, starting on March 20th 2017, time bars for each task and an 'x' for each milestone and sub-milestone up to milestone M3.

The Gantt-Chart of figure 7.2.2.3 shows a timeline, starting on April 29th 2017, time bars for each task and an 'x' for each milestone and sub-milestone up to milestone M4, the end of the project.

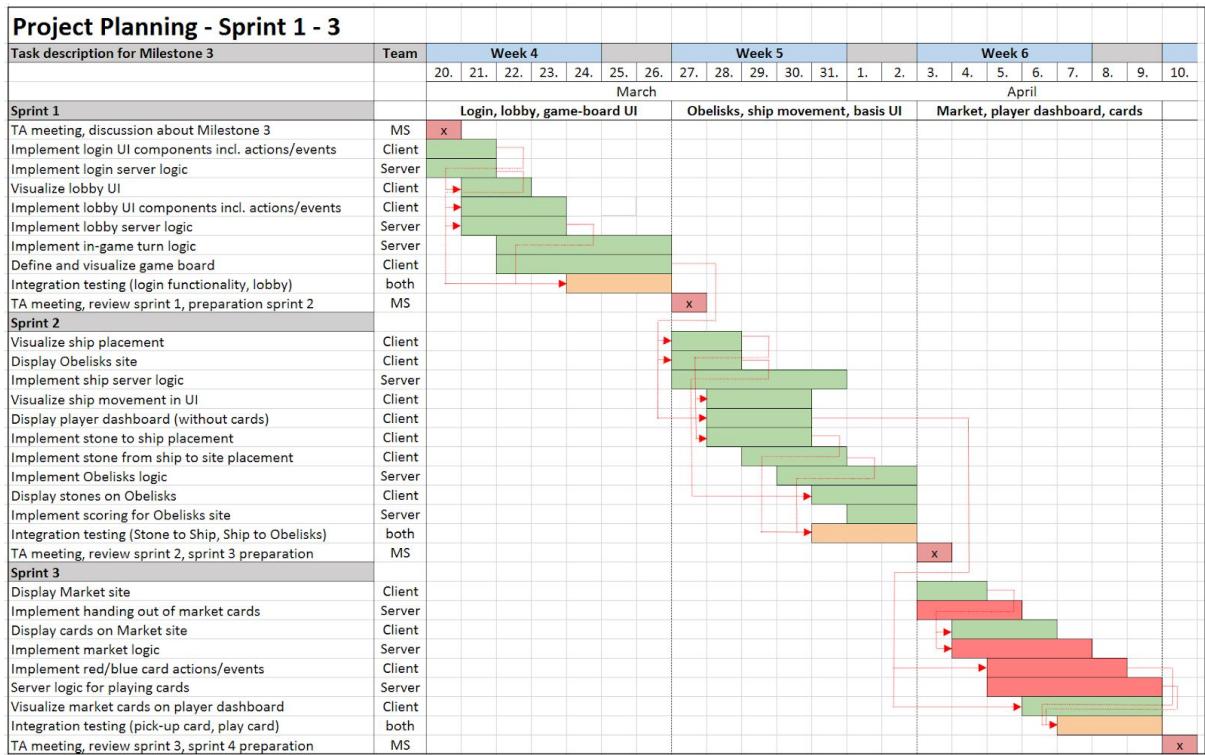


Figure 7.2.2.1

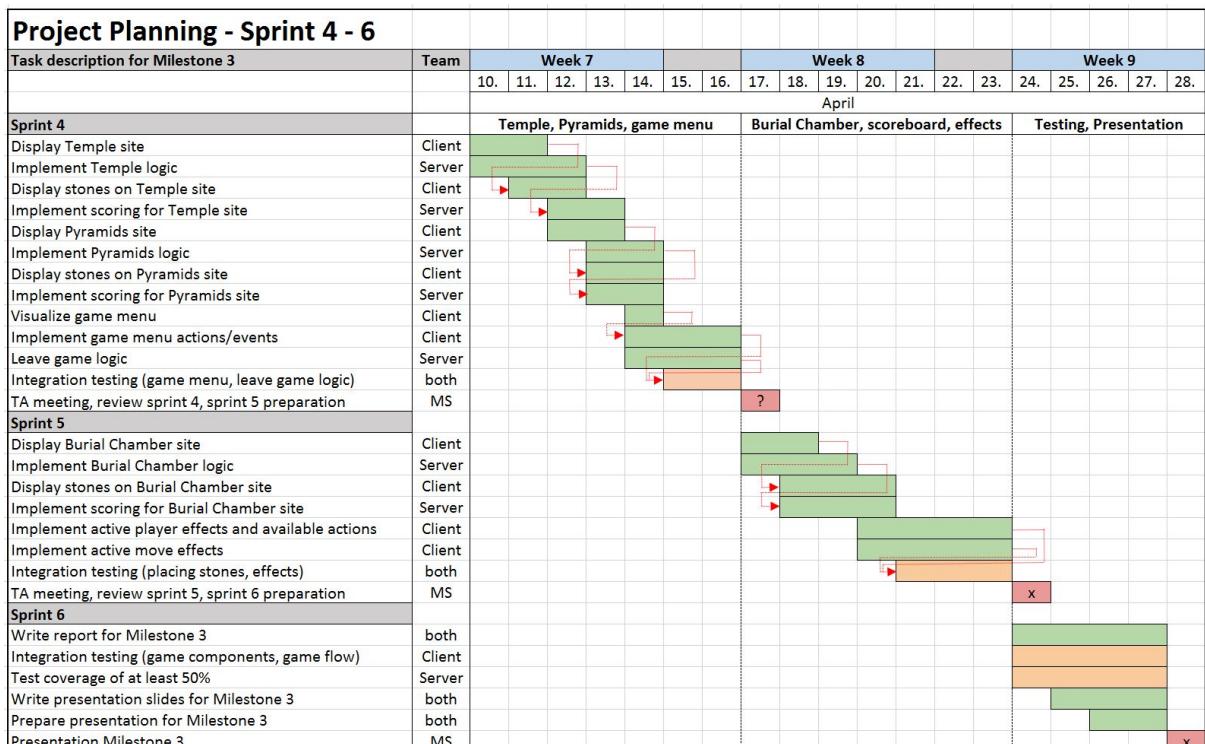


Figure 7.2.2.2

Project Planning - Sprint 7 - 8		Team	Week 10							Week 11						
			29.	30.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
		April							May							
Sprint 7			Test coverage, buffer, should & could features										Presentation, report			
TA meeting, review Milestone 3, Milestone 4 preparation	MS			x												
Visualize final scoreboard	Client															
Implement final scoreboard	Server															
Buffer for unfinished tasks during Milestone 3	both															
Optional: Implement hand-picked features (if enough time)	both															
Optional: Stress testing UI with random clicking	Client															
Integration testing (scoreboard)	both															
TA meeting, review sprint 7, sprint 8 preparation	MS											x				
Sprint 8																
Write report for Milestone 4	both															
Integration testing (game components, game flow)	Client															
Test coverage of at least 75%	Server															
Write presentation slides for Milestone 4	both															
Prepare presentation for Milestone 4	both															
Presentation Milestone 4	MS														x	

Figure 7.2.2.3

## 7.3 Redefined Time Planning for Milestone 4

Due to restructure and general changes concerning our time planning, we had to redefine the already created time planning, especially the sprint seven and eight. Some tasks were already finished (e.g. the scoreboard), whereas others had to be postponed (some of the market site logic). For this reason we highlighted the tasks we had to finish during the last milestone with a red color in the gantt chart from sprint one to three. (Figure 7.2.2.1)

We came across some issues while implementing the drag and drop functionality. In the beginning of Milestone 3, we decided to use the dragula-functionality, in order to create the drag and drop as we imagined it. For several weeks we used and expanded it until it became a huge source of buggy code with lots of work-arounds. At some point we weren't able to continue implementing important functions, because dragula wasn't working properly. First, we tried to restructure and rewrite the depending functions. After wasting too much time, it left us with no other choice but cancelling dragula out of our implementation. Therefore, we had to search for alternatives. Obviously, this led to some chaos in our time planning. For this reason, we decided to postpone some of the implementation tasks of the market site (Table 7.3.1.1 & figure 7.3.2.1) and use the gained time to restructure the drag and drop depending features.

### 7.3.1 Redefined Task List for Milestone 4

Tasks for Milestone 4	Team	From	To	Duration
<b>Sprint 7</b>				
TA meeting, review MS 3, preparation MS 4	MS	1.5	1.5	MS
Implement pick-up of market cards	Server	29.4	30.4	15h
Implement blue market card actions	Client	29.4	1.5	20h
Server logic for playing blue cards	Server	30.4	2.5	20h
Implement stone from ship to site placement in chosen order (lever)	Client	30.4	2.5	15h
Minor styling improvements	Client	5.5	7.5	10h
Test all possible moves	both	2.5	5.5	3h
Debug detected errors	both	2.5	7.5	5h
Integration Testing (market site/cards)	both	5.5	7.5	5h
TA meeting, review sprint 7, sprint 8 preparation	MS	8.5	8.5	MS
<b>Sprint 8</b>				
Write report for Milestone 4	both	8.5	11.5	20h
Integration testing (game flow)	Client	5.5	7.5	15h
Test coverage of at least 75%	Server	8.5	11.1	15h
Write presentation slides for Milestone 4	both	9.5	11.1	10h
Prepare presentation Milestone 4	both	10.5	11.5	8h
Presentation Milestone 4	MS	12.5	12.5	MS

Table 7.3.1.1

### 7.3.2 Redefined Gantt Chart for Milestone 4

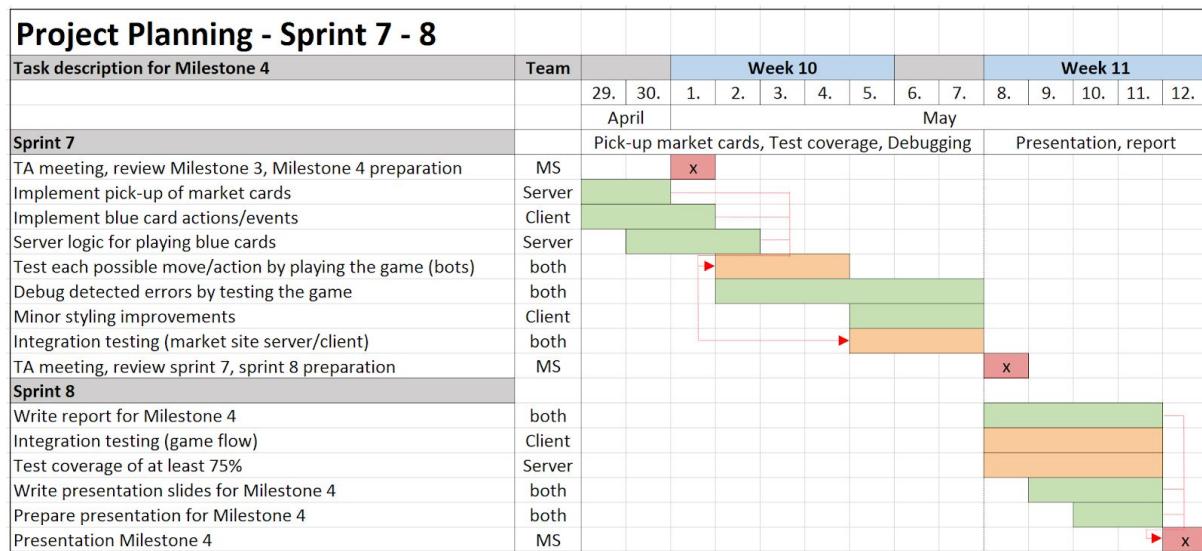


Figure 7.3.2.1

### 7.3.3 Issue Tracker Client

This section contains all of our issues on the client side. The issues are the broken down version of the tasks featured in the task list for the fourth milestone (Table 7.3.1.1) and the gantt chart (Figure 7.3.2.1)

## Closed issues:

<input type="checkbox"/>	⌚ fix playing 2 blue market cards at once	bug	in progress	
	#120 by Brudi8 was closed 2 hours ago	 Milestone 4		
<input type="checkbox"/>	⌚ In Lobby: change bot game buttons	styling		
	#119 by dawwys was closed a minute ago	 Milestone 4		
<input type="checkbox"/>	⌚ Add mute-music-button to the lobby	styling		
	#118 by dawwys was closed a minute ago	 Milestone 4		
<input type="checkbox"/>	⌚ Property 'players' does not exist on type 'TopRightComponent'.		in progress	
	#117 by nairboon was closed 2 hours ago	 Milestone 4		
<input type="checkbox"/>	⌚ fix stone visualization over fire animation for burial chamber	bug	in progress	
	#116 by Brudi8 was closed 2 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ do sail card check for all players	bug		
	#115 by dawwys was closed a day ago			
<input type="checkbox"/>	⌚ Implement game manual buttons and modals on all screens		in progress	
	#114 by Brudi8 was closed 3 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ fix visualization of all modals (leave, scoreboard, new game, game instructions)	bug	in progress	
	styling			
	#113 by Brudi8 was closed 3 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ Styling for leave, manual and music pause button	styling		
	#112 by dawwys was closed 2 days ago			
<input type="checkbox"/>	⌚ when sail card has been played, make player only able to place stone on ships that can sail afterwards	bug		
	#111 by dawwys was closed 2 days ago			
<input type="checkbox"/>	⌚ fix hammer card styling	in progress	styling	
	#110 by Brudi8 was closed 3 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ 1-Bot-Game Market Bug	bug	in progress	
	#109 by spazzyt was closed a minute ago	 Milestone 4		
<input type="checkbox"/>	⌚ Can't play lever card - says "You can't play sail card" xD	bug	in progress	
	#108 by Brudi8 was closed 2 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ implement game manual modal	in progress	styling	
	#107 by Brudi8 was closed 7 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ fix popovers for top-right players	bug	styling	
	#106 by Brudi8 was closed 7 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ Frontend crashes if red card is picked			
	#105 by nairboon was closed 3 days ago	 Milestone 4		
<input type="checkbox"/>	⌚ verify acceptance of all 'must' user stories			
	#104 by nairboon was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ number on quarry does not go to 0 when taking stones with 1 left	bug	in progress	
	#103 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ prepare presentation	Presentation		
	#102 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ write presentation slides	Presentation		
	#101 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ write report for milestone 4	Report		
	#100 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ test game flow	testing		
	#99 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ testing market site and cards	testing		
	#98 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ minor styling improvements	in progress	styling	
	#97 by snoweZz was closed 2 minutes ago	 Milestone 4		
<input type="checkbox"/>	⌚ debug detected errors by playing the game (WRITE ISSUES!)	bug		
	#96 by snoweZz was closed 2 minutes ago	 Milestone 4		

<input type="checkbox"/>	<b>test all possible moves</b>	<span>in progress</span>	<span>testing</span>	#95 by snoweZz was closed 3 minutes ago	Milestone 4	
<input type="checkbox"/>	<b>implement blue card actions</b>	<span>in progress</span>		#94 by snoweZz was closed 3 minutes ago	Milestone 4	
<input type="checkbox"/>	<b>12 Open ✓ 91 Closed</b>				Author ▾ Labels ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾	
<input type="checkbox"/>	<b>define arriving harbor</b>			#93 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>defined departing harbour</b>			#92 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>create demo buttons in In-Game screen for testing functions</b>			#91 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Implement point assessment hover for pyramid site</b>			#90 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>style market card icons in playerfield</b>			#89 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>add quarry stones and icon to playerfield</b>			#88 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>add stone sled number and icon to playerfield</b>			#87 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>add score for player in playerfield</b>			#86 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>implement active and inactive playerfield effect</b>			#85 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>add active and inactive functions for market card icons in playerfield</b>			#84 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>add counter for every market card icon in playerfield</b>			#83 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>implement clickfunction for blue market card icons</b>			#82 by Brudi8 was closed 3 days ago	Milestone 3	

<input type="checkbox"/>	<b>⌚ implement hover function for card icons</b>	#81 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ define order of card icons in playerfield</b>	#80 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ display playernames in playerfield</b>	#79 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ get playernames from backend</b>	#78 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ general player field styling</b>	#77 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ define playerfield structure</b>	#76 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ logic for displaying stones on burial chamber site</b>	#75 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ div structure for burial chamber</b>	#74 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ Implement router functionality to lobby screen for leave game button</b>	#73 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ visualize stone placement on burial chamber</b>	#72 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ add point assessment hover for burial chamber</b>	#71 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ display burial chamber site</b>	#70 by Brudi8 was closed 3 days ago		
<input type="checkbox"/>	<b>⌚ receive round information from backend</b>	#69 by Brudi8 was closed 3 days ago		

<input type="checkbox"/>	<b>⌚ 12 Open ✓ 91 Closed</b>	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	<b>⌚ implement leave game button</b>	#68 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ style game menu</b>	#67 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ Visualize game menu</b>	#66 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ implemented continuous stone placement on pyramid site</b>	#65 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ add slot ids in ships for stones <span style="background-color: #d9e1f2; border: 1px solid #2e3436; padding: 2px;">in progress</span></b>	#64 by Brudi8 was closed a day ago					
<input type="checkbox"/>	<b>⌚ implement stone from ship to site logic for temple site(displaying stones in corresponding order) <span style="background-color: #d9e1f2; border: 1px solid #2e3436; padding: 2px;">in progress</span></b>	#63 by Brudi8 was closed a day ago					
<input type="checkbox"/>	<b>⌚ implement stone from ship to site logic for pyramid site (displaying stones in corresponding order) <span style="background-color: #d9e1f2; border: 1px solid #2e3436; padding: 2px;">in progress</span></b>	#62 by Brudi8 was closed a day ago					
<input type="checkbox"/>	<b>⌚ added smoke animation for temple site</b>	#61 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ display stones on pyramids site</b>	#60 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ Display Pyramids site</b>	#59 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ add stack number for temple site</b>	#58 by Brudi8 was closed 3 days ago					
<input type="checkbox"/>	<b>⌚ Implement point assessment hover for temple site</b>	#57 by Brudi8 was closed 3 days ago					

<input type="checkbox"/>	<a href="#">Display stones on Temple site</a>	#56 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">display temple site</a>	#55 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">implement active/inactive player functionality</a>	#54 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">receive game update from backend</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#53 by Brudi8 was closed a day ago	Milestone 3	
<input type="checkbox"/>	<a href="#">receive round information from backend (new round)</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#52 by Brudi8 was closed 2 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">receive game start information from backend</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#51 by Brudi8 was closed 2 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">function for random market cards on site</a>	#50 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">add click functionality for market cards on market site (handing out of market cards)</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#49 by Brudi8 was closed a day ago	Milestone 3	
<input type="checkbox"/>	<a href="#">add click functionality for market card icons</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#48 by Brudi8 was closed a day ago	Milestone 3	
<input type="checkbox"/>	<a href="#">hover functionality for market card icons</a>	#47 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">visualize market cards on player field</a>	#46 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<a href="#">implement red card actions</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#45 by Brudi8 was closed a day ago	Milestone 3	
<input type="checkbox"/>	<a href="#">add zoom functionality for market cards</a>	#43 by Brudi8 was closed 3 days ago	Milestone 3	

<input type="checkbox"/>	<a href="#">12 Open</a>	<a href="#">✓ 91 Closed</a>	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	<a href="#">display cards on market site</a>	#42 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">add market explanation hover</a>	#41 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">visualize market site</a>	#40 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">add point assessment hover for obelisks</a>	#39 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Implement sled counter functionality</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#38 by Brudi8 was closed a day ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Implement drag and drop functionality for ships</a>	#36 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Implement stone to ship placement</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#35 by Brudi8 was closed a day ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Add obelisk counter</a>	#34 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Display stones on Obelisks</a>	#33 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Display Obelisks site</a>	#32 by Brudi8 was closed 3 days ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Style departing harbour</a> <span style="background-color: #007bff; color: white; padding: 2px;">in progress</span>	#31 by Brudi8 was closed a day ago	Milestone 3					
<input type="checkbox"/>	<a href="#">Visualize ship placement</a>	#30 by Brudi8 was closed 3 days ago	Milestone 3					

<input type="checkbox"/>	<b>Highlight In-Game structure</b>	#29 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Structure the In-Game screen</b>	#28 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Define and visualize game board</b>	#27 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Implement router functionality to game screen and login screen</b>	#26 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Add 'quickstart' button</b>	#25 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Add 'logout' button</b>	#24 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Add 'game screen' button</b>	#23 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Implement 'new game' modal</b>	#22 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Add 'new game' button</b>	#21 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Implement list of players table <small>archive</small></b>	#20 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Implement list of games table</b>	#19 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Site structure and general styling</b>	#18 by Brudi8 was closed 3 days ago	Milestone 3	
<input type="checkbox"/>	<b>Create demo lobby for testing</b>	#17 by Brudi8 was closed 3 days ago	Milestone 3	

<input type="checkbox"/>	<span>① 12 Open ✓ 91 Closed</span>	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	<b>Implement lobby UI components incl. actions/events</b>	#16 by Brudi8 was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>Implement router functionality to lobby screen</b>	#15 by Brudi8 was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>Site structure and general styling <small>duplicate</small></b>	#14 by Brudi8 was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>Remove 'name' field</b>	#13 by Brudi8 was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>Implement login UI components incl. actions/events <small>duplicate</small></b>	#12 by Brudi8 was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>Login Screen: Enter key should log in <small>enhancement</small></b>	#11 by dawwys was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>Login Screen button width <small>bug</small></b>	#10 by dawwys was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>lobby ui routing</b>	#9 by nairboon was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>new game view template</b>	#8 by nairboon was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>lobby view template</b>	#7 by nairboon was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>login view template</b>	#6 by nairboon was closed 3 days ago	Milestone 3				
<input type="checkbox"/>	<b>lobby background styling</b>	#5 by nairboon was closed 3 days ago	Milestone 3				

<input type="checkbox"/>	<b>⌚ Define and visualize game board</b>	
	#4 by nairboon was closed 3 days ago	Milestone 3
<input type="checkbox"/>	<b>⌚ Implement lobby UI components incl. actions/events</b>	
	#3 by nairboon was closed 3 days ago	Milestone 3
<input type="checkbox"/>	<b>⌚ Visualize lobby UI</b> Epic	
	#2 by nairboon was closed 3 days ago	Milestone 3
<input type="checkbox"/>	<b>⌚ Implement login UI components incl. actions/events</b>	
	#1 by nairboon was closed 3 days ago	Milestone 3

### 7.3.4 Issue Tracker Server

This section contains all of our issues on the server side. The issues are the broken down version of the tasks featured in the task list for the fourth milestone (Table 7.3.1.1) and the corresponding gantt chart (Figure 7.3.2.1).

## Closed issues:

	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
① 10 Open ✓ 55 Closed						
② Game owner can't leave game in lobby <span style="background-color: red; border: 1px solid black; padding: 2px;">bug</span>						
#65 by dawwys was closed 27 minutes ago						
③ check user story acceptance <span style="background-color: pink; border: 1px solid black; padding: 2px;">M2-feedback</span>						
#63 by nairboon was closed 4 days ago				M3		
④ SonarQube testing coverage proof screenshot <span style="background-color: purple; border: 1px solid black; padding: 2px;">Presentation</span> <span style="background-color: green; border: 1px solid black; padding: 2px;">Report</span>						
#62 by nairboon was closed 11 minutes ago				M3		
⑤ example unit test <span style="background-color: green; border: 1px solid black; padding: 2px;">Report</span>						
#61 by nairboon was closed 37 minutes ago				M3		
⑥ gantt chart for M4 plan <span style="background-color: purple; border: 1px solid black; padding: 2px;">Presentation</span> <span style="background-color: green; border: 1px solid black; padding: 2px;">Report</span>						
#59 by nairboon was closed 8 minutes ago				M3		
⑦ assign each issue to smb, also for the finished ones <span style="background-color: pink; border: 1px solid black; padding: 2px;">M2-feedback</span>						
#58 by nairboon was closed 3 minutes ago				M3		
⑧ Add acceptance criterions to user stories <span style="background-color: pink; border: 1px solid black; padding: 2px;">M2-feedback</span> <span style="background-color: green; border: 1px solid black; padding: 2px;">Report</span>						
#57 by nairboon was closed 4 days ago				M3		
⑨ Click sequence diagram <span style="background-color: pink; border: 1px solid black; padding: 2px;">M2-feedback</span> <span style="background-color: green; border: 1px solid black; padding: 2px;">Report</span>						
#56 by nairboon was closed 13 hours ago				M3		
⑩ Move all unfinished tasks before the M3 deadline to M4						
#55 by nairboon was closed 2 minutes ago				M3		
⑪ implemented continuous stone placement on pyramid site						
#53 by Brudi8 was closed 10 days ago						
⑫ add slot ids in ships for stones						
#52 by Brudi8 was closed 10 days ago						
⑬ implement stone from ship to site logic for temple site(displaying stones in corresponding order)						
#51 by Brudi8 was closed 10 days ago						
⑭ implement stone from ship to site logic for pyramid site (displaying stones in corresponding order)						
#50 by Brudi8 was closed 10 days ago						
⑮ added smoke animation for temple site						
#49 by Brudi8 was closed 10 days ago						
⑯ display stones on pyramids site						
#48 by Brudi8 was closed 10 days ago						
⑰ Display Pyramids site						
#47 by Brudi8 was closed 10 days ago						
⑱ add stack number for temple site						
#46 by Brudi8 was closed 10 days ago						
⑲ Implement point assessment hover for temple site						
#45 by Brudi8 was closed 10 days ago						
⑳ Display stones on Temple site						
#44 by Brudi8 was closed 10 days ago						
㉑ display temple site						
#43 by Brudi8 was closed 10 days ago						
㉒ implement active/inactive player functionality						
#42 by Brudi8 was closed 10 days ago						
㉓ receive game update from backend						
#41 by Brudi8 was closed 10 days ago						
㉔ receive round information from backend (new round)						
#40 by Brudi8 was closed 10 days ago						
㉕ receive game information from backend						
#39 by Brudi8 was closed 10 days ago						
㉖ function for random market cards on site						
#38 by Brudi8 was closed 10 days ago						

	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/> ① 10 Open ✓ 55 Closed						
② <b>Decision Objekte im Frontend abholen</b> #36 by since4 was closed 13 hours ago						
② <b>Wie erfahren Clients wer sie sind und ob sie zur Zeit am Zug sind?</b> #34 by since4 was closed 13 hours ago						
② <b>Client Reaction auf einen Broadcast des Servers.</b> #33 by since4 was closed 13 hours ago						
② <b>Unsere GameComponent und das Game Modell.</b> #32 by since4 was closed 13 hours ago						
② <b>Unsere GameComponent und das Game Modell.</b> #31 by since4 was closed 14 days ago						
② <b>Decision Klasse: Datenstruktur abgleichen.</b> #30 by since4 was closed 13 hours ago						
② <b>QuaryToSledAction</b> #28 by nairboon was closed 13 hours ago M3						
② <b>refactor stoneMovefactory</b> #27 by nairboon was closed 12 days ago						
② <b>Sled siteboard</b> #25 by nairboon was closed 13 hours ago M3						
② <b>Quary "Siteboard"</b> #24 by nairboon was closed 13 hours ago M3						
② <b>Dock a Ship to Market Test</b> #23 by nairboon was closed 13 hours ago M3						
② <b>implement Market SiteBoard</b> #22 by nairboon was closed 13 hours ago						
<hr/>						
② <b>implement server logic for playing violet cards</b> #21 by nairboon was closed 4 minutes ago M4						
② <b>implement handling out market cards</b> #20 by nairboon was closed 13 hours ago						
② <b>Big main game loop integration test</b> #18 by nairboon was closed 13 hours ago Sprint 1						
② <b>Test executing valid moves</b> #17 by nairboon was closed 13 hours ago						
② <b>Test available Actions/Moves for a new game</b> #16 by nairboon was closed 13 hours ago						
② <b>Obelisk executeMove</b> #14 by nairboon was closed 13 hours ago Sprint 2						
② <b>setup TDD stub for a REST query</b> #12 by nairboon was closed on 20 Mar Sprint 1						
② <b>/users/{userId}/logout POST</b> #11 by nairboon was closed 13 hours ago Sprint 1						
② <b>/users/{userId}/login POST</b> #10 by nairboon was closed 13 hours ago Sprint 1						
② <b>/users POST</b> #9 by nairboon was closed 13 hours ago Sprint 1						
② <b>/games/{gameId}/leave POST</b> #8 by nairboon was closed 13 hours ago Sprint 1						
② <b>/games/{gameId}/join POST</b> #7 by nairboon was closed 13 hours ago Sprint 1		1				
② <b>/games/{gameId} GET</b> #6 by nairboon was closed 13 hours ago Sprint 1						

	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
① 10 Open ✓ 55 Closed						
② <b>Implement in-game turn logic</b> Epic						
#5 by nairboon was closed 13 hours ago						
③ <b>Implement lobby server logic</b> Epic						
#4 by nairboon was closed 13 hours ago						
④ <b>Implement login server logic</b> Epic						
#3 by nairboon was closed 13 hours ago						
⑤ <b>/game REST GET endpoint</b>						
#2 by nairboon was closed 13 hours ago		Sprint 1				
⑥ <b>/game REST POST endpoint</b>						
#1 by nairboon was closed 13 hours ago		Sprint 1				

# 8 Testing

Our process of development did not solely rely on implementing all the necessary features. We simultaneously verified our code with the use of unit tests. Writing these tests was fundamental for thinking through the corresponding requirements and design.

The goal was to write the test cases first and subsequently write the code until all unit tests pass. The output of each test was used as a guideline on how to structure and further develop/improve our code. As a next step we extended our tests step by step, simultaneously expanding the related part of the code. That way, we were able to check the impact of each expansion.

Using this technique, our code was significantly improved which leads to fewer time-investment in maintenance. Additionally the architecture is easier to understand and has a better structure overall. Redundancy and unnecessary code snippets are being eliminated.

When further manual testing revealed any bugs, we first identified the unit test that should have prevented the bug and extended the test to cover this case.

## 8.1 Example Unit Test

In Figure 8.1.1 below we tested the joinGame() method. First we created a new user “Brudi9” and used that user in the tests below.

The standard case test expects an “ok” status when there is actually a testGame with a user that is currently in this game. The call to contentType verifies whether a Json is returned or not.

The other three comparisons starting at line 177, 183 and 189 are edge cases. We compare the id of the game and expect an isNotFound status if that game could not be found. If there is no valid test user token then the status will be forbidden. In the last case we combine the two edge cases before and return an isNotFound status.

```

165
166 @Test
167 public void joinGame() throws Exception {
168
169     User testUser = userService.createUser( username: "Brudi9" );
170
171     //standard case
172     this.mockMvc.perform(post( urlTemplate: "/game/" + testGame.getId() + "/" )
173         .param("token", testUser.getToken())
174         .contentType(contentType))
175         .andExpect(status().isOk());
176
177     //testGame is not found
178     this.mockMvc.perform(post( urlTemplate: "/game/" + -1 + "/join" )
179         .param("token", testUser.getToken())
180         .contentType(contentType))
181         .andExpect(status().isNotFound());
182
183     //user token deviant
184     this.mockMvc.perform(post( urlTemplate: "/game/" + testGame.getId() + "/join" )
185         .param("token", "Hacker")
186         .contentType(contentType))
187         .andExpect(status().isForbidden());
188
189     //testGame not found && user token deviant
190     this.mockMvc.perform(post( urlTemplate: "/game/" + -1 + "/join" )
191         .param("token", "")
192         .contentType(contentType))
193         .andExpect(status().isNotFound());
}

```

Figure 8.1.1

## 8.2 Final Test Coverage



Figure 8.2.1: Test coverage

Figure 8.2.1 displays our final test coverage on SonarQube. Our coverage supports our code and meets the threshold. However, the reported coverage is only a snapshot of the coverage.

In Figure 8.2.2 the blue graph shows the coverage history whereas the purple graph shows the history of lines-of-code. Thus, it is evident from Figure 8.2.2 that we mostly followed Test-Driven-Development since the coverage is mostly steadily while the overall codebase was growing. The coverage drop around Apr 23, was caused due to the addition of the websocket functionality, which was accompanied by locally passing unit-tests, but we couldn't get them to run on Jenkins. Later, after we finished the core game functionality, we resumed debugging the issue, which was related to timing issues of the asynchronous test suite, and got the test suite working again on Jenkins. This is visible in the sharp coverage increase around Apr 30.



Figure 8.2.1: Coverage History

## 8.3 Fast Forward Function

It took us some time to figure out a method capable of displaying the fast-forward function. At the same time we tried to use the function as a tool to test our implementation on the client side. We came up with a very simplified solution of an automated testing bot.

The fast-forward function is implemented as two bots playing against each other. The bots are playing at a much higher average speed compared to a normal user. Within 1.5 minutes the game will be finished if played by two bots.

In the final version deployed on firebase, we removed the bot functionality again, since it is not a core functional requirement of the game.

# 9 Development Infrastructure

This chapter contains the links to our published game on firebase, the backend on heroku and the software code on GitHub.

## 9.1 Published Game on Firebase

Our link to the published game-application hosted on Firebase:

<https://bigdata-17.firebaseio.com/>

## 9.2 Backend on Heroku

Our backend on Heroku:

<https://sopra-fs17-group13.herokuapp.com/>

## 9.3 Software Code on GitHub

Our link to the software code, hosted on GitHub:

<https://github.com/sealuzh/sopra-fs17-group13-client>

<https://github.com/sealuzh/sopra-fs17-group13-server>

# 10 Glossary

The glossary in table 10.1 explains the meaning of terms officially used in the game manual, as well as terms defined by us.

Term	Meaning
Active Player	The player whose turn it currently is during an active game.
End Screen	The screen that appears after the game has ended; from here, players may go back to the Lobby screen.
Guest in the Lobby Screen	A player that has already logged in, but has not yet joined any game lobby.
Information Box	Area in the in-game screen in which the player receive information during the game.
In-Game Screen	The screen shown when actively participating in a round of the game until the game's end.
Lobby Screen	The screen where all the currently running games are shown and can be joined. Reached after logging in on the Login screen.
Login Screen	The first screen presented to a new arrival upon entering the website; here, they are prompted to enter their user data before being able to enter the Lobby screen.
Market Card	34 in total; a maximum of 4 are available on the market at the same time. There are 4 colors of market cards available, varying in functionality: red, green, blue, and purple.
New Arrival to the Website	A player that has just arrived at the login screen of the website and is not yet logged in.
Player	A player that is participating in an active game.
Player Field	The 4 zones in the corners of the in-game screen.
Round Card	21 in total; 7 each for two, three and four player-games respectively. Round cards determine which 4 of the 8 ship tokens to use in each round. Only 6 are used per game.
Scoring Track Board	The board used to keep track of players' scores during the game.
Ship Token	8 ship tokens varying in size from one to four slots for stones. Players place stones on the ship tokens to transport them to the site boards.

Site Board	4 site boards in total. These are the locations where stones will be placed in order to gain points.
Stone	120 wooden blocks (called “stones”), 30 in each of the 4 game colors: black, white, brown, and gray.
Stone Quarry	Each player has a stone quarry. At the beginning of the game it holds 30 stones each.
Supply Sled Token	Each player has one; there, he can store up to five stones at a time. One in each of the 4 game colors: black, white, brown, and gray.

Table 10.1

# 11 List Of Tables

- Table 3.1.1 User Stories Index
- Table 3.1.2 User Stories M1
- Table 3.2.1 User Stories Index
- Table 3.2.2 User Stories M2
- Table 3.3.1 User Stories Index
- Table 3.3.2 User Stories M3
- Table 4.1.1 REST Interface Specification
- Table 4.2.1.1 WebSocket Specification
- Table 5.1 Risk Analysis
- Table 7.1.1.1 Task List From M1 To M2
- Table 7.2.1.1 Task List From M2 To M4
- Table 7.3.1.1 Task List From M3 To M4
- Table 10.1 Glossary

# 12 List Of Figures

- Figure 2.1.1 Domain Model
- Figure 2.2.1 Class Diagram Overview
- Figure 2.2.2 Game Class
- Figure 2.2.3 TurnController
- Figure 2.2.4 Summary Of Figure 3.2.3 and Figure 3.2.5
- Figure 2.2.5 MoveFactory
- Figure 2.2.6 Command Interface
- Figure 4.2.1 WebSocket - REST Interface
- Figure 4.2.2.1 WebSocket Interaction Diagram
- Figure 6.1.1.1 Login Screen M1
- Figure 6.1.2.1 Lobby Screen M1
- Figure 6.1.3.1 New Game Screen M1
- Figure 6.1.4.1 In Game Screen M1
- Figure 6.1.5.1 End Game Screen M1
- Figure 6.2.1.1 Login Screen M2
- Figure 6.2.2.1 Lobby Screen M2
- Figure 6.2.3.1 New Game Screen M2
- Figure 6.2.4.1 In Game Screen M2
- Figure 6.2.5.1 End Game Screen M2
- Figure 6.3.1.1 Login Screen M3
- Figure 6.3.2.1 Lobby Screen M3
- Figure 6.3.3.1 New Game Screen M3
- Figure 6.3.4.1 In Game Screen M3
- Figure 6.3.5.1 End Game Screen M3
- Figure 6.3.6.1 Click Sequence Diagram Presentation
- Figure 6.3.6.2 Click Sequence Diagram Report
- Figure 6.4.1.1 Login Screen M4
- Figure 6.4.2.1 Lobby Screen M4
- Figure 6.4.3.1 New Game Screen M4
- Figure 6.4.4.1 In Game Screen Instructions M4
- Figure 6.4.4.2 In Game Screen M4
- Figure 6.4.5.1 End Game Screen M4
- Figure 7.1.2.1 Gantt Chart From M1 To M2
- Figure 7.2.2.1 Gantt Chart From M2 To M3 Sprint 1 To 3
- Figure 7.2.2.2 Gantt Chart From M2 To M3 Sprint 4 To 6
- Figure 7.2.2.3 Gantt Chart From M2 To M3 Sprint 7 To 8
- Figure 7.3.2.1 Gantt Chart From M3 To M4
- Figure 8.1.1 Unit Test For joinGame-Method
- Figure 8.2.1 Test Coverage M4
- Figure 8.2.2 Test Coverage M4: History
- Figure 13.1: Action package
- Figure 13.2: Factory package
- Figure 13.3 Bluecards package
- Figure 13.4: Marketcards package

# 13 Auto Generated Class Diagram Final Version

In this chapter we include the final version of our class diagram. It was created with IntelliJ. It represents parts of the final structure of our web application. Illustrating that we mostly followed the planned architecture.

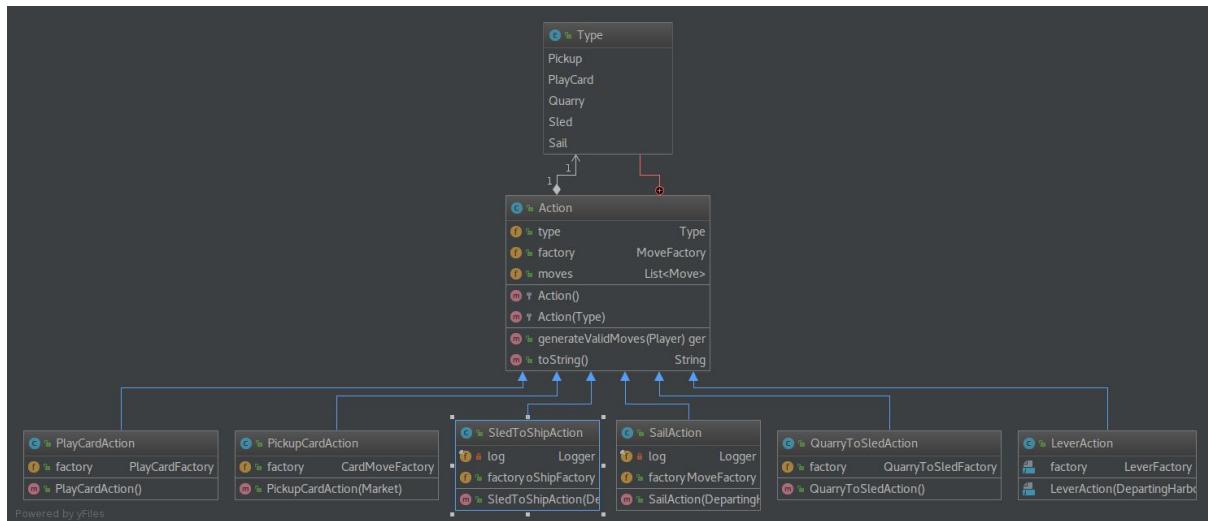


Figure 13.1: Action package

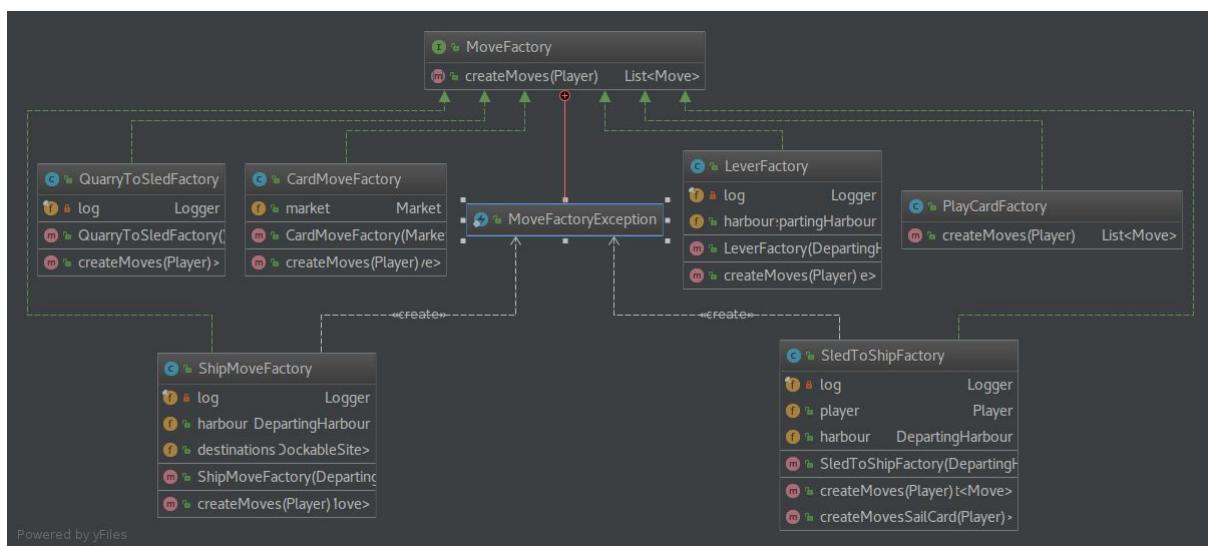


Figure 13.2: Factory package

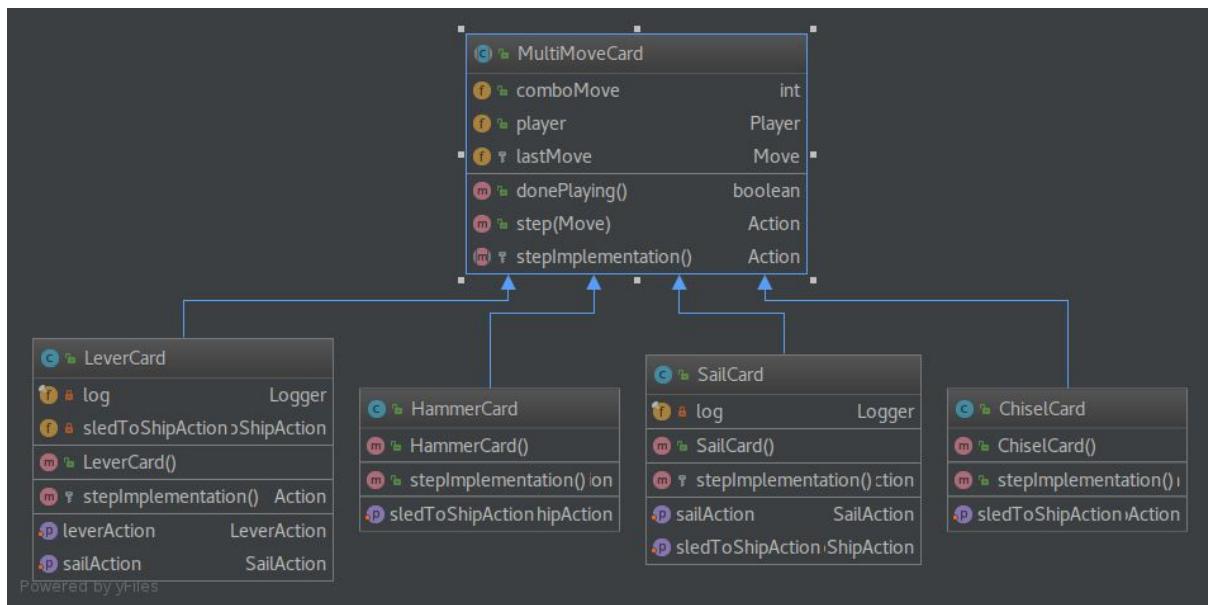


Figure 13.3 Bluecards package

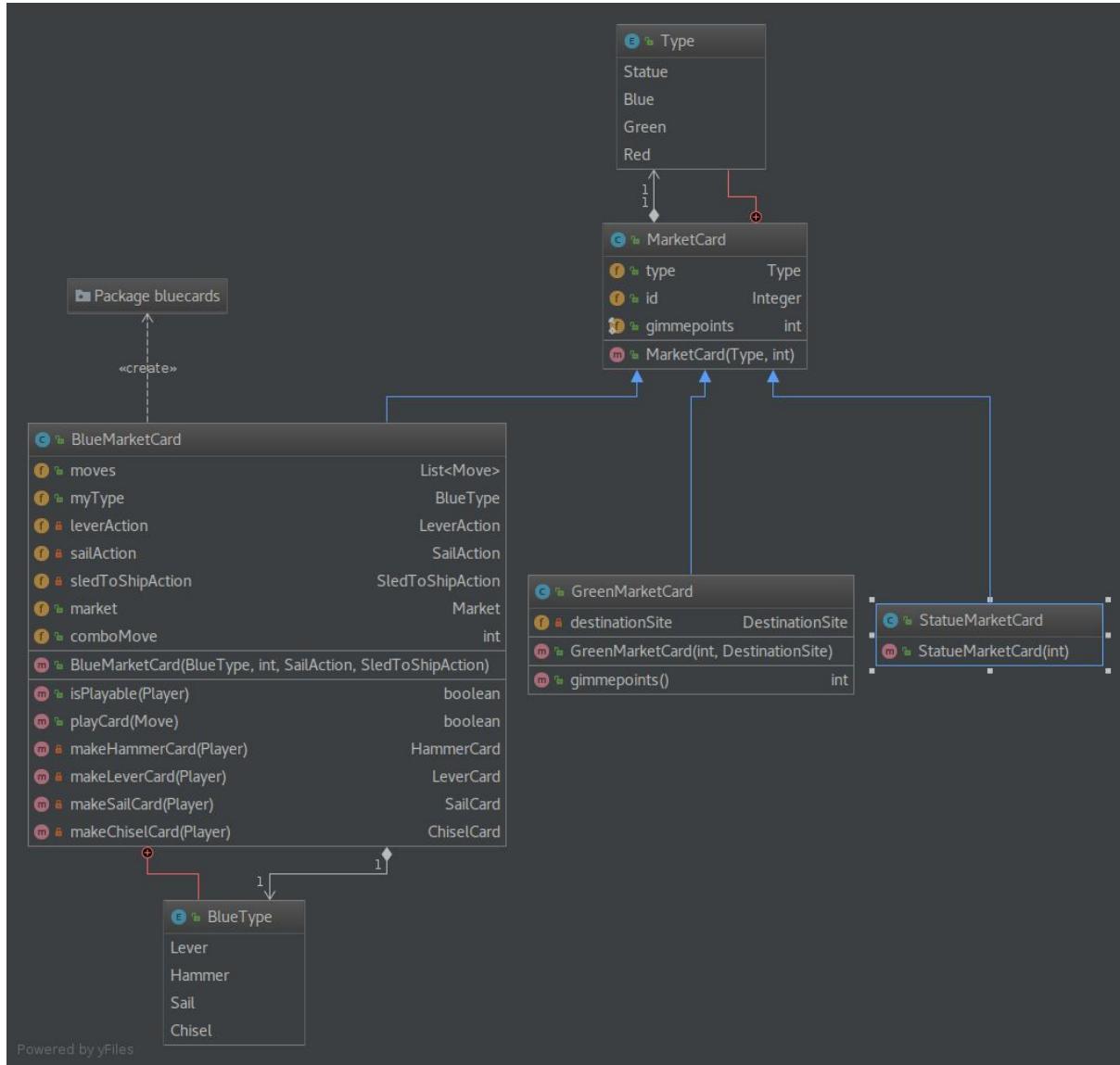


Figure 13.4: Marketcards package