

# **Release 2 Dokumentation**

**Product Owner:**  
Gianluca Voss

**Scrum Master:**  
Rohhullah Walizada

**Entwickler:**  
Birk Magnussen  
Gianluca Voss  
Kapil Karki  
Patrick Michla

## Inhaltsverzeichnis

Erste Ideen.....	4
Mockups.....	4
Armeemanager.....	4
Spielgeschehen.....	9
Domain & User Stories.....	14
Armee speichern.....	14
Armee laden.....	15
Spiel starten.....	16
Menü Knopf.....	17
Chat minimieren.....	18
Chat benutzen.....	19
TicTacToe spielen.....	20
Snake spielen.....	20
Chuck Norris Witze.....	21
Sprints.....	22
3. Sprint.....	22
Vom letzten Sprint übernommen.....	22
TB – 10 Select Chat Target.....	22
TB – 11 View Chat Target.....	23
TB – 87 Code Cleanup.....	23
TB – 88 Fat Jar.....	24
Stories und Tasks.....	24
TB – 53 Basic FXML Armymanager.....	24
TB – 55 Save Configuration.....	25
TB – 56 Load Configuration.....	25
TB – 57 Increase and Decrease Units.....	26
TB – 58 Show Configuration.....	26
TB – 61 Buttons.....	27
TB – 63 Basic FXML InGame.....	27
TB – 64 InGame Chat.....	28
TB – 65 Transitions.....	29
TB – 68 TicTacToe.....	29
TB – 71 Whitemode.....	30
TB – 78 Textur-Manager.....	30
TB – 79 Sound-Manager(manage part).....	31
TB – 80 Animated textures.....	31
TB – 81 Create Unit Textures.....	31
TB – 83 Sound – Manager(IO).....	31
TB – 85 Test UI/Util.....	32
TB – 86 Request Armsaves.....	32
TB – 89 Server Requests.....	33
TB – 91 Logger.....	33
TB – 92 PlayerList update.....	33
Bugs.....	34
TB – 90 Handshake Error.....	34
TB – 93 Loading saved LoginData.....	34

TB – 94 Errorhandler Input.....	35
Burndown – Diagramm.....	35
Zum Abschluss .....	36
4. Sprint.....	37
Tasks und Stories.....	37
TB - 82 Create Terrain Textures.....	37
TB - 96 ShowJoinedPlayer.....	37
TB - 99 Server Messages.....	38
TB - 100 Show initial Game.....	38
TB - 101 ChatWindow Movements.....	39
TB - 103 Show Images in ArmyManager.....	40
TB - 108 Configure before Game.....	40
TB - 109 Minimap.....	40
TB - 110 Move GameField.....	41
TB - 111 SmallChat.....	41
TB - 112 NotificationHandler Color.....	42
TB - 115 Update PlayerList.....	42
TB - 116 Minimap Rendering.....	42
TB - 120 ArmyManager Start Game with Config.....	43
Bugs.....	43
TB - 95 NotificationHandler.....	43
TB – 105 ChatAppearence.....	44
TB - 107 ArmyManager Chat Size.....	44
TB - 113 Whitemode.....	44
TB - 114 Icon in Fullmode.....	44
TB - 118 ArmyRequestTests.....	44
Nicht geschafft.....	45
TB – 77 Javadoc.....	45
TB - 84 Zuendorf Meme.....	45
TB - 119 TicTacToeTest.....	45
Burndown – Diagramm.....	46
Zum Abschluss.....	46
Resultate.....	48
Stand der Veröffentlichung.....	48
Lobby.....	48
Armeemanager.....	50
Spielgeschehen.....	53
Schlusswort.....	55

# Erste Ideen

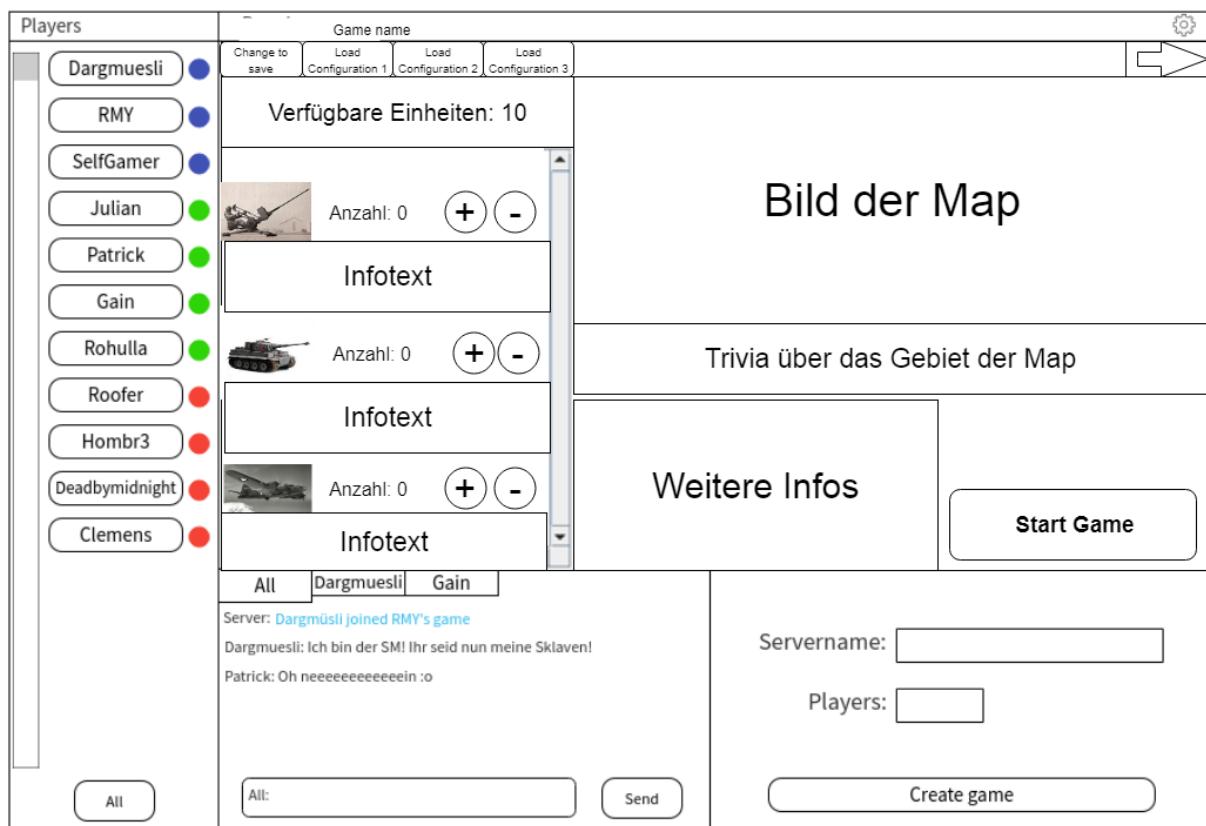
## Mockups

Erste Ideen wie die gewünschten Features aussehen könnten.

### Armeemanager

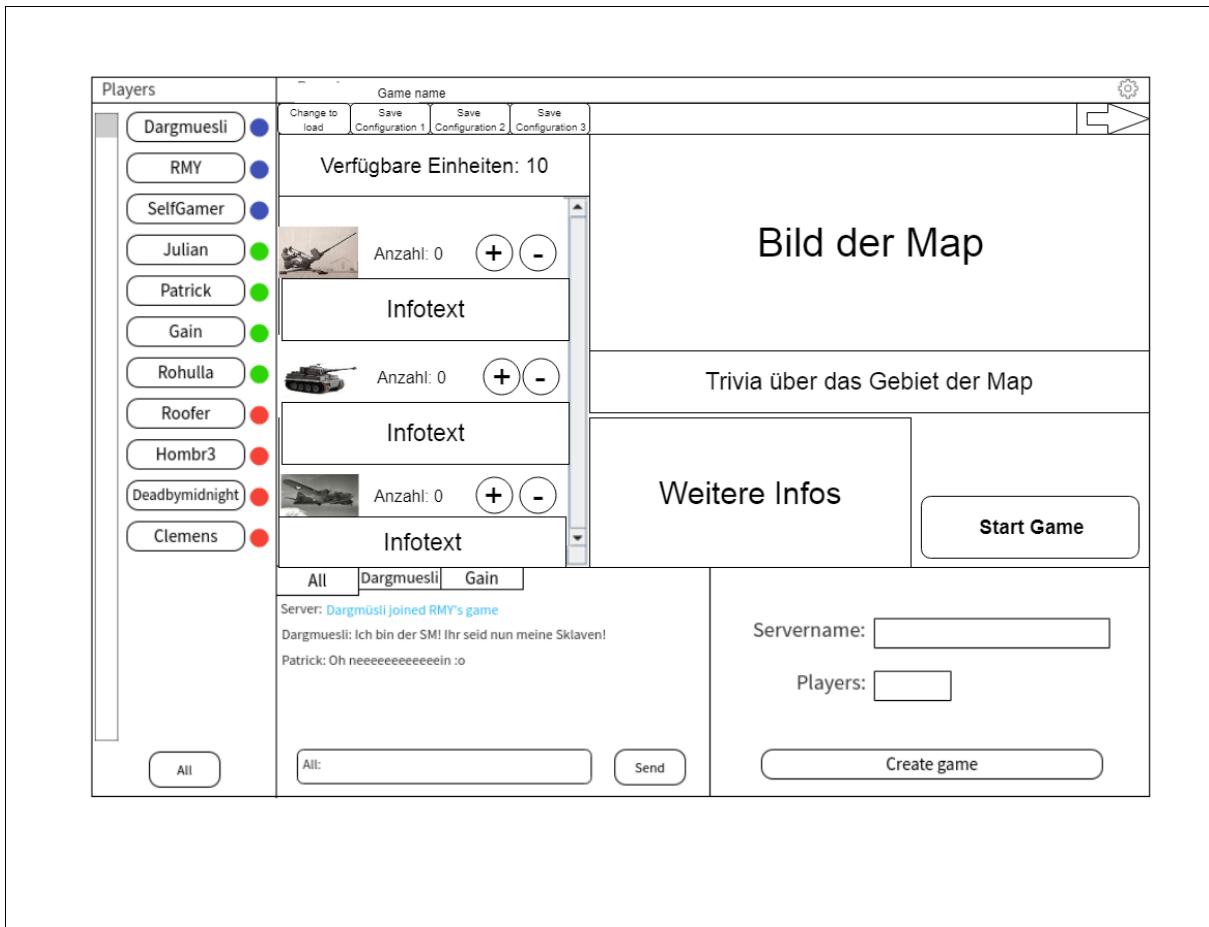
Der Armeemanager wird in den vorhandenen Bildschirm eingebettet, sodass es weniger überflüssige Übergänge und Bildschirmwechsel gibt, die das Spielerlebnis einschränken könnten.

Es gibt hier die Möglichkeit mit den Knöpfen gespeicherte Armeen zu laden oder wenn man auf den „Change to Save“ - Knopf drückt, die ausgewählte Armee zu speichern. Um die Armee zu verändern kann man außerdem die Plus - und Minus - Knöpfen verwenden. Rechts oben ist ein Pfeil mit dem man zurück zur Lobby gelangen kann. Darunter sehen wir offene Felder, die für weitere Features beschreiben, aber es ist nicht sicher, ob diese mit dem Server realisierbar sind.



Drückt man den Knopf „Start Game“, startet das Spiel und der Bildschirm wechselt zum Spielgeschehen.

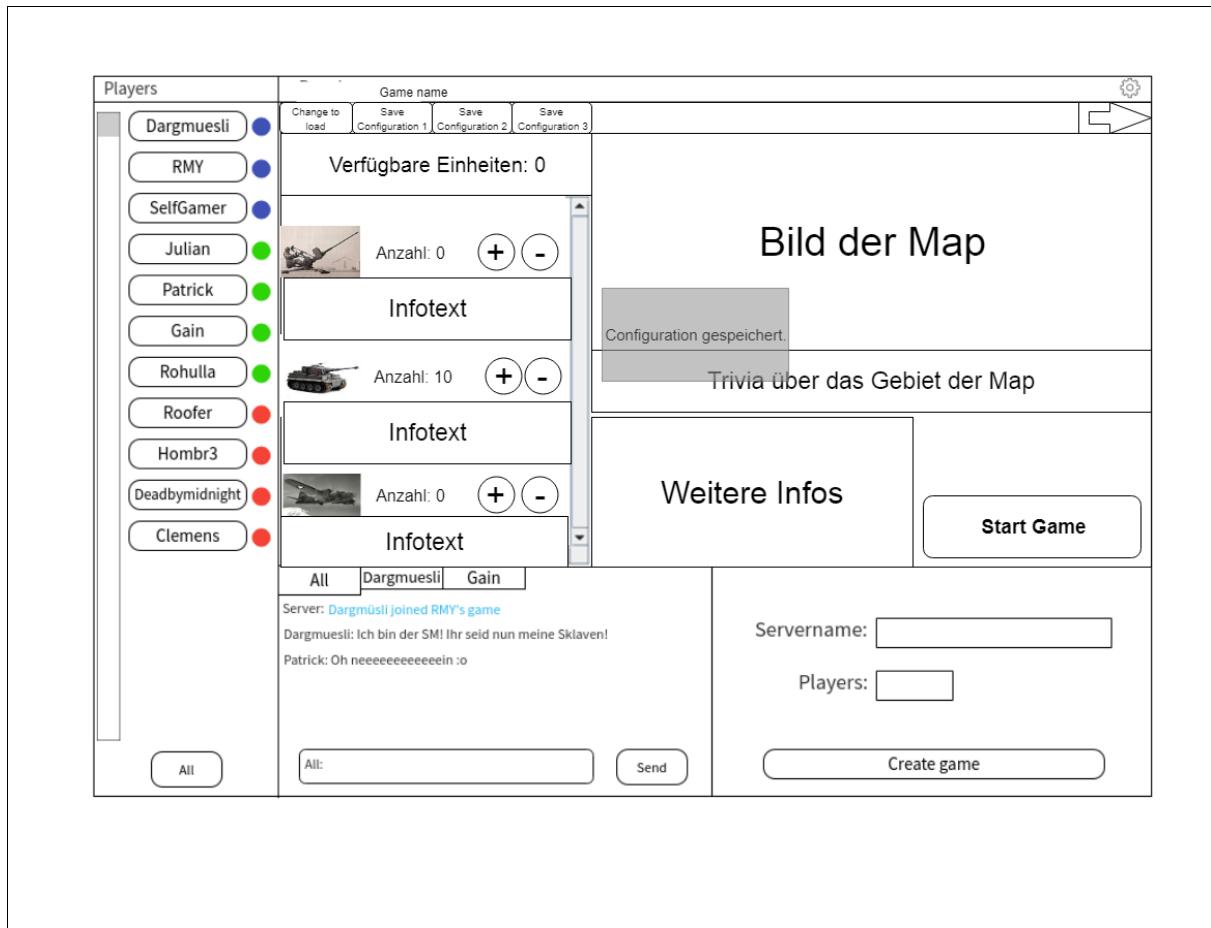
Hier das Mockup, wenn man auf den „Change to Save“ - Knopf gedrückt hat.



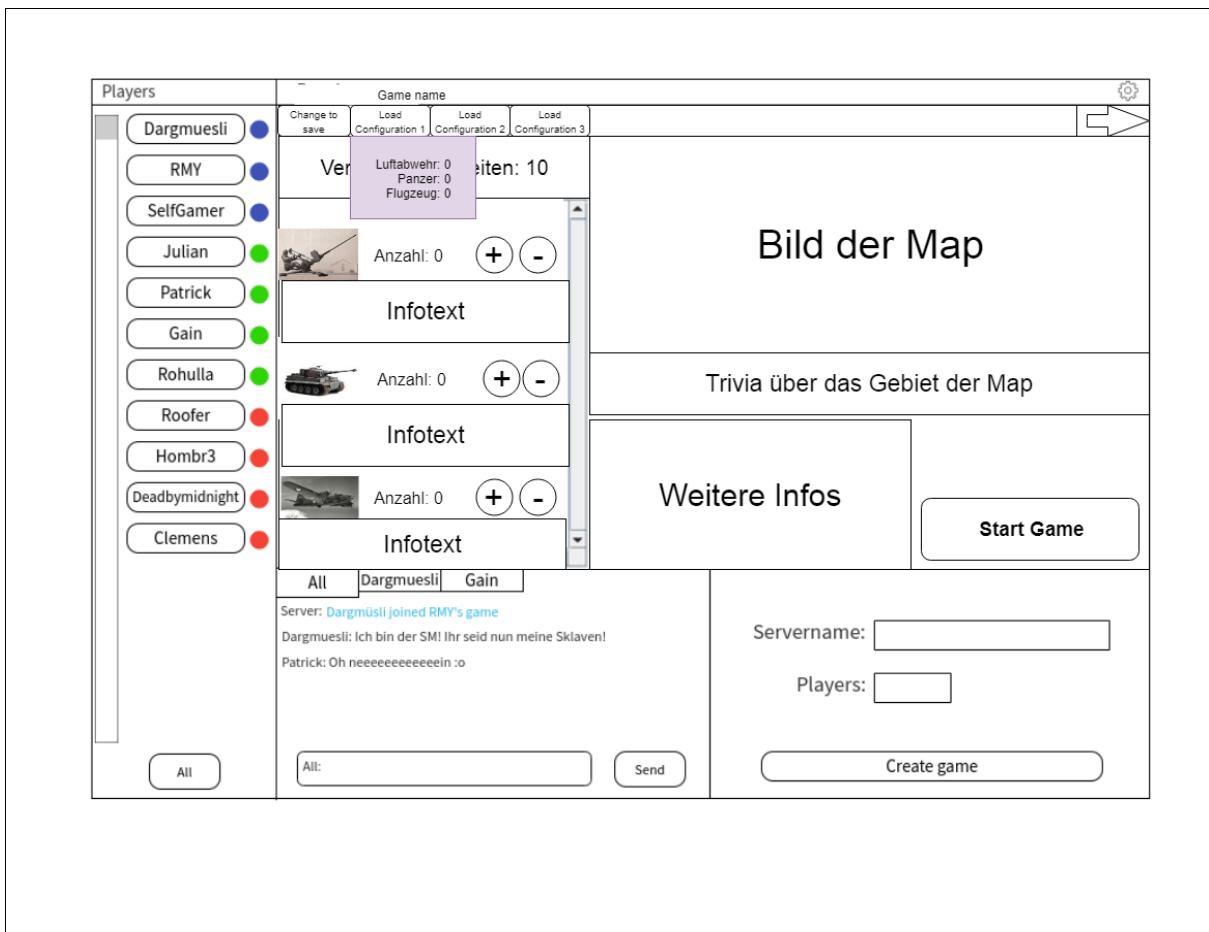
Hier das Mockup, wenn die Einheiten verteilt wurden.

Players		Game name				
		Change to load	Save Configuration 1	Save Configuration 2	Save Configuration 3	
		Verfügbare Einheiten: 0				
		Anzahl: 0 <input type="button" value="+"/> <input type="button" value="-"/> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Infotext</div> Anzahl: 10 <input type="button" value="+"/> <input type="button" value="-"/> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Infotext</div> Anzahl: 0 <input type="button" value="+"/> <input type="button" value="-"/> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Infotext</div>				
		<input type="button" value="All"/> <input type="button" value="Dargmuesli"/> <input type="button" value="Gain"/>				
		<p>Server: Dargmüsli joined RMY's game            Dargmuesli: Ich bin der SM! Ihr seid nun meine Sklaven!            Patrick: Oh neeeeeeeeeeeein :o</p>				
		<p><b>Bild der Map</b></p> <p>Trivia über das Gebiet der Map</p> <p>Weitere Infos</p> <p><b>Start Game</b></p>				
		<p>Servername: <input type="text"/></p> <p>Players: <input type="text"/></p> <p><b>Create game</b></p>				
<input type="button" value="All"/>		<input type="text" value="All:"/> <input type="button" value="Send"/>				

Hier das Mockup, wenn man auf „Safe Configuration 1“ geklickt hat.



Möchte man die Armeekonfiguration speichern oder laden, ist es wichtig zu wissen, was sich auf dem Speicherstand befindet. Aus diesem Grund soll es möglich sein sich dieses anzeigen zu lassen. Durch Rechtsklick auf den „Load Configuration 1“ Knopf wird deswegen ein Fenster aufgeklappt, in dem sich diese Informationen befinden.



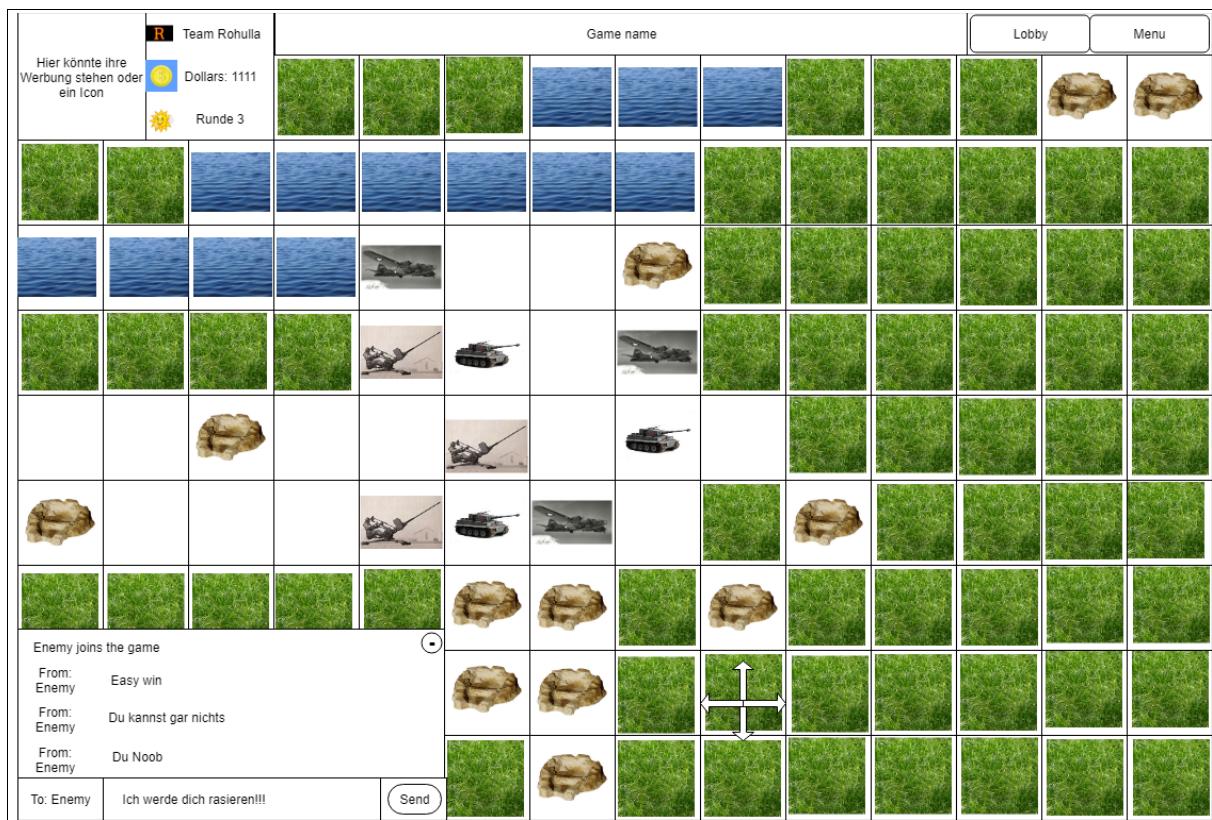
## Spielgeschehen

Am einfachsten ist es das Spielfeld in Kästchen aufzuteilen und jedem Kästchen eine Textur zu zuteilen, je nachdem um was für ein Feld es sich handelt. Einheiten können auch auf dem Spielfeld platziert werden. Um die Kästchen hier zu verdeutlichen sind sie mit Abtrennungen markiert. Diese sollten in dem Ergebnis nicht mehr vorhanden sein.

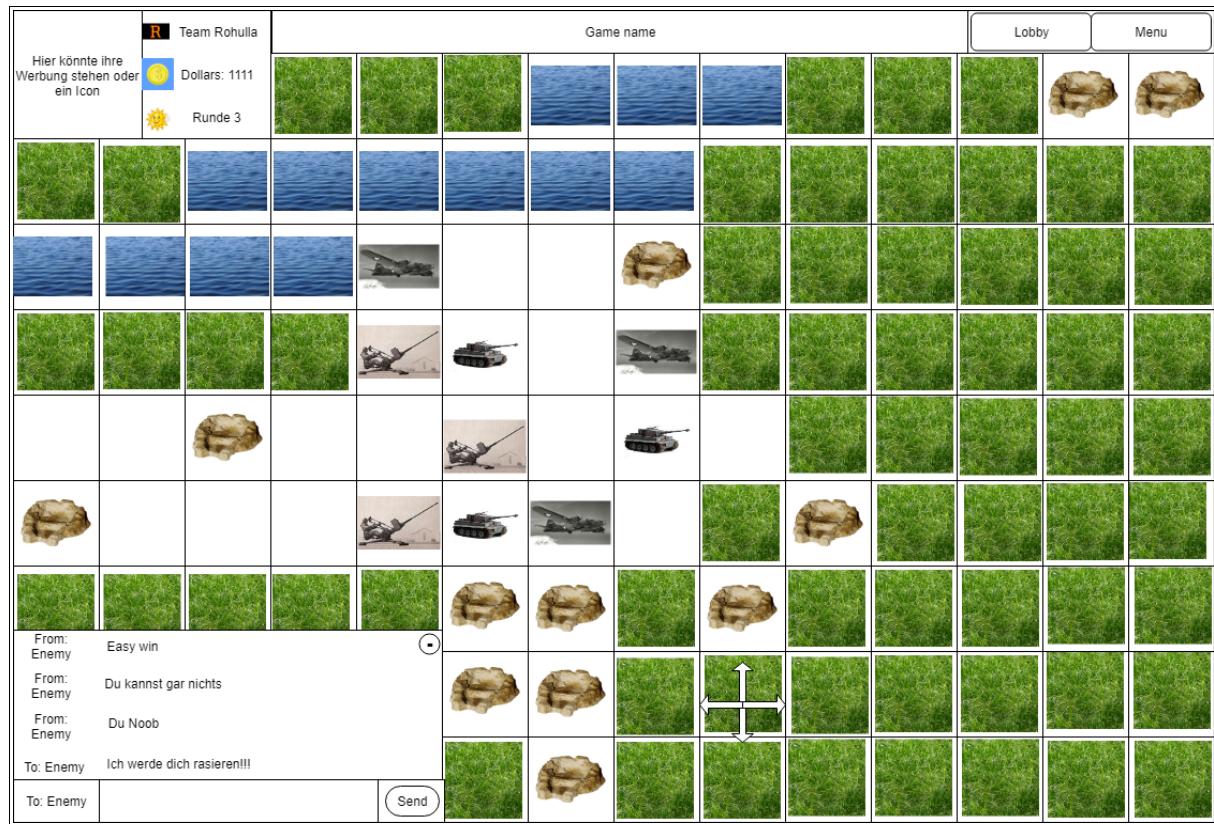
Links oben könnte ein selbst gewähltes Icon oder Spielinformationen angezeigt werden.

Rechts oben sind Knöpfe für das Zurückkehren in die Lobby und das Öffnen eines Menüs vorhanden.

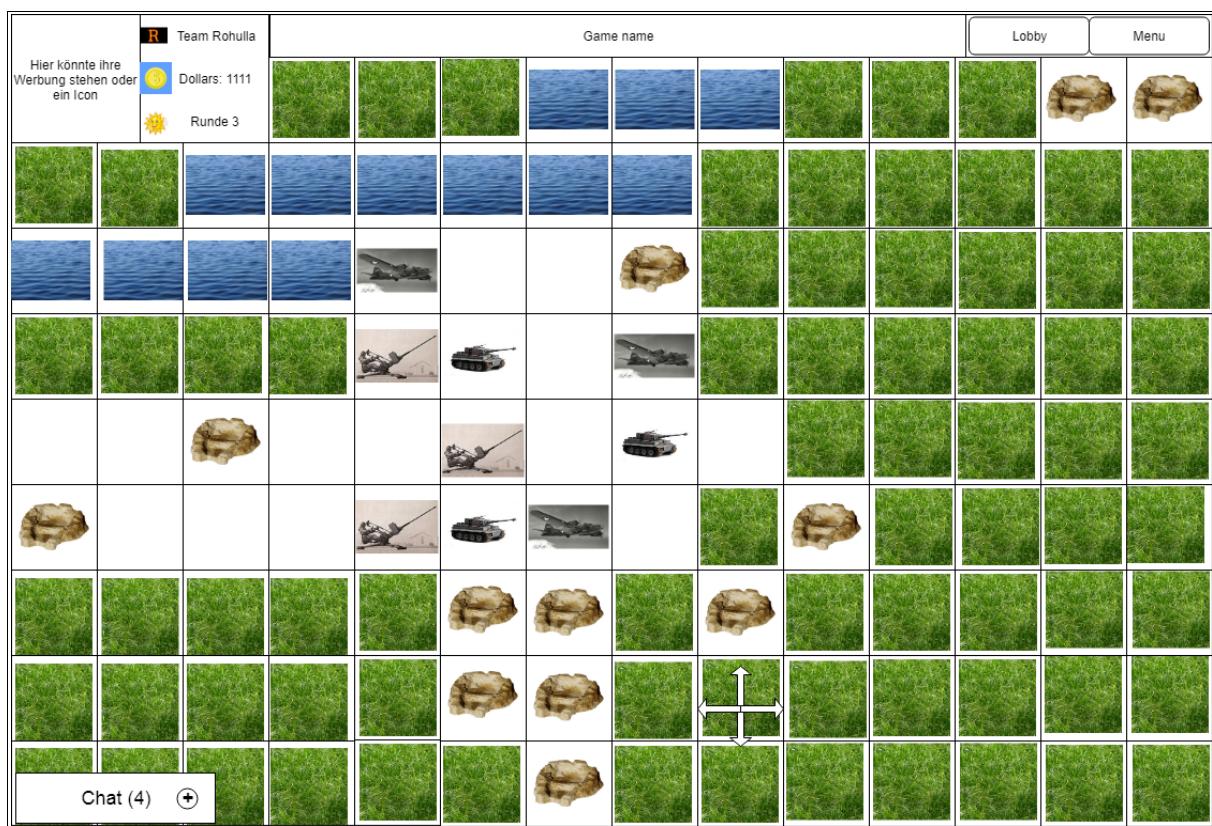
Links unten sieht man den Chat in dem man Nachrichten mit seinen Kontrahenten austauschen kann. Außerdem wird man über ihn informiert, sobald ein Spieler dem Spiel beitritt. Der Chat kann verkleinert werden aber benachrichtigt den Spieler trotzdem, wenn eine Nachricht ankommt oder ein Ereignis auftritt.



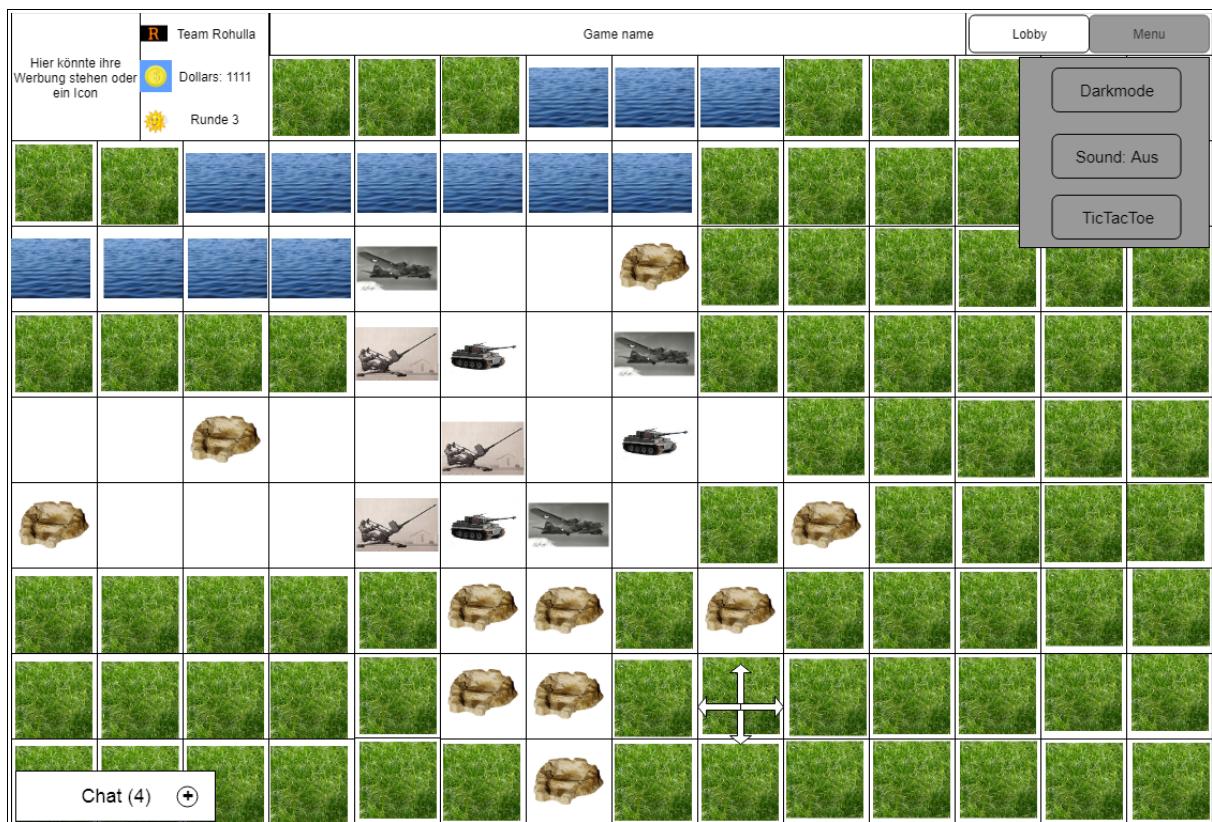
Auf diesem Bild sieht man das Mockup, nachdem auf "send" geklickt, oder die Enter-Taste gedrückt wurde.



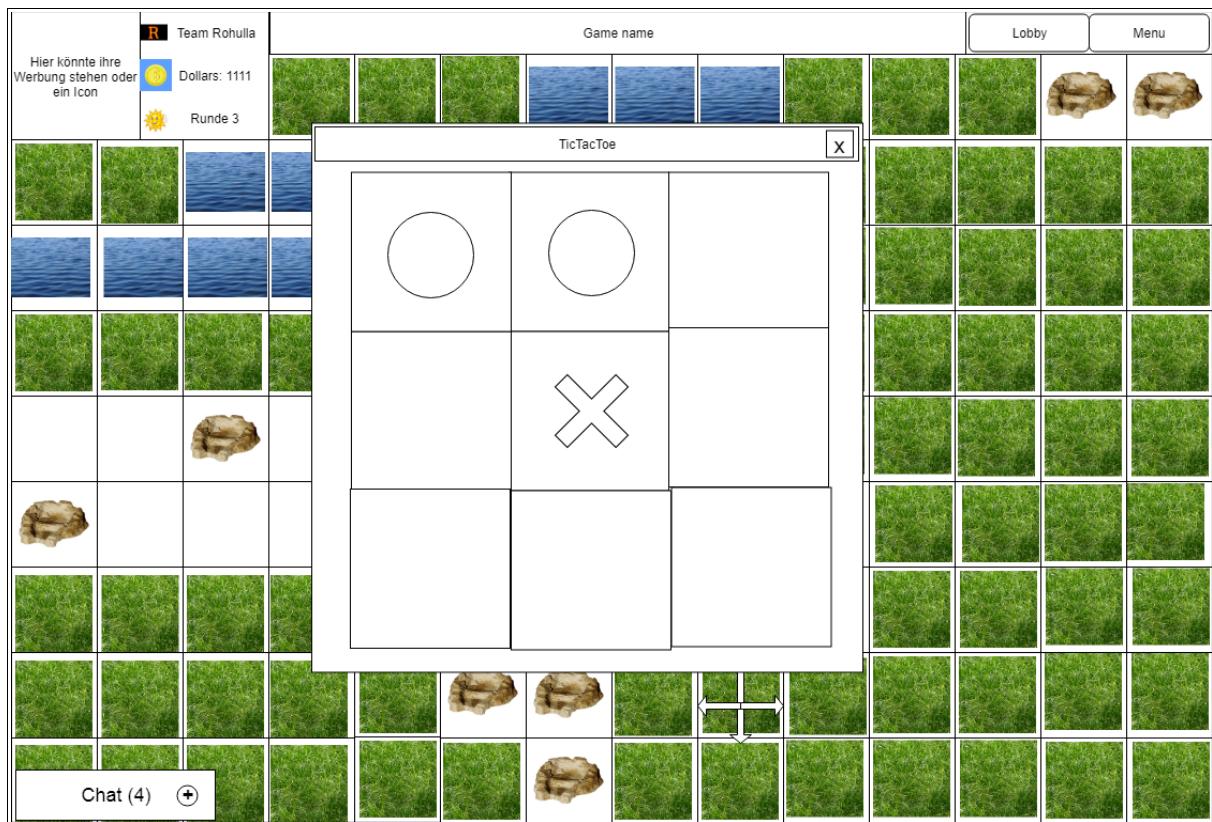
Auf diesem Mockup wurde der Chat mit dem dazugehörigen Knopf verkleinert. Hierbei wird in der Klammer angezeigt, wie viele neue Nachrichten und Ereignisse eingegangen sind, seit dem man den Chat verkleinert hat. Durch klicken auf den Plus - Knopf öffnet sich wieder der große Chat.



In diesem Mockup ist zu sehen, was passieren soll, wenn man auf den „Menu“ - Knopf klickt. In diesem Menü sind dann Zusatzfeatures vorhanden, wie ein heller/dunkler Modus, ein TicTacToe Spiel oder die Möglichkeit den Sound auszuschalten.



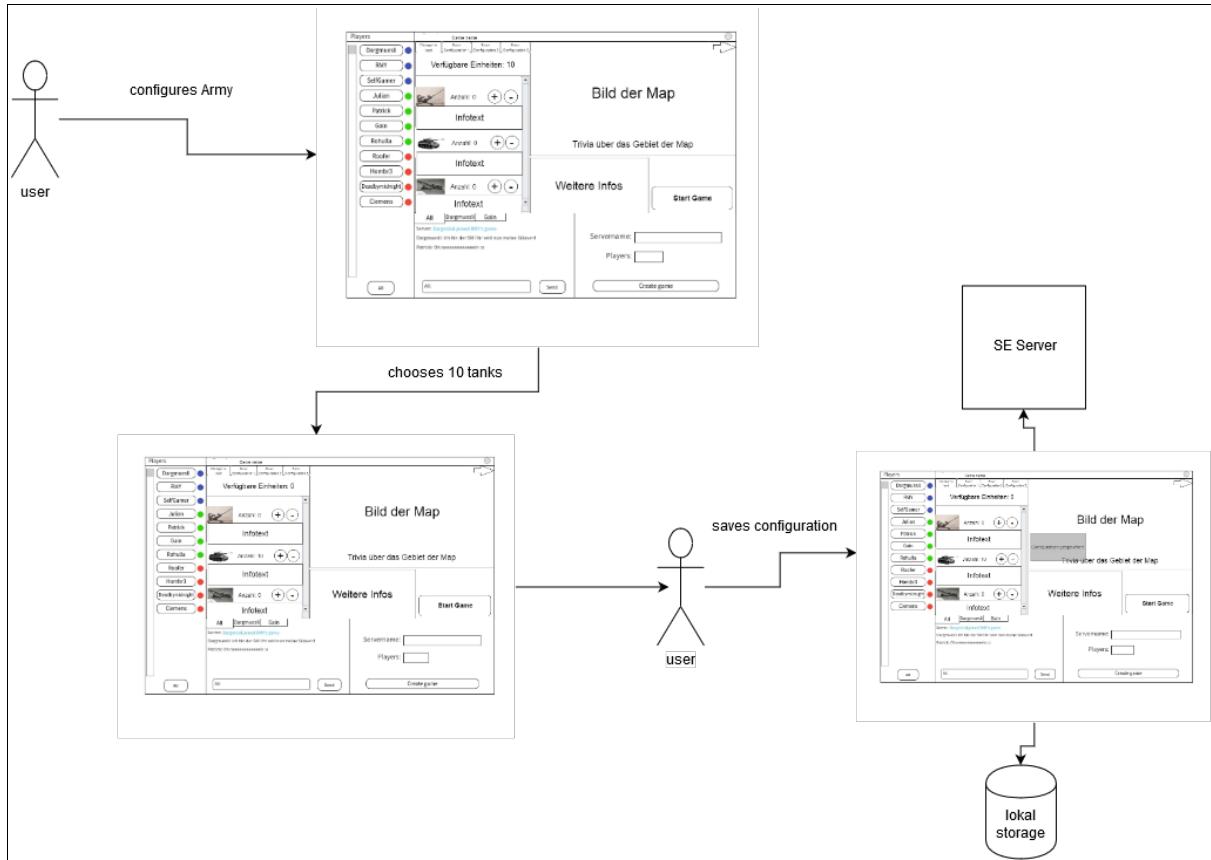
Drückt man nun auf den „TicTacToe“ - Knopf, so öffnet sich ein neues Fenster über dem Spielfeld. In diesem Fenster kann man TicTacToe gegen den Computer spielen, während man auf den Zug seines Gegners wartet. Das könnte folgendermaßen aussehen:



## Domain & User Stories

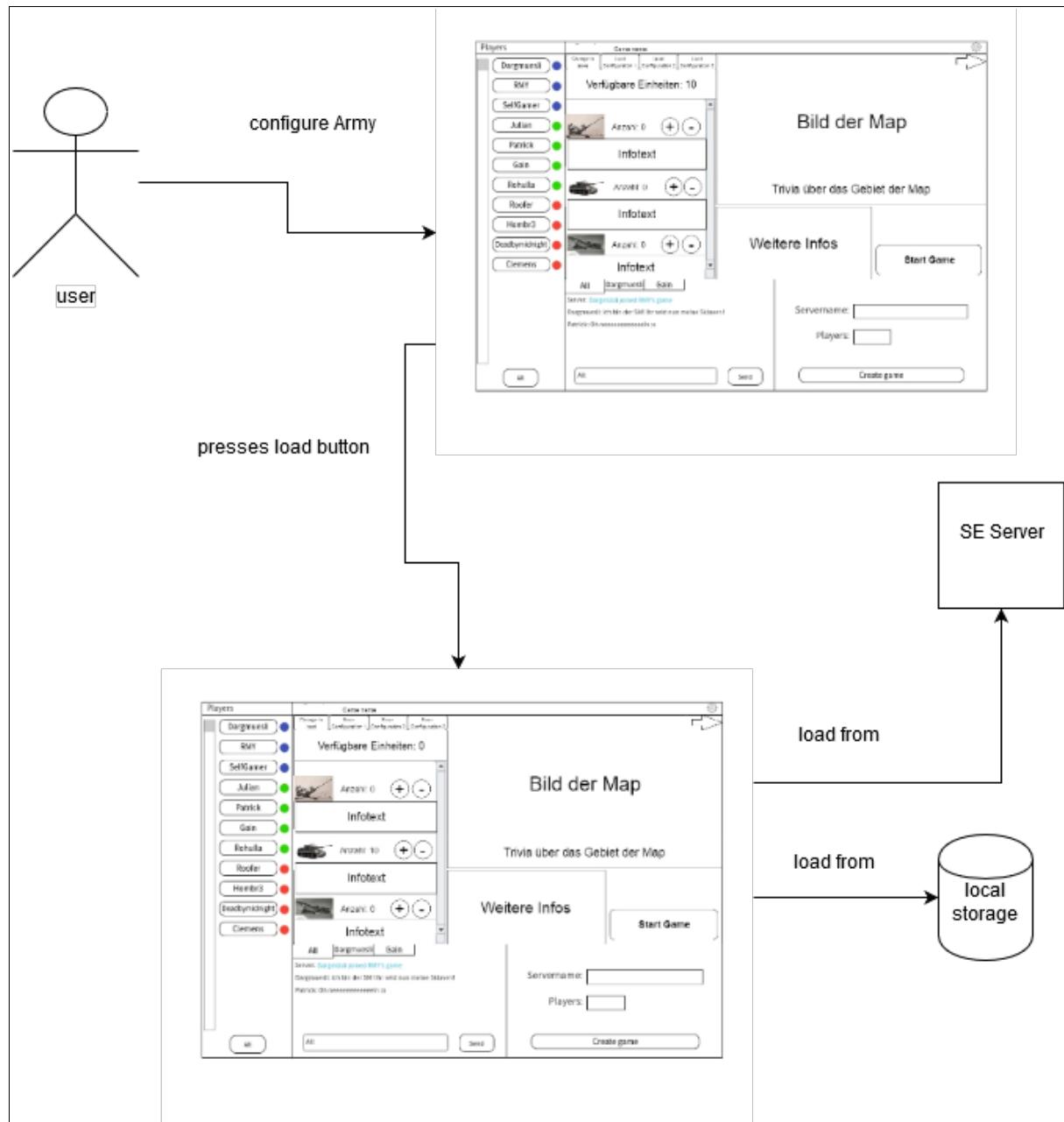
### Armee speichern

In dieser Story ist der Benutzer in dem Armeemanager. Dort wählt er zehn Panzer und speichert die Konfiguration. Die Information über die Konfiguration wird dann auf dem lokalen Speicher und auf dem Server abgelegt.



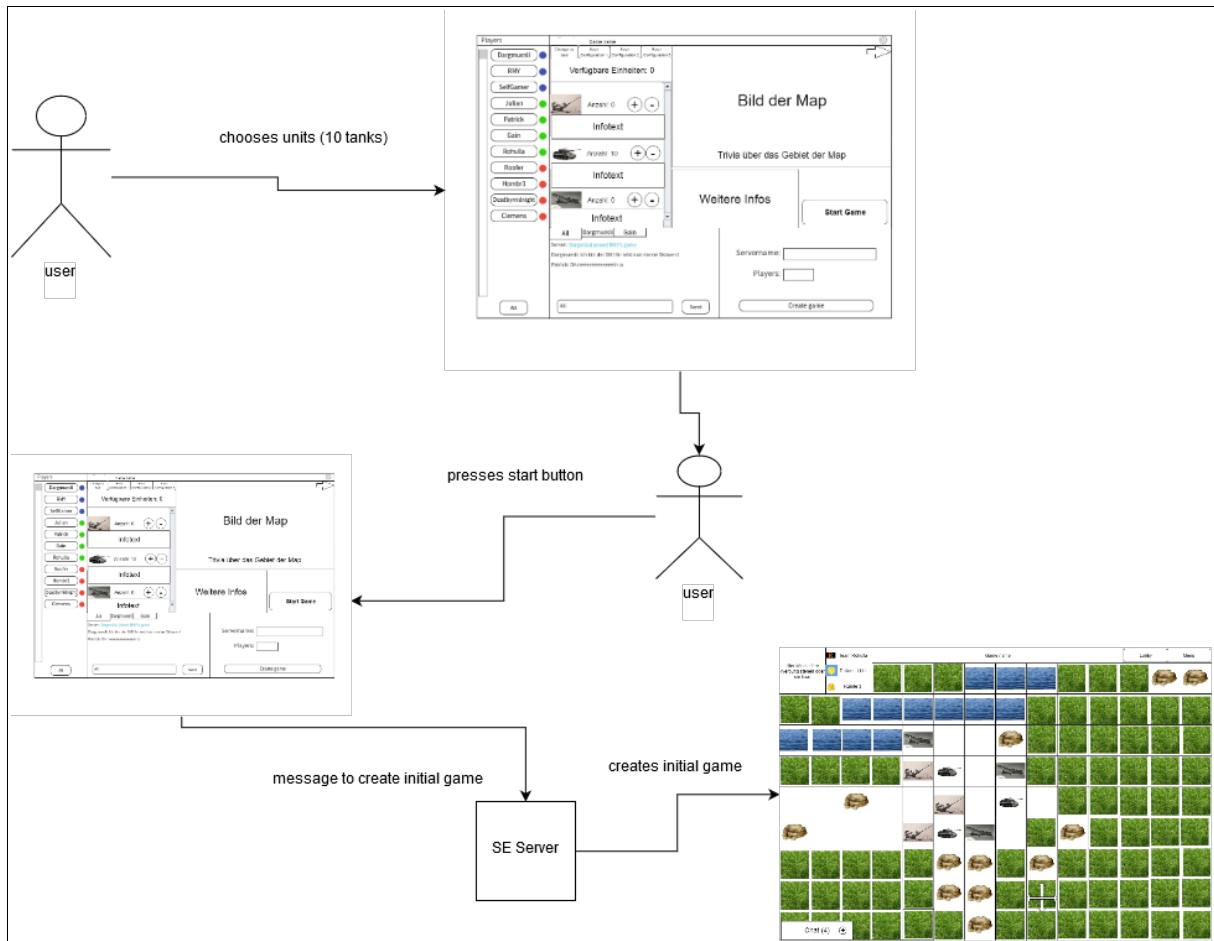
## Armee laden

Der Benutzer befindet sich im Armeemanager. Dort klickt er auf den „Load Configuration 1“ - Knopf und es wird die Konfiguration vom lokalen Speicher oder Server geladen und taucht auf dem Bildschirm auf.



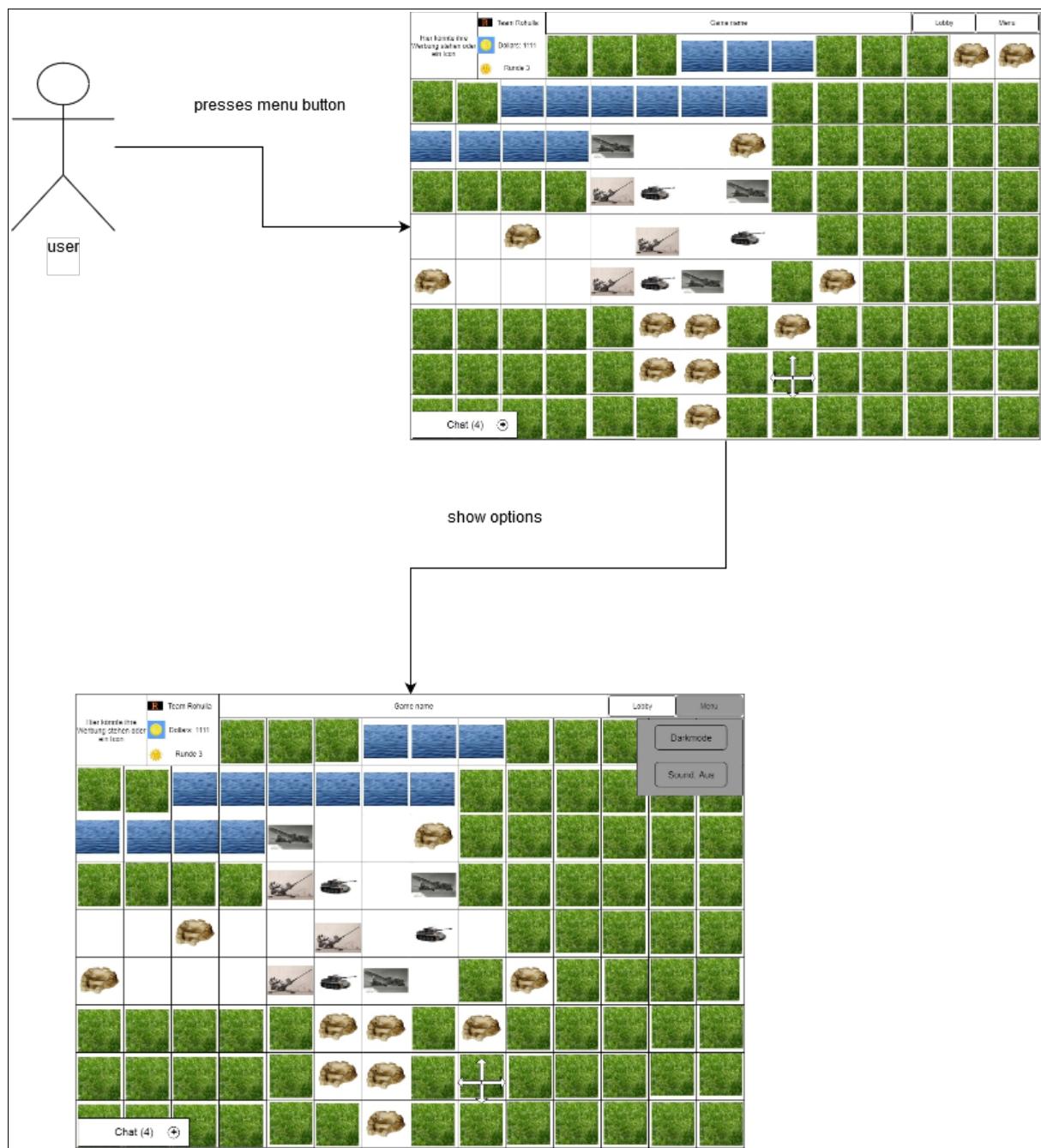
## Spiel starten

Nachdem der Benutzer in dieser Story zehn Panzer ausgewählt hat, drückt er den „Start“ - Knopf. Die Informationen der Armeekonfiguration werden an den Server geschickt. Daraufhin schickt der Server die Informationen über das initiale Spielgeschehen an den Client des Benutzers und dieser erzeugt dann dieses Geschehen.



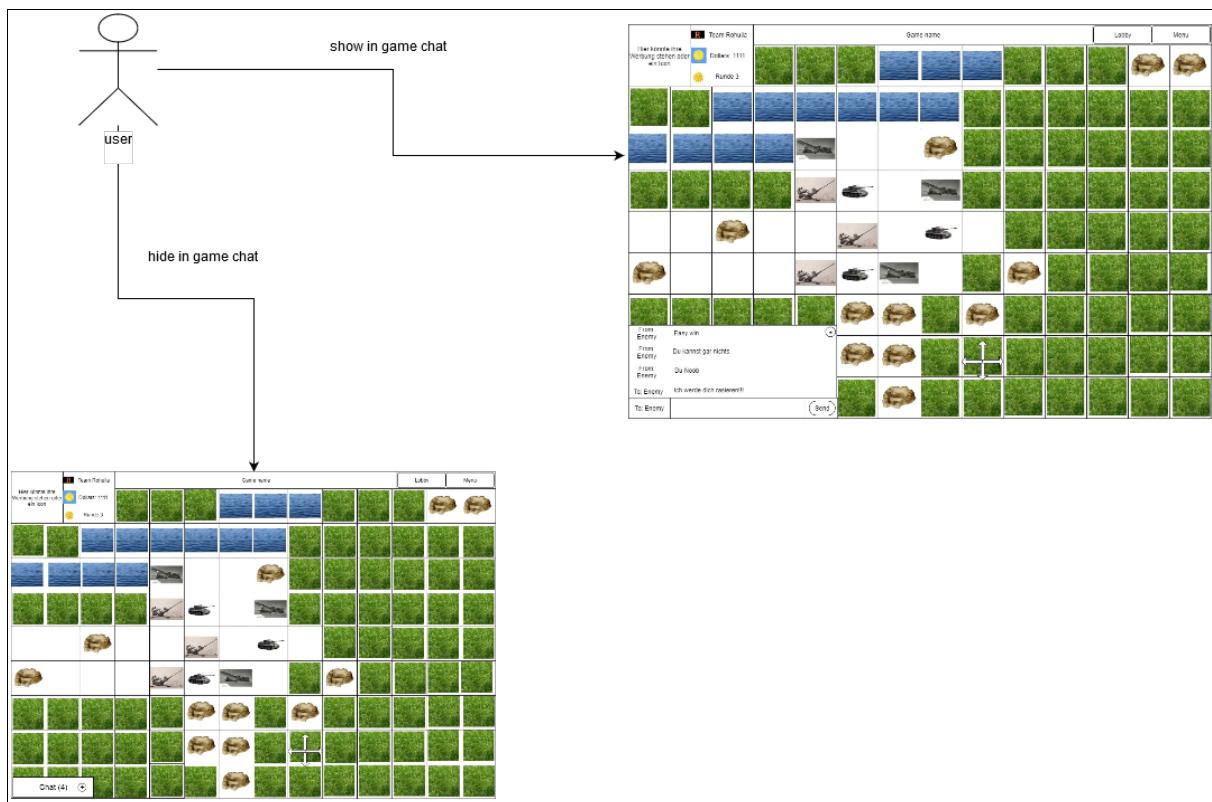
## Menü Knopf

Der Benutzer klickt auf den „Menu“ - Knopf und es öffnet sich das Menü.



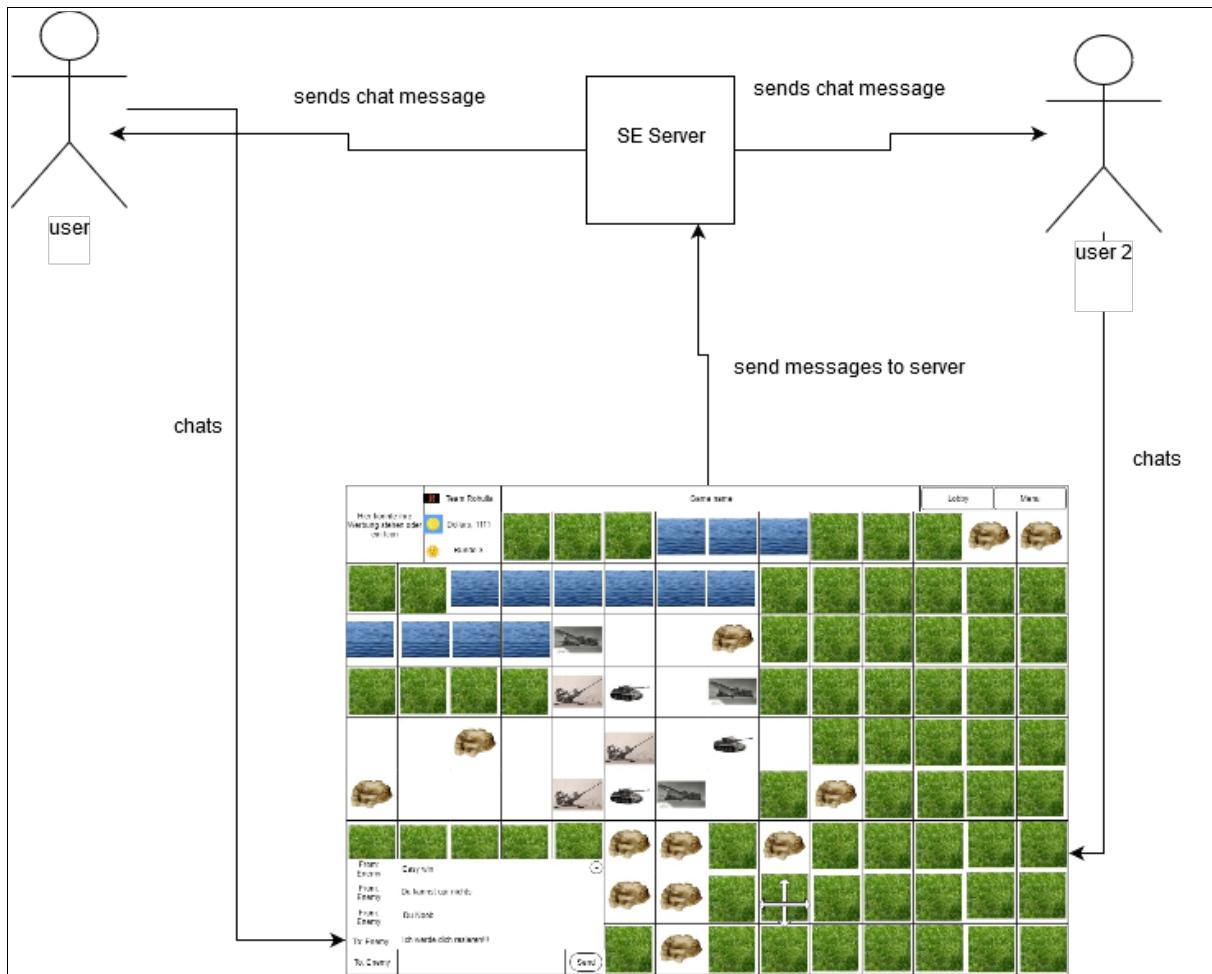
## Chat minimieren

Hier sieht man, dass der Benutzer den Chat im Spielgeschehen verkleinern und vergrößern kann.



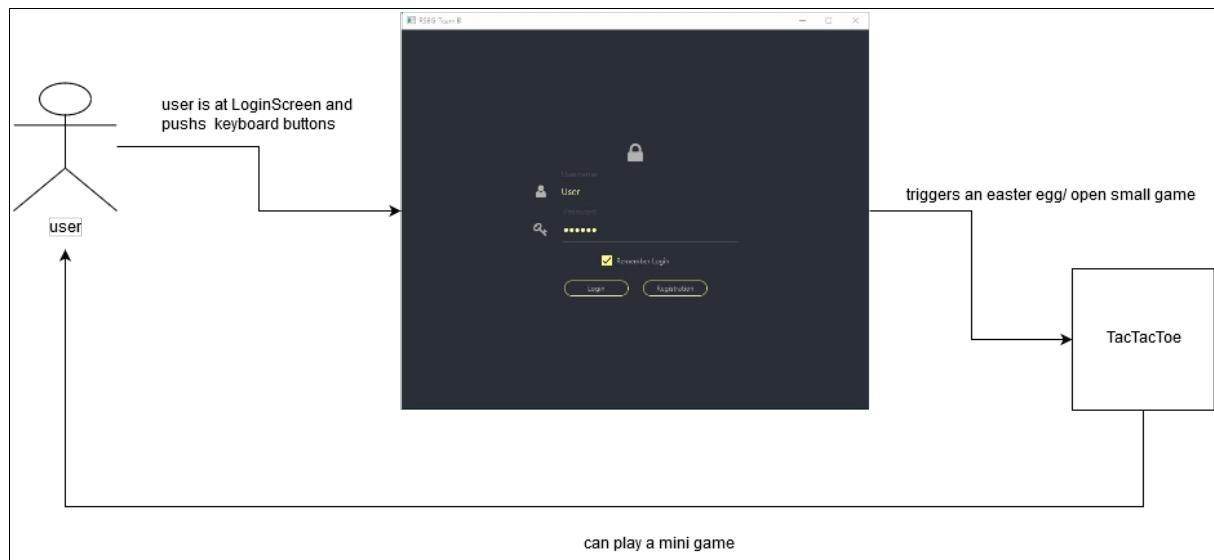
## Chat benutzen

Der Benutzer will von seinem Client eine Nachricht an den Client eines anderen Spielers schicken, also klickt er den „Send“ Knopf. Der Client sendet die Nachricht an den SE Server, welcher dann die Nachricht an den Client des gewünschten Spielers schickt.



## TicTacToe spielen

Mit einer Tastenkombination oder einem Knopfdruck öffnet sich ein Fenster in dem man TicTacToe spielen kann.



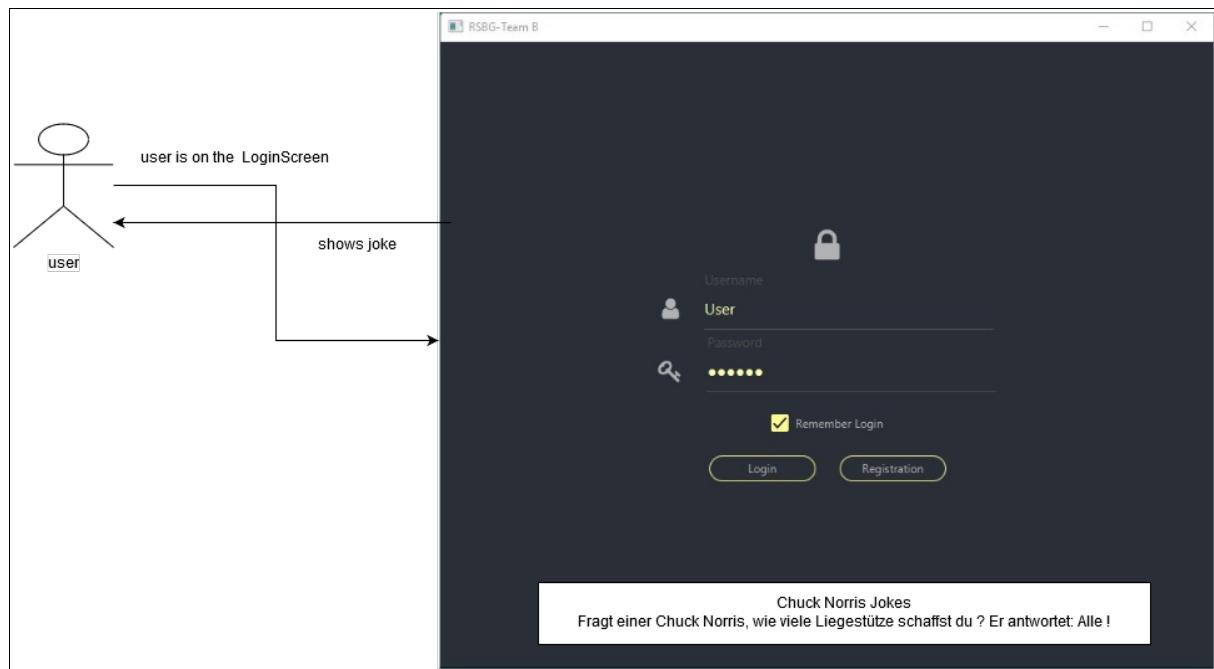
## Snake spielen

Mit einer Tastenkombination oder einem Knopfdruck öffnet sich ein Fenster in dem man Snake spielen kann.



## Chuck Norris Witze

Bei Inaktivität im Login Bildschirm tauchen am unteren Rand verschiedene Chuck Norris Witze auf und verschwinden nach einiger Zeit wieder.



## Sprints

Dieses Release ist das zweite und enthält also den dritten und den vierten Sprint, da jedes Release zwei Sprints umfasst.

### 3. Sprint

Das Ziel dieses Sprints war es den Armeemanager zu implementieren, sowie die Grundlagen für das Spielgeschehen zu schaffen und darauf zu achten, dass mehr als 60% des Codes durch Tests abgedeckt werden.

Der Sprint lief vom 9. Juni 2019 22:25 bis zum 20. Juni 2019 14:33 Uhr, da der Scrummaster den Sprint aus Versehen nicht nach den vollen zwei Wochen enden ließ, sondern schon vorher. Es gab 26 Issues in Jira mit insgesamt 79 Storypoints auf fünf Entwickler. Zusätzlich gab es noch Tasks und Bugfixes, die Zeit in Anspruch genommen haben, die allerdings nicht in den Storypoints reflektiert werden.

Im zweiten Sprint sind folgende Stories nicht fertig geworden und wurden deswegen in diesen Sprint übernommen.

### **Vom letzten Sprint übernommen**

#### **TB – 10 Select Chat Target**

Schätzung: 7,5

Beschreibung: As a user on the main screen I want to be able to click on buttons named as existing players or "all" to set the chat's target to the corresponding value.

Für diesen Sprint wurde die Beschreibung geändert zu:

Start

Der Benutzer ist in der Lobby, er kann die Spielerliste, sowie die Liste der aktuellen Spiele sehen. Darüber hinaus ist es ihm möglich mit anderen Spielern zu chatten, ein neues Spiel zu generieren oder sich wieder auszuloggen.

Aktion

Der Spieler klickt auf einen Knopf in der Spielerliste.

Ende

Nun hat der Benutzer einen anderen Benutzer zum chatten ausgewählt.

Die andere Story, die nicht rechtzeitig geschafft wurde, war:

## **TB – 11 View Chat Target**

Schätzung: 7,5

Beschreibung: As a user on the main screen I want to be able to see the current chat target as a chat tab to decide whether it is correct or not.

Die Beschreibung wurde geändert zu:

Start

Der Benutzer ist in der Lobby, er kann die Spielerliste, sowie die Liste der aktuellen Spiele sehen, darüber hinaus ist es ihm möglich mit anderen Spielern zu chatten oder ein neues Spiel zu generieren oder sich wieder auszuloggen.

Aktion

Der Benutzer möchte den aktuellen Chat mit dem gewünschtem Benutzer bzw. den gesamten Chat zu allen Benutzern (Spielern) in einer Registerkarte (menu) sehen.

Ende

Dem Benutzer wird eine Registerkarte angezeigt, mit dem gesamten Chat zu allen Benutzern. In einer anderen Registerkarte wird der vorher auserwählte Benutzer (Spieler) zum chatten angezeigt.

Die originale Beschreibung ließen wir in Klammern unter der neuen im Jira, damit es nicht zu Verwirrungen kommt.

Diese Stories wurden in die Aufgabenverteilung des Sprints mit einberechnet, sodass alle Aufgaben der Entwickler zeitlich zu schaffen waren. Des weiteren hat die Leitung des letzten Releases noch Aufgaben erstellt, die sie in ihrem Release nicht geschafft haben aber noch nötig sind. Diese wurden in den dritten Sprint mit aufgenommen.

## **TB – 87 Code Cleanup**

Schätzung: 3

Beschreibung:

As a developer I want the code to be clean which means fewest IDE hints about unclean codestyle so that I can focus on my main tasks, that is kept cleaner in turn.

Der Entwickler hat für diesen Task fünf Stunden gebraucht. Er meinte, es habe etwas länger gedauert, da er auch einige Klassen bearbeitet habe um deren Funktion zu verbessern.

## **TB – 88 Fat Jar**

Schätzung: 2

Beschreibung:

As a user I want to be able to click on a single jar file without dependecies so that the application starts wherever I like.

Auch dieser Task hat geringfügig länger gedauert als von den letzten Leitern erwartet. Der Entwickler nutzte drei Stunden um die Aufgabe zu bewältigen, da er mehrere Methoden versucht habe und dafür noch Teile der Gradle Dokumentation habe lesen müssen.

## **Stories und Tasks**

Folgende Aufgaben wurden neu erstellt und in den ersten Sprint des zweiten Releases mit aufgenommen. Für alle Aufgaben waren vorhandene Mockups und Domain-, sowie Userstories im Jira vorhanden.

## **TB – 53 Basic FXML Armymanager**

Schätzung: 7

Beschreibung:

Screen layout files for the Armymanager needs to exist, so that GUI elements can be added. The mainscreen should be bigger and be able to be in fullscreen.

Start

Der Benutzer ist in der Lobby und möchte seine Armee erstellen.

Aktion

Der Benutzer drückt auf einen Knopf und gelangt zum Armeemanagers.

Ende

Dem Benutzer wird eine Grundstruktur des Armeemanagers angezeigt, ohne die zu konfigurierbaren Einheiten.

Diese Story wurde auf sieben Stunden geschätzt, da es noch Unteraufgaben wie die Vollbildfunktion und die Anpassung de Lobby gab. Design und FXML können oft auch länger dauern, da sich die Wahrnehmung von Schönheit oft unterscheidet und einige Änderungswünsche erwartet wurden. Der Entwickler gab für die ganze Aufgabe allerdings

nur drei Stunden und 45 Minuten benötigte Zeit an. Er begründete die vorzeitige Fertigstellung mit seiner Erfahrung und schneller Übereinkunft bei Designfragen mit den Verantwortlichen. Außerdem half das zur Verfügung gestellte Mockup von Seite 4 bei der Bearbeitung der Aufgabe.

## TB – 55 Save Configuration

Schätzung: 5

Beschreibung:

Start

Der Benutzer ist im Armeemanager, dieser hat die Möglichkeit seine Armee zu konfigurieren.

Aktion

Der Benutzer drückt auf einen Knopf, der die Armee speichert und danach auf einen anderen, der die Knöpfe umwandelt.

Ende

Die Armeekonfiguration ist gespeichert und es werden Ladeknöpfe anstelle der Speicherknöpfe angezeigt.

(The buttons for saving the Armyconfiguration need to be added and implemented. The configuration should be saved locally and on the server. The button for changing to load configuration should also be added. Use the existing class for serialization.)

Auch diese Story wurde wesentlich schneller bearbeitet als erwartet. Die Schätzung war fünf Stunden, der Entwickler brauchte nur eine. Der Entwickler begründete, dass es nicht so viel Arbeit gewesen wäre und er bereits Bausteine bei einer anderen Aufgabe baute, was zu einer erheblichen Vereinfachung geführt habe.

Für die Story standen die dazugehörigen Mockups des Armeemanagers, sowie die Domainstory von Seite 14.

## **TB – 56 Load Configuration**

Schätzung: 5

Beschreibung:

Start

Der Benutzer ist im Armeemanager, dieser hat die Möglichkeit seine Armee zu konfigurieren.

Aktion

Der Benutzer drückt auf einen Knopf und kann somit seine Armeekonfiguration laden,. Er kann auch auf einen anderen Knopf drücken um zwischen Lade- und Speicherfunktion zu wechseln.

Ende

Der Benutzer hat seine (zuvor gespeicherte) Konfiguration geladen oder wechselt zur Speicherfunktion.

(Implement the change to load button and the buttons for loading the configuration. Try to load the configuration locally and if it doesn't exist, request it from the server. Use the existing class for serialization.)

Hier war der Entwickler ebenfalls sehr schnell. Er brauchte eine Stunde und vierzig Minuten für diese Story, da er viel der Vorarbeit bereits bei TB – 89 erledigt hatte und diese Zeit auch für diesen Task angegeben hat. Dadurch ist die Zeitverteilung bei den Aufgabe TB – 55, TB – 56 und TB – 89 vermischt und müsste zusammengerechnet werden.

Die Domainstory von Seite 15 war im Jira verfügbar.

## **TB – 57 Increase and Decrease Units**

Schätzung: 4

Beschreibung:

Start

Der Benutzer ist im Armeemanager. Hier kann er seine Armee konfigurieren. Er hat eine maximal vorgegebene Anzahl Einheiten und ihm wird eine Liste von der Art der Einheiten dargestellt. Ihm stehen Panzer, Flugzeuge und eine Abwehr zur Verfügung. Per Plus - oder Minus - Knopf kann er die Anzahl der gewünschten Einheiten erhöhen oder verringern.

Aktion

Er drückt zehn mal einen beliebigen Plus – Knopf,

Ende

Der Benutzer hat sich eine Armee ausgewählt.

(Button for increase and decrease of the units must be created. Also on click or enter the available and chosen units should change. There are 10 available Units.)

Nach drei Stunden und zehn Minuten Arbeit konnte der Entwickler diese Story fertigstellen und kam damit relativ genau an die Schätzung. Es gab keine Probleme bei der Bearbeitung und der Entwickler hatte sogar noch Zeit ein kleines Zusatzfeature zu implementieren. Alle Knöpfe, die nichts mehr verändern werden ausgegraut um diese Tatsache zu verdeutlichen.

## **TB – 58 Show Configuration**

Schätzung: 4

Beschreibung:

Start

Der Benutzer ist im Armeemanager. Hier kann er seine Armee konfigurieren bzw. eine alte Konfiguration laden. Ihm stehen dafür der Ladeknopf und der Speicher - Knopf zur Verfügung.

Aktion

Der Benutzer drückt mit Rechtsklick auf den Ladeknopf bzw. Speicher - knopf. Es öffnet sich ein kleines Fenster.

Ende

Dem Benutzer wird angezeigt (im Fenster), welche Konfiguration zum Laden vorhanden, bzw. welche Konfiguration gespeichert ist.

(The save and load buttons should show the configuration that they saved/load on right click.)

Diese Story wurde ziemlich genau geschätzt. Der Entwickler benötigte drei Stunden und 40 Minuten um die Funktion zu implementieren und wickelte damit nur mit 20 Minuten von der initialen Schätzung ab.

## **TB – 61 Buttons**

Schätzung: 2

Beschreibung:

Start

Der Benutzer ist im Armeemanager oder im Spiel selbst, er möchte wieder zurück zur Lobby gelangen. Im Spiel selbst ist ein Menü vorhanden.

Aktion

Er drückt auf einen Knopf und kommt wieder zur Lobby zurück. Beim drücken auf das Menü werden verschiedene Aktionen angezeigt.

Ende

Der Benutzer ist wieder in der Lobby oder hat das Menü im Spiel geöffnet.

(Add and implement the button, that gets you out of the Armymanager and out of the game back to the mainScreen.

Also add a menubutton ingame similar to the cog menu button in the mainscreen.)

Auch bei dieser Story war die Schätzung recht genau. Der Programmierer brauchte zweieinhalb Stunden für die Story. Er habe etwas länger benötigt, da er sich nach eigenen Angaben noch nicht so gut mit dem vorhandenen Code auskenne.

## **TB – 63 Basic FXML InGame**

Schätzung: 3

Beschreibung:

Start

Der Benutzer ist in der Lobby und möchte das Spielfeld sehen / Spiel starten.

Aktion

Der Benutzer drückt in der Lobby auf den Start - Knopf.

Ende

Es wird dem Spieler die Grundstruktur des Spielfeldes mit Platz für Chat, Name usw. angezeigt.

(Create basic FXML for the ingame screen so that GUI elements can be added. The screen should be able to go into fullscreen modus.)

Diese Story wurde in dreieinhalb Stunden abgearbeitet. Die Schätzung war sehr genau, da der Bearbeiter die zusätzliche halbe Stunde für Probleme mit Travis angab. Er kannte die Travis Funktionen noch nicht und habe Zeit benötigt um Fehlermeldungen zu beheben. Für die Aufgabe stand das Mockup des Spielgeschehens im Jira zur Verfügung, damit der Entwickler genau weiß, welche FXML Elemente gebraucht werden.

## TB – 64 InGame Chat

Schätzung: 6

Beschreibung:

Start

Der Benutzer ist im Spiel. Er möchte mit andern während des Spieles chatten können.

Aktion

Der Benutzer klickt auf einen Knopf im Chatfenster und es wird groß angezeigt, beim klicken auf einen anderen Button wird es wieder kleiner angezeigt.

Ende

Der Benutzer kann nun chatten und zwischen großem und keinem Chatfenster per Klick auf einen Knopf ändern.

(Add and implement the ingame chat. All necessary components for the chat already exist. The chat should have a small mode and a big mode. You can only send and read messages in big mode but you can see the amount of sent messages in small mode.)

Die Bearbeitung dieser Aufgabe betrugen fünfeinhalb Stunden und waren damit auch sehr nah an der ursprünglichen Schätzung. Im Jira sind Mockup und Domainstories für diese Funktion vorhanden.

## **TB – 65 Transitions**

Schätzung: 2

Beschreibung:

Start

Der Benutzer ist im Armeemanager bzw. im Spiel selbst. Er möchte diesen nun verlassen bzw. zum Armeemanager zurückgelangen.

Aktion

Er drückt auf den „Zurück“ - Knopf und gelangt zum Armeemanager bzw. auf den „Start Game“ - Knopf und startet das Spiel.

Ende

Durch eine Transition werden die Aktionen ausgeführt, nun ist der Benutzer im Spiel oder im Armeemanager.

(Add and the start game button in the Armymanager and the Lobby button ingame. Use existing fades to tranistion from one screen to another.)

Die benötigte Zeit für diese Story wurde überschätzt. Der Entwickler stellte die Aufgabe in gerade mal zehn Minuten fertig, da die Funktionen, die für diese Aufgabe verwendet wurden, alle schon zur Verfügung standen. Außerdem wären sie sehr leicht zu benutzen gewesen, wodurch der Entwickler sehr schnell sein konnte.

## **TB – 68 TicTacToe**

Schätzung: 7

Beschreibung:

Start

Der Benutzer ist in der Lobby und wartet auf andere Benutzer.

Aktion

Der Benutzer drückt auf eine bestimmte Tastenkombination.

Ende

Es öffnet sich ein Fenster und er kann TicTacToe spielen.

(Create small window where the user can play TicTacToe.)

Statt der geschätzten sieben Stunden, beendete der Entwickler die Aufgabe nach zehn. Nach seinen Angaben habe er nur länger gebraucht, da er langsam sei.

## TB – 71 Whitemode

Schätzung: 5

Beschreibung:

Start

Der Benutzer ist im Anmeldebildschirm/ Lobby/ Spiel und möchte das Design ändern.

Aktion

Der Benutzer drückt auf einen Knopf.

Ende

Die Farben ändern sich von dunkel zu hell.

Create application stylesheets that theme a white background with dark text and input elements with a nice design. To fade scenes, use slideNextScene funktion in util. Buttons need to be added in the menus.

Bei dieser Aufgabe hat sich das Team massiv verschätzt. Der verantwortliche Programmierer nahm 15,5 Stunden, statt der geschätzten fünf in Anspruch um die Aufgabe zu erledigen. Die erste halbe der 15,5 benötigten Stunden, verwendete der Programmierer um existierende Stylesheets zu kopieren und die Farben zu verändern. Danach gab er drei Stunden an um herauszufinden, wie man diese Stylesheets zu Laufzeit verändert. Weitere drei Stunden wurden verwendet um das Hamburgermenü zu implementieren und die Knöpfe um zwischen dem hellen und dem dunkeln Modus zu wechseln. Die nächsten vier Stunden habe er einige angefallene Probleme behoben. Durch die ganzen Änderungen soll es dann einen großen Merge - Konflikt und Travis - Probleme gegeben habe, die auch drei Stunden in Anspruch genommen haben. Zuletzt brauchte der Programmierer noch zwei Stunden um einige Fehler und Stilfehler zu beheben.

Die Bearbeitung der Aufgabe dauerte viel länger aber der Programmierer nahm sich die Zeit um sie komplett zu beenden und hat vorbildlich seine Pflichten erledigt.

## **TB – 78 Textur-Manager**

Schätzung: 7

Beschreibung:

Modul, bei dem per String - Queries Texturen für Units / Terrain als JavaFX - Pane angefragt werden können. Diese werden zu Programmstart automatisch von der Festplatte in den RAM geladen.

Der Entwickler dieser Aufgabe brauchte nur drei der sieben verfügbaren Stunde um sie zu Ende zu bringen. Er habe genau gewusst und vorher geplant was, er machen wolle, wodurch er viel schneller als erwartet fertig gewesen wäre.

## **TB – 79 Sound-Manager(manage part)**

Schätzung: 5

Beschreibung:

Modul, bei dem per String-Query Sound Objekte angefragt und geladen werden. Hier wird es auch eine Option geben, die Sounds Live von der Festplatte zu laden, um zu vermeiden, Soundfiles >100MB im RAM zu halten.

Bei dieser Aufgabe gab der Entwickler an, dass es erstaunlich leichte Bibliotheken gebe, wodurch er nur zwei Stunden brauchte um sie zu erledigen.

## **TB – 80 Animated textures**

Schätzung: 7

Beschreibung:

Erweiterung von TB - 78 und dem Texturmanager um animierte Texturen zu ermöglichen.

Rendering sei eines der Spezialgebiete dieses Entwicklers, weswegen er nur zweieinhalb Stunden nutzte um diese Aufgabe fertigzustellen.

## **TB – 81 Create Unit Textures**

Schätzung: 8

Beschreibung:

Erweiterung von TB - 78 und dem Texturmanager um Units hinzuzufügen.

Das Erstellen der Texturen ging auch wesentlich schneller als erwartet, da der Entwickler doch keine Pixelart sondern 3D – Rendering benutzte. Dadurch wurden aus den erwarteten acht Stunden nur dreieinhalb.

## **TB – 83 Sound – Manager(IO)**

Schätzung: 5

Beschreibung:

Für den Soundmanager eine IO-API, die dann Sounds lädt, dekodiert und abspielt.

Der Grund, dass zu Beendigung dieses Tasks nur zweieinhalb Stunden benötigt wurden ist der selbe wie bei TB – 79. Die Bibliotheken seien erstaunlich leicht gewesen.

## **TB – 85 Test UI/Util**

Schätzung: 4

Beschreibung:

Es müssen Test für die Klassen in den Packages UI und Util geschrieben werden, da diese noch nicht ausreichend bzw. überhaupt noch nicht getestet wurden. Es müssen mindestens 60% Code Coverage erreicht werden.

Für diesen Task brauchte der Entwickler wesentlich länger als erwartet, da es massive Probleme mit dem kopflosen Modus beim Testen gab, was viel Zeit in Anspruch genommen habe.

## **TB – 86 Request Armysaves**

Schätzung: 2

Beschreibung:

Start

Der Benutzer ist im Armeemanager und möchte seine Armeekonfiguration vom Server speichern bzw. laden.

Aktion

Er drückt auf den Speicher- bzw. Ladeknopf.

Ende

Es wird die gewünschte Konfiguration vom Server geladen bzw. im Server gespeichert.

(To save and load Armyconfigurations from the server, a class to sent and request the data need to be written. Integrate the code in the exististing requests framework.)

Der Entwickler dieses Tasks berichtete, dass er mit drei Stunden und 40 Minuten länger brauchte als erwartet, da er noch Funktionen habe ergänzen müssen, um die Konfigurationen anzuzeigen.

## **TB – 89 Server Requests**

Schätzung: 6

Beschreibung:

Klassen für die Server Requests müssen geschrieben werden. Für folgende Funktionen:

POST /army  
GET /army  
GET /army/id  
PUT /army/id  
DELETE /army/id  
GET /army/units

Wie bereits erwähnt müssten die Zeiten von TB – 55, TB – 56 und TB – 89 addiert und dann mit der Summe der Schätzungen verglichen werden, da der Entwickler die Aufgaben vermischt und dadurch die Zeitangaben inkorrekt sind. Für diesen Task wurden zehn Stunden und 50 Minuten angegeben. Für TB – 55 eine Stunde und für TB – 56 eine Stunde und 40 Minuten. Zusammengerechnet ergibt das 13 Stunden und 30 Minuten. Die Summe der

Schätzungen sind 16 Stunden. Also war der Entwickler insgesamt zweieinhalb Stunden schneller als erwartet, was bei einer so hohen Schätzung relativ genau ist. Er war etwas schneller, da für die Requests schon ein Framework existierte und die meiste Zeit dafür verwendet wurde die neuen Request – Klassen für die benötigten Funktionen für die Bearbeitung von TB – 55 und TB – 56 anzupassen.

## **TB – 91 Logger**

Schätzung: 4

Beschreibung:

In jeder Klasse, in der der Logger zum Einsatz kommt wird nur logger.error(e) aufgerufen, dieser Aufruf muss mit logger.error(e.getStackTrace) ersetzt werden, damit in der LogDatei die komplette Fehlermeldung zur Fehlerbehebung vorhanden ist.

Mit drei benötigten Stunden war der Entwickler etwas schneller als erwartet aber die Schätzung war trotzdem recht genau.

## **TB – 92 PlayerList update**

Schätzung: 1

Beschreibung:

Die jetzige PlayerList zeigt nur die vorhandenen Player beim Start des Programms, es werden keine neu dazugekommenen Spieler angezeigt.

Die vorhandenen Methoden zum erstellen der PlayerList in dem MainController müssen ergänzt werden.

Wie erwartet, benötigte der Verantwortliche genau eine Stunde für diese Aufgabe.

## **Bugs**

Zusätzlich zu den Stories und Tasks gab es diesen Sprint drei Bugs die behandelt wurden.

## **TB – 90 Handshake Error**

Beschreibung:

Start

Wenn man ein neues Spiel erstellt und sich danach abmeldet taucht ein Fehler auf.

Aktion

Es muss die Fehlerursache gefunden werden und der Fehler behoben werden.

Zu überprüfen ist die WebSocketClient Klasse.

Die vollständige Fehlermeldung findet man im Logger.

Ende

Fehler gefunden und beseitigt.

Dieser Fehler wurde innerhalb von einer halben Stunde gefunden und behoben und entstand durch ein falsch gesetztes statisches Feld.

## **TB – 93 Loading saved LoginData**

Beschreibung:

Start

Benutzer ist im Anmeldeschirm und wechselt zum Registrierungsbildschirm und möchte sich registrieren.

Aktion

Benutzer registriert sich und möchte sich in Anmeldeschirm anmelden.

Ende

Die gespeicherte Logindaten werden nicht beim verlassen des Registrierungsbildschirms und zurückkehren auf den Anmeldeschirm nicht geladen.

Für diesen Bug benötigte der Programmierer zwei Stunden und 20 Minuten, da er die Aufgabe nutze, um noch den Code zu verschönern und verständlicher zu gestalten. Die gewünschten Änderungen wurden erledigt und zusätzlich noch das Feature implementiert, sodass die Daten aus dem Registrierungsbildschirms gespeichert werden.

## **TB – 94 Errorhandler Input**

Beschreibung:

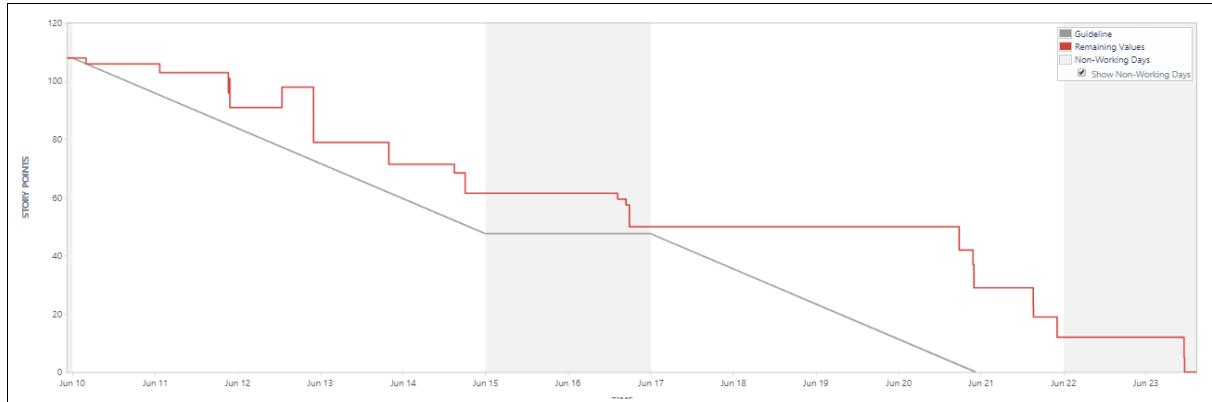
Man unterscheidet zwischen verschiedenen Fehlermeldungen. Zum Einen, die durch falsche Benutzereingaben entstehen und andere, die aufgrund von Fehlern im Client auftauchen.

Für die jeweiligen Fehler werden andere Ausgaben mit anderen Farben für die Unterscheidung angezeigt.

Bsp. Fehler sind Rot, Warnungen Gelb und Infos blau/weiß.

Diese Funktion hätte vorher schon funktionieren sollen. Da das allerdings nicht der Fall war, wurde sie noch einmal als Bug hinzugefügt. 50 Minuten wurden von dem Verantwortlichen gebraucht um diesen zu reparieren.

## Burndown – Diagramm



Am Ende des 3. Sprints ergab sich dieses Burndown – Diagramm. Wie bereits erwähnt wurde der Sprint nicht mit dem richtigen Endzeitpunkt eingestellt, wodurch die Erwartungslinie etwas zu früh endet und nicht genau durch das eigentlich Diagramm verläuft. Diese Tatsache schadet der Anschaulichkeit allerdings nicht.

Es ist zu sehen, dass die Entwickler die zwei Wochen durchgehend an dem Projekt gearbeitet haben und alle gestellten Aufgaben beenden konnten.

Am 12. Juni wurden zwei Stories dem Sprint hinzugefügt, da einem Entwickler eine zu dem Zeitpunkt nicht machbare Aufgabe zugewiesen wurde und seine anderen Aufgaben bereits erledigt hatte. Die nicht machbare Aufgabe war TB – 82. Es gab noch keine Informationen über die Terrain Felder, weswegen diese Aufgabe am 20. Juni aus dem Sprint entfernt wurde.

Im Zeitraum vom 17. bis zum 20.Juni wurde keine Story erledigt, aber die Entwickler schafften in dieser Zeit Tasks und Bugfixes, welche allerdings nicht in dem Diagramm angezeigt werden.

## Zum Abschluss

Der erste Sprint der zweiten Veröffentlichung verlief einwandfrei. Das Burndown – Diagramm sieht beispielhaft aus und die Fertigstellung aller Aufgaben ist durch eine hervorragende Leistung und Zusammenarbeit des gesamten Teams geschafft worden. Es ist eindeutig zu sehen, dass es eine gute Idee ist, die Aufgabenverteilung über den kompletten Sprint von vorneherein zu bestimmen. Dadurch war es für alle Entwickler direkt zum Anfang des Sprint möglich auch zu arbeiten und es wurde garantiert, dass alle Aufgaben möglichst schnell geschafft werden. Keiner musste mehr bis zum nächsten Teamtreffen warten, falls er seine Aufgaben schon schnell erledigen konnte und vorarbeiten möchte. Zudem konnte damit weitgehend vermieden werden, dass einige Entwickler viel mehr Arbeiten müssen als andere und es zu Unstimmigkeiten kommt.

Alles in allem verlief der Sprint also sehr erfolgreich und es bleibt viel Zeit für extra Features, da die Entwickler viel schafften.

## 4. Sprint

In diesem Sprint stand das Spielgeschehen in Fokus. Der Armeemanager war größtenteils implementiert, aber es mussten noch einige Bugs repariert werden. Durch gute Vorarbeit blieb noch Zeit, die für zusätzliche Funktionen investiert wurden.

Der Sprint startete am 23. Juni 8:00 Uhr und lief bis zum 7. Juli 9:23 Uhr. Es gab insgesamt 28 Aufgaben in Form von Stories, Tasks und Bugs, wovon schließlich 25 geschafft und drei in den nächsten Sprint aufgenommen wurden. Von den 63 vorhandenen Storypoints wurden damit 61 abgearbeitet.

### **Tasks und Stories**

Auch hier standen wieder die Mockups, Domain- und Userstories zu Verfügung, die bei den ersten Ideen zu sehen sind.

#### **TB - 82 Create Terrain Textures**

Schätzung: 8

Beschreibung:

Hier werden die Texturen für die verschiedenen Terrain - Tiles gezeichnet. Es gibt eine Vielzahl von verschiedenen Texturen für passende Terrains.

Diese Aufgabe hat mehr Zeit in Anspruch genommen als erwartet. Obwohl der Entwickler sehr begabt in diesem Feld ist, hat er extra Zeit investiert um die Aufgabe phantastisch zu bewältigen um nur Texturen der höchsten Qualität abzugeben. Er nutzte insgesamt 14 Stunden und 30 Minuten, um ästhetisch schöne Texturen zu erstellen und dafür zu sorgen, dass es nahtlose Übergänge zwischen den gleichen, sowie anderen Texturen gibt. Außerdem fügte er eine Rendering Technik hinzu, wodurch die Kästchen flüssiger ineinander überlaufen und das Spielfeld nicht so steif wirkt. Es stellte sich heraus, dass diese ganzen Punkte eine größere Herausforderung darstellten, als zunächst gedacht.

## **TB - 96 ShowJoinedPlayer**

Schätzung: 2

Beschreibung:

Start

Benutzer hat das Spiel gestartet und ist nun im Spiel.

Aktion

Benutzer spielt gerade.

Ende

Es wird oben links sein Name und seine Farbe angezeigt.

Außerdem werden neu dazugekommene Spieler angezeigt, die Anzeige verschwindet nach ein paar Sekunden.

Der Entwickler kannte sich bereits mit dem Verfahren aus, wodurch er nur 30 Minuten für die Aufgabe benötigte.

## **TB - 99 Server Messages**

Schätzung: 3

Beschreibung:

Start

Benutzer hat ein Spiel erstellt und ist im Armeemanager.

Aktion

Benutzer klickt auf den "Start Game"-Button.

Ende

Der Server schickt das initiale Spielgeschehen. Der Client empfängt die Nachricht und verarbeitet diese zu einem für diesen Client verständliches Modell.

Auch die Bewältigung dieser Aufgabe dauerte länger als geschätzt. Der Entwickler gab als einen Grund an, dass es lange gedauert habe den teils unverständlichen Code von anderen Entwicklern zu verstehen. Bei dem Code, den er benutzt hat, nahm er dann einige Änderungen vor, was auch Zeit kostete. Allerdings konnte er die Aufgabe schließlich nach fünf Stunden und 20 Minuten beenden.

## **TB - 100 Show initial Game**

Schätzung: 15

Beschreibung:

Start

Benutzer hat ein Spiel erstellt und ist im Armeemanager.

Aktion

Benutzer klickt auf den "Start Game"- Knopf.

Ende

Das initiale Spielgeschehen wird dem Nutzer angezeigt mit vorhandenen Texturen.

Diese Aufgabe wurde nach 14 Stunden und 36 Minuten abgeschlossen und wurde damit sehr genau geschätzt für so eine hohe Zeitangabe. Der Bearbeiter hatte kein Kommentar hinzuzufügen, es lief also alles glatt und es kam zu keinen unerwarteten Entwicklungen.

Mit dieser Aufgabe wurde die Mindestanforderung des initialen Spielgeschehens komplett erfüllt. Die Vorarbeit dabei war die Verarbeitung des Nachrichten des Servers mit TB - 99 und das Erstellen der Texturen aus TB – 80, TB – 81 und TB – 82.

## **TB - 101 ChatWindow Movements**

Schätzung: 8

Beschreibung:

Start

Der Benutzer ist gerade in einem Spiel und spielt (InGame).

Aktion

Der Benutzer ändert die Position des Chatfenster und kann die Größe des Chats verändern.

Ende

Der Chat ist an einer anderen Position und die Größe hat sich auf die vom Benutzer eingestellte Größe geändert.

Bei dieser Aufgabe wurde sich mächtig verschätzt. Der Entwickler gab 16 Stunden und 30 Minuten zur Fertigstellung an.

Die ersten eineinhalb Stunden verwendete er um eine neue Klasse zu erstellen mit der man die Größe des Chats verändern können sollte und testete die Resultate. Die nächsten eineinhalb Stunden nutze er dann um neue Berechnungen zu erwähnter Klasse hinzuzufügen. Danach

arbeitete er nach eigenen Angabe zweieinhalb Stunden an einer vernünftigen Logik für die Größenänderung, sowie der Funktion das Chatfenster zu verschieben und dem Hinzufügen von Schatten bei den Bewegungen. Weitere zwei Stunden verwendete er um die Berechnungen bei die Größenänderungen erneut zu überarbeiten, Schatten zu reparieren und Merge - Konflikte zu beheben. Die Folgenden zwei Stunden implementierte er Änderungswünsche. Die letzten sieben Stunden reparierte er die Chats im Armeemanager und im Spiel, die durch die Änderungen nicht mehr sichtbar waren. Außerdem behob er einige Fehler die durch das Wechseln der Bildschirme beim Chat entstanden, sowie andere Fehler, die die Tests durch die Änderungen fehlschlagen ließen. Trotz einiger Rückschläge bei diese Aufgabe kam der Entwickler seinen Pflichten nach und verwendete so viel Zeit wie nötig um sie zufriedenstellend zu beenden.

## **TB - 103 Show Images in ArmyManager**

Schätzung: 2

Beschreibung:

Start

Der Benutzer ist im Armeemanager und konfiguriert seine Armee.

Aktion

Er klickt auf die Knöpfe neben den animierten Bildern der jeweiligen Einheiten um diese auszuwählen bzw. zu erhöhen oder verringern.

Ende

Der Benutzer hat seine Armee konfiguriert und sieht zu dem die animierten Bilder der Einheiten.

Diese Aufgabe konnte der Bearbeiter schnell lösen, er benötigte lediglich eine Stunde, statt der geschätzten zwei und lieferte ein zufriedenstellendes Ergebnis ab.

## **TB - 108 Configure before Game**

Schätzung: 4

Beschreibung:

Start

Benutzer ist in der Lobby und möchte das erste Mal einem Spiel beitreten.

Aktion

Er klickt auf den „Join“ - Knopf.

Ende

Benutzer ist in einem extra Armeemanager mit einem „Join Game“ Knopf.

Bei dieser Aufgabe gab es einige Kommunikationsprobleme. Es war gedacht, dass der Entwickler auch gleich den Knopf zum Beitreten implementiert, wodurch die hohe Schätzung entstand. Der Entwickler tat dies allerdings nicht und es wurde beim Einbinden in den Master nicht bemerkt. Deswegen wurde noch TB – 120 erstellt um die fehlende Funktion nachzubessern. Für die unfertige Aufgabe wurde eine Stunde Arbeit angegeben.

## **TB - 109 Minimap**

Schätzung: 4

Beschreibung:

Start

Der User ist im Spiel.

Aktion

Er öffnet das Hamburgermenü und klickt auf einen Minimap - Knopf.

Ende

Es öffnet sich am rechten unteren Rand eine Minimap.

Der Entwickler gab genau die geschätzten vier Stunden als seine benötigte Zeit an. Die Minimap befindet sich vorerst auf der linken obere anstatt der rechten unteren Hälfte, da der Entwickler aus zeitlichen Gründen es nicht geschafft hat, die Karte dynamisch zu programmieren und sie an die richtige Stelle zu setzen.

## **TB - 110 Move GameField**

Schätzung: 5

Beschreibung:

Start

Benutzer hat ein neues Spiel erstellt und hat dieses gestartet.

Aktion

Benutzer klickt auf das Spielfeld und bewegt es.

Ende

Benutzer ist immer noch im Spiel und nun hat sich das Spielfeld bewegt und man sieht eine andere Stelle der Spielkarte.

Diese Aufgabe wurde in vier Stunden und 35 Minuten fertiggestellt. Die Schätzung war also recht genau. Bei der Bearbeitung gab es keine Probleme und die gewünschte Funktion ist vorhanden.

## **TB - 111 SmallChat**

Schätzung: 3

Beschreibung:

Start

Benutzer ist in einem der Spielfenster und der Chat ist im kleinen Modus.

Aktion

Der Benutzer bekommt Nachrichten.

Ende

Dem Benutzer werden im Chat (Small-Mode) Benachrichtigungen angezeigt. Mit einer kleinen Zahl soll die Anzahl der empfangenen Nachrichten dargestellt werden.

Auch diese Aufgabe konnte innerhalb der geschätzten Zeit erledigt werden. Nach zwei Stunden und 35 Minuten stellte der Entwickler sie fertig.

## **TB - 112 NotificationHandler Color**

Schätzung: 2

Beschreibung:

Fehlernachrichten sollen in rot angezeigt werden. Informationen in blau und Warnungen in orange.

Bei dieser Aufgabe wurde keine Zeit angegeben. Sie wurde allerdings fertiggestellt und in das Programm integriert.

## **TB - 115 Update PlayerList**

Schätzung: 1

Beschreibung:

Die Spielerliste soll automatisch bei Veränderung aktualisiert werden.

Diese Aufgabe wurde in TB – 120 erledigt, da es für den Entwickler nicht separat ging.

## **TB - 116 Minimap Rendering**

Schätzung: 6

Beschreibung:

Es müsste ein JavaFX - Pane Rendern erstellt werden, die dann von der Minimap übernommen werden.

Diese Aufgabe wurde in gerade mal einer halben Stunde erledigt. Die Schätzung wich so stark von der Schätzung ab, da der Entwickler genau wusste was er machen muss und es für ihn noch leichter war als erwartet, da bereits ein Karten – Objekt zur Verfügung stand und er es für diese Aufgabe verwenden konnte.

## TB - 120 ArmyManager Start Game with Config

Schätzung: 1

Beschreibung:

Es muss noch die Funktion implementiert werden, mit der man mit der gewünschten Armee das Spiel starten kann.

Für diese Aufgabe wurde auf eine Stunde Arbeit geschätzt und genau diese Stunde protokollierte der verantwortliche Entwickler auch für die Aufgabe selbst. Allerdings gab er noch 8 Stunden und 35 Minuten für weitere Erledigungen an, die nicht direkt die Aufgabe sind, der verantwortliche Entwickler jedoch für nötig hielt.

Nach eigenen Angabe verbrachte er die ersten eineinhalb Stunden damit, „Comedy-Code“ zu dechiffrieren. Danach genannte Stunde mit der Aufgabe und weitere fünf Stunden und 50 Minuten damit den extra Armeemanager neu zu schreiben, dass es dafür nicht eine eigene FXML – Klasse mehr gibt sondern nur der normale Armeemanager mit einem extra „Join Game“ Knopf aufgerufen wird. Dabei überarbeitete er auch gleich die CSS, damit es da keine unnötigen Duplikate mehr gibt. Schlussendlich verbrachte er noch eine Stunde und 15 Minuten mit debuggen.

## Bugs

### TB - 95 NotificationHandler

Beschreibung:

Start

Benutzer ist in der Lobby, im Spiel oder im Armeemanager.

Aktion

Benutzer macht etwas, wodurch ein Fehler erzeugt wird.

Ende

Dem Benutzer wird eine Fehlermeldung angezeigt.

Dieses Feature gab es ursprünglich schon. Beim Mergen ist es anscheinend verschwunden und wurde deswegen als Bugfix hinzugefügt. Der Entwickler, der mit dieser Aufgabe anvertraut wurde, stellte es innerhalb von einer Stunde wieder her und fügte die Farberweiterung hinzu. Die Aufgabe wurde nicht zu voller Zufriedenheit fertiggestellt, musste aber aus Zeitgründen so übernommen werden.

## **TB – 105 ChatAppearence**

Beschreibung:

Im Chat ist ein langer blauer Balken, dieser müsste in einer einheitlichen Farbe zum ganzen Spiel sein d.h. Grau bzw. die Farbe vom CSS.

Dieser kleine Fehler hat sich aus dem letzten Release eingeschlichen aber war kein großen Problem. Ein Entwickler brauchte eine halbe Stunde um ihn zu beheben.

## **TB - 107 ArmyManager Chat Size**

Beschreibung:

Die Größe des Chats im Armeemanager müsste angepasst werden. Dieser soll kleiner werden, da er die Einheiten überdeckt.

Der Fehler wurde leicht behoben. Allerdings lässt sich die benötigte Zeit nicht richtig erfassen, da der Entwickler von TB – 101 diesen Fehler behob, als er den Chat reparierte.

## **TB - 113 Whitemode**

Beschreibung:

Whitemode funktioniert nicht, wenn man die Spielerlisten aktualisiert.

Auch dieser Fehler wurde in TB – 120 gelöst.

## **TB - 114 Icon in Fullmode**

Beschreibung:

Der Button zum togglen des Fullscreens wird durch den Scenenwechsel falsch.

Innerhalb von 15 Minuten konnte das Problem von einem Entwickler gelöst werden. Es gab keine Schwierigkeiten.

## **TB - 118 ArmyRequestTests**

Beschreibung:

Es müssen Test für dieses Klassen ergänzt werden.

Nach drei Stunden Arbeit hatte der Entwickler die Aufgabe erledigt und funktionierende Tests nachgereicht.

## **Nicht geschafft**

## **TB – 77 Javadoc**

Schätzung: 12

Beschreibung:

In der Javadoc werden alle Programmabläufe, Klassen, Methoden und die verwendeten Bibliotheken erklärt und das Zusammenspiel dieser bezogen auf das gesamt Projekt.

## **TB - 84 Zuendorf Meme**

Schätzung: 2

Beschreibung:

Start

Man ist in einem der verschiedenen Szenen (Lobby/Login usw.)

Aktion

Es wird eine bestimmte Tastenkombination ausgeführt.

Ende

Irgendwo taucht Zündorf (Meme) auf und trinkt Bier.

Diese beiden Aufgaben hat ein Entwickler nicht rechtzeitig vor Ende des Sprints geschafft, da er viel Zeit mit Bugs und seinen anderen Aufgaben verbrachte. Für TB – 77 Javadoc wurden jedoch bereits die Grundlagen gelegt und sollten im nächsten Release vervollständigt werden. Auch für TB – 84 stand kurz nach dem Sprint eine Pull - Request, konnte aber nicht in dieses Release eingebunden werden.

## TB - 119 TicTacToeTest

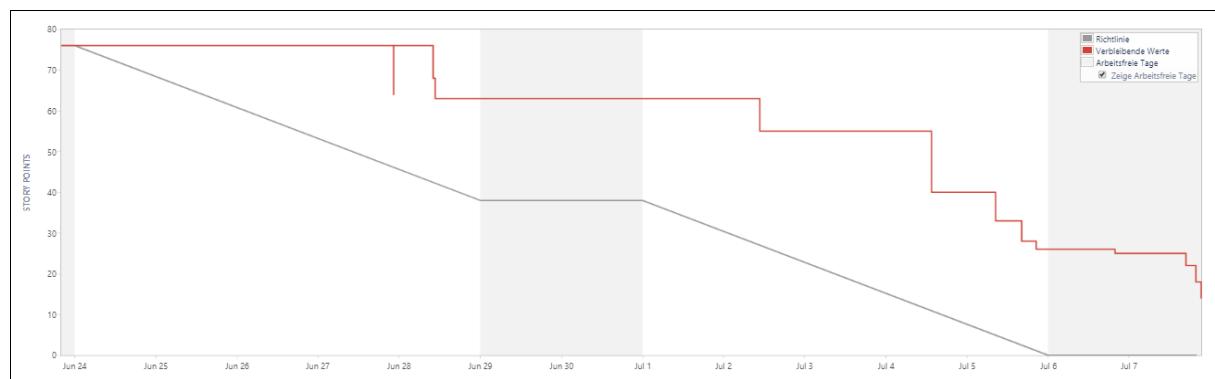
Schätzung: 2

Beschreibung:

Es müssen Test für dieses Feature geschrieben müssen, es muss eine Coverage von min. 60% erreicht werden.

Der Bearbeiter dieser Aufgabe befand sich bis kurz vor Ende des Sprints im Urlaub. Er schaffte es trotzdem allen seinen Aufgaben nachzukommen bis auf diese eine. Da auch ohne diese Tests der überprüfte Teil des Codes über 60 Prozent betrug, lief es zwar nicht optimal, aber der Ablauf war noch vertretbar.

## Burndown – Diagramm



Das Burndown – Diagramm des vierten Sprints sieht nicht ganz so gut aus wie das aus dem dritten. Es sieht aus, als wäre die erste Woche des Sprints nicht besonders viel geschafft worden. Das Diagramm beinhaltet allerdings keine Bugs oder Tasks, sondern nur Stories. Die erste Woche wurde sich besonders auf die Bugs und Stories konzentriert und wird also nicht gut in diesem Diagramm repräsentiert. Außerdem war ein Entwickler im Urlaub und konnte deswegen zum Anfang des Sprints nicht viel schaffen. Er kam in den letzten Tagen des Sprints zurück und konnte fast alle seine Aufgaben fertigstellen, was auch dazu führt, dass das Diagramm in den letzten Tagen und bis so kurz vor Ende sinkt. Es vielen leider einige Fehler im Master an, die behoben werden mussten, welche zunächst nicht auffielen. Sie konnten jedoch alle bis auf einen behoben werden.

Am 27. Juni ist ein plötzlicher Abfall direkt gefolgt von einem erneuten Ansteigen des Graphen zu sehen. Das passierte dadurch, dass ein Teammitglied aus Versehen eine Story aus dem Sprint zog. Sie wurde direkt wieder eingefügt, aber dieses Missgeschick ist in dem Diagramm zu sehen.

## Zum Abschluss

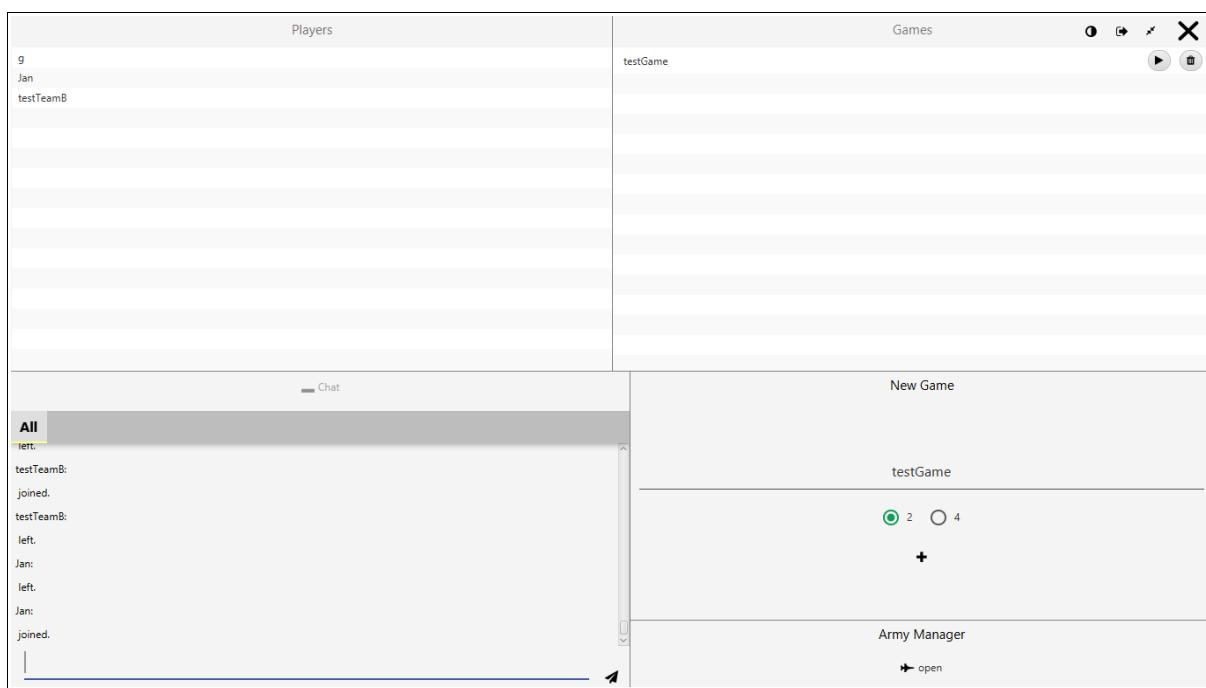
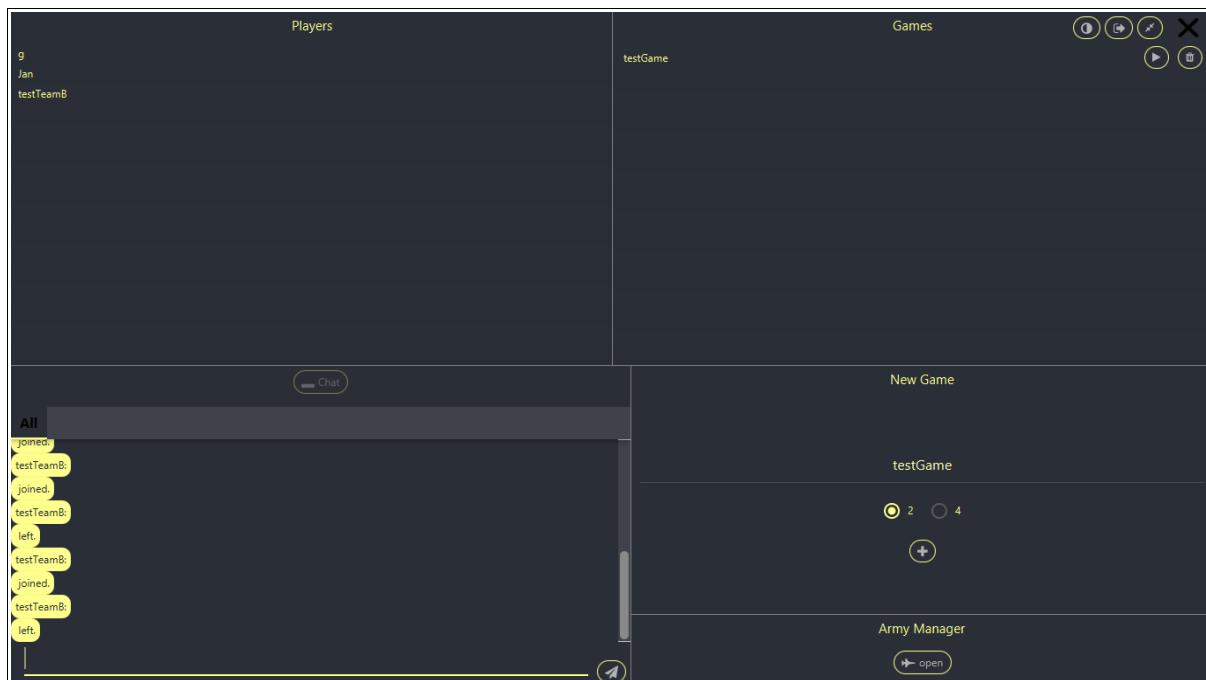
In diesem Sprint verlief nicht alles so einwandfrei und nach Plan wie im dritten. Die vorhandenen Bugs und der Urlaub eines Entwicklers wirkten sich auf den Zeitplan aus und sorgten für Verzögerungen, die letztendlich dazu führten, dass nicht alle geplanten Features realisiert wurden. Außerdem gab es einen kleinen Bug mit den Websockets, der noch nicht behoben wurde und ins nächste Release aufgenommen wird.

Insgesamt lässt sich jedoch sagen, dass der Sprint trotzdem sehr erfolgreich war, da die Mindestanforderungen hervorragend gemeistert und zahlreiche Extrafeatures implementiert wurden. Das Team arbeitete wieder ausgezeichnet zusammen und unterstützte sich gegenseitig bei Problemen, was das gute Endergebnis ermöglichte.

# Resultate

## Stand der Veröffentlichung

### Lobby



Zuerst ist das Ergebnis der Spielloby noch mal zu sehen. Das Ergebnis ist hier einmal im hellen und einmal im dunklen Modus. Man kann zwischen den Modi durch einen Klick auf den linken Knopf im Hamburger Menü wechseln. Für die Knöpfe wurden die Designs so gewählt, dass ihre Funktion intuitiv erkennbar ist. Es besteht allerdings noch die Möglichkeit in einem folgenden Release einen Schriftzug hinzuzufügen, wenn man die Maus über einen der Knöpfe verweilen lässt.

Im Vergleich zum letzten Release wurde auch die Farbe des Chats angepasst, damit er besser in die Lobby passt. Zudem wurde die Funktion hinzugefügt, dass wenn man sich in einem Chat befindet, wird man durch ein kleines Zeichen auf den Tabs des Chats benachrichtigt, falls noch eine Nachricht empfangen wird.

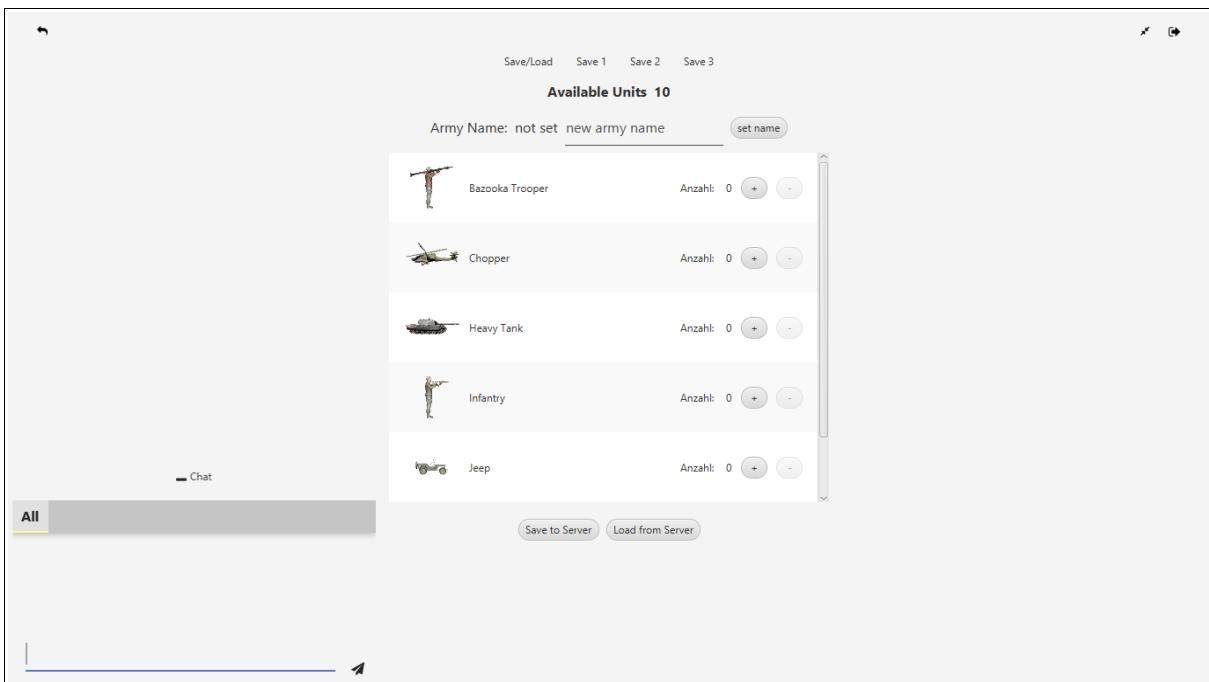
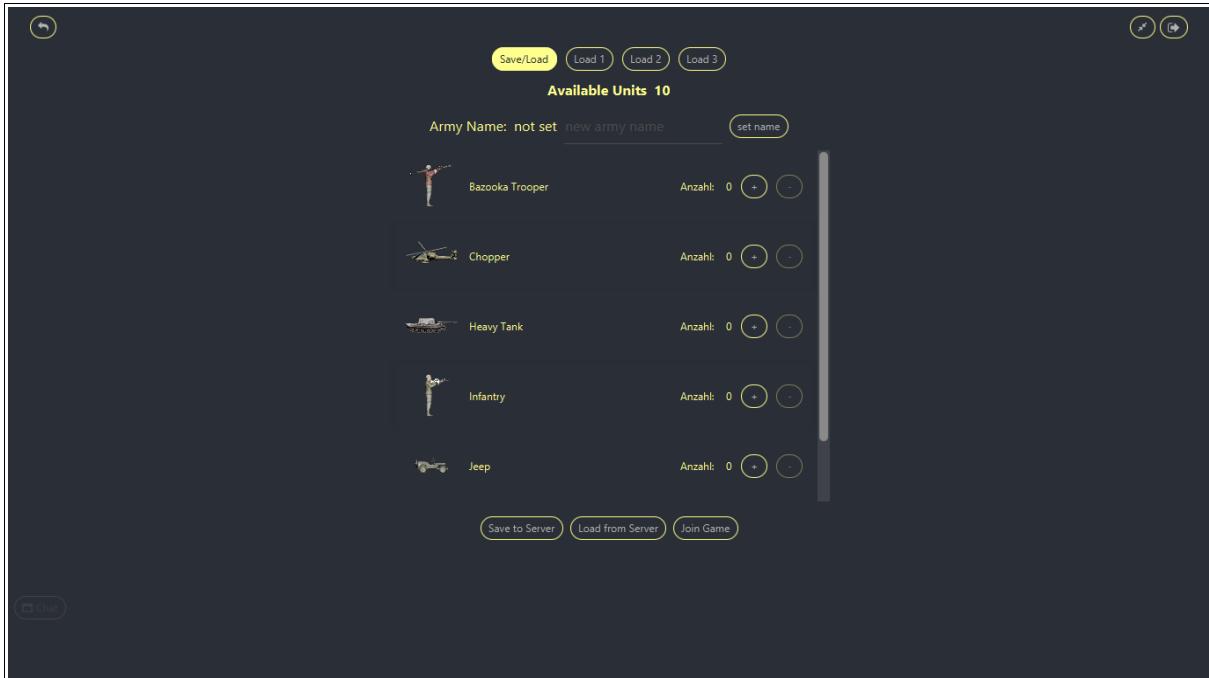
Außerdem wurden die Knöpfe oben rechts im Bildschirm durch ein Hamburger Menü zusammengefasst. Unten rechts wurde der Knopf um in den Armeemanager zu kommen hinzugefügt. Die Spielerliste wird nun automatisch aktualisiert und es sind immer die aktiven Spieler zu sehen. Versucht man einem Spiel beizutreten, gelangt man erst in den Armeemanager um sich die Armee für das Spiel auszusuchen.

Des Weiteren wurde die Möglichkeit implementiert das Spiel in Vollbild zu spielen und die Spielerfahrung erheblich zu verbessern. Auch hier wurde wieder auf ein intuitiv erkennbares Design gesetzt. Der Knopf rechts im Hamburger Menü kann genutzt werden um in diesem Fall das Vollbild zu verlassen. Würde man sich nicht im Vollbild befinden, wären die Pfeile genau anders herum.

Diese ganzen Änderungen an der Spielloby sind direkt für den Nutzer erkennbar. Es gab weitere Änderungen und Optimierungen im Code wie zum Beispiel der Anfang des Hinzufügens von Javadoc, einer Dokumentationssoftware, die automatisch Dateien aus dem Quellcode erstellt. Diese dokumentiert, was neu hinzugefügt und verändert wurde um eine bessere Übersicht über den Code für jeden Entwickler zu schaffen. Dadurch können die Entwickler effektiver arbeiten und können besser Änderungen von anderen Entwicklern nachvollziehen. Sollten neue Entwickler dem Team beitreten, könnten diese auch wesentlich besser eingeleitet werden.

Es gibt weitere kleine Performanceoptimierungen, die jedoch keinen direkten Einfluss auf den Nutzer haben und somit hier nicht weiter Ausgeführt werden.

## Armeemanager



Der Armeemanager unterscheidet sich sehr von dem eigentlichen Mockup. Die erste Idee den Armeemanager in die Spielloby einzubinden um Bildschirmwechsel zu vermeiden konnte nicht umgesetzt werden. Aus dem Grund, dass das Mockup auf der Version kurz vor Ende der ersten Veröffentlichung basiert und es einige große unangekündigte Designänderungen direkt vor Ende der Veröffentlichung gab. Zu diesem Zeitpunkt waren die Mockups schon erstellt.

Bei dem ursprünglichen Design wäre es möglich gewesen den Armeemanager direkt in die Lobby einzubinden weil in dem Fenster, wo die Spiele angezeigt wurden, genug Platz dafür vorhanden war. Nach der Änderung allerdings nicht mehr und man hätte den ganzen Code wieder umschreiben müssen um diese Idee umzusetzen, was aus zeitlichen Gründen nicht möglich gewesen wäre. Außerdem gab es mehr Einheiten, als erwartet und weitere Funktionen des Servers, die vorher nicht bekannt waren, wie den Namen einer Armee setzen, was den Platz in der Spielloobby hätte knapp werden lassen. Aus diesen Gründen wurde sich dazu entschieden den Armeemanager in einen eigenen Bildschirm zu packen. Das Grunddesign des Mockups wurde trotzdem übernommen.

Oben links im Armeemanager befindet sich ein Knopf um in die Lobby zurückzugelangen. Oben rechts hingegen sind Knöpfe für die Vollbildfunktion und zum Verlassen des Spiels vorhanden.

In der Mitte oben sind wie geplant die Knöpfe um die aktuell ausgewählte Armee zu speichern, wie auf dem Bild im hellen Modus zu erkennen. Bei Klick auf den „Save/Load“ Knopf wechselt die Speicherfunktion zur Ladenfunktion, wie es bei der dunklen Version geschah. Diese Knöpfe sind für das lokale Speichern und Laden verantwortlich. Anders als auf dem Mockup wurden noch unten in der Mitte Knöpfe für das exklusive Speichern und Laden auf dem Server hinzugefügt, da sich diese Idee als sehr praktisch erwies. Jede gespeicherten Armee muss benannt werden, was mit dem dafür vorgesehenen Eingabefeld unter den verfügbaren Einheiten gesetzt werden kann. Diese Funktion fehlte im Mockup, wurde aber hinzugefügt, da der Server diese Funktion bereitstellt und das vorher nicht bekannt war.

In dem Bild der dunklen Version gibt es rechts neben den genannten Knöpfen einen weiteren Knopf mit „Join Game“, der in der hellen Version nicht zu sehen ist. Das liegt daran, dass der Armeemanager im dunklen Modus geöffnet wurde, indem man einem Spiel beitritt und somit die Funktion zum Spielgeschehen zu gelangen vorhanden ist, während man zu dem Bild des Armeemangers im hellen Modus durch den extra „Armeemanager“ Knopf in der Spielloobby gelang. Diese Funktion hat den Grund, dass der Spieler so sehr einfach direkt vor dem Spiel eine Armee auswählen kann und nicht gezwungen ist sie vorher zu speichern aber trotzdem die Freiheit behält seine Armeen zu verändern und zu planen ohne einem Spiel beizutreten.

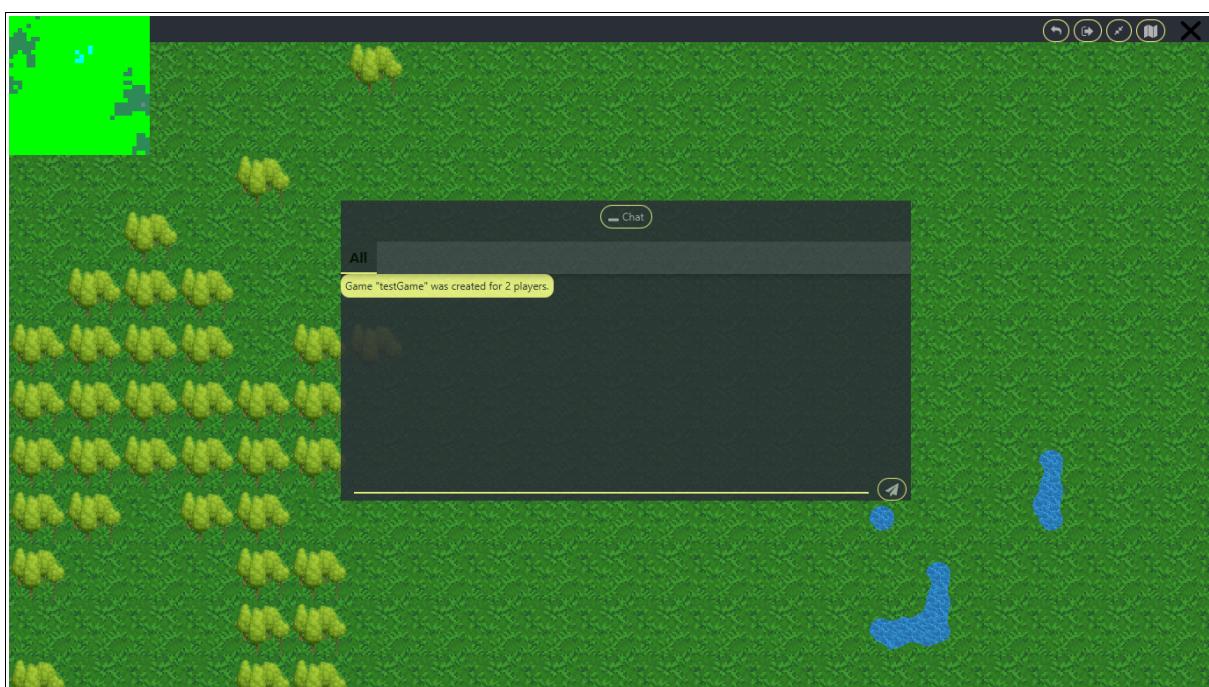
Im Armeemanager gibt es auch ein Chatfenster. Das Chatfenster ist als Knopf implementiert, sodass es nur unten links in der Ecke zu sehen ist, wenn man in den Armeemanager geht, damit es nicht stört oder ablenkt, während man an seinen Armeen arbeitet. Jedoch ist die Möglichkeit vorhanden auf den Chat zu klicken, wodurch er sich öffnet wie in dem Bild mit der hellen Modus. Der Chat ist leicht durchsichtig, während man ihn nicht fokussiert, damit er wieder in den Hintergrund gelangt, wenn man gerade nicht schreiben möchte und von der Größe frei veränderbar. Es ist sogar möglich den ganzen Bildschirm mit dem Chat zu füllen, wenn man gerade einen angeregten Chat führt.

Der Chat ist standardmäßig im globalen Modus, es ist aber auch möglich private Chats zu starten, indem man „/w USER“ schreiben würde, wobei USER den Benutzernamen darstellen soll. Eine weitere kleine Funktion ist, dass ein Meme auftaucht, wenn man „//“ in den Chat eingibt.

Im Zentrum des Bildschirms befinden sich dann die Einheiten mit Knöpfen um die Anzahl zu verändern. Die Textfelder verändern sich dynamisch, je nachdem was man dort auswählt. Knöpfe, die nichts mehr verändern können, sollte man sich klicken, werden grau hinterlegt um den Nutzer nicht zu verwirren und weiter das intuitive Design zu verbessern.

Es gibt sechs verschiedene auswählbare Einheiten für die alle eigene Texturen von einem Entwickler erstellt wurden. Es gibt den „Bazooka Trooper“, den „Chopper“, den „Heavy Tank“, die „Infantrie“, den „Jeep“ und den „Light Tank“. Für den „Chopper“ gibt es die Animation, dass sich die Rotorblätter bewegen.

## Spielgeschehen





Das Spielgeschehen ist von der Idee des Mockups gleich geblieben. Am Design hat sich hier jedoch auch einiges verbessert. Wie beim Mockup beschrieben wurde das Spielfeld in Kästchen aufgeteilt. Die Abtrennung zwischen den einzelnen Kästchen wird nun aber nicht durch ein schwarzes Gitter realisiert sondern die Texturen gehen flüssig ineinander über um das ganze Spielfeld nicht so starr und unecht aussehen zu lassen. Diese Änderung kam durch die Erkenntnis, dass ein Gitter nicht gut in das Spielfeld passt und die Möglichkeit bestand es mit den zu sehenden flüssigen Übergängen zu verwirklichen.

Die Einheiten werden über das Spielfeld gelegt, man kann also bei jeder Einheit das Spielfeld darunter noch sehen. Darauf wurde geachtet, damit das flüssige Design dadurch nicht verloren geht. Auf diesen Bildern sind die Einheiten nicht mehr zu sehen, da zum Zeitpunkt der Aufnahmen der Server verändert wurde und das Programm erst auf diese Veränderungen angepasst werden muss, damit die Einheiten wieder geladen werden.

Der Chat ist ähnlich wie im Armeemanager. Man hat die Möglichkeit ihn nur in einem kleinen Modus links unten am Bildschirmrand zu lassen oder ihn aufzuklappen. Im aufgeklappten Modus kann man die Größe trotzdem beliebig verändern und sogar das ganze Chatfenster an jede beliebige Stelle ziehen, wie es auf dem zweiten Bild zu sehen ist. Vergleicht man das zweite und das dritte Bild kann man auch den Unterschied erkennen, wenn man den Chat fokussiert. Der Chat ist dann wesentlich weniger durchsichtig, damit man die Nachrichten besser lesen kann. Auch hier wird darauf geachtet, dass das Spielgeschehen flüssig aussieht und die Idee des Designs nicht negativ beeinflusst.

Oben rechts in dem Hamburger-Menü befinden sich von links nach rechts die Knöpfe um zurück in die Lobby zu gelangen, dann um das Spiel zu verlassen, für die Vollbildfunktion und zum Schluss um die Minimap. Die Minimap ist eine Funktion, die schon im Mockup des Armeemanagers vorhanden war. Da es wegen der Serverfunktionen nicht möglich war die Minimap schon im Armeemanager anzuzeigen, wurde entschieden, dies im Spielgeschehen zu implementieren. Die Minimap ist besonders sinnvoll, da das Spielfeld beweglich ist aber man

keinen kompletten Überblick bekommt.

Die aktuelle Runde, verschiedene Währungen oder ein Icon am linken oberen Rand wie im Mockup wurden noch nicht implementiert, da es zu wenig Informationen darüber gab und es sinnvoller erschien, es direkt richtig zu implementieren, wenn alle Einzelheiten bekannt sind um doppelte Arbeit zu vermeiden und Effizienz zu steigern.

## **Schlusswort**

Abschließend lässt sich sagen, dass das Release sehr erfolgreich verlief. Die gestellten Anforderungen wurden ausgezeichnet umgesetzt und viele kleine Extras und Funktionen haben ihren Weg in die Veröffentlichung gefunden, was zu einem sehr guten Ergebnis führt.

Es gab einige Abweichungen von den Mockups und den Ideen bevor die Serverdokumentation veröffentlicht wurden, sowie durch die Änderungen in den letzten Minuten vor dem ersten Release. Im Endeffekt hatten diese Änderungen keinen negativen Einfluss auf das Design, da existierende Ideen erweitert und erneut Überarbeitet wurden, wodurch das Ergebnis sich weiter verbesserte und die Designideen einen klareren Plan annahmen.

Das Spiel ist noch nicht perfekt aber alle Entwickler arbeiten hart daran es mit jedem Release besser wird um am Ende ein Produkt zu präsentieren, dass so gut wie möglich mit den verfügbaren Ressourcen und Zeit ist.