

PROJEKTDOKUMENTATION

Release III

UNIVERSITÄT KASSEL
SOFTWARE ENGINEERING I SS 19

**Projekt: Rundenbasiertes Strategiespiel
RBSG - Enhanced Wars**

GRUPPE G



EnGineerinG
<Problem/>

SCRUM MASTER: JAN MÜLLER
PRODUCT OWNER: KEANU STÜCKRAD

8.7.2019 - 4.8.2019

1 Vorwort

Das Release III von Gruppe G des Projekts RBSG, ein rundenbasiertes Strategiespiel, soll im Rahmen des Moduls „Software Engineering I“ im Fachbereich „Software Engineering“ im Sommersemester 2019 mit dieser Dokumentation festgehalten werden.

Das Vorgehen in diesem Modul ist so gegliedert, dass es vier Releases mit jeweils zwei Sprints über vier Monate gibt. Das Projekt begann am 13.5.2019 und wird bis zum 1.9.2019 laufen. Ein Client für einen Klon vom Klassiker „Advanced Wars“ soll entwickelt werden. Dabei ist das Team über die vier Releases immer in einen Scrum Master, einen Product Owner und mehrere Developer eingeteilt.

Die Dokumentation wird eine Einleitung in das Release, danach die Zusammenfassung der beiden Sprints und eine abschließende Releaseanalyse enthalten. In der Einleitung sollen die Anforderungen des Kunden, der letzte Standpunkt und die neuen MockUps und User Stories erläutert werden. In den Sprintdokumentationen wird kurz das jeweilige Ziel dargelegt und darauf sollen alle User Stories aufgelistet werden. Abschließend wird es jeweils eine Sprintanalyse mit einer Zeit- und Vorgangsübersicht geben. Die Releaseanalyse wird aufzeigen, ob das Resultat den Anforderungen entspricht und ob das Team effektiv gearbeitet hat.

Inhaltsverzeichnis

1 Vorwort	1
2 Release III	4
2.1 Releaseanforderungen	4
2.2 Standpunkt des letzten Releases	5
2.2.1 Login	5
2.2.2 Lobby	7
2.2.3 Army Manager	8
2.2.4 Waiting Room (Gamelobby)	10
2.2.5 Ingame	12
2.2.6 Weitere Features	14
2.3 MockUps	14
2.3.1 Lobby	15
2.3.2 Waiting Room (Gamelobby)	15
2.3.3 Ingame	17
2.3.3.1 Einheit auswählen	18
2.3.3.2 Einheit bewegen	20
2.3.3.3 Einheit angreifen	20
2.3.3.4 Spielende	21
2.3.3.5 Kontextmenu	23
2.3.3.6 Weitere Features	23
2.4 Domain Stories	25
3 Sprint V	25
3.1 Sprintziel	25
3.2 User Stories	25
3.2.1 Waiting Room - Spielbeitritt anpassen	25
3.2.2 Ingame - Spielfeld	26
3.2.3 Ingame - Phase beenden	26
3.2.4 Ingame - Einheit auswählen	26
3.2.5 Ingame - Einheit bewegen	27
3.2.6 Ingame - Einheit angreifen	27
3.2.7 Ingame - Spielstatus anzeigen	27
3.2.8 Ingame - Minimap anzeigen	27
3.2.9 Ingame - Chatintegration	28
3.2.10 Ingame - Game Over	28
3.2.11 Ingame - Game Won	28
3.2.12 Lobby - Spactating	28
3.2.13 Ingame - Spactating	29
3.2.14 Ingame - Spectator Over	29
3.2.15 Ingame - Einheiteninformationen	29
3.2.16 Ingame - Musiksteuerung	29
3.3 Tasks	30
3.3.1 Servernachrichten	30
3.3.2 Einheiteninformationen im Datenmodell	30

3.3.3 Ingame - Kontextmenü-Framework	30
3.4 Zeitübersicht	31

2 Release III

Das Release III erstreckt sich über die vier Wochen vom 8.7.2019 bis zum 4.8.2019. Die zwei Sprints gehen vom 8.7.2019 bis zum 21.7.2019 und vom 22.7.2019 bis zum 4.8.2019.

Das Team ist in diesem Release wie gefolgt aufgeteilt:

- Scrum Master:  Jan Müller
- Product Owner:  Keanu Stückrad
- Developer:
 1.  Georg Siebert
 2.  Juri Lozowoj
 3.  Omar Sood
 4.  Tobias Klipp

2.1 Releaseanforderungen

Vom Kunden wurden folgende Features gefordert:

1. Client: Waiting Room (Gamelobby)
 - Unterstützung eines Chats
 - Armee soll auch nach dem Spielbeitritt ausgewählt werden können
 - Spieler müssen ein Bereit senden können
 - Spieler müssen das Spiel starten können
2. Client: Ingame
 - Bewegen von Einheiten
 - Angreifen von Einheiten
 - Unterstützung einer Minimap
 - Unterstützung eines Chats
 - Anzeige aller Spieler
 - Anzeige der aktuellen Runde/Phase
 - Spieler müssen das Spiel verlassen können
 - Anzeige des Gewinners
3. Client: Lobby
 - Spieler muss einem Spiel über Beobachtermodus beitreten können
4. Qualitätssicherung
 - C0 Testabdeckung von 75%

2.2 Standpunkt des letzten Releases

Das Release II erstreckte sich vom 10.6.2019 bis zum 7.7.2019. Nach dem 7.7. waren folgende Teile des Clients fertig gestellt:

- Login
 - * Anmelden
 - * Registrieren
- Lobby
 - * Spiel erstellen/beitreten
 - * Anzeige der angemeldeten Spieler
 - * Anzeige der aktiven Spiele
 - * Unterstützung eines Chats
 - * Ausloggen
- Army Manager
 - * Konfigurieren eigener Armeen
 - * Speichern von Armeen lokal und auf dem Server
- Waiting Room (Gamelobby)
 - * Unterstützung eines Chats
 - * Anzeige der beigetretenen Spieler
 - * Zurück zur Lobby
- Ingame
 - * Anzeige des initialen Spielgeschehens
 - * Zurück zur Lobby

Weiterhin wurde bereits eine C0 Testabdeckung von mindestens 60% erreicht.

2.2.1 Login

Im Login kann der User sich anmelden oder sich sogar registrieren (siehe Abbildung 1).

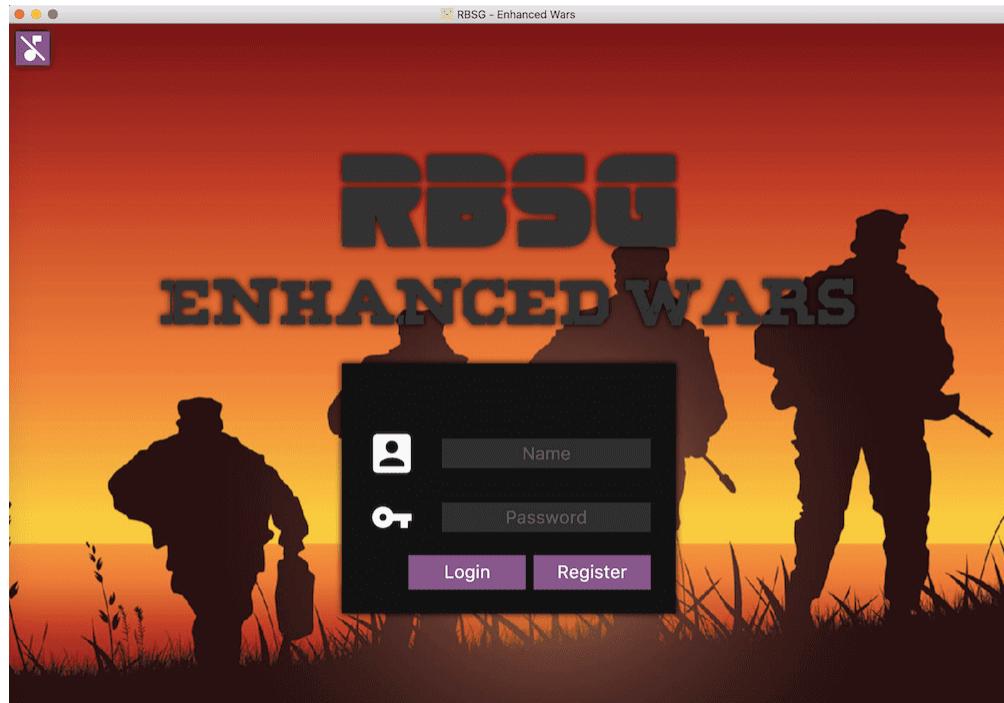


Abbildung 1: Login Szene

Versucht man sich anzumelden oder zu registrieren, erscheint ein Lade Indicator (siehe Abbildung 2) und die Buttons und Textfelder werden disabled. Macht der User dabei einen Fehler oder kann keine Verbindung zum Server hergestellt werden, erscheint eine Fehlermeldung (siehe Abbildung 2). Der Login ist internationalisiert und der Untertitel „Enhanced Wars“ ist animiert. Weiterhin kann man die Musik über den Button ein- und ausschalten.

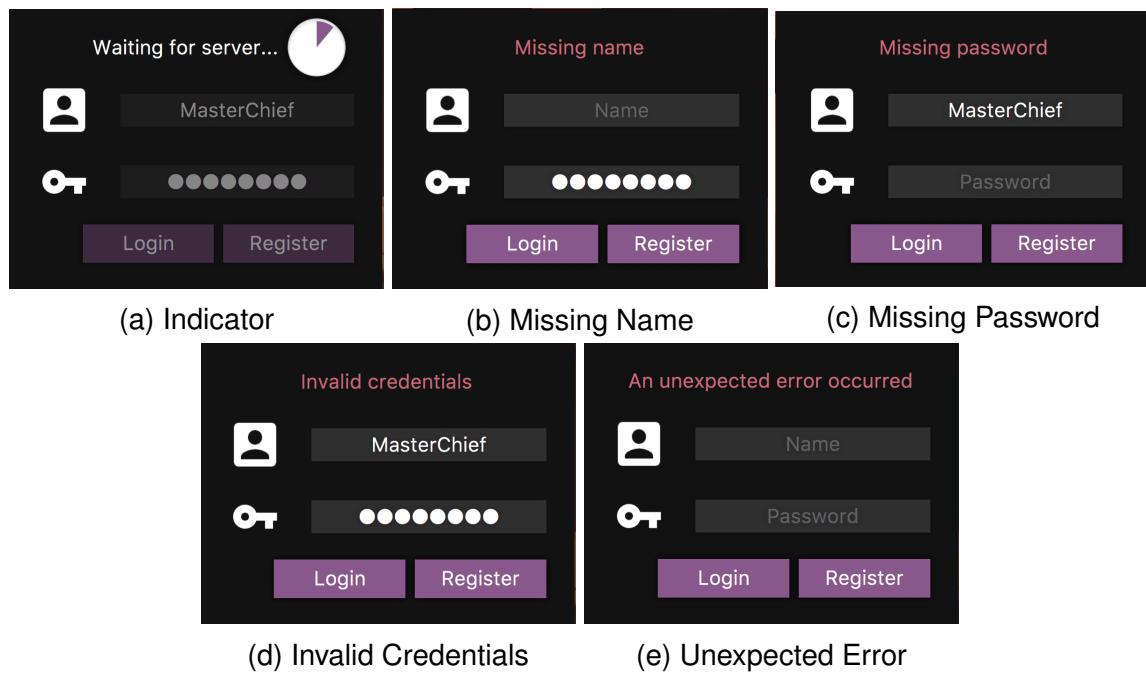


Abbildung 2: Login Form

2.2.2 Lobby

Die Lobby zeigt eine Liste von allen Spielern (die nicht im Spiel sind) und allen Spielen an. Diese aktualisieren sich, wenn der Server die entsprechende Nachricht schickt. Man kann sich über die Buttons oben rechts ausloggen, die Sprache ändern oder die Musik an- und ausschalten. Der Chat wird unten links angezeigt. Rechts sieht man eine Liste mit allen vollständigen Armeen. Hat man keine vollständige Armee ausgewählt, kann man kein Spiel erstellen (siehe Abbildung 3).

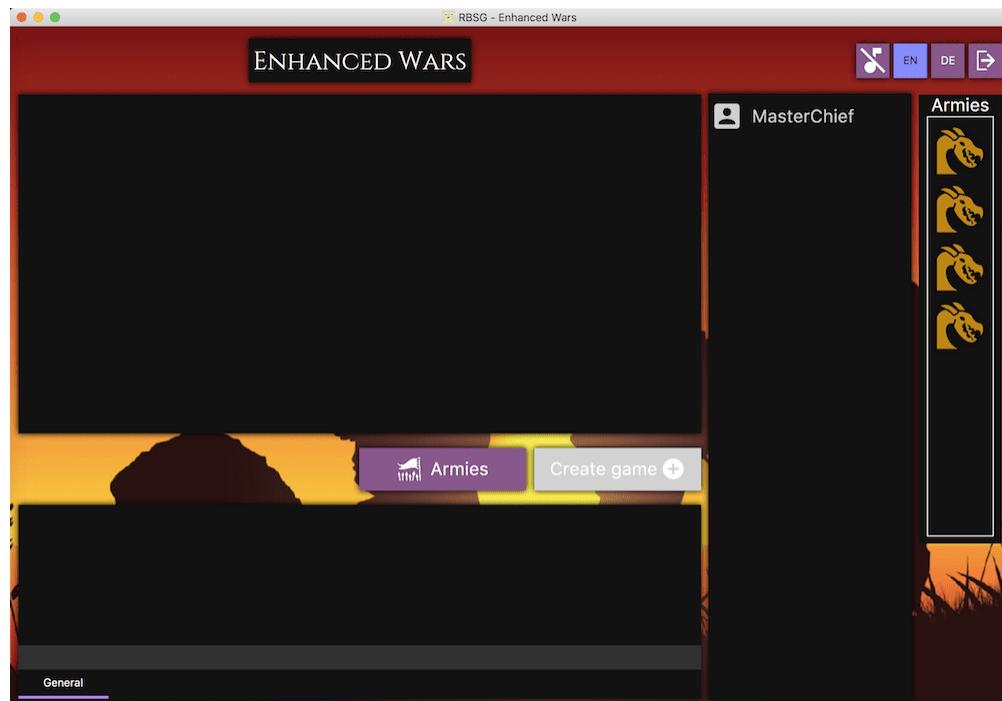


Abbildung 3: Lobby Szene: Keine Armee ausgewählt

Hat man eine Armee ausgewählt, kann man ein Spiel erstellen oder einem Spiel beitreten (siehe Abbildung 4 oder 6). Spiel erstellen geht über Create Game (siehe Abbildung 5). Der User hat die Auswahl zwischen einem 2-Spieler-Spiel und einem 4-Spieler-Spiel. Danach wird ein Autojoin getriggert. Die Lobby ist auch internationalisiert (vgl. Abbildung 5 mit 6).

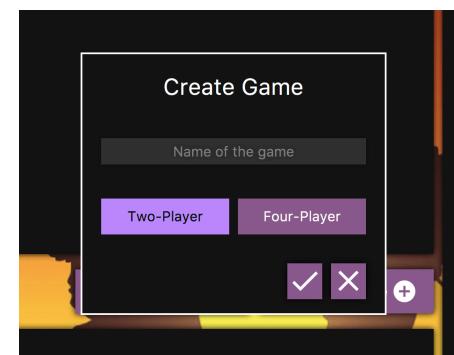


Abbildung 4: Create Game

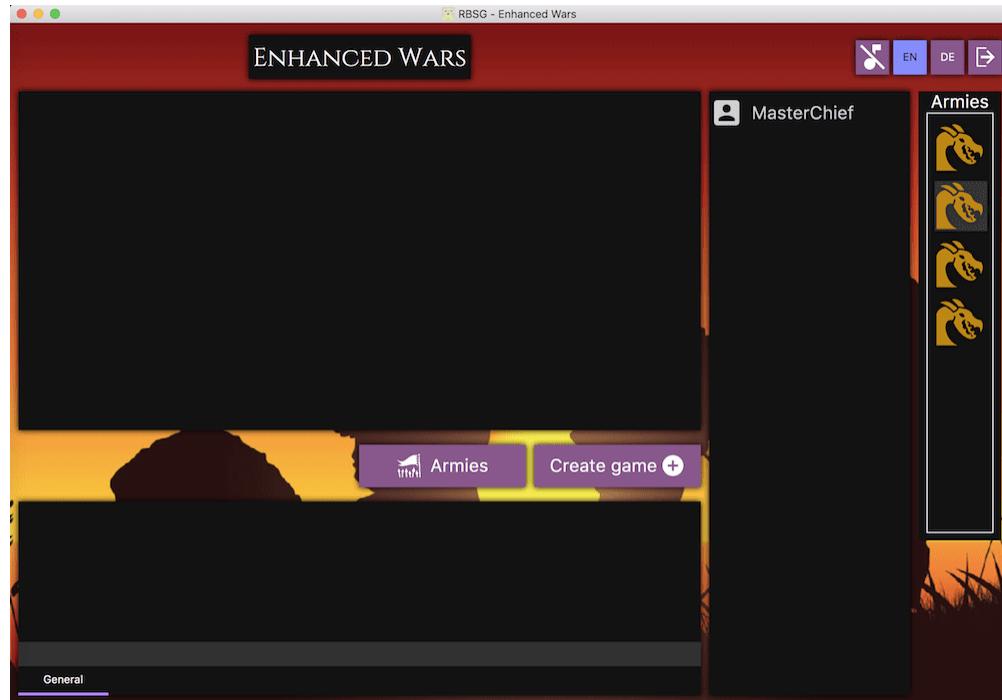


Abbildung 5: Lobby Szene: Armee ausgewählt

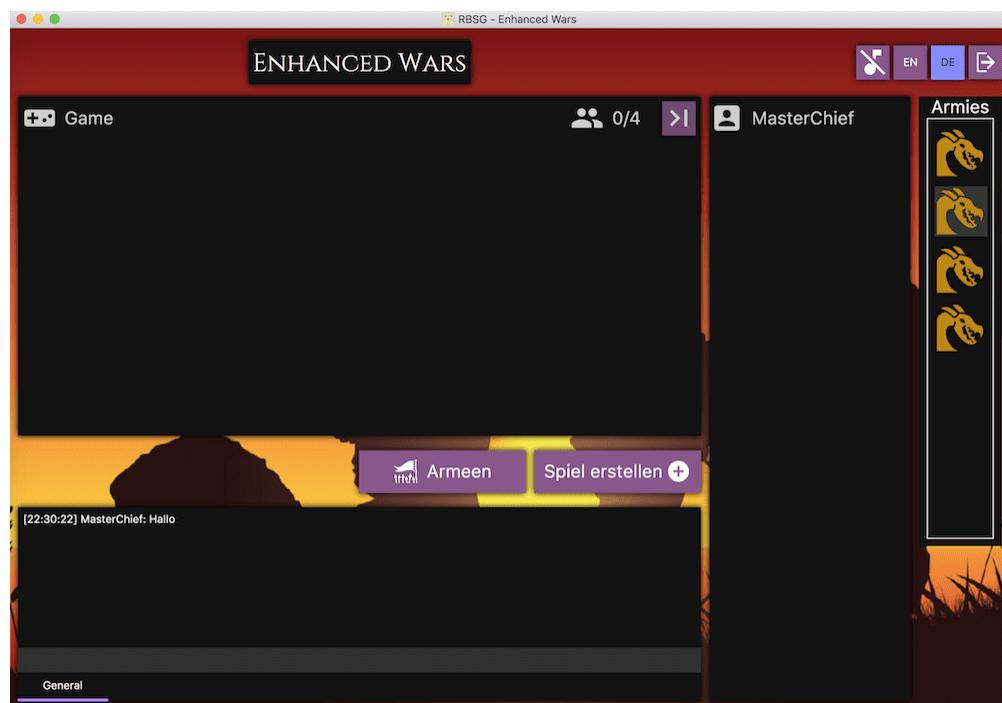


Abbildung 6: Lobby Szene: Sprache DE

2.2.3 Army Manager

Im Army Manager sieht der User eine Liste der Einheiten, die man wählen kann, und eine Liste der Einheiten, die in der gerade rechts ausgewählten Armee sind. In der Mitte steht der Name der ausgewählten Armee, ein + und ein - Button, um Einheiten der Armee hinzuzufügen oder um sie zu löschen und ein Zähler, der anzeigt, wie voll die Armee ist.

Neben der Einheitenliste ist oben eine Vorschau der gerade ausgewählten Einheit. Neben diesem Feld werden die Properties der Einheit angezeigt und welche Einheiten die ausgewählte Einheit angreifen kann. Hier stehen bei keiner Auswahl Fragezeichen (siehe Abbildung 7), ansonsten

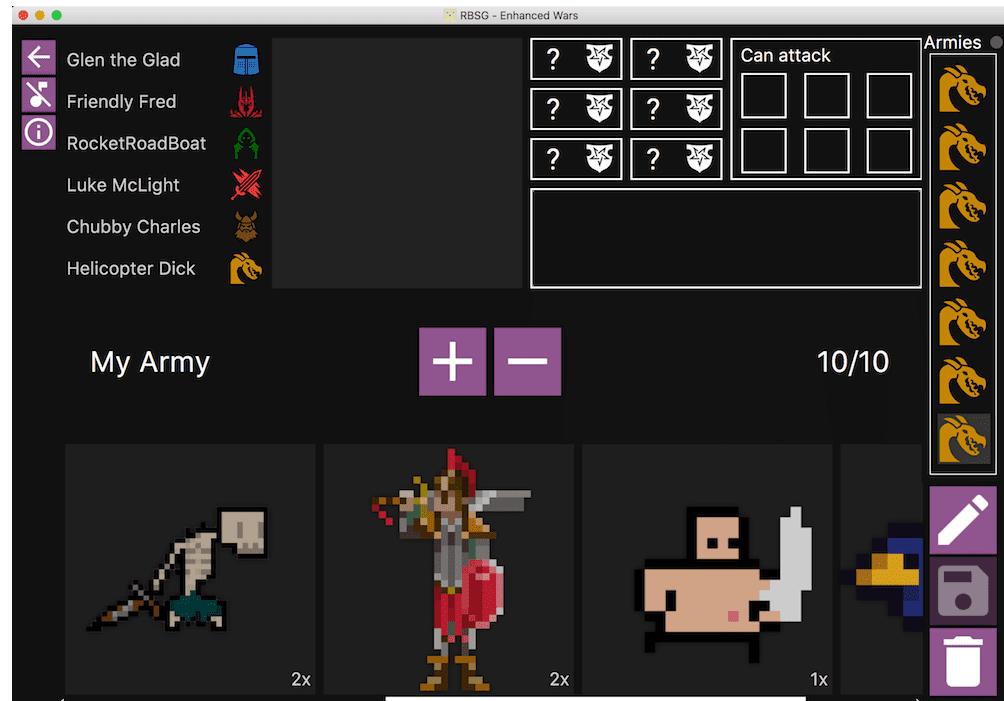


Abbildung 7: Army Manager Szene: Keine Einheit ausgewählt

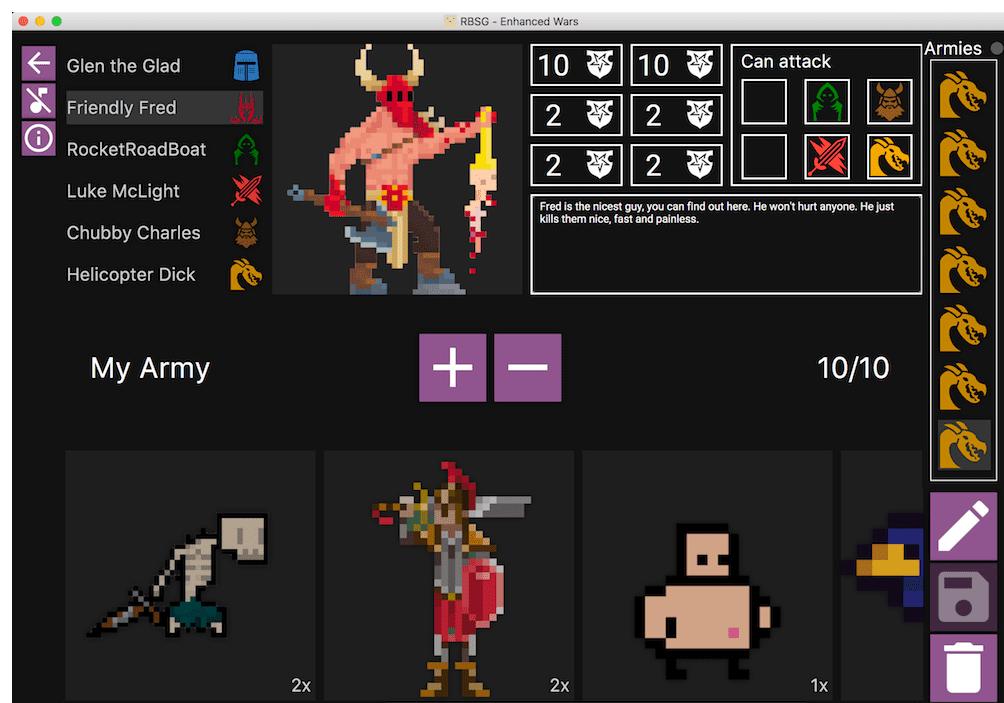


Abbildung 8: Army Manager Szene: Einheit ausgewählt

Werte (siehe Abbildung 8). Darunter steht der Lore Text der ausgewählten Einheit. Oben rechts sind Buttons, um zurück zur Lobby zu gehen, um die Musik ein- oder auszuschalten und ein Button der die Properties in einem neuen Menu erklärt (siehe Abbildung 37). Unten rechts sind Buttons, um eine Armee komplett

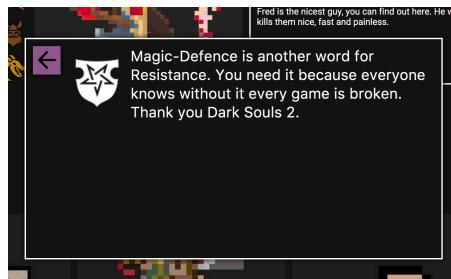


Abbildung 9: Property Info

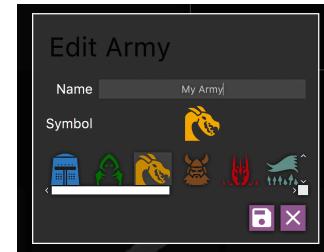


Abbildung 10: Armee bearbeiten

zu löschen, um sie zu speichern (wird disabled, wenn sich nichts geändert hat (siehe Abbildung 7), wenn der User etwas ändert wieder enabled (siehe Abbildung 11)) und ein Button, um den Armeenamen zu ändern oder das Icon zu wechseln. Dazu öffnet sich ein weiteres Menu (siehe Abbildung 10).

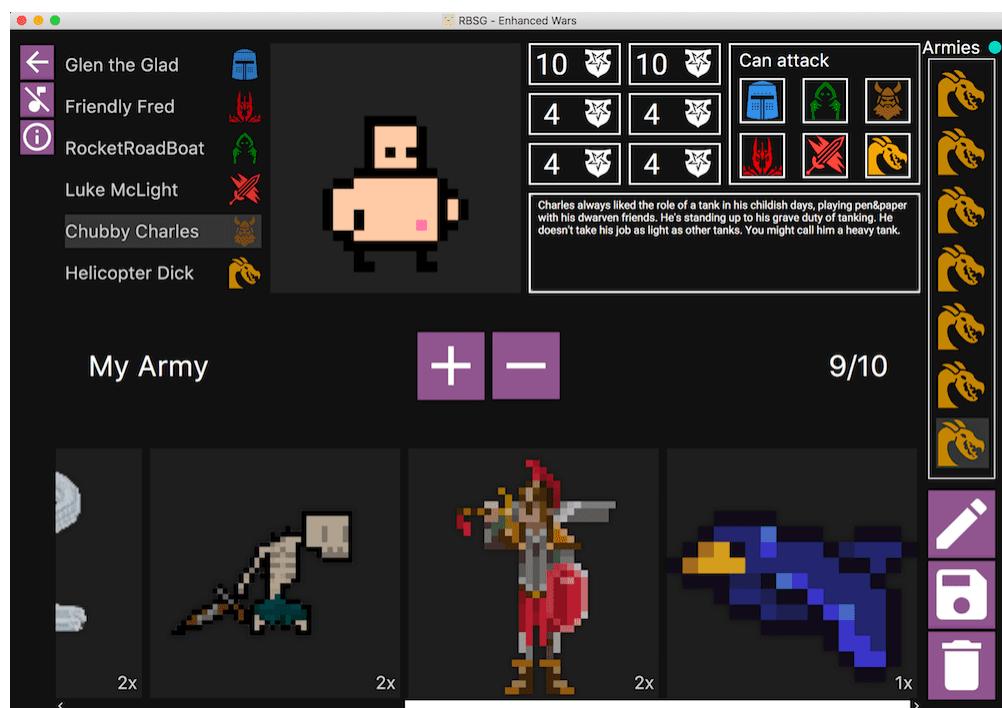


Abbildung 11: Army Manager Szene: Speicherbar

2.2.4 Waiting Room (Gamelobby)

Tritt der User dem Warteraum bei, dann sieht er oben rechts Buttons. Einmal einen Button zum Verlassen des Spiels in die Lobby, ein Button, um die Musik ein- und auszuschalten, und ein Button, um Informationen anzuzeigen. Da dieser Button noch keine spezielle

Funktion hat, wechselt man mit jenem Testweise auf das Spielfeld (Ingame). Oben in der Mitte wird eine Kartenvorschau angezeigt. Links und rechts daneben sind die Playercards. Je nachdem ob man ein 2- Spieler-Spiel oder ein 4-Spieler-Spiel gestartet hat, sind dort zwei oder vier Spielerkarten (siehe Abbildung 12 und 13).

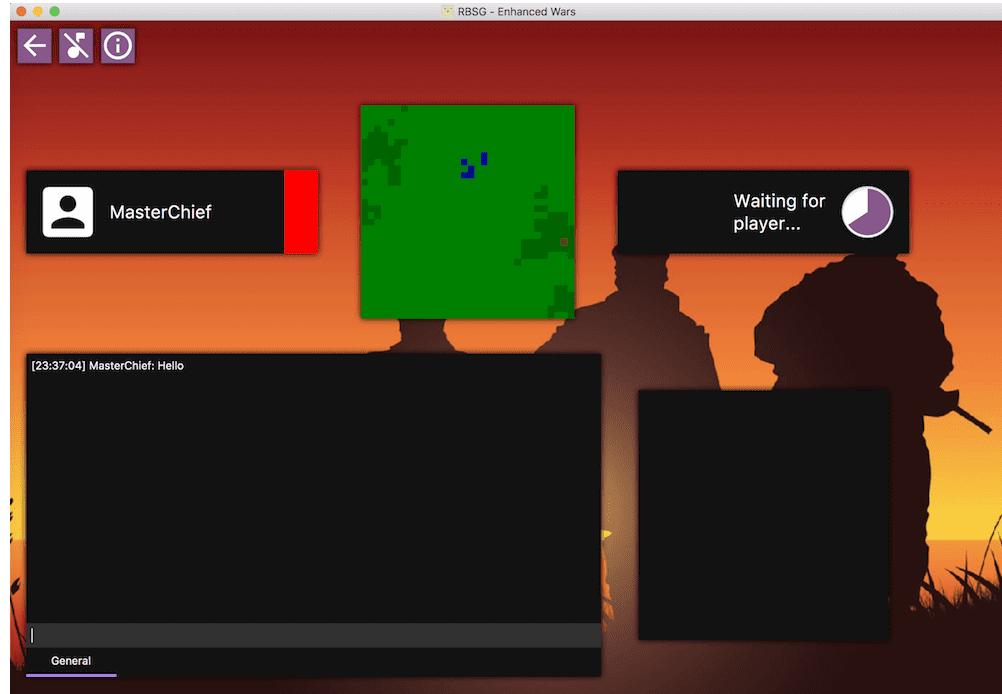


Abbildung 12: Waiting Room Szene: 2 Spieler

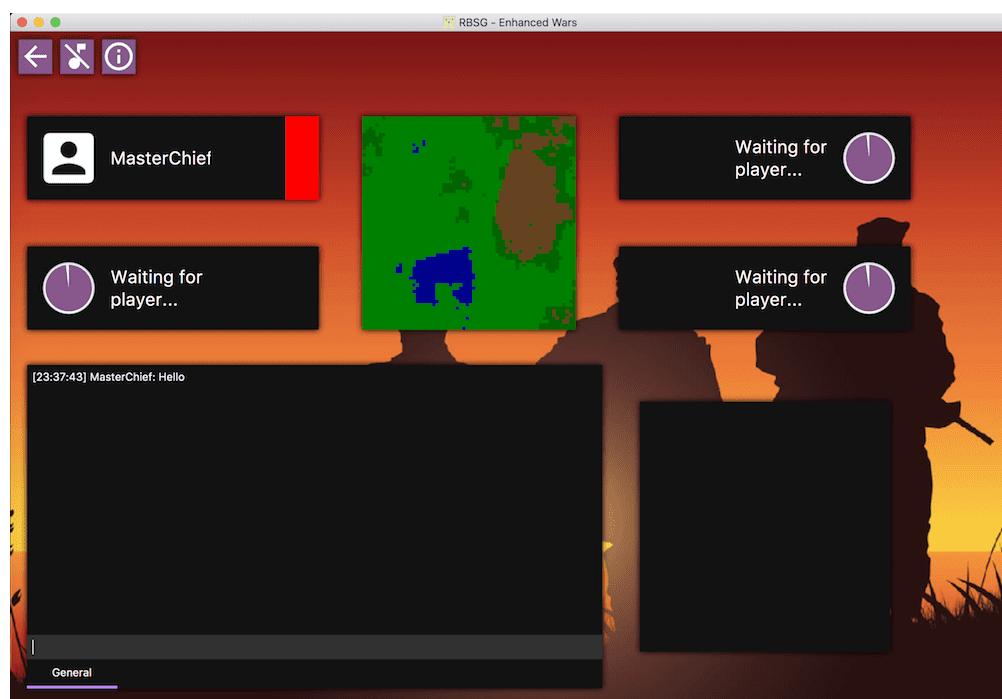


Abbildung 13: Waiting Room Szene: 4 Spieler

Auf den Spielerkarten stehen der Name des jeweiligen Spielers und der User sieht die Farbe. Ist noch kein Spieler da, wird ein Indikator angezeigt. Unten ist der Chat und daneben eine freie Pane für ein Minigame, dass noch implementiert werden soll.

2.2.5 Ingame

Die Ingame Szene zeigt bisher nur die Karte und das initiale Spielgeschehen. Die initialen Einheiten werden auch angezeigt. Weiterhin gibt es einen Button, der zurück in die Lobby geht. Zwei weitere Buttons kontrollieren das Zoomen (siehe Abbildung 14, 15 und 16).

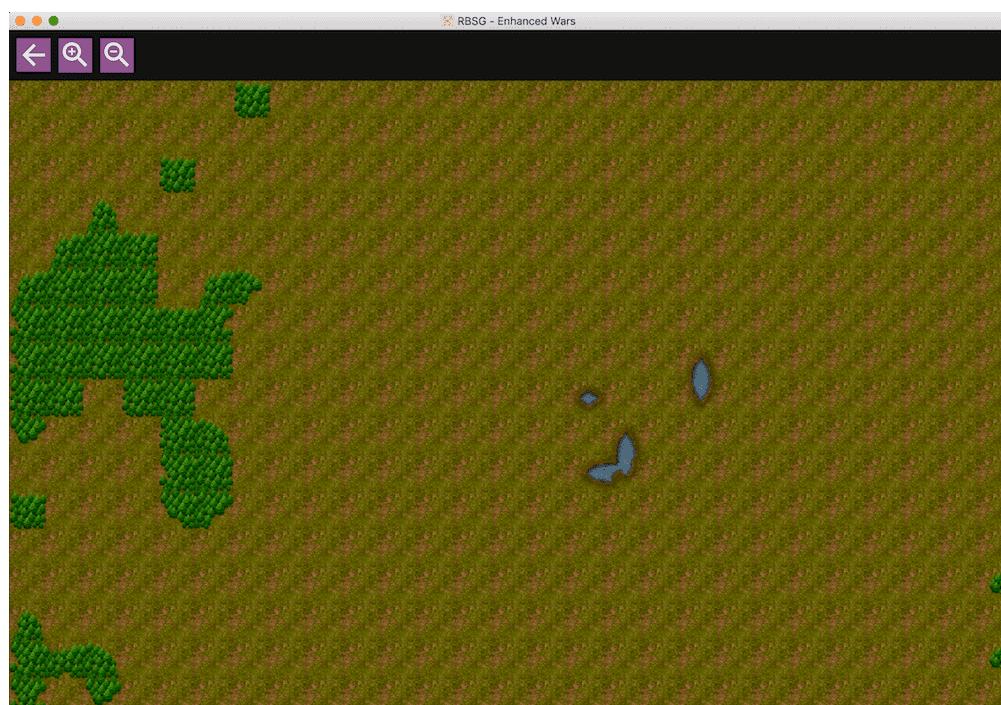


Abbildung 14: Ingame Szene

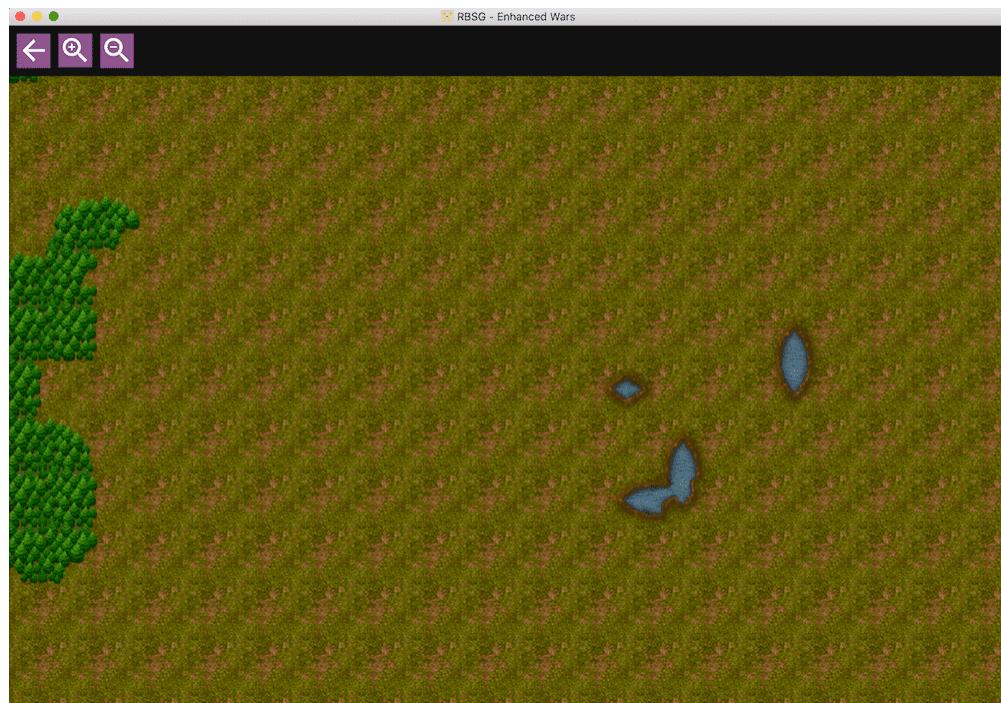


Abbildung 15: Ingame Szene: Zoom In

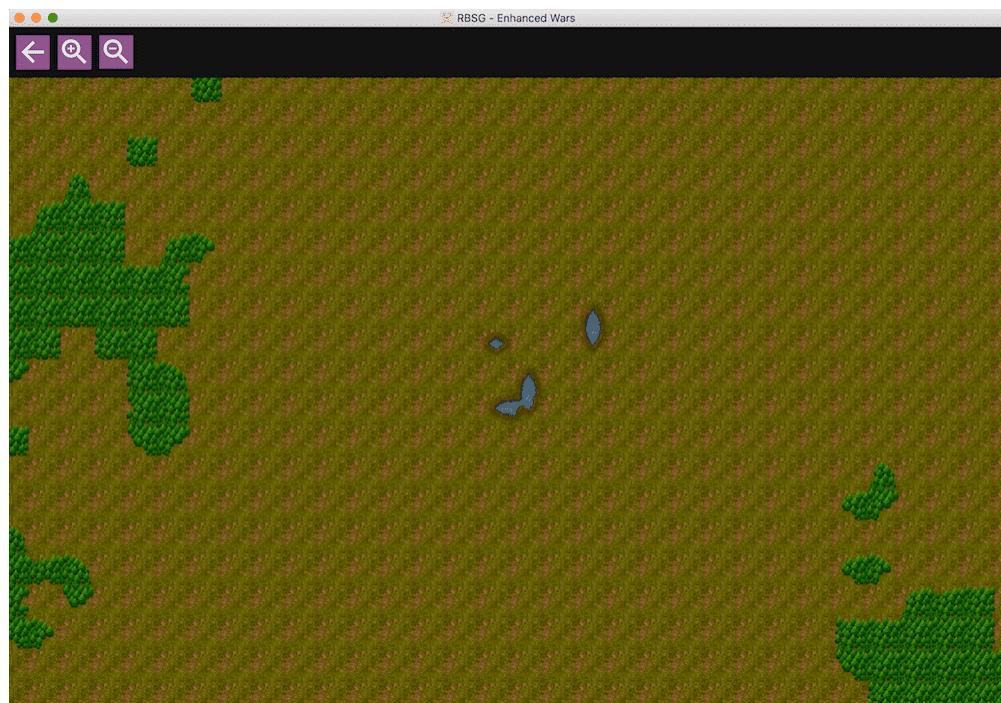


Abbildung 16: Ingame Szene: Zoom Out

2.2.6 Weitere Features

Weiterhin wurde seit den ersten beiden Releases ein Dark Theme Styling implementiert, wie man auf allen vorangegangenen Abbildungen sehen kann. Musik und Internationalisierung war auch völlig optional und funktioniert nun in jeder Szene. Der User kann auch nach Belieben Chuck Norris Zitate über den Chat-Befehl „\chuckMe“ im Chat versenden und weiterhin wird nun angezeigt, wann jede Nachricht geschickt wurde. Wenn man eine private Nachricht bekommt, wird dies auch auf dem jeweiligen Tab mit der secondary Farbe signalisiert. Die Kartenvorschau im Warteraum war auch optional, genauso wie das Zoomen Ingame. Den Armeenamen, sowie das Icon dazu ändern zu können, war auch optional. Der Autojoin nach dem Create Game war auch nicht verlangt, funktioniert nun aber bestens. Ein weiteres sehr wichtiges Feature ist, dass der Client den Spieler nun Fehlermeldungen (siehe Abbildung 17 und 18) schickt, wenn der Server einen WebSocket unerwartet schließt. Man wird je nach Fehler in den Login oder die Lobby zurückgeschickt. Außerdem gibt es ein Fenster, in dem der User noch einmal aussuchen kann, ob er sich wirklich ausloggen möchte oder ob er wirklich das Spiel verlassen möchte (siehe Abbildung 23 und 24).

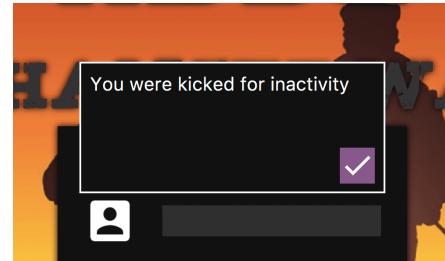


Abbildung 17: Login Fehler

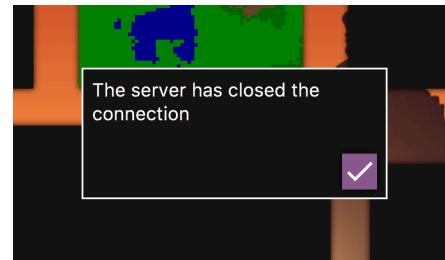


Abbildung 18: Spielfehler

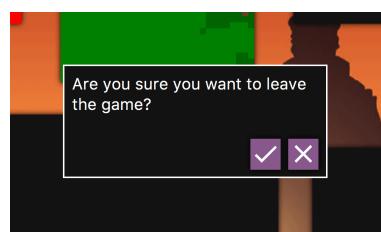


Abbildung 19: Spiel verlassen

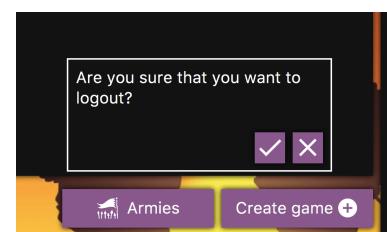


Abbildung 20: Logout

Die C0 Testabdeckung betrug zum Ende des Releases II 77%.

97% classes, 77% lines covered in package 'de.uniks.se19.team_g'			
Element	Class, %	Method, %	Line, %
project_rbsg	97% (380/389)	80% (1586/1967)	77% (7915/10223)

Abbildung 21: C0 Testabdeckung: Siehe Line, %

2.3 MockUps

Aufgrund der Releaseanforderungen wurden für das Kundentreffen am 1.7.2019 MockUps zu der neuen Ingame Szene erstellt. Weiterhin gab es auch noch MockUps zu Lobby und Waiting Room. Diese wurden nach dem Erscheinen der Serverdokumentation noch er-

weitert. Die MockUps sind im Stil des Dark Theme und sollen an die UI aus den vorherigen Releases anknüpfen.

2.3.1 Lobby

In der Lobby muss zum Stand aus dem Release II (siehe Kapitel 2.2.2) ein Button hinzugefügt werden, der den User einem Beobachtungsmodus beitreten lässt, der ihn direkt in die Ingame Szene weiterleitet. Des Weiteren soll die Armeeauswahl von der Lobby entfernt werden (und in den Waiting Room verlegt werden).

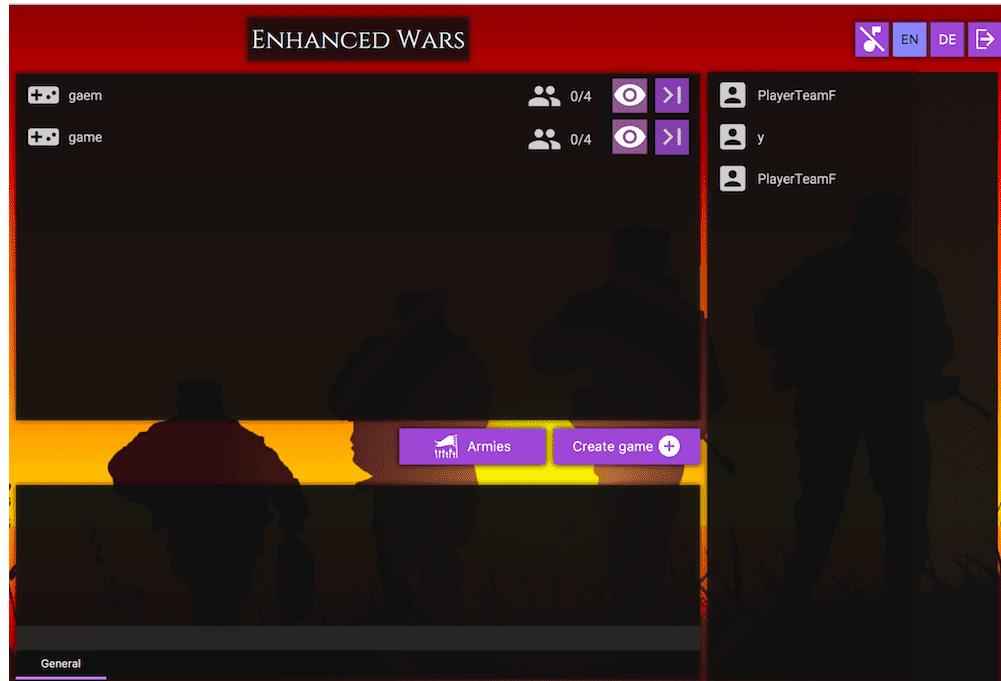


Abbildung 22: Beobachtungsmodus Lobby

2.3.2 Waiting Room (Gamelobby)

Da der Waiting Room im Release II (siehe Kapitel 2.2.4) schon größtenteils vorhanden war, muss nun nur noch das Ready geben hinzugefügt werden. Dies soll bei uns passieren, wenn man eine Armee auswählt (siehe Abbildung 24). Die Armeeauswahl muss deswegen von der Lobby in den Waiting Room umgezogen werden. Spieler, die bereits bereit sind, sollen in ihrer Playercard in der secondary Farbe hinterlegt werden und die Icons sollen sich schwarz färben. Wenn alle Spieler das Ready gegeben haben, wird ein Overlay angezeigt, in dem ein Indicator lädt, bis zur Ingame Szene gewechselt wird (siehe Abbildung 25). Auf den MockUps ist auch ein optionales Feature zu sehen. Dies ist das TicTacToe Spiel unten rechts, das gegebenenfalls implementiert werden soll.

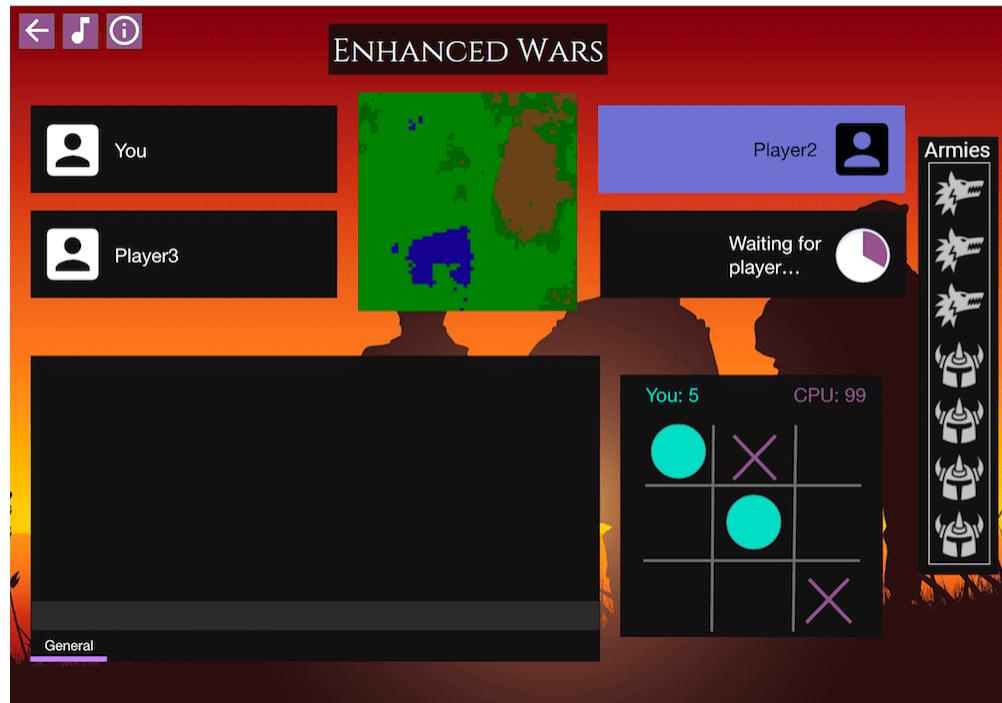


Abbildung 23: User nicht bereit

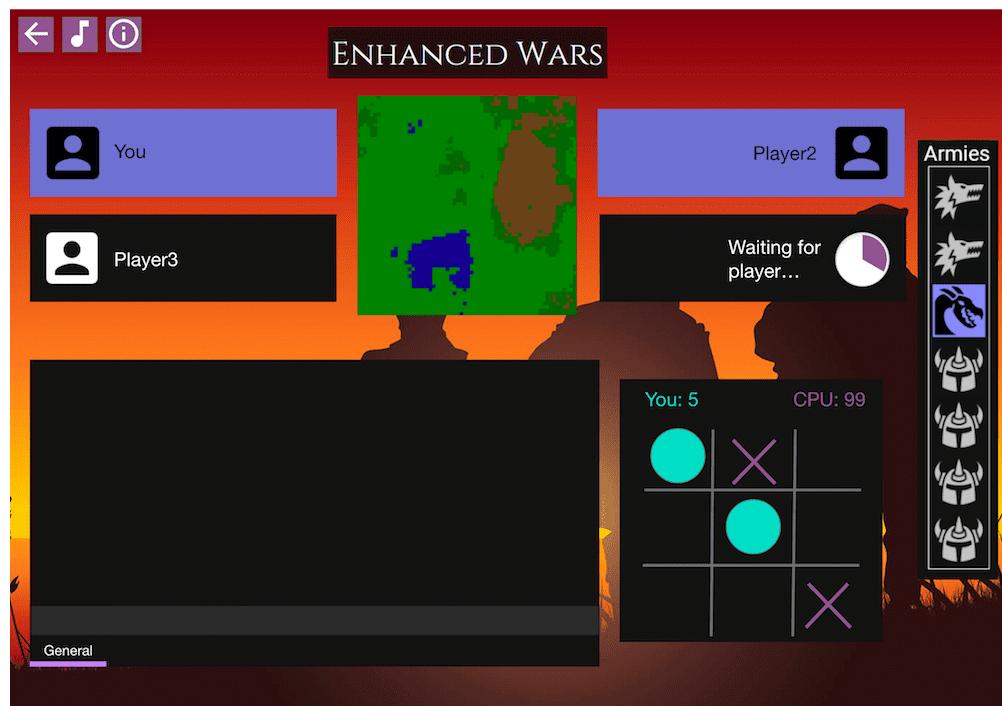


Abbildung 24: User bereit

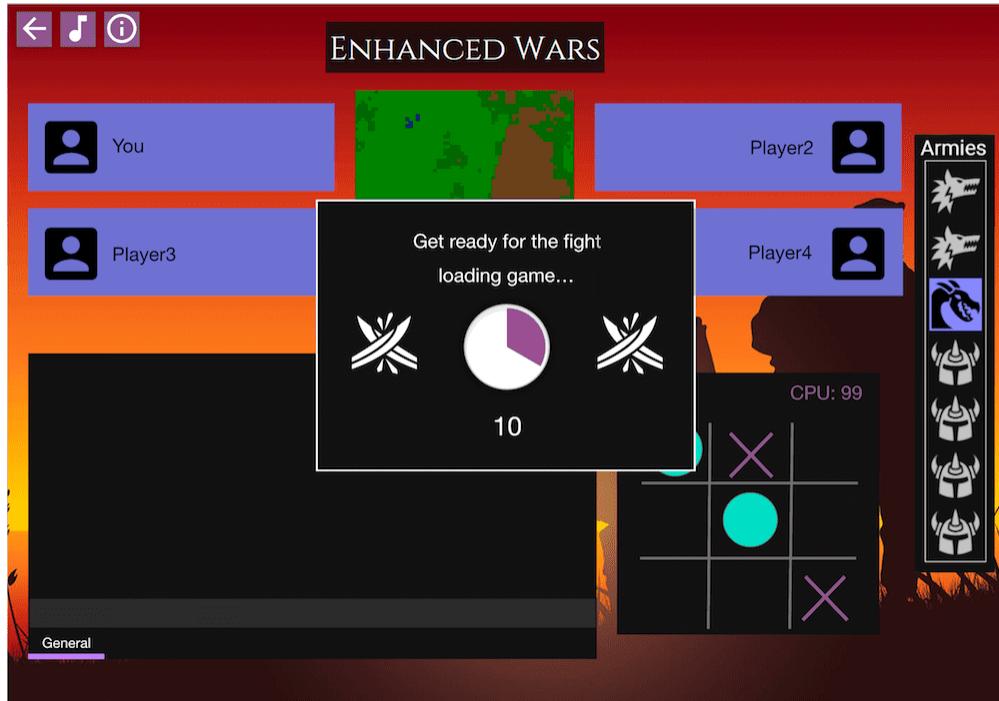


Abbildung 25: Spielstart

2.3.3 Ingame

Bisher zeigt die Ingame Szene nur das initiale Spielgeschehen. In diesem Release soll dies aber hauptsächlich erweitert werden. Zuerst sollen der Chat, die Minimap und die Spieler angezeigt werden. Diese Elemente müssen auch eingeklappt werden können über die jeweiligen Buttons oben rechts im Fenster. Die aktuelle Runde und die aktuelle Phase sollen oben mittig angezeigt werden. Der aktuelle Spieler soll in seiner Playercard mit der secondary Farbe hinterlegt werden. Links oben kommt der Musik Button zum ein- und ausschalten der Musik hinzu.

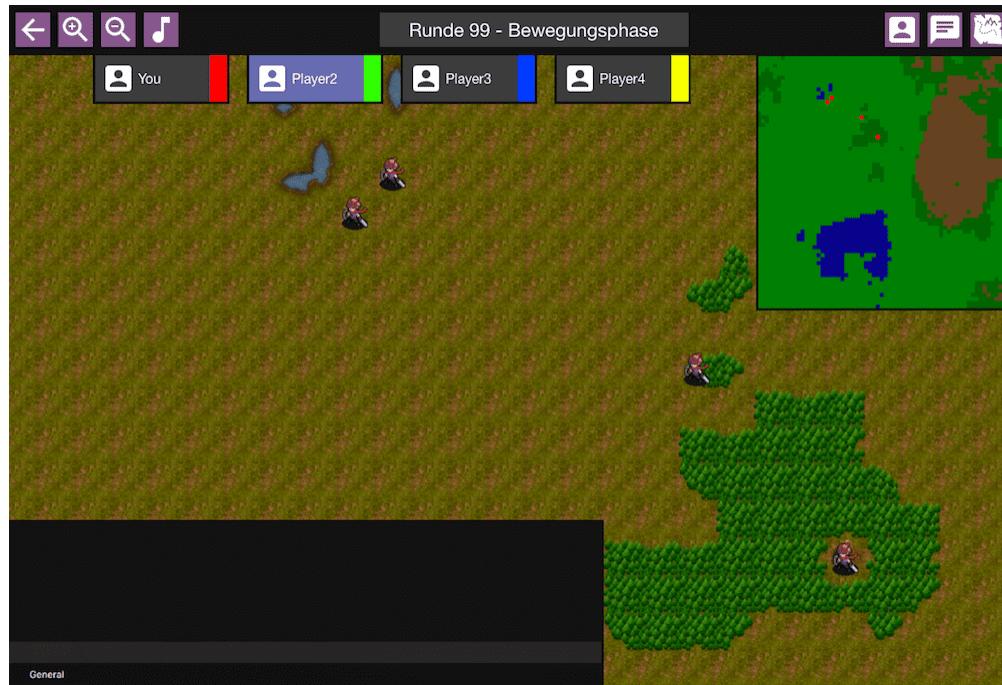


Abbildung 26: Ingame

2.3.3.1 Einheit auswählen

Der Server stellt den Spielern drei Phasen zur Verfügung, wenn ein Spieler an der Reihe ist. Eine Bewegungsphase, in der mindestens eine Einheit bewegt werden muss, eine Angriffsphase, die übersprungen werden kann und eine restliche Bewegungspunktephase, die übersprungen werden kann.



Abbildung 27: Phase 1: Auswählen

Danach beendet der jeweilige Spieler seine Runde. Wenn der User selbst an der Reihe ist und in einer seiner drei Phasen auf eine Einheit klickt soll der Bewegungsradius oder Angriffsradius angezeigt werden.



Abbildung 28: Phase 2: Auswählen



Abbildung 29: Phase 3: Auswählen

In Phase 1 soll zusätzlich zum Bewegungs- auch der Angriffsradius angezeigt werden (siehe Abbildung 27). In Phase 2 soll nur der Angriffsradius angezeigt werden (siehe Abbildung 28). In Phase 3 soll ein verkürzter Bewegungsradius ohne Angriffsradius angezeigt werden (siehe Abbildung 29).

2.3.3.2 Einheit bewegen

Nachdem der User in Phase 1 oder 3 eine Einheit ausgewählt hat, und er über die Felder hovert, auf die er seine Einheit stellen könnte, muss der kürzeste Pfad berechnet werden. Ein Pfeil soll dahin angezeigt werden. Klickt der User auf ein Feld, bewegt sich seine Einheit dorthin.



Abbildung 30: Bewegen (Phase 1)

Später soll das Kontextmenü dazu ergänzt werden und dem User die Möglichkeit geben seinen Zug doch noch zu widerrufen. Ist ein Gegner an der Reihe und der Server sendet eine bewegte Einheit, dann soll auch erst die Bewegungsreichweite angezeigt werden und danach (kurz verzögert) die Einheit bewegt werden.

2.3.3.3 Einheit angreifen

Nachdem der User in Phase 2 eine Einheit ausgewählt hat, kann er eine gegnerische Einheit neben ihm auswählen (insofern eine vorhanden ist) und sie angreifen. Später soll dies auch noch über ein Kontextmenü bestätigt werden, um dem User die Möglichkeit zu geben, seinen Angriff doch nicht durchzuführen. Weiterhin soll ein zweites Kontextmenü angezeigt werden, indem der User nochmal alle Properties beider Einheiten sieht und somit doppelt entscheiden kann, ob er den Angriff durchführt. Ist ein Gegner in seiner

Phase 2, soll bei einem Angriff von ihm, wenn er vom Server empfangen wird, auch die Angriffsreichweite angezeigt werden und kurz danach der Angriff durchgeführt werden.



Abbildung 31: Angreifen



Abbildung 32: Angreifen bestätigen

2.3.3.4 Spielende

Wenn ein Spieler verloren hat, da er keine Einheiten mehr hat, soll das Icon und der Name in seiner Playercard schwarz werden. Bei diesem User soll ein Overlay angezeigt werden,

in dem steht, dass er verloren hat. Außerdem kann er über Buttons entscheiden, ob er sich ausloggt, in die Lobby geht oder dem Spiel als Spectator beitritt (siehe Abbildung 33). Wenn nur noch ein Spieler übrig ist, hat er das Spiel gewonnen. Diesem User soll auch ein Overlay angezeigt werden, in dem er mit Buttons sich ausloggen, zurück zur Lobby gehen oder das Spiel in den sozialen Medien teilen kann (optionales Feature, siehe Abbildung 34). Bei anderen Usern, die im Spectatormodus sind, soll auch ein ähnliches Overlay mit den gleichen Buttons angezeigt werden (siehe Abbildung 35).

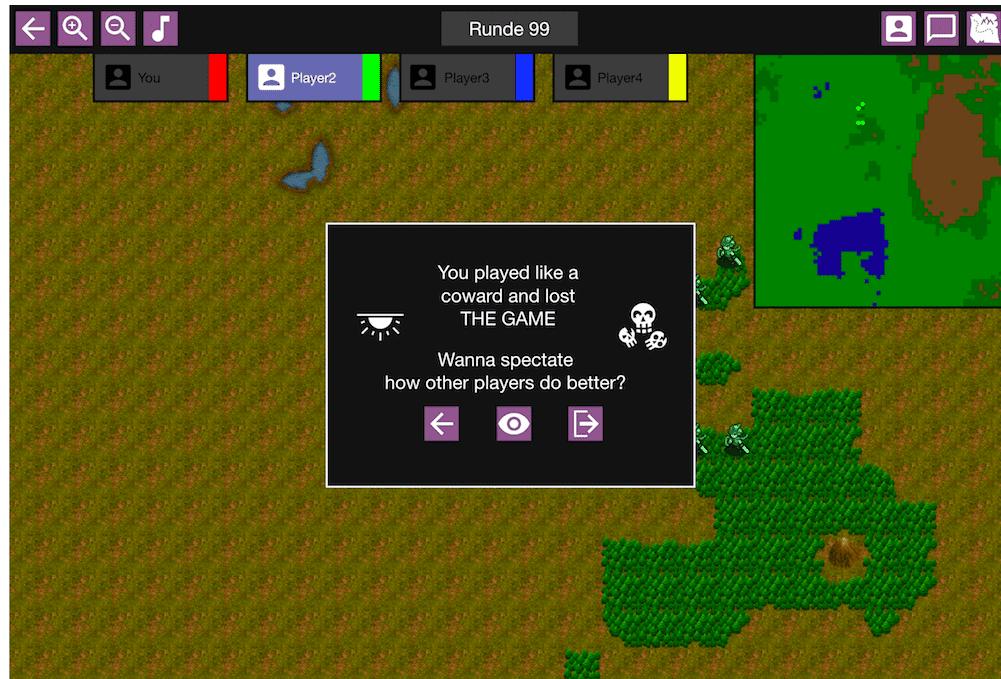


Abbildung 33: Spiel verloren



Abbildung 34: Spiel gewonnen



Abbildung 35: Beobachtungsmodus: Spiel vorbei

2.3.3.5 Kontextmenu

bla

2.3.3.6 Weitere Features

Es soll für den User möglich sein, über einen Button seine jeweilige Phase zu beenden und in die nächste zu wechseln. Nach Beenden aller drei Phasen, soll auch die Runde beendet werden. Wenn ein Gegner an der Reihe ist, muss der Phase beenden Button disabled sein. Beim Beenden der Phasen soll immer ein Overlay angezeigt werden (siehe Abbildung 36).



Abbildung 36: Phase beenden

Ein weiteres Feature ist, dass wenn der User über irgendeine Einheit hovert, alle Properties der Einheit angezeigt werden (siehe Abbildung 37).



Abbildung 37: Property Info

2.4 Domain Stories

Domain Stories wurden einmal zum Ready geben für den Waiting Room und einmal für das Bewegen von Einheiten Ingame angefertigt.

Die Domain Stories wurden vor dem Release der Serverdokumentation für das Kundentreffen erstellt. Deswegen zeigen sie leichte Abweichungen vom tatsächlichen Verhalten des Servers. Zum Beispiel sendet der Server Spiel starten erst, wenn irgendein Spieler den Befehl zum Spiel starten an den Server schickt, nachdem alle eine Ready gegeben haben. Wir nahmen an, dass der Server automatisch das Spiel startet, nachdem alle ready sind. Unsere Move Story trifft teilweise den Server. Der Unterschied zu unserer Story ist, dass wir vorher nicht davon ausgingen, dass eine Runde des Spielers in die drei Phasen unterteilt ist. Deswegen zeigt unsere Story auf dem MockUp oben rechts schon die bewegte Einheit, mit der jetzt eigentlich angegriffen werden sollte. Da dies vom Server her nicht funktioniert, müssen wir dies auf die drei Phasen unterteilen.

3 Sprint V

Der fünfte Sprint erstreckte sich über dem Zeitraum vom 8.7.2019 bis zum 21.7.2019. Es wurden 116 Storypoints für alle User Stories geschätzt.

3.1 Sprintziel

Es sollte nach dem fünften Sprint möglich sein:

- Dem Spiel im Waiting Room korrekt beizutreten
- Das Spielfeld und die Minimap mit den Einheiten darauf zu sehen
- Eine Einheit auszuwählen, anzugreifen und zu bewegen
- Den Spielstatus zu sehen und Phasen beenden zu können

Wenn noch genug Zeit übrig sein sollte, sollte die Ingame Chatintegration und das Spieldende noch implementiert werden.

3.2 User Stories

Die User Stories, die vom Product Owner erstellt wurden, sind wie gefolgt aufgebaut: Im Titel kann man sehen, zu welcher Szene die Story gehört. Anschließend werden die Story Points (Ein Storypoint entspricht einer Stunde) und der zugeteilte Entwickler genannt. Danach folgt das Ziel und die Story selbst. Das Ziel fasst die Subtasks, die der Scrum Master erstellt hat, zusammen.

3.2.1 Waiting Room - Spielbeitritt anpassen

Zuteilung Diese User Story wurde auf 5 Storypoints geschätzt und an Omar Sood zugeordnet.

Ziel Der Entwickler soll das Ready geben implementieren, wenn eine Armee ausgewählt wurde und das Spiel muss starten, sobald alle Spieler bereit sind.

Story Karli ist einem Spiel beigetreten und befindet sich im Wartreraum. Alle anderen Spieler sind bereit. Karli wählt eine Armee aus. Da nun alle Spieler bereit sind, startet das Spiel (und es findet ein Szenenwechsel statt).

3.2.2 Ingame - Spielfeld

Zuteilung Diese User Story wurde auf 13 Storypoints geschätzt und an Georg Siebert zugewieilt.

Ziel Der Entwickler soll es möglich machen, dass Einheiten angezeigt werden und dass jedes Feld klickbar ist.

Story Karli ist bereits einem Spiel beigetreten und sieht das Spielfeld. Karli sieht die Einheiten auf dem Spielfeld. Karli möchte nun auf ein Feld (einer Einheit) klicken.

3.2.3 Ingame - Phase beenden

Zuteilung Diese User Story wurde auf 3 Storypoints geschätzt und an Juri Lozowoj zugewieilt.

Ziel Der Entwickler soll einen Phase beenden Button hinzufügen und die jeweiligen Servernachrichten zum Phase beenden senden und abfangen können.

Story Karli befindet sich bereits in einem Spiel. Karli hat gerade einen Teil ihrer Armee bewegt (oder mit einem Teil angegriffen) und möchte nun die jeweilige Phase beenden. Dafür klickt Karli auf den entsprechenden Button (siehe Mockup). Die Phase wird beendet und die nächste tritt ein (bzw. der nächste Spieler ist an der Reihe).

3.2.4 Ingame - Einheit auswählen

Zuteilung Diese User Story wurde auf 13 Storypoints geschätzt und an Juri Lozowoj zugewieilt.

Ziel Der Entwickler soll das Anklicken und Auswählen der eigenen Einheiten implementieren. Danach soll die Bewegungs- und Angriffsreichweite sichtbar sein, je nachdem in welcher Phase der User sich befindet. Dies soll später auch für die Gegner angezeigt werden, wenn sie an der Reihe sind.

Story Karli befindet sich bereits in einem Spiel. Karli wählt eine Einheit durch Klicken auf ihre Graphik aus. Gültige Bewegungsziele werden blau und gültige Angriffsziele rot markiert (siehe Mockups).

3.2.5 Ingame - Einheit bewegen

Zuteilung Diese User Story wurde auf 13 Storypoints geschätzt und an Omar Sood zugeteilt.

Ziel Der Entwickler soll den kürzesten Pfad berechnen, wenn eine Einheit bewegt werden soll und muss die jeweiligen Nachrichten an den Server senden und auch abfangen können.

Story Karli befindet sich in der Bewegungsphase oder der restlichen Bewegungspunktephase und hat bereits auf eine Einheit geklickt und sieht alle Felder, auf welche die Einheit bewegt werden könnte (je nachdem ob Karli in der 1. oder 3. Phase ist, mit oder ohne rote Felder). Karli klickt auf ein Feld in Bewegungsreichweite und die Einheit wird bewegt.

3.2.6 Ingame - Einheit angreifen

Zuteilung Diese User Story wurde auf 5 Storypoints geschätzt und an Omar Sood zugeteilt.

Ziel Der Entwickler soll bei einem Angriff die jeweiligen Nachrichten an den Server senden und auch abfangen können.

Story Karli ist in der Angriffsphase und hat bereits eine Einheit ausgewählt. Karli sieht alle Felder in Rot, die angegriffen werden könnten. Karli wählt eine gegnerische Einheit auf diesen Feldern aus und greift sie an.

3.2.7 Ingame - Spielstatus anzeigen

Zuteilung Diese User Story wurde auf 8 Storypoints geschätzt und an Tobias Klipp zugeteilt.

Ziel Der Entwickler soll einen Button zum Ein- und Ausblenden der Spielerkarten Ingame implementieren. Der aktive Spieler soll hervorgehoben werden.

Story Karli befindet sich bereits in einem Spiel. In der Menuleiste sieht Karli einen Button zum Anzeigen der Spieler und die aktuelle Rundennummer. Karli klickt auf diesen Button. Eine Anzeige der Spieler wird eingeblendet (siehe Mockup). Der aktive Spieler ist hervorgehoben (in der selected-Color siehe Stylesheet bzw. MockUp).

3.2.8 Ingame - Minimap anzeigen

Zuteilung Diese User Story wurde auf 8 Storypoints geschätzt und an Tobias Klipp zugeteilt.

Ziel Der Entwickler soll einen Button zum Ein- und Ausblenden der Minimap implementieren. Die Minimap muss gegebenfalls auch (vom Waiting Room) implementiert werden.

Story Karli befindet sich bereits in einem Spiel. In der rechten oberen Ecke sieht Karli die Minimap, welche neben dem Terrain auch alle Einheiten anzeigt. Karli drückt den Ausblenden Button, um den Bereich hinter der Karte zu sehen. Die Karte wird ausgeblendet.

3.2.9 Ingame - Chatintegration

Zuteilung Diese User Story wurde auf 13 Storypoints geschätzt und an Tobias Klipp zugewieitet.

Ziel Der Entwickler soll einen Button zum Ein- und Ausblenden des Chats implementieren. Die Chat muss gegebenenfalls auch implementiert werden.

Story Karli befindet sich bereits in einem Spiel. Karli blendet den Chat durch Betätigen des zugehörigen Buttons ein (siehe Mockup). Karli schreibt eine Nachricht an die anderen Spieler. Karli kann die Antworten der anderen Spieler lesen.

3.2.10 Ingame - Game Over

Zuteilung Diese User Story wurde auf 2 Storypoints geschätzt und an Tobias Klipp zugewieitet.

Ziel Der Entwickler soll ein Overlay anzeigen, wenn ein Spieler verloren hat, auf dem der User in die Lobby oder den Login wechseln kann. Der User soll auch weiter zuschauen können. Seine Spielerkarte soll sich dunkel färben.

Story Karli hat keine Einheiten mehr und verliert das Spiel. Karli sieht, wie sich Karlis Playercard schwarz färbt (Name und Icon). Karli kann auswählen, ob in die Lobby gewechselt oder ob der Logout gerufen oder ob dem Spiel als Spectator beigetreten wird.

3.2.11 Ingame - Game Won

Zuteilung Diese User Story wurde auf 2 Storypoints geschätzt und an Tobias Klipp zugewieitet.

Ziel Der Entwickler soll ein Overlay anzeigen, wenn ein Spieler gewonnen hat, auf dem der User in die Lobby oder den Login wechseln kann.

Story Karli hat als einziger noch Einheiten und gewinnt das Spiel. Karli kann auswählen, ob in die Lobby gewechselt oder ob der Logout gerufen wird.

3.2.12 Lobby - Spectating

Zuteilung Diese User Story wurde auf 5 Storypoints geschätzt und an Juri Lozowoj zugewieitet.

Ziel Der Entwickler soll einen Button hinzufügen, der den User als Beobachter auf das Spielfeld weiterleitet.

Story Karli hat sich bereits eingeloggt und sieht die Lobby. Karli sieht die Spielliste und möchte dieses mal nicht selbst spielen. Karli drückt auf den Spactator-Button und wechselt nun zum Spielfeld.

3.2.13 Ingame - Spactating

Zuteilung Diese User Story wurde auf 13 Storypoints geschätzt und an Juri Lozowoj zugeteilt.

Ziel Der Entwickler soll die Ingame Szene aufbauen, ohne dass der User dort im Chat mitschreiben oder das Spielfeld bedienen kann.

Story Karli sieht bereits das Spielfeld. Karli kann weder Aktionen auf dem Spielfeld ausführen, noch im Chat mitschreiben. Karli merkt aber, dass alles andere funktioniert, holt Popcorn und genießt das Spiel.

3.2.14 Ingame - Spectator Over

Zuteilung Diese User Story wurde auf 2 Storypoints geschätzt und an Tobias Klipp zugeteilt.

Ziel Der Entwickler soll ein Overlay anzeigen, wenn ein Spieler gewonnen hat und das Spiel vorbei ist. Auf dem Overlay kann der User in die Lobby oder den Login wechseln.

Story Karli sieht, dass Bob als einziger noch Einheiten hat und das Spiel endet. Karli kann auswählen, ob in die Lobby gewechselt oder ob der Logout gerufen wird.

3.2.15 Ingame - Einheiteninformationen

Zuteilung Diese User Story wurde auf 8 Storypoints geschätzt und an Georg Siebert zugeteilt.

Ziel Der Entwickler soll ein UI Element erstellen, dass die Einheiteninformationen anzeigt, wenn man über die jeweilige Einheit hovert.

Story Karli befindet sich bereits in einem Spiel. Karli fährt mit seinem Mouse-Courser über eine Einheit. Es wird Karli (während die Maus auf der Einheit verweilt) eine Infobox mit wichtigen Informationen über diese Einheit angezeigt (siehe Mockups). Die Box sollte sich später, wie in TG-274 beschrieben, positionieren.

3.2.16 Ingame - Musiksteuerung

Zuteilung Diese User Story wurde auf 3 Storypoints geschätzt und an Omar Sood zugeteilt.

Ziel Der Entwickler soll einen Musik Button hinzufügen und die Musik soll sich genauso bedienen lassen, wie in den restlichen Szenen.

Story Karli befindet sich bereits in einem Spiel. Karli möchte die Musik ausschalten. Karli klickt auf den dafür vorgesehenen Button (siehe Mockup). Die Musik wird ausgeschaltet.

3.3 Tasks

Die Tasks wurden vom Scrum Master erstellt und sind nicht aus Usersicht beschreibbar, da sie nur technische Aufgaben sind. Aus diesem Grund unterscheiden sich Tasks nur in einer Sache von User Stories. Sie haben keine Story.

3.3.1 Servernachrichten

Zuteilung Diese Task wurde auf 8 Zeitstunden geschätzt und an Omar Sood zugeteilt.

Ziel Der Entwickler soll alle restlichen Servernachrichten abfangen, die in keiner User Story berücksichtig wurden. Dazu gehören unter anderem Änderungen des Datenmodells, welche durch Aktionen der Gegenspieler anfallen.

3.3.2 Einheiteninformationen im Datenmodell

Zuteilung Diese Task wurde auf 5 Zeitstunden geschätzt und an Tobias Klipp zugeteilt.

Ziel Der Entwickler soll die Enums UnitType und UnitTypeInfo zu einer Enum zusammenfügen und darauf achten, dass alle Funktionalitäten erhalten bleiben.

3.3.3 Ingame - Kontextmenü-Framework

Zuteilung Diese Task wurde auf 8 Zeitstunden geschätzt und an Georg Siebert zugeordnet.

Ziel Der Entwickler soll ein Kontext Framework erstellen.

3.4 Zeitübersicht

User Story/Task	Soll Zeit	Ist Zeit	Noch Zeit	Entwickler
Waiting Room - Spielbeitritt anpassen	5 h	13 h 9 min	-	Omar Sood
Ingame - Spielfeld	13 h	8 h 22 min	-	Georg Siebert
Ingame - Phase beenden	3 h	15 h 20 min	-	Juri Lozowoj
Ingame - Einheit auswählen	13 h	8 h 46 min	4h 14 min	Juri Lozowoj
Ingame - Einheit bewegen	13 h	10 h 54 min	1 h	Omar Sood
Ingame - Einheit angreifen	5 h	-	5 h	Omar Sood
Ingame - Spielstatus anzeigen	8 h	18 h 45 min	-	Tobias Klipp
Ingame - Minimap anzeigen	8 h	-	8 h	Georg Siebert
Ingame - Chatintegration	13 h	-	13 h	Tobias Klipp
Ingame - Game Over	2 h	-	2 h	Tobias Klipp
Ingame - Game Won	2 h	-	2 h	Tobias Klipp
<hr/>				
Lobby - Spectating	5 h	-	5 h	Juri Lozowoj
Ingame - Spectating	13 h	-	13 h	Juri Lozowoj
Ingame - Spectator Over	2 h	-	2 h	Tobias Klipp
Ingame - Einheiteninformationen	8 h	-	8 h	Georg Siebert
Ingame - Musiksteuerung	3 h	-	3 h	Omar Sood
<hr/>				
Servernachrichten	8 h	-	8 h	Omar Sood
Einheiteninformationen im Datenmodell	5 h	-	5 h	Tobias Klipp
Ingame - Kontextmenü-Framework	8 h	-	8 h	Georg Siebert

Tabelle 1: Zeitübersicht fünfter Sprint