

UNIVERSITÄT KASSEL

PROJEKTDOKUMENTATION

RBSG - Release II

SOFTWARE ENGINEERING I SS19

GRUPPE G

Scrum Master: Omar Sood

Product Owner: Juri Lozowoj



15. August 2019

Diese Dokumentation beschreibt das zweite Release des Projektes RBSG von Team G. Dies umfasst die Auflistung und Erläuterung der Mockups und der Domain Stories. Beide Sprints werden dokumentiert und analysiert. Zum Schluss wird das Resultat des Releases mit den Zielen verglichen und es wird auf Probleme oder nicht erfüllte Anforderungen eingegangen.

Inhaltsverzeichnis

1	Ziel der Dokumentation	1
2	Das Projekt RBSG	1
3	Das Release II	1
3.1	Anforderung an das zweite Release	2
3.2	Stand zum Ende des Release I	2
3.2.1	Login	2
3.2.2	Lobby	3
3.2.3	Spiel erstellen	4
3.2.4	Spiel beitreten	4
3.2.5	Lobby Chat	5
3.2.6	Umgesetzte optionale Features	5
3.3	ZeitÜbersicht	6
3.4	Analyse des 7. Sprints	7
3.4.1	Burndown-Diagramm	7
3.4.2	Ausreiser	7
3.4.3	Abgeschlossene Vorgänge	8
3.4.4	Nicht abgeschlossene Vorgänge	8
3.4.5	Entfernte Vorgänge	9
3.4.6	Fazit	10
4	Sprint IV	10
4.1	Sprintziel	10
4.2	User Stories	10
4.2.1	Core - ArmyBuilder - Speichern der Armeen	10
4.2.2	Core - ArmyBuilder - Löschen der Armee (Clear)	11
4.2.3	Appearance - ArmyBuilder	11
4.2.4	Optional - Ingame - Zoomen auf dem Spielfeld	11
4.2.5	Core - Ingame - Spiel verlassen	12
4.2.6	Optional - Waiting Room - Anzeigen der Karte (Preview)	12
4.2.7	Core - Login - Default Armeen erstellen und User auf erstellen von Armeen hinweisen	13
4.2.8	Core - Lobby - Deaktivieren der JoinGame-Buttons und des Create-Game-Buttons bei keiner ausgewählten Armee	13
4.2.9	Optional - ArmyBuilder - Bearbeiten der Armeeeinformationen	13
4.2.10	Optional - ArmyBuilder - Armeeeauswahl aus der Lobby Übernehmen	14
4.2.11	Optional - Create Game - Auto Join	14
4.2.12	Appearance - Create Game - Anpassen des Style's des Create-Game-Formulars	15
4.2.13	Core - Waiting Room - Ein Spieler tritt dem Spiel bei	15
4.2.14	Core - Waiting Room - Ein Spieler verlässt das Spiel	15
4.2.15	Optional - Waiting Room - „Player-Kasten“ wird mit der Spielerfarbe dargestellt	16
4.2.16	Appearance - Alerts	16
4.2.17	Core - ArmyBuilder - „CanAttack“ Übersicht	16

4.2.18	Optional - ArmyBuilder- Indikator ob Speichern nötig ist und Aktivieren des Save Buttons	17
4.2.19	Core - ArmyBuilder - Laden der Armeen	17
4.2.20	Optional - Chat - Anzeigen das eine Nachricht angekommen ist	17
4.2.21	Optional - ArmyBuilder - Bearbeiten der Armeeeinformationen	18
4.2.22	Core - ArmyBuilder - Autosave der Armeen bei verlassen des ArmyBuilders	18
4.2.23	Core - ArmyBuilder - Can Attack Übersicht	19
4.2.24	Optional - ArmyBuilder- Indikator ob Speichern nötig ist und Aktivieren des Save Buttons	19
4.2.25	Optional - Waiting Room - Anzeigen der Einheiten der ausgewählten Armee	19
4.2.26	Optional - Lobby - Anzeigen der Einheiten einer Armee	20
4.2.27	Optional - ArmyBuilder- Notifiaction für das Speichern der Armee und Waiting Animation.	20
4.2.28	Core - ArmyBuilder - Lokaler Speicherort für Armeen	20
4.3	Tasks	20
4.3.1	Task - SceneManger Überarbeiten	21
4.3.2	Task - Neues Musikstück einpflegen	21
4.3.3	Task - Spiel verlassen	21
4.3.4	Task - NullPointerException - Spielbeitritt	21
4.3.5	Task - Reaktion auf 503 vom Server	22
4.3.6	Task - ArmyManager Lore	22
4.3.7	Task - Zwischenrelease v0.3	22
4.3.8	Task - Speicherstände mergen	22
4.4	Bugs	23
4.4.1	Bug - lokales Speichern unter Windows	23
4.5	Zeitübersicht	24
4.6	Analyse des 4. Sprint	25
4.6.1	Burndown-Diagramm	25
4.6.2	Abgeschlossene Vorgänge	26
4.6.3	Nicht abgeschlossene Vorgänge	27
4.6.4	Fazit	28
5	Abschluss des 2. Releases	28
5.1	Der Armeemanager	29
5.1.1	Implementaion	29
5.1.2	Armee-Editor	30
5.1.3	Info zu den Attributen	31
5.2	Warteraum	32
5.3	Ingame	33
5.4	Lobby	34
5.4.1	Spiel erstellen Formular	35
5.5	Alerts	36

1 Ziel der Dokumentation

Die Dokumentation soll den Verlauf des Entwicklungsprozesses des Spiels RBSG für das zweite Release veranschaulichen und analysieren. Dies umfasst die Entstehung der Mockups und Domain Stories, welche die Grundlagen für die Entwicklung bilden. Es werden für alle Sprints das Sprintziel, die User Stories sowie eine Analyse des Sprintverlaufs erstellt. Abschließend werden die Mockups mit den Ergebnissen des Sprints verglichen. Dabei wird darauf eingegangen, wieso Features anders umgesetzt oder weggelassen wurden.

2 Das Projekt RBSG






Das Projekt RBSG findet im Rahmen der Veranstaltung Software Engineering I des Fachgebiets Software Engineering statt. Im Rahmen dieser Veranstaltung soll das Spiel RBSG in einzelnen Gruppen umgesetzt werden. Es basiert auf dem Spielprinzip des Nintendospieles Advanced Wars. Die Entwicklung ist dabei in vier Releases eingeteilt. Ein Release besteht jeweils aus zwei Sprints. Die Teams arbeiten und organisieren sich im Rahmen der agilen Projektmanagementmethode Scrum. In jedem Team werden entsprechend die Rollen des Product Owners und des Scrum Masters besetzt. Die restlichen Teammitglieder sind Entwickler.

3 Das Release II

Das zweiten Release knüpfte direkt an das Ende des ersten Releases an. Es begann am 10.06.2019 und endete am 09.07.2019. Der Entwicklungsprozess ist in zwei Sprints eingeteilt:

- Sprint III 10.06.2019 bis zum 23.06.2019
- Sprint IV 24.06.2019 bis zum 07.07.2019

Im zweiten Release gab es folgende Rollenverteilung:

- Product Owner:  Georg Siebert
- Scrum Master:  Juri Lozowoj
- Entwickler:
 -  Omar Sood
 -  Tobias Klipp
 -  Jan Müller
 -  Keanu Stückrad

3.1 Anforderung an das zweite Release

An das Release wurde folgende Anforderungen vom Kunden gestellt:

1. Armeemanager
 - Erstellen von Armeen
 - Speichern und Laden von Armeen (lokal und Server)
 - Es sollen mindestens 3 Armeen konfigurierbar sein
 - Es sollen 10 Einheiten pro Armee ausgewählt werden können
2. Client
 - es soll einem Spiel beigetreten werden können
 - der Client verarbeitet die Nachrichten vom Server korrekt
 - Es soll zurück zur Lobby navigiert werden können
 - ein Ingame Chat soll zur Verfügung stehen
3. Testabdeckung von 60%

3.2 Stand zum Ende des Release I

Das erste Release endete am 09.06.2019. Es wurden bereits folgende Komponenten fertiggestellt:

- Login
- Lobby
- Erstellen eines Spiels
- Beitreten eines Spiels ¹

3.2.1 Login

Im Login konnte ein Nutzer sich registrieren und anmelden. Dies ist auf Abb.1 zu sehen. Loggt sich ein Nutzer ein, wird ein animiertes Ladesymbol eingeblendet. Der Untertitel „Enhanced Wars“ ist animiert. Die Oberfläche ist internationalisiert.

¹Das Beitreten eines Spiels wurde auf dem Stand der Serverdoku Release I realisiert. Dort war ein Anmelden an den Websocket des Servers noch nicht möglich

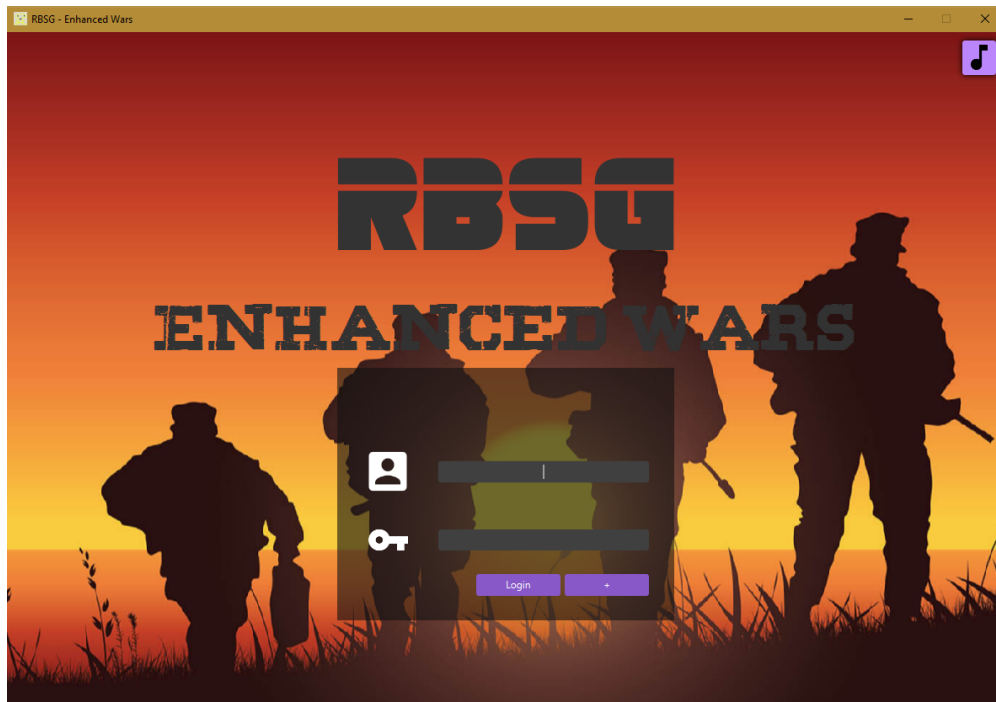


Abbildung 1: Login Release I

3.2.2 Lobby

Die Lobby zu Beginn des zweiten Releases wird in Abb. 2 dargestellt. In der Lobby werden die Spielerlisten und Spiellisten angezeigt. Diese aktualisieren sich, wenn der Server eine entsprechende Nachricht sendet. Es ist möglich über den Spiel erstellen-Knopf ein Spiel zu erstellen. Über den Logout-Knopf kann ein Nutzer sich abmelden.

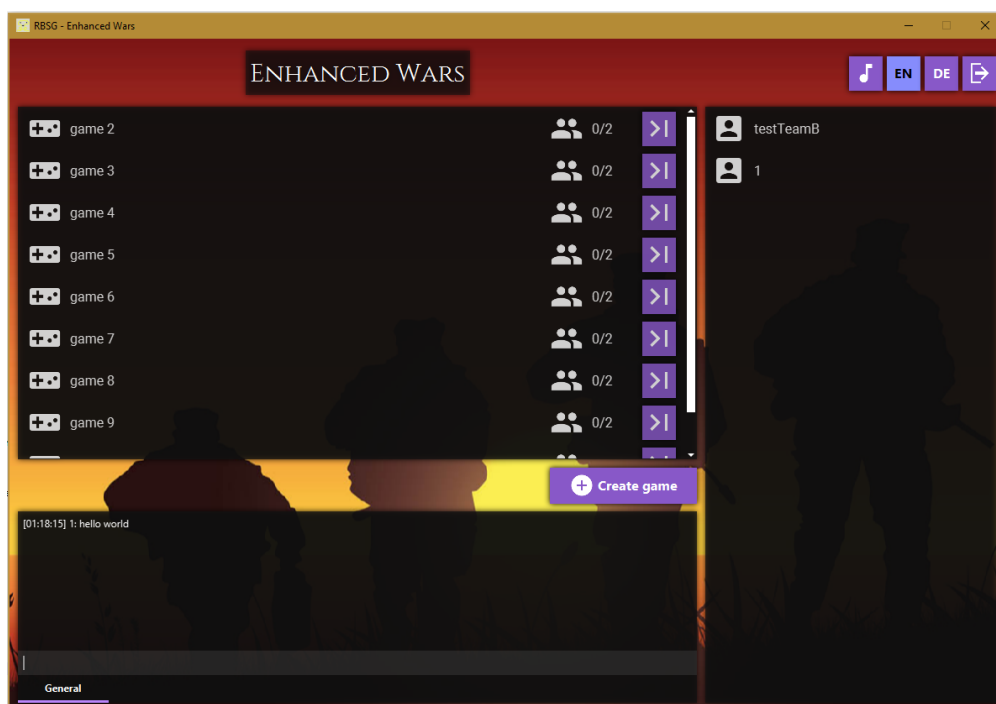


Abbildung 2: Lobby Release I

3.2.3 Spiel erstellen

Der in Abb. 3 dargestellte Spiel erstellen-Dialog wird über den entsprechenden Button in der Lobby aufrufen. Im Spiel erstellen-Fenster hat man die Möglichkeit, den Name des Spiels und die Anzahl der Spieler einzustellen.² Das Spiel wird über den Erstellen-Button erzeugt und über den Abbrechen-Button kann man den Dialog wieder verlassen.

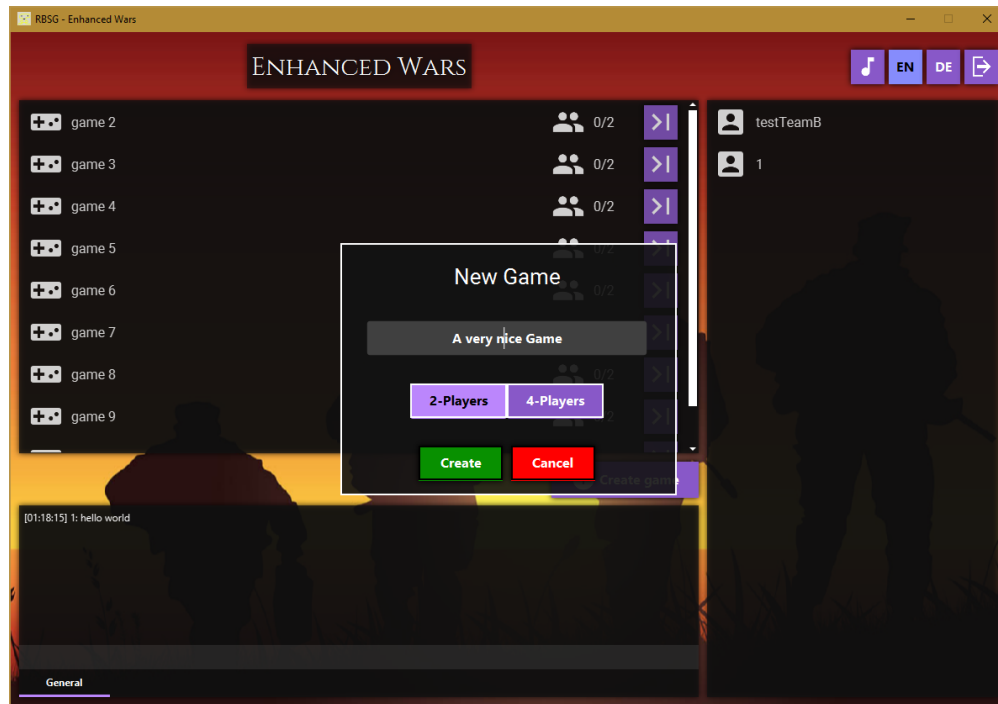


Abbildung 3: Spiel erstellen Release I

3.2.4 Spiel beitreten

Über den Knopf in der Spielliste kann man einem Spiel, entsprechend dem Serverstand des ersten Releases, beitreten. Nachdem man einem Spiel beigetreten ist, wechselt die Szene zum Warteraum des Spiels. Der Warteraum eines Spiels zum Stand nach dem ersten Release wird in Abb. 4 dargestellt.

²2-Spieler und 4-Spieler

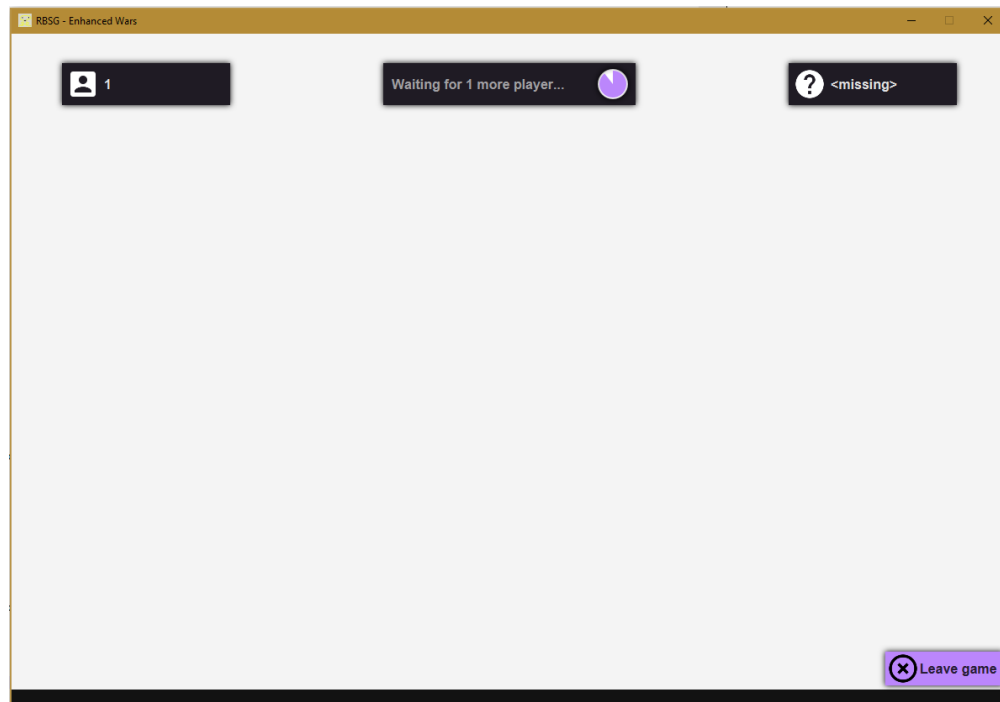


Abbildung 4: Warteraum eines Spiels Release I

3.2.5 Lobby Chat

Die Lobby verfügte bereits über einen Chat. Mit diesem ist es möglich Nachrichten in den General-Chat und in Private-Chats zu versenden.

3.2.6 Umgesetzte optionale Features

Im ersten Release wurden bereits folgende optionale Features umgesetzt:

- Internationalisierung, diese kann Über Buttons, siehe Abb. 5, in der Lobby eingestellt werden
- Musik, diese kann über einen Button (Abb. 6) an- und ausgestellt werden
- Ein Dark-Theme
- Über den Chat-Befehl „\chuckMe“ werden Chuck Norris Zitate an den Chat gesendet

Die Testabdeckung (C0) lag bei ~ 80 Prozent, womit die Anforderung für das zweite Release bereits grundlegend erfüllt ist. Das Ziel für das zweite Release war es, die Höhe der Testabdeckung aufrechtzuerhalten und weiter auszubauen.

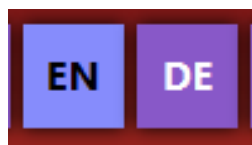


Abbildung 5: Button für die Internationalisierung Release I



Abbildung 6: Button für Musik Release I

3.3 ZeitÜbersicht

User Story	Story Points	Soll Zeit	Ist Zeit	Entwickler
Core - ArmyBuilder - Laden der Armeen	5	5h	16h 16min	Tobias Klipp
Core - ArmyBuilder - Speichern der Armeen	13	13h	34h 45min	Tobias Klipp
Core - ArmyBuilder - Hinzufügen der Einheiten	13	13h	7h 23m	Omar Sood
Core - ArmyBuilder - Entfernen der Einheiten	5	5h	39m	Omar Sood
Core - ArmyBuilder - zurück in die Lobby navigieren	3	3h	1h	Keanu Stückrad
Core - ArmyBuilder - Info Über Attribute	3	3h	5h	Keanu Stückrad
Core - Waiting Room - Anzeigen des Waiting Rooms (neues Layout)	5	5h	9h	Keanu Stückrad
Optional - ArmyBuilder - Musik an/aus	1	1h	30m	Keanu Stückrad
Core - Lobby - Auswahl der Armee	1	1h	18m	Omar Sood
Core - Ingame - Anzeigen der Spielfeldes	3	3h	7h 30m	Keanu Stückrad
Core - ArmyBuilder - Auswahl einer Einheit	5	5h	9h 59m	Omar Sood
Core - Waiting Room - Spiel beitreten	16	16h	15h 23m	Jan Müller
Appearance - ArmyBuilder	8	8h	-	Tobias Klipp
Core - Waiting Room - Chat	13	13h	6h 35m	Jan Müller
Core - ArmyBuilder - In den ArmyBuilder navigieren	3	3h	4h 7m	Omar Sood
Core - ArmyBuilder- Auswählen einer Armee	5	5h	1h 58m	Omar Sood

Tabelle 1: Übersicht der Zeiten für den 3. Sprint

3.4 Analyse des 7. Sprints

Der siebte Sprint wurde am 19.08.2019 beendet. Es wurden insgesamt - von 114 Story Points abgeschlossen. Demnach konnte nicht alle vorgesehenen User Stories abgeschlossen werden. Der Ablauf des siebten Sprints wird anhand des Burndown-Diagramms erläutert.

3.4.1 Burndown-Diagramm

In der ersten Woche entspricht das Burndown-Diagramm annähernd der dargestellten Idealkurve.

Betrachtet man zunächst das Burndown-Diagramm für den dritten Sprint, so nimmt in der ersten Woche die Anzahl der abgeschlossenen Story Points kaum zu. Ab der Mitte der zweiten Woche des Sprints beginnt der Graph schnell zu fallen und er endet mit den übrig gebliebenen 21 Story Points. Ein Grund für diesen Verlauf ist die Fehleinschätzung des Teams der zu erledigenden Arbeit, so wurde zum Beispiel das Erstellen der Karte auf drei Story Points geschätzt. Am Ende wurde für diese Task insgesamt 7,5 Stunden benötigt. Dieses Problem trat in diesen Sprint häufiger auf, da die Tasks vor Veröffentlichung der Serverdokumentation des zweiten Releases geschätzt wurden. Ein weiteres Problem, welches am Anfang des Sprints auftrat, waren die unklaren Anforderung an den Sprint, so wurden Features geplant welche mit dem damaligen Serverstand nicht möglich waren. Darauf konnten wir jedoch gut reagieren, indem wir den Entwicklern erst Tasks zuwiesen, welche diese bereits abarbeiten konnten. Während dessen arbeiten wir die restlichen Tasks auf und passten sie an den Serverstand an. Insgesamt können wir hier jedoch von einem guten Abarbeitung der Tasks sprechen, da die geforderten 80 Story Points erfüllt wurden, und einige Tasks sich im Nachhinein als noch aufwändiger heraus stellten.

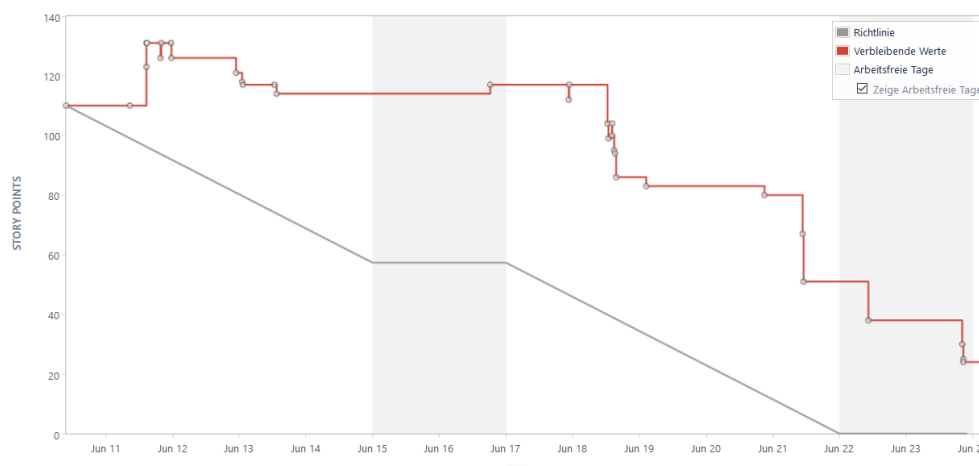


Abbildung 7: Burndown-Diagramm des Sprint 3

3.4.2 Ausreiser

Am 10. August wurde eine User Story im Umfang von drei Story Punkten nachträglich in den siebten Sprint aufgenommen. Es hatte sich nach der ersten Implementierung der “Künstliche Intelligenz“ (KI) gezeigt, dass ein zu einem sinnvollen Verhalten der KI die Betrachtung, der in User Story TG-414 beschriebenen Situation notwendig, ist.

Am 11. August wurde eine bereits abgeschlossene User Story (TG-409) erneut geöffnet. Dies war notwendig, da erst bei der Überprüfung des Pull Requests auffiel, dass dieses Feature nicht wie in der User Story vorgegeben implementiert war. Daher musste der Entwickler die User Story TG-409 erneut überarbeiten.

3.4.3 Abgeschlossene Vorgänge

In Abb. 9 sind die abgeschlossenen Vorgänge für den siebten Sprint zu sehen. Die an eine "Künstliche Intelligenz" gestellten Anforderungen waren nach dem siebten Sprint bereits erfüllt.

Künstliche Intelligenz -

Ein C0 Abdeckung von -% Zum Ende des dritten Sprints ist eine Code Coverage von -% vorhanden. Siehe Abb. 8.

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	97,7% (128/ 131)	75,3% (522/ 693)	68,9% (2076/ 3011)

Abbildung 8: Code Coverage vom 3. Sprint

Abgeschlossene Vorgänge		Im Vorgangsnavigator anzeigen			
Schlüssel	Zusammenfassung	Vorgangstyp	Priorität	Status	Story Points (77 → 89)
TG-109	Core - ArmyBuilder - Speichern der Armeen	Story	Highest	FERTIG	13
TG-113	Core - ArmyBuilder - Hinzufügen der Einheiten	Story	Medium	FERTIG	13
TG-115	Core - ArmyBuilder - Entfernen der Einheiten	Story	Medium	FERTIG	5
TG-117	Core - ArmyBuilder - Zurück in die Lobby navigieren	Story	High	FERTIG	3
TG-121	Core - ArmyBuilder - Info über Attribute	Story	Low	FERTIG	3
TG-123	Optional - ArmyBuilder - Musik an/aus	Story	Medium	FERTIG	1
TG-125	Core - Waiting Room - Anzeigen des Waiting Rooms (neues Layout)	Story	High	FERTIG	5
TG-126	Core - Lobby - Auswahl der Armee	Story	High	FERTIG	5 → 1
TG-130	Core - Ingame - Anzeigen der Spielfeldes	Story	Medium	FERTIG	3
TG-177	Core - ArmyBuilder - Auswahl einer Einheit	Story	Medium	FERTIG	5
TG-178	Core - Waiting Room - Spiel beitreten	Story	High	FERTIG	- → 16
TG-180	Core - Waiting Room - Chat	Story	High	FERTIG	13
TG-181	Core - ArmyBuilder - In den ArmyBuilder navigieren	Story	Medium	FERTIG	3
TG-194 *	Tests optimieren	Aufgabe	Medium	FERTIG	-
TG-195 *	Pull Request - stylingLoginCreate	Aufgabe	Highest	FERTIG	-
TG-201 *	Core - ArmyBuilder - Auswählen einer Armee	Story	Highest	FERTIG	5

Abbildung 9: Abgeschlossene Vorgänge vom 3. Sprint

3.4.4 Nicht abgeschlossene Vorgänge

Im dritten Sprint wurden User Stories im Umfang von insgesamt 18 Story Points nicht abgeschlossen. Die entsprechenden Stories sind in Abb. 10 aufgelistet. Die Tasks werden in den vierten Sprint verschoben. Bei den nicht abgeschlossenen Tasks handelt es sich fast ausschließlich um optionale oder weniger umfangreiche Aufgaben.

Optional - Waiting Room - Anzeigen einer Karten Preview Dieser Task konnte nicht abgeschlossen werden, da er durch die Fertigstellung der Spielkarte blockiert wurde. Dies verzögerte aber nicht den Entwicklungsverlauf, da der Entwickler an anderen Aufgaben arbeiten konnte.

Optional - Ingame - Zoomen auf dem Spielfeld Die Task wurde im dritten Sprint begonnen, aber nicht fertig gestellt. Dies kam durch den unerwarteten größeren Aufwand beim Erstellen des Spielfeldes.

Core - Ingame - Spiel verlassen Auch diese Task konnte auf Grund des höheren Aufwands beim Erstellen des Spielfeldes noch nicht abgeschlossen werden.

Vorgänge Nicht Abgeschlossen		Im Vorgangsnavigator anzeigen			
Schlüssel	Zusammenfassung	Vorgangstyp	Priorität	Status	Story Points (19)
TG-124	Optional - Waiting Room - Anzeigen der Karte (Preview)	Story	Medium	AUFGABEN	3
TG-131	Optional - Ingame - Zoomen auf dem Spielfeld	Story	Medium	AUFGABEN	13
TG-132	Core - Ingame - Spiel verlassen	Story	Medium	AUFGABEN	3
TG-172	Test Coverage verbessern	Aufgabe	Medium	AUFGABEN	-
TG-174 *	REST Kommunikation	Aufgabe	Low	AUFGABEN	-

Abbildung 10: Nicht Abgeschlossene Vorgänge vom 3. Sprint

3.4.5 Entfernte Vorgänge

Aus dem dritten Sprint wurden insgesamt 22 Story Points entfernt und in den vierten Sprint verschoben, da wir bereits schon abschätzen konnten, dass diese nicht erreicht werden.

Core - ArmyBuilder - Laden der Armeen Bei dieser Task wurde nach der ersten Woche deutlich, dass dies nicht mehr im dritten Sprint erreicht werden kann. Der Entwickler hat anfangs Schwierigkeiten, sich in das Projekt einzuarbeiten. Die Ursache dafür war, dass der Entwickler im vorgehenden Release Product Owner war und so nicht mit der Quellcodebasis vertraut war. Dies wurde beim ersten Teammeeting erkannt. Als Reaktion darauf setzte sich der Entwickler und der Scrum Master zusammen, um den Quellcode aufzuarbeiten und so das entstandene Problem zu vermindern.

Core - ArmyBuilder - Löschen der Armeen Siehe 3.4.5 Core - ArmyBuilder - Laden der Armeen.

Optional - Waiting Room - Anzeigen der ausgewählten Einheiten Diese Task wurde erst im Verlauf des dritten Sprints erstellt. Da alle Entwickler ausgelastet waren und andere Tasks höhere Prioritäten hatten wurde dieser verschoben.

Appearance - Army Builder Im dritten Sprint war es uns wichtig Funktionalitäten bereit zu stellen. Deswegen wurden andere Tasks zuerst abgearbeitet. Das Anpassen der Oberfläche wurde auf den nächsten Sprint verlegt.

Vorgänge entfernt aus Sprint		Im Vorgangsnavigator anzeigen			
Schlüssel	Zusammenfassung	Vorgangstyp	Priorität	Status	Story Points (14 → 22)
TG-108	Core - ArmyBuilder - Laden der Armeen	Story	Highest	AUFGABEN	5
TG-118	Core - ArmyBuilder - Löschen der Armee (Clear)	Story	Medium	AUFGABEN	1
TG-129	Optional - Waiting Room - Anzeigen der Einheiten der ausgewählten Armee	Story	Medium	AUFGABEN	8
TG-179	Appearance - ArmyBuilder	Story	Medium	AUFGABEN	→ 8

Abbildung 11: Entfernte Vorgänge vom dritten Sprint

3.4.6 Fazit

Obwohl das gesetzte Ziel von 130 Story Points nicht erreicht wurde, handelt es sich dennoch um ein gutes Sprintergebnis. Es konnte ein Großteil der geforderten Features für das zweite Release bereits abgeschlossen werden. Im nächsten Sprint kann sich deshalb mehr auf optionale Features konzentriert werden. Die aufgetretenen Probleme wurden rechtzeitig erkannt und konnten so auf ein Minimum reduziert werden.

4 Sprint IV

4.1 Sprintziel

Im vierten und letzten Sprint des zweiten Releases haben wir uns drei Ziele gesetzt. Als erstes sollen natürlich die aus dem dritten Sprint übernommenen Aufgaben abgeschlossen werden. Der zweite Punkt ist das Einarbeiten der restlichen Features für das zweite Release. Obwohl ein Großteil der geforderten Features bereits implementiert war, fehlten jedoch noch Punkte wie das Visualisieren eines Spielerbeitritts im Warteraum. Zuletzt wollen wir noch die Oberfläche überarbeiten und die Nutzererfahrung der Anwendung verbessern.

4.2 User Stories

Der Aufbau der User Stories wird in Abschnitt ?? erläutert.

4.2.1 Core - ArmyBuilder - Speichern der Armeen

Ziel Der Entwickler soll das Speichern von Armeen lokal und serverseitig implementieren.

Story Albert befindet sich in dem ArmyBuilder. Albert hat eine Armee konfiguriert. Albert klickt auf den Speichern Button. Die Armeen werden lokal und auf dem Server gespeichert.

Zugeteilter Entwickler Diese User Story bearbeitet weiterhin Tobias Klipp.

Schätzung Da das Feature von den Entwicklern als komplex eingeschätzt wurde, erhielt es 15 Story Points.

Verlauf der Entwicklung Dieser Task wurde im vierten Sprint erneut geöffnet, da es noch Erweiterung gab, welche nicht zum Ende des dritten Sprints in die Quellcodebasis gezogen wurden.

4.2.2 Core - ArmyBuilder - Löschen der Armee (Clear)

Ziel Im ArmyBuilder soll ein Button vorhanden sein mit dem man eine Armee komplett leeren kann.

Story Albert befindet sich in dem ArmyBuilder. Es ist eine Armee mit Einheiten ausgewählt. Albert klickt auf den Trash-Button. Alle Einheiten der ausgewählten Armee werden aus dieser entfernt. Es wird noch keine Änderungen gespeichert.

Zugeteilter Entwickler Diese Aufgabe hat Tobias Klipp bekommen, da er das Laden und Speichern der Armee realisiert.

Schätzung Das Feature wurde auf einen Story Point geschätzt.

Verlauf der Entwicklung Dieses Feature wurde mit einem Story Point stark unterschätzt. Es wurden 6h und 45min für die Vollendung der Story benötigt. Die Anforderungen für das Löschen waren hier spezieller, da die Armee auf dem Server gelöscht werden sollte, jedoch lokal noch vorhanden sein soll. Zudem erschwerte das Erstellen von Tests den Task.

4.2.3 Appearance - ArmyBuilder

Ziel Es soll der ArmyBuilder an das Dark Theme angepasst werden.

Story Albert befindet sich in der Lobby. Albert klickt auf den ArmyBuilder Button. Albert befindet sich nun im ArmyBuilder und es sind alle nötigen Daten geladen und werden angezeigt. Der ArmyBuilder wird entsprechend dem Style der Mockups angezeigt.

Zugeteilter Entwickler Als Entwickler wurde Keanu Stückrad diese Aufgabe zugewiesen.

Schätzung Das Feature wurde auf acht Story Points geschätzt.

Verlauf der Entwicklung Der Entwickler benötigte insgesamt 14h für die Bearbeitung der Story. In der Story wurde noch der Style des „Dark Themes“ global angewandt, was zusätzliche Arbeitszeit kostete. Des Weiteren mussten Tests und Mergekonflikte behoben werden, was die Bearbeitungszeit weiter in die Länge zog.

4.2.4 Optional - Ingame - Zoomen auf dem Spielfeld

Ziel Es soll ein Vergrößern und Verkleinern der Spielkarte möglich sein.

Story Albert befindet sich in einem Spiel welches bereits gestartet ist. Es ist ein Vergrößern-Button vorhanden. Albert klickt auf den Verkleinern-Button. Das Spielfeld wird um eine Stufe herangezoomt.

Zugeteilter Entwickler Diese Aufgabe bearbeitete Keanu Stückrad, da er auch für die Implementierung des Spielfeldes verantwortlich war.

Schätzung Das Feature wurde auf 15 Story Points geschätzt, da es noch unklar war wie man das Zoomen realisiert werden kann.

Verlauf der Entwicklung Für diese Aufgabe wurden 5h Stunden benötigt. Eigentlich wurden für diese Story 15h angesetzt. Diese Fehleinschätzung resultierte daraus, dass dieses Problem für die Entwickler noch unbekannt war und somit eine hohe Einarbeitungszeit angesetzt wurde.

4.2.5 Core - Ingame - Spiel verlassen

Ziel Es soll möglich sein ein Spiel zu verlassen.

Story Albert befindet sich in einem gestartetem Spiel. Albert hat den Early Rush vermasselt. In der Ingame Scene ist ein Leave-Game-Button vorhanden. Albert klickt auf den Leave-Game-Button und verlässt das Spiel ordnungsgemäß. Albert kommt zurück Lobby zurück.

Zugeteilter Entwickler Als Entwickler wurde Keanu Stückrad dieser Aufgabe zugeteilt.

Schätzung Das Feature wurde auf drei Story Points geschätzt

Verlauf der Entwicklung Es wurde insgesamt eine Arbeitszeit von 1h für diese Story benötigt. Es wurden 3h für diese Task angesetzt. Der Entwickler konnte diese schneller abschließen, da bereits auf vorhandene Programmlogik aufgebaut werden konnte. Die Schätzung auf 3h wurde auf der Grundlage getroffen, dass nicht klar, welche Aspekte beim „Aufräumen“ zu beachten sein werden.

4.2.6 Optional - Waiting Room - Anzeigen der Karte (Preview)

Ziel Im Warteraum soll es eine Vorschau für die Karte des kommenden Spiels angezeigt werden.

Story Albert befindet sich in der Lobby. Albert tritt einem Spiel aus der Spieliste bei. Albert wird beim Spiel angemeldet und befindet sich nun im Waiting Room. Es wird eine Vorschau für das kommende Spielfeld angezeigt.

Zugeteilter Entwickler Diese Story wurde von Jan Müller bearbeitet.

Schätzung Das Feature wurde auf drei Story Points geschätzt, da es auf das Anzeigen des Spielfeldes aufbaut und sich daran orientiert werden kann.

Verlauf der Entwicklung Da diese Aufgabe auf dem Erstellen der Spielkarte aufbaut, konnten viele Konstrukte daraus übernommen werden. Deshalb wurden nur 2h von den geplanten 3h benötigt.

4.2.7 Core - Login - Default Armeen erstellen und User auf erstellen von Armeen hinweisen

Ziel Es sollen Defaultarmeen erstellt werden, damit der User sofort einem Spiel beitreten kann, wenn dieser das Spiel zum ersten mal spielt. Der Nutzer wird dabei auch darauf hingewiesen, das er die Armeen auch bearbeiten kann.

Story Albert befindet sich im Login Bereich und hat sich gerade registriert. Albert gibt seine Logindaten ein und meldet sich an. Albert wird erfolgreich eingeloggt. Albert befindet sich nun in der Lobby. Es gibt bereits Defaultarmeen und Albert wird darauf hingewiesen, dass er die Armeen Über den Armee-Button bearbeiten kann.

Zugeteilter Entwickler Diese Story wurde von Omar Sood bearbeitet.

Schätzung Das Feature wurde auf drei Story Points geschätzt.

Verlauf der Entwicklung Diese Aufgabe wurde von den Entwicklern korrekt eingeschätzt, da wir hierfür 2h und 50 min benötigten.

4.2.8 Core - Lobby - Deaktivieren der JoinGame-Buttons und des Create-Game-Buttons bei keiner ausgewählten Armee

Ziel Es soll nicht möglich sein einem Spiel beizutreten, wenn man keine valide Armee ausgewählt hat.

Story Albert befindet sich in der Lobby. Albert hat keine Armee ausgewählt es ist keine vollständige Armee vorhanden. Die JoinGame-Buttons der Spielliste und der Create-Game Button sind deaktiviert. Albert klickt auf einen der JoinGame-Buttons. Albert bekommt einen Dialog, welcher ihm anzeigt, dass er eine vollständige Armee braucht, um einem Spiel beizutreten.

Zugeteilter Entwickler Diese Story wurde von Omar Sood bearbeitet.

Schätzung Das Feature wurde auf drei Story Points geschätzt.

Verlauf der Entwicklung Der Entwickler schloss die Aufgabe eine Stunde schneller ab als geplant.

4.2.9 Optional - ArmyBuilder - Bearbeiten der Armeeinformationen

Ziel Im ArmyBuilder solle es die möglichkeit geben, das Icon und den Namen der Armee zu bearbeiten.

Story Albert befindet sich im ArmyBuilder. Unter der Armeeliste befindet sich ein Bearbeiten-Button. Albert hat eine Armee ausgewählt und klickt auf den Bearbeiten-Button. Es öffnet sich ein Dialog zum Bearbeiten der Armeeeinformationen. Es ist möglich den Namen der Armee zu bearbeiten. Es ist möglich das Icon der Armee aus einer Liste vordefinierter Icons auszuwählen. Armeen werden mit einem Default Name und einem Default Icon erstellt, falls nichts näher spezifiziert ist. Wird die auf den Speichern-Button geklickt, so wird die Armee gespeichert. Wird auf den Cancel Button geklickt, so wird die View geschlossen. Es wird nichts gespeichert.

Zugeteilter Entwickler Diese Story wurde von Omar Sood bearbeitet.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Der Aufwand wurde von den Entwicklern gut eingeschätzt, da eine Arbeitszeit von 4h und 24min benötigt wurde.

4.2.10 Optional - ArmyBuilder - Armeeauswahl aus der Lobby Übernehmen

Ziel Als Komfortfunktion soll es möglich sein die ausgewählte Armee mit in den ArmyBuilder zu Übernehmen.

Story Albert befindet sich in der Lobby. Es ist eine Armee in der Lobby ausgewählt. Albert klickt auf den Armies-Button. Albert kommt in den ArmyBuilder. Im ArmyBuilder ist die Armee ausgewählt, welche bereits in der Lobby ausgewählt war.

Zugeteilter Entwickler Diese Story wurde von Omar Sood bearbeitet.

Schätzung Das Feature wurde auf einen Story Point geschätzt.

Verlauf der Entwicklung Der Aufwand wurde von den Entwicklern gut eingeschätzt. Mit 1h und 18min liegen wir leicht über der geschätzten Stunde.

4.2.11 Optional - Create Game - Auto Join

Ziel Als Komfortfunktion soll es möglich sein, nach dem ein Spiel erstellt wurde diesem direkt beizutreten.

Story Albert befindet sich im Create Game Dialog. Albert hat eine Spielerzahl ausgewählt und einen Namen für die Armee eingegeben. Albert klickt auf Spiel erstellen. Albert tritt dem eben erstellten Spiel automatisch bei und befindet sich jetzt im Waiting Room für das Spiel.

Zugeteilter Entwickler Diese Story wurde von Omar Sood bearbeitet.

Schätzung Das Feature wurde auf 1 Story Points geschätzt.

Verlauf der Entwicklung Der Aufwand wurde von den Entwicklern perfekt eingeschätzt. Es gab eine Punktlandung mit 1h.

4.2.12 Appearance - Create Game - Anpassen des Style's des Create-Game-Formulars

Ziel Das Create Game Formular soll an den aktuellen Style angepasst werden.

Story Albert befindet sich in der Lobby. Albert klickt auf den Create-Game-Button. Es öffnet sich das CreateGame Formular. Die Buttons für Create Game erstrahlen im Glanze unseres Styles. (Mockups)

Zugeteilter Entwickler Diese Story wurde von Jan Müller bearbeitet.

Schätzung Das Feature wurde auf drei Story Points geschätzt.

Verlauf der Entwicklung Der Aufwand wurde von den Entwicklern gut eingeschätzt. Der Entwickler war mit 2h und 38min schneller fertig als geschätzt.

4.2.13 Core - Waiting Room - Ein Spieler tritt dem Spiel bei

Ziel Wenn ein Spieler das Spiel im Waitingroom beitrifft, wird dies visuell dargestellt.

Story Albert befindet sich im Waiting Room eines Spiels. Es befindet sich noch kein weiterer Mitspieler im Spiel. Carlie tritt dem Spiel bei. Carlie wird in einer der Player-Karten angezeigt.

Zugeteilter Entwickler Diese Story wurde von Keanu Stückrad bearbeitet. Keanu hat bereits im letzten Release an den Boxen für die Spieler gearbeitet.

Schätzung Das Feature wurde auf drei Story Points geschätzt.

Verlauf der Entwicklung Der Entwickler liegt mit 3,5h leicht über der Schätzung.

4.2.14 Core - Waiting Room - Ein Spieler verlässt das Spiel

Ziel Wenn ein Spieler das Spiel im Waitingroom verlässt, wird dies visuell dargestellt.

Story Albert befindet sich im Waiting Room von Alberts Spiel. Carlie befindet sich ebenfalls in Alberts Spiel. Carlie hat noch ein Date und kann deshalb nicht mitspielen. Carlie verlässt das Spiel. Carlie wird aus seiner Player-Karte entfernt. Alle anderen Mitspieler bleiben in Ihren Player-Karten.

Zugeteilter Entwickler Diese Story wurde von Keanu Stückrad bearbeitet. Keanu hat bereits im letzten Release an den Boxen der Spieler gearbeitet und den auch das Beitreten eines Spielers bearbeitet, war er die beste Wahl für diese Aufgabe.

Schätzung Das Feature wurde auf zwei Story Points geschätzt.

Verlauf der Entwicklung Da der Entwickler bereits Erfahrung mit der Problematik hatte, wurde diese Aufgabe mit 1h schneller als geschätzt abgeschlossen.

4.2.15 Optional - Waiting Room - „Player-Kasten“ wird mit der Spielerfarbe dargestellt

Ziel Spieler im Waiting Room werden mit ihren entsprechenden Farben dargestellt.

Story Albert befindet sich in der Lobby. Albert tritt Carlies Game bei. Albert befindet sich nun in Carlies Game Waiting Room. Die Player-Karten werden mit den Farben des jeweiligen Spielers dargestellt.

Zugeteilter Entwickler Diese Story wurde von Keanu Stückrad bearbeitet. Keanu hat bereits im letzten Release an den Boxen der Spieler gearbeitet und auch das Beitreten und Verlassen bearbeitet.

Schätzung Das Feature wurde auf zwei Story Points geschätzt.

Verlauf der Entwicklung Da der Entwickler bereits Erfahrung mit der Problematik hatte, wurde diese Aufgabe mit 1h schneller als geschätzt abgeschlossen.

4.2.16 Appearance - Alerts

Ziel Die Popup-Meldungen sollten dem Dark Theme angepasst werden.

Story Albert befindet sich in der Lobby. Albert klickt auf Logout. Es öffnet sich ein Overlay, welches die Alerts ersetzt. Das Overlay erstrahlt im Glanze unseres Styles. Es sollen alle Alerts durch die Overlays ersetzt werden.

Zugeteilter Entwickler Diese Story wurde von Jan Müller bearbeitet.

Schätzung Das Feature wurde auf drei Story Points geschätzt.

Verlauf der Entwicklung Für die Bearbeitung dieser Aufgabe wurden ca. 9h benötigt. Der Entwickler realisierte eine entsprechende Funktion um einfach und individuell Meldungen generieren zu können. Dies wurde durch den Scrum Master bestätigt, da wir dadurch im späteren Entwicklungsverlauf Zeit und Ressourcen sparen können.

4.2.17 Core - ArmyBuilder - „CanAttack“ Übersicht

Ziel Die Detailansicht soll in um eine Can Attack-Ansicht erweitert werden.

Story Albert befindet sich im ArmyBuilder. Albert wählt eine Einheit aus der Unit Selection aus. Es wird die Detail View zur Einheit angezeigt. In der Detail View ist eine Übersicht über die Einheiten, die von der ausgewählten Einheit angegriffen werden können, vorhanden.

Zugeteilter Entwickler Diese Story wurde von Omar Sood bearbeitet. Die Aufgabe wurde Omar zugeteilt, da dieser die Detailansicht schon in einer vorhergehen Story realisiert hat.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Da der Entwickler bereits Erfahrungen mit der Detailansicht hat konnte die Aufgabe mit 3h und 49min schneller abgeschlossen werden.

4.2.18 Optional - ArmyBuilder- Indikator ob Speichern nötig ist und Aktivieren des Save Buttons

Ziel Die Detailansicht soll in um eine Can Attack-Ansicht erweitert werden.

Story Albert befindet sich in der Lobby. Albert hat noch keine Änderungen an den Armeen vorgenommen. Der Indikator zum Speichern ist grau. Der Save-Button ist Disabled. Albert fügt eine Einheit seiner Armee hinzu. Der Indikator zum Speicher hat die Farbe von Primary Two. Der Save-Button ist Enabled.

Zugeworbener Entwickler Diese Story wurde von Omar Sood bearbeitet. Die Aufgabe wurde Omar zugeworben, da dieser schon viel im ArmyBuilder implementiert hat.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Der Entwickler war mit 2h und 42min doppelt so schnell wie geschätzt. Die Story wurde im positiven Sinne von den Entwicklern überschätzt.

4.2.19 Core - ArmyBuilder - Laden der Armeen

Ziel Die gespeicherten Armeen sollen Lokal und vom Server geladen werden.

Story Albert befindet sich im Login Bereich. Albert loggt sich erfolgreich beim Spiel ein. Die Lokal/auf dem Server gespeicherten Armeen werden geladen. Albert befindet sich in der Lobby, die geladenen Armeen werden in der Armeenliste angezeigt.

Zugeworbener Entwickler Diese Story wurde von Tobias Klipp bearbeitet. Die Aufgabe wurde Tobias zugeworben, da dieser bereits das Laden implementiert hat und somit bereits Erfahrung für diese Aufgabe besitzt.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Die Story benötigte eine Arbeitszeit von insgesamt 16h und 16min. Die Schwierigkeit bei der Implementierung war, dass die gespeicherten Armeen auf dem Server und Lokal zusammengeführt werden. Der Entwickler hatte auch Probleme damit, das Feature umzusetzen, da die Kommunikation mit dem Server noch neu war und er sich in die Thematik einarbeiten musste.

4.2.20 Optional - Chat - Anzeigen das eine Nachricht angekommen ist

Ziel Der Nutzer soll im Chat darauf aufmerksam gemacht werden, dass er eine neue Nachricht erhalten hat, in einem inaktiven Chat.

Story Albert befindet sich in der Lobby. Albert hat mit Carlie chattet und wechselt in den General-Chat. Carlie schickt eine Nachricht an Albert. Der Chat-Tab von Carlie wird bei Albert hervorgehoben, so dass Albert erkennt, dass eine neue Nachricht angekommen ist.

Zugeteilter Entwickler Die Story wurde Jan Müller zugeteilt, da diese bereits für die Implementierung des Chats zuständig war.

Schätzung Das Feature wurde auf drei Story Points geschätzt.

Verlauf der Entwicklung Der Entwickler konnte die Story mit 1h 40min schneller abschließen als erwartet.

4.2.21 Optional - ArmyBuilder - Bearbeiten der Armeeeinformationen

Ziel Der Nutzer soll im Chat darauf aufmerksam gemacht werden, dass er eine neue Nachricht erhalten hat, in einem inaktiven Chat.

Story Albert befindet sich im ArmyBuilder. Unter der Armeeliste befindet sich ein Bearbeiten-Button. Albert hat eine Armee ausgewählt und klickt auf den Bearbeiten-Button. Es öffnet sich ein Dialog zum Bearbeiten der Armeeeinformationen. Es ist möglich den Namen der Armee zu bearbeiten. Es ist möglich das Icon der Armee aus einer Liste vordefinierter Icons auszuwählen. Armeen werden mit einem Default Name und einem Default Icon erstellt, falls nichts näher spezifiziert ist. Wird die auf den Speichern Button geklickt, so wird die Armee gespeichert. Wird auf den Cancel-Button geklickt, so wird die View geschlossen. Es wird nichts gespeichert.

Zugeteilter Entwickler Die Story wurde Omar Sood zugeteilt.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Mit 4h 24m war der Entwickler schneller als geplant.

4.2.22 Core - ArmyBuilder - Autosave der Armeen bei verlassen des ArmyBuilders

Ziel Änderungen an den Armeen sollen automatisch nach verlassen des ArmyBuilders gespeichert werden.

Story Albert befindet sich im ArmyBuilder. Albert hat ein paar Armeen erstellt und bearbeitet. Albert hat seine Änderungen noch nicht gespeichert. Albert klickt auf den Zurück-Button. Es werden alle Armeen gespeichert. Albert wird in die Lobby zurück navigiert.

Zugeteilter Entwickler Die Story wurde Tobias Klipp zugeteilt.

Schätzung Das Feature wurde auf einen Story Point geschätzt.

Verlauf der Entwicklung Der Entwickler benötigte 15 Minuten, um die Aufgabe abzuschließen. Die Aufgabe wurde schneller abschlossen als erwartet.

4.2.23 Core - ArmyBuilder - Can Attack Übersicht

Ziel In der Detailansicht einer Einheit, soll es eine Can Attack Übersicht geben.

Story Albert befindet sich im ArmyBuilder. Albert hat ein paar Armeen erstellt und bearbeitet. Albert hat seine Änderungen noch nicht gespeichert. Albert klickt auf den Zurück-Button. Es werden alle Armeen gespeichert. Albert wird in die Lobby zurück navigiert.

Zugeteilter Entwickler Die Story wurde Omar Sood zugeteilt. Omar hatte bereits die Detailansicht für Einheit implementiert, womit er die meiste Erfahrung in diesen Bereich hat.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Der Arbeitsaufwand für diese Story belief sich auf 3h und 49m. Damit war der Entwickler schneller als geplant.

4.2.24 Optional - ArmyBuilder- Indikator ob Speichern nötig ist und Aktivieren des Save Buttons

Ziel Im ArmyBuilder soll es ein visuelles Feedback geben, ob aktuellen Armeen gespeichert sind.

Story Albert befindet sich in der Lobby. Albert hat noch keine Änderungen an den Armeen vorgenommen. Der Indikator zum Speichern ist Grau. Der Save-Button ist Disabled. Albert fügt eine Einheit seiner Armee hinzu. Der Indikator zum Speicher hat die Farbe von Primary Two. Der Save-Button ist Enabled.

Zugeteilter Entwickler Die Story wurde Omar Sood zugeteilt.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Die Story wurde mit 2h und 42 in der Hälfte der geplanten Zeit erledigt.

4.2.25 Optional - Waiting Room - Anzeigen der Einheiten der ausgewählten Armee

Story Albert befindet sich in der Lobby und hat eine Armee ausgewählt. Albert tritt einem Spiel bei. Albert wird erfolgreich beim Spiel angemeldet und befindet sich im Waiting Room des Spiels. Entsprechend dem Mockup werden die Einheiten der ausgewählten Armee angezeigt.

Schätzung Das Feature wurde auf acht Story Points geschätzt.

Verlauf der Entwicklung Die Story wurde nicht im vierten Sprint begonnen und wird auch nicht in den nächsten Sprint übernommen werden.

4.2.26 Optional - Lobby - Anzeigen der Einheiten einer Armee

Story Albert befindet sich im ArmyBuilder. Es sind Armeen vorhanden. Albert hovers über eine Armee. Es erscheint ein Overlay, in welchem die Einheiten der Armeen sehen kann.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Die Story wurde nicht im vierten Sprint begonnen und wird auch nicht in den nächsten Sprint übernommen werden.

4.2.27 Optional - ArmyBuilder- Notification für das Speichern der Armee und Waiting Animation.

Story Albert befindet sich in der Lobby. Albert hat eine Armee bearbeitet. Albert klickt auf den Speicher Button. Der Speicherprozess beginnt. Es wird eine Warteanimation angezeigt. Der Speicherprozess endet. Es wird eine Notification angezeigt mit dem Status des Speicherns.

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Da der Speicherprozess nur sehr wenig Zeit benötigt, ist das Anzeigen eines Ladebalkens nicht nötig. Die Story wird deshalb nicht umgesetzt.

4.2.28 Core - ArmyBuilder - Lokaler Speicherort für Armeen

Ziel Der Speicherort der Armeen soll lokal an die Betriebssysteme Windows, Unix und MacOS angepasst werden.

Story Albert befindet sich im ArmyBuilder. Albert klickt auf Speichern. Die Armeen werden lokal gespeichert an einem von uns definierten betriebssystemabhängigen Speicherort. z.B. Windows: AppData

Schätzung Das Feature wurde auf fünf Story Points geschätzt.

Verlauf der Entwicklung Das Feature konnte in 30 Minuten fertig gestellt werden. Die Story wurde erst im fünften Sprint vom Entwickler geschlossen. Die Funktionalität war jedoch bereits im zweiten Release vorhanden.

4.3 Tasks

Auch im vierten Sprint wurden einzelne Tasks erstellt, die keiner User Story zugeordnet waren. Da keine Story Punkte auf Tasks geschätzt werden konnten, wurden lediglich Zeitschätzungen eingetragen.

4.3.1 Task - SceneManger Überarbeiten

Ziel Der Szenenwechsel sollte soweit überarbeitet werden, dass bestimmte Szenenwechsel, z. B. vom der Lobby zum ArmyBuilder, möglich sind, ohne dass die Controller terminiert werden und damit die Websocketverbindung zum Server geschlossen wird.

Zugeteilter Entwickler Jan Müller bearbeitete diesen Task.

Schätzung Dieser Task wurde auf acht Stunden geschätzt.

Verlauf der Entwicklung Da der Entwickler auf der Arbeit in einer anderen User Story aufbauen konnte, benötigte er mit 1h und 35 Minuten deutlich weniger Zeit.

4.3.2 Task - Neues Musikstück einpflegen

Ziel Das bisher verwendete Musikstück sollte durch ein anderes 8-Bit Musikstück ersetzt werden.

Zugeteilter Entwickler Dieser Task wurde von Keanu Stückrad bearbeitet.

Schätzung Für die Implementierung wurde eine Stunde geschätzt.

Verlauf der Entwicklung Der Task konnte nach einer Stunde vom Entwickler abgeschlossen werden.

4.3.3 Task - Spiel verlassen

Ziel Das Verlassen des Spiel sollte dahingehend verändert werden, dass beim Verlassen der Spielwebsocket nicht mehr vom Client geschlossen wird. Stattdessen soll unsere Applikation nach der leaveGame-Nachricht an den Server, auf das Schließen des Websockets durch den Server warten.

Zugeteilter Entwickler Jan Müller bearbeitete diesen Task.

Schätzung Es wurden fünf Stunden für diesen Task geschätzt.

Verlauf der Entwicklung Mit einer Stunde wurde der Task deutlicher schneller als geschätzt abgeschlossen.

4.3.4 Task - NullPointerException - Spielbeitritt

Ziel Wenn bei einem Spielbeitritt der Server mit einer NullPointerException reagiert, soll diese in unserer Applikation aufgefangen werden. Es soll ein Szenenwechsel in die Lobby erfolgen und dem Nutzer soll eine Fehlermeldung angezeigt werden.

Zugeteilter Entwickler Dieser Task wurde Jan Müller zugewiesen.

Schätzung Der Task wurde auf drei Stunden geschätzt.

Verlauf der Entwicklung Der Entwickler benötigte eine Stunde um diesen Taks abzuschließen.

4.3.5 Task - Reaktion auf 503 vom Server

Ziel Wenn der Spieler aufgrund von Inaktivität mit einer 503 Response ausgeloggt wird, soll der Nutzer wieder zum Login geleitet werden und eine Fehlermeldung soll angezeigt werden.

Zugeteilter Entwickler Jan Müller bearbeitete diesen Task.

Schätzung Die Schätzung für diesen Task betrug eine Stunde.

Verlauf der Entwicklung Tatsächlich benötigte der Entwickler eine Stunde und 30 Minuten für diesen Task.

4.3.6 Task - ArmyManager Lore

Ziel Die Einheiten im ArmyBuilder sollten neue Bilder sowie Beschreibungstexte erhalten, die in einem Fantasyuniversum angesiedelt sind.

Zugeteilter Entwickler Dieser Task wurde Omar Sood zugeordnet.

Verlauf der Entwicklung Für diesen Task benötigte der Entwickler 10 Stunden und 40 Minuten.

4.3.7 Task - Zwischenrelease v0.3

Ziel Zur Vorbereitung der Abgabe auf dem master Branche, sollte ein eigener Zwischenrelease Branche erstellt. Dieser dient dazu, die Integration aller entwickelten Features sicherzustellen. Von diesem Branche soll ein Pull Request an den master Branche erstellt werden.

Zugeteilter Entwickler Jan Müller bearbeite diesen Task.

Schätzung Es wurden zwei Stunden für diesen Task geschätzt.

Verlauf der Entwicklung Dieser Task wurde nach 15 Minuten abgeschlossen, da die Integration der Features funktionierte.

4.3.8 Task - Speicherstände mergen

Ziel Beim Laden der Armeen sollen die lokalen und serverseitig gespeicherten Armeen sinnvoll gemerged werden. Wenn zwei Armeen die gleiche ID haben, werden die Units und der Name der lokalen Armeen durch die remote Armeen überschrieben. Das Icon der lokal gespeicherten Armeen soll übernommen werden. Lokale Armeen ohne IDs sollen ebenfalls übernommen werden.

Zugeteilter Entwickler Dieser Task wurde Tobias Klipp zugewiesen.

Schätzung Für diesen Task wurden fünf Stunden geschätzt.

Verlauf der Entwicklung Der Entwickler konnte auf die Arbeit einer anderen User Story aufbauen und benötigte daher nur 30 Minuten für diesen Task.

4.4 Bugs

In einem Fall wurde nach der Implementierung einer Funktionalität festgestellt, dass diese nicht korrekt funktioniert. Dafür wurde in Jira ein Bug-Task angelegt.

4.4.1 Bug - lokales Speichern unter Windows

Ziel Unter Windows schlug das lokale Speichern von Armeen schlug fehl und sollte behoben werden.

Zugewiesener Entwickler Der Bug wurde Omar Sood zugewiesen.

Verlauf der Entwicklung Der Bug wurde nach einer Stunde behoben.

4.5 Zeitübersicht

User Story	Story Points	Soll Zeit	Ist Zeit	Entwickler
Core - ArmyBuilder - Laden der Armeen	5	5h	16h 16min	Tobias Klipp
Core - ArmyBuilder - Speichern der Armeen	13	13h	34h 45min	Tobias Klipp
Core - ArmyBuilder - Löschen der Armee (Clear)	1	1h	6h 45m	Tobias Klipp
Optional - Waiting Room - Anzeigen der Karte (Preview)	3	3h	2h 2m	Jan Müller
Optional - Ingame - Zoomen auf dem Spielfeld	13	13h	7h	Keanu Stückrad
Core - Ingame - Spiel verlassen	3	3h	1h	Keanu Stückrad
Appearance - ArmyBuilder	8	8h	14h	Keanu Stückrad
Core - Login - Default Armeen erstellen und User auf erstellen von Armeen hinweisen	3	3h	4h 8m	Omar Sood
Core - Lobby - Deaktivieren der JoinGame-Buttons und des Create-Game-Buttons bei keiner ausgewählten Armee	3	3h	2h	Omar Sood
Optional - Chat - Anzeigen das eine Nachricht angekommen ist	3	3h	1h 40m	Omar Sood
Optional - ArmyBuilder - Bearbeiten der Armeeeinformationen	5	5h	4h 24m	Omar Sood
Optional - ArmyBuilder - Armeerauswahl aus der Lobby Übernehmen	1	1h	1h 18m	Omar Sood
Optional - Create Game - Auto Join	1	1h	1h 16m	Omar Sood
Appearance - Create Game - Anpassen des Style's des Create-Game-Formulars	3	3h	2h 38m	Jan Müller
Core - Waiting Room - Ein Spieler tritt dem Spiel bei	3	3h	3h 30m	Keanu Stückrad
Core - Waiting Room - Ein Spieler verlässt das Spiel	2	2h	1h	Keanu Stückrad
Optional - Waiting Room - „Player-Kasten“ wird mit der Spielerfarbe dargestellt	2	2h	1h	Keanu Stückrad
Core - ArmyBuilder - Autosave der Armeen bei verlassen des ArmyBuilders	1	1h	15m	Tobias Klipp

User Story	Story Points	Soll Zeit	Ist Zeit	Entwickler
Appearance - Alerts	3	3h	8h 55m	Omar Sood
Core - ArmyBuilder - „CanAttack“ Übersicht	5	5h	3h 49m	Omar Sood
Appearance - Alerts	3	3h	8h 55m	Jan Müller
Optional - ArmyBuilder- Indikator ob Speichern nötig ist und Aktivieren des Save Buttons	5	5h	2h 42m	Omar Sood

Tabelle 2: Übersicht der Zeiten für den 4. Sprint

4.6 Analyse des 4. Sprint

Der vierte Sprint wurde am 07.07.2019 abgeschlossen. Es wurden insgesamt 86 von 109 Story Points abgeschlossen. Es wurden alle Kernfunktionalitäten und viele optionale Features umgesetzt. Im Folgenden wird noch einmal der Verlauf des Sprints analysiert und eine Übersicht über die nicht abgeschlossenen User Stories gegeben.

4.6.1 Burndown-Diagramm

In Abb. 12 ist das Burndown-Diagramm für den vierten Sprint zusehen. Insgesamt ist der Verlauf des Burndown-Graphen so wie vorgesehen. Es gibt jedoch zwei Ausläufer. Der erste findet am Anfang des Sprints statt, wo am 25.06.2019 die Anzahl der Story Points ansteigt. Die Ursache dafür ist, dass der Scrum Master eine abgeschlossene Story erneut geöffnet hatte, damit ein Entwickler seine Zeiten nachtragen kann. Die Ursache ist als lediglich ein organisatorischer Fehler. Der letzte Ausreißer ist das Ende des Sprints, da die Kurve nicht auf Null Story Points fällt. Es waren also noch User Stories nicht abgeschlossen. Das hatte zwei Gründe. Zum Einen hatte die Entwickler bereits Ihre geforderte Arbeitszeit absolviert und es handelte sich nur noch um optionale Tasks, weshalb deren Bearbeitung nicht als zwingend notwendig angesehen wurde. Zum anderen haben sich der Product Owner und der Scrum Master bei den Aufgaben für den vierten Sprint verschätzt und deshalb zu viele Aufgaben generiert.

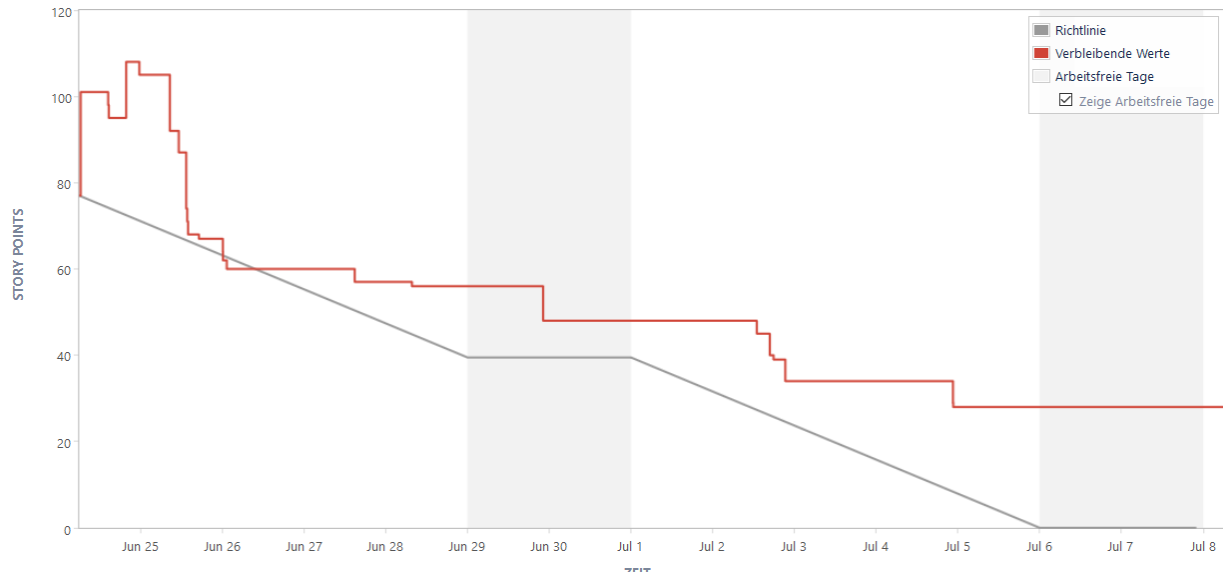


Abbildung 12: Burndown-Diagramm des Sprint 4

4.6.2 Abgeschlossene Vorgänge

In Abb. 13 sind die abgeschlossenen Vorgänge für den vierten Sprint zu sehen. Es konnten alle geforderten Features für das zweite Release umgesetzt werden. Es konnten noch folgende optionale Features umgesetzt werden:

- eine Kartenvorschau im Warteraum
- Zoomen auf dem Spielfeld
- Benachrichtigung für ungelesene Nachrichten
- Bearbeiten von Armeeeinformationen
- Im Warteraum werden die Spieler mit entsprechender Farbe dargestellt
- ein Indikator für das Speichern im Armeemanager





































































Abgeschlossene Vorgänge		Im Vorgangsnavigator anzeigen			
Schlüssel	Zusammenfassung	Vorgangstyp	Priorität	Status	Story Points (86)
TG-108	Core - ArmyBuilder - Laden der Armeen	 Story	 Highest	 FERTIG	5
TG-109 *	Core - ArmyBuilder - Speichern der Armeen	 Story	 Highest	 FERTIG	13
TG-118	Core - ArmyBuilder - Löschen der Armee (Clear)	 Story	 Medium	 FERTIG	1
TG-124 *	Optional - Waiting Room - Anzeigen der Karte (Preview)	 Story	 Medium	 FERTIG	3
TG-131 *	Optional - Ingame - Zoomen auf dem Spielfeld	 Story	 Medium	 FERTIG	13
TG-132 *	Core - Ingame - Spiel verlassen	 Story	 Medium	 FERTIG	3
TG-174 *	REST Kommunikation	 Aufgabe	 Low	 FERTIG	-
TG-179	Appearance - ArmyBuilder	 Story	 Medium	 FERTIG	8
TG-200	Core - Login - Default Armeen erstellen und User auf erstellen von Armeen hinweisen	 Story	 Medium	 FERTIG	3
TG-202	Core - Lobby - Deaktivieren der JoinGame-Buttons und des Create-Game-Buttons bei keiner ausgewählten Armee	 Story	 Medium	 FERTIG	3
TG-203	Optional - Chat - Anzeigen das eine Nachricht angekommen ist	 Story	 Medium	 FERTIG	3
TG-206	Optional - ArmyBuilder - Bearbeiten der Armeinformationen	 Story	 Medium	 FERTIG	5
TG-207	Optional - ArmyBuilder - Armeeauswahl aus der Lobby übernehmen	 Story	 Medium	 FERTIG	1
TG-208	SceneManager überarbeiten	 Aufgabe	 Medium	 FERTIG	-
TG-214	Optional - Create Game - Auto Join	 Story	 Medium	 FERTIG	1
TG-215	Appearance - Create Game - Anpassen des Style's des Create-Game-Formulars	 Story	 Medium	 FERTIG	3
TG-216	Core - Waiting Room - Ein Spieler tritt dem Spiel bei	 Story	 Medium	 FERTIG	3
TG-217	Core - Waiting Room - Ein Spieler verlässt das Spiel	 Story	 Medium	 FERTIG	2
TG-218	Optional - Waiting Room - "Player-Kasten" wird mit der Spielerfarbe dargestellt	 Story	 Medium	 FERTIG	2
TG-220	Core - ArmyBuilder - Autosave der Armeen bei verlassen des ArmyBuilders	 Story	 Medium	 FERTIG	1
TG-221	Appearance - Alerts	 Story	 Medium	 FERTIG	3
TG-222	Core - ArmyBuilder - "CanAttack" Übersicht	 Story	 Medium	 FERTIG	5
TG-223	Optional - ArmyBuilder- Indikator ob Speichern nötig ist und Aktivieren des Save Buttons	 Story	 Medium	 FERTIG	5
TG-251	Speicherstände mergen	 Aufgabe	 Medium	 FERTIG	-
TG-252 *	Zwischenrelease v0.3	 Aufgabe	 Medium	 FERTIG	-
TG-253 *	NullPointerException - Spielbeitritt	 Aufgabe	 Medium	 FERTIG	-
TG-254 *	Spiel verlassen	 Aufgabe	 Medium	 FERTIG	-
TG-279 *	Reaktion auf 503 vom Server	 Aufgabe	 Medium	 FERTIG	-
TG-280 *	lokales Speichern unter Windows	 Bug	 Medium	 FERTIG	-
TG-290 *	ArmyManager Lore	 Aufgabe	 Medium	 FERTIG	-
TG-291 *	Neues Musikstück einpflegen	 Aufgabe	 Medium	 FERTIG	-

Abbildung 13: Abgeschlossene Vorgänge in Sprint 4

4.6.3 Nicht abgeschlossene Vorgänge

Im vierten Sprint wurden insgesamt 23 Story Points nicht umgesetzt. Unter diesen befindet sich kein Kernfeature. Die nicht abgeschlossenen User Stories werden nun im Nachfolgenden betrachtet. Eine Übersicht über diese ist in Abb. 14 zu finden.

Optional - Waiting Room - Anzeigen der Einheiten der ausgewählten Armee In dieser Story sollten die Einheiten der ausgewählten Armee im Warteraum angezeigt werden. Es stellte sich jedoch heraus, dass dies im nächsten Sprint wieder verworfen werden müsste, weshalb andere Aufgaben priorisiert wurden, um Arbeitszeit nicht in unnötige Features zu stecken. Diese Story wird im fünften Sprint nicht weitergeführt.

Optional - Lobby- Anzeigen der Einheiten einer Armee Ziel dieser Story war es die Einheiten der Armeen in der Lobby anzuzeigen. Auch diese Story wurde verworfen. Die Gründe dafür waren die selben wie in 4.6.3. Die Story wird im fünften Sprint nicht weitergeführt.

Optional - ArmyBuilder - Notification für das Speichern der Armee und Waiting Animation Im Armeemanager sollte während des Speichervorgangs ein Fortschrittsbalken angezeigt werden und eine Benachrichtigung für das erfolgreiche Speichern erstellt werden. Da der Speichervorgang jedoch immer sehr schnell abgeschlossen wurde, ist das Anzeigen der eben genannten Visualisierungen verworfen wurden.

Core - ArmyBuilder - Lokaler Speicherort für Armeen Dieses Feature ist bereits im zweiten Release enthalten. Es wird noch unter den nicht abgeschlossenen Vorgängen aufgeführt, weil die Vorgang im Jira nicht rechtzeitig abgeschlossen wurde.

Vorgänge Nicht Abgeschlossen		Im Vorgangsnavigator anzeigen			
Schlüssel	Zusammenfassung	Vorgangstyp	Priorität	Status	Story Points (23)
TG-129	Optional - Waiting Room - Anzeigen der Einheiten der ausgewählten Armee	Story	Medium	AUFGABEN	8
TG-172 *	Test Coverage verbessern	Aufgabe	Medium	AUFGABEN	-
TG-213	Optional - Lobby - Anzeigen der Einheiten einer Armee	Story	Medium	AUFGABEN	5
TG-224	Optional - ArmyBuilder- Notifiaktion für das Speichern der Armee und Waiting Animation.	Story	Medium	AUFGABEN	5
TG-225	Core - ArmyBuilder - Lokaler Speicherort für Armeen	Story	Medium	AUFGABEN	5

Abbildung 14: Nicht Abgeschlossene Vorgänge in Sprint 4

4.6.4 Fazit

Wir sehen den vierten Sprint als erfolgreich an, da alle restlichen Kernfunktionalitäten und viele weitere optionale Features abgeschlossen werden konnten. Das von uns gesetzte Sprintziel konnte somit erreicht werden. Von den 23 Über gebliebenen Story Points wurden 18 verworfen, der Product Owner und Scrum Master haben bei diesen Übersehen sie aus dem Jira zu entfernen. Die übrigen fünf Story Points wurden im Jira nur nicht diesem Sprint zugeordnet, da der Entwickler die Aufgabe zu spät abgeschlossen hat. Somit bleiben am Ende auch keine Aufgaben Über, womit die Schätzung für den 4. Sprint eingehalten werden konnte.

Code Coverage Abschließend sei noch zu erwähnen, dass auch diesen Sprint die Quellcodeabdeckung weiter ausgebaut werden konnte. In Abb. 15 ist eine Übersicht Über die finale Code Coverage zum Ende von Release 2. Wir erreichen eine Abdeckung von insgesamt 76%, womit die Anforderung von 60% CO Coverage erfüllt ist.

Overall Coverage Summary			
Package	Class, %	Method, %	Line, %
all classes	97,4% (151 / 155)	80,4% (686 / 853)	76,9% (3035 / 3945)

Abbildung 15: Quellcodeabdeckung Release 2

5 Abschluss des 2. Releases

Das zweite Release wurde am 07.07.2019 abgeschlossen. Im nachfolgenden Teil werden nun die Mockups mit der Implementierung verglichen. Dabei wir auf Unterschiede und nicht realisierte Features eingegangen und die Ursachen erläutert.

5.1 Der Armeemanager

Zu Beginn des Release wurde folgende Anforderungen an den Armeemanager gestellt:

- Erstellen und Konfigurieren von Armeen
- Armeen dürfen maximal 10 Einheiten enthalten
- Speichern soll Lokal und auf dem Server erfolgen
- Speichern von mindestens 3 Armeen

Diese Anforderungen wurden zum Ende des zweiten Releases abgedeckt. Die Armeen werden automatisch erstellt und es werden dem Nutzer 7 zu Konfiguration zur Verfügung gestellt. Verändert werden können die Armeen Über die „+“ und „-“ Buttons. Gespeichert werden kann Über den Speichern Button. Die 7 Armeen werden dabei lokal und auf dem Server gespeichert. Die Umsetzung war dabei in 3 Kernbereiche aufgeteilt. Diese werden in Abb. 16 gezeigt.

- Der Einheitenauswahl mit Detailansicht
- Der Armeeübersicht
- Der Armeeauswahl

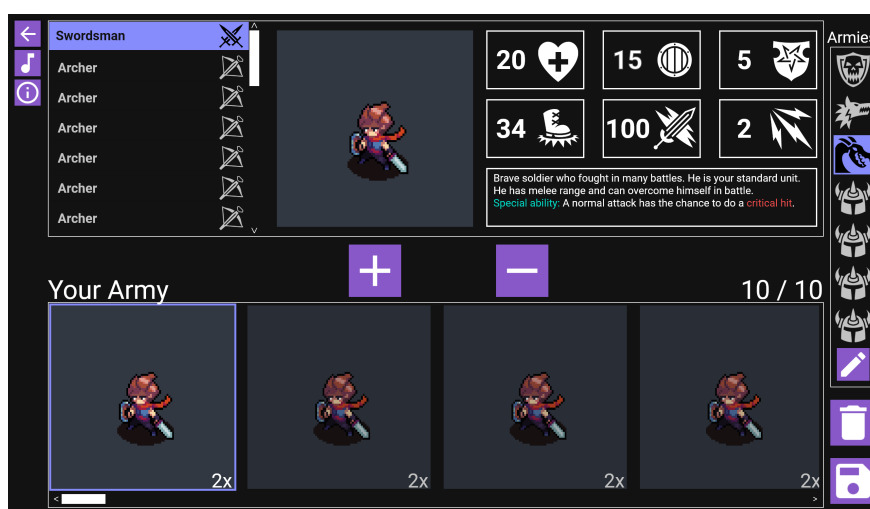


Abbildung 16: Mockup für den geplanten Armeemanager

5.1.1 Implementaion

Die Umsetzung dieser Bereiche hat dabei fast 1:1 statt gefunden. Auf Abb. 17 ist die abschließende Implementation des Armeemangers zu sehen. Die Kernbereiche wurden wie geplant umgesetzt. Im Design und der Anordnung sind dabei leichte Unterschiede zu erkennen. Die Rahmen um die Einheitenauswahl und der Armeeauswahl sind entfallen, da dies zu einer Übersichtlicheren Oberfläche beigetragen hat. Der Edit-Button zum Ändern der Armeeeinformationen wurde aus der Armeeauswahl herausgezogen, um ihn in eine Reihe mit den anderen Buttons zu bringen. Der Detailansicht der Einheiten wurde eine Can Attack-Ansicht hinzugefügt, da sich durch die erhaltenen Servernachrichten eine solche Übersicht als sinnvoll erwies. Es wurden noch die folgenden zusätzlichen Features implementiert:

- Lore und Flavour für alle Einheiten
- Individuelle Armeeicons und Namen
- Clear Button zum schnellen „aufräumen“ von Armeen
- Indikator für das Speichern von Armeen

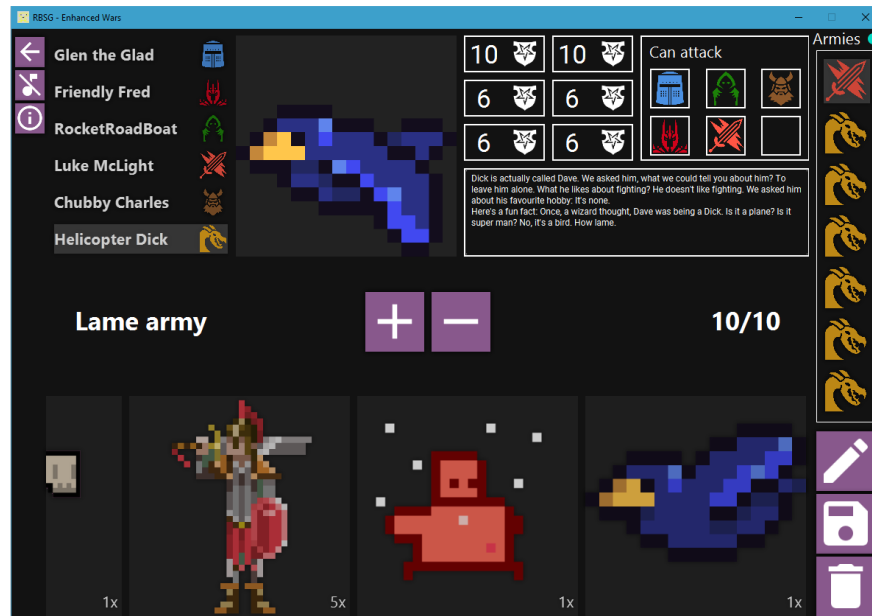


Abbildung 17: Realisierung des Armeemanagers am Ende von Release 2

5.1.2 Armee-Editor

Im Kundengespräch zu Beginn des 2. Releases wurde vom Kunden noch die Anforderung gestellt die Icons der Armee individuell gestalten zu können. Dafür wurde das Mockup in Abb. 18 erstellt. Die Umsetzung ist in Abb. 19 zu sehen. Die Implementierung unterscheidet sich vom Mockup nur in den Farben und den Schließen-Button. Der Schließen-Button wurde im Mockup vergessen und wurde in der Realisierung nachgetragen. Der farbliche Unterschied ist der zeitlichen Begrenzung geschuldet. Die Funktionalitäten wurden vom Entwickler fertig gestellt, so ist es möglich den Armeenamen zu bearbeiten und auch das Icon individuell aus einer vorgefertigten Liste auszuwählen. für den Design Aspekt fehlte am Ende des Releases die Zeit.

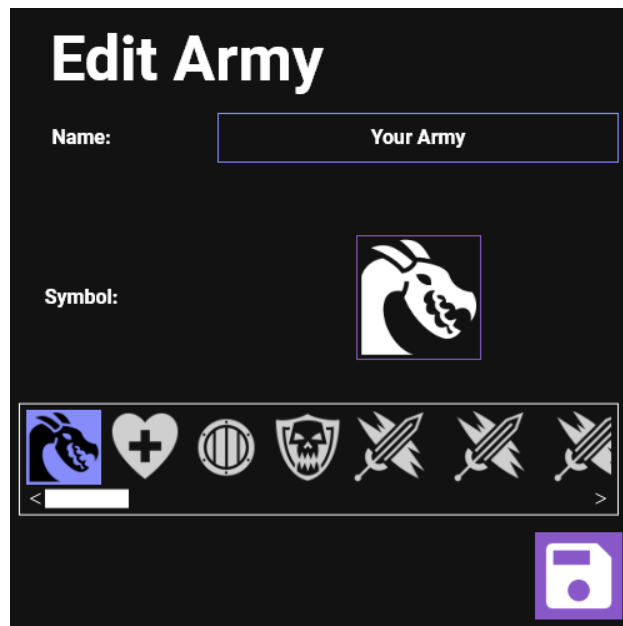


Abbildung 18: Mockup des Armeeditor

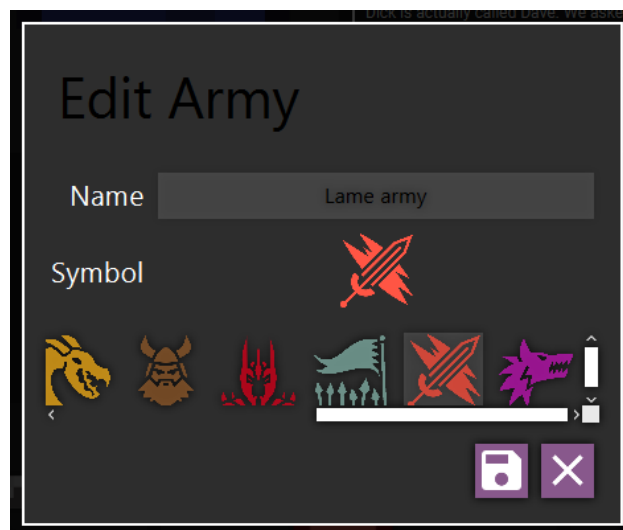


Abbildung 19: Implementierung des Armeeditor

5.1.3 Info zu den Attributen

Als zusätzliches Feature wurde noch die eine Übersicht zu den Attributen erstellt, welche die Einheit besitzt. Die umgesetzte Oberfläche ist in Abb. 20 zu sehen. Geplant war die in Abb. 21 zu sehende Oberfläche. Diese wurde so umgesetzt.

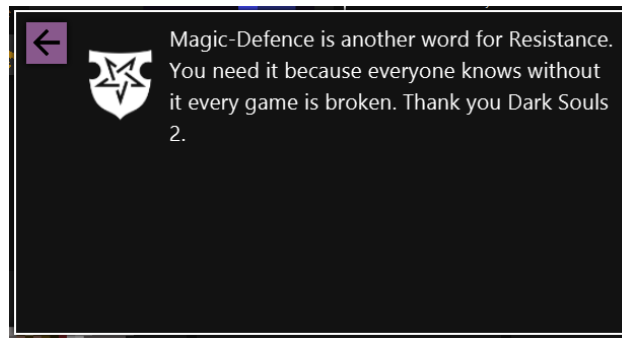


Abbildung 20: Übersicht für die Attribute der Einheiten



Abbildung 21: Mockup Übersicht Über die Attribute

5.2 Warteraum

Zu Beginn des zweiten Releases wurde ein Warteraum geplant, in welchen man nach Spielbeginn kommt. Das Mockup ist in Abb. 22 zu sehen. Es wurden fast alle geplanten Features im Warteraum umgesetzt, die Implementierung ist in Abb. 23 zu sehen. Den Warteraum haben wir dabei in 3 Komponenten unterteilt:

- das Anzeigen der Spieler
- den Chat
- die Kartenvorschau

Diese Komponenten wurden auch wie geplant umgesetzt. Als Zusatz wird noch in den Spielerkarten die Farbe angezeigt, welcher ein Spieler für das kommende Spiel hat. Am rechten Rand sollte noch eine die Einheitenliste für die ausgewählte Armee angezeigt werden. Dieses Feature wurde nicht mehr umgesetzt, da es im nächsten Release wieder verworfen worden wäre und wir uns so auf andere Sachen konzentriert haben.



Abbildung 22: Mockup des Warteraum

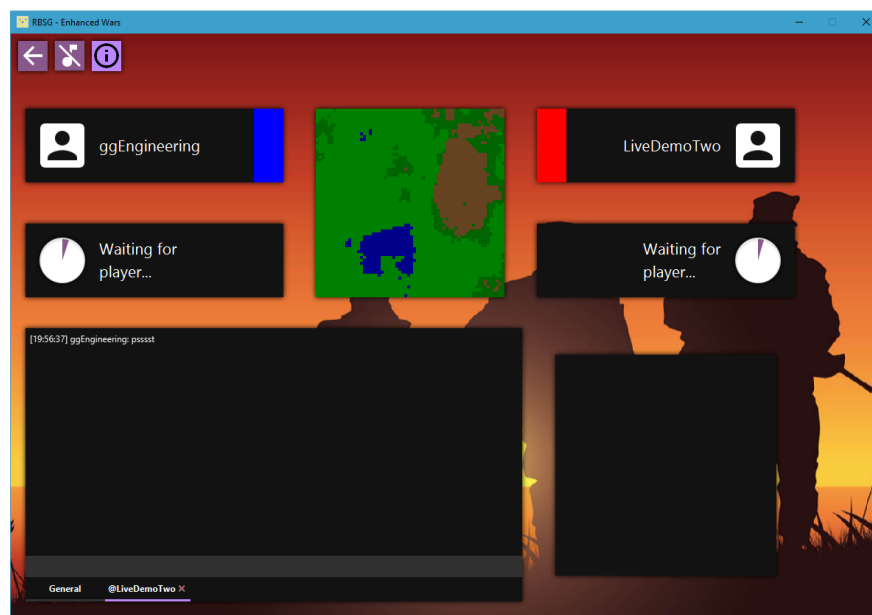


Abbildung 23: Umsetzung des Warteraums im 2. Release

Damit sind auch folgende Anforderungen umgesetzt:

- Initiales Spielgeschehen anzeigen (indirekt über die Kartenvorschau)
- Spieler, die ein Spiel beitreten
- Von Spiel zurück in die Lobby gehen
- Ingame Chat

5.3 Ingame

Als das 2. Release geplant wurde, war den Product Owner nicht bewusst, dass auch die Implementierung einer Karte zu diesem Release dazu gehört. Deshalb wurde kein Mockup

für diese Situation angefertigt. Trotzdem war es uns möglich eine Ingame Oberfläche zu erstellen. Diese ist in Abb. 24 zu sehen. Umgesetzt wurde hier:

- das Zeichnen der Karte
- das Zeichnen der Startpositionen der Einheiten
- Zoomen auf dem Spielfeld
- Verlassen des Ingames



Abbildung 24: Umsetzung der Ingame Oberfläche im 2. Release

Durch die Implementierung der Ingame Oberfläche wird die Anforderung das initiale Spielgeschehen anzuzeigen erfüllt. Es sollte auch möglich sein, vom Spiel zurück in die Lobby zu gehen. Dies ist über den zurück-Button in der linken oberen Ecke möglich. Das Zoomen wird über die zwei nebenliegenden Buttons realisiert. Bei einem Zwei-Spieler-Spiel ist ein 3-Stufiger Zoom möglich und bei einem Vier-Spieler-Spiel ein 4-Stufen Zoom. Die Einheiten werden auf ihren Startpositionen auf der Karte dargestellt. Eine Besonderheit bei der Darstellung der Karte ist, dass die Übergänge zwischen Biomen flüssig ist. So werden bei Übergängen andere Graphiken gewählt, um diese flüssig darzustellen.

5.4 Lobby

Die Lobby war von den Anforderungen dieses Releases nicht betroffen. Um jedoch ein angenehmeres Bedienverhalten zu gewährleisten, haben wir der Lobby noch eine Armeerauswahl hinzugefügt, da eine Armee benötigt wird um einem Spiel beizutreten. Zu sehen ist die Umsetzung auf Abb. 25. Die Icons der Armee werden dabei je nach Konfiguration, individuell dargestellt. Die Oberfläche ist auch Responsive gehalten, so wird der ausgewählte Eintrag hervorgehoben. Ist keine Armee ausgewählt, so wird der Spiel erstellen-Button und auch die Spiel beitreten-Buttons deaktiviert.

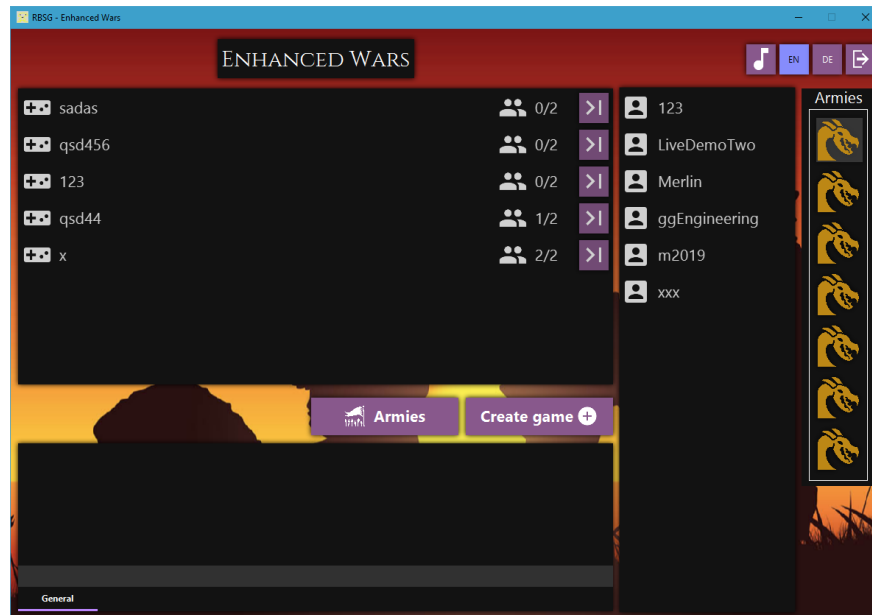


Abbildung 25: Lobby mit Armeerauswahl am Ende des 2. Releases

5.4.1 Spiel erstellen Formular

In diesen Release wurde auch das Spiel erstellen Formular optisch aufgearbeitet, dies kann man in Abb. 26 sehen. Es ist nun an das Dark Theme angepasst wurden.

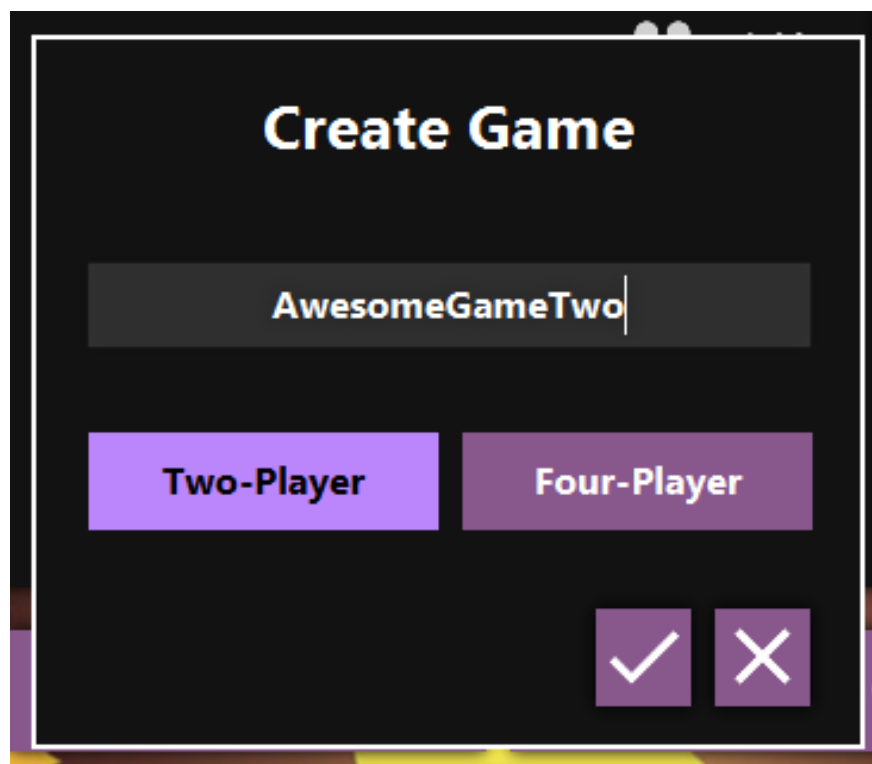


Abbildung 26: Optisch Überarbeitetes „Spiel erstellen“ Formular

5.5 Alerts

Im Verlauf des zweiten Releases entschieden wir uns dazu die Alerts optische an unsere Thema anzupassen. Anfangs wurden die von JavaFX generierten Alerts genutzt, diese sind in Abb. 27 zu sehen. Diese wurden dann von uns durch die in Abb. 28 zu sehenden Alerts ersetzt. Diese werden nun mit dem von uns erstellten Dark Theme dargestellt.

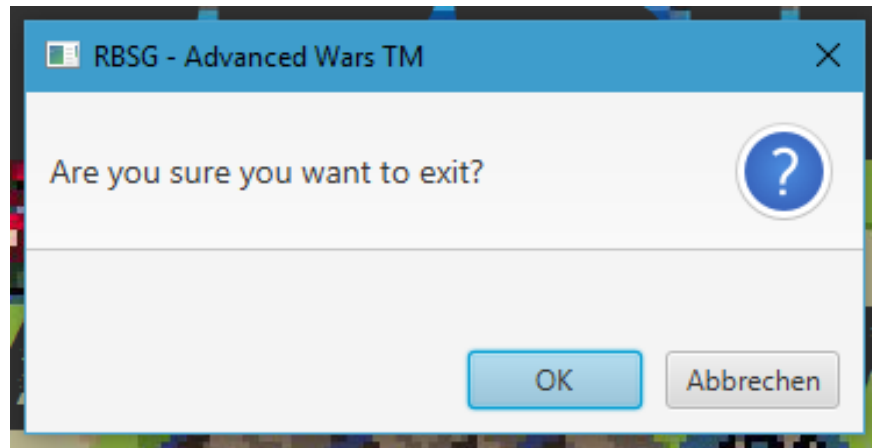


Abbildung 27: Alerts von JavaFX

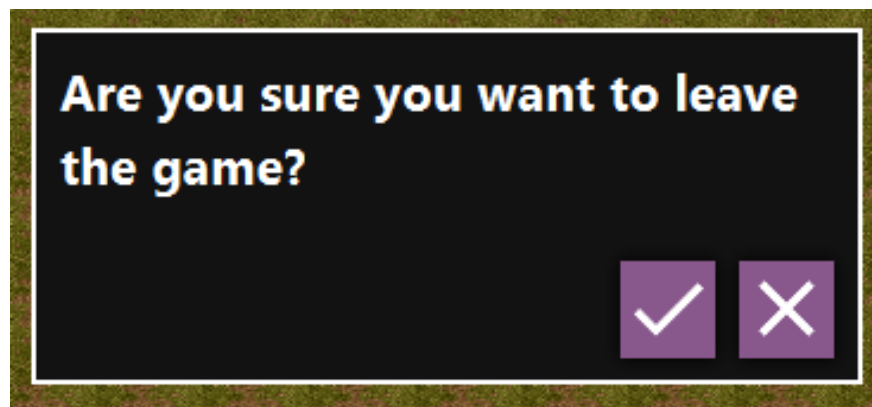


Abbildung 28: Überarbeitet Alerts im Dark Theme

Abbildungsverzeichnis

1	Login Release I	3
2	Lobby Release I	3
3	Spiel erstellen Release I	4
4	Warteraum eines Spiels Release I	5
5	Button für die Internationalisierung Release I	5
6	Button für Musik Release I	6
7	Burndown-Diagramm des Sprint 3	7
8	Code Coverage vom 3. Sprint	8
9	Abgeschlossene Vorgänge vom 3. Sprint	8
10	Nicht Abgeschlossene Vorgänge vom 3. Sprint	9
11	Entfernte Vorgänge vom dritten Sprint	10
12	Burndown-Diagramm des Sprint 4	26
13	Abgeschlossene Vorgänge in Sprint 4	27
14	Nicht Abgeschlossene Vorgänge in Sprint 4	28
15	Quellcodeabdeckung Release 2	28
16	Mockup für den geplanten Armeemanager	29
17	Realisierung des Armeemanagers am Ende von Release 2	30
18	Mockup des Armee-Editor	31
19	Implementierung des Armee-Editor	31
20	Übersicht für die Attribute der Einheiten	32
21	Mockup Übersicht Über die Attribute	32
22	Mockup des Warteraum	33
23	Umsetzung des Warteraums im 2. Release	33
24	Umsetzung der Ingame Oberfläche im 2. Release	34
25	Lobby mit Armeeauswahl am Ende des 2. Releases	35
26	Optisch Überarbeitetes „Spiel erstellen“ Formular	35
27	Alerts von JavaFX	36
28	Überarbeitet Alerts im Dark Theme	36

Tabellenverzeichnis

1	Übersicht der Zeiten für den 3. Sprint	6
2	Übersicht der Zeiten für den 4. Sprint	25