**Release 4 Documentation**

**Project Owner:**

Vadim Simukov

**Scrum Master:**

Christopher Bartalos

**Developer:**

Jonas Thelemann

Birk Magnussen

Gianluca Voss

Rohhullah Walizada

Patrick Michla

Christopher Bartalos

**Table of contents**

[1 Introduction 6](#_Toc18768617)

[2 First Ideas 7](#_Toc18768618)

[2.1 Mock-ups 7](#_Toc18768619)

[2.1.1 Army Manager in the Game Lobby 8](#_Toc18768620)

[2.1.2 Health Point Bars 9](#_Toc18768621)

[2.1.3 Visualization of Health Reduction 10](#_Toc18768622)

[2.1.4 Player Colour Outline 11](#_Toc18768623)

[2.1.5 AI Takeover 12](#_Toc18768624)

[2.2 Domain/User Stories 13](#_Toc18768625)

[2.2.1 Army Manager in the Game Lobby 14](#_Toc18768626)

[2.2.2 Health Point Bars 15](#_Toc18768627)

[2.2.3 Visualization of Health Reduction 16](#_Toc18768628)

[2.2.4 AI takeover 17](#_Toc18768629)

[3 Sprints 18](#_Toc18768630)

[4 Sprint Seven 18](#_Toc18768631)

[4.1 Earlier Issues 18](#_Toc18768632)

[4.1.1 TB-45: 3.3.5 Chat Encryption 18](#_Toc18768633)

[4.1.2 TB-76: Memes 19](#_Toc18768634)

[4.1.3 TB-77: Javadoc 20](#_Toc18768635)

[4.2 Issues from Sprint Six 21](#_Toc18768636)

[4.2.1 TB-138: Properly Handle REST-Response Status Codes 21](#_Toc18768637)

[4.2.2 TB-141: Add Handler to Ingame Chat from GameSocket 21](#_Toc18768638)

[4.2.3 TB-152: Logging out Does Not Work on Crashed Sockets. Allow for Logging In Again. 22](#_Toc18768639)

[4.2.4 TB-153: Evaluate and Fix Failing Integration Tests 22](#_Toc18768640)

[4.2.5 TB-158: Closing the Login Screen with Stored Credentials Deletes Stored Credentials 23](#_Toc18768641)

[4.2.6 TB-165: Make Notifications Dynamic and Unify Their Language 23](#_Toc18768642)

[4.2.7 TB-169: Fix Bug That Allowed Bad Login Credentials 23](#_Toc18768643)

[4.3 Issues 7. Sprint 25](#_Toc18768644)

[4.3.1 TB-170: Show Health Points 25](#_Toc18768645)

[4.3.2 TB-171: Show Movement Point 25](#_Toc18768646)

[4.3.3 TB-172: Border of Units 26](#_Toc18768647)

[4.3.4 TB-173: Armys in Game Lobby 26](#_Toc18768648)

[4.3.5 TB-174: Show Initial Health Bars (Sub-Task) 27](#_Toc18768649)

[4.3.6 TB-175: Health Point Reduction (Sub-Task) 27](#_Toc18768650)

[4.3.7 TB-176: AutoMode 27](#_Toc18768651)

[4.3.8 TB-177: AI First Stage 28](#_Toc18768652)

[4.3.9 TB-178: AI Second Stage 29](#_Toc18768653)

[4.3.10 TB-179: AI Third Stage 29](#_Toc18768654)

[4.3.11 TB-180: Implement Bot Creation 30](#_Toc18768655)

[4.3.12 TB-181: Bot Usage 30](#_Toc18768656)

[4.3.13 TB-182: Different Bot Difficulties 31](#_Toc18768657)

[4.3.14 TB-184: Show Turn After Initialization 31](#_Toc18768658)

[4.3.15 TB-185: Minimap Unit Positions 32](#_Toc18768659)

[4.3.16 TB-189: Turn UI Further Improvements 32](#_Toc18768660)

[4.3.17 TB-191: WinScreen Further Improvements 33](#_Toc18768661)

[4.3.18 TB-192: Units Can Walk Into Other Units 33](#_Toc18768662)

[4.3.19 TB-193: Remove Statics from all Sockets and make them instanceable 34](#_Toc18768663)

[4.3.20 TB-194: GameSocket Disconnect After Finishing a Game 35](#_Toc18768664)

[4.3.21 TB-196: Army Manager Bug Fixes 35](#_Toc18768665)

[4.3.22 TB-197: Fix Logout Logic 36](#_Toc18768666)

[4.4 Burndown Chart 37](#_Toc18768667)

[4.5 Sprint End 39](#_Toc18768668)

[5 Sprint Eight 40](#_Toc18768669)

[5.1 Unfinished Issues from Sprint Seven 40](#_Toc18768670)

[5.2 Issues Sprint Eight 40](#_Toc18768671)

[5.2.1 TB-183: Unit Move/Attack 40](#_Toc18768672)

[5.2.2 TB-195: Implement AI Difficulties 41](#_Toc18768673)

[5.2.3 TB-199: Identify and Fix AI Error 41](#_Toc18768674)

[5.2.4 TB-200: Incorrect Connection to a Game 42](#_Toc18768675)

[5.2.5 TB-201: Feature: Tetris 42](#_Toc18768676)

[5.2.6 TB-202: Feature: Pong 42](#_Toc18768677)

[5.3 Burndown Chart 43](#_Toc18768678)

[5.4 Sprint End 44](#_Toc18768679)

[6 Results 45](#_Toc18768680)

[6.1 Game Lobby 45](#_Toc18768681)

[6.2 Health Bars 46](#_Toc18768682)

[6.3 Health Bars reduction 47](#_Toc18768683)

[6.4 Player Colour Outline 47](#_Toc18768684)

[6.5 AutoMode 48](#_Toc18768685)

[6.6 AI and Bots 48](#_Toc18768686)

[6.6.1 Bots 48](#_Toc18768687)

[6.6.2 AI 49](#_Toc18768688)

[7 Closing Words 49](#_Toc18768689)

**Image Directory**

[Figure 1: Button For Army Manager 8](#_Toc18768690)

[Figure 2: In Game Health Bars 9](#_Toc18768691)

[Figure 3: Health reduction 10](#_Toc18768692)

[Figure 4: Player Colour Outlier 11](#_Toc18768693)

[Figure 5: AI Takeover Button 12](#_Toc18768694)

[Figure 6: Army Manager Inside Game Lobby 14](#_Toc18768695)

[Figure 7: Units Health In Game 15](#_Toc18768696)

[Figure 8: Visual Health Reduction 16](#_Toc18768697)

[Figure 9: AI Takeover Process 17](#_Toc18768698)

[Figure 10: Burndown Chart Sprint Seven: Story Points 37](#_Toc18768699)

[Figure 11: Sprint Seven: Issues 37](#_Toc18768700)

[Figure 12: Burndown Chart Sprint Seven: Original Time Estimate 38](#_Toc18768701)

[Figure 13: Burndown Chart Sprint Seven: Issue Count 38](#_Toc18768702)

[Figure 14: Burndown Chart Sprint Eight: Story Points 43](#_Toc18768703)

[Figure 15: Burndown Chart Sprint Eight: Original Time Estimate 44](#_Toc18768704)

[Figure 16: Burndown Chart Sprint Eight: Issues Count 44](#_Toc18768705)

[Figure 17: Implementation: Game Lobby 45](#_Toc18768706)

[Figure 18: Implementation: Health Bars 46](#_Toc18768707)

[Figure 19: Implementation: Health Bars reduction 47](#_Toc18768708)

[Figure 20: Implementation: Player Colour Outline 47](#_Toc18768709)

[Figure 21: Implementation: AutoMode 48](#_Toc18768710)

[Figure 22: Bot creation 48](#_Toc18768711)

# Introduction

In this documentation you will read full documentation of the forth release of module Software Engineering 1 project RBSG.

This documentation does not contain information about previous releases. If you are looking for details regarding releases 1 to 3, please visit previous documentations.

As stated in previous releases, this project uses Jira software with all it’s necessary tools for project management.

In this documentation we will use Jira terms like a task, a bug or a story and many more to classify unfinished parts of the project.

# First Ideas

The first ideas of this release will be listed in the Mock-ups to follow.

## Mock-ups

In this chapter we will look at the developed Mock-ups which were introduced at the meeting with the customers.

Mock-ups are used by designers mainly to acquire feedback from users about designs and design ideas early in the process. Mock-ups are “very early prototypes” made of cardboard or otherwise low-fidelity material. In our case we designed our Mock-ups on a computer and printed them on simple paper sheets.

Since Mock-ups are very early prototypes, results may vary.

### Army Manager in the Game Lobby

Release three brought to us exciting new ideas. Game lobby was one of those ideas. Game lobby was introduced as a step before joining the game fully. Players could choose their respectful armies before each game and make themselves ready for the game to come. After enough players joined for the game and completed the steps explained earlier the game could be initialized.

In our previous design player could choose one out of three armies previously saved on the device. It was inconvenient for the player who joined a game without saved armies. To solve that problem a new design for the game lobby was introduced. New game lobby will have a way to create new armies even when a player has already joined the game. For that a new button will be added to the lobby which will open the army manager as shown in Figure 1.

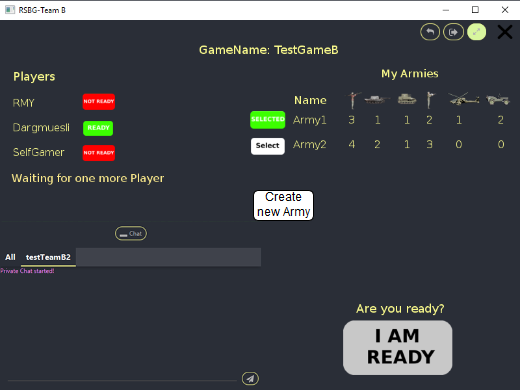


Figure 1: Button For Army Manager

### Health Point Bars

The Idea of health bars or any other way of visually indication the health points of each unit has already originated in the first release and was logically put on hold before it could be implemented in the game.

Each unit has same amount of health points and in strategic games they should not be guessed by the player. To find the right strategy player must know the amount of health points his units have. The player wants to get into the game and have the needed information to be presented right away without completing extra steps. For that we introduce health bars. Each unit will have their health points shown to the players playing the game as shown in Figure 2.



Figure 2: In Game Health Bars

### Visualization of Health Reduction

The next design idea is a continuation of health bars shown in the game.

As discussed in the previous Mock-up, players want to have useful information presented instantly to them. Health point reduction is easily one of the most useful information you must show the player. It is one of the greatest satisfaction factors to see an enemy unit health points being lower than allied unit as shown in Figure 3.



Figure 3: Health reduction

### Player Colour Outline

Like in Chess all the pieces in the game have the same look. The only difference is who controls the units on the game field. To make it easier for the players to recognize their units, the game should provide help. Best way to help player find their units is to provide them information visually.

We decided to give units outline in the colour provided to them from server as they join the game.



Figure 4: Player Colour Outlier

### AI Takeover

Implementing the Artificial Intelligence (AI) is the biggest part of the release four. There are almost no design preferences given from the customer. It should be possible for the AI to take over the game for the player playing it and otherwise. The customer did not specify how the AI should behave in certain situation except for it should do everything possible to win the game against its enemy, be it a player or the AI itself.

Our take on the AI takeover is shown in Figure 4. The takeover will be displayed as a button “AutoMode” on top of the user interface.



Figure 5: AI Takeover Button

## Domain/User Stories

User stories are short, simple descriptions of a feature in the system under development told from the perspective of the person who desires this new capability. This person is normally a user of the system or even a customer who pays for the solution.

In our case the domain/user stories do not come directly from user. User stories are designed by the user’s specific functions given some situations. They are of course highly linked to our specific model of the project.

### Army Manager in the Game Lobby

The process of creating a new army inside the game lobby should be easy and without any problems for the user.

It could be implemented as a button which will transition game lobby window into army manager. After player is done deciding his next new army, he should be able to transition himself back into game lobby.

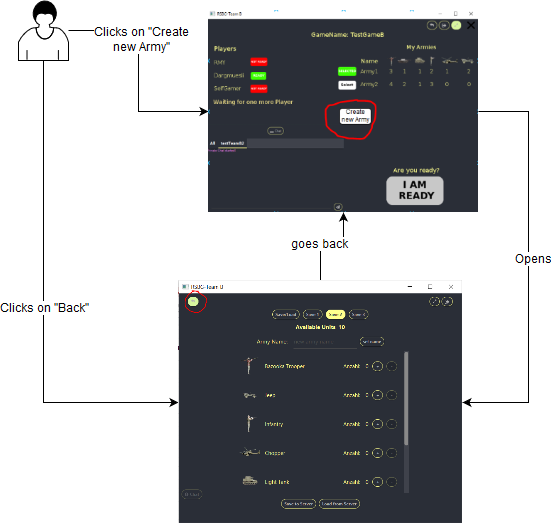


Figure 6: Army Manager Inside Game Lobby

### Health Point Bars

In a strategy game most of the time players do not expect any kind of extra steps to show them the information about the health points of their units. Players want to get the specific information right away and it should be very clear for the player. Colour green for health points of any kind is very prominent in the gaming industry and is also the primary colour for our planet earth. Green is the best colour to show that the units are full of health and ready help the player defeat his enemy.

It should be held simple. After the game has started, all the units should show the health points of each unit.

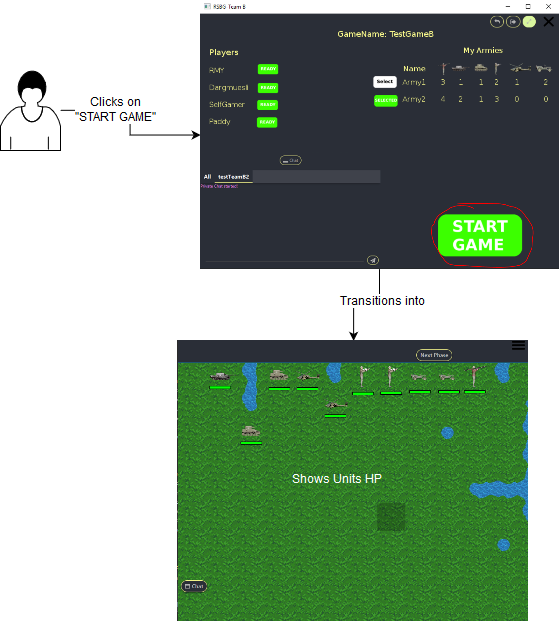


Figure 7: Units Health In Game

### Visualization of Health Reduction

Visualizing health reduction is the next evolution to health bars and a very important part of the whole game. To properly strategize you next move it is crucial to know the health points of your enemy units.

Before health reduction is shown the user or his enemy must attack a unit. After attack has finished the remaining health points should be shown in green as before and lost health points in red.

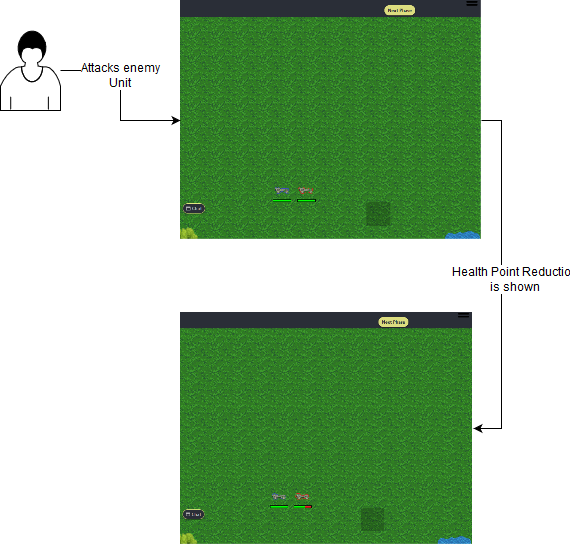


Figure 8: Visual Health Reduction

### AI takeover

AI takeover is a concept not very often used in gaming. Most of the time the players want to be engaged in gaming himself. This concept of AI takeover is found in mobile games. Mobile games differ slightly from Console gaming or PC gaming. Focus of PC/Console games are player in your home where all the distractions are kept to a minimum. Mobile games are played not on the couch, but mostly in places like the buses or trains. So, the AI takeover makes sense so you can let the AI do some tedious things while you are busy with the cashier. After your distraction you can take over the game again.

A button should allow the AI to take over your turn and the turn to follow if you decide not to press it again.

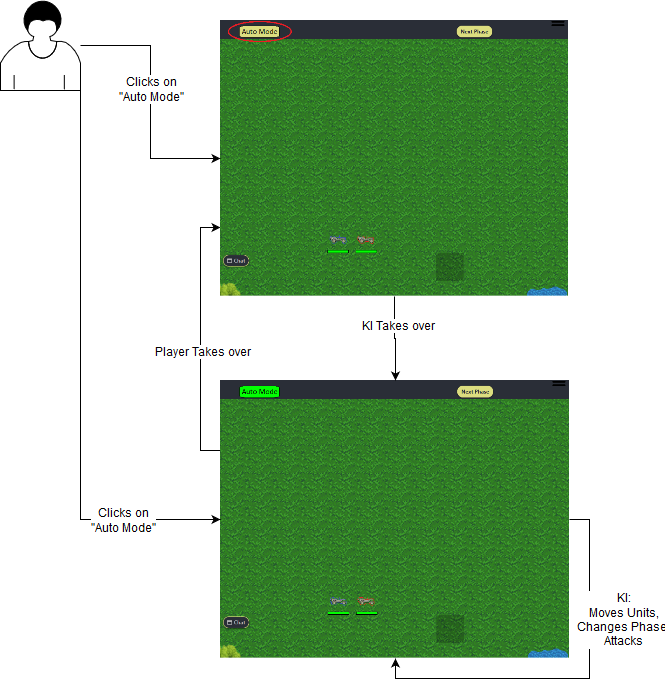


Figure 9: AI Takeover Process

# Sprints

A sprint is a set of development activities conducted over a pre-defined period, usually one to four weeks. The interval is based on product complexity, risk assessment, and degree of oversight desired. Sprint speed and intensity are driven by the selected duration of the sprint. Risk is assessed continuously, and adequate risk controls and responses put in place.

We chose to have sprints of 2 weeks length.

# Sprint Seven

This sprint has its focus set on the implementation of AI and finishing urgent issues not finished in sprint 6. The secondary focus is to create a stable intuitive User Interface (UI) for users of this game.

## Earlier Issues

Issues that were not finished in previous sprints in creation order.

### TB-45: 3.3.5 Chat Encryption

Type: Task

Estimated Time: 2d 4h

Priority: Low

Description:

As a user I want to be able to send encrypted chat messages to other users of this application and decrypt read decrypted messages to me by default to ensure my data's privacy.

This is a special feature. An option to show the own and remote public key should exist. A private key must be generated by each client for this purpose. Key negotiation seems only possible via a few first unencrypted chat messages that cannot be interpreted by other game clients.

Status: Done (Sprint eight)

Time Logged: 1d 2h

Developer assigned to this task had to drop out from the Project. In sprint eight another developer took over the development. At first the developer took 2 hours to rework the code for encryption. After that the developer was working 5 hours total on the handling of the encrypted messages and their serialization. Writing the tests took 3 hours.

### TB-76: Memes

Type: Story

Estimated Story Points: 5

Estimated Time: 5h

Priority: Low

Description:

Start:

Der Benutzer ist im Registrierungsbildschirm oder Anmeldebildschirm ...

Aktion:

Der Benutzer gibt falsche Daten oder schon vorhandene Daten ein.

Ende:

Es taucht ein Meme auf, dadurch wird der Benutzer verspottet und auf seine unzulässige Eingabe verwiesen.

Translated description:

Start:

User is in register/login window…

Action:

User does not enter correct information or already used login data.

End:

A meme pops up which ridicules the user referring his invalid input.

Status: Done (Sprint eight)

Time Logged: 4h

### TB-77: Javadoc

Type: Task

Estimated Time: 1d 4h

Priority: Lowest

Description:

In der Java-Dokumentation/Literatur (Javadoc) werden alle Programmabläufe, Klassen, Methoden und die verwendeten Bibliotheken (Librarys) erklärt und das Zusammenspiel dieser bezogen auf das gesamte Projekt. (Ebenfalls wird auch das Zusammenspiel von diesen, bezogen auf den gesamten Projekt, erläutert.)

Translated description:

Java documentation/literature (Javadoc) contains all the explanations to program flows, classes, methods and used libraries, and the interactions between them. (Likewise all the interactions related to the whole project will be explained)

Status: Done (Sprint eight)

Time Logged: 1d 3h

## Issues from Sprint Six

Here is a list of issues that were not finished specifically in sprint 6 in creation order.

### TB-138: Properly Handle REST-Response Status Codes

Type: Task

Estimated Time: 4h

Priority: Medium

Description:

Code TODO (AbstractRestRequest L65)

L65: Handle result status codes

Status: Done (Sprint eight)

Time Logged: No Time Logged

### TB-141: Add Handler to Ingame Chat from GameSocket

Type: Task

Estimated Time: 4h

Priority: Medium

Description:

Code TODO (GameSocket L140 + L152)

Changed to: Code TODO (GameSocket L244 + L421)

L244: maybe handler to chat window

L421: receive chat messages

Status: Done (Sprint eight)

Time Logged: 5h

### TB-152: Logging out Does Not Work on Crashed Sockets. Allow for Logging In Again.

Type: Bug

Priority: Medium

Description:

Logging out does not work on crashed sockets. Allow for logging in again.

Status: Done (Sprint eight)

Time Logged: 3h 45m

### TB-153: Evaluate and Fix Failing Integration Tests

Type: Bug

Priority: Medium

Description:

Evaluate and fix failing integration tests.

Status: Done (Sprint eight)

Time Logged: 1d 4h 30m

Developer assigned was not able to identify this bug. After 4 hours of work another developer took over. This person was able to find the problems in a mere hour and started solving the issues. It took him 3 hours to repair the problem and find bad tests. Developer also took time to fix and add missing tests.

### TB-158: Closing the Login Screen with Stored Credentials Deletes Stored Credentials

Type: Bug

Priority: Medium

Description:

Closing the login screen with stored credentials deletes stored credentials.

Status: Done

Time Logged: 30m

### TB-165: Make Notifications Dynamic and Unify Their Language

Type: Task

Estimated Time: Not Specified

Priority: Medium

Description:

Make notifications dynamic and unify their language .

Status: Done

Time Logged: 1d 3h 35m

A lot of time in this task consumed by fixing UI tests headless mode as well as updating them after other issues completion.

### TB-169: Fix Bug That Allowed Bad Login Credentials

Type: Bug

Priority: Medium

Description:

Fix bug that allowed bad login credentials

Status: Done

Time Logged: 1h 30m

## Issues 7. Sprint

In the following chapter you will find a list of issues from sprint seven in creation order.

### TB-170: Show Health Points

Type: Story

Estimated Story Points: 6

Estimated Time: 6h

Priority: Medium

Description:

Health point of every unit should be shown.

The game has started, and the units are shown. Each unit has health bars. The amount of health points is shown.

Status: Done (Sprint eight)

Time Logged: 1d 6h 30m

This story took considerably longer because developer faced unexpected issues with JavaFX. Time was incorrectly logged into story instead of sub-task TB-174.

### TB-171: Show Movement Point

Type: Task

Estimated Time: 6h

Priority: Medium

Description:

The amount of movement points after the unit is slightly moved should be shown correctly.

Status: Done

Time Logged: 0h

Developer worked on this issue before it was created.

### TB-172: Border of Units

Type: Task

Estimated Time: 4h

Priority: Medium

Description:

Players should be able to distinguish which units belong to them.

Status: Done

Time Logged: 7h

Developer took more time than expected because of research for proper implementation.

### TB-173: Armys in Game Lobby

Type: Bug

Priority: Medium

Description:

The army you just created in army manager in the game lobby is not chosen as your current army.

Status: Done

### TB-174: Show Initial Health Bars (Sub-Task)

Type: Sub-Task

Estimated Time: 3h

Priority: Medium

Description:

Show health bars with health points on them for each unit.

Status: Done (Sprint Eight)

Time Logged: No Time Logged

Developed did not specify how much time was spent for this sub-task.

### TB-175: Health Point Reduction (Sub-Task)

Type: Sub-Task

Estimated Time: 3h

Priority: Medium

Description:

When attacked, units lose health. This should be visible in the health bars.

Status: Done (Sprint Eight)

Time Logged: 3h

### TB-176: AutoMode

Type: Story

Estimated Story points: 3

Estimated Time: 3h

Priority: Highest

Description:

The AI takes over the control of the game for you. Player cannot interact with the game except for the "AutoMode" button, chat and moving the game field. AI can only be activated at the start of a round in which nothing was changed. After a change, like moving a unit, button should be deactivated.

Start:

The players are ready, and the game starts.

Action:

The players click on AutoMode button.

End:

The AI takes over the control of the game.

Status: Done

Time Logged: 3h

### TB-177: AI First Stage

Type: Task

Estimated Time: 1h

Priority: Highest

Description:

Implementing AI: First Stage

AI can move all units in move turn and end it.

Status: Done

Time Logged: 1h 30m

Developer needed 30 minutes to create a mocked test.

### TB-178: AI Second Stage

Type: Task

Estimated Time: 1h

Priority: Highest

Description:

Implementing AI: Second Stage

AI can identify enemy units and attack them.

Status: Done

Time Logged: 2h

Second Stage of AI took half an hour longer as expected. Developer also added a mocked test which took 30 minutes.

### TB-179: AI Third Stage

Type: Task

Estimated Time: 1d 2h

Time estimation was discussed and agreed upon with the developer team and developer designated. Developer was not sure how much time was needed for the AI.

Priority: Medium

Description:

Implementing AI: Third Stage

AI actually gets intelligent and tries to win the game.

Status: Done

Time Logged: 2d 3h 30m

Developer took considerably more time than expected. Developer needed to build a solid foundation to be able to work further on the AI. Writing test was also very time consuming. It took developer 1h 30m to think up the AI “Nagato” and start constructing it. Another 8h and 55m to fix initial bugs and AI logic. Next 5 hours were needed for calculation important things like attack/movement patterns and damage predictions. After it was all done developer fixed merge conflicts and wrote tests.

### TB-180: Implement Bot Creation

Type: Task

Estimated Time: 5h

Priority: Highest

Description:

New server feature: bot creation.

Creation of a bot needs to be implemented.

Status: Done

Time Logged: 6h 15m

Overall it took developer almost the expected amount if time. Fixing merge conflicts occupied extra 35m.

### TB-181: Bot Usage

Type: Task

Estimated Time: 5h

Priority: High

Description:

Bot usage should be implemented.

Bot needs an army.

Bot needs to be taken over by the AI.

Status: Done (Sprint eight)

Time Logged: 3d 1h 20m

This task consumed considerably more resources than expected. Many things needed to be researched and tested. At first developer talked with AI developer. After understanding how the AI works GameSocket needed to be able to distinct between a Bot and a Player. Many bugs were found and fixed. Most of the time was needed to establish proper communication between bot GameSocket and AI. Last few hours were needed to create tests and add delete bot functionality.

### TB-182: Different Bot Difficulties

Type: Task

Estimated Time: 1d 2h

Priority: Low

Description:

Different bot difficulty levels can be added.

Status: Done (Sprint eight)

Time Logged: No Time Logged

This task was finished but developer did not log the time in this Task, but rather in TB-181.

### TB-184: Show Turn After Initialization

Type: Bug

Priority: Medium

Description:

Turns are not shown in the first turn. So, players can’t see whose turn it is on "round 1".

Status: Done

Time Logged: 10m

### TB-185: Minimap Unit Positions

Type: Bug

Priority: Medium

Description:

Unit movement is not displayed on the mini map this needs to be fixed. So that players can see where their units are.

Status: Done

Time Logged: 1h

### TB-189: Turn UI Further Improvements

Type: Task

Estimated Time: Not Specified

Priority: Medium

Description:

Turn Ui does not scale with screen size. This needs to be fixed. Also make Ui look more appealing.

Status: Done

Time Logged: 3h

Since no time was estimated no comparison could be done.

### TB-191: WinScreen Further Improvements

Type: Task

Estimated Time: Not Specified

Priority: Medium

Description:

Win screen does not scale with the screen size.

Also, Label does not correctly fit into the space of the trophy as it should. This needs to be fixed.

Status: Done

Time Logged: 6h

Since no time was estimated no comparison could be done.

### TB-192: Units Can Walk Into Other Units

Type: Bug

Priority: Highest

Description:

Title and comment:

"Es gibt zusätzlich noch einen Bug, dass die Units in andere Unit reinlaufen können

zumindest wird das Ganze im Overlay gehighlighted (vorallem in die previousTileMapById gepackt), und das stört die AI, die sich aufs Overlay verlässt

zusätzlich müssen ja auch noch angreifbare Units (auch außerhalb der MP) angezeigt werden, aber nicht in der previousTileMapById landen"

Translated description:

Title and comment:

“There is also a bug present, that the units can walk into other units, at least it is what the overlay shows (especially it is put inside previousTileMapById), and it is disrupting the AI which relies on the Overlay.

Also, the unit that could be attacked (that are in range of Movement Points) should be shown, but not be putted inside previousTileMapById”

Status: Done

Time Logged: 2h

### TB-193: Remove Statics from all Sockets and make them instanceable

Type: Task

Estimated Time: 1d 7h

Priority: Highest

Description:

"Wir haben folgendes Problem: in den GameSocket gibt es static Sachen, welche nicht static sein dürfen. Es ist sehr wichtig, dass alle static Sachen aus dem GameSocket herausgenommen werden und die Sockets instanziiert. Bzw. nicht nur GameSocket, andere auch."

Translated description:

“We have following problem: GmesSocket has static things which should not be there. It is very important to take out all the static things out of GameSockets and instantiate the sockets or rather not only GameSocket but the other sockets too.”

Status: Done

Time Logged: 5h 45m

Took much less time as estimated, because some very problematic points were fixed with ease.

### TB-194: GameSocket Disconnect After Finishing a Game

Type: Bug

Priority: Medium

Description:

Bugreport:

Wenn man ein Game Completed und in die Lobby kommt, connected der Websocket nicht.

Vermutlicher Lösungsansatz:

Man muss, beim "Back to Lobby" Knopf vom Win Screen explizit von GameSocket disconnecten.

Translated description:

Bugreport:

When you complete a game and go back into the lobby, web socket doesn’t connect.

Presumable solution approach:

You must explicitly disconnect from GameSocket when you click “Back to Lobby” button on the win screen.

Status: Done (Sprint eight)

Time Logged: 3h 20m

Time was consumed to find the reason of the bug. After consultation with other developers, the problem was found inside web socket. Developer understood the problem and could resolve it.

### TB-196: Army Manager Bug Fixes

Type: Bug

Priority: Medium

Description:

Army manager bug fixes

Status: Done

Time Logged: 30m

### TB-197: Fix Logout Logic

Type: Bug

Priority: Medium

Description:

When a player tries to logout, while he is still in game, the game should ask if the player wants to concede the game.

Status: Done

Time Logged: 30m

## Burndown Chart

After completing the sprint, we encountered a bug in Jira. The story points shown in burndown chart do not add up for the stories created. The only stories created for this sprint are TB-76, TB-170 and TB-176 which add up to **14** story points, but the story point count in the chart goes up to **53** points total (Figure 10).

With further investigation it was found that some of the tasks which came from previous releases, had story points given to them. Those tasks have at some point been stories with a huge amount of story points given to them. After converting them to tasks the story points did not disperse, they stayed till the last sprint. The 2 tasks distorting the burndown chart are TB-45 and TB-77 with 30 and 12 story points respectively (Figure 11).

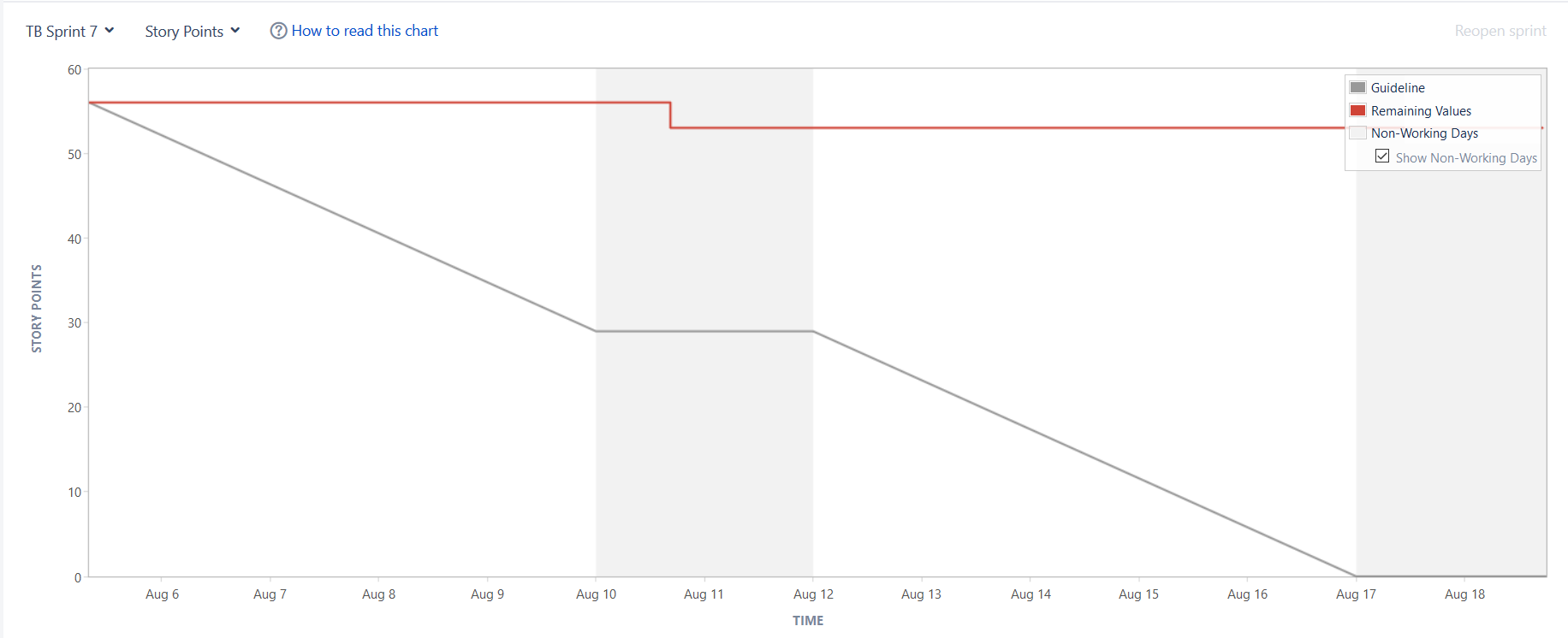


Figure 10: Burndown Chart Sprint Seven: Story Points

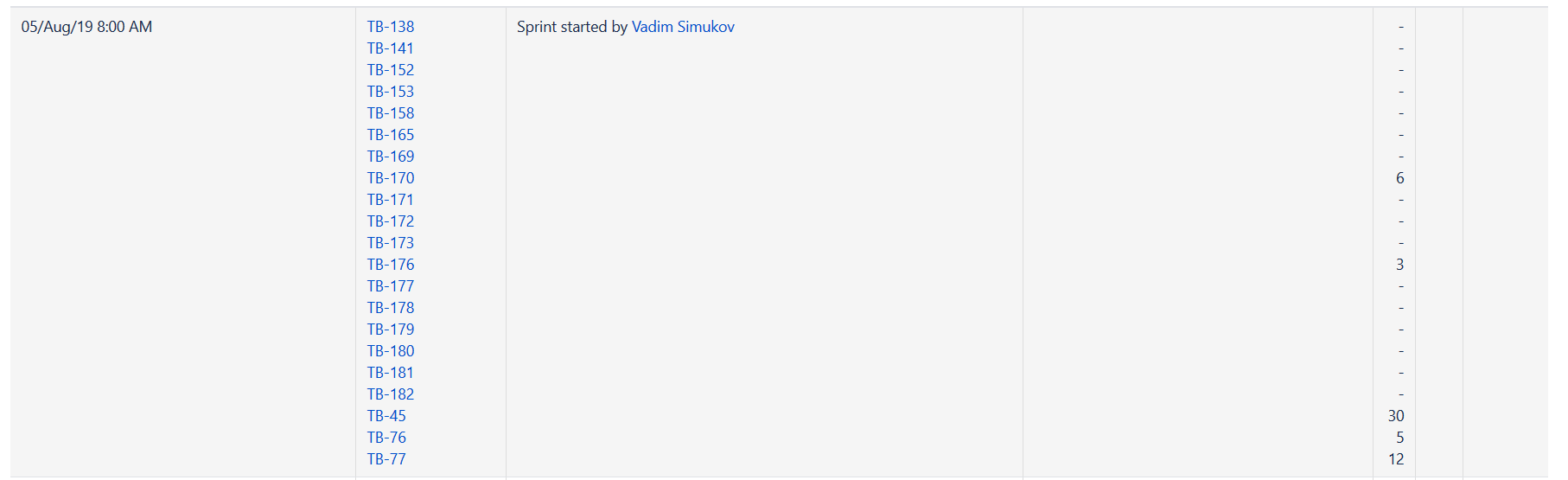


Figure 11: Sprint Seven: Issues

For clarification purposes and to get real view about sprint seven we will have to add burndown charts for estimated time (Figure 12) and issue count (Figure 13).

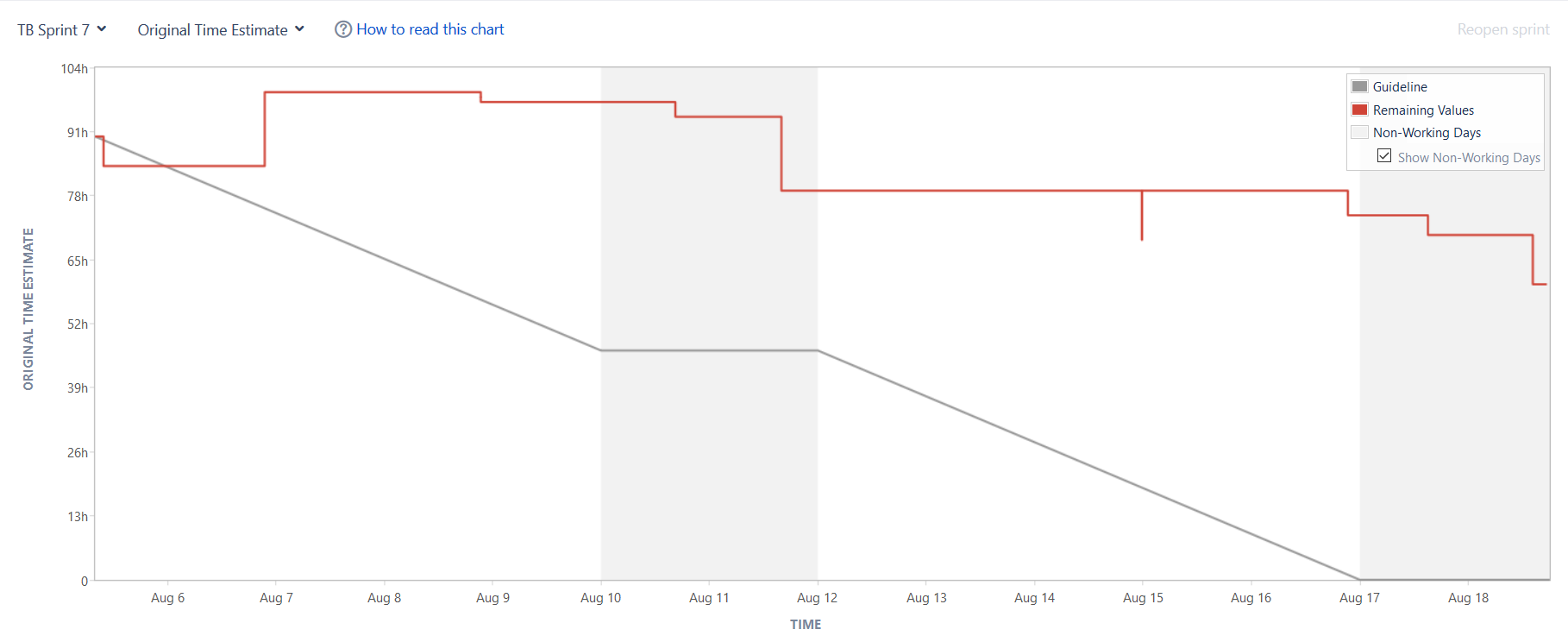


Figure 12: Burndown Chart Sprint Seven: Original Time Estimate

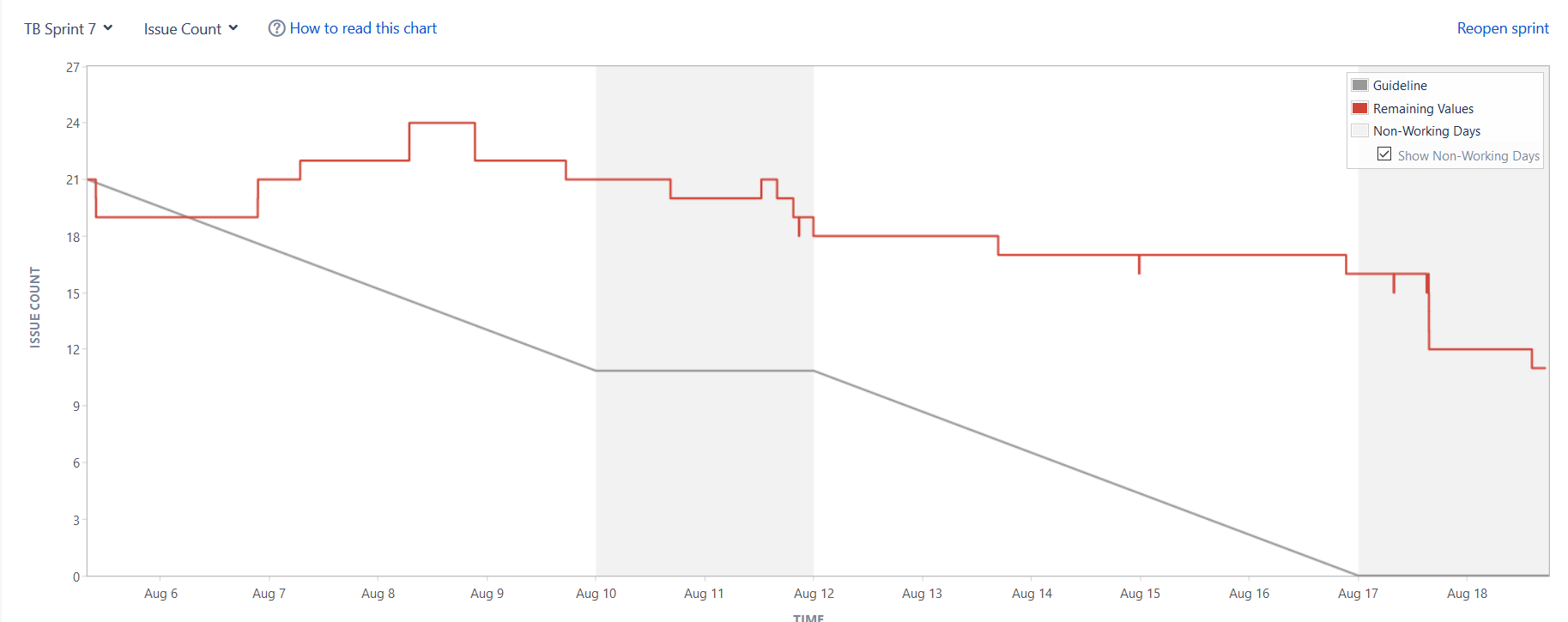


Figure 13: Burndown Chart Sprint Seven: Issue Count

## Sprint End

In the figures above we can clearly see that the estimated work was wrong. It turned out to be too much work for the developers working on this project. Many issues had to be postponed into sprint eight.

Estimation of time needed to finish the tasks, was discussed between all developers. Every developer could state how time he would need for the task at hand. Mean of all estimations was set as “Original Estimate” in Jira. Many issues were finished on time and many needed more time than expected due to their complexity.

# Sprint Eight

## Unfinished Issues from Sprint Seven

Unfinished issues from sprint seven will be put into sprint eight and will continued to be worked on until finished. Unfinished issues were moved and have “Done in Sprint Eight” as their status. To list them all here would be too redundant. A simple list will show the unfinished issues.

Unfinished issues:

* TB-45
* TB-76
* TB-77
* TB-138
* TB-141
* TB-152
* TB-153
* TB-170
* TB-174
* TB-175
* TB-181
* TB-182
* TB-194

## Issues Sprint Eight

Tasks, stories and bugs of this sprint will primarily be tackling issues of usability and tidiness of the program.

### TB-183: Unit Move/Attack

Type: Bug

Priority: Medium

Description:

Get the range of the display of the move fields up to one more field to each side and display it differently so that a player knows a unit can move "x" far and can attack a unit in that range.

Status: Done

### TB-195: Implement AI Difficulties

Type: Task

Estimated Time: 1d 7h

Priority: Highest

Description:

AI needs different difficulty levels for the players to test their strategic skills.

Status: Done

Time Logged: 6h

Task was finished early, because the developer already thought of the AI difficulties beforehand and only needed to implement them.

### TB-199: Identify and Fix AI Error

Type: Bug

Priority: Medium

Description:

While in AutoMode, after some time you get error message:

"GameSocket - InGameError: Unit with id Unit@UNIT\_ID" does not exist"

Status: Done

Time Logged: 1h

### TB-200: Incorrect Connection to a Game

Type: Bug

Priority: Medium

Description:

After finishing a game, with either a win or a loss, and you try to join another game you get an "incorrect connection" message from GameSocket.

Status: Done

### TB-201: Feature: Tetris

Type: Task

Estimated Time: 2h

Priority: Low

Description:

Add a new feature that das not influence the game but can be played in the meantime.

Status: Not Done

Developer did not have enough time to finish his task on time. Time management was a big issue and the team was not able to help the developer in question with the completion of this task.

### TB-202: Feature: Pong

Type: Task

Estimated Time: 2h

Priority: Low

Description:

Add Pong as feature into the game. Find a creative way to start it.

Status: Done

Time Logged: 5h

Developer in question had to rework his code due to bad usability.

## Burndown Chart

Burndown charts of sprint eight look more pleasant to the eye. In the first week we see almost no progress in the development. That is due to bugs and tasks being very complicated to find and solve. After the first week see rapid fall in story points (Figure 14), reason being the bug described in Chapter 4.4.

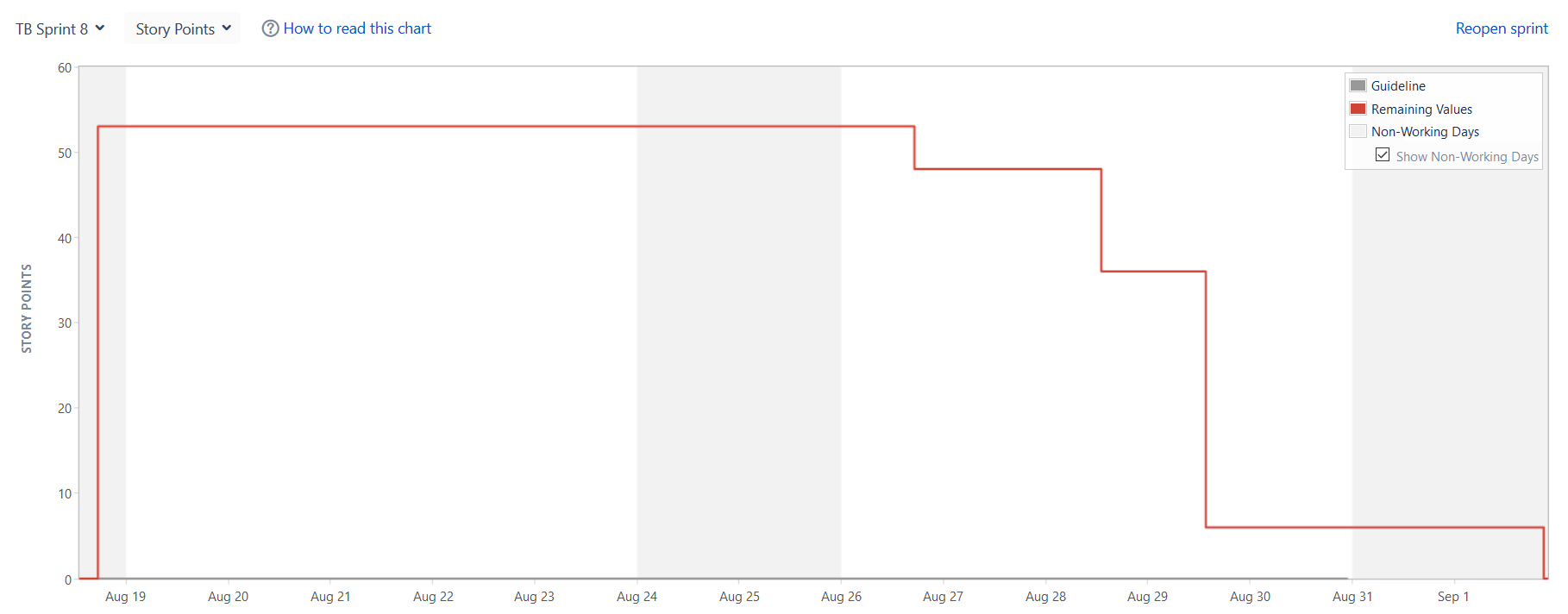


Figure 14: Burndown Chart Sprint Eight: Story Points

For the same reason we see a decline in time estimated chart (Figure 15). At the start new bugs were found and added to sprint.

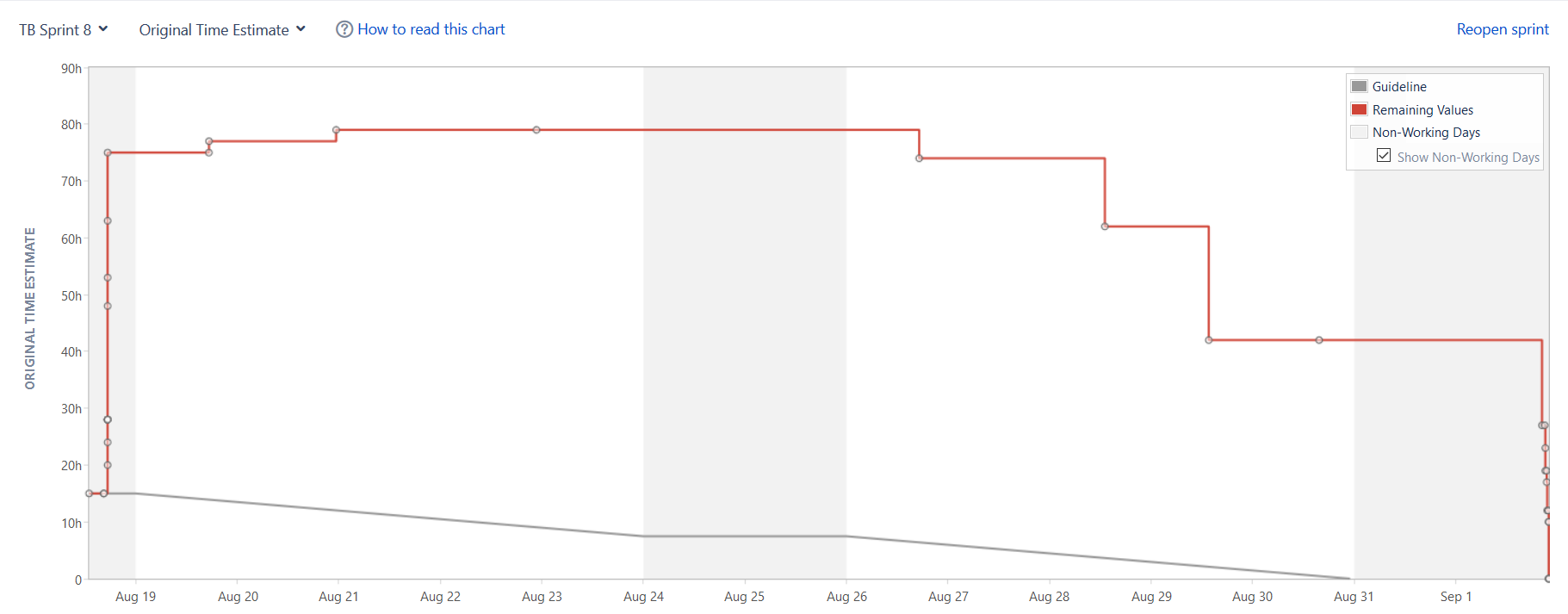


Figure 15: Burndown Chart Sprint Eight: Original Time Estimate

Many crucial bugs were finished at the last day of sprint eight (Figure 15/Figure 16). Many of these bugs were deeply rooted inside the code. Many considerations needed to be found which made the way to solve the problems very thorny. All the issues had to be properly tested for the sake of making a very stable product. This made the charts look very rushed.

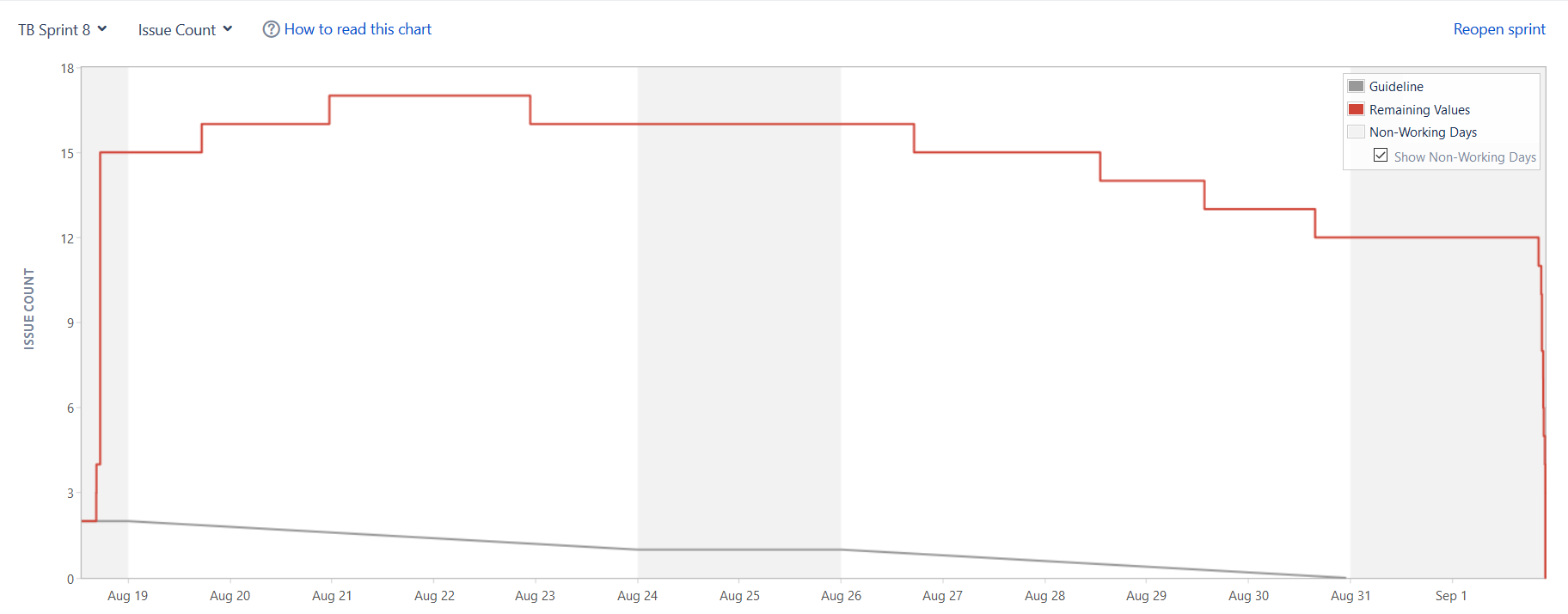


Figure 16: Burndown Chart Sprint Eight: Issues Count

## Sprint End

Sprint ended with all high priority Tasks, Bugs and Story being done. Some of the low priority features like Tetris were not finished on time and will not be present at the final product. Deep rooted bugs were found and solved but took a lot of time to solve.

# Results

## Game Lobby

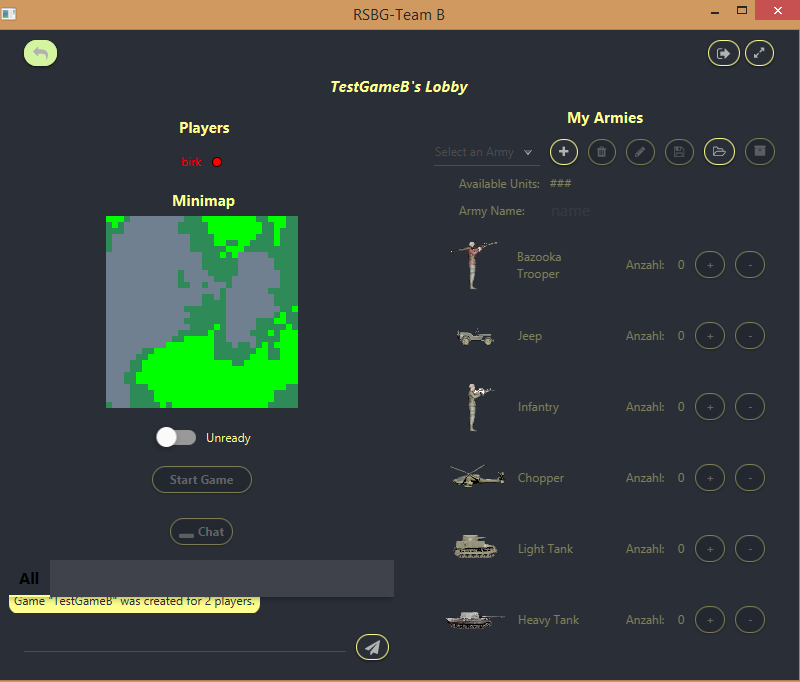


Figure 17: Implementation: Game Lobby

Implementation of game lobby (Figure 17) is vastly different to the mock-up presented (Figure 1).

Biggest difference being the position of army manager inside game lobby. In the previous release the player could select 1 out of 3 armies saved beforehand. If the player wanted a new army, he had to leave the lobby and inside the main screen go to army manager to create new army. Then he had to leave the manager and go back inside the game. In the Mock-up we tried to simplify the process of army creation by placing a “Army Manager” button inside the game lobby.

In comparison to the Mock-up we found this way to be way more efficient and fluent for the players. In this implementation we can not only choose existing army from the server but also from the device. Choosing from the device is done through the button with an opened folder icon. As you can browse through your whole device, there is an unlimited supply of armies to choose from. The armies need to be created first. To create a new army, you click on the “Plus” button and type a new name into the “Army Name” text field. After selecting your 10 units you save the army and ready yourself up with the switch button titles “Unready”. New addition to the game lobby is the preview of the map player are going to play on. This allows the players to create new strategy based on the map presented

## Health Bars

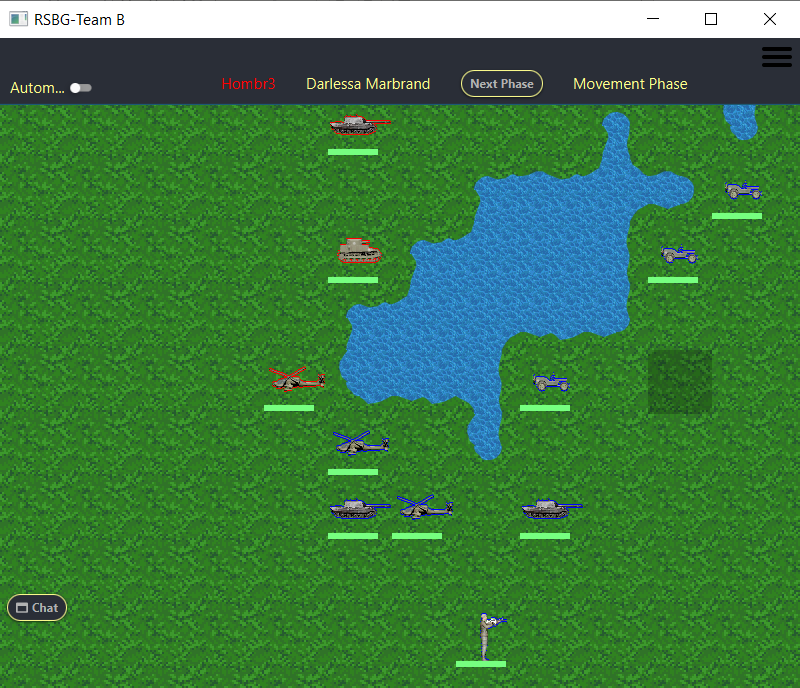


Figure 18: Implementation: Health Bars

There are not that many differences in the implementation to the Mock-up as it is fairly straight forward. We wanted the players to see the Health Points of their units and we decided to add green bars to show them.

Implementation has been almost exactly as planned (Figure 18). Only the colour was changed to a slightly brighter green. In comparison to the Mock-up.

## Health Bars reduction

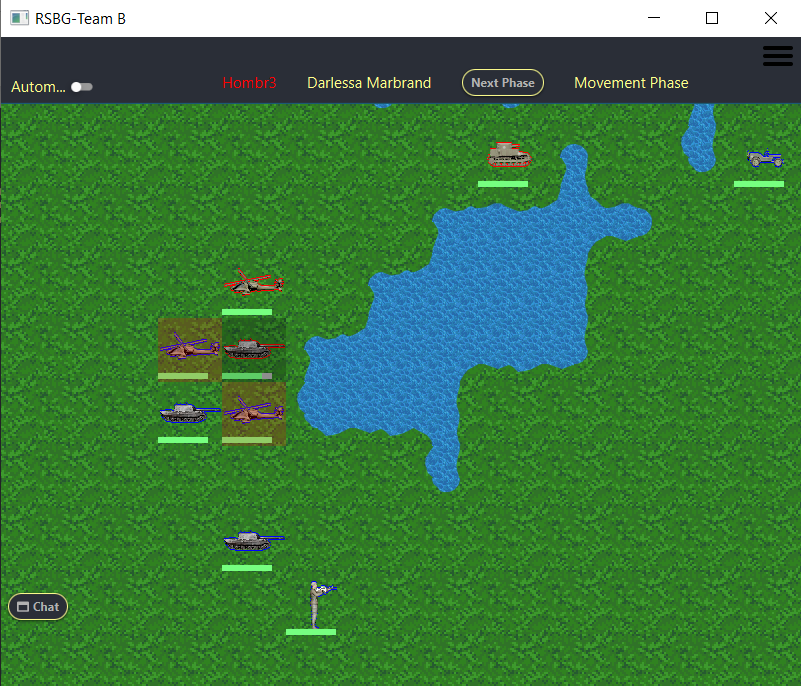


Figure 19: Implementation: Health Bars reduction

Health bars reduction did not receive many changes from the Mock-up as well.

Colour of the missing health points was changed to grey (Figure 19), which I personally did not find very useful. Product owner was not able to clarify what colour should be used for missing health until the product had to be presented, so no changes could be implemented.

## Player Colour Outline



Figure 20: Implementation: Player Colour Outline

Player colour outline was implemented as planned.

The colours are assigned by the server to players joining the game as players. We use those colours to generate outline for each unit (Figure 20). Colours being “RED”, “Blue”, “YELLOW” and “GREEN”. Each player will have unit outline in the colour designated.

## AutoMode

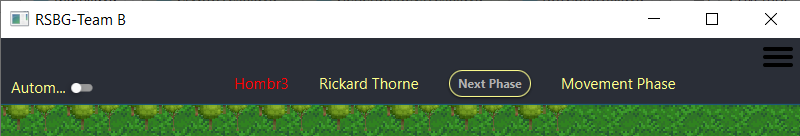


Figure 21: Implementation: AutoMode

AutoMode is only the visual aspect of the AI takeover. It was planned to be implementing using a button, but it made more sense to implement it as a Switch-button (Figure 21). This way it should be easier for the player to see in the AutoMode is active.

AI takeover can only be done if the player did not move any of his units. AutoMode can play takeover for any army it was given to it.

## AI and Bots

### Bots

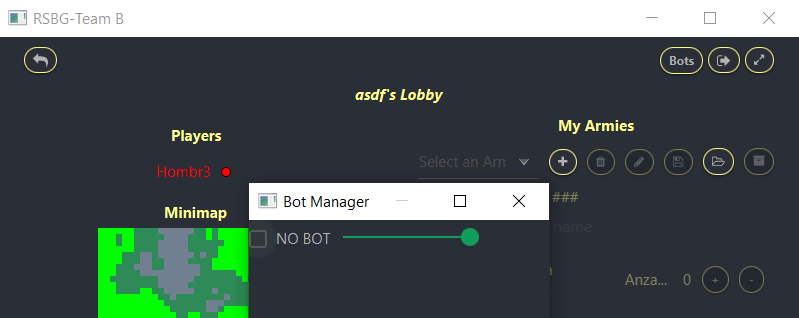


Figure 22: Bot creation

Bots are non-player players added to the game to compete with you instead of a real player.

There was no Muck-up created to show how the bot should be created. Instead only the creation of such bot was important rather than looks.

The developer had creative freedom for the ability to add a Bot-player to the game. He decided to create a button which will open a small window which will allow the player to add one or more bots depending on the size of the game player joined (Figure 22). Clicking on the checkbox near the “NO BOT” text will add a bot to the game. The name of the bot is randomly generated.

The difficulty of the bot can be chosen. On the slider, the difficulties rank from easy to normal to difficult. Default position of the slider for the bot difficulty is the most difficult one to play against.

### AI

The AI had no Mock-ups presented, since the AI had to be there but there is usually no indication that it is present except for written code.

The AI was developed by one developer which focused only on the AI for this release. He was able to create four types of AI.

The first AI is used for AutoMode and easiest bot difficulty. It can take over for the player regardless of the units the player chose. Its basic functionality lies on “find and destroy”. Each unit tries to move forward and attack and destroy a unit. If there are allied unit nearby, they try to gang up on unit to secure the kill.

The second AI is designated for the normal bot difficulty. It uses more advanced strategic approach for the game play. It tries to use the advantages of the units as well as terrain. For example, if the Units have a lot of movement points, but the enemy has very few, it will take on a position where it can attack and return to the out of range position of the enemy.

The third AI is the most challenging one. It uses a specific strategic approach. It uses only one type of army, 7 heavy tanks and 3 choppers. Since the heavy tank are very strong overall and only them and bazooka trooper being able to attack choppers, make them very valuable. Bazooka trooper being very poor at moving and weak at defending are not very desirable units. Choppers are very strong units because only the 2 Unit-types stated above can attack them. Third AI depends heavily on the terrain and waits for the enemy to attack it on the terrain best suited for itself.

# Closing Words

Overall the release 4 was finished without big issues. All the requirements were implemented without an issue with a couple of extra features hear and there like the guess number game and chat encryption.

The AI was implemented as planned and the bots can be created. The AI can take over the game for the player and the health points are visible. We can identify the player units and choose an army before the game starts. Many small and big bugs were created and many more resolved. There are some differences to the Mock-ups, but we think the differences are for the better.

The Game was finished to the point that it works smoothly. Player can without holding his hand start a game chat, with another player openly or in private. He can create and join the game, start it against other players or against a bot. he can let the AI take over his turn or finish the game for him, though no guarantee that it is going to win the game, since the AI must work with what it has. At a point in the game the player can stop the AI and start playing himself. After finishing the game, you return to main screen and can start a new game.

The developers poured all their available time into the game to make is good as it can be with the limited time provided. We are very proud with this wonderful game that we created together.