



# Software Engineering Project: Tower Defense

Team Members:

Fabian Braun, Luca Rutschmann, Nicolas Wagner

Period: 3. + 4. Semester

Course: TINF18B4

Lecturer: Prof. Kay Margarethe Berkling

## Inhalt

1	General Information.....	6
2	Blog Entry Week 1: Project Vision .....	7
2.1	About this Project.....	7
2.2	Subdivision.....	7
2.3	Dependencies.....	7
2.4	Comments .....	8
3	Blog Entry Week 2: Team Roles & Working With Jira .....	9
3.1	Team Roles – RUP Terminonlogy .....	9
3.2	Technology Choice.....	9
3.3	Working with Jira.....	9
3.4	Tower Defense – Progress.....	9
3.5	Game Engine – Concept .....	10
3.6	Map – Concept .....	11
3.7	Comments .....	12
4	Blog Entry Week 3: SRS and UCD .....	13
4.1	SRS and UCD .....	13
4.2	Edit (07.12.2019) – Update of UCD .....	13
4.3	Comments .....	13
5	Blog Entry Week 4: Use Case Specifications.....	15
5.1	UCS – Start the game from the menu .....	15
5.2	UCS – Start waves.....	15
5.3	Development Progress .....	15
5.4	Comments .....	16
6	Blog Entry Week 5: Narratives .....	17
6.1	Narratives .....	17
6.2	UC: Start the game from the menu .....	17
6.3	UC: Start waves .....	17
6.4	Edit (01.12.2019) – Feature File running.....	17
6.5	Comments .....	18
7	Blog Entry Week 6: Project Management Tool & Our Game Engine .....	19
7.1	JIRA – Introduction .....	19
7.2	Jira Integration in Android Studio (IDE).....	19
7.3	GIT Integration in Jira .....	20
7.4	Game Engine – General.....	20
7.5	Game Engine – “MatchField” .....	21

7.6	Game Engine – Enemy & Tower .....	21
7.7	Game Engine – “Map Structure” .....	22
7.8	Game Engine – Application .....	23
7.9	Comments .....	24
8	Blog Entry Week 7: Retrospective .....	26
8.1	Retrospective .....	26
8.2	What went well? .....	26
8.3	What can be improved? .....	26
8.4	How to improve? .....	27
8.5	Comments .....	27
9	Blog Entry Week 8: Class Diagram .....	28
9.1	Class Diagram – First Try .....	28
9.2	Class Diagram – Second Try .....	29
9.3	Class Diagram – Conclusion .....	29
9.4	Game Progress – Designs, Bullet-Animations and more! .....	30
9.5	Comments .....	30
10	Blog Entry Week 9: MVC .....	32
10.1	Our Framework .....	32
10.2	MVC – Diagram .....	32
10.3	Automatic UI Testing – Feature Files, Cucumber & Espresso .....	33
10.4	Comments .....	34
11	Blog Entry Week 10: Midterm .....	35
11.1	Midterm – Links .....	35
12	Blog Entry Week 11: Revision & New Use Case Descriptions .....	36
12.1	Revision .....	36
12.2	Descriptions of new Use Cases .....	36
12.3	Revision of Midterm Issues .....	37
13	Blog Entry Week 12: Risk Management .....	38
13.1	Our Risk Management Table .....	38
13.2	Use Case Overview .....	38
13.3	Title-Soundtrack .....	38
13.4	Comments .....	40
14	Blog Entry Week 13: Function Points / Estimation .....	41
14.1	Function Points – What’s that? .....	41
14.2	Cumulated Flow Chart .....	43
14.3	Time Estimation .....	43

14.4	Progress Gameengine.....	44
14.5	Comments .....	45
15	Blog Entry Week 14: Testing.....	46
15.1	Testing .....	46
15.2	New Feature: Choose Difficulties!.....	47
15.3	New Feature: Select between different towers!.....	47
15.4	Conclusion .....	48
15.5	Comments .....	49
16	Blog Entry Week 15: Refactoring.....	50
16.1	Fowler's Refactoring.....	50
16.2	UC Upgrade and Sell Towers .....	50
16.3	UC Edit Settings .....	51
16.4	Comments .....	51
17	Blog Entry Week 16: Design Patterns.....	53
17.1	Design Patterns .....	53
17.2	The use of design patterns in our project .....	54
17.3	New Tower: Freezer .....	55
17.4	New Tower: Boombastic .....	56
17.5	Comments .....	57
18	Blog Entry Week 17: Metrics .....	58
18.1	Metrics – what is that? .....	58
18.2	Cyclomatic Complexity .....	58
18.3	Junit Testing Metric.....	59
18.4	JavaDocs Coverage Metric.....	59
18.5	Code which is not going to be refactored .....	60
18.6	Metrics – Conclusion .....	61
18.7	Big Progress in our CI/CD.....	61
18.8	New Tower: Plasmarizer.....	62
18.9	New Tower: Assaultlaser .....	62
18.10	Summary.....	63
18.11	Comments .....	63
19	Blog Entry Week 18: Optimization .....	64
19.1	Introduction.....	64
19.2	New Enemy: Car .....	64
19.3	New Music.....	64
19.4	UX Test Preparation .....	64

19.5	Comments .....	65
20	Blog Entry Week 19: Retrospective .....	66
20.1	Retrospective.....	66
20.2	Solutions.....	66
20.3	New Enemy: Plane.....	66
20.4	New Boss Enemies.....	67
20.5	Further Adjustments .....	67
21	Blog Entry Week 20: Installation .....	68
21.1	Installation.....	68
21.2	Running through last Tests.....	68
21.3	Tech Stack.....	68
22	Blog Entry Week 21: Final.....	70
22.1	Final – Links .....	70
23	Statistics.....	72
23.1	Statistics – Introduction.....	72
23.2	Time Data .....	81
23.3	Function Points – Estimation.....	83
23.4	Function Points – Time spent .....	84
23.5	Time spent by Sprint.....	85
23.6	Time spent by Phase.....	86
24	Related Links.....	87

## 1 General Information

This document contains the documentation of the project “Tower Defense” which was worked on by Fabian Braun, Luca Rutschmann and Nicolas Wagner.

Beside all chapters, which are equal to the blog, a lot of links are located at the end of this document which relate to websites, the repository, and other important sites.

Note:

As our team made use of “pair programming/working” very often, the committed lines of code to the repository and the statistics might not be 100% accurate. Nonetheless, our team thinks that every member participated in the project significantly.

Aliases on GitHub:

GitHub Name	Real Name
Fabian1699	Fabian Braun
Charon1502	Luca Rutschmann
Niwa99 / thjg	Nicolas Wagner

## 2 Blog Entry Week 1: Project Vision

### 2.1 About this Project

Our team focuses on the development of a simple "Tower-Defense" game for Android smartphones. In this game, a map with a path, slots next to the path, a spawn-area and a goal-area are given. Enemies (e.g. tanks) spawn in the spawn-area and try to reach the goal-area by following the given path. A player has to build towers in the given slots which can fight against the enemies by shooting at them. One can win this game by defending against all enemies. In addition to this skeletal structure, it's planned to have some values and parameters such as health points for enemies and towers, a variable amount of enemies that may pass the goal, fire rate, range and strength parameters for towers and a movement speed for enemies.

### 2.2 Subdivision

Because this project will probably require a lot of effort, one has to split this project into smaller subprojects. At first appearance, our team would divide the project into following subprojects: Design; Conceptional game-engine; Interface between Design and the game engine.

Due to missing experience with the development of Android apps, this subdivision can vary when planning the project in more detail.

### 2.3 Dependencies

The app will only be applicable on Android smartphones (version 5.0+) which is why the project-team wants to make use of Android Studio as an IDE. Additionally, the following products/services/languages will be used for the development-process:

- GitHub for version-controlling
- Jira for project-management
- Java, XML, SQL as programming languages
- Photoshop/Gimp as design-tools

## 2.4 Comments

### 4 Gedanken zu „Week 1 – Project Vision“



**Max Mustermann** sagt:

6. Oktober 2019 um 21:20 Uhr

Hi,

great idea but the amount of work that will need to be done sounds frightening. Couple of questions: Are you sure yet whether you want to create the app in a 3D-format game or 2-dimensional? Maybe Gimp isn't sufficient for that. Blender might be suited as well (idk)? Heard that there are plenty of java 3d-packages, <https://www.lwjgl.org/> just to name one of them. Don't take my advice too serious, after all, I am not a game developer! Georg



**Nicolas Wagner** sagt:

7. Oktober 2019 um 0:47 Uhr

◀ Antworten

Hi Georg,

thanks for your feedback and hints!

Probably, we will develop the game in 2D, which hopefully lets us focus on the game engine even more. Nonetheless, we will keep the idea of updating to 3D in mind for the future – thanks for this hint!



**Marius** sagt:

7. Oktober 2019 um 12:12 Uhr

◀ Antworten

Hi,

I like the idea of developing a Tower-Defense-Game, since I enjoy this kind of games. Through the comment of Georg and your answer of developing the game in 2D I assume you will have the player look with a bird's eye view onto the map. Therefore I would like to give you the tip of keeping the designs simple and focus on the game engine. To do so you could stay with simple geometrical figures in various colors for the Towers and enemies and upgrade to detailed graphics if you are well in time and everything is running.

Hope this can be helpful for your Project.

Marius



**Fabian Braun** sagt:

7. Oktober 2019 um 13:46 Uhr

◀ Antworten

Hi Marius,

you are totally right, we want to do in a bird's-eye view. Thank you for the tip about simple graphics, of course, we will start with simple designs and animations at the beginning of development. We know design takes a lot of time and we won't focus on that as there is not enough time! Also simple graphics can look very fancy if you do it right 😊

◀ Antworten

### 3 Blog Entry Week 2: Team Roles & Working With Jira

#### 3.1 Team Roles – RUP Terminonlogy

According to the RUP terminology, we decided to assign the roles as following:

- Project Manager:  
Nicolas Wagner
- Scrum Master:  
Fabian Braun
- Software Architect:  
Fabian Braun
- Implementer (Game Engine):  
Fabian Braun, Luca Rutschmann
- Implementer (Surface/Overlay):  
Nicolas Wagner, Nico Ziener
- Designer:  
*will be assigned in the future when needed*
- Tester/Reviewer:  
Luca Rutschmann, Nico Ziener

[Change 18th Oct. 2019: Fabian Braun listed as "Scrum Master" instead of "Project Manager"]

#### 3.2 Technology Choice

As described in the first-week-blog-entry, our team mainly wants to make use of Java and XML with Android Studio as IDE for this project.

Already, two Jira-issues exist which are connected with our choice of technology: [DHTD-2 \(Create structure of gameengine for Tower, Enemies and MatchField\)](#)<sub>[80]</sub> & [DHTD-5 \(create structure of gameengine for the Map\)](#)<sub>[81]</sub>

#### 3.3 Working with Jira

Jira is our project-management-tool and can be found at: <http://jira.dh-towerdefense.de>

All tasks will be displayed in separated "sprints". Sprints contain a pool of tasks for a single week.

Using this system, sprints depict workflows. Phases cannot be displayed in Jira yet, but we are currently working on this problem.

Our whole issues list can be found at: <http://jira.dh-towerdefense.de/projects/DHTD/issues>

#### 3.4 Tower Defense – Progress

Within the last week, our team specified the idea of the tower-defense app.

The app structure will look like that:

Opening the app, one will see a startscreen which functions as a home menu. There, one will be able to navigate to the settings or to the game.

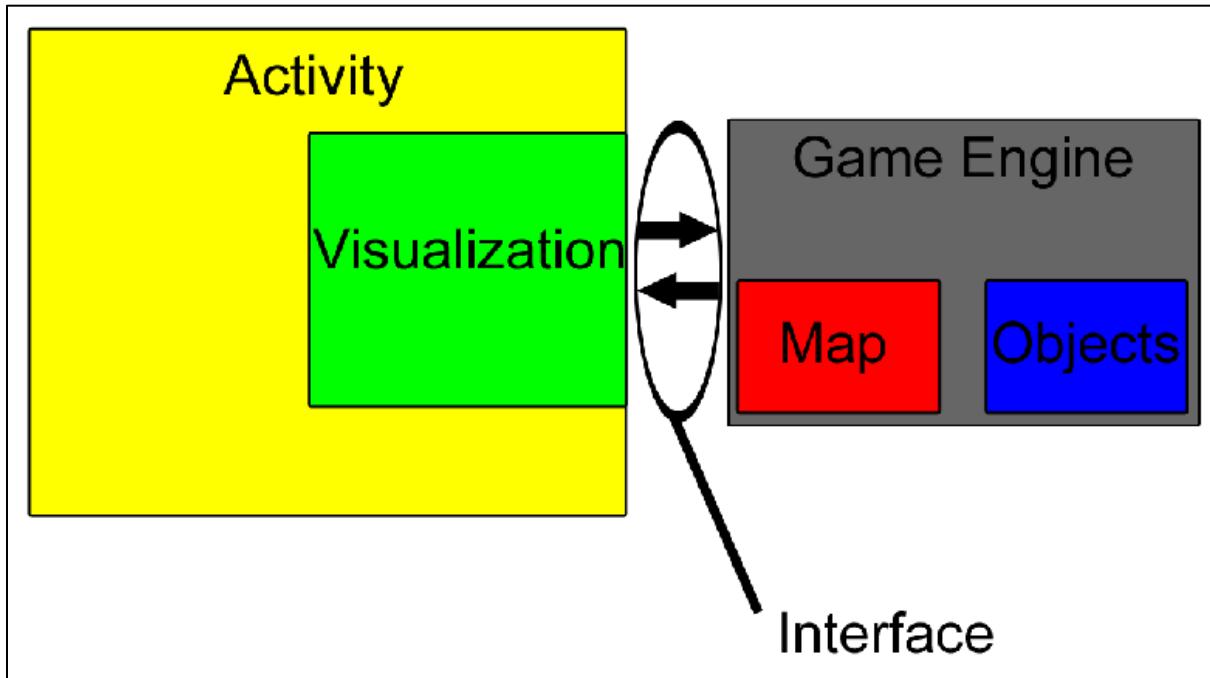
### 3.5 Game Engine – Concept

The tower defense game needs to be split up into two main parts: Game Engine and Activities.

The Game Engine module controls the map and the objects on non-graphical basis. Every interaction between objects will be simulated there.

The other main module is the activity including the visualization. This module is responsible for the visual layout on the screen, the visualization of objects and animations.

Those two modules need to interact through an interface, which receives the game data and passes the information on to the activity-module which visualizes the calculated data.



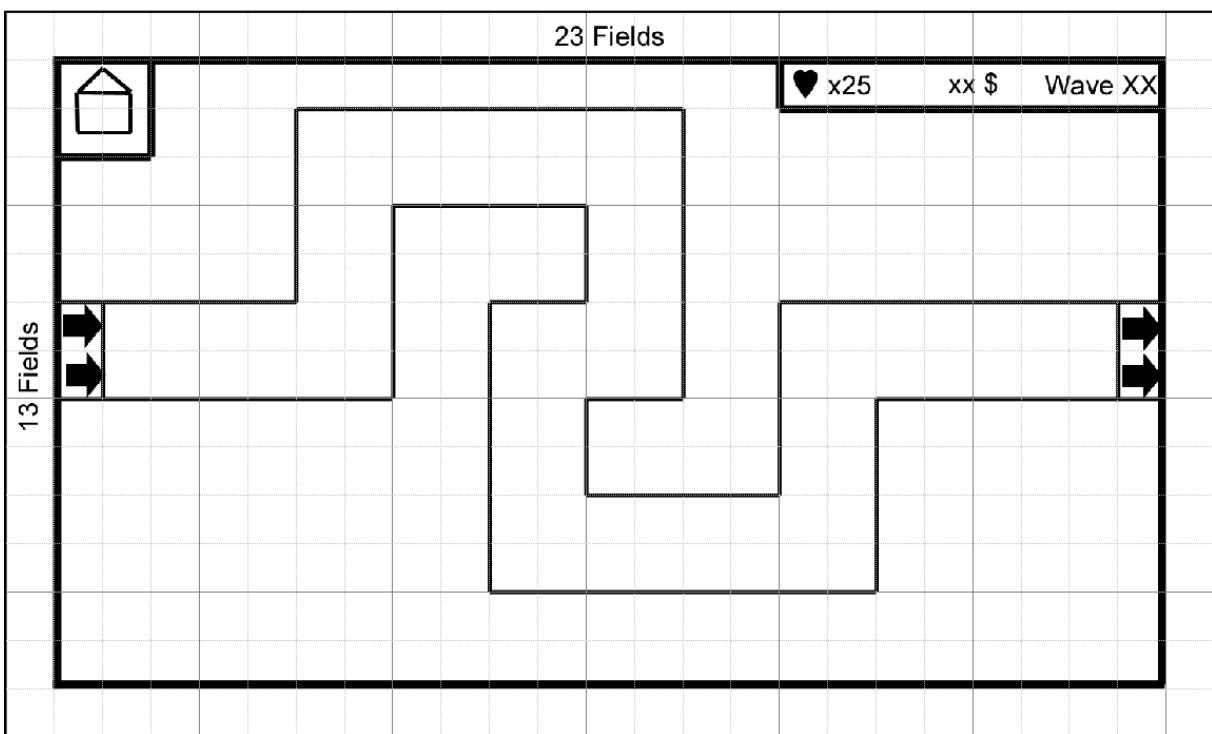
### 3.6 Map – Concept

The concept of a tower-defense map can be seen on the right.

The whole map will be divided into 299 (287) single fields.

With a button on the top left corner, a player can pause the game and navigate back to the startscreen or the settings. On the top right corner, there is information about the player-lives, the amount of money and the current wave displayed. The rest of the screen is used as play-area.

Beginning on the left side, enemies will spawn and follow the drawn way in the direction of the right side. A player is able to locate towers (which need an unused space of 2x2 fields) on the whole map except the drawn enemy-way. For the beginning development, our team will work with this map-draft as the given map, but in the future, we probably implement an algorithm that allows a random map creation for more variety.



### 3.7 Comments



**MAPHYNN** sagt:  
15. Oktober 2019 um 19:32 Uhr

Hey Tower-Defenders,  
I have to say I'm impressed about the structure and concepts you already made and I'm really looking forward on your progress. But I have to ask you how do you have planned to work with two Project Managers in a team of four? How will this work out? I am looking forward to hear from you.  
Best regards  
the MAPHYNN-Team



**Nicolas Wagner** sagt:  
17. Oktober 2019 um 8:32 Uhr

Antworten

Hi Team MAPHYNN,  
thanks a lot for your feedback and your question.  
The reason for why we decided to have two project managers is pretty simple.  
In contrast to the project of the 2nd semester, everyone has to develop some code which is why there cannot be just one person who focuses on project-management.  
Because project-tasks also require a lot of effort, Fabian takes care of the scrum-part and I assume the tasks of all project-management stuff around scrum. In fact, both of us will have enough time for developing code then.  
I hope this answers your question.  
Nicolas



**Dominik Lange** sagt:  
28. Oktober 2019 um 14:23 Uhr

Antworten

Hey Tower-Defenders,  
I really like your idea and I'm looking forward on your final result.  
Your block entry is well structured and interesting to read. You gave a clear description of your team roles and the technology you use.  
Is there any more detailed planning on your realisation of the interface? Which technology will combine your game engine with the activity?  
Could you explain your management tool to share and develop your code?  
And one more little annotation: the map concept (last paragraph) can't be seen on the right, but on the left. 😊  
Best regards and keep going!  
easyFinance-Team

Antworten

## 4 Blog Entry Week 3: SRS and UCD

### 4.1 SRS and UCD

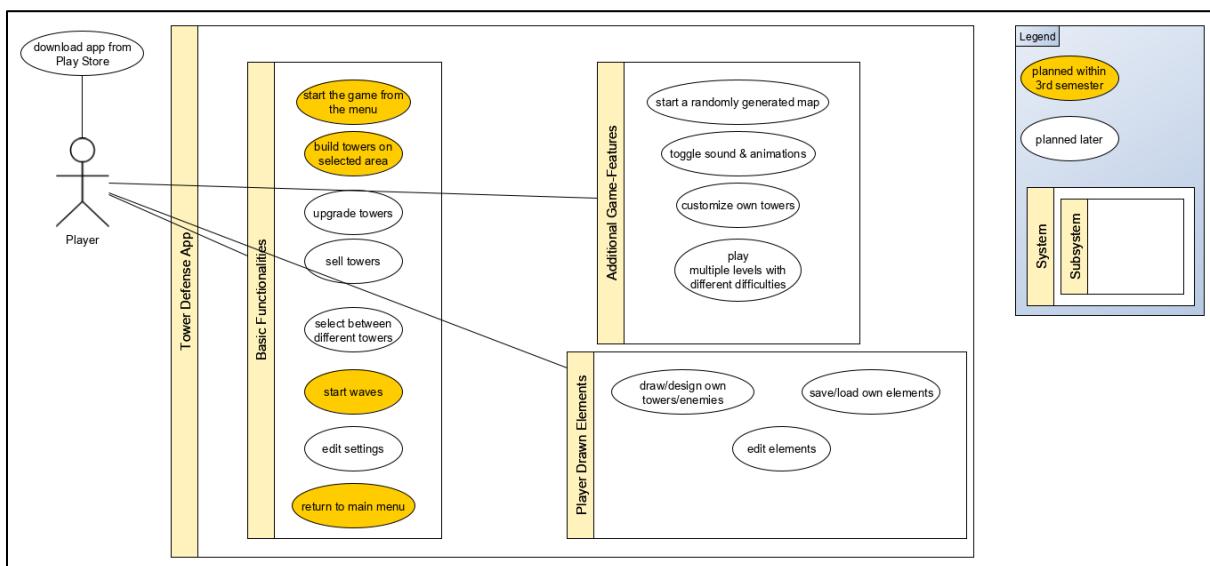
During the third week, our team managed to write the Software Requirements Specification document and to model the Use Case Diagram.

The SRS can be found here: [SRS on GitHub](#)<sup>[29]</sup>

The UCD can be found here: [UCD on GitHub](#)<sup>[79]</sup>

Please keep in mind that our SRS is not ready yet and gets completed within the next weeks.

Below, one can see our Use Case Diagram. The yellow marked Use Cases will be developed within the 3rd semester.



### 4.2 Edit (07.12.2019) – Update of UCD

As we planned our use cases in the beginning of the 3rd semester, we did not think about our use case "edit settings" in detail. Now, we realized that we do not have any adjustable settings yet which is why we removed "edit settings" of our ToDo-list for this semester. Probably, this use case will be implemented within the 4th semester.

### 4.3 Comments



nanathalie sagt:

20. Oktober 2019 um 22:58 Uhr

Hi Tower Defense Team,  
 your srs and your version control looks fine to us.  
 We like that you defined an element outside your system boundary. Your use case diagram looks precise to us. But we can not find the labels of the meaning of your colours. Maybe you want to add that and dependencies in your usecase diagram.  
 Kind regards,  
 Nathalie from Orchestra Team



Nicolas Wagner sagt:

21. Oktober 2019 um 20:58 Uhr

Antworten

Hi Nathalie,  
 thanks for your feedback!  
 You are completely right – we forgot to add a legend in the UCD. Thank you for the hint.  
 We just added the updated UCD which includes a legend on the right side.  
 Kind regards,  
 Nicolas

Antworten



**GreenClothaWay-Team** sagt:  
20. Oktober 2019 um 23:42 Uhr

Hi Tower Defense Team,

your project vision in your SRS is well elaborated, but in general your SRS could be more detailed in some places.  
Your OUCD looks fine, but we don't understand why some use cases are highlighted in colour.

Best regards,

GreenClothaWay-Team

Antworten



**Nicolas Wagner** sagt:  
21. Oktober 2019 um 21:05 Uhr

Hi Team GreenClothaWay,  
thank you for your feedback.

You are completely right – in our UCD, a legend is missing. We just added the updated UCD version which includes a legend on the right side.  
Currently, our SRS does not contain that much information because we still do not exactly know which game functionalities we are going to implement and how we will implement those. We do work on the research and hope to complete the SRS very soon.  
Kind regards,  
Nicolas

Antworten



**Georg Hertzsch** sagt:  
21. Oktober 2019 um 0:06 Uhr

Hi,

I can only add to what's been mentioned before. Your UCD is perfectly fine but giving a more details report on the interfaces-part would make thing more understandable. If it is for reasons that you just don't know how to build up your for instance user interfaces, than that would make sence but otherwise please go into more detail on that one.

„Our target group is pretty wide-ranging from teenagers up to older adults. In order to have fun playing this game, tactical thinking is required. Different levels provide a great challenge for people at any age.“  
Don't you think that is a little too broadly defined? Do you have to be the Ego Shooter guy or the strategy game boy.

Thanks for letting me giving you that advice.

Georg

Antworten



**Nicolas Wagner** sagt:  
21. Oktober 2019 um 21:20 Uhr

Hi Georg,  
thanks for your feedback!

As you assumed, we are not sure yet how to create our components in detail. This is actually the reason for why such things are not explained elaborately yet, but explanations and details will follow soon!

Concerning the target group, we chose our phrasing intentionally because our expression is actually true. One does not have to be a shooter- or a strategy-guy or a gamer at all in order to have fun playing this game. Our game (hopefully) will be designed to match many people's expectations on a „time-killing“ game. We do not want to focus on for instance a young target group – at least, it is one of our aims to reach a wide-ranging target group.

Nonetheless, your thought is definitely appropriate due to the fact that a lot of games exist which focus on specific target groups. Thank you for your nice question!

Kinds regards,  
Nicolas

Antworten

## 5 Blog Entry Week 4: Use Case Specifications

### 5.1 UCS – Start the game from the menu

Our first Use Case Specification is about starting the game from the menu page. It can be found here:

[UCS - Start the game from the menu](#)<sup>[33]</sup>

### 5.2 UCS – Start waves

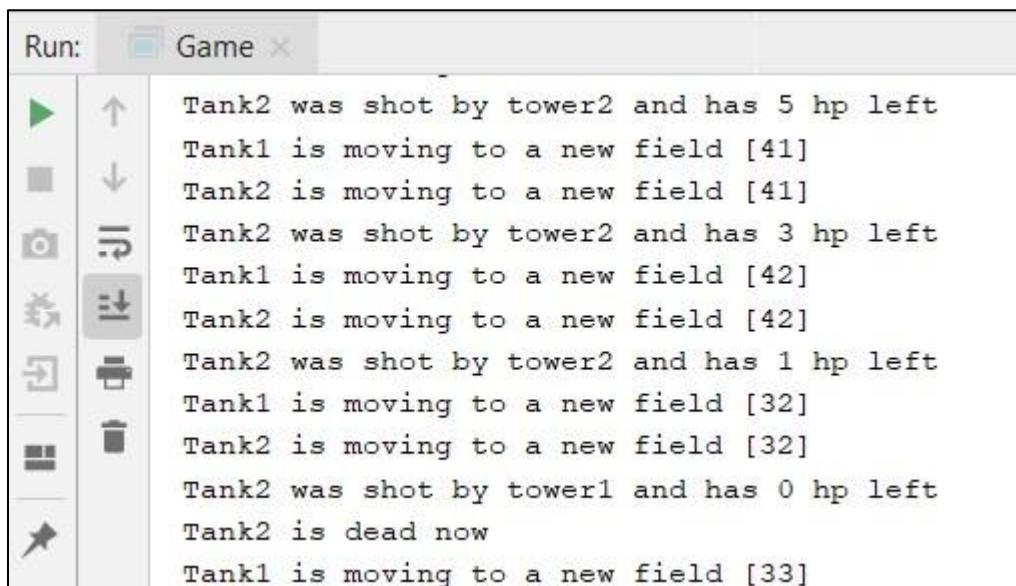
Our second Use Case Specification is about starting the waves from within the game. It can be found here: [UCS - Start waves](#)<sup>[34]</sup>

### 5.3 Development Progress

Since our last publication of our development-progress, we worked on some game-logical features.

At the moment, our game engine is able to start a game, load enemies and towers and simulate the typical procedure of a game. On the screenshot, one can see our game running with given towers and enemies as well as the defined map of week 2.

Within the next weeks, our team will focus on the UI so that objects can be represented visually.



The screenshot shows a terminal window titled "Run: Game". The window contains a list of log messages describing the actions of two tanks. The log starts with "Tank2 was shot by tower2 and has 5 hp left" and continues through several moves and shots until "Tank2 is dead now". The terminal interface includes a toolbar with icons for file operations like open, save, and delete, and a scroll bar on the right side of the log area.

```
Tank2 was shot by tower2 and has 5 hp left
Tank1 is moving to a new field [41]
Tank2 is moving to a new field [41]
Tank2 was shot by tower2 and has 3 hp left
Tank1 is moving to a new field [42]
Tank2 is moving to a new field [42]
Tank2 was shot by tower2 and has 1 hp left
Tank1 is moving to a new field [32]
Tank2 is moving to a new field [32]
Tank2 was shot by tower1 and has 0 hp left
Tank2 is dead now
Tank1 is moving to a new field [33]
```

## 5.4 Comments



Tobias sagt:  
28. Oktober 2019 um 9:47 Uhr

Hello Towerdefense Team,  
apart from some Spelling mistakes i can't find much to criticize you on.  
The handdrawn Mockup for the Start Game UseCase should be improved.  
Try to avoid declaring Topic X.1 if there is no X.2.  
It is nice that you made Progress in Development, yet the Screenshot of the console output is missing almost any context required to understand it.  
The Contents of the Specifications are done well, the Flows are coherent and are easy to understand.  
Kind regards,  
Tobias



Tobias sagt:  
28. Oktober 2019 um 12:12 Uhr

◀ Antworten

Your activity Diagram does not conform with UML standards



Fabian Braun sagt:  
29. Oktober 2019 um 18:50 Uhr

◀ Antworten

Hi Tobias,

Yes, our hand-drawn mockup is not very fancy, we will update the picture as we modelled it in Android Studio in the next weeks. As we had some progress in the backend of our gameengine we will focus now on the user interface. Maybe we can add more information about the game engine next time so that it is clear to everyone what's going on there. We will correct the UML diagram and the topics. Thanks for your constructive feedback!

◀ Antworten



Ted's Team sagt:  
30. Oktober 2019 um 11:43 Uhr

Hey Tower-Defenders,  
looks like you're on to something big, keep up that vision. We really liked your menu mockup, but maybe you should've used Google Mockups or something similar.  
Your activity diagrams and narratives diverge a little. For example, your AD shows that the „Next wave“ button is clickable after all enemies have spawned, but the narrative shows that the next wave is only beginning after none of the enemies from the previous wave are left. Maybe you can adapt that to the button only being clickable when it's actually doing something.  
Also we're wondering how the different difficulties are varying.  
Stay classy,  
Ted's Team



Nicolas Wagner sagt:  
6. November 2019 um 21:54 Uhr

◀ Antworten

Hi Ted's Team,  
thanks a lot for your constructive feedback and your suggestions to improve.  
We completely agree that our menu-mockup was not fancy yet. Recently, we re-created the mockup so that it looks more fancy now.  
Concerning the activity diagram(s), we are still working on it.  
Thanks again for your proposals!  
Kind regards,  
Nicolas

◀ Antworten



Samuel Oechsler sagt:  
6. November 2019 um 11:13 Uhr

Hi there, I am sorry having to say That I can not locate your linked use-cases. You should probably fix the links there. Once they are back online, I will have a look at them.  
Other than that your development progress seems impressive, considering that you guys are creating a game and an in-house engine.

Regards,  
Sam from Cozy



Nicolas Wagner sagt:  
6. November 2019 um 17:44 Uhr

◀ Antworten

Hi Samuel,  
I am sorry for the inconvenience concerning the links. Due to a revision of our documents on github, the links have been outdated, but now, they are now up to date again.  
Thanks for that hint!  
Within the next weeks, we will try to progress more far so that some more interesting information about the game (-development) can be published. Thanks for your interest!  
Kind regards,  
Nicolas

◀ Antworten

## 6 Blog Entry Week 5: Narratives

### 6.1 Narratives

This week, we created some narratives to our use-cases, which we specified in the last week.

Using the gherkin syntax, our narratives look like this:

### 6.2 UC: Start the game from the menu

```
Feature: Start the game from the menu

Background:
I started the application

Scenario: start a game with easy settings
Given I am on the "main" page
When I press the "start game" button
Then I am on the "choose difficulty" dropdown
And I choose "easy" by clicking on it
Then I am on the "game" page
And the map with easy settings is loaded

Scenario: start a game with medium settings
Given I am on the "main" page
When I press the "start game" button
Then I am on the "choose difficulty" dropdown
And I choose "medium" by clicking on it
Then I am on the "game" page
And the map with medium settings is loaded

Scenario: start a game with hard settings
Given I am on the "main" page
When I press the "start game" button
Then I am on the "choose difficulty" dropdown
And I choose "hard" by clicking on it
Then I am on the "game" page
And the map with hard settings is loaded
```

### 6.3 UC: Start waves

```
Feature: Start a new wave in the game

Background:
I started the application
I started the game by clicking "start game" and choosed a difficulty

Scenario: open new operation dialog
Given I am in the game activity and have a running game
When I press the spawnpoint
And The last wave completely passed the spawnpoint
And There is a wave left
Then The next wave will start
```

### 6.4 Edit (01.12.2019) – Feature File running

On 1st december 2019, we managed to be able to run feature files.

Because Tower Defense is an Android-app, we had to search for a possibility to integrate cucumber/gherkin into Android Studio.

In the beginning, we tried to run feature-file-tests the basic way using "Cucumber for Java", "Gherkin" and JUnit. After hours of research and testing, we were not able to get cucumber running the basic way because of several errors (for which not even stackoverflow had a useful solution).

Researching even more, our team got to know Espresso (Android-UI-testing) and GreenCoffee (Android library which combines cucumber and espresso). Using this library, we finally could automate our first feature file which starts the game from the menu. At the moment, we cannot implement feature-file-tests for other use cases because those are not fully developed yet.

In the following video, you can see our first successful ui-test.

[Video](#) [78]

## 6.5 Comments



@rawbean sagt:  
5. November 2019 um 11:43 Uhr

Hellas guys,

your Gherkin looks fine to me. You used syntax and layout correctly and all necessary functions are mentioned.  
So there won't be any further options than choosing the difficulty?

I would have liked some more explanation in free text.

Keep it up guys.

Cheers,

@rawbean (Foody, B4)

Antworten



Nicolas Wagner sagt:  
6. November 2019 um 22:05 Uhr

Hi rawbean,

thank you for your feedback.

You are completely right: For the first game version, we plan to implement the choice of the difficulty only.

In future versions, there might be some functionalities to choose different (custom) maps, levels and randomly created conditions. Because nobody in our team does have experience in game-development yet, we do not want to overestimate our possibilities and focus even more on realistic aims.

Some updates concerning our development-progress will follow within the next weeks.

Kind regards,

Nicolas

Antworten



Marlon sagt:  
5. November 2019 um 22:36 Uhr

Hello Team-Towerdefense,

I also liked your work of this week. You did a great job by defining your Gherkin-features. They clearly show what's important for your use cases and how they will be tested.  
Did you already start implementing the steps to run these tests? And will there be more options for a new game in the future?

Apart from the required tasks for this week. How far is your development? Last week you wrote about drawing some UI stuff. Did you have some first images by now?

Looking forward to more!

Yours, Marlon  
from SaSEp/Clairvoyance-Team

Antworten



Nicolas Wagner sagt:  
6. November 2019 um 22:15 Uhr

Hi Marlon,

thanks for praising our work!

At the moment, we are working on some common tests for our game-engine. The tests of the .feature-files are not in progress yet.

Concerning the options when starting a game, I want to cite my reply of the previous comment:

„In future versions, there might be some functionalities to choose different (custom) maps, levels and randomly created conditions. Because nobody in our team does have experience in game-development yet, we do not want to overestimate our possibilities and focus even more on realistic aims.“

Sadly, we could not achieve any further progress in development so far because of our e-portfolio presentations and a lot of other work for other courses.  
Hopefully, we will be able to create some game-designs and the implementation of those up to next Thursday.

Kind regards,  
Nicolas

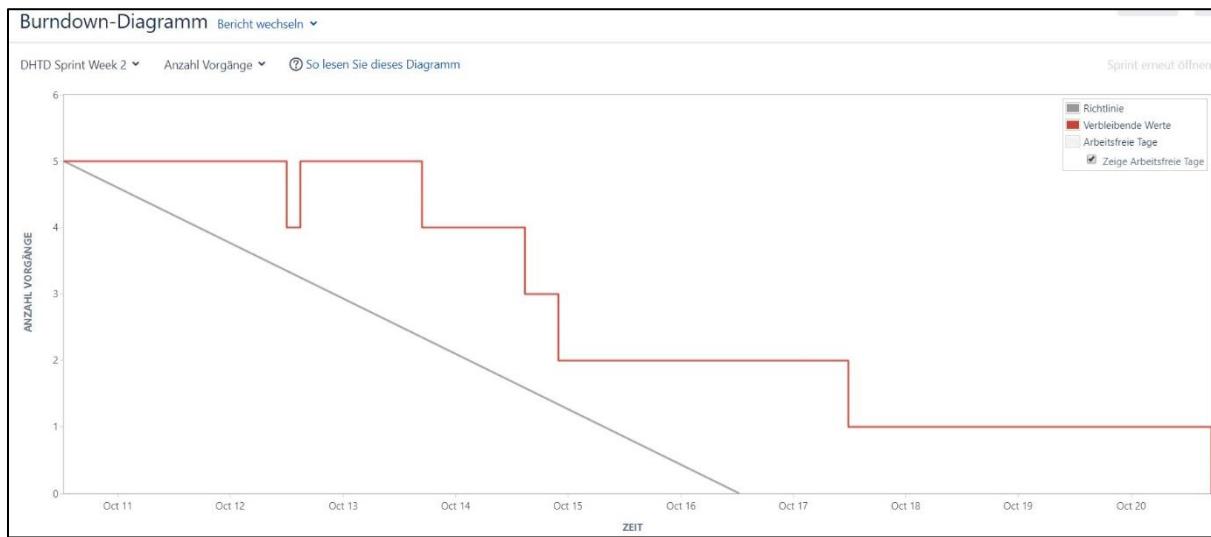
## 7 Blog Entry Week 6: Project Management Tool & Our Game Engine

### 7.1 JIRA – Introduction

We use Jira for project management which allows several Scrum-functionalities.

Our sprints last for one week each in order to fit the weekly updates on our blog and the given tasks of the lectures.

Because of initial difficulties, we had some problems using Jira in order to get a fancy burndown-chart. The burndown-chart of week 2 is displayed on the right. Because Jira does not support the public chart-generation, there is no public access to live diagrams available.



### 7.2 Jira Integration in Android Studio (IDE)

The screenshot shows the Android Studio interface with a floating window for Jira integration. The left side of the window has a search bar and a list of tasks: "Create New Task", "DHTD-19: Blog Entry Week 6", "DHTD-20: Refactor project management tool in order to match ...", "DHTD-22: Blog Entry Week 7", and "Default task". The right side shows a detailed view of the "DHTD-20" task. It includes fields for "Summary" (Refactor project management tool in order to match requirements), "Id" (DHTD-20), "Created at" (Thu Nov 07 15:51:43 CET 2019), "Updated at" (Mon Nov 11 08:11:47 CET 2019), and a code editor containing the following Java code:

```

on( x: 4, y: 2),
on( x: 3, y: 4),
on( x: 6, y: 4),
on( x: 7, y: 2),
on( x: 10, y: 2);

```

### 7.3 GIT Integration in Jira

DH-TowerDefense / DHTD-21

Hello World Demo: Visualization of simple gameengine elements

**Details**

- Typ: Implementation
- Priorität: High
- Status: IN ARBEIT (Arbeitsablauf anzeigen)
- Lösung: Nicht erledigt
- Stichwörter: Keine
- Sprint: DHTD Sprint Week 6
- Phase: Elaboration

**Personen**

- Bearbeiter: Fabian Braun
- Mir zuweisen
- Autor: Fabian Braun
- Stimmen: 0 Für Vorgang stimmen
- Beobachter: 1 Vorgang beobachten

**Beschreibung**

Fabian1699 erstellte einen Commit auf dem Branch **master** des Repositories Tower-Defense - Vor 1 Stunde  
[DHTD-21] Visualisation of Enemies, Towers and the Map

+extension for Game-Engine: class Bullet to represent a projectile

**Änderungshistorie**

- Modified app/src/androidTest/java/de/dhbw/towerdefense/TowerTest.java
- Modified app/src/main/java/de/dhbw/activities/GameActivity.java
- Modified app/src/main/java/de/dhbw/game/Game.java

**Daten**

- Erstellt: Vor 3 Tagen
- Aktualisiert: Vor 1 Stunde

**Zeitverfolgung**

- Geschätzte: 1d
- Verbleibende: 2h
- Protokolliert: 6h

**Agil**

- Aktiv Sprint: DHTD Sprint Week 6 endet 11/Nov/19
- Auf Board anzeigen

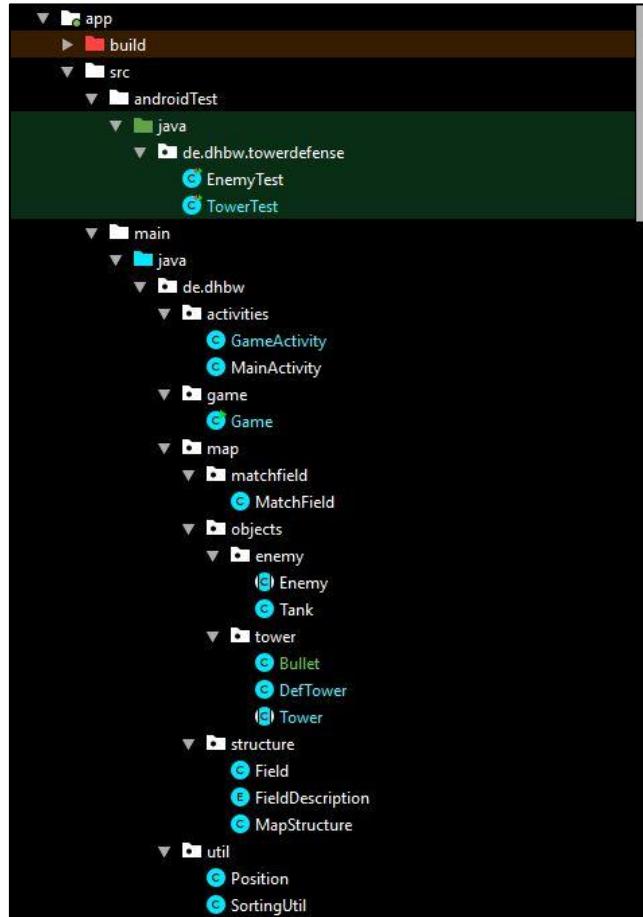
### 7.4 Game Engine – General

When we talk about the "Game Engine" we focus on the inner core of our app which is doing the logical part. This engine provides our objects at the moment without a physical extension. That will change in the next weeks, but the behavior of our main objects like towers and enemies will still be the same. To get an overview of the development just have a look at our class setup:

At the top, you can see some testing classes for JUnit. Actually, we only test the logic of the classes Enemy and Tower, because of the artificial intelligence they include.

As we described it in our process, we handle two Activities currently, the MainActivity which is the starting page with a menu and the GameActivity where the game is running at.

The 'Game' class creates a simple demo of our game which is helpful for us to see if everything is working as we expect.



## 7.5 Game Engine – “MatchField”

The first part of the engine is the class "MatchField". This class keeps all the enemies and towers and controls them. The main part is about moving the enemies and let the towers shoot them. So every Object needs a different thread to be able to act independently on the map. An example of starting the towers looks like that:

- As long as there are enemies on the map our towers will be able to fire
- after one shoot we clear the map by removing dead enemies
- if there are no enemies left on the map, the towers will be stopped (the map contains all enemies of a wave even if they are not spawned yet)
- the timer task we used there, also have to observe the fire rate of a tower

This class "MatchField" is, in conclusion, our game controller, so it also knows when the game is over and so on.

```
/*
 * all towers are shooting in an own thread. The speed is defined by the time this task is repeated.
 * int fireRate defines how much seconds the tower sleeps between two shoots
 */
public void fireTowers() {
    towers.stream().forEach(t -> {
        TimerTask task = () -> {
            if(enemies.size() > 0) {
                t.fire(enemies);
                removeDeadEnemies();
            } else {
                stopTowers();
            }
        };
        t.setTask(task);
        timer.scheduleAtFixedRate(task, delay: 1000, period: t.getFireRate() * 1000);
    });
}
```

## 7.6 Game Engine – Enemy & Tower

The abstract class Enemy contains all basic actions an enemy has to perform. So it contains the logic to move on the given path the 'MapStructure' defines and it also knows about the healthpoints, moving speed and other criteria. The move action can move the enemy one pixel on the given path, the enemy class makes sure that it's moving in the right direction. But the move action is called by the MatchField to control the speed by scheduling a timer.

More interesting is the artificial intelligence of our towers. Before a shoot, we need to calculate for every enemy which already spawned on the map the distance to our tower. Then we need to decide if the enemy is in the shooting range. If there are multiple enemies in range we focus on the nearest. In most tower defense games the focus only changes when the enemy is out of range, so towers fire multiple times on one enemy. The logic is not hard to implement but we are not quite sure if we want this behavior or not. So maybe you can give a feedback about which behavior you would expect. Currently, a tower focuses on the nearest enemy and calculates that before every shoot. The distance for all enemies is collected in a map, where we wrote our own util to sort the map by value. So we have the possibility to get the next enemy to the nearest if the first one was shoot at the same time and is already dead... So you see there is lot we had to think about.

```

public Enemy getNearestEnemy(List<Enemy> enemies) {
    if(!enemies.isEmpty()) {
        Map<Enemy, Integer> distanceToEnemy = new HashMap<~>();
        for (Enemy enemy : enemies) {
            int distance = getDistance(enemy.getPositionX(), enemy.getPositionY());
            distanceToEnemy.put(enemy, distance);
        }
        List<Enemy> sortedByDistance = sortingUtil.getSortedListBySortingMapByValue(distanceToEnemy);
        for (Enemy enemy : sortedByDistance) {
            if (enemy.isAlive()) {
                return enemy;
            }
        }
    }
    return null;
}

```

## 7.7 Game Engine – “Map Structure”

Another big part is our MapStructure. We need to create a map with lots of fields. Every field has one of the conditions: free, path or tower. So a tower can only be placed on a free field, but then the field is set to tower. A field can also be free again by deleting a tower. But the path can not be changed while the game is running. At the moment we set a fixed path by defining different positions at the beginning, where the fields should be a part of the path. But the map Structure does also include more logic. It is the representation of the map and has methods like getNextFieldForEnemyToMove() and so on.

Creating a random path will be very easy, with the foundation we have here. But in the beginning, we won't focus on that.

We don't want to bore you with some more detailed information yet. There is some more stuff like the Bullet class which represents the projectiles a tower can shoot, but enough is enough.

```

// sets the FieldDescription for the Map (says which Field belongs to the Path and which not)
private FieldDescription[][] createPath() {
    // There is only one fix Path with the following Coordinates
    FieldDescription[][] fieldDescription = new FieldDescription[columns][rows];
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 6; j++) {
            for (Position pos : path) {
                if(pos.equals(new Position(i,j))) {
                    fieldDescription[i][j] = FieldDescription.PATH;
                    break;
                } else {
                    fieldDescription[i][j] = FieldDescription.FREE;
                }
            }
        }
    }
    return fieldDescription;
}

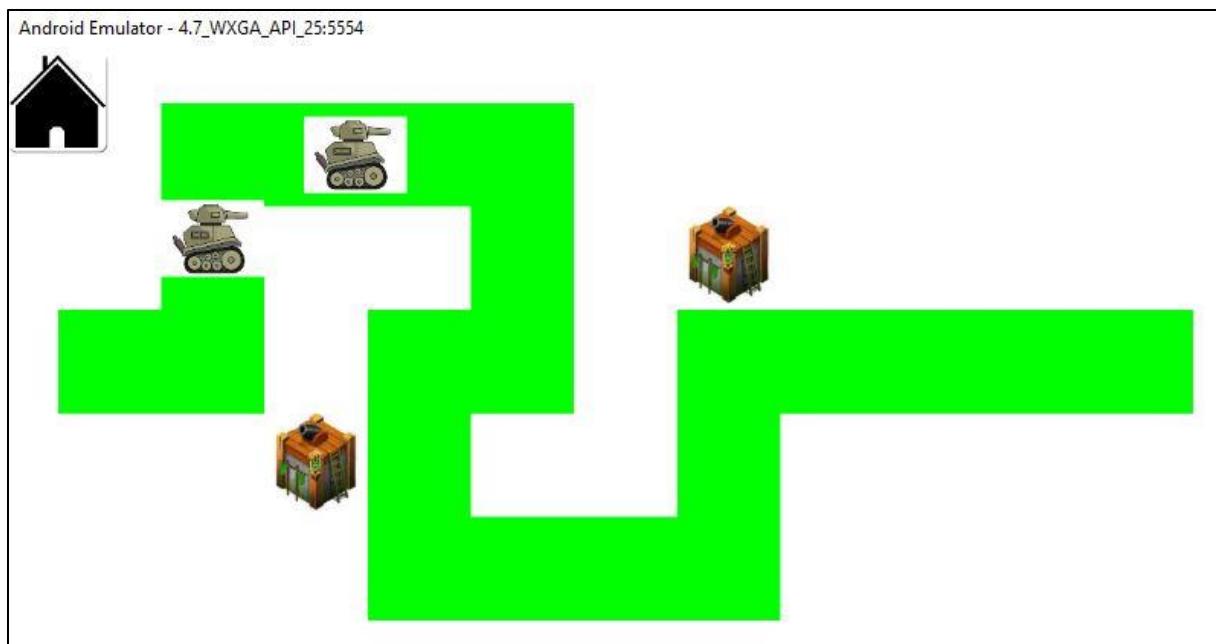
```

## 7.8 Game Engine – Application

The current result of our game-engine is only textbased in the console. We saw that before in week 4 as a preview, now you know a bit more how it works. The game is already running like intended. So enemies move, towers shoot and also the game stops if all enemies passed the target or died before.

```
Run: Game ×
▶ ↑ Tank2 was shot by tower2 and has 5 hp left
■ ↓ Tank1 is moving to a new field [41]
📸 ⏪ Tank2 is moving to a new field [41]
Tank2 was shot by tower2 and has 3 hp left
📸 ⏪ Tank1 is moving to a new field [42]
Tank2 is moving to a new field [42]
📸 ⏪ Tank2 was shot by tower2 and has 1 hp left
📸 ⏪ Tank1 is moving to a new field [32]
📸 ⏪ Tank2 is moving to a new field [32]
📸 ⏪ Tank2 was shot by tower1 and has 0 hp left
📸 ⏪ Tank2 is dead now
📸 ⏪ Tank1 is moving to a new field [33]
```

Next step will be to visualize our objects and just to give you a little preview, yes we are already working on that! 😊



## 7.9 Comments



**Simon** sagt:  
14. November 2019 um 16:08 Uhr

Hello Towerdefense Team,  
it is really cool to see a burndown chart of your sprint up front, also I see that you have managed the Integration into the IDE too.  
It would be cool to see how much time each individual of you have spent on which category though.  
Maybe you have implemented it but not showed it, it would be nice to know that.  
All in all you did very well and you should keep up the good work!

Sincerely

Manu & Simon  
Gyrogame Team



**Nicolas Wagner** sagt:  
15. November 2019 um 12:08 Uhr

Antworten

Hi Simon,  
thanks for your feedback!  
As far as we know, our project-management-tool Jira sadly does not support any chart-functionality in order to show the time of each team-member spent on certain categories/workflows (even not using external plugins).  
In return, we are able to generate charts which show the whole time of each member spent on all tasks and charts which demonstrate the total amounts of tickets for each category.  
We will keep our search for a suitable chart-plugin on in order to have more possibilities in the future.

Kind regards,  
Nicolas

Antworten



**Marlon** sagt:  
12. November 2019 um 14:57 Uhr

Hello guys,  
great to hear about your progress.  
The part about your scrum workflow appears a bit short. But I also know that there isn't very much to write about and your screenshots speak for themselves. So I assume its OK.  
Thank you for giving an introduction to your game-engine. Now it becomes more clear what you are doing. About your question on which enemy to focus: I would also expect the tower to focus on the nearest enemy, like you already implemented it.  
But I have a question about your implementation. Why did you implement it that way? It looks like you are creating some overhead by doing it this way, because of the extra lists, maps and loops you are using. I would adapt the fire method the following way:

```
public boolean fire(List enemies) {  
    Optional<Enemy> nearestEnemy = enemies.stream()  
.filter(this::isEnemyInRange)  
.filter(Enemy::isAlive)  
.min(this::compareEnemyDistance);  
if (nearestEnemy.isPresent()) {  
    Enemy enemy = nearestEnemy.get();  
    enemy.reduceHealthPoints(damage);  
    System.out.println(enemy.getLabel() + " was shot by " + label + " and has " + enemy.getHealthPoints() + " hp left");  
    return true;  
}  
return false;  
}  
  
private int compareEnemyDistance(Enemy e1, Enemy e2) {  
    return Integer.compare(getDistance(e1.getPositionX(), e1.getPositionY()),  
getDistance(e2.getPositionX(), e2.getPositionY()));  
}
```

This is only a suggestion. I suppose this would be more lightweight and performant especially when there are more enemies in the game at a later stage. Also it looks more readable to me, but this can also be a personal preference. As this is your project I don't want to influence you too much, just show my solution. If you already have some good reasons for doing it the other way, you can ignore this suggestion.

Keep on working!

Yours, Marlon  
from SaSEp/Clairvoyance-Team

**Nicolas Wagner** sagt:

13. November 2019 um 23:32 Uhr

Antworten

Hi Marlon,  
we are amazed – thanks a lot for your extensive feedback and suggestion!

In our team, we already discussed about your solution for the fire-method. I want to demonstrate our considerations:  
Your solution is probably more performant, just like you said.

At the moment, your code focuses on just one enemy which is nearest to a tower. For the current situation, of course, your solution would work out.

In the future, there will be towers that support shooting multiple enemies (we did not write something about this feature yet). In this case, your code presumably would not work out anymore because of the focus of just one enemy. Using our (current) solution, we could probably support towers shooting at multiple enemies more easily.  
This is why this part is implemented this way.

Nonetheless, we will test your solution/way of thinking on performance soon because your idea seems very useful at least at the moment.  
We will keep you up to date in the following blog entries.

Oh and don't worry, you don't influence „too much“. We really appreciate every new idea because it improves the whole game in the end. 😊

Kind regards,  
Nicolas

**Marlon** sagt:

14. November 2019 um 10:54 Uhr

Antworten

Hi Nicolas,

if you want to target multiple enemies, you could simply replace the `.min()` function with `.sorted(this::compareEnemyDistance).limit(x)`. Then you will get a list of targets to fire on or use the `forEach()`-function of the stream directly to execute the fire-code .

Yours,  
Marlon

**Fabian Braun** sagt:

14. November 2019 um 12:06 Uhr

Antworten

Hey Marlon,

I tested the two different implementations with some unit tests and with one million enemies in range, yours took about 0.2 seconds while our implementation needed 4.5 seconds. So you were totally right, thanks for the help! The performance won't make a big difference as we don't have more than 20 Enemies on the map in one wave, but it is nice to read and in both cases the better implementation.

Kind regards,  
Fabian

Antworten

## 8 Blog Entry Week 7: Retrospective

### 8.1 Retrospective

In the 7th week, we had a retrospective supported by an external coach.

There are three main questions which we had to answer:

1. What went well?
2. What can be improved?
3. How can certain things be improved?

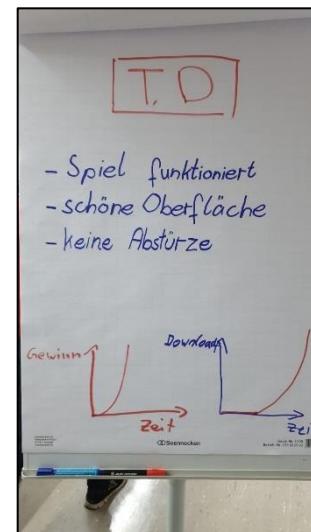
In the Software-Engineering lecture, those questions become answered by the use of Flip-Charts.

#### 8.2 What went well?

Sadly, we misunderstood this question which is why the content on the Flip-Chart is incorrect. Thinking about the questions, we can conclude that the following aspects went well:

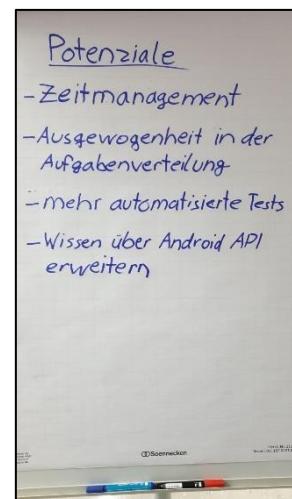
- Beginning of the development-process
- All tasks are done
- Successfully working with Git, Jira and Android
- Most often detailed blog entries

For the future we hope to create a working game with a nice design. Of course, there should not occur any exception. As a result, the game should provide a lot of fun so that the game gets downloaded by many users later on.



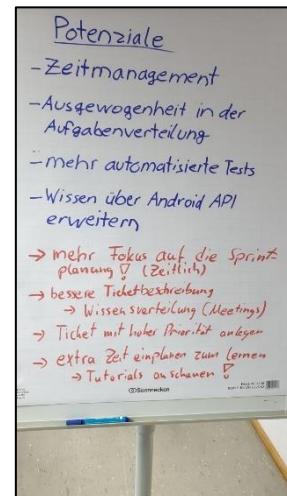
#### 8.3 What can be improved?

- Time-Management
- Balance concerning distribution of tasks
- more automatic testing
- Extend knowledge concerning the Android API



## 8.4 How to improve?

- Focus more on the sprint-planning in order to achieve better time-management
- Improve the ticket description so that every task is clearly defined and can be done by everyone
  - Distribution of knowledge in periodical meetings
- Create unit-testing-tickets with high priority
- Plan some extra time for the learning process, e.g. watching tutorials



## 8.5 Comments



**Simon** sagt:

14. November 2019 um 16:20 Uhr

Hello Towerdefense Team,

It is good to hear that you have understood your potential and managed to build a plan from that. We hope that everything you worked out can be implemented without any hassle. And even when you misunderstood a question, you have clearly defined goals that seem possible to reach. Good luck on your journey!

Regards,

Manu & Simon  
Gyrogame Team

Antworten



**Team Itemize** sagt:

17. November 2019 um 20:49 Uhr

Heya Team Towerdefense,

Even though you had a misunderstanding at first, you seem to have thought a lot about what went well at first. It's nice to see that your project itself works properly. Also the solutions you thought of for your current problems seem good and doable.

Good luck for the next time!

Team Itemize

Antworten

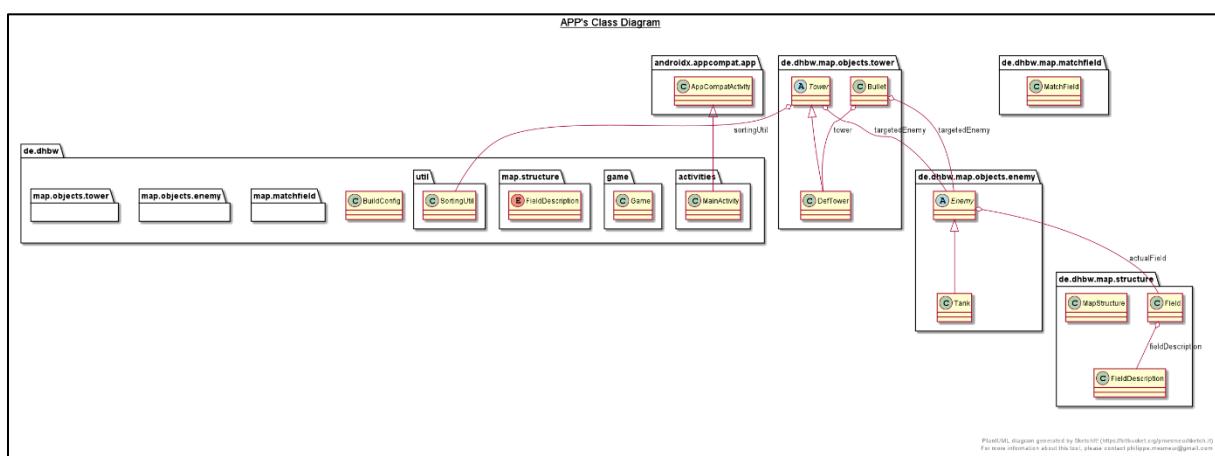
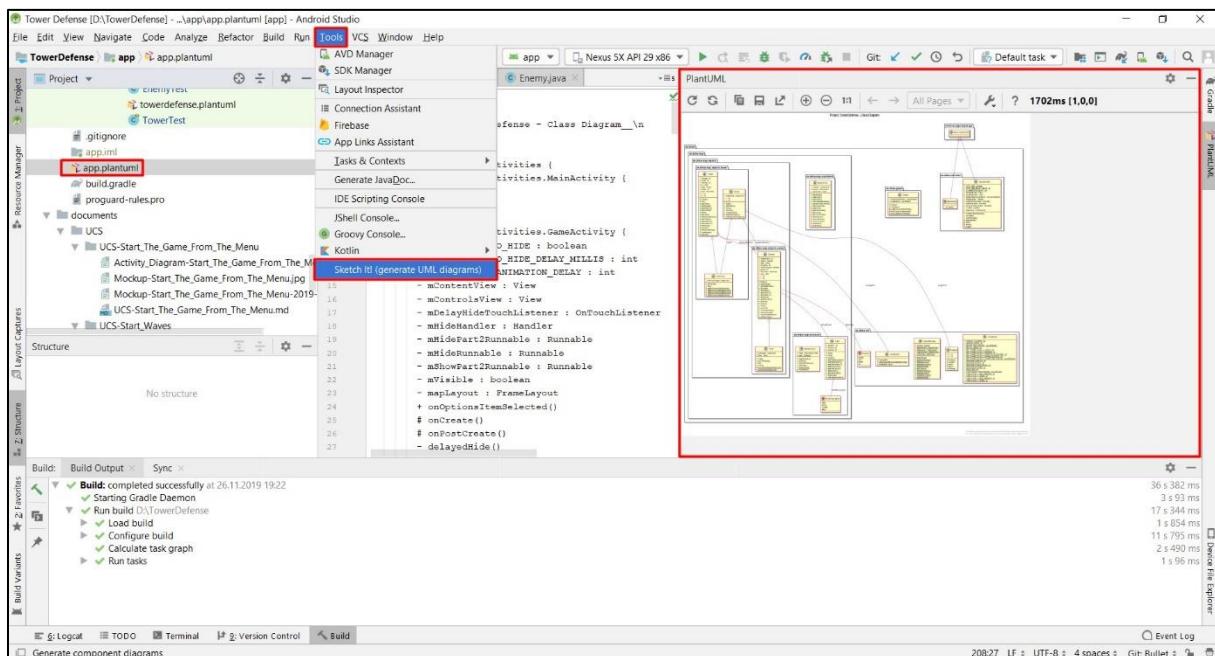
## 9 Blog Entry Week 8: Class Diagram

### 9.1 Class Diagram – First Try

This time, we want to present our class diagram of our current project.

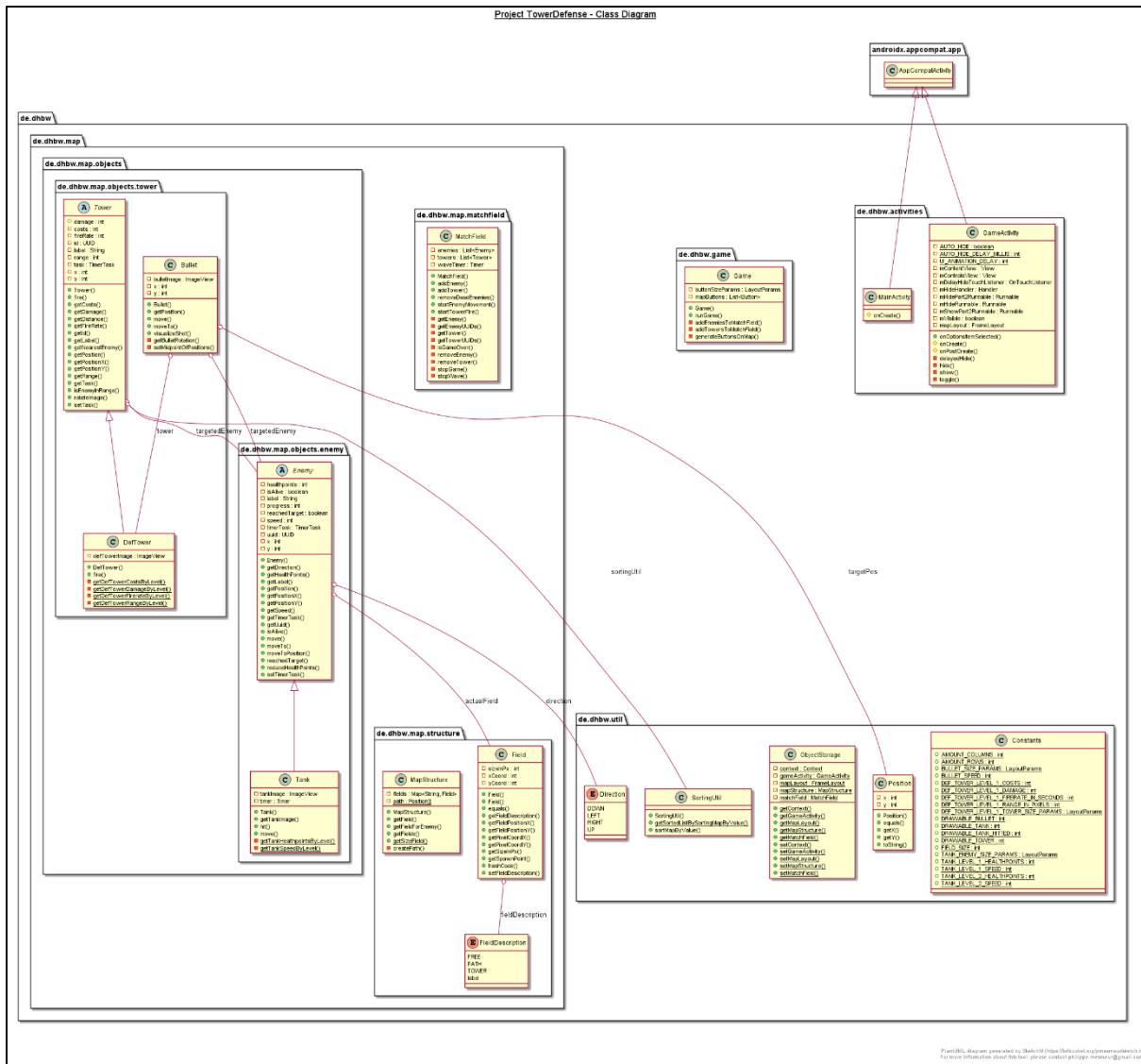
Because Android Studio does not have any tool which generates a class diagram itself, we had to search for an external tool or at least a tool which is integrable into Android Studio. Eventually, there is a tool called "Sketch It!" which should provide a class diagram generation functionality with the use of "plantuml" (also an Android Studio plugin which provides a view in which the diagram will be displayed) and "GraphViz" (which models the whole diagram).

As we simply generated our class diagram clicking on a button, the following result appeared:



## 9.2 Class Diagram – Second Try

Obviously, this is not the result which we did expect. Because of that, we had to adjust the whole generated class-diagram file, which took about one more hour. Finally, we managed to get a result as following:



## 9.3 Class Diagram – Conclusion

Concluding, we are really happy with our adjusted version of the class diagram. It clearly shows the scope and the dependencies of our classes.

If anyone has the same problem generating the class diagram out of Android Studio using the mentioned tools, fell free to contact us.

## 9.4 Game Progress – Designs, Bullet-Animations and more!

Our team is really happy to be able to present a big progress since the last update on the blog.

On the one hand, three designs have been created: tank, tower & map-background.

On the other hand, some animation-functionalities have been added:

- tanks turn themselves when changing the direction
- towers focus on the nearest enemy while shooting
- bullets have been added which are fired by a tower

The bullets are not self-designed yet.

Until now, a tank got instantly hit when a tower performed its "fire"-method. With the new bullet functionality, a tower is able to transfer its information (e.g. target position, damage, ...) onto a bullet which gets fired after transferring the information. After that, the bullet calculates the path to its target and moves into the target-direction. Only once the bullet reached its target, the tank is damaged, which improves the logical game-mechanism.

Moreover, our team achieved a structure which allows us to assign specific level-parameters to tanks and towers, which will be important for the future.

Here is a first impression of our current game-prototype:

[Video](#)[77]

## 9.5 Comments



Arerrac sagt:

26. November 2019 um 16:53 Uhr

Hi Team Towerdefense,

as my predecessor already mentioned, your progress is looking really auspicious.  
Keep up the good work!

We're looking forward for further objects and their animations.

Sincerely yours  
Your Arrerac Team

Antworten

Nicolas Wagner sagt:

26. November 2019 um 20:08 Uhr

Hi Team Arrerac,

thanks for praising our progress!  
We will keep our work on :)!

Kind regards,  
Nicolas

Antworten



**Bookly** sagt:  
25. November 2019 um 10:06 Uhr

Hi Towerdefense-Team,

your class diagram looks great and definitely generated. But can you add a screenshot of how you generated it? (In the GC it might be needed .Not drawn by hand but generated with tool, Proof of above (screenshot or tool list)'

Do you have a database by the way (for users etc.)? If so, your DB scheme is missing. If not: job well done 😊

Best wishes,

bookly, B4

↪ Antworten



**Nicolas Wagner** sagt:  
26. November 2019 um 20:02 Uhr

Hi Team Bookly,

thank you for your positive feedback and the hint concerning the GC as well.

You are completely right: We forgot about proofing the use of tools with screenshots. Recently, I added the screenshot which shows our Sketch-It! tool, the generated plantUML-files and of course the integrated class diagram-animation.

Up to now, we do not have any database yet because we want to focus on the game itself first. Sadly, it is too early to focus on some nice rankings or multiplayer functionalities 😊

Kind regards,  
Nicolas

↪ Antworten



**Cozy** sagt:  
25. November 2019 um 13:45 Uhr

Hi Team Towerdefense,

Your progress so far looks promising.

The Class Diagram is well structured and detailed. Good job !

As a minor complaint i would love to see a legend, as many people propably dont know what PlantUMLs shapes and letters mean

Keep going!  
Hannes, Team Cozy

↪ Antworten



**Nicolas Wagner** sagt:  
26. November 2019 um 20:06 Uhr

Hi Hannes,

thanks for your praise!

Concerning the suggested legend:

Do you think it is necessary to explain each plantUML component used in our diagram?

Actually, plantUML only makes use of common UML-standards which should be readable for everyone who knows UML.

Hopefully, this explanation could satify your complaint. If that's not the case, please let us know!

Kind regards,  
Nicolas

↪ Antworten

## 10 Blog Entry Week 9: MVC

### 10.1 Our Framework

Since Android is our only chosen framework and databases are not used for Tower-Defense, CRUD is not applicable for our project.

Android itself provides a simple use of CRUD-technologies though.

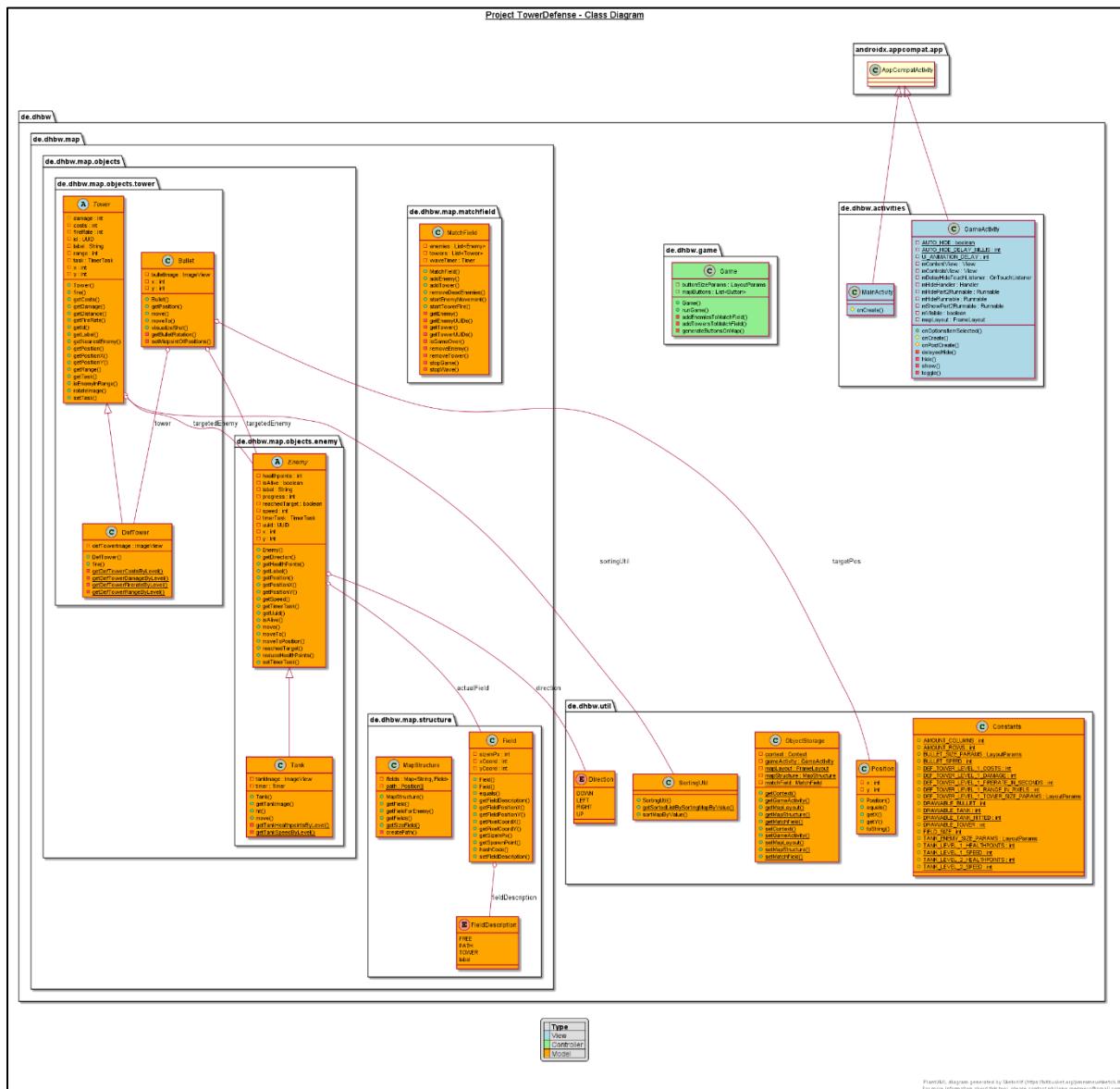
### 10.2 MVC – Diagram

In this week, we worked on subdividing our class diagram into Model-, View- and Controller-sections in order to match the MVC model.

The subdivision happened using plantUML again which provides a functionality to colorize classes. We decided on orange for model, blue for view and green for controller.

For this purpose, a SAD (Software Architecture Document) has been created and written which can be found [here](#)<sup>[30]</sup>.

According to the basic [Android-MVC-model](#)<sup>[72]</sup>, we subdivided our classes as following.



### 10.3 Automatic UI Testing – Feature Files, Cucumber & Espresso

After many hours of researching and testing, we finally got UI-tests using Gherkin Feature Files, Espresso, Cucumber and GreenCoffee running.

Because Android is our used framework, we installed plugins called "[Cucumber for Java](#)"<sup>[73]</sup> and "[Gherkin](#)"<sup>[74]</sup> in order to keep UI-tests as simple as possible. Sadly, UI-tests could not be set up using this simple configuration, because several errors (java.lang.IllegalStateException: No instrumentation registered! Must run under a registering instrumentation & cucumber.runtime.CucumberException: Failed to instantiate class & java.lang.NoClassDefFoundError: gherkin/formatter/Formatter Caused by: java.lang.ClassNotFoundException: gherkin.formatter.Formatter) occurred.

Probably, those issues are related to new android libraries which do not interact well with cucumber anymore.

Researching even more, a solution using [Espresso](#)<sup>[75]</sup> (Android UI testing) and [GreenCoffee](#)<sup>[76]</sup> (Combination of Cucumber & Espresso) seemed pretty sensible because of its topicality.

In the blog entry of week 5, working with those libraries is described in more detail (including a demo video).

## 10.4 Comments



**Itemize** sagt:  
2. Dezember 2019 um 10:02 Uhr

Heya Team Towerdefense,

after looking over your SAD we must say that everything looks well done, nothing missing.  
We're only missing the answer for if your framework is able to create simple CRUDs.  
The detailed section for your tests is also pretty nice, keep up the good work!

Team Itemize

Antworten



**Nicolas Wagner** sagt:  
2. Dezember 2019 um 17:15 Uhr

Hi Team Itemize,  
thanks for your feedback and the helpful hint – we forgot about mentioning our framework.  
Recently, a new box/section called „Our Framework“ has been added which includes our description concerning CRUD.  
Hopefully, this answers your question.

Kind regards,  
Nicolas

Antworten



**Manuel** sagt:  
2. Dezember 2019 um 10:51 Uhr

Hey Team Towerdefense,

I miss a DB model in your SAD, otherwise it looks good for me.  
Otherwise I can only join Itemize.  
I really liked your blog, keep it up.

Best regards,  
Manuel@Orchestra

Antworten



**Nicolas Wagner** sagt:  
2. Dezember 2019 um 18:03 Uhr

Hi Manuel,  
thank you for your feedback!  
For the Tower-Defense game, we did not plan to use a database which is why there is not any data-structure existing in our project-files.

Kind regards,  
Nicolas

Antworten



**Anonymous** sagt:  
5. Dezember 2019 um 10:50 Uhr

Hi there,

your SAD seems up to standards and features all the necessary parts needed. Since you do not have any outbound connections with external service the database model mentioned above (previous comment) is probably non existent, which is fine. The MVC structure is also there and explained well.

Regards  
– Sam

## 11 Blog Entry Week 10: Midterm

### 11.1 Midterm – Links

Today, our team wants to present the progress achieved in the 3rd semester to you. Within 10 weeks, the following work has been done.

Blog Entries:

- [Week 1 - Vision, Dependencies](#)<sup>[4]</sup>
- [Week 2 - RUP Team Roles, Technology, First Development-Progress](#)<sup>[5]</sup>
- [Week 3 - SRS, UCD](#)<sup>[6]</sup>
- [Week 4 - Use Case Specifications, Spoiler of Game Engine](#)<sup>[7]</sup>
- [Week 5 - Feature Files \(& Cucumber Test\)](#)<sup>[8]</sup>
- [Week 6 - Project Management Tool \(Jira\), Insights into our Game Engine](#)<sup>[9]</sup>
- [Week 7 – Retrospective](#)<sup>[10]</sup>
- [Week 8 - Class Diagram, Game-Designs/-Animations/-Prototype](#)<sup>[11]</sup>
- [Week 9 - MVC Model](#)<sup>[12]</sup>
- [Week 10 - Midterm Overview](#)<sup>[13]</sup>

Additional Links:

- [Jira](#)<sup>[2]</sup>
- [GitHub](#)<sup>[3]</sup>
- [Time Statistics](#)<sup>[27]</sup>
- [Excel Sheet for Time Statistics](#)<sup>[28]</sup>
- [SRS \(Software Requirement Specification\)](#)<sup>[29]</sup>
- [SAD \(Software Architecture Document\)](#)<sup>[30]</sup>
- [UCD \(Use Case Diagram\) \[Actually, the first two Use Cases could have been split into smaller Use Cases which is why there are only 4 UCS right now.\]](#)<sup>[68]</sup>
- [UCS1 - "Start the game from the menu"](#)<sup>[33]</sup>
- [UCS2 - "Start waves"](#)<sup>[34]</sup>
- [UCS3 - "Build towers on selected area"](#)<sup>[35]</sup>
- [UCS4 - "Return to main menu"](#)<sup>[36]</sup>
- [.feature File UCS1 \(Start the game from the menu\)](#)<sup>[43]</sup>
- [.feature File UCS2 \(Start waves\)](#)<sup>[44]</sup>
- [Demo/Prototype \(on appetize.io\)](#)<sup>[69]</sup>
- [Midterm Presentation – Handout](#)<sup>[70]</sup>
- [Midterm Presentation – Slides](#)<sup>[71]</sup>

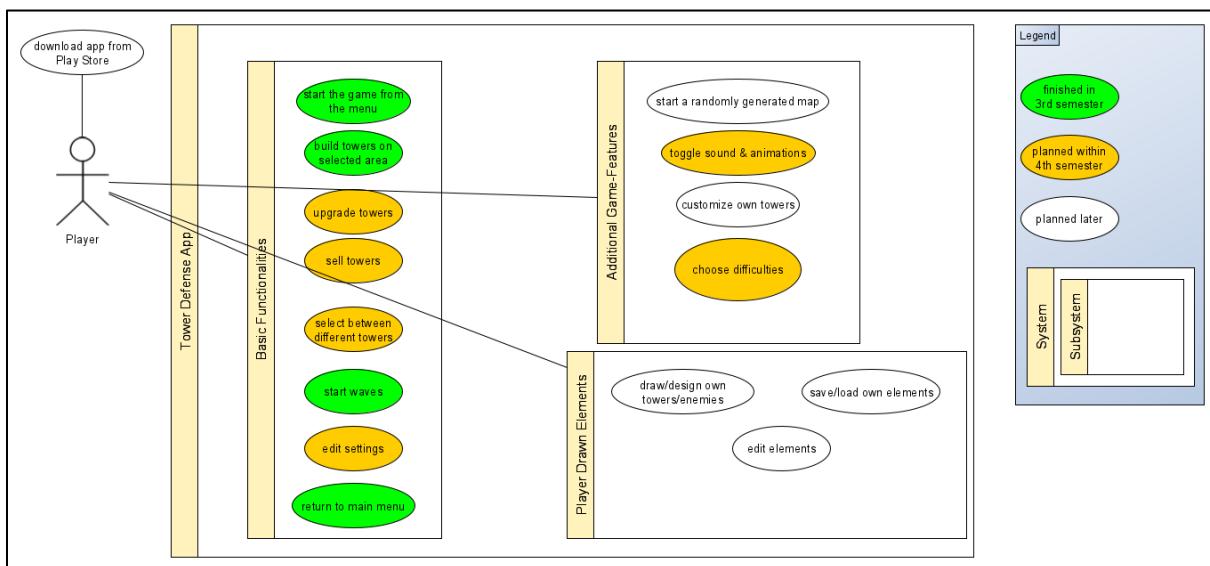
## 12 Blog Entry Week 11: Revision & New Use Case Descriptions

### 12.1 Revision

Welcome back to our blog!

So far, we are still happy with our project-idea and therefore, our team wants to continue with the initially idea of the project.

In fact, this means that this project still will be about creating a Tower Defense game starting without any high-level framework and based on Android. There is no reason for changing our choice of technology and as a result, we can present the following Use Case Diagram (and our scope for this semester) to you. There is only one small change: our Use-Case "play multiple levels with different difficulties" changed to "choose difficulties".



### 12.2 Descriptions of new Use Cases

Of course, the Use Cases mentioned above have to be specified. In the following, you can find the links to our new Use Case descriptions (GitHub), which are planned to be implemented within this semester.

- [Upgrade Towers<sup>\[37\]</sup>](#)
- [Sell Towers<sup>\[38\]</sup>](#)
- [Select between different Towers<sup>\[39\]</sup>](#)
- [Edit Settings<sup>\[40\]</sup>](#)
- [Toggle Sound & Animations<sup>\[41\]</sup>](#)
- [Choose Difficulties<sup>\[42\]</sup>](#)

### 12.3 Revision of Midterm Issues

Moreover, there are some issues which have been mentioned in our midterm-feedback. Below, there is a list of the issues and our problem-solutions for each issue.

1. ***The Use-Cases do not directly link to their corresponding feature-files.***

The Uses Cases "[Start the game from the menu](#)"<sup>[33]</sup> and "[Start waves](#)"<sup>[34]</sup> now contain the links to their corresponding feature file.

2. ***Some links on the midterm-blog-entry do not work.***

The corrupted link on the blog-entry to the UC "Build Towers" is working now. Moreover, the SRS now contains the links to the UCs "Build towers on selected area" and "Return to main menu".

3. ***The film of seeing feature-files working could not be found.***

The film of the running feature-files is located in our [blog entry of week 5](#)<sup>[8]</sup> or directly following this link: [Link to the film](#)<sup>[60]</sup>

4. ***The activity diagrams have not been created according to UML.***

Now, all four activity diagrams are refactored according to UML ([Diagram1](#)<sup>[61]</sup>, [Diagram2](#)<sup>[62]</sup>, [Diagram3](#)<sup>[63]</sup>, [Diagram4](#)<sup>[64]</sup>).

5. ***A RUP-gantt-chart is missing.***

Sadly, Jira does not provide any gantt-report-functionality and free external plugins neither.

6. ***Time-charts (per UC and per phase) are missing.***

The mentioned time-charts can be generated automatically by anyone by following this link: [Link to the report-section](#).<sup>[65]</sup>

Being there, one has to choose the value "Time spent" in the field "Time field to report on". In the field above ("Statistiktyp"/Stats-type), there are several possibilities given in order to get time-stats. For example, one can select "Bearbeiter", "Use Case", "Phase", "Vorgangstyp" or "Sprint" there.

IMPORTANT NOTE: The generated time statistics by Jira do not count the accurate time of each team member spent. Because of that, we simultaneously maintain an excel-sheet which precisely counts our time spent.

Of course, all those important charts will be included in the final written-hand-in.

As an example, here are the pie-charts "[Time spent by Use Case](#)"<sup>[66]</sup> and "[Time spent by Phase](#)"<sup>[67]</sup> of 24th of April 2020.

7. ***The Sprint Names should be more specific and contain a catchphrase.***

The sprint names are now refactored according to the given convention.

## 13 Blog Entry Week 12: Risk Management

### 13.1 Our Risk Management Table

ID	Risk Name	Risk Description	Risk Probability of Occurrence (1-3)	Risk Impact (1-3)	Risk Factor	Risk Mitigation	Person in Charge of Tracking	Number-Definitions: 1 = low 2 = medium 3 = high
1	Performance problems	As we do not have a lot of experience in developing Android games, there might be performance issues with our implementation	2	3	6	provide the possibility to turn off unnecessary animations and testing on several devices	Fabian	
2	Lack of time	Because of a huge workload in other lectures, team members might not have enough time to develop the use cases as planned	2	2	4	good and foresighted time management; setting priorities in order to achieve important goals	Luca	
3	Lose team member	Losing a team member (e.g. due to failing exams)	1	3	3	Relieve group member if such a scenario becomes plausible; distribution of knowledge between all members	Luca	
4	Data loss (GitHub)	Technical problems or wrong usage can lead to data loss on GitHub	1	3	3	backup on multiple devices/servers and responsible usage of Git-tools	Nicolas	
5	Publish problems	Unability to publish the app on Google Play	1	3	3	Publishing the app on our own server	Fabian	
6	Server loss	Jira and the blog are hosted on an own server	1	2	2	Regular backups distribution of knowledge between all members to reduce the impact	Nicolas	
7	Illness	Due to illness some team members could not be able to work anymore	2	1	2	no use of deprecated functionalities and updating only if necessary	Luca	
8	Updates of tools	Because of software updates some used functionalities could not be useable anymore	1	2	2		Nicolas	

### 13.2 Use Case Overview

Our use case overview table is demonstrated below. As our first use case actually counts more than 40h hours, we decided to split it up into the "real" UC 1 and UC 0. In our time statistics, the work of creating the game engine is counted for UC 1. As the game engine is not directly a part of the first use case, we created the new entry called "UC 0" which counts all the time spent on working for the game engine. This way, the time spent on UC 1 should be pretty accurate.

For the use cases of the second semester, we are looking forward to have more precise data.

UC	UC description	Documentation (h)	Coding (h)	Testing (h)	Warm-Up time (h)	Total (h)	FP
0	GameEngine - Foundation	0	42	9	0	51	---
1	start the game from the menu	5	4	3	5	12	?
2	start waves	5	2	0,5	0	7,5	?
3	build towers on selected area	2	4,5	0	0	6,5	?
4	return to main menu	1,5	0	0	0	1,5	?

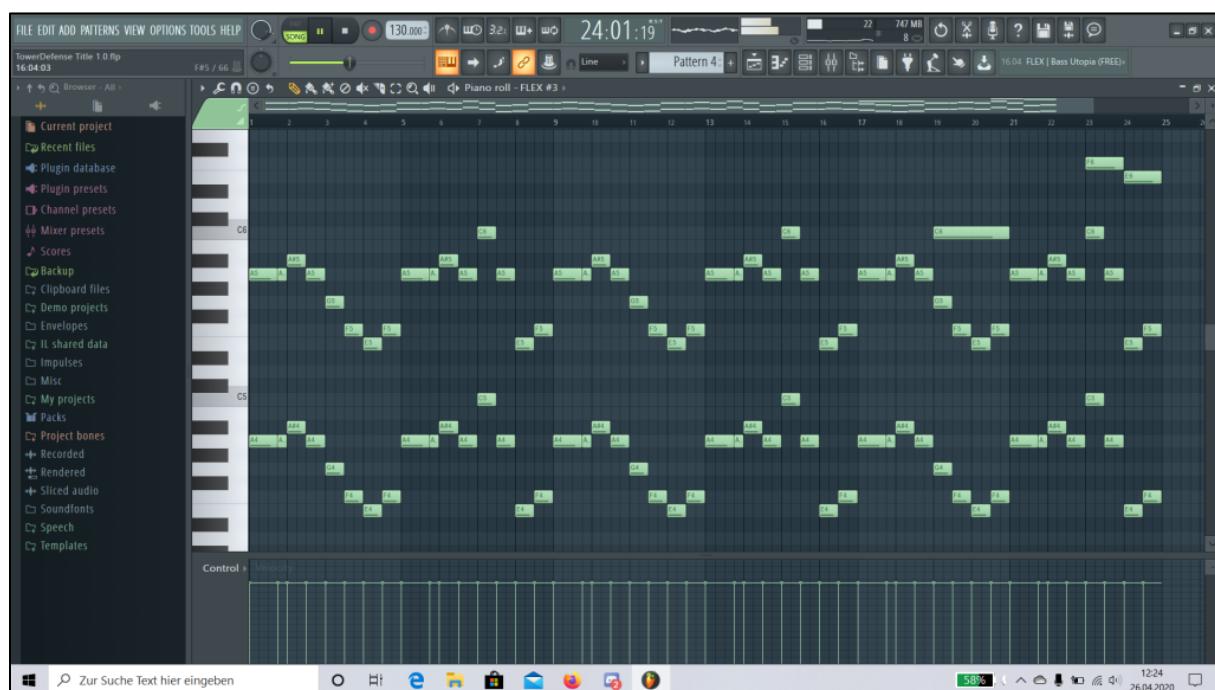
### 13.3 Title-Soundtrack

One big aspect we see for a successful game is the right atmosphere. For that part we do not only focus on design and game mechanics. The atmosphere is also created by sounds. As we have a team member who already has some experience in composing, we decided to make our own soundtrack. Therefore the Software FL Studio was used. After some experiments we now have a result which we call the main theme. This title music will be played in the main menu of the application. That should give the user a good first impression when the game is just started. When there is time left, we will extend our soundtrack to have diversified music also in the background of the game.

As we are developing a tower defense game, where tanks are moving around and towers are firing on them, the music has to create a dark atmosphere. But a good soundtrack also includes some kind of epic melody which pushes the player to focus and continue. Music and the corresponding effect on humans is a lot of science but it surely can help to create an emotional connection to the players.

FL Studio is a professional software for composers and offers almost unlimited possibilities for every kind of music. As we already owned this software, we tried to simulate the moving tanks by using some string instruments with low undertones. By overlaying the melodies we get some nice chords out of it, that may create kind of an epic feeling. The following picture shows an overview from the soundtrack in FL Studio. There you can see all melodies from the different instruments and also filters for regulating the volume. The second picture shows the notes of our main melody, which joins in the middle of the soundtrack.

[Audio]<sub>[59]</sub>



### 13.4 Comments



**AniflixApp-Team** sagt:  
26. April 2020 um 21:33 Uhr

Hello Tower Defense-Team,  
We think you did a good job this week. Your risk management table and your use case overview are clearly understandable.  
We also like the soundtrack you created. We look forward to see more of your work.

Kind regards,  
AniflixApp-Team

Antworten



**Fabian Braun** sagt:  
28. April 2020 um 11:34 Uhr

Hi AniflixApp-Team,  
thanks for that feedback, we appreciate your commendation!

Yours sincerely,  
Tower Defense-Team

Antworten



**Nathalie** sagt:  
27. April 2020 um 16:16 Uhr

Hi Tower Defense-Team,  
you identified a lot of risks in your riskmanagement table and figured out mitigation strategies. You should probably order your table by the risk factor. So your risk management looks great.  
Your UC table looks fine.  
Your soundtrack gives me some 90s adventure game feelings, so i like that.

Keep up the great and creative work.  
Best regards,  
Nathalie from orchestra-team

Antworten



**Fabian Braun** sagt:  
28. April 2020 um 11:41 Uhr

Hello Orchestra-Team,  
thanks for your prasing feedback. We will reorder our table soon, as you mentioned. We are glad to hear, that our soundtrack causes the right feelings. Hopefully we have enough time to continue our creative kind of work! 😊

Kind regards,  
Tower Defense-Team

Antworten



**Nicolas Wagner** sagt:  
30. April 2020 um 11:43 Uhr

Hello Team Orchestra,  
our risk management table is updated and the risks are sorted in descending order now. Thanks again for the hint!

Kind regards,  
Tower Defense-Team

Antworten

## 14 Blog Entry Week 13: Function Points / Estimation

### 14.1 Function Points – What's that?

Welcome to our blog entry of week 13. Today, we mainly want to present the function point calculations and our estimation to you.

First of all: **What are "Function Points" and what are they useful for?**

In every project, estimation and planning are very important aspects. Most often, it is difficult to estimate the workload of a ticket or a use case according to one's "gut feeling". Because of that, tools and methods such as functions points or story points exist.

Summarized, several aspects are considered for each use case while calculating the function points. One differentiates the following aspects:

- EI: External Input
- EO: External Output
- EQ: External Queries
- ILF: Internal Logical Files
- EIF: External Interface Files

Depending on the category, RETs (Record Element Types), DETs (Data Element Types) or FTRs (File Type References) need to be analyzed according to a use case and the results should be listed in an overview table. Moreover, one can conclude a complexity out of the gained values as demonstrated in the table on the right.

Using the function points calculator "[TINY TOOLS](#)"<sup>[58]</sup>, function points can be calculated based on the analyzed data.

Below, you can see our function point calculations of the first 4 use cases (last semester) and of the current 6 use cases (this semester). Those have been calculated with our complexity adjustment table (on the right).

(The original function points excel table can be [viewed on GitHub](#)<sup>[28]</sup>).

Tables				ILF and EIF Complexity Matrix			Complexity Adjustment Table						
				1-19 DETs	20-50 DETs	51+ DETs	ITEM	COMPLEXITY ADJUSTMENT QUESTIONS					
				1	2	Avg	1	No influence	1	2	3	4	Essential
RETs	1	Low	Low	Low	Low	Avg	1	Does the system require reliable backup and recovery?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	2-5	Low	Avg	High			2	Are data communications required?	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	6+	Avg	High	High		High	3	Are there distributed processing functions?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Visible groupings				EI Complexity Matrix			4	Is performance critical?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
FTRs	1-4 DETs	5-15 DETs	16+ DETs	5	Will the system run in an existing, heavily utilized operational environment?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	0-1	Low	Low	Avg	6	Does the system require on-line data entry?					<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	2	Low	Avg	High	7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?					<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	3+	Avg	High	High	8	Are the master files updated on-line?					<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
EO and EQ* Complexity Matrix				9	Are the inputs, outputs, files or inquiries complex?					<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
FTRs	1-5 DETs	6-19 DETs	20+ DETs	10	Is the internal processing complex?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	0-1	Low	Low	Avg	11	Is the code to be designed reusable?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	2-3	Low	Avg	High	12	Are conversion and installation included in the design?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	4+	Avg	High	High	13	Is the system designed for multiple installations in different organizations?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
				14	Is the application designed to facilitate change and ease of use by the user?					<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Domain Characteristic Table | FPC Calculator

UC1: Start the game from the menu	RET	DET	FTR	Complexity	Amount
External Inputs	---	2	1	low	3
External Outputs	---	3	3	low	6
External Queries	---	0	0	low	0
Internal Logical Files	2	5	---	low	7
External Interface Files	0	0	---	low	0
Function Points			82		

RETs	DETs	FTRs
	Play-Button, MainActivity	MainActivity
	Field-Buttons, Moving Tanks, generated Map	GameActivity, MapStructure, Game
	---	---
Tanks, Map	Tanks, Map, Fields, MatchField, Game	
	---	---

UC2: Start Waves	RET	DET	FTR	Complexity	Amount
External Inputs	---	1	2	low	3
External Outputs	---	2	2	low	4
External Queries	---	0	0	low	0
Internal Logical Files	3	5	---	low	8
External Interface Files	0	0	---	low	0
Function Points			81		

RETs	DETs	FTRs
	wave-button	GameActivity, Game, MatchField
	wave-button, Tanks	GameActivity, Game, MatchField
	---	---
Game, MatchField, Enemies	Game, MatchField, Enemies, Tanks, MapStructure	
---	---	

UC3: Build Towers on selected Area	RET	DET	FTR	Complexity	Amount
External Inputs	---	2	3	average	5
External Outputs	---	2	2	low	4
External Queries	---	0	0	low	0
Internal Logical Files	2	2	---	low	4
External Interface Files	0	0	---	low	0
Function Points			64		

RETs	DETs	FTRs
	Field-Buttons, Plus- Symbol	MatchField, Tower, Plus-Symbol
	Tower, MatchField	MatchField, Tower
	---	---
Game, MatchField	Game, MatchField	
---	---	

UC4: Return to main menu	RET	DET	FTR	Complexity	Amount
External Inputs	---	2	2	low	4
External Outputs	---	1	0	low	1
External Queries	---	0	0	low	0
Internal Logical Files	2	2	---	low	4
External Interface Files	0	0	---	low	0
Function Points			44		

RETs	DETs	FTRs
	Main-Menu-Button, GameActivity	GameActivity, Game
	MainActivity	---
	---	---
GameActivity, Game	GameActivity, Game	
---	---	

UC5: Upgrade Towers	RET	DET	FTR	Complexity	Amount
External Inputs	---	2	2	low	4
External Outputs	---	3	2	low	5
External Queries	---	1	1	low	2
Internal Logical Files	3	3	---	low	6
External Interface Files	0	0	---	low	0
Function Points			80		
Estimation: 10,1h					

RETs	DETs	FTRs
	Tower-Selection, Upgrade-Button	Tower, GameActivity
	Tower, MatchField, GameActivity	Tower, GameActivity
	Tower-Stats	Tower
GameActivity, Tower, Level-Constants	GameActivity, Tower, Upgrade-UI	
---	---	

UC6: Sell Towers	RET	DET	FTR	Complexity	Amount
External Inputs	---	2	2	low	4
External Outputs	---	3	2	low	5
External Queries	---	1	1	low	2
Internal Logical Files	3	3	---	low	6
External Interface Files	0	0	---	low	0
Function Points			80		
Estimation: 10,1h					

RETs	DETs	FTRs
	Tower-Selection, Sell- Button	Tower, GameActivity
	Tower, MatchField, GameActivity	Tower, GameActivity
	Tower-Stats	Tower
GameActivity, Tower, MatchField	GameActivity, Tower, Sell-UI	
---	---	

UC7: Select between different Towers	RET	DET	FTR	Complexity	Amount
External Inputs	---	2	2	low	4
External Outputs	---	2	2	low	4
External Queries	---	2	1	low	3
Internal Logical Files	3	3	---	low	6
External Interface Files	0	0	---	low	0
Function Points			79		
Estimation: 9,86h					

RETs	DETs	FTRs
	Tower, Position	Tower, Game
	Tower-Selection, Select-Button	Tower, GameActivity
	Money, Price	Game
GameActivity, Game, Tower-Template	GameActivity, Game, Money	
---	---	

UC8: Edit Settings	RET	DET	FTR	Complexity	Amount
External Inputs	---	1	2	low	3
External Outputs	---	1	1	low	2
External Queries	---	1	1	low	2
Internal Logical Files	3	2	---	low	5
External Interface Files	0	0	---	low	0
<b>Function Points</b>					<b>58</b>
Estimation: 4,3h					

UC9: Toggle Sound & Animations	RET	DET	FTR	Complexity	Amount
External Inputs	---	1	1	low	2
External Outputs	---	2	1	low	3
External Queries	---	1	1	low	2
Internal Logical Files	3	3	---	low	6
External Interface Files	0	0	---	low	0
<b>Function Points</b>					<b>66</b>
Estimation: 6,4h					

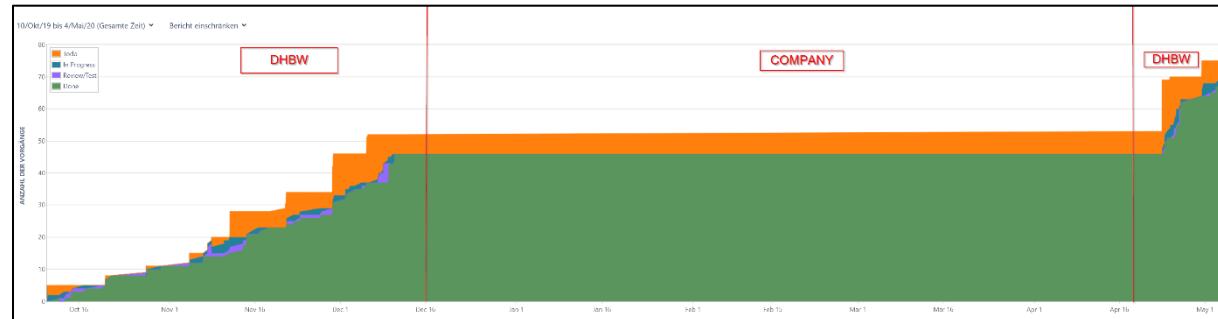
UC10: Choose Difficulties	RET	DET	FTR	Complexity	Amount
External Inputs	---	3	1	low	4
External Outputs	---	3	1	low	4
External Queries	---	1	1	low	2
Internal Logical Files	3	3	---	low	6
External Interface Files	0	0	---	low	0
<b>Function Points</b>					<b>76</b>
Estimation: 9,1h					

RETs	DETs	FTRs
	Menu-Button	GameActivity, Game
	Settings-UI	GameActivity
	Settings	SharedPreferences
SharedPreferences, GameActivity, Game	Settings, GameActivity	
---	---	

RETs	DETs	FTRs
	Settings-Buttons	GameActivity
	Settings-Buttons, changed Settings	GameActivity
	Settings	SharedPreferences
SharedPreferences, GameActivity, MainActivity	Settings, GameActivity, MainActivity	
---	---	

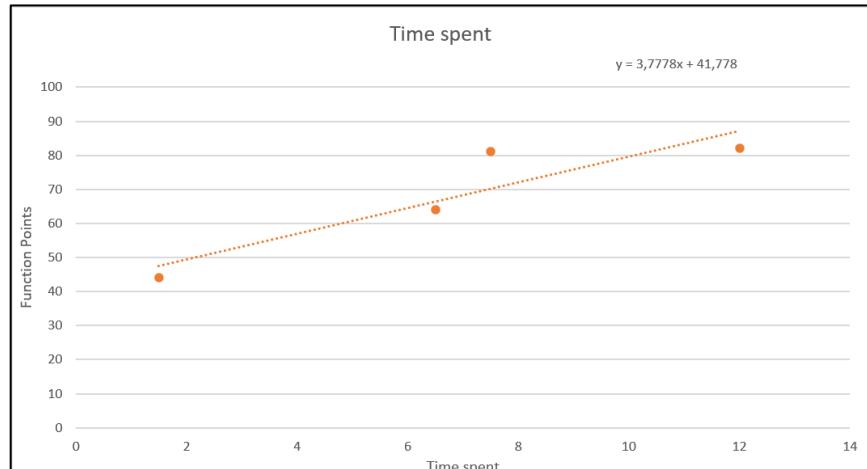
RETs	DETs	FTRs
	MainActivity, Difficulty, Play-Button	MainActivity
	GameActivity, Game, Game-Settings	GameActivity
	Statistics	SharedPreferences
SharedPreferences, MainActivity, GameActivity	Statistics, GameActivity, MainActivity	
---	---	

## 14.2 Cumulated Flow Chart



## 14.3 Time Estimation

As you can see in our cumulated diagram above, our working-speed is very high at the moment in contrast to the last semester. Although we expect to not keep this speed up (due to exams and rising workload in other lectures), it is probable that all our use cases could become finished in a basic way within four to six weeks. This is the result of our calculations based on the values which one can see in the chart on the right.

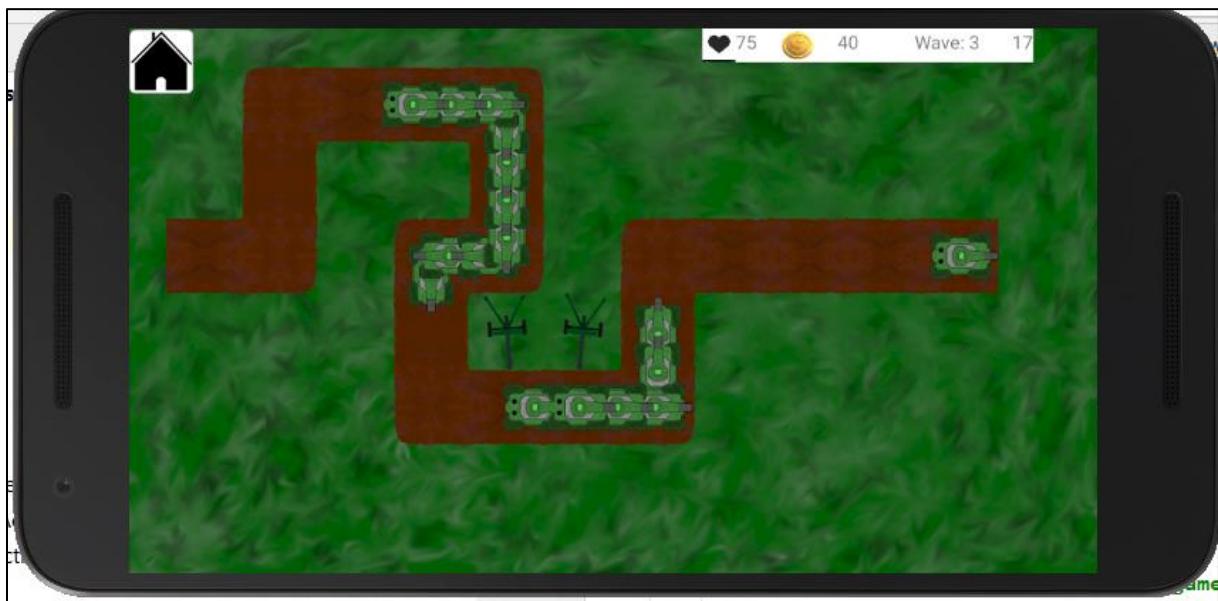


If we are able to stick to this plan, we will use the remaining time for optimizations, improving the performance and getting the app ready for publication.

#### 14.4 Progress Gameengine

Since we are still at the beginning of the semester, we have more time than usual when exams are getting closer. That's why we do the sprints with a big workload and our app begins to grow. The priority for the next weeks is to get our defined Use Cases implemented and have a well working game at the end of the month. Because we also want to have enough time for the design and to adjust the right difficulty in the game, so that users are not bored after some time. For now, we can say there is a big progress we made over the last days and some UCs are almost finished.

One finished functionality/feature we achieved this week is our status bar. You can see it in the picture below. This bar shows the current state of the game. This includes the lifepoints, the current money and also some information about the waves, like the wave number and the time until the next waves start. Looks like an easy job, right? That's not quite true, because of some big changes we did in the background. Of course, if you buy, sell or upgrade a tower you need to have a possibility to change the amount of money in a consistent way. If you kill enemies, they have to drop some money. And if an enemy passes the target it has to decrease the players lifepoints. Lifepoints also gave us the possibility to win or lose the game, which was not defined yet. Another hidden but implemented feature is our concept of waves and matches. A wave consists of a list of enemies, where every enemy can be of a different level or type, as well as information about spawnrate. The whole Match exists of a list of waves and further information but we need also an efficient way to compose a match or rather each wave of it. To have a consistent model we used abstract classes and we also work on some refactorings. In summary we build a little framework to make our next steps in implementation easier. But the current state already made the game kind of playable. So that were just a few words of our current work. There may be some very cool updates in the next weeks! 😊



## 14.5 Comments



**Anonymous** sagt:  
6. Mai 2020 um 9:11 Uhr

Hey guys,  
your blog entry looks very nice and thought out. You pointed out your calculations very detailed and the screenshot of your game is giving an overview how you progressed so far and it looks amazing! I hope you can progress like this to reach your goals.

Keep up your good work,  
Team LogicGame

Antworten



**Nicolas Wagner** sagt:  
6. Mai 2020 um 9:38 Uhr

Hi Team LogicGame,  
thank you for the positive feedback!  
Because of our progress concerning the game engine, we expect to present many more features to you very soon :)!

Kind regards,  
Nicolas

Antworten



**Anonymous** sagt:  
6. Mai 2020 um 11:55 Uhr

Hi,  
your blogpost seems to meet the grading criteria but I think the Cumulated Flow Chart should show the time per workflow eg. implementation deployment etc. so that you can estimate the ratio between coding and everything else.  
yours MAPHYNN

Antworten



**Nicolas Wagner** sagt:  
6. Mai 2020 um 13:57 Uhr

Hi Team MAPHYNN,  
thank you for your feedback and for the hint concerning the Cumulated Flow Chart.  
Sadly, our project management tool (Jira) does not support the generation of a cumulated flow chart with a workflow-view.  
In a few weeks, we are going to publish several pie-charts instead which offer more detailed data (also concerning the workflow, phase etc.).

Kind regards,  
Nicolas

Antworten



**Gyrogame Team** sagt:  
6. Mai 2020 um 11:58 Uhr

Hello Tower Defense Team,  
you have a very in detailed description on how you calculated your function points! No matter how hard we look, we cannot find anything to criticize in this post, except for maybe more details about the use cases themselves.  
Keep up the great work!  
  
Regards,  
The Gyrogame team

Antworten



**Nicolas Wagner** sagt:  
6. Mai 2020 um 14:15 Uhr

Hi Team Gyrogame,  
thanks for your feedback!  
I understand your aspect concerning the details of our use cases.  
So far, we tried to specify the use cases as detailed as possible, but as we are developing a game in which all components interact with each other anyhow, more detailed specifications could have restricted our progress. We hope that our use cases are understandable nonetheless :).  
  
Kind regards,  
Nicolas

Antworten

## 15 Blog Entry Week 14: Testing

### 15.1 Testing

Welcome back to our blog! This week, we are dealing with several aspects concerning testing.

Testing becomes more and more important in today's way of development. In our project, we want to focus on three different test-types: UI testing, JUnit testing & UX testing.

UI testing is important in order to check if the game-UI works properly. Such tests have already been implemented and can be seen in our [blog entry of week 5](#)<sup>[8]</sup>.

For our game engine, it makes sense to use JUnit for testing.

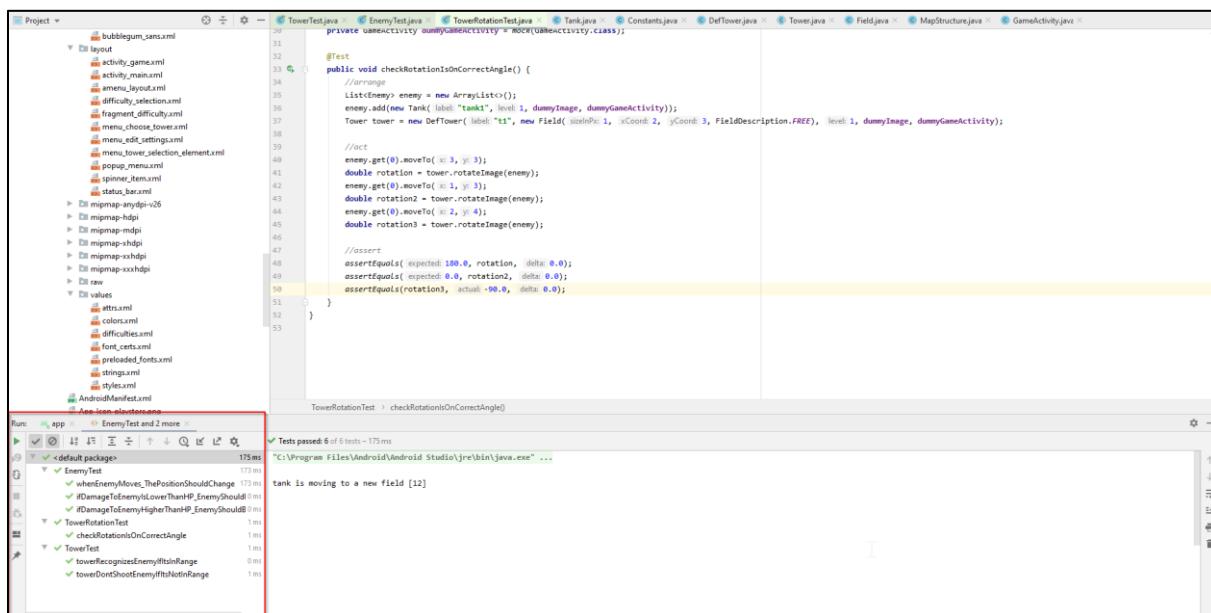
In late phases of development, we plan to run through some UX-tests so that we receive some feedback about the self-explanation aspect of our app.

The blog entry of this week mainly deals with JUnit testing.

In our repository on GitHub, a folder called "test" exists ([click to check out the folder](#)<sup>[57]</sup>). This folder contains some JUnit tests to check if our game engine is still working. From now, those tests will run and the build-process will start if we push some commits up to our GitHub repository. The test-results of each test-on-push are located in our [Android-CI area](#)<sup>[56]</sup>. Moreover, there is a screenshot of our successfully runned JUnit tests below. Our [build.gradle file](#)<sup>[55]</sup> shows the integration of our chosen test-libraries.

At the moment, it is not possible to create a Unit test which covers all aspects of a use case because of permanent connections to our UI. Nonetheless, we think that our Unit tests cover the most necessary components of most use cases.

Furthermore, our [test-plan document](#)<sup>[31]</sup> is also available on GitHub now.



```

private GameActivity dummyGameActivity = mock(GameActivity.class);
private Enemy enemy;
private Tower tower;

@Test
public void checkRotationsOnCorrectAngle() {
    //Arrange
    List<Enemy> enemy = new ArrayList<>();
    enemy.add(new Tank(label: "tank1", level: 1, dummyImage, dummyGameActivity));
    Tower tower = new DefTower(label: "t1", new Field(id: screenW / 1, xCoord: 2, yCoord: 3, FieldDescription.FREE), level: 1, dummyImage, dummyGameActivity);

    //Act
    enemy.get(0).moveTo(x: 3, y: 3);
    double rotation = tower.rotateImage(enemy);
    enemy.get(0).moveTo(x: 1, y: 3);
    double rotation2 = tower.rotateImage(enemy);
    enemy.get(0).moveTo(x: 2, y: 4);
    double rotation3 = tower.rotateImage(enemy);

    //Assert
    assertEquals(expected: 180.0, rotation, delta: 0.0);
    assertEquals(expected: 0.0, rotation2, delta: 0.0);
    assertEquals(rotation, actual: -90.0, delta: 0.0);
}

```

Tests passed: 6 of 6 tests – 175ms  
T:\Program Files\Android\Android Studio\jre\bin\java.exe" ...  
tank is moving to a new field [12]

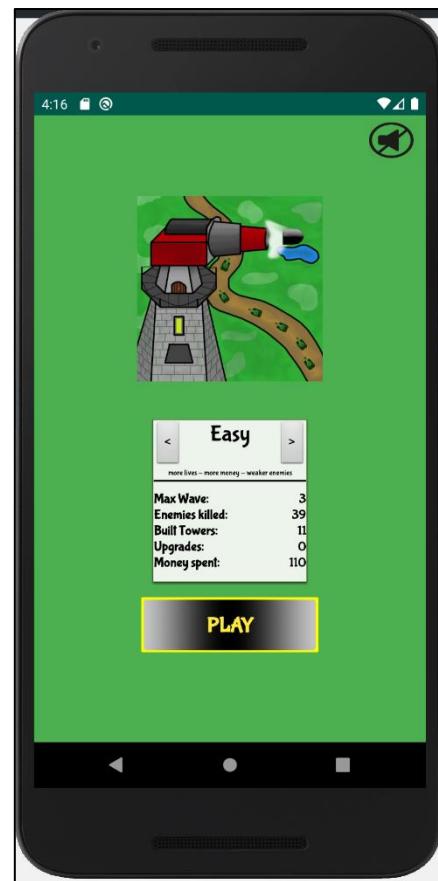
## 15.2 New Feature: Choose Difficulties!

In the period of sprint 13, the use case "[Choose Difficulties](#)"<sup>[42]</sup> has been developed.

From now, a user is able to see a box on the main activity. In this box, three panels for three difficulties can be displayed: easy, medium and hard. If a user clicks on the play-button, the game will start with the currently chosen difficulty. Depending on the difficulty, the amount of start-money, the amount of lives and the composition of waves can differ.

Those panels were developed by making use of Fragments and a ViewPager. For each difficulty-panel, a fragment has been created and each fragment is embedded into the ViewPager which is a part of the main activity and which makes the box become visible.

In addition to the new difficulty feature, we implemented statistics into the game. After winning or losing a game, the values "Max Wave", "Enemies killed", "Built Towers", "Upgrades" and "Money spent" will be saved if the recent result is higher than the current high score. Those stats are displayed on the main activity in the difficulty-box as well, because the statistics are saved in dependence of the chosen difficulty.



The statistics become saved using the Android Shared Preferences. This functionality allows developers to save small amount of data into automatically created files by the key-value principle. In the future, we are going to make use of the Shared Preferences one more time when developing the game-settings.

Probably, the design we are going to change the design of the main page later on.

## 15.3 New Feature: Select between different towers!

Also in the period of sprint 13, the use case "[Select between different towers](#)"<sup>[39]</sup> has been developed.

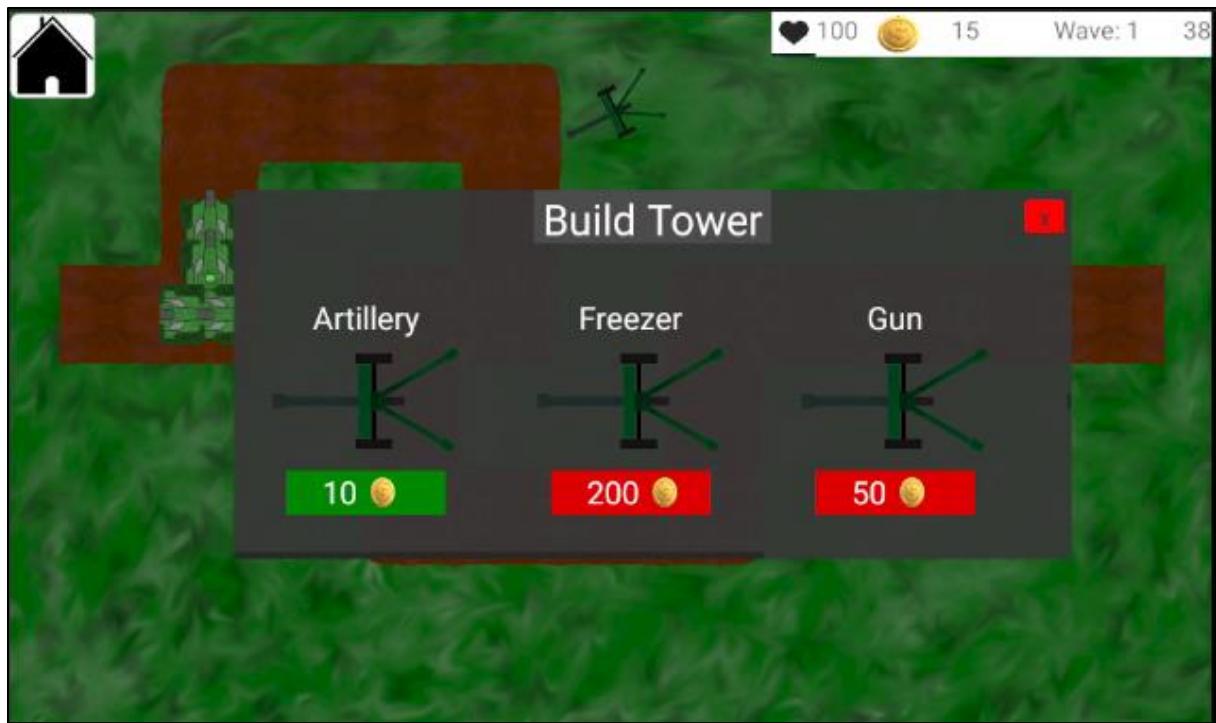
Since this Use Case is implemented, it is possible to open a popup window and build a selected tower onto a field.

To open the build-popup, it is necessary to double click onto a free field. Within this popup, users are able to view all implemented towers. For each tower, the name, its image and its price is shown. By clicking on the price, a tower will be built. Of course, building a tower is only possible when the player has enough money. The price field will be green if the player has enough money. Otherwise, such a price field will be red. When clicking on a red price-tag, nothing will happen. Otherwise, the popup closes and the tower will be built. If a player does not want to build a tower, it is possible to close the popup by clicking on the close button in the right top corner.

The popup window becomes generated by the use of two XML-files. One XML-file is used for creating the frame and the other one creates the content.

This allows us to use the first layout file for all popups we want to have because we planned to create two more popups for upgrading /selling towers and for changing the settings. Hence, this separation saves us some time in the future.

In the picture, you can see how this popup looks like right now. At the moment, all towers have the same image. The reason: Other towers are not implemented yet but they will be there very soon.



#### 15.4 Conclusion

All in all, we are really happy with our progress. Even our project is not that handy to test components as actually required, we think to be able to manage adequate testing.

Moreover, we are looking forward to keep the speed of development up so that the main functionalities will be developed before the end of the semester. Probably, there will be another publication of the next use cases in the blog entry of next week.

Thanks for your interest and see you soon!

## 15.5 Comments



Marlon sagt:  
11. Mai 2020 um 17:27 Uhr

Hi Team Towerdefense,  
great to hear about your progress in testing your game. Your JUnit tests look good although I can't find the actual test runs in your GitHub actions, maybe you could link to one concrete example.  
That you can't test a whole use case with one JUnit test isn't a problem to my mind as this would be the task of an integration test or something similar. Units tests should only test very small parts of your software, I believe. Testing more complex aspects will be handled by other tests.  
Also your new features sound really good. I liked how you tried to give an explanation on how everything works together.  
  
Looking forward to see more.  
  
Yours,  
Marlon from Clairvoyance



Nicolas Wagner sagt:  
11. Mai 2020 um 18:29 Uhr

Antworten

Hi Marlon,  
thank you for the feedback!

I have to admit that the paragraph about unit testing does not highlight our main intention. In the grading-criteria, it says that we should choose one use case and write tests for each path in the corresponding activity diagram. This is actually what we are not able to cover yet because of several connections to the UI.

Concerning the testing in our GitHub actions, I have to agree that we do not have an optimal solution yet. Currently, we are using a third party action called „Android GitHub Action“. In the GitHub-Cl area (e.g. [https://github.com/nawa99/Tower-Defense/runs/662499930?check\\_suite\\_focus=true](https://github.com/nawa99/Tower-Defense/runs/662499930?check_suite_focus=true)), this action becomes runned, but there is not any obvious reference to our Unit tests being runned. Looking into the plugin sources (<https://github.com/Vukan-Markovic/Github-Android-Action/blob/master/github/workflows/main.yml>), it becomes more clear that the tests are runned.

I am really sorry for this laboriousness! Probably, we will switch to another tool like Travis lateron.

We will keep you up to date :).  
Kind regards,  
Nicolas

Antworten



Nico sagt:  
13. Mai 2020 um 11:01 Uhr

Hi Towerdefens Team,  
you did a nice job with your Unit-tests yet!  
I really like how you use aliases for the test description, so everyone can see what a test does or what test fails. Even if he doesn't know the code.  
I know there are some difficulties with continuous testing, but I'm looking forward that you will do your way.

Best regards and keep going  
Nico

Antworten



Nicolas Wagner sagt:  
13. Mai 2020 um 11:53 Uhr

Hi Nico,  
thanks for your feedback!  
Hopefully, we can set up the CI completely according to our wishes and needs very soon.

Kind regards,  
Nicolas

Antworten



Anonymous sagt:  
13. Mai 2020 um 11:12 Uhr

Hi Team Towerdefense,  
your JUnit tests are looking really good. JUnit is a good tool and doesn't create too much of a hassle.(at least in our experience ;D)  
The tests are pretty detailed and give a good view about the future of your project.  
We are looking forward to your results  
Sincerely  
PiPossible



Nicolas Wagner sagt:  
13. Mai 2020 um 12:17 Uhr

Antworten

Hi Team PiPossible,  
thank you for your feedback!  
We definitely agree that JUnit is a great tool for software engineering in general!

Kind regards,  
Nicolas

Antworten

## 16 Blog Entry Week 15: Refactoring

### 16.1 Fowler's Refactoring

Hi everyone and welcome back to our blog!

This week deals with a specific refactoring project. In order to achieve active awareness of refactoring whenever we are writing code, we should refactor a given project according to Fowler's idea of refactoring.

Below, the refactoring-projects of each team-member are linked.

- Fabian: <https://github.com/Fabian1699/SE-Refactoring>
- Luca: <https://github.com/Charon1502/Refactoring>
- Nicolas: <https://github.com/niwa99/Refactoring-Project-SE>

Of course, this is not all what we have done so far. Following, new developments of our tower defense game are described.

### 16.2 UC Upgrade and Sell Towers

According to this Use-Case several functions were added.

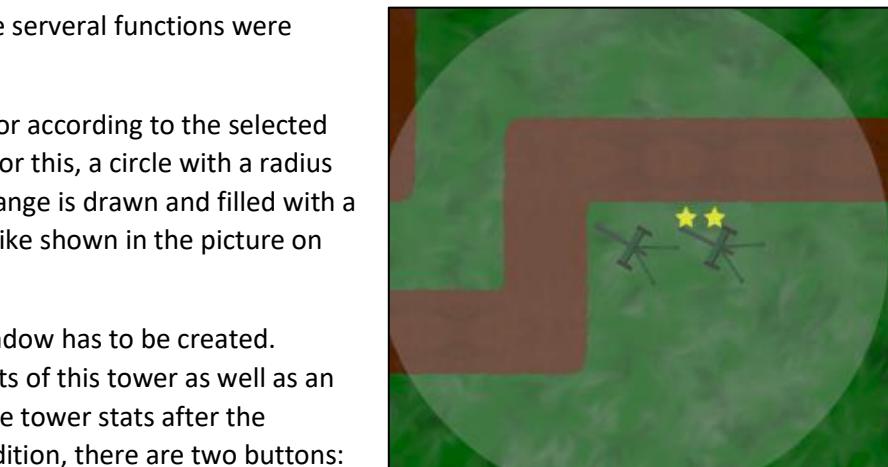
First of all, a range indicator according to the selected tower has been created. For this, a circle with a radius depending on the tower range is drawn and filled with a transparent background (like shown in the picture on the left).

Furthermore, a popup window has to be created.

Within this popup, the stats of this tower as well as an image of the tower and the tower stats after the upgrade are shown. In addition, there are two buttons:

one to sell the selected tower and one to upgrade the tower (like shown in the second picture on the left).

When clicking on the sell button, one receives the half of the current value back. When clicking on the upgrade button, money according to the



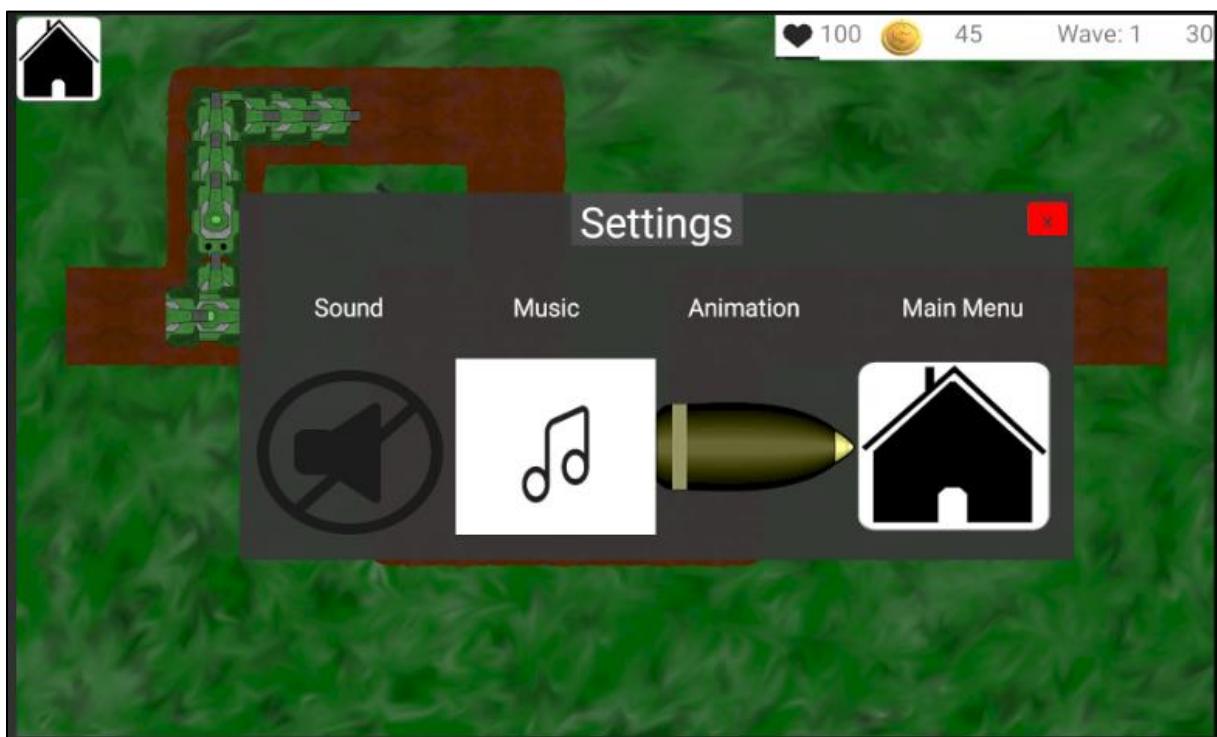
amount of the price is decreased. Right now, the maximum level of all towers is three. According to this, the text of the upgrade button changes to "max reached" and there will nothing happen when clicking on the tower.

The last necessary feature to add within this Use Case is a sign to show the level of each tower. We decided on displaying stars for the tower-levels. When upgrading a tower to level two, one star appears on the top of the tower. When upgrading to level three, a second star is added and when selling this tower, all stars disappear.

### 16.3 UC Edit Settings

The use case Edit Settings should give the player the possibility to pause the game and change some important settings. Therefore by clicking the pause button a menu is shown where the player can toggle sound, background music and animations. Another button beams the player back to the main menu. If the dialog is cancelled the game will continue exactly at the same time it was paused. This feature costed us unfortunately many hours, because of our current use of Java Timers which cannot be paused. So we had to calculate a delay for the towers for their next shoot and also for the waves to spawn and so on. Otherwise towers would shoot directly after continuing although they may have a pause time of some seconds. That would allow the player to cheat more or less. To calculate the delay, every object has to log its time of the last action to be able to retrace the necessary pause time for the next action if the pause button is clicked.

Even if the layout and also the Icons will be adjusted in the next weeks, we are now finished with implementing the functionalities of all our planned use cases. The game is fully playable now. But there will be some more cool updates including more towers and enemies with special abilities to improve the game flow.



### 16.4 Comments

 **GreenClothaWay-Team** sagt:  
19. Mai 2020 um 8:07 Uhr

hey dudes  
i saw all pf you committed only on master.  
try ro make branches with speaking names for every step you implement and the do a pull request. its so easy that its not even too much effort for this kind of repo/project.

 **Nicolas Wagner** sagt:  
20. Mai 2020 um 11:04 Uhr 

Hi Team GreenClothaWay,  
thanks for your feedback!  
Sadly, we failed to see the aspect of using branches in this project and cannot change the use of branches anymore but we are using a lot of branching in our real tower defense project :).

Kind regards,  
Nicolas



**Marlon** sagt:  
19. Mai 2020 um 16:13 Uhr

Hey guys,

overall your refactoring looks nice, just a few small remarks. Maybe you could also include the IDE specific files in your .gitignore, they could lead to problems when others clone your repo, although not important for this small example. Next, some of you renamed the statement-method, I know it's not the best name for that function, but I'm not sure if it's a good idea to do this in a refactoring, as „old“ code that relies on this method will be broken after that change. At last, your tests include everything I believe, but a few more scenarios would be even better.

Apart from that refactoring, great to see that you had time to work on two more use-cases. Our team wasn't that productive this week.

Looking forward to see more.

Yours,  
Marlon from Clairvoyance

Antworten



**Nicolas Wagner** sagt:  
20. Mai 2020 um 11:44 Uhr

Hi Marlon,

thanks for your detailed feedback!

Your hint concerning the .gitignore is really good, thanks!

Some of us renamed the statement method because we did not think that this method name is chosen well and wanted to make the whole code as clean as possible. If the given project is actually a part of a bigger project, the IDE will refactor the remaining snippets of code as well.

Kind regards,  
Nicolas

Antworten



**Mr CardGame** sagt:  
25. Mai 2020 um 10:19 Uhr

Hey

I looked at your repos, it looks like you did a good job. I just think you could have made the output string a little more general so you don't always have to manually type / t for tabs for every other word. I would only define a constant for this Luca.

Best regards  
Mr CardGame

Antworten



**Luca Rutschmann** sagt:  
25. Mai 2020 um 11:46 Uhr

Hi Mr CardGame,

that's a good point you mentioned here.

Thanks for your feedback. ^^

Kind regards  
Luca

Antworten

## 17 Blog Entry Week 16: Design Patterns

### 17.1 Design Patterns

Hi and welcome back to our blog!

Today, our blog entry deals with design patterns as well as with our further progress concerning the tower defense game functionalities. Let's start with design patterns.

Design patterns should be used in order to have clearly structured code. A lot of design patterns exist such as "Singleton", "Factory Method", "Abstract Factory" or "Decorator". In our opinion, the following websites give a good introduction into the design pattern topic:

- <https://www.oodesign.com>
- <https://www.philippauer.de/study/se/design-pattern.php>

In our project, we made use of several design patterns from the very beginning. Due to that, we could not find any part in our code for which the usage of another design pattern would be sensible right now. Nonetheless, we want to show to you that design patterns are really important to us and therefore, you can find our current class diagram below. Underneath the class diagram, we will give several explanations on where design patterns are applied.

Note: As our class diagram grew by a lot within the last months, it is not really readable in the preview below anymore. By opening the image in a new tab, it might become more readable.



## 17.2 The use of design patterns in our project

### 1. Factory

*Creates objects without exposing the instantiation logic to the client and Refers to the newly created object through a common interface. (odesign.com)*

The tower defense app makes use of three interfaces so far.

The first interface is called "**IStatusBar**". It is responsible for the status right at the top while being in the game. This interface enables us to interact with the status bar more easily.

The second interface is called "**ISettingsManager**". This interface is used for all functionalities which enable a user to toggle the app settings. Using this interface, we can always call just one method and provide a setting type as an enum which makes the change of settings very handy.

The third interface is called "**IMoneyListener**" and controls the change of money while being ingame. For example, if a player upgrades a tower, the money-change information is passed from the UpgradeAndSell-activity to the game.

Concerning the naming style for our project, all interfaces start with an "I" and all abstract classes start with an "A" to immediately recognize the type of those classes. Especially, this convention simplifies reading outside of Android Studio, e.g. on GitHub.

### 2. Abstract Factory

*Offers the interface for creating a family of related objects, without explicitly specifying their classes. (odesign.com)*

Abstraction is one of the most important design patterns in the tower defense game. It is used in many parts of the project. In this section, we want to point out the most interesting usages of abstraction.

- **Tower:** A player is able to build several different towers such as an ArtilleryTower, a FreezerTower or a BoombasticTower. All those towers consist of common functions (coordinates, image-attribute, damage, range, ...) and tower-specific functions (freezing, area damage, tower rotation, ...).
- **Bullet:** A bullet becomes fired by a tower. Commonly, a bullet has got attributes like coordinates, a target position and a target enemy. In its specific implementations (Bomb, SnowFlake, Projectile, ...), changing functionalities (freezing, detecting multiple enemies, ...) are defined.
- **Enemy:** Enemies also share several attributes such as coordinates, life points, revenue and many more. In its implementations, more specific things are defined, e.g. the rotation of its image, its behaviour when being hit or its values.

"AMatch", "AWave", "AMenu" and "ATimerUsage" are more examples of abstraction in our project and the whole structure of using this design pattern enables us to create and adjust new implementations of all those types, which is a very big advantage when adding new features or adjusting existing features for more than one implementation. As a result, it reduces the amount of code massively.

### **3. Object Pool**

*reuses and shares objects that are expensive to create. (oodesign.com)*

In an earlier state of our project, we had a class called "ObjectStorage". In this class, all needed object-instances for keeping the app operating have been maintained (e.g. GameActivity, Game, MatchField and many more). Some weeks/months ago, this was a very easy-to-use solution but performance intensive as well because the Android framework is not designed for holding a lot of instances which are not permanently necessary. Because this design pattern is not suitable for our project anymore, we got rid of it in the beginning of the 4th semester.

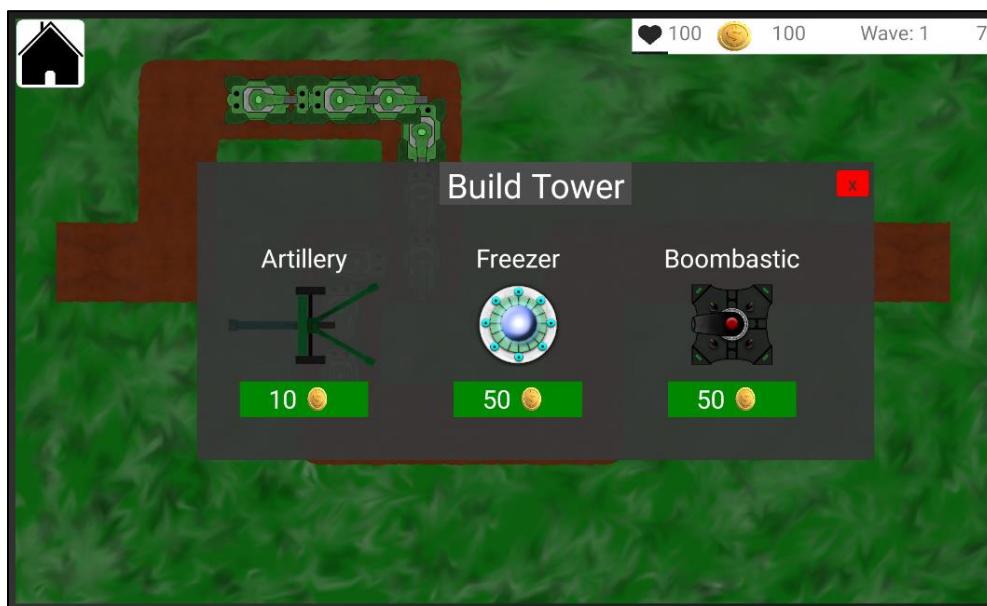
### **Conclusion**

Summarizing, we can conclude that design patterns are already used in our project from the beginning of development, which probably enables us to develop new features comparatively quickly. Because of some restrictions due to our framework and the requirements of the app itself, we are not trying to apply as many design patterns as possible and stick to the patterns which suit best to our project.

### 17.3 New Tower: Freezer

Last sprint we implemented of course some new features. As our planned UCs are already implemented, we worked this week on new Towers. We think this is currently the most important part, because towers with different special effects allow the player to think about tactical methods to play the game and defeat especially the hard waves. Maybe you have seen our first Tower in previous pictures and clips. We call it Artillery and it includes the basics we expect of a tower. It simply shoots with one bullet at one enemy which is located in the range of the Tower.

The second tower is called "Freezer" which comes with the ability to freeze enemies. It makes less damage but slows a hit enemy down for a few seconds. The slowness effect decreases continuously and after some time the enemy gets back to its full speed. For moving the enemies, we use simple timertasks. But here we had to improvise a little bit and our solution is about a timertask which works recursively by calling itself with a specific delay that decreases with every movement of the enemy, so this effect disappears in a smooth way. The Bullet is visible as a little snowflake.



#### 17.4 New Tower: Boombastic

"Boombastic" builds the third tower and lives up to its name. It shoots with a cannon ball that causes area damage in a given range. The tower focuses at one enemy but the canon ball will end in an explosion which causes damage to near enemies. Of course, the damage decreases according to the distance an enemy has to the explosion.

We mentioned that we use a lot of abstract classes. Here we had a big benefit and were able to implement these towers and their bullets within a few hours and little code because we only had to extend the normal behavior by the special effects. So we are already working on more towers. Be Excited!



## 17.5 Comments



**Bookly** sagt:  
25. Mai 2020 um 22:35 Uhr

Hey Team Towerdefense,

I'm actually stunned how well your application is structured. I already developed some Android Native and especially with bigger application it's not that easy to build such a framework of activities/fragments. I looked at your Class Diagram and I get why you say that you don't want to implement something like a „mediator“ because it simply does not make sense in your context.

You mentioned that you got rid of a „object manager“ which is very common among beginner android devs, because it's such an easy solution. However it's a solution to a problem you are not supposed to have in first place because the android framework is not designed to work this way (for example treating activities like classes in java). The only thing I noticed is that I could not check your Project Management on Jira.

Do I really have to create an account for this, or can't you just make it public?

Best Progress I've seen so far 😊

Kind regards  
Team Bookly

Antworten



**Nicolas Wagner** sagt:  
27. Mai 2020 um 14:37 Uhr

Hi Team Bookly,

thanks a lot for your feedback!

We really appreciate that!

For us, it is a great feedback that you as an Android developer confirm our decisions regarding the implementation of several design patterns.

Concerning Jira, we do not have such good news... Sadly, one cannot make the whole tool public so that you cannot see everything in detail, but using the following link, you can see at least all our tickets and generate some charts by yourself: <http://jira.dh-towerdefense.de/projects/DHTD/issues?filter=allissues>

Hopefully, that is okay for you 😊

Kind regards,  
Nicolas

Antworten



**GreenClothaWay** sagt:  
26. Mai 2020 um 8:39 Uhr

Hey guys,

you did a good job. I get that its hard to implement design patterns because lots of them already existing.

Best regards,  
GreenClothaWay-Team

Antworten



**Nicolas Wagner** sagt:  
27. Mai 2020 um 14:39 Uhr

Hi Team GreenClothaWay,

thanks for your feedback and for agreeing with our decisions concerning the design patterns!

Kind regards,  
Nicolas

Antworten

## 18 Blog Entry Week 17: Metrics

### 18.1 Metrics – what is that?

Hi everyone and welcome back to our blog!

This week, we are dealing with metrics. For analyzing our metrics, we make use of an Android Studio plugin/tool called "[Metrics-Reloaded](#)"<sup>[54]</sup>. This plugin offers a lot of possibilities to analyze the metrics of a project.

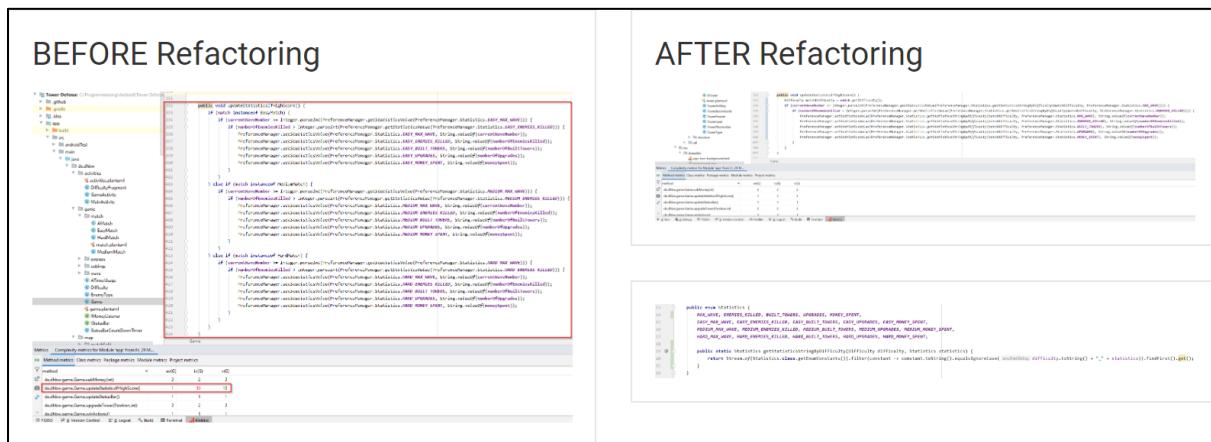
In particular, we want to deal with three metrics this week: Cyclomatic Complexity (CYC), JUnit Test Metric and JavaDocs Coverage Metric. Explanations to those metrics and refactoring examples in our project follow below.

### 18.2 Cyclomatic Complexity

The cyclomatic complexity measures the complexity of code. For this, the code is structured into a flow graph diagram. In this diagram, edges, nodes and exit points are counted and calculated into the complexity value. On the website [tutorialspoint.com](http://tutorialspoint.com), "[What is Cyclomatic Complexity?](#)"<sup>[53]</sup> is explained very well.

In our tower defense project, just a few methods with high complexity values exist. In the following, we refactored our method "Game.updateStatisticsIfHighScore()". Before refactoring, the method got a complexity value of 10. After refactoring, the complexity was reduced to 3.

(Open the screenshots in a new tab for better resolution.)



The "updateStatisticsIfHighScore()" method will update the high score for the currently played game-difficulty if the scores of the current game are higher than the saved high scores.

Before refactoring, the difficulty of the match had to be determined by checking the type of the match using "instanceof". According to the defined difficulty, the new score got saved by the explicitly defined key (e.g. `EASY_MAX_WAVES`, `MEDIUM_ENEMIES_KILLED`, ...). In fact, always the same code was runned using different keys for saving the values.

After refactoring, we pass on the difficulty of a match as well as the required high score type to a method which generates the needed keys for saving the score values. This way, we are able to save up about two thirds of code when looking at the `updateStatisticsIfHighScore` method.

Probably, this solution still can be improved but it shows pretty well how the cyclomatic complexity can be reduced by a lot.

You can also check out our [corresponding commit](#)<sup>[49]</sup> for all changes concerning this metric-refactoring.

### 18.3 Junit Testing Metric

Another metric for analyzing one's code is called "JUnit Testing Metric". As the name already states, numbers concerning JUnit tests are used in order to calculate values.

As an example, our plugin offers two specific values per class: "Number of JUnit test assertions" (JTA) and "Number of JUnit test methods" (JTM). Beside the total amount of both measurements, the tool provides some average values so that these values help to track the density of testing.

Up to now, we got a JTA of 16 (total) or 0.29 (average) and a JTM of 13 (total) or 0.24 (total). A before/after comparison of some refactoring and addition of tests can be viewed below.

BEFORE Refactoring				AFTER Refactoring			
				Metrics JUnit testing metrics for Project 'Tower Defense' fro...			
				Class metrics Package metrics Module metrics Project metrics			
class		JTA	JTM	class		JTA	JTM
de.dhbw.towerdefense.BulletTest		1	1	de.dhbw.towerdefense.BulletTest		6	3
de.dhbw.towerdefense.EnemyTest		3	3	de.dhbw.towerdefense.EnemyTest		3	3
de.dhbw.towerdefense.LaserRayTest		7	5	de.dhbw.towerdefense.LaserRayTest		7	5
de.dhbw.towerdefense.steps.Test_Steps_StartTheGameF		0	0	de.dhbw.towerdefense.steps.Test_Steps_StartTheGameF		0	0
de.dhbw.towerdefense.Test_StartTheGameFromTheMen		0	1	de.dhbw.towerdefense.Test_StartTheGameFromTheMen		0	1
de.dhbw.towerdefense.TowerRotationTest		3	1	de.dhbw.towerdefense.TowerRotationTest		3	1
de.dhbw.towerdefense.TowerTest		2	2	de.dhbw.towerdefense.TowerTest		2	2
de.dhbw.util.Constants		0	0	de.dhbw.util.Constants		0	0
de.dhbw.util.DifficultyFragmentAdapter		0	0	de.dhbw.util.DifficultyFragmentAdapter		0	0
de.dhbw.util.Direction		0	0	de.dhbw.util.Direction		0	0
de.dhbw.util.Position		0	0	de.dhbw.util.Position		0	0
de.dhbw.util.PreferenceManager		0	0	de.dhbw.util.PreferenceManager		0	0
de.dhbw.util.PreferenceManager.Statistics		0	0	de.dhbw.util.PreferenceManager.Statistics		0	0
de.dhbw.util.SortingUtil		0	0	de.dhbw.util.SortingUtil		0	0
<b>Total</b>		<b>16</b>	<b>13</b>	<b>Total</b>		<b>21</b>	<b>15</b>
Average		0,29	0,24	Average		0,38	0,27

In particular, we created some more JUnit tests in order to cover most implementations of our tower-shooting-functionalities.

As a result, both values increased so that we got a JTA of 21 (total) or 0.38 (average) and a JTM of 15 (total) or 0.27 (average) now.

The commits [add-JUnit-Test-Bomb](#)<sup>[50]</sup> and [add-JUnit-Test-SnowFlake](#)<sup>[51]</sup> contain the recent changes.

Resulting, one can say that the higher those values, the better the test coverage and the metric values.

### 18.4 JavaDocs Coverage Metric

As a third metric, we want to introduce into the JavaDocs Coverage Metric. This metric is a very simple one and points out which parts of code still need some documentation. Those values can be generated in total as well as in percentage (see screenshots below).

In our project, we strictly followed some rules of clean code from the beginning and named methods most often meaningfully. Because of that, JavaDocs were redundant. In order to show to you how this metric works, you can find a comparison of before/after code below.

BEFORE Refactoring						AFTER Refactoring						
Metrics Javadoc coverage metrics for Project 'Tower Defens...'						Metrics Javadoc coverage metrics for Module 'app' from Sa...						
	Method metrics	Class metrics	Interface metrics	Package metrics	Module metrics		Method metrics	Class metrics	Interface metrics	Package metrics	Module metrics	
package		Jc	Jf	JLOC	Jm		package		Jc	Jf	JLOC	Jm
de.dhbw.activities	0,00%	3,12%	9	4,17%		de.dhbw.activities	0,00%	3,12%	9	4,17%		
de.dhbw.game	0,00%	0,00%	0	0,00%		de.dhbw.game	0,00%	0,00%	0	0,00%		
de.dhbw.game.match	0,00%	0,00%	0	0,00%		de.dhbw.game.match	0,00%	0,00%	0	0,00%		
de.dhbw.game.popups	0,00%	0,00%	0	0,00%		de.dhbw.game.popups	0,00%	0,00%	0	0,00%		
de.dhbw.game.settings	0,00%	0,00%	0	0,00%		de.dhbw.game.settings	0,00%	0,00%	0	0,00%		
de.dhbw.game.wave	0,00%	0,00%	0	0,00%		de.dhbw.game.wave	0,00%	0,00%	0	0,00%		
de.dhbw.map.matchfield	0,00%	0,00%	16	13,64%		de.dhbw.map.matchfield	0,00%	0,00%	94	100,00%		
de.dhbw.map.objects.bullet	0,00%	0,00%	12	8,51%		de.dhbw.map.objects.bullet	14,29%	0,00%	248	81,25%		
de.dhbw.map.objects.enemy	0,00%	0,00%	10	4,88%		de.dhbw.map.objects.enemy	0,00%	0,00%	173	92,68%		
de.dhbw.map.objects.tower	0,00%	0,00%	27	6,52%		de.dhbw.map.objects.tower	0,00%	0,00%	271	61,80%		
de.dhbw.map.structure	0,00%	0,00%	36	33,33%		de.dhbw.map.structure	0,00%	9,09%	85	82,61%		
de.dhbw.towerdefense	0,00%	0,00%	0	0,00%		de.dhbw.towerdefense	0,00%	0,00%	0	0,00%		
de.dhbw.towerdefense.steps	0,00%	0,00%	0	0,00%		de.dhbw.towerdefense.steps	0,00%	0,00%	0	0,00%		
de.dhbw.util	0,00%	0,00%	5	4,76%		de.dhbw.util	14,29%	0,83%	89	94,74%		
<b>Total</b>			<b>115</b>			<b>Total</b>			<b>969</b>			
Average	0,00%	0,29%	8,21	6,10%		Average	3,45%	0,87%	69,21	47,41%		

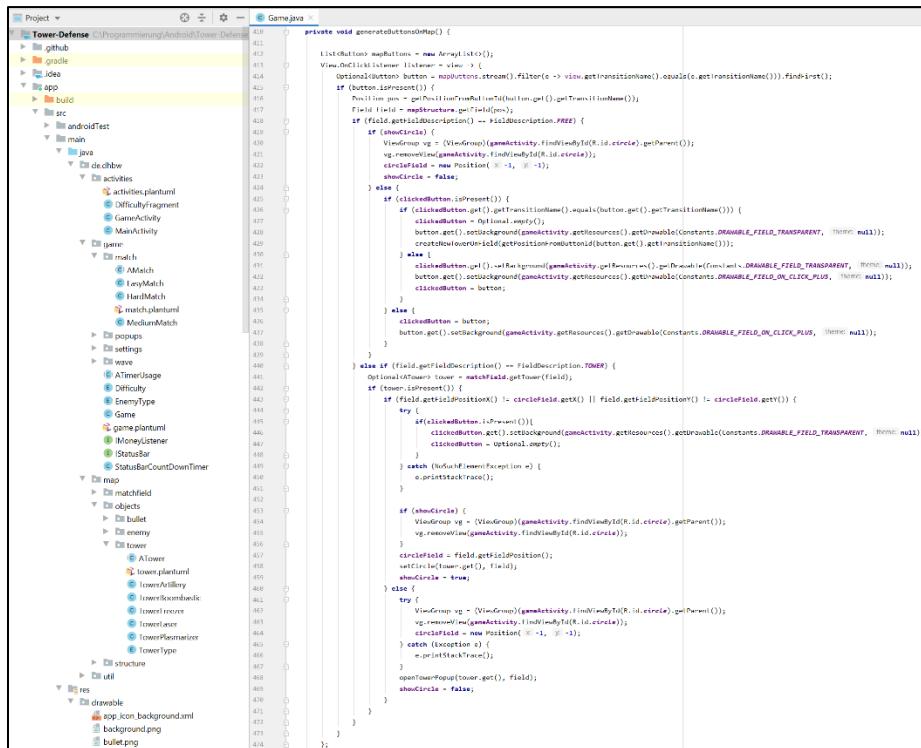
The [corresponding commit](#) [52] can be viewed on GitHub as well.

As one can see, our average Javadoc coverage (right column) increased from 6.1% to 47.41% while adding JavaDocs for about the half of all methods/packages.

In the metric overview, it is pretty useful to see what packages/classes/methods/interfaces still need some JavaDocs.

All in all, this metric is not that important in our case but probably, it is a nice example to demonstrate because of its easy way of increasing quality.

## 18.5 Code which is not going to be refactored



```

private void generateButtonsOnMap() {
    List<ViewGroup> viewGroups = new ArrayList();
    View.OnClickListener listener = view ->
        (Optional<Button> button = buttonGroups.stream().filter(o -> view.getTransitionName().equals(o.getTransitionName())).findFirst());
        if(button.isPresent()) {
            if(button.get().getTag() == null) {
                button.setTag(button.get().getTransitionName());
            }
            if(button.get().getTag().equals(button.get().getTransitionName())) {
                if(button.get().getTag().contains("circle")) {
                    ViewGroup vg = (ViewGroup)gameActivity.findViewById(R.id.circle);
                    vg.removeAllViews();
                    vg.addView(button);
                    vg.setDrawingCacheEnabled(true);
                    vg.setWillNotDraw(true);
                } else {
                    if(button.get().getTag().contains("button")) {
                        if(button.get().getTag().equals(button.get().getTransitionName())) {
                            if(button.get().getTag().contains("circle")) {
                                ViewGroup vg = (ViewGroup)gameActivity.findViewById(R.id.button);
                                vg.removeAllViews();
                                vg.addView(button);
                                vg.setDrawingCacheEnabled(true);
                                vg.setWillNotDraw(true);
                            } else {
                                if(button.get().getTag().contains("square")) {
                                    ViewGroup vg = (ViewGroup)gameActivity.findViewById(R.id.square);
                                    vg.removeAllViews();
                                    vg.addView(button);
                                    vg.setDrawingCacheEnabled(true);
                                    vg.setWillNotDraw(true);
                                } else {
                                    if(button.get().getTag().contains("triangle")) {
                                        ViewGroup vg = (ViewGroup)gameActivity.findViewById(R.id.triangle);
                                        vg.removeAllViews();
                                        vg.addView(button);
                                        vg.setDrawingCacheEnabled(true);
                                        vg.setWillNotDraw(true);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Metrics Complexity metrics for Module 'app' from Fr. 20 M...

	Method metrics	Class metrics	Interface metrics	Package metrics	Module metrics	Project metrics
method	ev(G)	iv(G)	v(G)			
de.dhbw.game.Game(GameActivity)	1	1	1			
de.dhbw.game.Game.generateButtonsOnMap()	1	10	19			

Above, you can see our example of code which is not going to be refactored in order to reduce its complexity value of 19.

The "generateButtonsOnMap()" method is responsible for generating the field structure. It is one of our most fundamental game engine methods which is never going to be touched anymore because the time needed for refactoring the method is disproportionate to the profit. The method works fine and we have not seen any bugs in the past.

## 18.6 Metrics – Conclusion

All in all, we can summarize that metrics can help to find weak code spots in general. On the other hand, code snippets highlighted by the tool need to be analyzed by a developer first in order to decide whether the metrics should be improved or not.

In our case, the metrics analysis is only performed locally because most common metrics do not fit very well into the style of programming in the Android framework. As we do have our own conventions which are adapted to programming Android apps, this decision is okay in our opinion.

## 18.7 Big Progress in our CI/CD

Beside the use of metrics, a good CI/CD (Continuous Integration / Continuous Delivery) pipeline becomes more and more important.

Until now, our CI was not really good. Instead of setting up a good CI tool (such as Jenkins or Travis), we used the GitHub Actions functionality where we have set up an Android CI with a third party tool. Though the tool built and tested the tower defense project, it acted as a black box and nobody was able to have a closer look into what is going on there. Of course, we could have done this better earlier, but as our team wanted to push ahead the progress of the tower defense game first and as nobody of us had any experience with CI/CD tools, the previous solution was obviously the best one for us. But now, this changed!

### Introduction into our use of TravisCI

TravisCI is one of the most popular tools when talking about CI/CD. This is actually the reason for why we decided on using this tool from now.

After a lot of hours dealing with getting Travis to build our Android application, we finally managed to have an CI which does the following:

1. Clean and build the Android project
2. Run all JUnit tests (instrumentation tests do not work out properly with our required SDK version)
3. Generate an apk file (Android installation file)
4. Send a push notification to our discord of build/test status for a better overview and an active flow of information
5. deploy the app to our release section on GitHub so that one can download the app to one's own Android device and test it in reality

As you might imagine, this is a big step for us and enables a lot of possibilities in the future (e.g. automatic deployment of the app to Google Play Store).

Of course, all those new information are added in our testplan document on GitHub which also contains a badge with the current build-status of our app such as the following:

 build passing

### 18.8 New Tower: Plasmarizer

A recently added tower is called Plasmarizer. The Plasmarizer is a tower with a special shooting mechanism. It shoots a PlasmaBall onto the nearest targetable enemy in range. As soon as the PlasmaBall hits the targeted enemy, the bullet can jump to other enemies in a specific range. With each further hit enemy, the damage of the ball decreases.



### 18.9 New Tower: Assaultlaser

Another new tower is called Assaultlaser. This tower does not shoot a typical bullet as all other towers. As the name already says, the Assaultlaser creates a LaserRay. The LaserRay focuses the nearest enemy in range and shoots as long as the targeted enemy is damaged by the LaserRay. If the enemy moves out of the LaserRay or if it becomes killed, the AssaultLaser will stop shooting and searches for the next enemy. If not explicitly targeted enemies move into the LaserRay, those enemies will also be damaged.

To be honest, this tower can become very strong when positioning the tower cleverly. Due to this fact, the game balancing (which will come up very soon) needs to be done carefully.

Because pictures cannot demonstrate the whole progress up to now, we uploaded our current app to [appetize.io](https://appetize.io) so that you can play by yourself, if you want. Here is the link:

<https://appetize.io/app/w4nbd746n40k2wajrfzp03vpuc>



## 18.10 Summary

All in all, we are happy again to present so many new stuff to you. Beside the progress in the game itself, we managed to make use of metrics and we set up a well working CI. In fact, this means that most requirements of a good software engineering project are fulfilled and we can focus on the following tasks from now:

- game balancing
- implementation of some details
- optimization
- usability tests
- getting the project ready for the final presentation

That's it for now. See you next week!

## 18.11 Comments



Alexandra Stober sagt:  
3. Juni 2020 um 0:44 Uhr

Hello Team Tower Defense,  
I really enjoyed reading your blog entry. It was well structured and clear which metrics you chose, how you implemented these changes and why.  
You also explained a spot in your code where metrics suggest that you should implement a change, but you decided not to. All in all, well done. I think you fulfilled the grading criteria.  
Your progress in your CI/CD, is an important step and it is great that you got it to work even though none of you has experience with setting CI/CD up. You mentioned the google play store. Do you plan to publish your app once it is done?  
It was interesting to read about your future plans. I look forward to reading your next blog.  
Kind regards,  
bookly

Antworten



Nicolas Wagner sagt:  
4. Juni 2020 um 13:37 Uhr

Hi Team bookly,  
thank you for the extensive feedback!  
At the moment, our plan is to publish our app in the Google Play Store after the provisional end of development (in a few weeks).  
We will keep you up to date :)!  
  
Kind regards,  
Team Tower Defense

Antworten



Sam sagt:  
3. Juni 2020 um 11:01 Uhr

Hi there, in short we do not have anything to complain.  
You mention two metrics and explain them very well afterwards,  
including the improvements you guys have gained due raising these.  
– Team cozy

Antworten



Nicolas Wagner sagt:  
4. Juni 2020 um 13:34 Uhr

Hi Team cozy,  
thanks for your feedback concerning the use of metrics!  
  
Kind regards,  
Team Tower Defense

Antworten

## 19 Blog Entry Week 18: Optimization

### 19.1 Introduction

Hi everyone and welcome back to our blog!

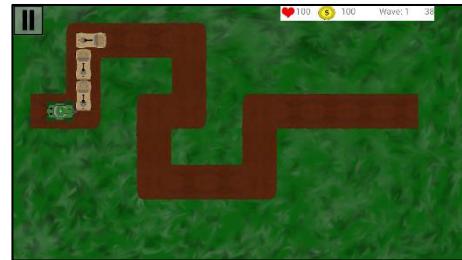
This week, we are not going to deal with any specific topic. However, that does not mean that we do not work on anything ;)!

As we are working on several optimizations concerning our code, there is not much to show today. Nonetheless, we are happy to present 3 topics to you which we worked on: "New Enemy: Car", "New Music" and "UX Test Preparation".

### 19.2 New Enemy: Car



On the left, you can see the newly implemented enemy: Car. As simple as it sounds, the car is just another enemy with differing values. For example, the car moves faster but has less lifepoints in return. On the right, you can see how the cars look like in



the real game.

### 19.3 New Music

Beside the work of last weeks, our team member Fabian worked on 4 new soundtracks as ingame music. At the moment, those are not finished yet. However, we want to present 2 of them to you today:

[Audio<sup>\[82\]</sup>](#)

[Audio<sup>\[83\]</sup>](#)

The first soundtrack does not have the same dark atmosphere as the other 3 soundtracks, but we think that it gives a good variety. If you want to explore all soundtracks, you have to listen to them in the game 😊

Enjoy!

### 19.4 UX Test Preparation

Very soon, the tower defense app achieves the status in which it can be tested by outsiders. As we already decided in the past, UX (usability) tests seem to be most sensful in our case as a third test type.

Before the app becomes delivered to selected testers, a UX testplan has to be created. In our elaboration, the testplan is reflected on a Google formular sheet, which can be accessed using the following link: <https://forms.gle/JjKckiCuGKEAeqC56>

This formular begins with a small instruction on how to download and install the app and goes on over several app-specific questions. This process is divided into many sections such as "Installation", "Play!", "Main Menu", "Ingame", "Popups" and "Overall game". For each section, we provide some questions so that hopefully after some UX tests, we get some feedback on which components of the app we still have to work on.

If you want, you can already follow the Google formular link above in order to find the current download link for installing the app. If you can free some time and own an Android device, we would be really happy and appreciate if you could run through the UX test as well. By doing that, you would help us a lot! But if you cannot run through the test right now, that's okay as well!

That's it for now. See you next week!

## 19.5 Comments



**Philipp@Keycloud** sagt:  
9. Juni 2020 um 9:35 Uhr

Hi folks,  
I spent some minutes testing your game.  
Let me provide some feedback which did not find a correct spot inside your UX testplan.

- You should add you guys as app developer so one popup less at installation.
- For me it was not intuitive how I can place towers. How should I have known that I have to click an empty spot first?
- Please, add a speed up button
- For me filthy casual one questions comes up: Is it even possible to win hard mode? I mean you cannot control the focus of the towers
- The enemies and towers look quite good but the map itself needs some love ...

Best regards,  
Philipp@KeyCloud



**Nicolas Wagner** sagt:  
10. Juni 2020 um 16:48 Uhr

Antworten

Hi Philipp,  
thanks a lot for your feedback! This extensive feedback and test help us a lot improving our app!  
Concerning your mentioned additions:

- The whole installation process will move to Google Play in the future which will include the signation of the app. But agreed: This is a necessary adjustment.
- What would you suggest how a tower should be built? Without making use of drag and drop, this solution seemed most intuitive to us.
- I feel your pain :D. Due to our current engine, it is not possible to add this function until the final hand in. But it's definitely on our ToDo list for further improvements.
- Every difficulty becomes designed so that one can win the game – „difficult“ as well ;). It depends a lot on how you place towers and how you combine the special functionalities of them. That's actually also a point which makes the game different in comparison to other tower defense games.
- The map design is on our ToDo list.

Again, thanks for your detailed feedback! We will keep working on that!

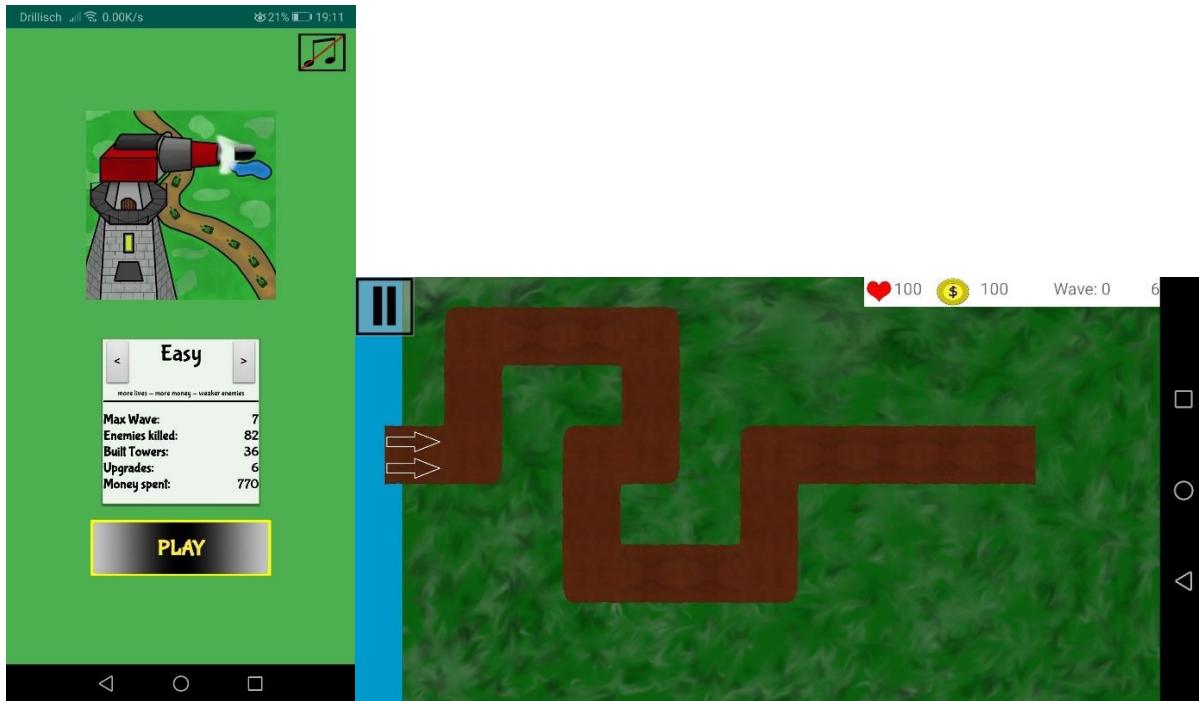
Kind regards,  
Team Tower Defense



**Nicolas Wagner** sagt:  
11. Juni 2020 um 19:18 Uhr

Antworten

For the sake of completeness, here are the screenshots of the installed app provided by Philipp of team KeyCloud:



## 20 Blog Entry Week 19: Retrospective

### 20.1 Retrospective

Hi everyone and welcome back to our blog!

As already talked about in week 7, we had another retrospective this week guided by an external coach. Again, three questions had to be thought of:

1. What went well?
2. What can be improved?
3. How can certain things be improved?

Looking at our new results below, one can see that the mentioned aspects differ in comparison to the aspects mentioned in week 7. Let's have a look at our new aspects!

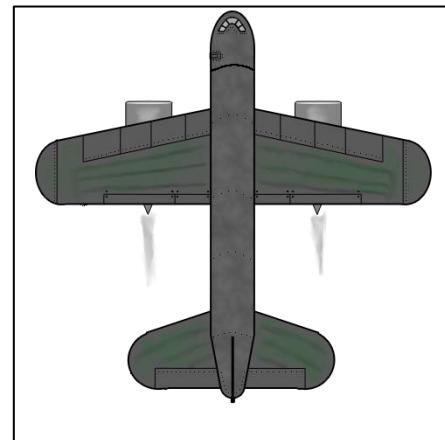
### 20.2 Solutions

Positive Aspects	Negative Aspects	Actions
<b>TowerDefense:</b> Planung, Teamarbeit, Programmieren, Blog, Aufgabenverteilung, Kreativität	<b>TowerDefense:</b> -sehr hoher workload -schwere behebbare bugs	<b>TowerDefense:</b> - Grundbausteine zu 3. legen - mehr mit Jira von Anfang an arbeiten - andere Frameworks zur Hilfe nehmen
<ul style="list-style-type: none"> <li>• Planning</li> <li>• Teamwork</li> <li>• Programming</li> <li>• Blog</li> <li>• Distribution of tasks</li> <li>• Creativity</li> </ul>	<ul style="list-style-type: none"> <li>• very high workload</li> <li>• bugs which are difficult to fix</li> </ul>	<ul style="list-style-type: none"> <li>• work on the elementary components with all team members</li> <li>• use Jira as project management tool from the beginning</li> <li>• make use of other frameworks</li> </ul>

### 20.3 New Enemy: Plane

Another achievement in this sprint is the newly implemented plane-enemy. Planes are very different in comparison to tanks and cars:

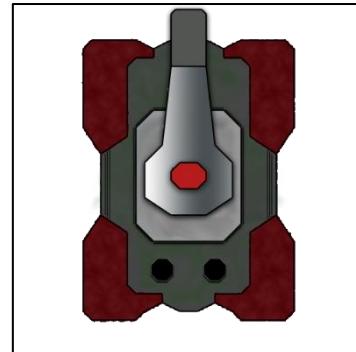
- planes spawn on a random field on the left side of the match field
- they do not follow the path - they just "fly" straight to the right side of the match field
- planes can only be shot by specified towers (at the moment: Artillery and Freezer)



## 20.4 New Boss Enemies

In addition to the planes, we also created new boss enemies which are based on tanks. Those boss enemies are planned to be spawned after some waves so that there is also an enemy which is not that easy to kill.

Fighting against boss enemies, one can remark a pretty unexpected feature: When a boss enemy is about to die, a car enemy will spawn at the position of the boss enemy. This makes the boss stronger than it actually looks like.



## 20.5 Further Adjustments

Last but not least, we worked on some more adjustments. Those are the most important achievements:

- The game of the difficulty "easy" got designed so that 30 waves are playable now.
- Each enemy now has a lifepoints progress bar.
- A "Download" button can be found at the top of this blog. By clicking on it, a recent version of the app will be downloaded as an .apk installation file.
- Some more small bugs became fixed and some code-architectural adjustments have been made.
- Our code quality is available online here: <https://codeclimate.com/github/niwa99/Tower-Defense>

Of course, there is also a badge provided by the tool:



That's it for this week. See you next time!

## 21 Blog Entry Week 20: Installation

### 21.1 Installation

Hi and welcome back to our blog! This blog entry rather deals with some organizational topics in contrast to the reports of several previous weeks. And it's the last one before the final blog entry!

First of all, here is a short **instruction** on how to **install the app on an Android device**:

1. Using your Android device, download the app using the "Download" button at the top of this page or using the following link: <https://download.dh-towerdefense.de>
2. Run the -apk file by clicking on it
3. If your device runs with a rather old Android version, the app will probably just become installed.  
If your device runs with a rather recent Android version, you will have to pass some "security" popups in which you agree that you really want to install our app. This security-check is caused because our app cannot be downloaded in the PlayStore yet.
4. That's it! You can simply run Tower Defense as any other app on your device. Have fun!

(Grading Information: The proof of installation by another team can be found on the blog entry of week 18 by following the link: <https://dh-towerdefense.de/week-18-optimization/#comment-86>)

### 21.2 Running through last Tests

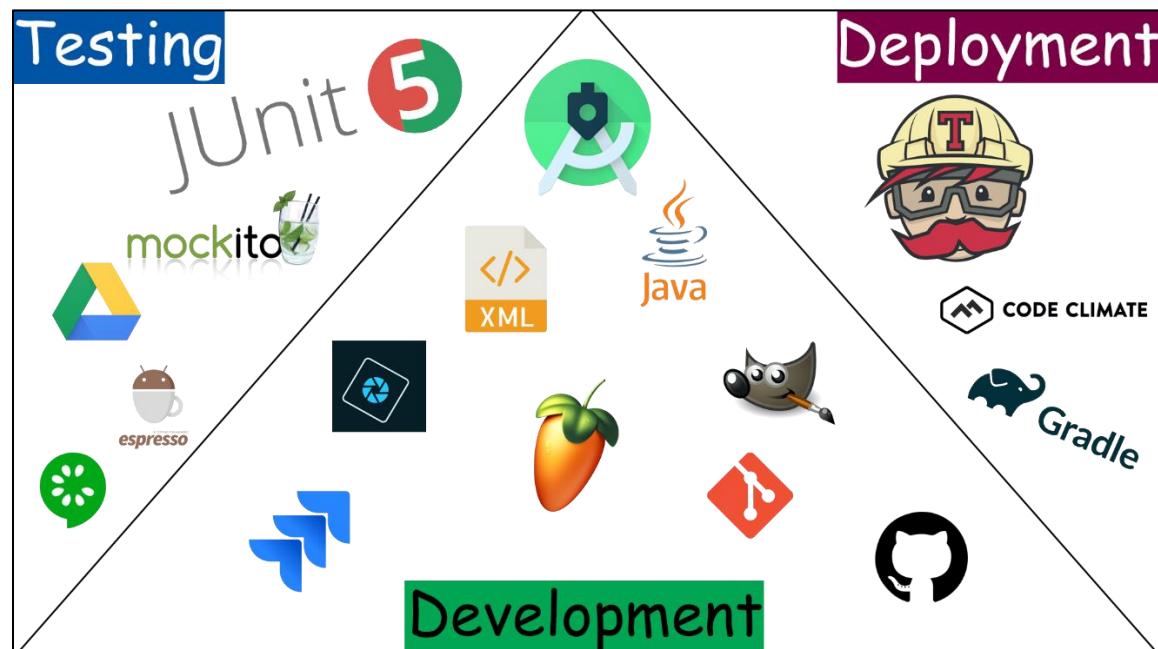
As the time of the project in class is ending soon, we manage to get some people testing our app concerning usability. Usability testing is a very important aspect because it shows up the satisfaction of potential users.

The results of our usability test can be accessed using the following link: <https://ux-test-results.dh-towerdefense.de>

Moreover, the [testplan document](#)<sup>[31]</sup> is finally finished. It was completed by adjusting the workflow and the overall use of CodeClimate and TravisCI.

### 21.3 Tech Stack

Last but not least, we want to show our final tech stack to you. To be honest, it does not seem like we made use of so many tools and technologies throughout the whole project, but here it is:



And now... that's it.

It is our last blog entry with thematic content. Thanks a lot to everybody who followed us throughout both semesters.

Next week, you will get to see the final blog post, which will include links to all blog posts and links to important documents and sites as well.

See you there!

## 22 Blog Entry Week 21: Final

### 22.1 Final – Links

Hi everybody,

as announced last time, today's blog entry contains just all important links to previous blog entries and other important documents and sites. Within 20 weeks, our team was able to achieve the following:

#### **Blog Entries:**

- [Week 01: Vision, Dependencies<sup>\[4\]</sup>](#)
- [Week 02: RUP Team Roles, Technology<sup>\[5\]</sup>](#)
- [Week 03: SRS, UCD<sup>\[6\]</sup>](#)
- [Week 04: Use Case Specifications<sup>\[7\]</sup>](#)
- [Week 05: Feature Files, Cucumber Test Video<sup>\[8\]</sup>](#)
- [Week 06: JIRA Integrations<sup>\[9\]</sup>](#)
- [Week 07: Retrospective \(#1\)<sup>\[10\]</sup>](#)
- [Week 08: Class Diagram, Prototype<sup>\[11\]</sup>](#)
- [Week 09: MVC Model, Feature Files running<sup>\[12\]</sup>](#)
- [Week 10: Midterm Links<sup>\[13\]</sup>](#)
- [Week 11: Revision, new UCD and Use Case Descriptions<sup>\[14\]</sup>](#)
- [Week 12: Risk Management<sup>\[15\]</sup>](#)
- [Week 13: Function Points<sup>\[16\]</sup>](#)
- [Week 14: Testing \(JUnit\)<sup>\[17\]</sup>](#)
- [Week 15: Refactoring Project<sup>\[18\]</sup>](#)
- [Week 16: Design Patterns<sup>\[19\]</sup>](#)
- [Week 17: Metrics, CI/CD-Progress<sup>\[20\]</sup>](#)
- [Week 18: UX Test Preparation \(and proof of Installation comment\)<sup>\[21\]</sup>](#)
- [Week 19: Retrospective \(#2\)<sup>\[22\]</sup>](#)
- [Week 20: Installation, Tech Stack<sup>\[23\]</sup>](#)

#### **All about Code:**

- [Repository on GitHub<sup>\[3\]</sup>](#)
- [TravisCI<sup>\[25\]</sup>](#)
- [Code Analytics on CodeClimate<sup>\[26\]</sup>](#)

#### **Project Management & Statistics:**

- [JIRA Project Management<sup>\[2\]</sup>](#)
- [Statistics and Reports<sup>\[27\]</sup>](#)
- [Excel Time Sheet for Statistics<sup>\[28\]</sup>](#)

#### **Software Specifications:**

- [SRS \(Software Requirements Specification\)<sup>\[29\]</sup>](#)
- [SAD \(Software Architecture Document\)<sup>\[30\]</sup>](#)
- [UCD \(latest Use Case Diagram\)<sup>\[32\]</sup>](#)
- [Testplan<sup>\[31\]</sup>](#)

**Use Cases:**

- UC-01: Start the game from the menu<sup>[33]</sup>
- UC-02: Start waves<sup>[34]</sup>
- UC-03: Build towers on selected area<sup>[35]</sup>
- UC-04: Return to main menu<sup>[36]</sup>
- UC-05: Upgrade towers<sup>[37]</sup>
- UC-06: Sell towers<sup>[38]</sup>
- UC-07: Select between different towers<sup>[39]</sup>
- UC-08: Edit settings<sup>[40]</sup>
- UC-09: Toggle sound and animations<sup>[41]</sup>
- UC-10: Choose difficulties<sup>[42]</sup>

**Feature Files:**

- Feature File: Start the game from the menu<sup>[43]</sup>
- Feature File: Start waves<sup>[44]</sup>

**App online testable:**

- Tower Defense on appetize.io<sup>[45]</sup>

**Final Presentation:**

- Final Presentation (Slides)<sup>[46]</sup>
- Final Handout<sup>[47]</sup>
- Video of Tower Defense App<sup>[48]</sup>

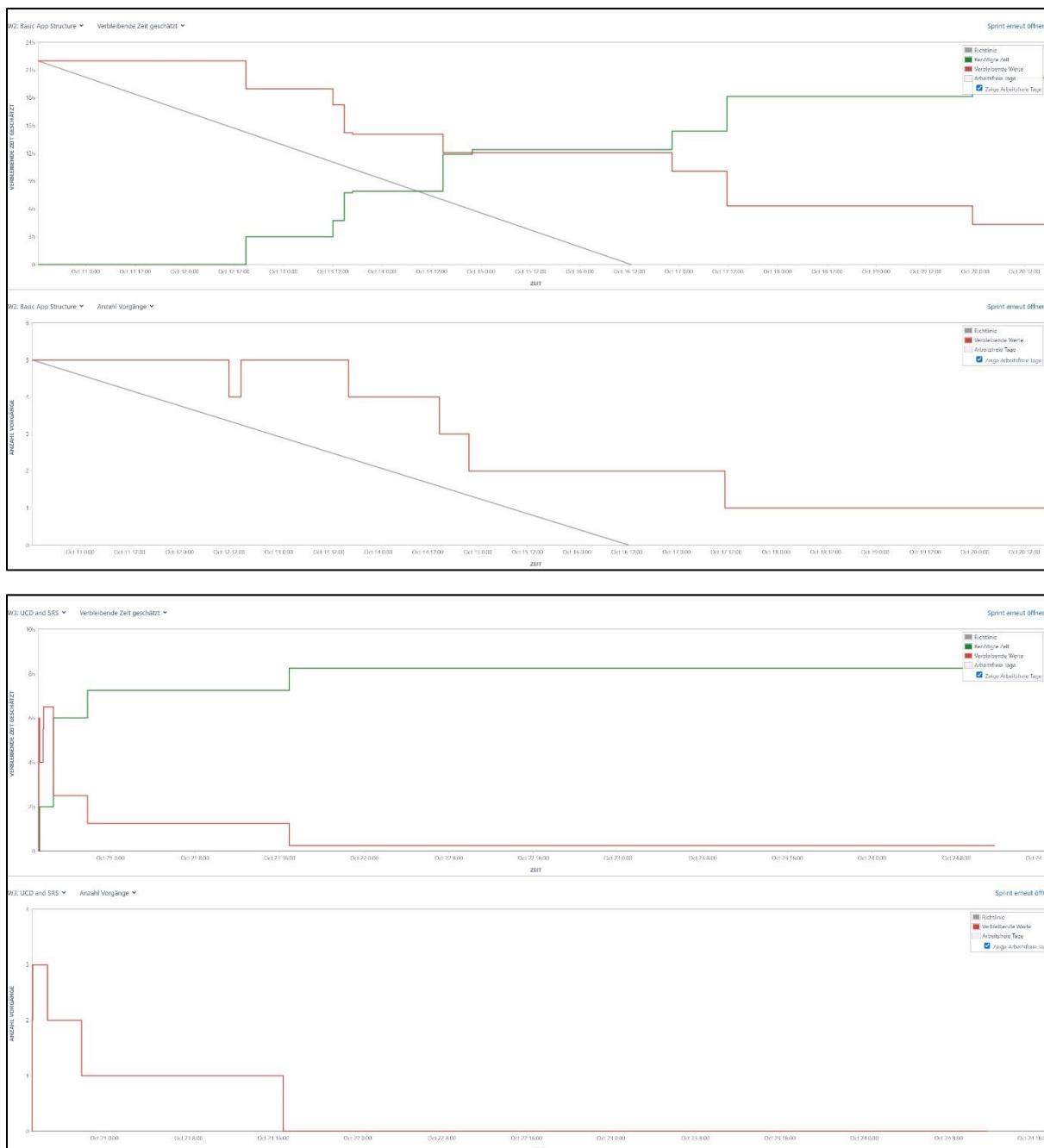
## 23 Statistics

### 23.1 Statistics – Introduction

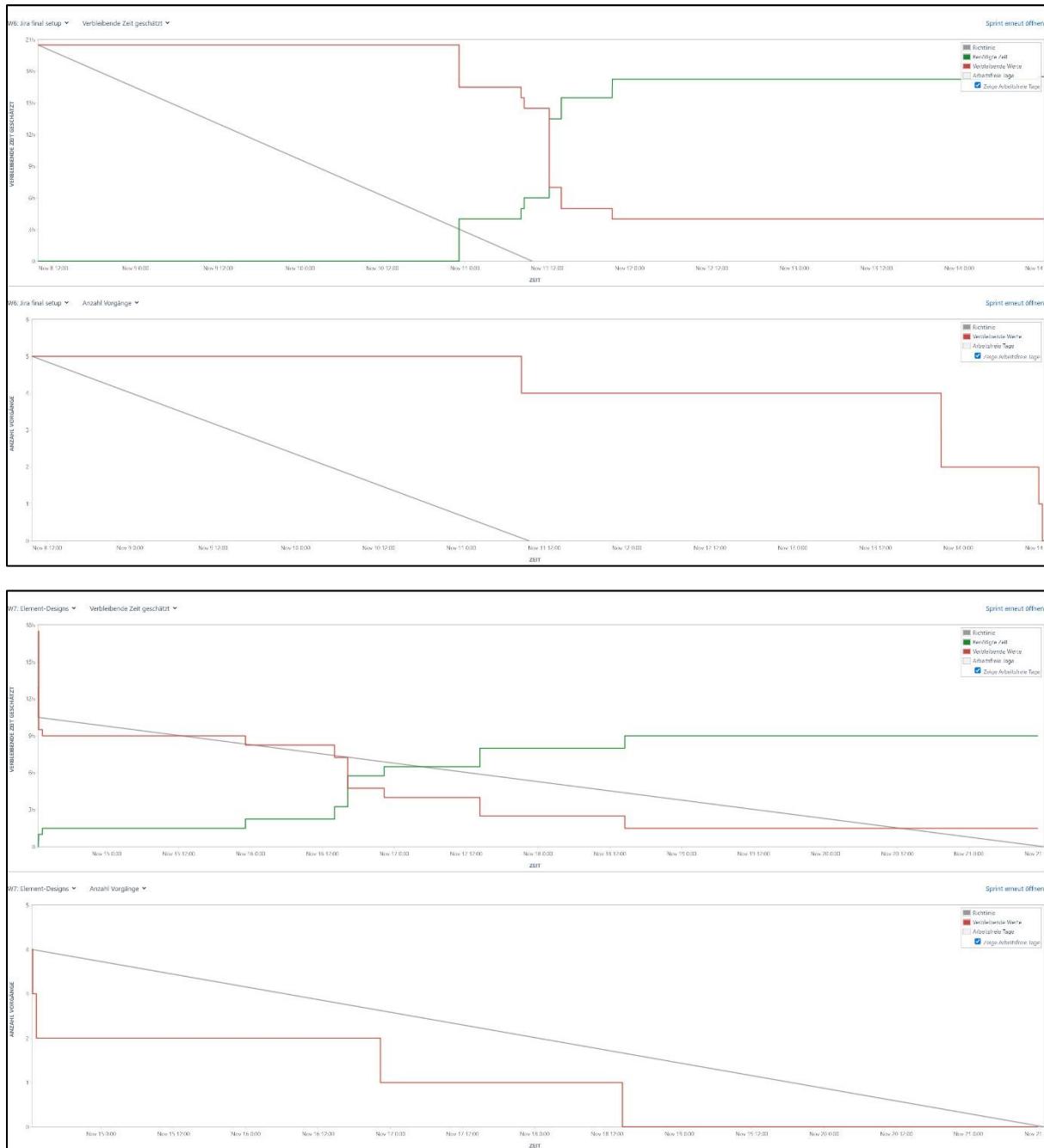
The whole "Tower Defense" project is developed making use of Scrum. Each week of the semester is demonstrated by one single sprint. For the first week, there is not any sprint existing because the project was not set up yet.

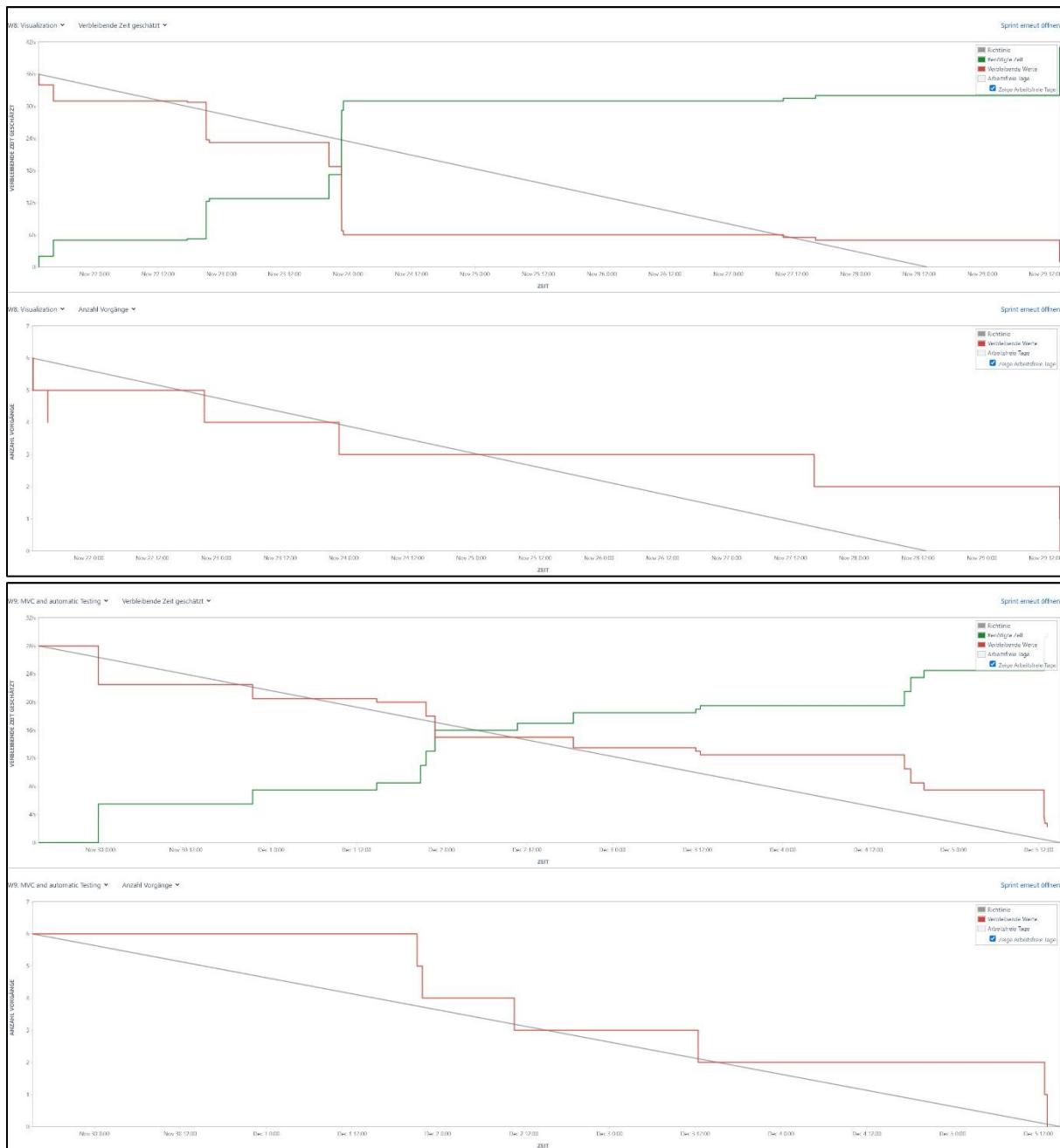
Within the next weeks, our team had to get along with Jira at first which is why our first burndown-diagrams are not pretty nice.

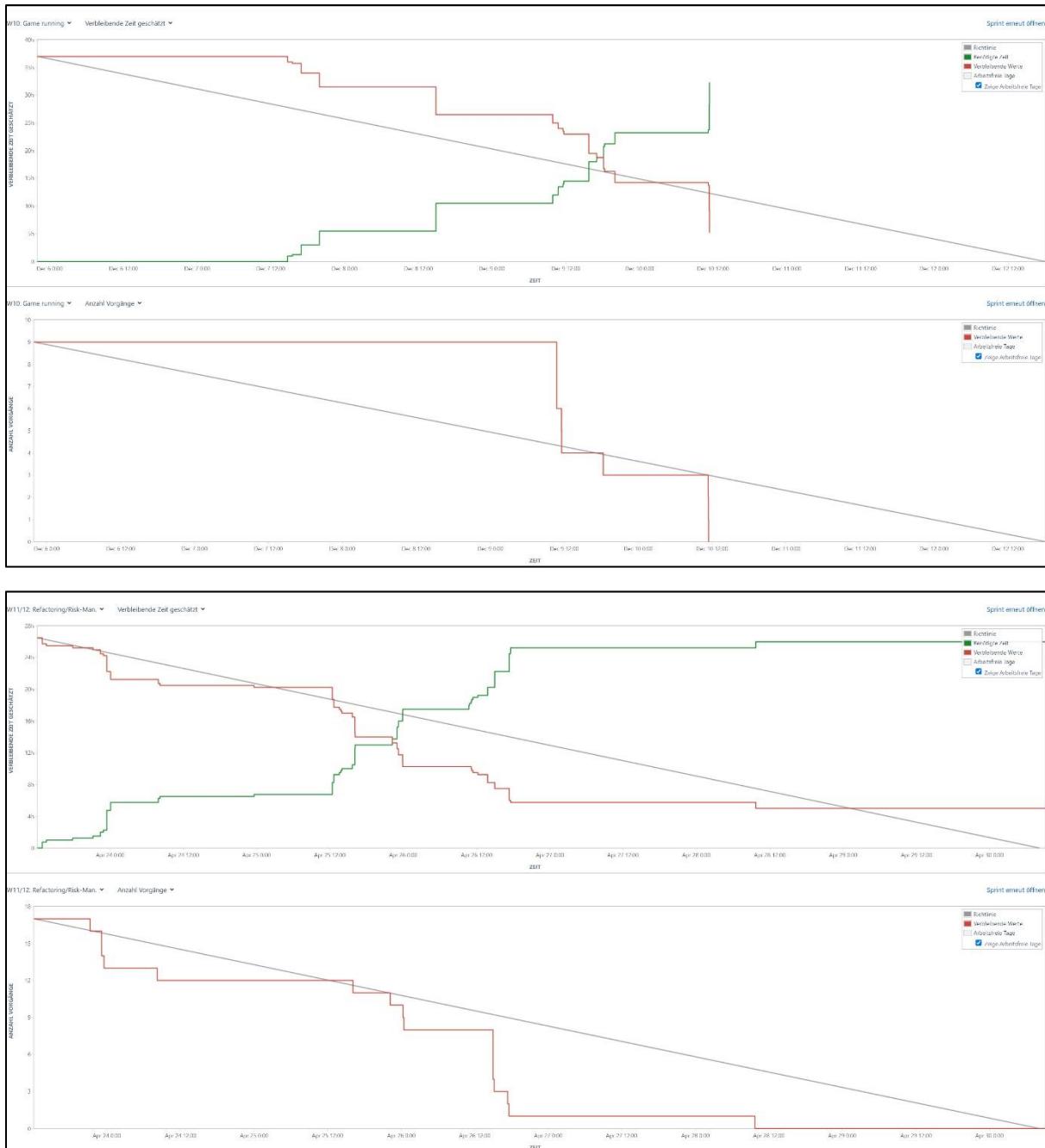
In the following, you can see our burndown-diagrams. (Click on the right to move on to the next burndown) In our opinion, the diagrams of week 13 and 21 look best.

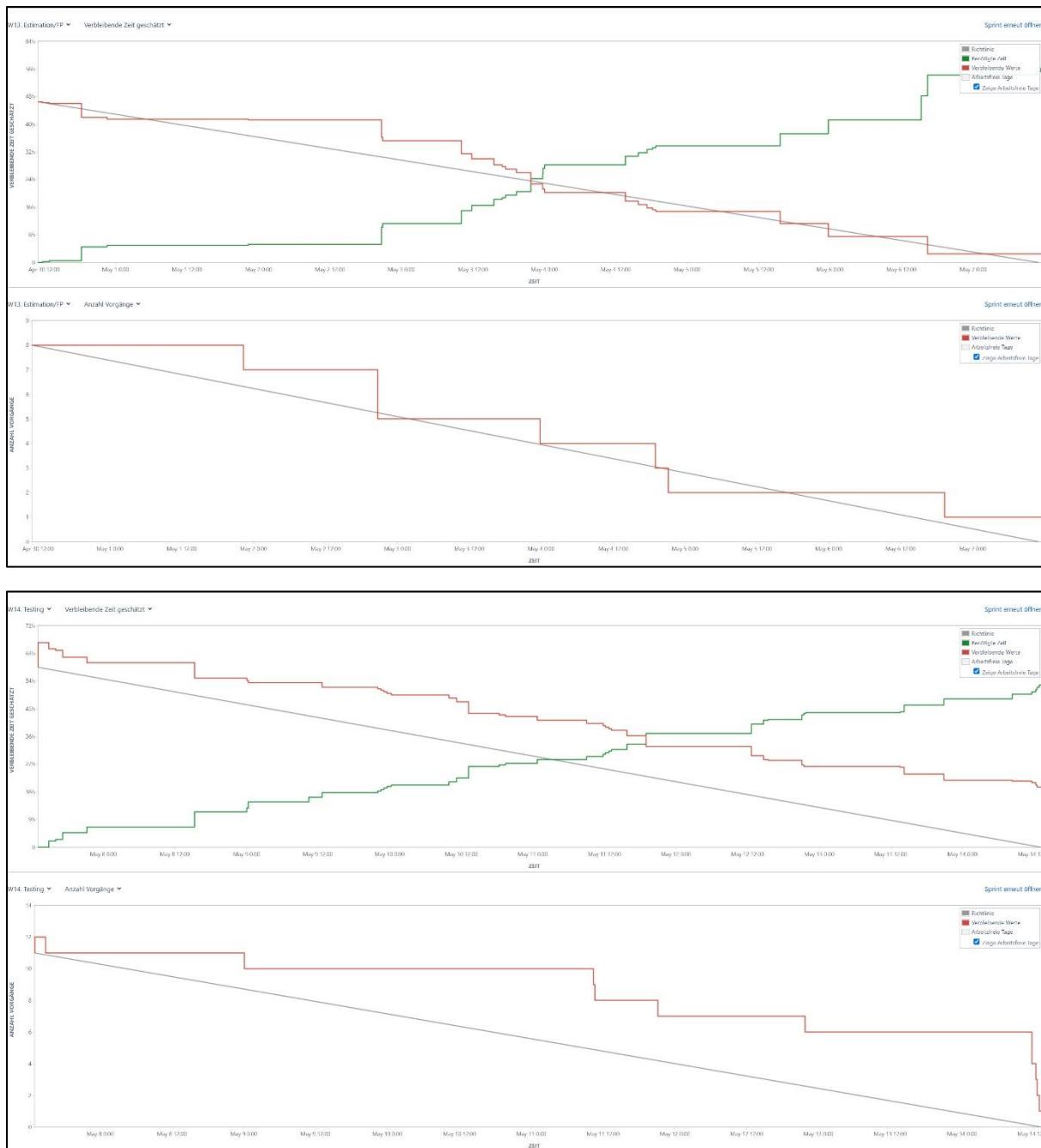


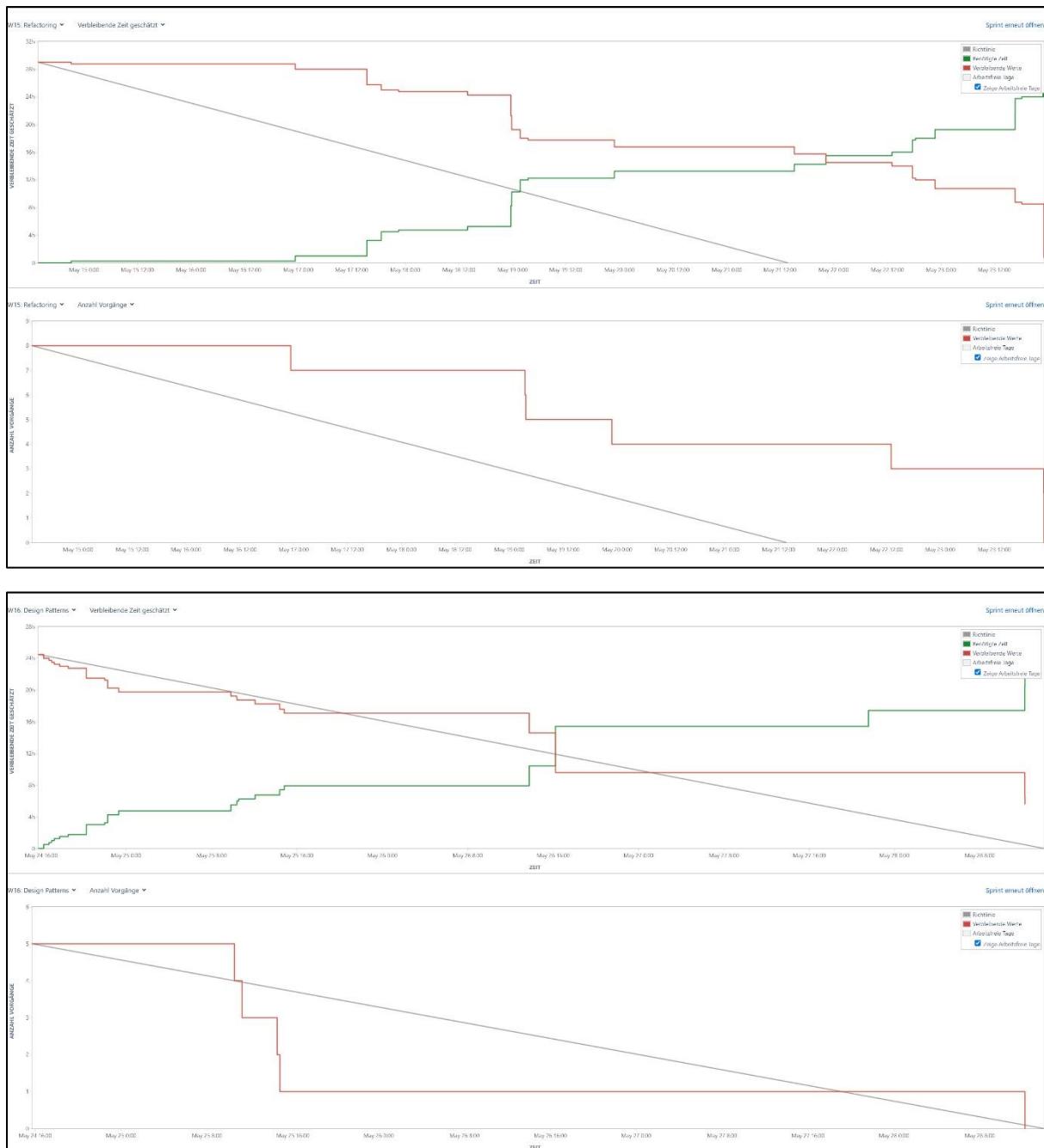


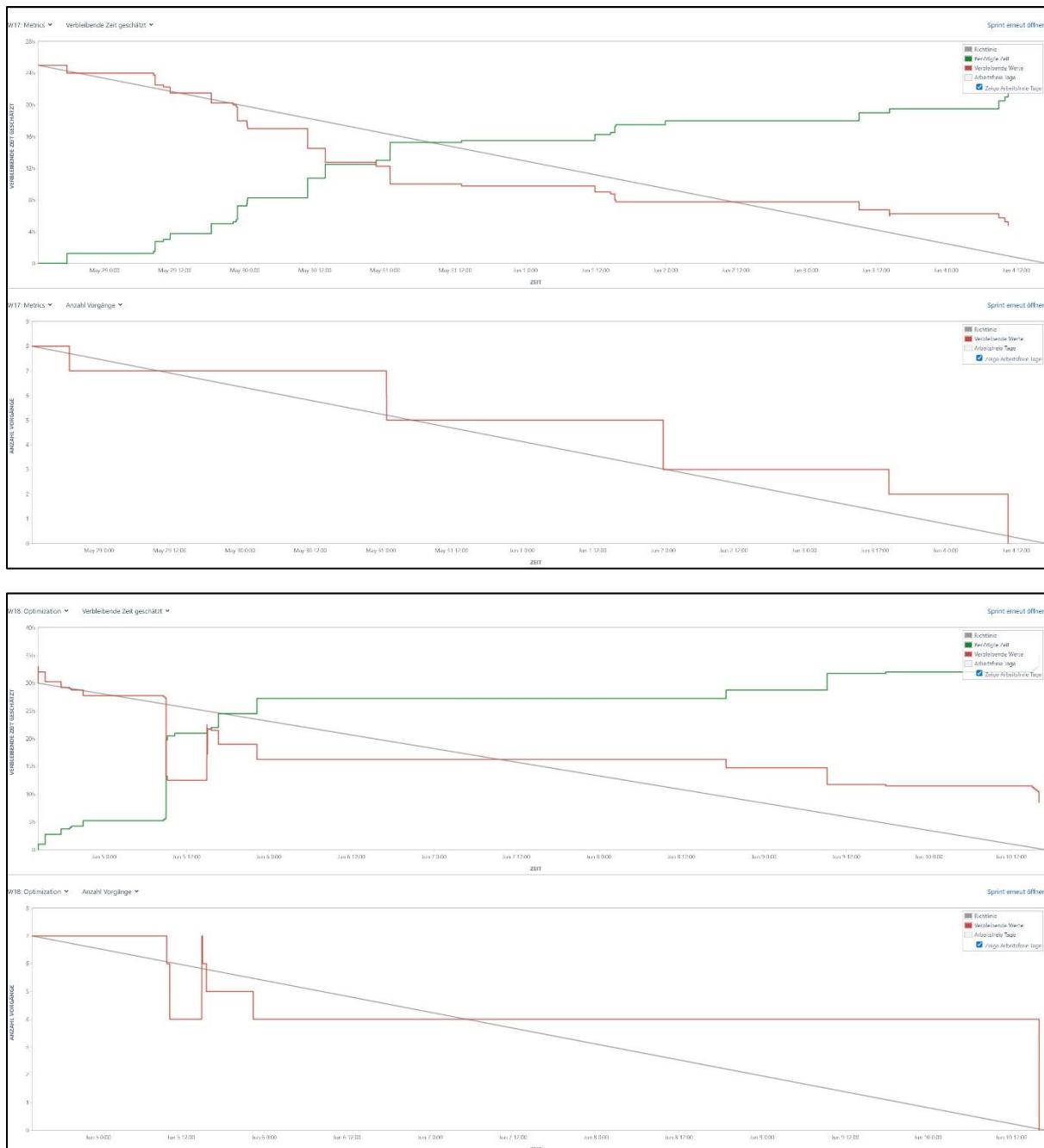


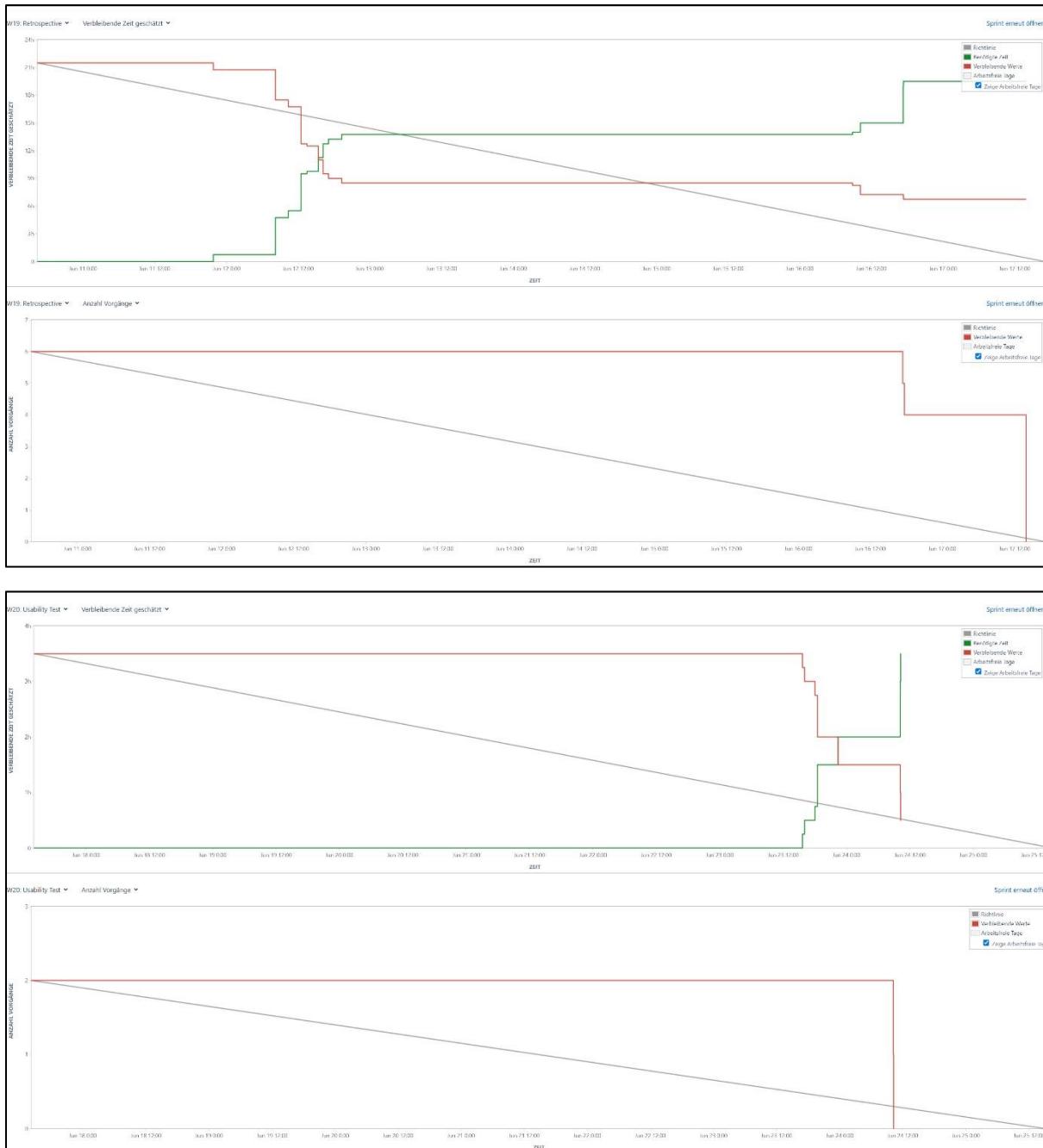










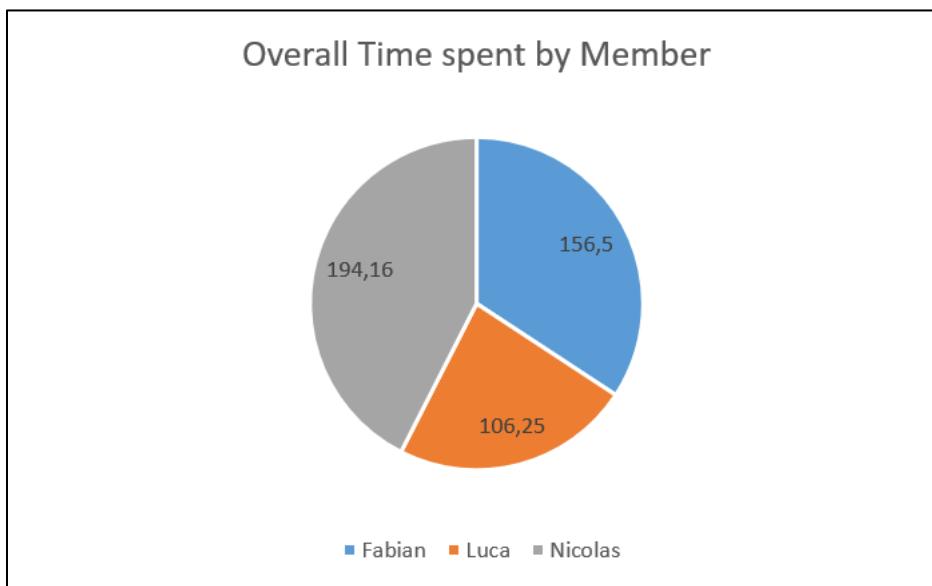


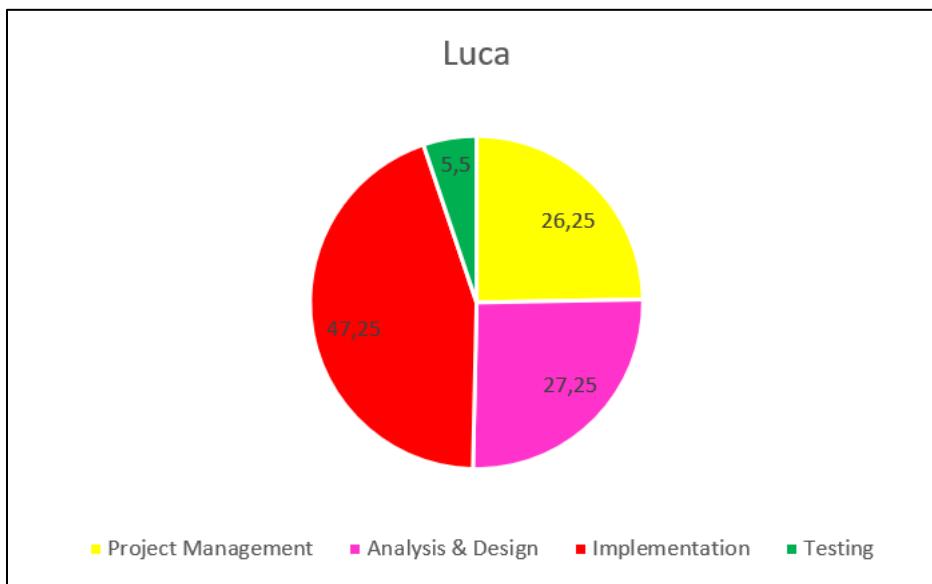
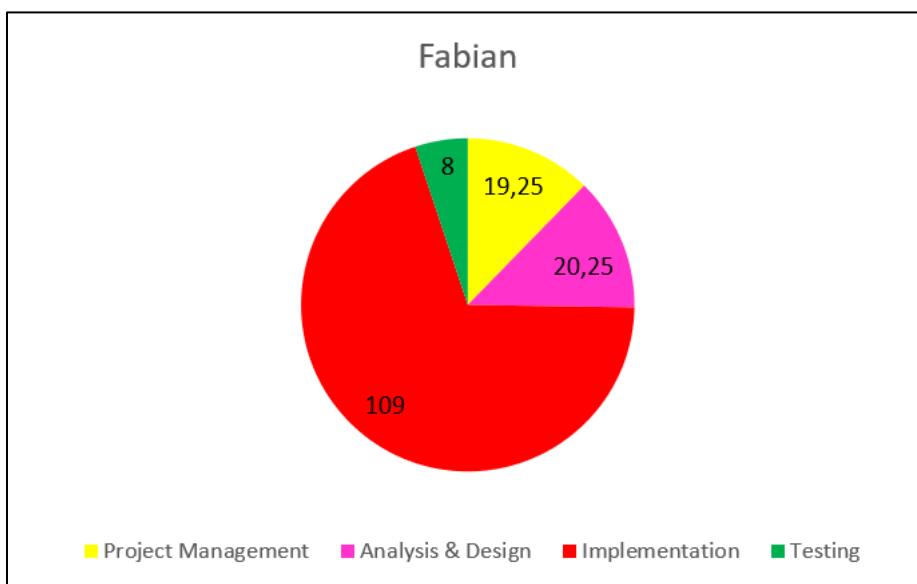
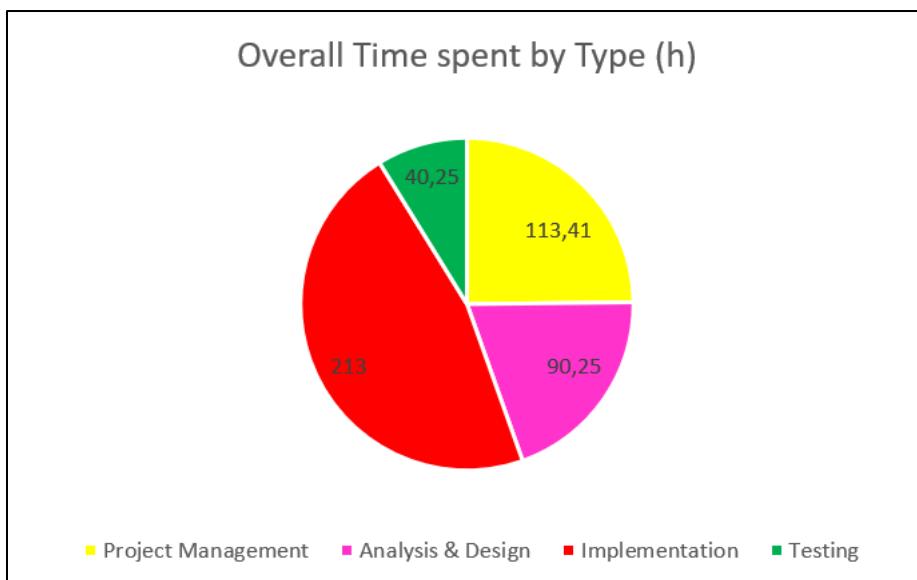


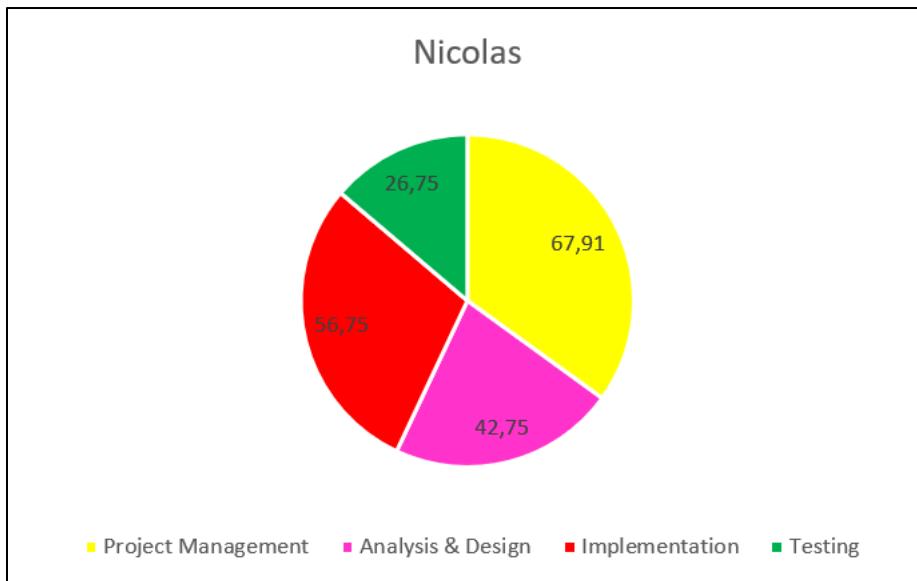
## 23.2 Time Data

As Jira supports time-tracking, every team member tracked on how much work was done at which time.

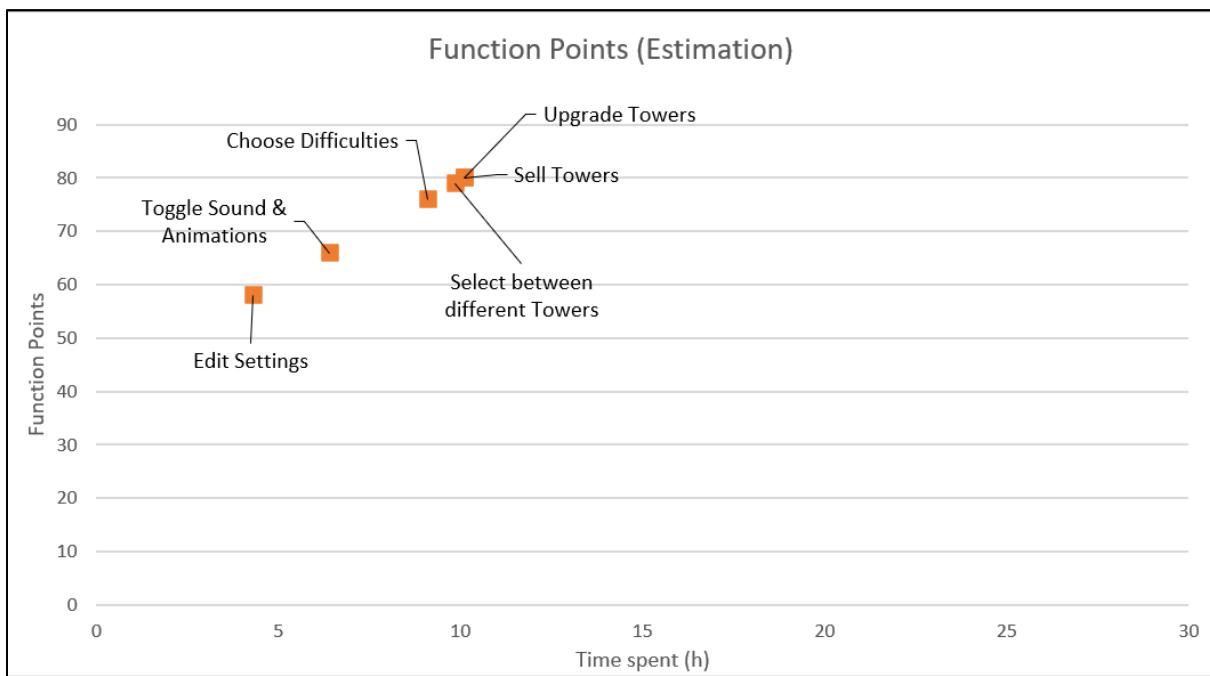
The following charts show the time spent on each work-type in more detail. (click on the right to move on to the next diagram)



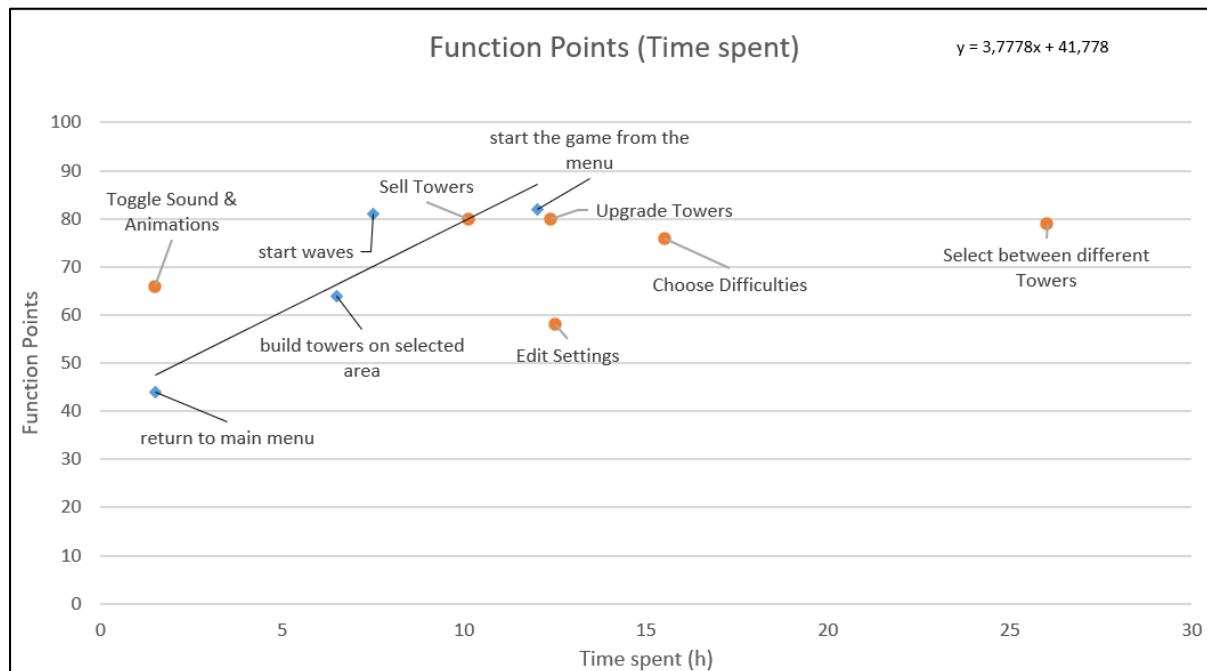




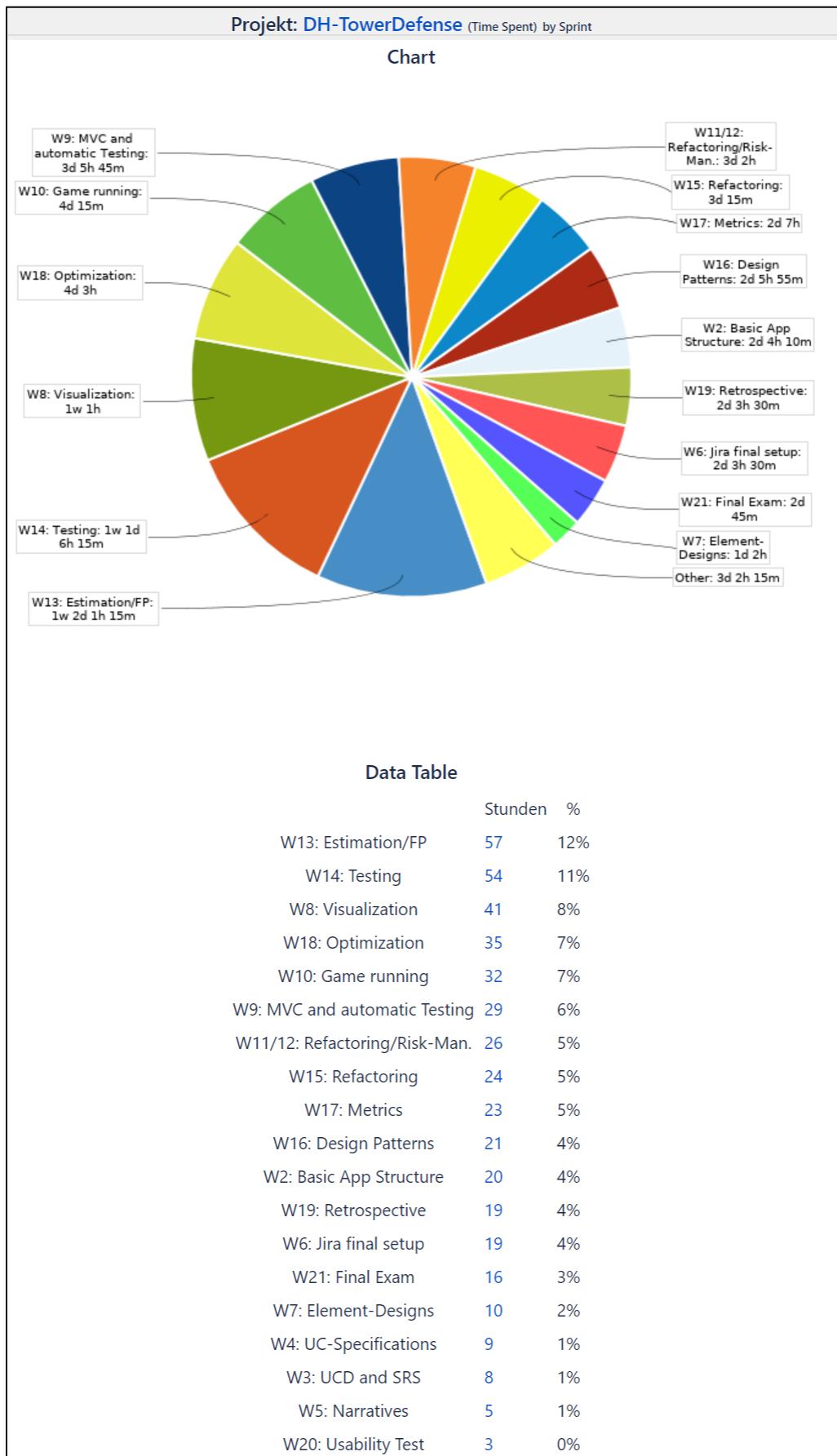
### 23.3 Function Points – Estimation



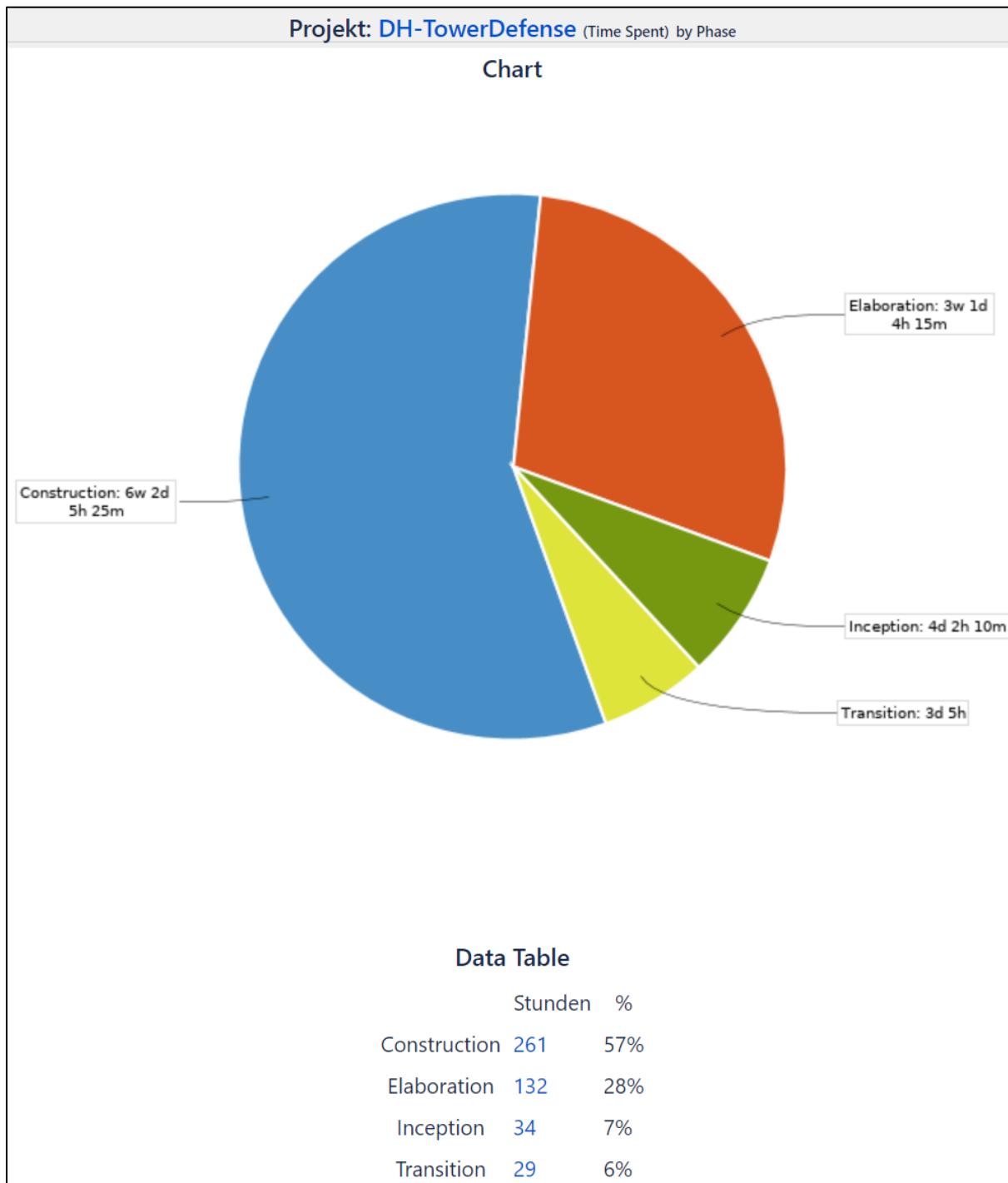
## 23.4 Function Points – Time spent



## 23.5 Time spent by Sprint



## 23.6 Time spent by Phase



## 24 Related Links

- [1] <https://dh-towerdefense.de>
- [2] <https://jira.dh-towerdefense.de>
- [3] <https://github.com/niwa99/Tower-Defense>
- [4] <https://dh-towerdefense.de/week-1-project-vision/>
- [5] <https://dh-towerdefense.de/week-2-team-roles-and-working-with-jira/>
- [6] <https://dh-towerdefense.de/week-3-srs-and-ucd/>
- [7] <https://dh-towerdefense.de/week-4-use-case-specifications/>
- [8] <https://dh-towerdefense.de/week-5-narratives/>
- [9] <https://dh-towerdefense.de/week-6-project-management-tool-and-our-game-engine/>
- [10] <https://dh-towerdefense.de/week-7-retrospective/>
- [11] <https://dh-towerdefense.de/week-8-class-diagram/>
- [12] <https://dh-towerdefense.de/week-9-mvc/>
- [13] <https://dh-towerdefense.de/week-10-midterm/>
- [14] <https://dh-towerdefense.de/week-11-revision-new-use-case-descriptions/>
- [15] <https://dh-towerdefense.de/week-12-risk-management/>
- [16] <https://dh-towerdefense.de/week-13-function-points-estimation/>
- [17] <https://dh-towerdefense.de/week-14-testing/>
- [18] <https://dh-towerdefense.de/week-15-refactoring/>
- [19] <https://dh-towerdefense.de/week-16-design-patterns/>
- [20] <https://dh-towerdefense.de/week-17-metrics/>
- [21] <https://dh-towerdefense.de/week-18-optimization/>
- [22] <https://dh-towerdefense.de/week-19-retrospective/>
- [23] <https://dh-towerdefense.de/week-20-installation/>
- [24] <https://dh-towerdefense.de/week-21-final/>
- [25] <https://travis-ci.org/github/niwa99/Tower-Defense>
- [26] <https://codeclimate.com/github/niwa99/Tower-Defense>
- [27] <https://dh-towerdefense.de/statistics/>
- [28] <https://github.com/niwa99/Tower-Defense/blob/master/midterm-presentation/TimeSheet.xlsx>
- [29] <https://github.com/niwa99/Tower-Defense/blob/master/documents/SRS.md>
- [30] <https://github.com/niwa99/Tower-Defense/blob/master/documents/SAD.md>
- [31] <https://github.com/niwa99/Tower-Defense/blob/master/documents/Testplan.md>
- [32] <https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UseCaseDiagram-2020-06-12.png>
- [33] [https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start\\_The\\_Game\\_From\\_The\\_Menu/UCS-Start\\_The\\_Game\\_From\\_The\\_Menu.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start_The_Game_From_The_Menu/UCS-Start_The_Game_From_The_Menu.md)
- [34] [https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start\\_Waves/UCS-Start\\_Waves.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start_Waves/UCS-Start_Waves.md)
- [35] [https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Build\\_Towers\\_On\\_Selected\\_Area/UCS-Build\\_Towers\\_On\\_Selected\\_Area.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Build_Towers_On_Selected_Area/UCS-Build_Towers_On_Selected_Area.md)
- [36] [https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Return\\_To\\_Main\\_Menu/UCS-Return\\_To\\_Main\\_Menu.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Return_To_Main_Menu/UCS-Return_To_Main_Menu.md)
- [37] [https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Upgrade\\_Towers/UCS-Upgrade\\_Towers.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Upgrade_Towers/UCS-Upgrade_Towers.md)
- [38] [https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Sell\\_Towers/UCS-Sell\\_Towers.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Sell_Towers/UCS-Sell_Towers.md)

- [39][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Select\\_Between\\_Different\\_Towers/UCS-Select\\_Between\\_Different\\_Towers.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Select_Between_Different_Towers/UCS-Select_Between_Different_Towers.md)
- [40][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Edit\\_Settings/UCS-Edit\\_Settings.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Edit_Settings/UCS-Edit_Settings.md)
- [41][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Toggle\\_Sound\\_And\\_Animations/UCS-Toggle\\_Sound\\_And\\_Animations.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Toggle_Sound_And_Animations/UCS-Toggle_Sound_And_Animations.md)
- [42][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Choose\\_Difficulties/UCS-Choose\\_Difficulties.md](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Choose_Difficulties/UCS-Choose_Difficulties.md)
- [43]<https://github.com/niwa99/Tower-Defense/blob/master/app/src/androidTest/assets/features/StartTheGameFromTheMenu.feature>
- [44]<https://github.com/niwa99/Tower-Defense/blob/master/app/src/androidTest/assets/features/StartWaves.feature>
- [45]<https://play.dh-towerdefense.de>
- [46][https://github.com/niwa99/Tower-Defense/blob/master/final-presentation/Presentation\\_Final.pdf](https://github.com/niwa99/Tower-Defense/blob/master/final-presentation/Presentation_Final.pdf)
- [47][https://github.com/niwa99/Tower-Defense/blob/master/final-presentation/Handout\\_Final.pdf](https://github.com/niwa99/Tower-Defense/blob/master/final-presentation/Handout_Final.pdf)
- [48]<https://dh-towerdefense.de/Video.mp4>
- [49]<https://github.com/niwa99/Tower-Defense/commit/29b359e281fa26b59598bcc57e6c6a3b623b2d03>
- [50]<https://github.com/niwa99/Tower-Defense/commit/228fb972aed3ea1106576e995db02eed7545353c>
- [51]<https://github.com/niwa99/Tower-Defense/commit/0c9b3cfb88c59649abd5ac969d97cd118977082f>
- [52]<https://github.com/niwa99/Tower-Defense/commit/8e569827e013d58915513ce342fb081962008d3c>
- [53][https://www.tutorialspoint.com/software\\_testing\\_dictionary/cyclomatic\\_complexity.htm](https://www.tutorialspoint.com/software_testing_dictionary/cyclomatic_complexity.htm)
- [54]<https://plugins.jetbrains.com/plugin/93-metricsreloaded>
- [55]<https://github.com/niwa99/Tower-Defense/blob/master/app/build.gradle>
- [56]<https://github.com/niwa99/Tower-Defense/actions>
- [57]<https://github.com/niwa99/Tower-Defense/tree/master/app/src/test/java/de/dhbw/towerdefense>
- [58][http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP\\_Calc.html?tCountVal=0#FPCalc](http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html?tCountVal=0#FPCalc)
- [59][https://dh-towerdefense.de/wp-content/uploads/2020/04/tower\\_defense\\_title\\_soundtrack.mp3](https://dh-towerdefense.de/wp-content/uploads/2020/04/tower_defense_title_soundtrack.mp3)
- [60]<https://dh-towerdefense.de/wp-content/uploads/2019/11/FeatureFilesRunning.mp4>
- [61][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start\\_The\\_Game\\_From\\_The\\_Menu/Activity\\_Diagram-Start\\_The\\_Game\\_From\\_The\\_Menu.jpg](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start_The_Game_From_The_Menu/Activity_Diagram-Start_The_Game_From_The_Menu.jpg)
- [62][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start\\_Waves/Activity\\_Diagram-Start\\_Waves.jpg](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Start_Waves/Activity_Diagram-Start_Waves.jpg)
- [63][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Build\\_Towers\\_On\\_Selected\\_Area/Activity\\_Diagram-Build\\_Towers\\_On\\_Selected\\_Area.jpg](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Build_Towers_On_Selected_Area/Activity_Diagram-Build_Towers_On_Selected_Area.jpg)
- [64][https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Return\\_To\\_Main\\_Menu/Activity\\_Diagram-Return\\_To\\_Main\\_Menu.jpg](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UCS-Return_To_Main_Menu/Activity_Diagram-Return_To_Main_Menu.jpg)
- [65]<http://jira.dh-towerdefense.de/secure/ConfigureReport!default.jspa?selectedProjectId=10000&projectOrF>

- [\[66\] !\[\]\(1ad53017baba922c95a6c4c0e7fbd21c\_img.jpg\)](https://dh-towerdefense.de/wp-content/uploads/2020/04/2020-04-24-07_50_35-TimeSpentByUseCase.png)
- [\[67\] !\[\]\(1b82a9d16c0a94acede90e029ee28e0f\_img.jpg\)](https://dh-towerdefense.de/wp-content/uploads/2020/04/2020-04-24-07_51_09-TimeSpentByPhase.png)
- [\[68\] !\[\]\(dfb753276996173d13fb95b507cc7171\_img.jpg\)](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UseCaseDiagram-2019-12-07.png)
- [\[69\] !\[\]\(9406d437e2a8e698fd2ed2c7e70b9a08\_img.jpg\)](https://appetize.io/app/cm2h2wjjz37m3ppa18umxhy34hc)
- [\[70\] !\[\]\(b1c53fdea0da4e1370e49fdb7a23145d\_img.jpg\)](https://github.com/niwa99/Tower-Defense/blob/master/midterm-presentation/Handout-Midterm.pdf)
- [\[71\] !\[\]\(caa0645b6b46fce406dabc4fdf10ee75\_img.jpg\)](https://github.com/niwa99/Tower-Defense/blob/master/midterm-presentation/Presentation-Midterm.pdf)
- [\[72\] !\[\]\(d421e4f7870933dd88a7d7f36c1393af\_img.jpg\)](https://camo.githubusercontent.com/6f98bc0763986e3df5ed8f36d094884989d7d62f/68747470733a2f2f63646e2d696d616765732d312e6d656469756d2e636f6d2f6d61782f3830332f312a49395750636e70474e754934436a7878726b50302d672e706e67)
- [\[73\] !\[\]\(787b0041f4fd678acaf44535a037ff82\_img.jpg\)](https://plugins.jetbrains.com/plugin/7212-cucumber-for-java)
- [\[74\] !\[\]\(9f86cd0295c718e5bb9870356a8276fa\_img.jpg\)](https://plugins.jetbrains.com/plugin/9164-gherkin)
- [\[75\] !\[\]\(86a196b7381a5ff2fc48a1ee6e9aaafe\_img.jpg\)](https://developer.android.com/training/testing/espresso)
- [\[76\] !\[\]\(f0248f07d73a6f3fe45c3835214a934e\_img.jpg\)](https://github.com/mauriciotogneri/green-coffee)
- [\[77\] !\[\]\(cea000c74a0bb1670c1214a7c0b08bbc\_img.jpg\)](https://dh-towerdefense.de/wp-content/uploads/2019/11/demoVideo-2019-11-23.mp4)
- [\[78\] !\[\]\(3ab00c18284632fda9e739c95e479fe6\_img.jpg\)](https://dh-towerdefense.de/wp-content/uploads/2019/11/FeatureFilesRunning.mp4)
- [\[79\] !\[\]\(841575275c4110efb7023d91057712a5\_img.jpg\)](https://github.com/niwa99/Tower-Defense/blob/master/documents/UCS/UseCaseDiagram-2019-10-20.png)
- [\[80\] !\[\]\(5a8f085dd40c43bbc4a4de11f156cf82\_img.jpg\)](http://jira.dh-towerdefense.de/projects/DHTD/issues/DHTD-2)
- [\[81\] !\[\]\(e7b0512d7d5f72bbd7cc55a27282dc7e\_img.jpg\)](http://jira.dh-towerdefense.de/projects/DHTD/issues/DHTD-5)
- [\[82\] !\[\]\(8857a35eca2487b875a40a5af3429c6e\_img.jpg\)](https://dh-towerdefense.de/wp-content/uploads/2020/06/tower_defense_soundtrack_1.mp3)
- [\[83\] !\[\]\(aca02cf80bbb8244ea1d9aa87be9f3d0\_img.jpg\)](https://dh-towerdefense.de/wp-content/uploads/2020/06/tower_defense_soundtrack_4.mp3)