LeRuKa

A 2D Jump'n'Run Game

VISION

„Ein Jump'n'Run Spiel für mobile Androidgeräte, in dem sich der Pizzalieferant Leandro durch Karlsruhes Baustellen kämpfen muss, um pünktlich die Pizza liefern zu können.

Leandro kann sich öffentlich mit anderen Lieferanten messen."

www.leruka.wordpress.com

Leif Bernsdorf

Kassandra Frank

Ruth Weber

Contents

1. 2015

1.1 October

About (2015-10-02 16:59)

Leruka as a project by Kassandra, Ruth and Leif, part of our course "Software Engineering" at the DHBW Karlsruhe. It is a jump'n'run game for Android devices. In this blog we will give you constant updates and news.

This is just a short excerpt for the about page.

Contact (2015-10-02 16:59)

Just write us your message and we will reply as soon as possible! If you prefer you can also send us an e-mail at [1]leruka@web.de.

[contact-form][contact-field label="Name" type="name" required="1"/][contact-field label="Email" type="email" required="1"/][contact-field label="Website" type="url"/][contact-field label="Comment" type="textarea" required="1"/][/contact-form]

1. `mailto:leruka@web.de`

This is just a short excerpt for the contact page.

Project Vision (2015-10-02 17:32)

Hi there!

We, Leif, Kassandra and Ruth, are students at the cooperative university of Karlsruhe.

We plan on creating a simple mobile game for Android devices. Of course we know it is not easy to develope an app so we want to keep it simple. So we start with the easy tasks and expand it later.

We want to make an Arcade/ Jump'n'Run game an the game mechanic should be dodging obstacles. There are different levels which you can play and if you are a logged user you could find your name in the highscore with other gamers.

Additional features could be that you can wrote a message to other gamers, rate the different levels or collect points to get special prices.

The main technologies we will use are Java and a SQL-Database.

At the moment leruka is our codename and when we have decided on a name for our app, we will let you know. :-)

Markus (2015-10-05 07:11:46)
Hi, Congratulations! I think you had a great idea building such a game. I expect your game will be single player. So I suggest you to think about an online leader board, where players could upload there score to compete with others. Good luck! Markus from timax

leruka (2015-10-05 09:53:44)
Thank you! We thought about something like the leader board but at the beginning we want to keep it simple. Ruth from leruka

binzmrc (2015-10-06 10:28:35)
Hey there, This project sounds like a lot of fun. Jump'n'Run is one of my favorite types of games, so I'm looking forward to see more about it. I would suggest you to implement something special to make your game unique and exciting. Good luck! Yours, Marc (Chezz - The Rating Game)

leruka (2015-10-06 10:33:28)
Thank you! We will think about your suggestion. leruka

Team Schwartwolfland (2015-10-15 10:20:29)
Hey, that is a very interesting idea and sounds like the game could be lots of fun. But why did you chose specifically a jump'n'run game? Do you have any experiences with it, because it sounds rather difficult. Yours, Team Schwartzwolfland

leruka (2015-10-15 10:33:14)
Hey Schwartzwolfland, we decided on a Jump'n'Run game because we are all interested in these type of games. Leif did have some experience in developing Jump'n'Run games. Leruka

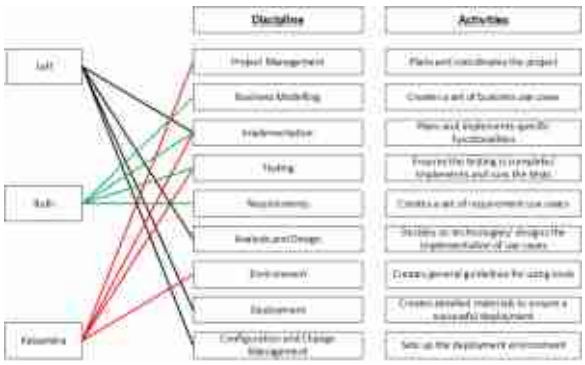
Midterm summary | leruka (2015-12-22 21:42:12)
[…] HW1: Vision […]

Final Presentation – leruka (2016-06-10 20:47:04)
[…] Project Vision […]

W2 - Team roles and technologies (2015-10-09 12:57)

Hi,

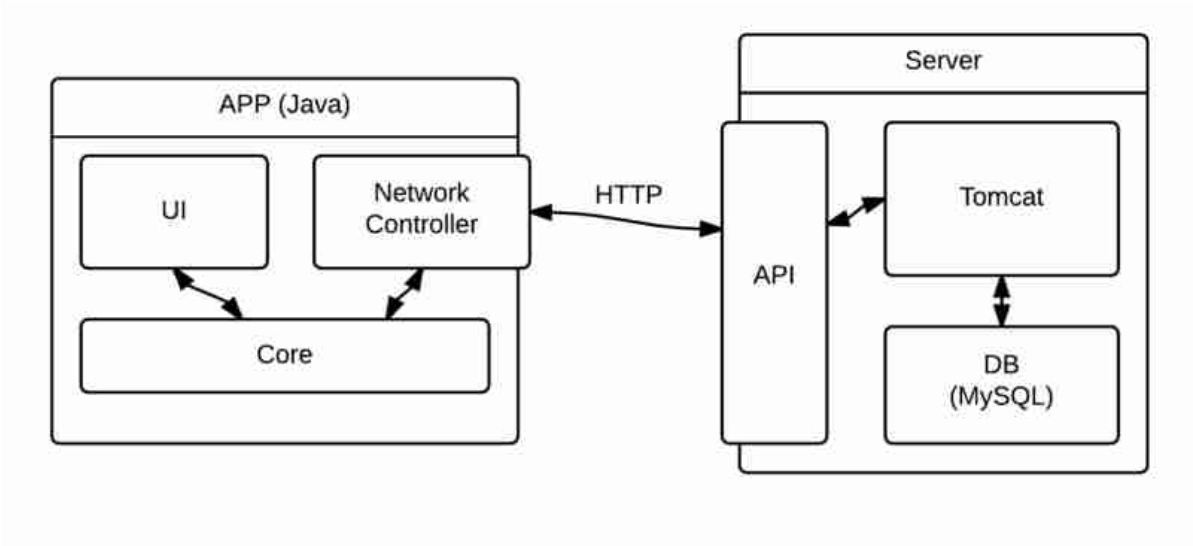
this week we decided our role allocation as the following:



[1]
our role allocation using RUP

In addition to that, we defined the technologies and the architecture of the APP. We will create a Java-Android APP and will use a Tomcat server with a MySQL database for storing user data. The APP will communicate with the Server via a HTTP connection. For each user, the database will store his username, password and score. The following chart illustrates this architecture:

[2]



Greetings, Leif

1. https://leruka.files.wordpress.com/2015/10/rup1.png
2. https://leruka.files.wordpress.com/2015/10/technologien.png

mark_us (2015-10-13 15:28:19)
Hi leruka, you have clearly defined your team roles and show them in your diagram in a good readable way! I appreciate, that you will work on my suggestion and implement an online score. Can you tell me which technology you will use for building the API and displaying the data in web browsers? Best regards Markus (Timax)

leruka (2015-10-14 15:24:01)
Hi Markus, thanks for your kind comment! The scoreboard is a good idea, but we are currently not planning on displaying it in a web browser. The Tomcat will process all incomming HTTP requests. We will develop the exact definition of each possible request in the near future. Greetings, Leif from Leruka

mark_us (2015-10-14 16:51:47)
Hi Leif, so this was an misunderstanding of mine. Thanks for clearifying it. Markus

kanibar (2015-10-15 10:06:01)
Hi leruka, We like your plan of role distribution as everyone has the same amount of work (at least in terms of roles) and it can be seen easily, who has which role. The only thing, we can not see, is why you only store so few data. For evaluation purposes it might be helpful, to have more data, such as logins per day, user activities or dates of high scores. We're looking forward, to see your project evolve. Kaniba

leruka (2015-10-15 10:28:40)
Hey Kaniba, thank you for your comment. We didn't think about it before, but it is a good idea to store more data, like logins per day. So we could have a statistic and can see how often it is used. Leruka

Team Schwartwolfland (2015-10-15 11:40:23)
Hi Leruka, We think your role constellation is clear and easy to read. You used the RUP terminology very well. But we have the concern, that although everyone has the same amount of roles there might be an unbalance in terms of time, that has to be spent. Is there a backup plan, if it turns out, that someone has to do more than the others? Best regards, Team SWL

leruka (2015-10-22 10:56:01)
Hi! We distributed the roles that everybody has a larger role and a smaller role. And if it turns out that someone has to do more we will overthink our role allocation. Greetings Leruka

Midterm summary | leruka (2015-12-22 21:42:14)
[…] HW2: Team Roles and Technologies […]

Final Presentation – leruka (2016-06-10 20:46:52)
[…] to the blogentry […]

We're on GitHub! **(2015-10-16 14:24)**

Hi,

I just wanted to let you know that we are now on GitHub. The repository will contain the Code for our APP. For now, there is now code to show, but soon there will be some! Feel free to look at what we will commit or just create a branch and play around with our code! :)

https://github.com/Leruka/leruka

We will also refer to the repository when we want to show you some code we wrote.

Greetings, Leif

HW3: Software Requirement Specification (2015-10-22 11:29)

Hey there!

This week we started to write our SRS which you can see here: [1]SRS

Additionally you can see all our documents on GitHub: [2]GitHub

If you want to see only the Overall Use Case diagram look here: [3]Use Case

Greetings Kassandra

1. https://raw.githubusercontent.com/Leruka/leruka/master/docs/rup_srs.dot.pdf
2. https://github.com/Leruka/leruka/tree/master/docs
3. https://raw.githubusercontent.com/Leruka/leruka/master/docs/UseCaseSurvey.jpg

---

cookyjasper (2015-10-22 16:51:46)
Hello Team Leruka, I really like your Overall Use Case diagram, it's very simple and easy to understand. But I noticed two points in your diagram. First, the settings use case is missing a verb like "Change settings". And the second point is that dependencies like "you can only login if you're registered" should not be modelled in a use case diagram.Your SRS is well done, it contains every information needed to understand. Best regards, Jasper from team Cooky

Kassandra (2015-10-22 18:22:08)
Hey Jasper, Thank you for your comment. We changed the things you noticed. Best regards Kassandra

damobert the unicorn (2015-10-23 12:49:45)
I like your SRS as well. It gives all the information needed and is easy to understand. But I still got a question about your UCD: There is not connection between a user and the messaging user story. Why? The version control on git is fine as well. Best regards, damobert the unicorn

Kassandra (2015-10-23 13:10:48)
Hello damobert, thank you for your comment. We forgot the connection between user and message and didn't realize it. We will fix it! Thanks a lot. Best regards, Kassandra

Final Presentation – leruka (2016-06-10 20:46:26)
[...] to the blogentry [...]

HW4: Use Cases (2015-10-25 19:19)

Hey there,

today we want to show you our first two Use Cases for

- [1]View public highscore
- [2]Change settings

We also updated our [3]SRS so you can find the links to the Use Cases also in there.

Greetings Ruth

1. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewpublicHighscore.pdf?raw=true
2. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCSettingsChange.pdf?raw=true
3. https://raw.githubusercontent.com/Leruka/leruka/master/docs/rup_srs.dot.pdf

---

Andre (2015-10-27 11:28:06)
Hello team Leruka, your homework seems to meet all the grading requirement and I espacially like your detailed UCD for "change settings". You could maybe add the different areas like "user" or "system" to your UCD "view public highscore". Greetings André

Ruth (2015-10-27 17:49:07)
Hi André, thank you for your comment. We had the areas also in the "view public highscore" Use Case but it disappeared as we imported it in the template, so thanks for the hint. We will change it immediately. Best regards Ruth from Leruka

kaniba (2015-10-28 19:09:21)
Hi team Leruka, like André said before me your homework is pretty good, but we have to littel questions. First is there really no alternativ flow in the highscore part? For exampel if somebody uses the menu button on his smartphone. The second question would be if it would make sense to splitt the UCD of the "Change Settings" up to two diagramms to make it better readable. But really good work. Greetings, Sebastian - Team KaNiBa

Ruth (2015-10-29 09:11:43)
Hey Sebastian! Thanks for your comment. When we upload our next homework we will change the alternative flow. We wanted to write to be determined (tbd) instead of not applicable, so that we can think of it again. That we could split our "Change Settings" in more than one Use Case is a good idea. We will also change it when we will upload the next homework. Best regards, Ruth

cookymario (2015-10-29 00:30:24)
Hey team Leruka, i looked at your use cases and the activity diagrams, which were created for that. I really liked the structured separation of user, app and server and i think that it is clearly defined. Furthermore you have created nice mockups, which show how to use the application in future. Maybe you can describe your use cases in more detail to get some uncertainties out of way. Best regards, Mario from Team Cooky

Ruth (2015-10-29 09:17:51)
Hi Mario, I am happy that you liked our Use Cases and our mockups. The description of the Use Cases will come with the next homework, so I hope you will then look at it. Greetings Ruth

Team Schwartzwolfland (2015-11-05 13:54:50)
Hello Team Leruka, we looked at your use case documents and we think your activity diagrams are well structured. Also your mockups are created really detailed and we like your design. Like Team KaNiBa already mentioned maybe you can add a alternative flow for the highscore. Maybe there is even a more simple navigation possible. Greetings, Team Schwartzwolfland

Ruth (2015-12-08 16:34:02)
Hey Team Schwartzwolfland, it is a little late but still thank you for your comment. We think our alternative flows are enough. Best regards, Ruth from Leruka

Final Presentation – leruka (2016-06-10 20:47:06)
[…] HW4: Use Cases […]

1.2 November

HW5: Feature Files (2015-11-01 23:34)

Hi guys,

in the last few days we were thinking about functional testing. In the future we are going to use Calabash (Gherkin syntax) to test our UI elements and menus. For now we have created a few tests (.feature files) and covered two features with them: Viewing the public highscore and changing the user preferences. The first feature has only one scenario, but the second one has four of them.

The narratives are on the one side uploaded as [1]feature files, but are also directly added to our [2]use cases, so that you can compare them to the activity diagrams and check if the tests fit to our plans :)
Furthermore we have uploaded more use cases to our git!

- [3]View personal highscore
- [4]Rate a level
- [5]View the instructions

Greetings, Leif

1. https://github.com/Leruka/leruka/tree/master/features
2. https://github.com/Leruka/leruka/tree/master/docs
3. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewpersonalhighscore.pdf?raw=true
4. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCRatelevels.pdf?raw=true
5. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewInstruction.pdf?raw=true

---

DaBrauun (2015-11-02 12:08:09)
Hi leruka, nice work with the feature files :) theyre well written and comprehensive (the preferences one). I like that you used the "But" statement. Also you match the Grading Criteria. Did you manage to get them running though? - Just asking, because many are struggling with that. Kind regards, Bookster Daniel

Leif (2015-11-04 16:23:12)
Hi Daniel, thank you for your kind reply! Unfortunately we did not jet manage to get the tests running. Do you know someone who got it for Android APPs? Maybe they can help us. Greetings, Leif

effortManagement (2015-11-02 19:02:59)
Hi Leif, your Narrative looks good but there are syntax errors. "BUT" is not a Keyword. In this case you have to implement a complete new scenario. Kind regards, effortManagement

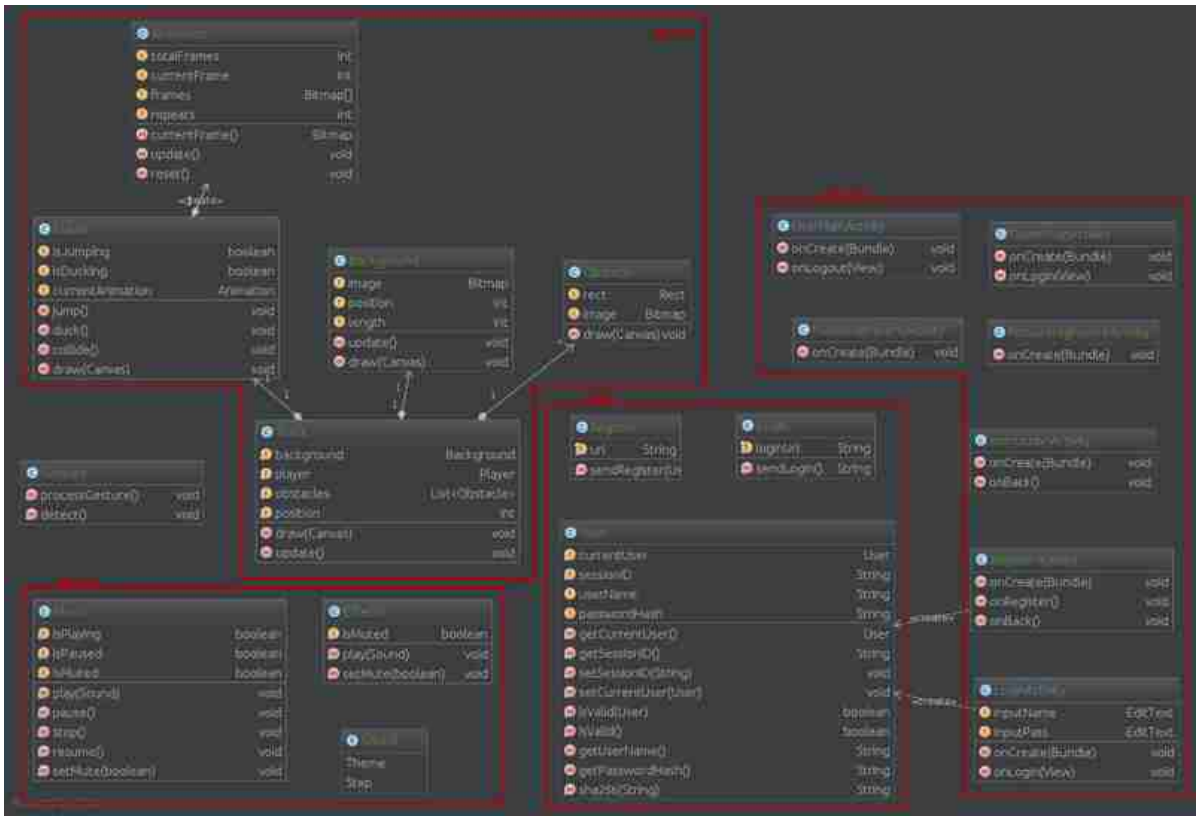
Leif (2015-11-04 16:31:39)
Hi! We have found "BUT" as a keyword on the official cucumber website. But we will research on how to implement it correctly. Thanks for your review! Greetings, Leif

Final Presentation – leruka (2016-06-10 20:47:08)
[...] HW5: Feature Files [...]

HW5: UML Class diagram (2015-11-10 23:27)

Hey guys!

We have thought about the classes we plan to implement and therefore created a UML class diagram. It holds the classes with their fields, methods and the packages they are in.[1]



We are happy about any feedback you give us :)

Greetings, Leif

1. https://leruka.files.wordpress.com/2015/11/leruka_class_diagram_packages.png

---

cookymario (2015-11-11 18:27:06)
Hi team Leruka, we think your class diagram shows nearly all relevant classes to implement a Jump 'n' Run game. Furthermore we were pleasantly surprised, that you separated your classes into relational areas. It facilitates the understanding of the relations in your application. I am very interested to see the outcome of this project! Greetings, Mario from team Cooky

Leif (2015-11-12 10:59:01)
Hi Mario, Thanks for your kind review! I am happy that you like our class diagram. Leif

kaniba (2015-11-12 10:08:39)
Hello guys, first of all your diagramm is very good and we liked the way you structured your diagramm in different sections. We only would recommend you to use a white background the next time because it is really hard to read on the dark one.

Leif (2015-11-12 11:01:34)
Hi Sebastian, you are right, it can be hard to read. I will surely change my theme before exporting the diagram next time. Thanks for your comment :) Leif

Andre (2015-11-12 10:12:26)
Hey team Leruka, I agree with the previous comment that you have included almost all the relevant classes. I would be interested in which tool you used to create your diagramm? Greetings, Andre

Leif (2015-11-12 11:05:42)
Hi André, we used IntelliJ to generate the diagram. Unfortunately I could not get Android Studio to do that, so we had to use the IntelliJ ultimate edition. We even tried serval plugins like PlantUML or SimpleUMLCE for Android Studio, but they did not work :/ The packages are drawn with gimp, just for a better overview ;) Greetings, Leif

Final Presentation – leruka (2016-06-10 20:47:11)
[...] HW5: UML Class diagram [...]

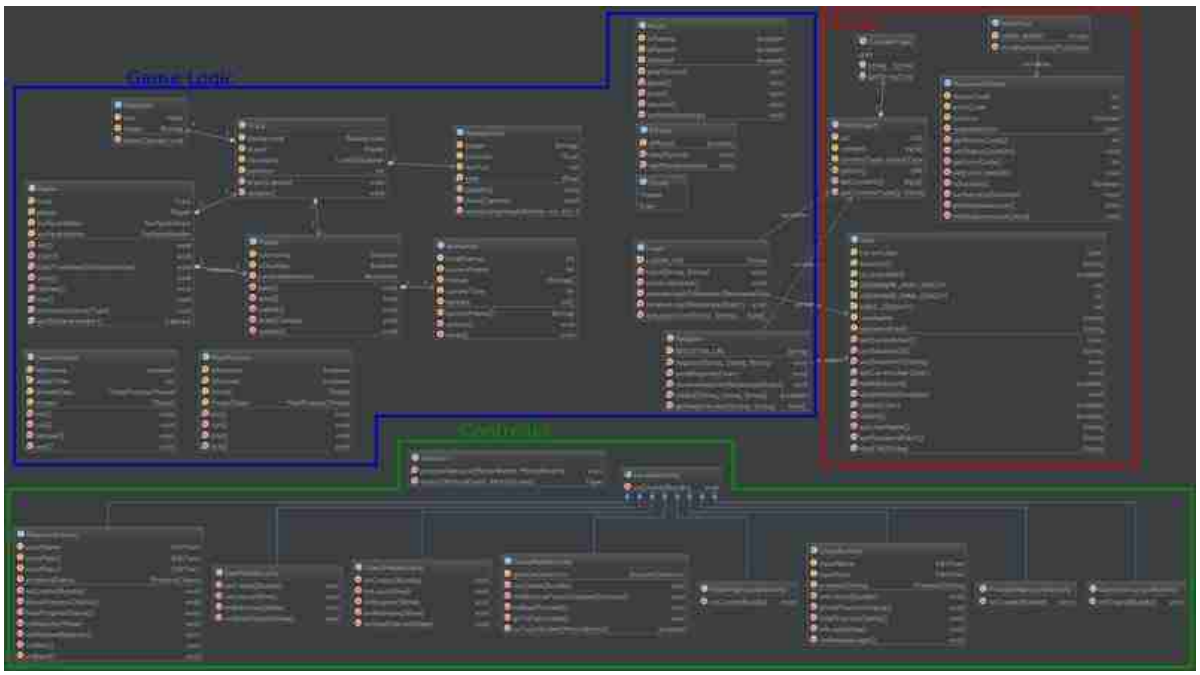
HW7: Software Architecture Document (2015-11-18 18:11)

Hey!

This week we uploaded our SAD. Here you can find it: [1]SAD.

Also here you can see the class diagram seperated in Model-View-Controller:

[2]



Greetings

Leruka

1. https://github.com/Leruka/leruka/blob/master/docs/SoftwareArchitectureDocument.pdf
2. https://raw.githubusercontent.com/Leruka/leruka/master/docs/classDiagram_logicalView.png

---

christopher the unicorn (2015-11-19 09:48:55)
Hi Team Leruka, I really like that you have a lot of classes but I think you put a lot of your classes in the wrong category. Most of the classes in your "controller" are actual models. Controllers are responsible to control the view with information drawn from the models. The only actual class doing that is probably "Gesture" as I would assume that it changes the view based on a gesture you make. All your other classes like User, Effects, Music, etc. actual model abstract classes and don't control anything (they don't change anything in the view). So maybe give it another look :) Greetings, Christopher from Chezz

> leruka (2015-11-19 12:17:03)
> Hi Christopher! Thanks for you comment! We decided to make Effects etc. as Controllers because it includes logic and in the definition was the point that logic is in controllers. But we will think about it again. Best regards Leruka

> Leif (2015-12-08 17:11:16)
> Hi Christopher! We have now updated our class diagram. You were right about all our controllers, which were none. We now use the new category "Game Logic" for the classes, which are neither controllers nor models. You can find the new version here: https://github.com/Leruka/leruka/blob/3b-1fa5aa91f5c86cfb9e381cb6524d0e5bc776b 8/docs/classDiagram _logicalView.png Greetings, Leif

cookymario (2015-11-19 12:00:15)
Hi Team Leruka, i read your Software Architecture Document and i like that you have described your application detailed with classes you want to use. But as Christopher already mentioned it before, i think you confused some controller classes with your models. The classes, which were defined already are obviously model classes. However the view classes seemed to be correctly. Kind regards, Mario from Team Cooky

> leruka (2015-11-19 12:25:41)
> Hey Mario! Thanks for your comment. As we answered to Christopher we decided to make all these classes Controllers because it includes logic. But as we said we will think about it again. Greetings Leruka

Daniel (2015-11-19 12:02:15)
Hello LeRuKa, We think that you got an easy to read SAD. It's compact and short. Your Class-Diagram is comprehendable easily. However in our opinion you should make up ur mind to the required wordload on your system. Greetings, Team TaskForge

> leruka (2015-12-08 16:43:45)
> Hey Daniel from Team Taskforge, we forgot to response to your comment, because we discused it with you in personal. So now, a little late, thanks for your comment. Greetings, Leruka

Final Presentation – leruka (2016-06-10 20:47:14)
[...] HW7: Software Architecture Document [...]

HW8: Jira (2015-11-24 18:42)

Hey guys!

We are now also on [1]JIRA.

On Friday we started our first sprint, which lasts two weeks.

So hopefully you will stay tuned and watch how our project progress.

Greetings

Leruka

1. http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=15

Felix (2015-11-25 11:04:48)
Hey Team Leruka, I really do like ur structure in your Sprints, you taged the RUP deciplines and assigned issues to certain persons. I cannot really see anything to improve on... sorry ;) Greets Felix

leruka (2015-11-25 16:29:49)
Hey Felix! Thanks for your comment! It is good that you do not see anything to improve ;-) Best regards Leruka

cookymario (2015-11-26 10:59:39)
Hey team Leruka, you clearly defined the tasks you want to do in the sprint and you assigned the tasks to your team members. But you should carefully take into consideration to estimate the work loads on all tasks in advance to get a nice burndown chart. Kind regards, Mario from Team Cooky

leruka (2015-12-08 14:14:29)
Hey Mario, thanks for your comment. We will try to follow your advice. Best regards, Leruka

TSWL (2015-12-04 22:02:02)
Hey Leruka , We read your jira and it Looks perfectly nice :) You fit all the grading criteria. I ... really dont see what i should criticise sooo just Keep up with the great work :D Best regards TSWL

Final Presentation – leruka (2016-06-10 20:47:16)
[...] HW8: Jira [...]

1.3 December

HW 9: "Gantt" (2015-12-02 20:13)

Hello,

this week we tried to export our tasks from Jira into MS Project to create a gantt diragram. It was a little bit difficult and we had to copy the tasks ourselves because MsProject didn't import our tasks.

[1]Here you can see our Gantt Diagram.

Greetings,

Kassandra

1. https://github.com/Leruka/leruka/blob/master/docs/Projektplan_Leruka.pdf?raw=true

---

mark_us the unicorn (2015-12-03 11:26:22)
Hi Team leruka, we the awesome unicorns have analyzed your Gantt Chart. It is pretty awesome. You fullfilled all the critera: there are Inception and Elaboration clearly seperated, all your tasks are associated to RUP workflows and you have Milestones set up. We recognized a small misstyping (I think it is) - your last "task" End of Elaboration lasts 16 hours. I think you wanted to type "0" to make it a milestone. You could quickly correct this. Best regards Mark _us the unicorn

Kassandra (2015-12-03 11:30:58)
Hi Mark _us, thank you for your comment. You are right! We created the milestone and afterwards we didn't recognize that it changed to 16 hours. We will correct it and reload the gantt chart. Greetings Kassandra

Andre (2015-12-03 11:30:06)
Hey team leruka, your Gantt diagramm seems pretty detailed, however it might be better to write the actual name of the homework instead of "Hausaufgabe HW1" for a better understanding of what you did. Greetings Team Gaming-Bets

Kassandra (2015-12-03 11:33:18)
Hey Andre, yes you are right that it would be better to write the actual name. But as we created some tasks we didn't know what the homework would be. We try do change the title of homework to the actual name and will update our gantt chart. Greetings Kassandra

cookydo (2015-12-03 11:32:05)
Hey Leruka guy and girls, You're GANTT chart looks very nice and I could find any task you've done. You assigned dates and resources to all of them, as expected. The only thing missing is the Inception milestone for the midterm presentations (or at least I could not find it). If you add this one, your chart satisfies all the grading criteria. Best regards, Dominik from Team Cooky

Kassandra (2015-12-03 11:38:51)
Hey Dominik, we have a Inception Milestone it is called "End of Inception". If you meant the Elaboration Milestone ("End of Elaboration") it is too long for a milestone as Markus informed us that it is misstakenly set up for 16 hours. Greetings Kassandra

Final Presentation – leruka (2016-06-10 20:47:19)
[…] HW 9: “Gantt” […]

Midterm summary (2015-12-22 21:42)

All Documents and Code are available on [1]GitHub

Our [2]Midterm Presentation

Here you can see our [3]handout

HW1: [4]Vision

HW2: [5]Team Roles and Technologies

HW3:

- [6]SRS
- [7]Overall Use Case Diagram

HW4:

- [8]Use Case Duck
- [9]Use Case Jump
- [10]Use Case Start Game
- [11]Use Case View Instruction
- [12]Use Case Register
- [13]Use Case Login
- [14]Use Case View Public Highscore
- [15]Use Case View Personal Highscore
- [16]Use Case Change Settings
- [17]Use Case Rate Levels

HW5: See feature files in the use case documents.

HW6: [18]Class Diagram

HW7: [19]SAD

HW8: [20]JIRA

HW9: [21] Gantt

1. https://github.com/Leruka/leruka
2. https://drive.google.com/file/d/0B9VagljXVQ-OYWhiUUZTWHpjNjQ/view?usp=sharing
3. https://drive.google.com/file/d/0B9VagljXVQ-Od19hNTgteFNyLXM/view?usp=sharing
4. https://leruka.wordpress.com/2015/10/02/project-vision/
5. https://leruka.wordpress.com/2015/10/09/w2-team-roles-and-technologies/
6. https://raw.githubusercontent.com/Leruka/leruka/master/docs/rup_srs.dot.pdf
7. https://raw.githubusercontent.com/Leruka/leruka/master/docs/UseCaseSurvey.jpg
8. https://raw.githubusercontent.com/Leruka/leruka/master/docs/rup_srs_UCDuck.pdf
9. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCJump.pdf
10. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCStartGame.pdf
11. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCViewInstruction.pdf
12. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCRegister.pdf
13. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCLogin.pdf
14. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCViewpublicHighscore.pdf
15. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCViewpersonalhighscore.pdf
16. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCSettingsChange.pdf
17. https://raw.github.com/Leruka/leruka/master/docs/rup_srs_UCRatelevels.pdf
18. https://raw.github.com/Leruka/leruka/master/docs/classDiagram_logicalView.png
19. https://raw.github.com/Leruka/leruka/master/docs/SoftwareArchitectureDocument.pdf
20. http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=15
21. https://github.com/Leruka/leruka/blob/master/docs/Projektplan_Leruka.pdf?raw=true

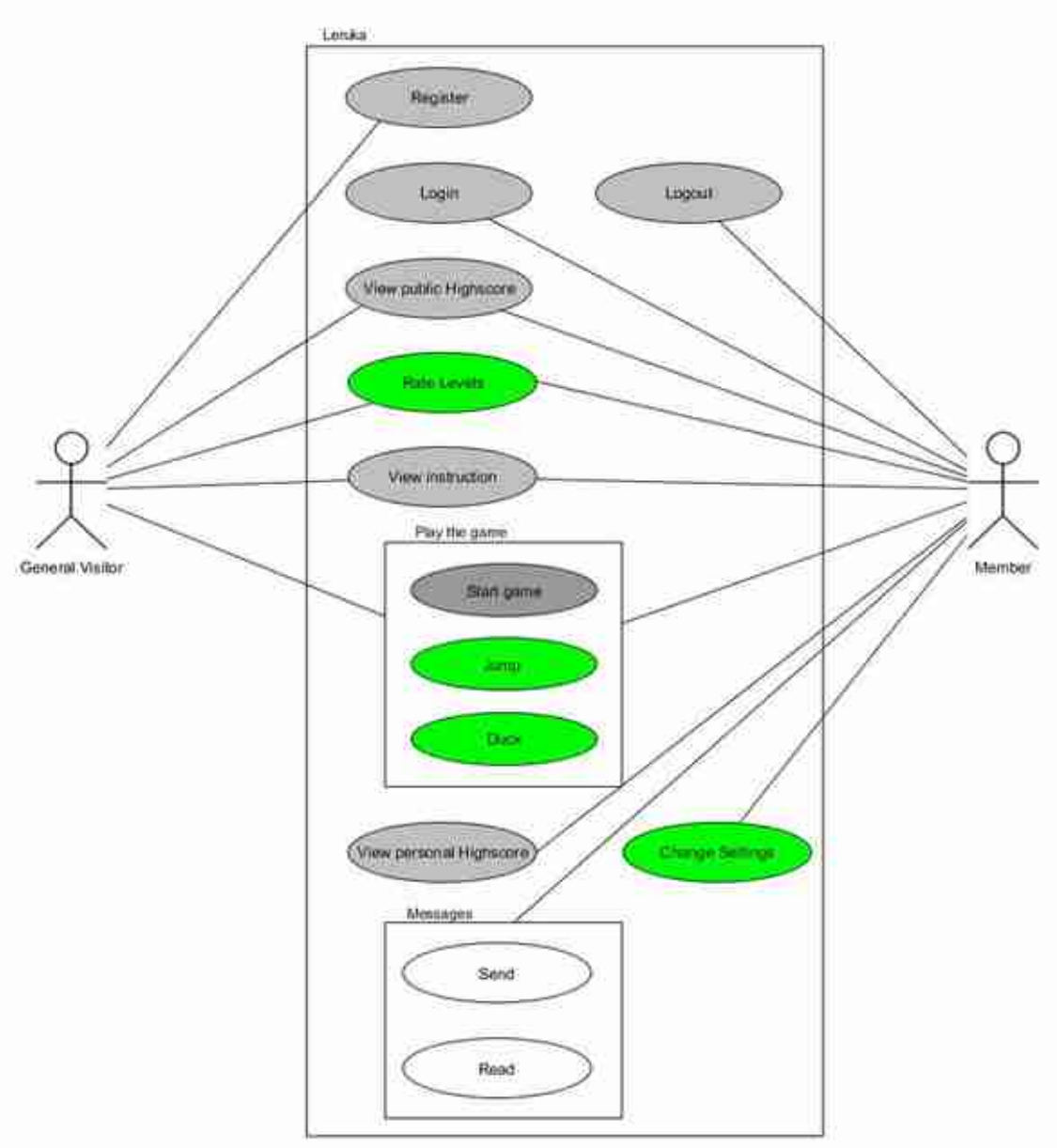
2. 2016

2.1 April

Updated overall use case diagram (2016-04-04 09:00)

Hey guys,

today I will show you our updated overall use case diagram.



The gray bubbles are the finished use cases. The green bubbles are the use cases we want to finish this semester. The white ones are optional and we will implement when we finished the green bubbles and

have enough time for it.

Greetings,

Ruth

SE_2_HW2: Risk Management, Time Estimation and new Use Case Documents – leruka (2016-04-08 20:04:23)
[…] Updated overall use case diagram […]

Final Presentation – leruka (2016-06-10 20:47:21)
[…] […]

SE_2_HW2: Risk Management, Time Estimation and new Use Case Documents (2016-04-08 20:04)

Hey there,

this week we will show you our Risk Management, which you can find [1]here. Also we made a time estimation for the old Use Cases, you can look at it [2]here.

Furthermore we filled out the Use Case document for the new Use Cases, we showed you in the [3]last blog entry:

- [4]Change settings
- [5]Duck
- [6]Jump

Take a look at it and if you want give us some feedback.

Greetings

Ruth

1. https://github.com/Leruka/leruka/blob/master/docs/Risk%20Management%20-%20risk.pdf?raw=true
2. https://github.com/Leruka/leruka/blob/master/docs/Use%20Case%20Estimation%20-%20Estimation.pdf?raw=true
3. https://leruka.wordpress.com/2016/04/04/updated-overall-use-case-diagram/
4. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCSettingsChange.dot.pdf?raw=true
5. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCDuck.pdf?raw=true
6. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCJump.pdf?raw=true

---

Nico (2016-04-11 11:08:40)
Hello leruka team, you have done good work, but there is one thing you could improve for future blogs. It would be great if you add a preview of the documents in your blog. Keep up with your good work. Greetings, bookster

Ruth (2016-04-11 11:22:16)
Hi team bookster, thanks for your kind comment. In future blogs we will try to add previews to our documents. Best Regards, LeRuKa

TaskForge Henning (2016-04-12 08:25:35)
Hi there, you're document for the risk management is pretty nice. You covered a lot of different cases and your documents are consistent. I also enjoy the mockups in the "duck" use case but I was wondering about the numbers in the upper right corner. What's that? Greetings, Henning

Ruth (2016-04-12 10:08:48)
Hey Henning, thanks for your comment. Yes these numbers are points. These points increases, when the play figure runs. You can see it in the Use Case Document Start Game (https://github.com/Leruka/leruka/blob/master/docs/rup _srs _UCStartGame.pdf?raw=true). Best Regards, Leruka

Final Presentation – leruka (2016-06-10 20:47:24)
[...] SE _2 _HW2: Risk Management, Time Estimation and new Use Case Documents [...]

SE_02_HW3: Function point estimation (2016-04-15 21:53)

Hello!

Today I want to show you our function point estimation.

First of all I will give you a little explanation about function points.

Function points are used for estimating the time that will be spent on a certain Use Case. They are calculated in reference on External Inputs, Outputs, Inquiries as well as Internal and External Logical Files from the User's view, thereby they do not depend on the used technology.
To estimate how long it will take to implement a Use Case, function points of UC that have been already implemented are put in relation to the time we spent on it.
We used the calculation for our function points from "[1]TINY TOOL".

Now to the actual work we done.

The following table shows our completed Use Cases with the time we needed to implement. You can also see the at the beginning explained references we calculated the function points on.

Use Case	Estimation (h)	Logged (h)	Transaction						Function Points
				DET	RET	FTR	Complexity	Number of	
Register	13		External Inputs	4	x	1	low	1	
			External Outputs	0	x	1	low	0	
			External Inquiries	8	x	1	low	2	
			Internal Logical Files	2	1	x	low	5	
			External Interface Fil	0	0	x	low	0	38,72
Login/Logout	11,5			DET	RET	FTR	Complexity	Number of	
			External Inputs	5	x	1	low	0	
			External Outputs	0	x	1	low	0	
			External Inquiries	7	x	1	low	2	
			Internal Logical Files	2	1	x	low	4	
			External Interface Fil	0	0	x	low	0	29,92
View public highscore	6			DET	RET	FTR	Complexity	Number of	
			External Inputs	1	x	1	low	0	
			External Outputs	0	x	1	low	1	
			External Inquiries	0	x	2	low	2	
			Internal Logical Files	1	2	x	low	5	
			External Interface Fil	0	0	x	low	0	39,6
View personal highscor	6,5			DET	RET	FTR	Complexity	Number of	
			External Inputs	1	x	2	low	0	
			External Outputs	0	x	2	low	1	
			External Inquiries	1	x	2	low	2	
			Internal Logical Files	2	2	x	low	6	
			External Interface Fil	0	0	x	low	0	45,76
View instruction	3			DET	RET	FTR	Complexity	Number of	
			External Inputs	1	x	1	low	0	
			External Outputs	0	x	1	low	0	
			External Inquiries	1	x	1	low	1	
			Internal Logical Files	1	1	x	low	2	
			External Interface Fil	0	0	x	low	0	14,96
Start game	21			DET	RET	FTR	Complexity	Number of	
			External Inputs	1	x	1	low	0	
			External Outputs	4	x	1	low	2	
			External Inquiries	12	x	3	high	3	
			Internal Logical Files	11	6	x	avg	14	
			External Interface Fil	0	0	x	low	0	146,88

The acronyms DET, RET and FTR mean Data Element Type, Record Element Type and File Type Reference. Through these we identified the complexity which can be low, average or high. The part "numbers of" are the numbers for the measurement parameter of the TINY TOOL.
[2]Here you can find the whole document which also includes the data for the new Use Cases. In this document you also see the estimation for the new Use Cases these get explained later in this blog entry.

For the calculation of function points with TINY TOOL you need some extra information about the whole application.

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE (No Influence 0 1 2 3 4 5 Essential)
1	Does the system require reliable backup and recovery?	1
2	Are data communications required?	5
3	Are there distributed processing functions?	0
4	Is performance critical?	3
5	Will the system run in an existing, heavily utilized operational environment?	0
6	Does the system require on-line data entry?	2
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	1
8	Are the master files updated on-line?	0
9	Are the inputs, outputs, files or inquiries complex?	3
10	Is the internal processing complex?	2
11	Is the code to be designed reusable?	1
12	Are conversion and installation included in the design?	0
13	Is the system designed for multiple installations in different organizations?	0
14	Is the application designed to facilitate change and ease of use by the user?	5

In this screenshot you can see what we choose for our project.

We calculated the function points for all our Use Cases. For example in the next picture you can see the table for the Use Case "Start game".

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	0	●	○	○
Number of User Outputs	2	●	○	○
Number of User Inquiries	3	○	○	●
Number of Files	14	○	●	○
Number of External Interfaces	0	●	○	○

This Use Case has 146,08 function points. [3]Here you can see the whole document.

Below you can see the other completed Use Cases...

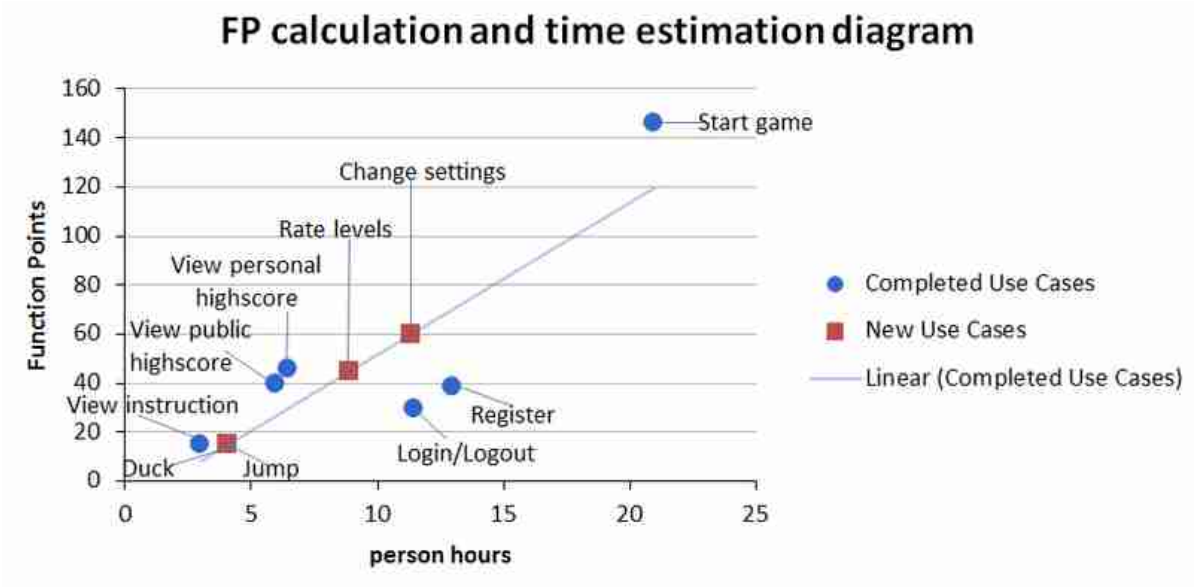
- [4]Register
- [5]Login/Logout
- [6]View public highscore
- [7]View personal highscore
- [8]View instruction

...and the new Use Cases.

- [9]Change settings
- [10]Duck
- [11]Jump
- [12]Rate levels

With the calculated function points we generated a diagram which shows the interact of the function points and the person hours.

We do not have any extreme outliner in our graph, the only on which is a bit different is the Use Case "Start game". This Use Case sounds not like there is much of work behind it, but it contains the whole game logic.



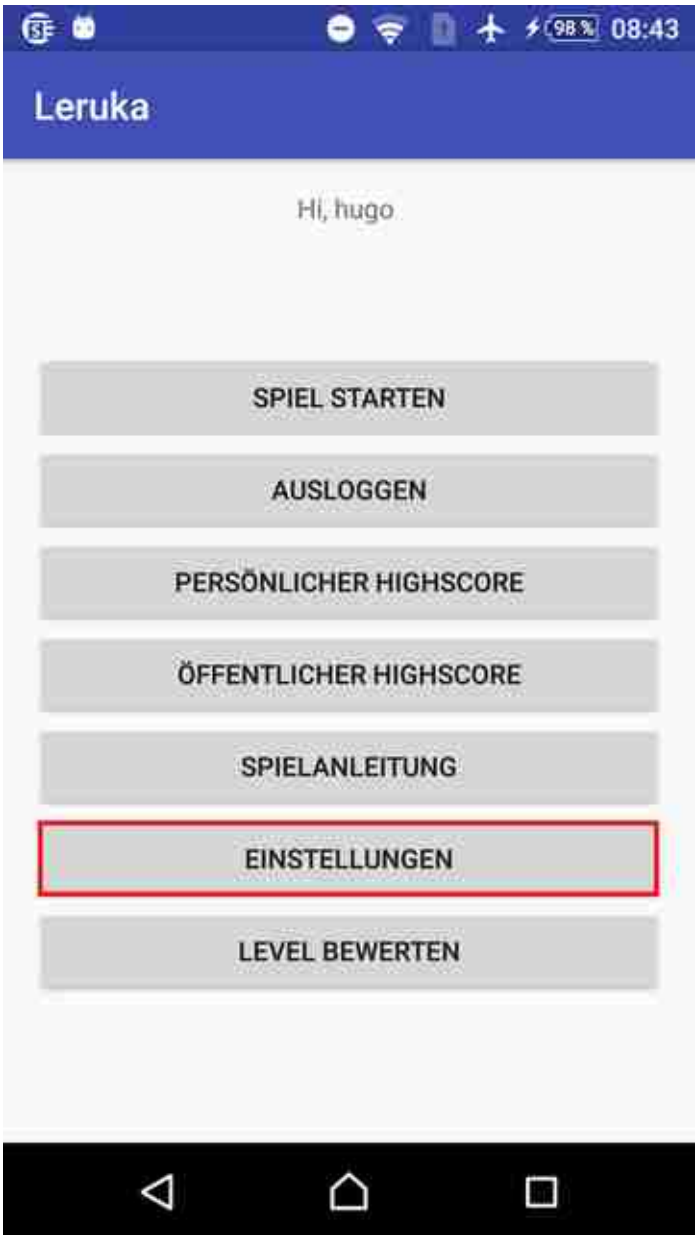
We used this graph to estimate the time for our remaining Use Cases. In the graph you can see these as

quadrats. You can only see three quadrats, because two Use Cases ("Duck" and "Jump") have the same amount of function points.

I hope you liked it and if you want give us some feedback.

UPDATE

We know updated [13]the document because the criteria changed. So now you can see what our DET, RET and FTR are.



Hauptmenü

Einstellungen

The red rectangles show that we have in change settings 7 Data Element Types.

Greetings,

Ruth

1. http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/harvey/FP_Calc.html
2. https://github.com/Leruka/leruka/blob/master/docs/Function%20Point%20Calculation%20-%20FP.pd

f?raw=true
3. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCStartGame.pdf?raw=true
4. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCRegister.pdf?raw=true
5. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCLogin.pdf?raw=true
6. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewpublicHighscore.pdf?raw=true
7. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewpersonalhighscore.pdf?raw=true
8. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewInstruction.pdf?raw=true
9. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCSettingsChange.pdf?raw=true
10. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCDuck.pdf?raw=true
11. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCJump.pdf?raw=true
12. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCRatelevels.pdf?raw=true
13. https://github.com/Leruka/leruka/blob/master/docs/Function%20Point%20Calculation%20-%20FP.pdf?raw=true

---

mark_us the unicorn (2016-04-19 12:15:11)
Hi LeRuKa, I had a look on your documents and what you did looks legit. You analysed your old use cases, got function points for them and used those FPs to calculate the time of your new cases. I can see this quite clear in your velocity diagram. For making it more easy to read you could have put labels to the points in the diagram with the name of the corresponding use cases. (Just as an idea) Keep it up! Yours Markus

Ruth (2016-04-20 11:42:40)
Hey Markus, thanks for your kind comment. In the first diagram I made I added the labels, but we have like 3 Use Cases with the same points and some others near by so it was a bit unclear which label is for which Use Case so I deleted them. But we will keep your advice in mind for the next diagrams. Best regards, Ruth from LeRuKa

mark_us the unicorn (2016-04-20 22:34:07)
Hey Ruth, I totally understand what you mean. Well, an interested person anyway needs to read all the documentation and then will understand which use case belongs to which point in the diagram. ;) Best regards Markus

marc the unicorn (2016-04-19 23:49:48)
Hey there LeRuKa, Your function point estimation looks very good. The post is very detailed which makes it easy to understand what you did and why. Your graph is quiet balanced so you got some good time and work management. The only thing I got to mention is the possibility of adding the formulas for the calculation of your function points. This would show the readers how you actually defined the different FP types (especially if the formulas are not the default ones). Best regards, Marc the unicorn

marc the unicorn (2016-04-19 23:51:57)
Whoops, typo: quiet –> quite ;)

Ruth (2016-04-20 11:46:12)
Hi Marc, thanks for your comment. we didn't explain it further because we kept to the calculation exactly like the slides from lecture. Greetings, Ruth

Time estimation – leruka (2016-05-04 10:29:17)
[...] SE _02 _HW3: Function point estimation [...]

Final Presentation – leruka (2016-06-10 20:46:43)
[...] Function Points [...]

SE_02_HW3 Test Driven Development (2016-04-25 21:18)

This week we started to use test driven development (TDD) for testing our app.

But first we tried it with a demo class named calculator. The main idea of TDD is that you first think about which things your methods should do and you write the tests for this. At this moment you only got the test cases and do not have any class and methods coded. Then you will write your code and let the tests run. Most of the tests will fail but you are writing more and more code until your method behaves like you expected while writing the tests.

So to see how we did TDD with a simple class named calculator watch this [1]file.

After we learned how TDD will work we tried it with the classes User, Registration and Login in our project. [2]Here you will find the implementet test cases.

When we run the tests not all passed but most of them.

Additionally we integrated it with our [3]build.gradle file so theres a task "test" to execute the JU-nit tests during the build of our app.

Comments and ideas to our TDD are welcome.

Greetings Kassandra from Leruka

1. https://github.com/Leruka/leruka/blob/master/docs/DemoTDD.pdf?raw=true
2. https://github.com/Leruka/leruka/tree/master/app/src/test/java/com/leruka/leruka/user
3. https://github.com/Leruka/leruka/blob/master/app/build.gradle

christopher the unicorn (2016-04-27 09:00:30)
Hi Kassandra, Looks like you figured TDD out pretty well. I really like the pdf you did for the example calculator and the fact that one of your tests failed means that your really thought about what your class should do and not just tested something you knew it already does. So I have absolutely no complains about testing. The only thing I'm missing is the belated homework from last week. How did you figure out what you can do this semester? Kind regards, Christopher from Team Chezz

Kassandra (2016-04-27 09:44:16)
Hi Christopher, thanks for your comment. About the belated homework we wasn't sure how detailed it has to be and decided to ask our prof the next lesson. We will do the belated homework as soon as possible when it is clear how detailed it has to be. Greetings, Kassandra

Andre (2016-04-27 11:37:50)
Hello Team Leruka, you demonstrated the use of TDD in android studio quite well and your unit tests seem to cover some important points. We also ran into problems with the build.gradle file so don't feel bad about it :P Greetings Team Gamingbet

Kassandra (2016-04-27 11:42:33)
Hi André, thanks for your comment. Did you get it now with the build.gradle file? If so can you give us help with it? Greetings, Kassandra

kaniba (2016-04-27 11:40:59)
Hi there, the TDD document looks very good and understandable. Also you have links to test files, wich are looking good. Philipp - Team KaNiBa

Kassandra (2016-04-27 11:45:08)
Hey Philipp, thanks for your comment. So you think that all is fine or is there anything to improve? Greetings, Kassandra

cookymario (2016-04-27 11:43:14)
Hi Team Leruka, I really like how you explained the way of using TDD for your example class "calculator". Futhermore I looked at your test classes. Did you wrote tests for all your user actions or only for some of them? Regarding to your problem with Gradle, you can try to add the following LOC to your build.gradle: //JUnit testCompile 'junit:junit:4.12' Best regards Mario Team Cooky

Kassandra (2016-04-27 11:50:23)
Hi Mario, thanks for your comment. At the moment we've got tests for all actions in the class "User". Additionally we wrote tests for the classes "Register" and "Login" but they don't cover all actions. Thanks for your help about Gradle but we've got the LOC you mentioned in our build.gradle but it doesn't work as expected. Greetings, Kassandra

Final Presentation – leruka (2016-06-10 20:47:26)
[…] SE _02 _HW3 Test Driven Development […]

SE_02_HW4: Fowler Refactor (2016-04-30 00:29)

Hey there!

This week we have applied to example project the refactor steps by Fowler.

Every member of our team has generated a repository on GitHub with the example code and commited every step, with that you can see step by step how refactoring works.

Following are the links to the repositories.

- [1]Leif
- [2]Kassandra
- [3]Ruth

Take a look and let us know in the comments below what you think about it.

Greetings,

Ruth

1. https://github.com/leifb/Fowler
2. https://github.com/kfrank12/Fowler
3. https://github.com/RuWe/Fowler-Refactor

Yannick Winter (2016-05-02 11:42:25)
Hey Guys, well done! Best Regards, Yannick

Ruth (2016-05-02 12:16:32)
Hey, thanks! :-D Greetings, Ruth from LeRuKa

Cheizmann (2016-05-04 10:17:30)
Hi Leruka, your commits are well structured and your messages are comprehensible. It's quite easy to see what changes you have done by only looking at your commits/commit messages. Well done! Best Regards, Chris The Unicorn from Chezz

Ruth (2016-05-04 10:21:35)
Hey Chris! Thanks for your kind comment. :-) Greetings, Ruth from Leruka

Final Presentation – leruka (2016-06-10 20:47:28)
[…] SE _02 _HW4: Fowler Refactor […]

2.2 May

Time estimation (2016-05-04 10:29)

Hey,

after we calculated the function points for our old and new use cases we can now estimate with this how much time we will need for the new use cases.

[1]Here you can see the diagramm again where we can see how long it will take for the new use cases. As you can see the use cases "Jump" and "Duck" have only about 4 hours but we think it would take more time because the game logic is more complex and cannot be estimatet by function points. So we think it will take about 30h because it is not only jump and duck. It is necessary to calculate collisions and more what is more complex than only count DETs, RETs and FTRs.

So we calculated that each of us will spend 5h per week on coding also 15h per week all together. With this estimation we could plan our new use cases and updating our gantt chart.

Vorgangsname	Dauer	Anfang	Ende
Construction	1 Tag	Mon 04.04.16	Mon 30.05.16
UC Change Setting	6 Tage	Mon 04.04.16	Son 10.04.16
UC Duck	16 Tage	Mon 11.04.16	Son 01.05.16
UC Jump	16 Tage	Mon 02.05.16	Son 22.05.16
UC Rate Levels	6 Tage	Mon 23.05.16	Son 29.05.16

For the use cases "Jump" and "Duck" we planned 3 weeks because in this time there are different days of holiday. Because we know that some of us would spend these days with family and friends we planned that we will lose some days of work.

So far about our new time planning for this semester.

EDIT:

So we did the rough planning for our use cases in the gantt chart above to see how long it will takes to do implement and document them. After this we expanded our project plan from last semester. So we added the use cases with the estimated time and planned them detailed. Because some tasks in "Jump" and "Duck" are very similar we decided to have a task called "Game" with the similar tasks from "Jump" and "Duck" as subtasks.

So you can find our expanded project plan [2]here.

Greetings Kassandra

1. https://leruka.wordpress.com/2016/04/15/se_02_hw3-function-point-estimation/
2. https://github.com/Leruka/leruka/blob/master/docs/Projektplan_Leruka_Erweitert.pdf?raw=true

Final Presentation – leruka (2016-06-10 20:46:40)
[…] Gantt chart (new) […]

SE_02_HW5: Design Pattern (2016-05-10 21:44)

Hey!

In the last few days we have implemented a new design pattern in our code. We choose to use [1]factory methods for creating different stages (levels). The factory method pattern allows us to create objects without exposing the creation logic to the client and only refer to the newly created object by using a common interface.

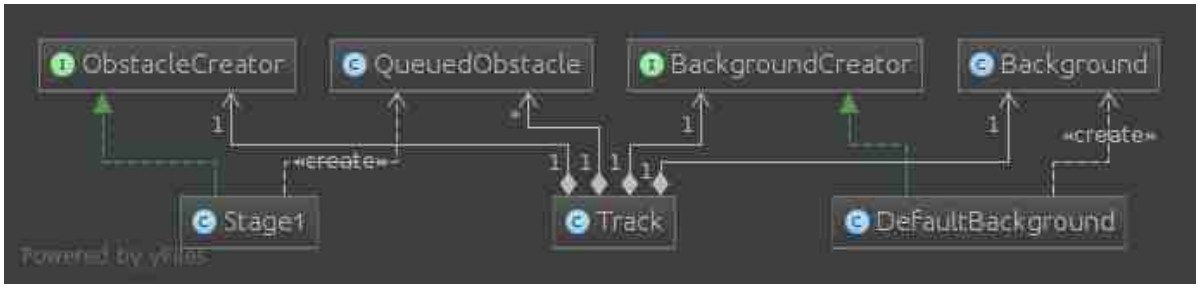
In our case, the class "Track" has been made abstract and now requires extending classes to implement certain methods that allow the creation of a stage. These methods define the shown background and the creation of obstacles. This way we can construct new stages by simply extending the class "Track", choosing a background and defining which obstacles should appear.

[code language="java"]
protected abstract Background createBackground();

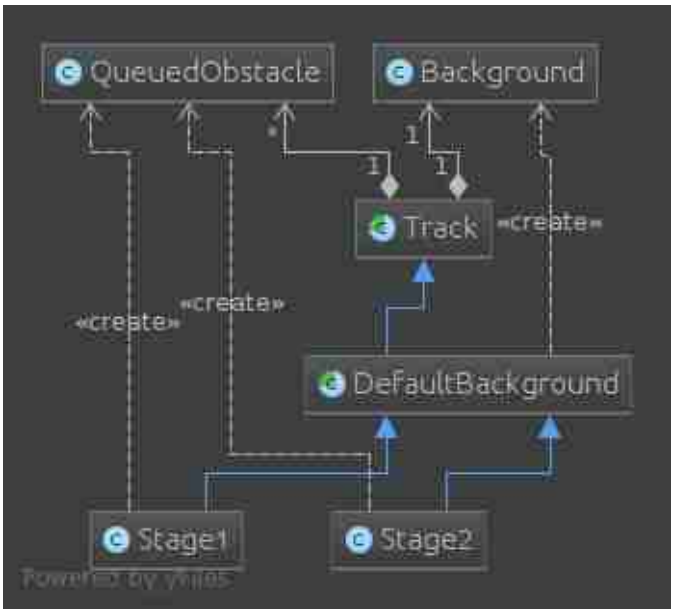
protected abstract QueuedObstacle createObstacle();
[/code]

Before using the factory methods, the class track was initiated with objects that provided the creation of a background or an obstacle. By using the new pattern we could simplify the code, delete two interfaces (one for creating a background, one for obstacles) and gave a solution for creating new stages in a very simple way.

Here you can see the class diagrams of how it was implemented before and after:



The implementation of stages without the pattern



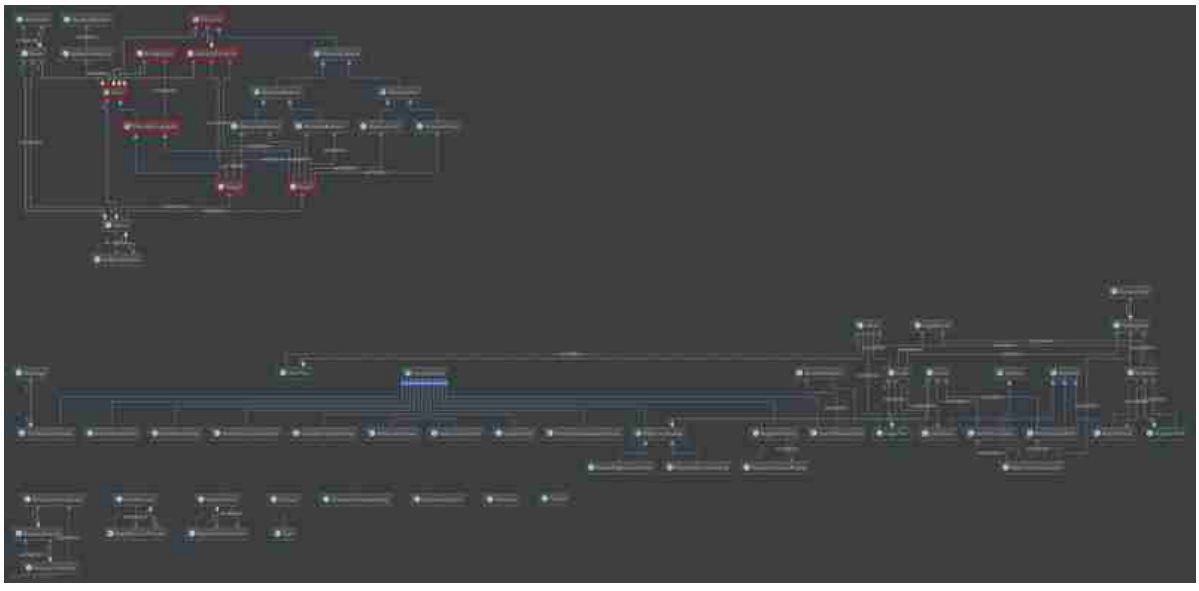
The implementation of stages with the pattern

As you can see in the second image, we even implement the two methods in separate classes. This way we can create different stages using the same background without having any code redundancy.

Please tell us what you think about this move!

EDIT

Here you can see our current overall class diagram:



Overall class diagram from 05/11/2016 (The red marked classes are part of the new pattern)

Greetings,

Leif

1. https://en.wikipedia.org/wiki/Factory_method_pattern

---

hongminhtruong (2016-05-11 09:27:16)
Hi, it's nice that you have explained in detail why this pattern is useful. One thing which is missing is the overall class diagram in which the pattern is highlighted. The rest is great! Best regards, Minh, Team MapIt

Kassandra (2016-05-11 12:01:40)
Hey Minh, thanks for your comment. We are working on our overall class diagramm so that it would contain all classes and would be readable anymore. We will add this picture later. Greetings Kassandra

Andre (2016-05-11 11:05:08)
The use of Factory pattern seems to work quite well with your project since the class diagramm has been simplified. You also explained good why you used the pattern so good job. Greetings André, Team Gaming-Bets

Kassandra (2016-05-11 12:02:31)
Hey André, thanks for your comment. Greetings Kassandra

Final Presentation – leruka (2016-06-10 20:46:55)
[...] Patterns [...]

SE_02_HW7: Test coverage (2016-05-30 19:57)

Hey guys,

this time I show you our test plan. Which you can find [1]here.

We now have over 20 % test coverage for our project you can see it on [2]SonarCube.

Please let us know what you think.

Greetings,

Ruth

1. https://github.com/Leruka/leruka/blob/master/docs/Test%20plan.pdf?raw=true
2. http://193.196.7.25/overview/coverage?id=leruka

---

mark_us (2016-06-01 09:07:36)
Amazing! You really did a great job covering your code with nearly 50 % tests! I also read through your test document, team leruka. You did a really good documentation about what your tests do, and what you expect. You even put a documentation about your metrics to the document. But while reading I noticed that you have some "tbd"s left. Regarding your environement, there are missing some system resources names. Maybe you could add them - seems not to be a big efffort. I also really like, that you have milestones for your testing. Maybe some of them can also be updated (coverage over 20 %). Best regards Markus

Ruth (2016-06-01 11:52:33)
Hey Markus! Thanks for your comment. I will update our test plan. You are right I also have to update the milestones I missed in when I uploaded the file. Best regards Ruth

TaskForge (2016-06-01 12:10:41)
Hi there, your test plan is well described and covers nearly everything you need to know. Maybe you can automate the Metrics tests from SonarQube when commiting code into your git repository. Greetings, the TaskForge team

Ruth (2016-06-01 13:41:08)
Hey team TaskForge, thanks for your comment. Our metrics tests run automatically when we start the sonarqube task in gradle. Because we think it should not do it every time. Greetings, Ruth from Leruka

Felix (2016-06-01 12:10:42)
Hi Leruka, I can not add something to your test plan, it is very well filed and seems logic. Also this test coverage is amazing! Keep this up! Greetz, Felix

Ruth (2016-06-01 12:49:50)
Hey Felix, thanks for your kind comment. :-) Best regards, Ruth

Final Presentation – leruka (2016-06-10 20:46:34)
[...] SE _02 _HW7: Test coverage [...]

SE_02_HW8: Metrics **(2016-05-31 15:28)**

Hey,

today we want to present our homework how we use metrics and which we use.

The first metric we want to show you is Cyclomatic Complexity.

We used [1]SonarQube to calculate the complexity of our project. For example in our Activity "ChangeSettingActivity" Sonar calculated 31 points. As you can see in the picture we got three if statements and one " & &" that makes 4 points of Cyclometic Complexity.

```
        // Change both
        if (changeName && changePass) {
            changeBoth(name, pw1, pw2, pwOld);
        }
        // Change name
        else if (changeName) {
            changeUsername(name, pwOld);
        }
        // Change pass
        else if (changePass) {
            changePassword(pw1, pw2, pwOld);
        }
        // Nothing to change
        else {
            Message.showMessage("Keine Änderungen");
            this.goToMainActivity();
        }
    }
```

As solution we changed the method by reorganize the structure of the method. As you can see in the picture we now have just 3 if-statements and avoided the " & &" so we decreased our Cyclomatic Complexity about one point.

```
// Change Name
if (changeName) {
    // Change Both
    if(changePass) {
        changeBoth(name, pw1, pw2, pwOld);
    } else {
        changeUsername(name, pwOld);
    }
}
// Change pass
else if (changePass) {
    changePassword(pw1, pw2, pwOld);
}
// Nothing to change
else {
    Message.showMessage("Keine Ã„nderungen");
    this.goToMainActivity();
}
```

[2]Here you can find the link to the commit so you can track our changes.

For our class "Hitbox" we didn’t change our method in the picture. Because we are in a statement and if we return directly the result of collides(r) we would interrupt the method directly so even if it is false.

```
public boolean collides(Hitbox hitbox) {
    for (Rect r : this.rects) {
        if (hitbox.collides(r)) return true;
    }
    return false;
}
```

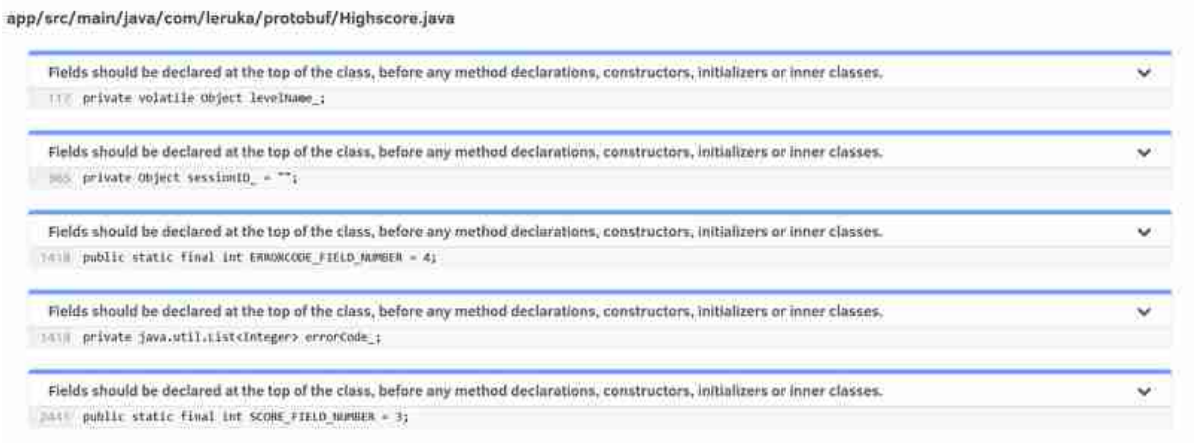
Our second metric we used is code style, which is shown us by [3]codacy. This you can see for example in our HttpPost class. There exists a method which name is starting with a uppercase letter.

We improved this by refactoring the name of the method.

```
abstract protected T createResponseObject(InputStream in);
```

[4]In this commit you can track what we changed exactly.

In this metric there are also some spotted issues we didn’t fix because these classes like that one in the picture are generated classes from Protobuf.



[5]Here you can find the test plan which includes also screenshots from our metrics.

Let us know what you think about it.

Greetings,

Leruka

1. http://193.196.7.25/overview/structure?id=leruka
2. https://github.com/Leruka/leruka/commit/1de66de13d9dbd56099a3f3587fcde6287a94bda
3. https://www.codacy.com/app/leruka/leruka/dashboard
4. https://github.com/Leruka/leruka/commit/c777291f4c82a18bd4bd94b59762cc828235547f
5. https://github.com/Leruka/leruka/blob/master/docs/Test%20plan.pdf?raw=true

---

mark_us (2016-06-01 08:52:46)
Hi team leruka! First let me say: good work! You really explained good understandable how your metrics work and what you did to refactor according to those metrics. I searched through your commits on GitHub to get an idea about what you refactored. Between all your commits (I know you are working a lot), it was a bit difficult to find all those which belong to this blog post. Maybe you could have posted links to all of the relevant commits. This would make it even more easy to follow your thoughts. Best regards Markus

Kassandra (2016-06-01 11:50:18)
Hey Markus, thanks for your comment. Yes your right! It would be better if we link the relevant commits of code before and after. We will add these links! Greetings Kassandra from Leruka

Felix (2016-06-01 11:54:11)
Hey Leruka, you really made clear what was your point and how you solved these issues. Good work,also your test plan is well filled. Greetz Felix

Kassandra (2016-06-01 12:00:44)
Hey Felix, thanks for your comment. We are glad that our test plan is well filled out. Greetings Kassandra from

Leruka

Kevin K. (2016-06-01 12:09:51)
Hi Leruka, I liked the fact, you embedded the refactored code, so the changes where quite easy to spot. You described the reasons for changing / not changing your code. The linked Test plan shows the test metrics in a clean easy to read style. Kind Regards, Kevin@TaskForge

leruka (2016-06-09 08:16:43)
Hey, thanks for your kind comment. It is good that you could reenact, what we did with our code. :-) Greetings, Leruka

Final Presentation – leruka (2016-06-10 20:46:57)
[…] SE _02 _HW8: Metrics […]

2.3 June

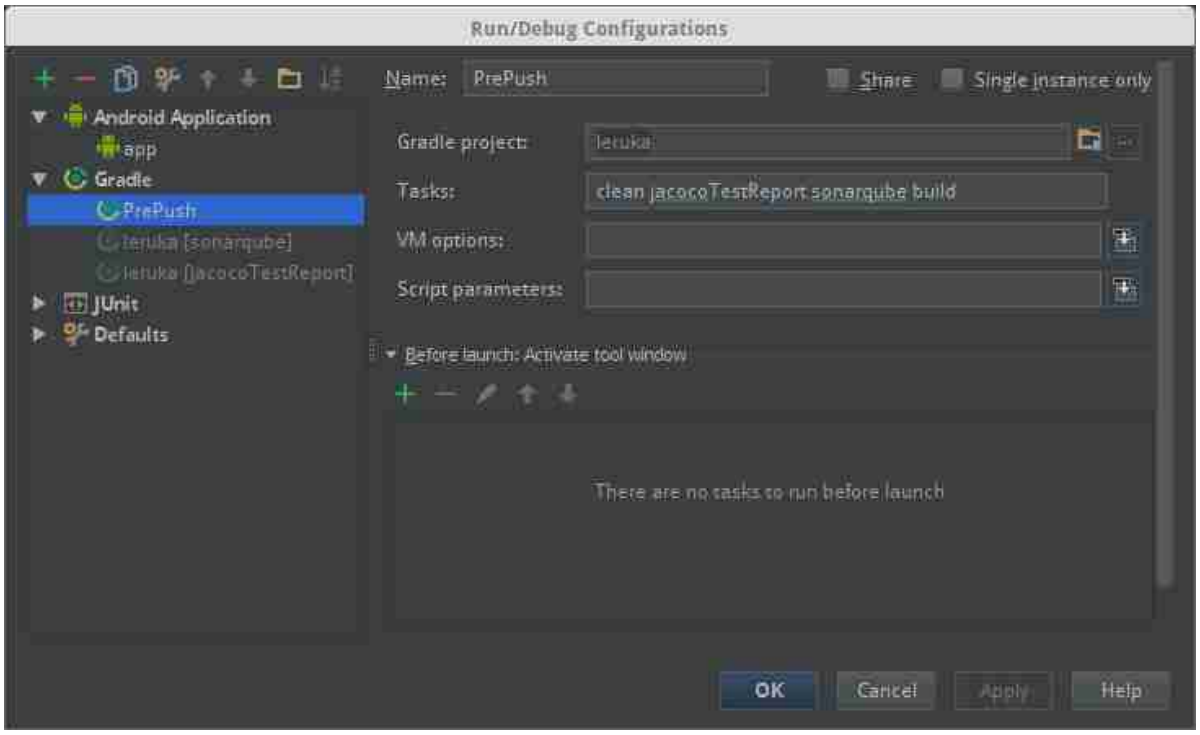
SE_02_HW9: Deployment (2016-06-06 18:37)

Hey!

In this entry we will explain you how we do our "automatic deployment".

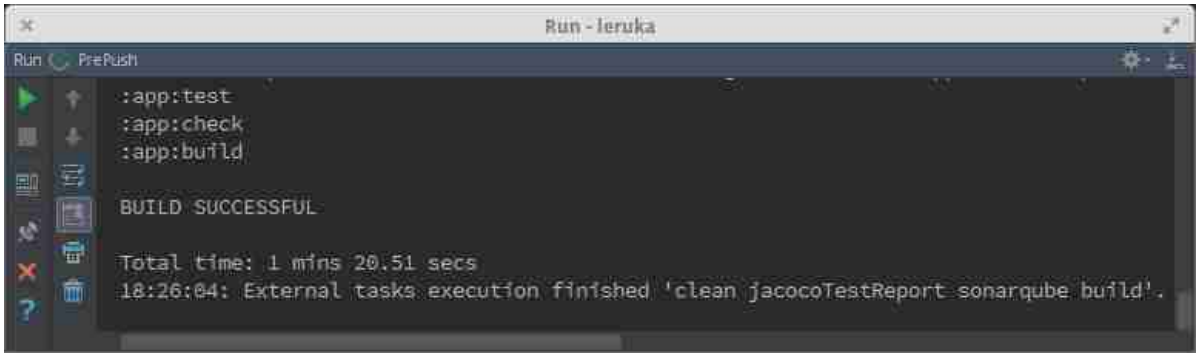
Because we implemented an Android app there is no chance for us to use Jenkins to auto deploy our app. Because of that we talked with our prof and she decided that we do not need to force it and lose to much time.

Our alternative is we write a "schnieke" run configuration :-D. You can see it in the screenshot.



This file is a IntelliJ internal configuration, which runs the tests, launch the jacoco plugin for reports and then uploads the reports to SonarQube.

In the following you can see how it succeeded.



Greetings,

Ruth

---

mark_us (2016-06-07 13:26:35)
Hi Ruth, I totally understand that you cannot do an autodeployment with an app. I even think this is not the idea behind it. Normally you would publish the app to the app store and fullfill their criteria. So what you did is quite perfect. I like your work and congratulate for everythinng you reached! Yours Markus

Leif (2016-06-10 14:52:31)
Hi Markus! Thanks for your kind comment! Glad to hear that you understand our problem. Greetings, Leif

Cheizmann (2016-06-10 14:48:49)
Hey leruka, We're quite in the same situation because it's also not possible to deploy our ios application Chezz automatically. But I like that you thought of an alternative instead! Regards, Chris the Unicorn Chezz - The Rating Game

Leif (2016-06-10 14:55:37)
Hey Chris, I think there is some way if you deploy your application to the Play Store / App Store, but as we did not jet publish our APP, that's not possible, too. At least it is not easy, sadly. Thanks for your comment! Leif

Final Presentation – leruka (2016-06-10 20:47:00)
[...] SE _02 _HW9: Deployment [...]

SE_02_HW10: Installation (2016-06-06 19:37)

Hey Guys!

As the code for our APP, the code for our server is public on Github, too. But we have never told you how to set up your very own leruka server. We want to change that today!

You will need a tomcat server with a local MySQL database. Guides how to install these can be found [1]here (tomcat) and [2]here (MySQL).

First of all the tomcat server needs to know the credentials of your MySQL database. It will search for them in the following file: "/home/leruka/serverHandler/credentials". The file has to look like the following example:

[sourcecode language="text"]db=leruka
user=*your user*
pass=*your password*[/sourcecode]

After you have done this, you will need two files: The tomcat war-file and the SQL creation-script. You will find these on the [3]"releases"-page of our lerukaServer Github repository. Download the files and do the following:

- Make sure that your MySQL server has no database named "leruka"
- Apply the SQL creation script (In MySQL Workbench: File > Open SQL Script...)
- Open the tomcat manager App ([4]how to)
- Under "Deploy" > "WAR file to deploy", upload the war file and click "Deploy"

Your own leruka server is now live and ready to use!

If you want the APP to use your new server, you will need to adjust the IP addresses. You find them in one file: "app/src/main/res/values/urls.xml".

Greetings,

Leif

1. https://tomcat.apache.org/tomcat-7.0-doc/setup.html
2. https://dev.mysql.com/doc/refman/5.7/en/installing.html
3. https://github.com/Leruka/lerukaServer/releases
4. https://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html

---

Felix (2016-06-10 09:07:54)
Hi Leruka, It worked!!! Great job, there were little issues but they all were easy to fix!! Issues regarding following things: I wasn´t quite sure which kind of file this is supposed to be: „/home/leruka/serverHandler/credential s“. Also maybe you can add an example to prove the setup worked, for example an http request in the browser or some thing like this. Greetz, Felix

Leif (2016-06-10 13:20:30)
That’s good to hear! The file is just plain text, there is no need for a file extension. In fact, the server will use javas „java.util.Properties“ package to parse the content. The text encoding will normally be detected automatically. The easiest way to check if the server is running is to fetch to global scores, because it is just an HTTP GET which does not require for any input. The URL is: http://address-or-ip.de:8080/leruka/score Be aware that you might not understand the structure of the response, cause we are using Googles protocol-buffers to transmit data. The HTTP status code will tell you, if everything worked. Thanks for testing the installation and commenting here! Greetings, Leif

Leif (2016-06-10 13:18:30)
That’s good to hear! The file is just plain text, there is no need for a file extension. In fact, the server will use javas "java.util.Properties" package to parse the content. The text encoding will normally be detected automatically. The easiest way to check if the server is running is to fetch to global scores, because it is just an HTTP GET which does not require for any input. The URL is: http://(your IP):8080/leruka/score Be aware that you might not understand the structure of the response, cause we are using Googles protocol-buffers to transmit data. The HTTP status code will tell you, if everything worked. Thanks for testing the installation and commenting here! Greetings, Leif

Final Presentation – leruka (2016-06-10 20:46:49)
[…] SE _02 _HW10: Installation […]

Final Presentation (2016-06-10 15:10)

Hi there!

We, Leif, Kassandra and Ruth, are students at the cooperative university of Karlsruhe.

We want to make an Jump'n'Run game an the game mechanic should be dodging obstacles. There are different levels which you can play and if you are a logged user you could find your name in the highscore with other gamers.

The semester is over and here are our results of the project.

Requirements

:

UC

- [1]Overall Use Case Diagram
- [2]UC Duck
- [3]UC Jump
- [4]UC Login
- [5]UC Rate Levels
- [6]UC Register
- [7]UC Change Settings
- [8]UC Start Game
- [9]UC View instruction
- [10]UC View personal highscore

- [11]UC View public highscore

SRS

[12]Software Requirements Specification

[13]to the blogentry

T
est Cases

- See feature files in the use case documents.
- [14]On Github

Test Log

testcoverage

- [15]feature files
- [16]SonarQube
- [17]Test with enduser evaluation
- [18]Test with enduser statistics

[19]to the blogentry

Test Plan

[20]RUP Testplan

Functional Test

- See feature files in the use case documents.
- [21]On Github

Unit Test and Test with enduser

- [22]Unit Tests
- Test with enduser:
  - [23]Test with enduser evaluation
  - [24]Test with enduser statistics
  - [25]Test with enduser survey

Test Driven Developement
[26]TDD

Project Management:

RUP gantt chart (past)

future long-term plan using FP estimation

Hours/ team member

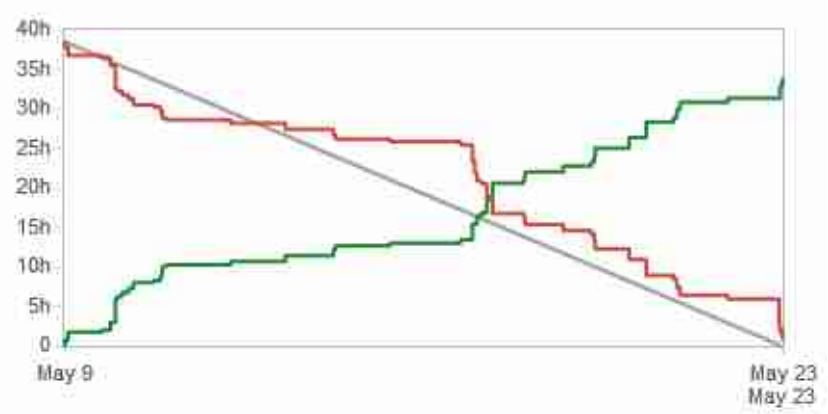
Team roles

- [27]Gantt chart (past)
- [28]Gantt chart (new)
  - [29]to the blog entry
- [30]Function Points
- [31]Hours/ team member
- [32]RUP Role allocation

Jira / burndown

- [33]Jira

Burndown:



(Sprint Const #1, Const #2 and Const #3 in the picture)

FP calculation

- [34]Function Points

Ability to Execute

Demo

- [35]Demo Download .apk on your smartphone and install it

Code

- [36]Github
- [37]Installation

Quality:

Architecture

SAD document

- [38]Architecture
  - [39]to the blogentry
- [40]Software Architecture Document
- [41]Patterns

Configuration Management/ Environmental Setup

See in [42]SRS part "Technologies".

Metrics

- [43]Metrics

Risk Management

- [44]Risk Managment

(living document – obsolete risks are ~~canceled~~ in the document)

Automated Testing

automatic with deploy

- [45]RUP Testplan

- [46]Automatic Deployment

Pattern

- [47]Patterns

Other:

Presentations

- [48]Midterm Presentation
- [49]Final Presentation

Block entries

- [50]Project Vision
- [51]W2 – Team roles and technologies
- [52]HW3: Software Requirement Specification
- [53]HW4: Use Cases
- [54]HW5: Feature Files
- [55]HW5: UML Class diagram
- [56]HW7: Software Architecture Document
- [57]HW8: Jira
- [58]HW 9: “Gantt”
- [59]Updated overall use case diagram
- [60]SE _2 _HW2: Risk Management, Time Estimation and new Use Case Documents
- [61]SE _02 _HW3: Function point estimation
- [62]SE _02 _HW3 Test Driven Development
- [63]SE _02 _HW4: Fowler Refactor
- [64]Time estimation
- [65]SE _02 _HW5: Design Pattern
- [66]SE _02 _HW7: Test coverage
- [67]SE _02 _HW8: Metrics
- [68]SE _02 _HW9: Deployment
- [69]SE _02 _HW10: Installation

1. https://raw.githubusercontent.com/Leruka/leruka/master/docs/UseCaseSurvey.jpg
2. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCDuck.pdf?raw=true
3. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCJump.pdf?raw=true
4. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCLogin.pdf?raw=true
5. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCRatelevels.pdf?raw=true
6. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCRegister.pdf?raw=true
7. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCSettingsChange.pdf?raw=true
8. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCStartGame.pdf?raw=true
9. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewInstruction.pdf?raw=true
10. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewpersonalhighscore.pdf?raw=true
11. https://github.com/Leruka/leruka/blob/master/docs/rup_srs_UCViewpublicHighscore.pdf?raw=true
12. https://github.com/Leruka/leruka/blob/master/docs/rup_srs.dot.pdf?raw=true
13. https://leruka.wordpress.com/2015/10/22/hw3-software-requirement-specification/
14. https://github.com/Leruka/leruka/tree/master/features
15. https://rawgit.com/Leruka/leruka/master/TestsKomplett.html
16. http://193.196.7.25/overview/coverage?id=leruka
17. https://github.com/Leruka/leruka/blob/master/docs/AuswertungderEndUserTests.xlsx?raw=true
18. https://github.com/Leruka/leruka/blob/master/docs/Statistik_Enduser_Umfrage.PNG?raw=true
19. https://leruka.wordpress.com/2016/05/30/se_02_hw6-test-coverage/
20. https://github.com/Leruka/leruka/blob/master/docs/Testplan.pdf?raw=true
21. https://github.com/Leruka/leruka/tree/master/features
22. https://github.com/Leruka/leruka/tree/master/app/src/test/java/com/leruka/leruka
23. https://github.com/Leruka/leruka/blob/master/docs/AuswertungderEndUserTests.xlsx?raw=true
24. https://github.com/Leruka/leruka/blob/master/docs/Statistik_Enduser_Umfrage.PNG?raw=true
25. https://github.com/Leruka/leruka/blob/master/docs/Befragung.pdf?raw=true
26. https://leruka.wordpress.com/2016/04/25/se_02_hw3-test-driven-development/
27. https://github.com/Leruka/leruka/blob/master/docs/Projektplan_Leruka.pdf?raw=true
28. https://github.com/Leruka/leruka/blob/master/docs/Projektplan_Leruka_Erweitert.pdf?raw=true
29. https://leruka.wordpress.com/2016/05/04/time-estimation/
30. https://leruka.wordpress.com/2016/04/15/se_02_hw3-function-point-estimation/
31. https://leruka.files.wordpress.com/2015/10/leruka1.pdf
32. https://github.com/Leruka/leruka/blob/master/docs/RUP.png?raw=true
33. http://193.196.7.27:8080/secure/RapidBoard.jspa?rapidView=15&view=planning.nodetail
34. https://leruka.wordpress.com/2016/04/15/se_02_hw3-function-point-estimation/
35. https://github.com/Leruka/leruka/releases
36. https://github.com/Leruka/leruka
37. https://leruka.wordpress.com/2016/06/06/se_02_hw10-installation/
38. https://github.com/Leruka/leruka/blob/master/docs/Technologien.png?raw=true
39. https://leruka.wordpress.com/2015/10/09/w2-team-roles-and-technologies/
40. https://github.com/Leruka/leruka/blob/master/docs/SoftwareArchitectureDocument.pdf?raw=true
41. https://leruka.wordpress.com/2016/05/10/se_02_hw5-design-pattern/
42. https://github.com/Leruka/leruka/blob/master/docs/rup_srs.dot.pdf?raw=true
43. https://leruka.wordpress.com/2016/05/31/se_02_hw8-metrics/
44. https://github.com/Leruka/leruka/blob/master/docs/RiskManagement-risk.pdf?raw=true
45. https://github.com/Leruka/leruka/blob/master/docs/Testplan.pdf?raw=true
46. https://leruka.wordpress.com/2016/06/06/se_02_hw9-deployment/

47. https://leruka.wordpress.com/2016/05/10/se_02_hw5-design-pattern/
48. https://drive.google.com/file/d/0B9VagljXVQ-OYWhiUUZTWHpjNjQ/view?usp=sharing
49. https://drive.google.com/file/d/0B9VagljXVQ-Oak42NkJneWlIclk/view?usp=sharing
50. https://leruka.wordpress.com/2015/10/02/project-vision/
51. https://leruka.wordpress.com/2015/10/09/w2-team-roles-and-technologies/
52. https://leruka.wordpress.com/2015/10/22/hw3-software-requirement-specification/
53. https://leruka.wordpress.com/2015/10/25/hw4-use-cases/
54. https://leruka.wordpress.com/2015/11/01/hw5-feature-files/
55. https://leruka.wordpress.com/2015/11/10/hw5-uml-class-diagram/
56. https://leruka.wordpress.com/2015/11/18/hw7-software-architecture-document/
57. https://leruka.wordpress.com/2015/11/24/hw8-jira/
58. https://leruka.wordpress.com/2015/12/02/hw-9-gantt/
59. https://leruka.wordpress.com/2016/04/04/updated-overall-use-case-diagram/
60. https://leruka.wordpress.com/2016/04/08/se_2_hw2-risk-management-time-estimation-and-new-use-case-documents/
61. https://leruka.wordpress.com/2016/04/15/se_02_hw3-function-point-estimation/
62. https://leruka.wordpress.com/2016/04/25/se_02_hw3-test-driven-development/
63. https://leruka.wordpress.com/2016/04/30/se_02_hw4-fowler-refactor/
64. https://leruka.wordpress.com/2016/05/04/time-estimation/
65. https://leruka.wordpress.com/2016/05/10/se_02_hw5-design-pattern/
66. https://leruka.wordpress.com/2016/05/30/se_02_hw6-test-coverage/
67. https://leruka.wordpress.com/2016/05/31/se_02_hw8-metrics/
68. https://leruka.wordpress.com/2016/06/06/se_02_hw9-deployment/
69. https://leruka.wordpress.com/2016/06/06/se_02_hw10-installation/