# Project plan+study diary

# Cygnus 2022

## version 1.4

| | | |
|---|---|---|
| TUT | Pervasive Computing | TIE-21106 Software Engineering Methodology |
| Author: Ignacio Laviña | | Printed: 12.03.2017 20:56 |
| Distribution: Team members, customer | | |
| | | |
| | | |
| | | |
| Document status: working on it | | Modified: 12.03.2017 20:56 |

## VERSION HISTORY

| Version | Date | Authors | Explanation (modifications) |
|---|---|---|---|
| 1.0 | 29.01.2017 | Ignacio L. | Initial version |
| 1.1 | 10.02.2017 | Victor G. | Added dependencies of stories |
| 1.2 | 12.02.2017 | Ignacio L. | Added review of sprint 1 |
| 1.3 | 06.03.2017 | Victor G. | Corrections of sprint 1 |
| 1.4 | 12.03.2017 | Ignacio L. | Added review of sprint 2 |

# TABLE OF CONTENTS

# 1. PROJECT RESOURCES

On this part of the document, a brief presentation of the team background and skills is exposed, also describing the process of the work flow, tools and technologies used during the project.

## 1.1 Personnel

This chapter of the document exposes the team members and a short view of their capacities, roles and other documentation needed.

The team is composed by four members: Ignacio Laviña, Victor Garcia, Iaroslav Gridin, Likai Ren.

| | |
|---|---|
| | **Ignacio Laviña Faustmann (Product owner)** |
| **Contact** | Ignacio.lavinafaustmann@student.tut.fi<br>+34677804196 |
| **Experience** | App developer and tester<br>(February 2015 – April 2016) |
| **Skills** | Team work<br>Active learning<br>Java<br>Creativity |
| **Interests** | New technologies, entrepreneurship |

| | |
|---|---|
| | **Víctor García Zarco (Scrum master)** |
| **Contact** | victor.garciazarco@student.tut.fi<br>+34666740213 |
| **Experience** | Project developer @ Demola Tampere<br>(October 2016 – January 2017)<br>Frontend & Backend lead developer @ Hightrack<br>(August 2015 – February 2016)<br>Talentum Startups @ Telefónica<br>(December 2014 – May 2015) |
| **Skills** | Frontend development (HTML5, CSS3, Backbone.js) |

| | |
|---|---|
| | Backend development (Java, C, PHP)<br>Photoshop<br>Project management |
| **Interests** | New technologies, entrepreneurship |

| | |
|---|---|
| | **Iaroslav Olegovich Gridin** |
| Contact | iaroslav.gridin@student.tut.fi<br>+358449165346 |
| Experience | Self-employed freelancer (2009-2015)<br>Research assistant @ TUT (March 2016 – December 2016) |
| Skills | Ruby on Rails backend development<br>C++, C, Ruby, Haskell, Go |
| Interests | Data exchange networks, video games |

| | |
|---|---|
| | **Likai Ren** |
| Contact | likai.ren@student.tut.fi<br>+358466143860 |
| Experience | Web Designer Intern @EasyMarketing Finland Oy Ab (July 2015-September 2015) |
| Skills | Programming Language: Python, Java, JavaScript, C++, C, C#, PHP;<br>Framework: Flask, Django, React, Bootstrap |
| Interests | Web development;<br>Ethical hacking |

### 1.1.1 Estimated contribution

| | |
|---|---|
| **Ignacio Laviña Faustmann** | 20h/sprint |
| **Víctor García Zarco** | 20h/sprint |
| **Iaroslav Gridin** | 20h/sprint |
| **Likai Ren** | 20h/sprint |

### 1.1.2 Team’s absence

| | |
|---|---|
| **Ignacio Laviña Faustmann** | 27 February – 4 March |
| **Víctor García Zarco** | 26 – 29 Jan, 27 Feb – 5 Mar |
| **Iaroslav Gridin** | |
| **Likai Ren** | |

## 1.2 Process description

The team members will have a meeting before each sprint to review the previous sprint, define goals and achievements, define the next sprint and split the task and work according to the Agilefant plan.

During the process, there will be active communication between the team members through slack, and other ways if it’s necessary. The team members are committed to be active in communication, and ask for others feedback or help if it becomes necessary.

The individual tasks are defined before each sprit in Agilefant, always with the team agreement. previous agreement. Some task will require a group meeting for developing together and solving problems.

## 1.3 Tools and technologies

*Table 1.1: Tools used in the project.*

| Purpose | Tool | Contact person | version |
|---|---|---|---|
| Documentation | MS Word (word processing)<br>office.microsoft.com | | 2015 |
| | ArgoUML (UML tool)<br>http://argouml.tigris.org/ | V.G.Z | 16.9 |
| Communication | Slack<br>http://slack.com | | 2.3.4 |
| | Outlook<br>https://outlook.live.com | | 2017 |
| | Mutt<br>http://www.mutt.org/ | | 20170113 (1.7.2) |
| Version management | Git<br>https://git-scm.com | V.G.Z | 2.11.0 |
| | GitLab<br>https://gitlab.rd.tut.fi/ | | 2017 |
| Project management | Agilefant<br>https://www.agilefant.com/ | I.L.F | 2017 |

| | | | |
|---|---|---|---|
| | One Drive<br>https://onedrive.live.com | | 2017 |
| Development | Processing<br>https://processing.org/ | | 3.2.3 |
| | ControlP5 library<br>http://www.sojamo.de/libraries/controlP5/ | | 2.2.6 |

As the duration of the project is less than 5 months if there is one new version of one tool/software we will ignore it, continuing with the current version (unless that version fixes security or important problems). The short duration of the project shouldn't be a problem for different versions.

**Version control repository**
The repository of the project is hosted in GitLab. The team have full access to it, while the customer will have only access to the *master* branch. Here, the customer will find the latest working version of the project.

**Agilefant**
Project management is done using Agilefant. Customer requirements are made into user stories, then they are converted to backlogs and distributed between sprints based on difficulty, dependencies and value. Then tasks based on backlogs are distributed among team members based on their capabilities and preferences. Team velocity is tracked and allows better time allocation in future.

**Processing**
The main development work is done by Processing, which is an integrated development environment (IDE) and a programming language for visual arts. Processing is open source and free to use in multiple platform, including Linux, Mac OS X and Windows. Processing can be used to create interactive programs with 2D, 3D or PDF output. With OpenGL integrated for accelerated 2D and 3D, Processing even has more than 100 libraries extending the core software.

## 1.4 SPRINT BACKLOGS

After analyzing the requirements given by the customer, some user stories have been made to manage them easily. Also, as Processing is a new technology for the team, the learning curve will increase exponentially from the beginning (personnel with experience in different fields). Because of that, the first phases of the projects will contain less workload than the final ones.

The picture below shows the dependencies between the different user stories. These dependencies are required to know when a new story can be started (based on the previous ones).

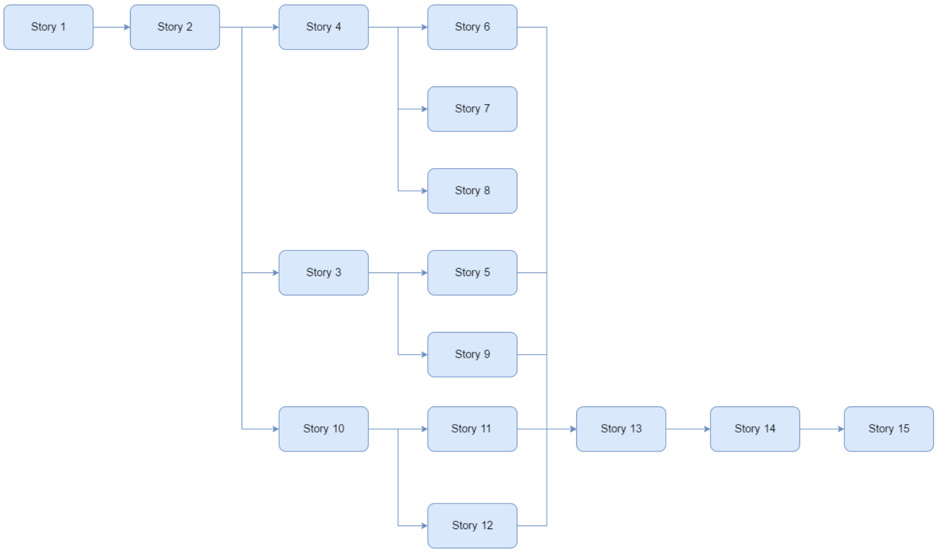


*Figure 1 - Dependencies between stories*

**NOTE**: More detailed information about the stories (with the linked requirements and tasks) can be found in Agilefant.

## 2. STUDY DIARY

### 2.1 Sprint 1

#### 2.1.1 What went well

In this first sprint there were many things that went well for the team:

The group started move on, the game has the welcome screen and part of the story game. Also the user can introduce the name that would be used for the story and the score of the game.

All the user stories were submited to agilefant and the work for the rest of the project was splited in the different sprints to have an initial idea about the work distribution.

Some creatives ideas come to the story of the game, and great drawings were implemented for the stories.

The team had continuos communication through the communication channels.

#### 2.1.2 What difficulties you had

The main difficulties were related to the processing learning, despite the team has coding skills, when facing a new programming environment some difficulties appear.

Thanks to some examples provided on the own processing environment and other tutorials found on the internet the team started learning and developing the first tasks.

### 2.1.3 What were the main learnings

During this sprint the team has achieved some learnings related to:

Scrum methodology: In contrast to the task defined in the first sprint, on the next sprints the task are much more specific and concretes. So splitting the work will be easier for the next parts of the project.

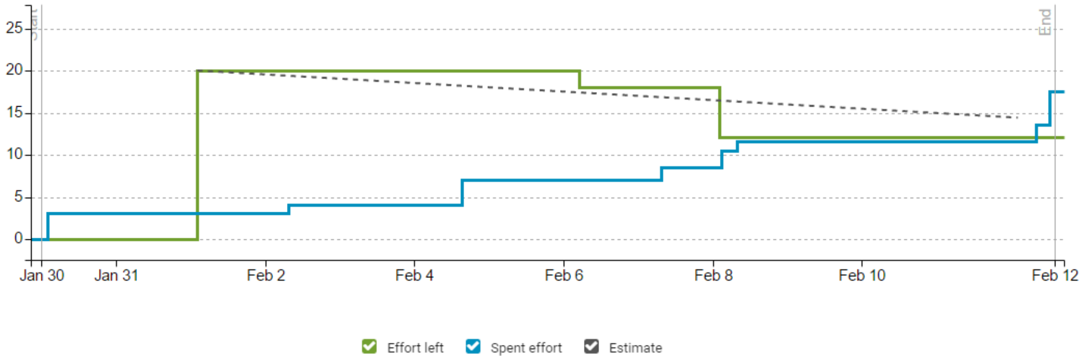
Git methodology: A part of the team wasn't used to Git methodology, so thanks to the team support and some tutorials all the team has now the knowledge of Git.

Processing: Learning processing while developing the game is something that will appear during all the parts of the project, as no one of the members had worked in processing before.

### 2.1.4 What did you decide to change for the next sprint

As is said above, for the next sprint the task are more specific, so will be easier to split the work and define responsibilities.

### 2.1.5 Burndown analysis



The burndown graphic shows the evolution of the sprint in terms of productivity. At the beginning, the team forgot to update the spent time so the effort left was not modified while the spent one was increasing. The distribution of the graphic is balanced, having at the end more work than the expected.

## 2.2 Sprint 2

### 2.2.1 What went well

During this Sprint we decided to develop fifteen user stories, the increase is quite significant from the first Sprint. This was possible thanks to the individual learning time that we took on the first Sprint. This made us more confident to develop much more stories and tasks.

In this Sprint we were able to include the map, the jet, enemies, islands and fuel repostages, and also implement the movement of all of this elements. This was great because is the main part of the game, and now the game has most of the difficult elements that we have to implement.

This also generates in the team sensation of success and motivates us to keep learning and developing

### 2.2.2 What difficulties you had

While it is true that we manage to implement the most diffcult parts, we found some problems that we needed to solve:

All the new elements included needed interaction between them, splittig the taks made neccessay to explain each part of the implementation in order to facilite the others members interact with all the parts.

When merging the individual branch into the main branch on Git called development, some merge conflicts appeared and it took some time to solve all of them in a succesfull way.

Also, in the graffic is possible to see that the main effort of the Sprint is spent at the beginning and at the end of the Sprint.

The team didn’t work on the project during the exams week, part of the team couldn’t work because of a trip, as it was programmed in the Project resources/team abscence. Others because of other courses and exams.

Also it is possible to notice that in the next week after exams the team restablish the work flow on Thursday, four days before the deadline with the most time consuming tasks. This could be a bad time management of the team.

### 2.2.3 What were the main learnings

This Sprint has allowed the team learn much more about processing. New functionalities of processing where implemented in this part.

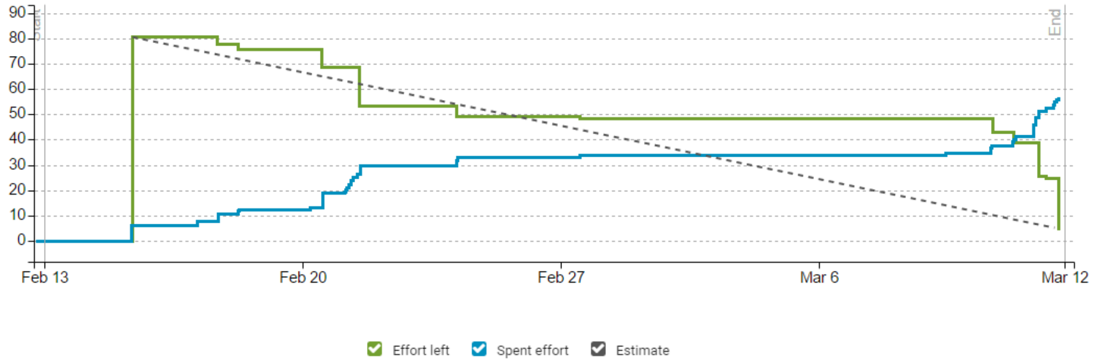
With the time, the team keep improving the team work methodology and the continuous communication through slack.

### 2.2.4 What did you decide to change for the next Sprint

For the next Sprint the team is going to take some time at the beginning to review the current code and give it a clear structure.

Also the team should try to redistribute the own way of spent effort in a more regular way, avoiding accumulate to much work on the end of the Sprint.

### 2.2.5 Burndown graphic



The burndown graphic shows the evolution of the project during the second sprint. At the begining, the results obtained were in line with the estimation of workload. Due to the exams week, after 27th of February there is a break in terms of work. Finally, the last days before the deadline the team had to work to finish on time, spending more effort than the estimated (due to some complications with the code). Also, there effort left at the end doesn't have a value of zero: there are two stories deferred, because the team agreed on to fix and refactor the current code before including new features.

## 3. RISK MANAGEMENT PLAN

The ID of the risks is defined by the pattern XY, where:

- X refers to the category of the risk.
  - P: Project management
  - T: Technologies
  - C: Customer
  - E: Environment
  - Pe: Personnel

*Table 4.1: Project risks.*

| **Risk ID** | **Description** | **Probability** | **Impact** |
|---|---|---|---|
| P1 | Bad scheduling of the project/sprints | 2 | 3 |
| T1 | Online tool not available | 2 | 2 |

| | | | |
|---|---|---|---|
| T2 | Learning of new technologies | 1 | 2 |
| T3 | HW problems with the equipment | 1 | 2 |
| C1 | Bad communication with the customer | 3 | 2 |
| C2 | Unclear requirements | 3 | 2 |
| C3 | Number of requirements increased | 2 | 1 |
| E1 | External attack to own systems | 1 | 3 |
| E2 | Internet connection lost | 1 | 1 |
| Pe1 | Short term absence | 2 | 3 |
| Pe2 | Long term absence | 1 | 3 |
| Pe3 | Bad communication within the team | 1 | 2 |
| Pe4 | Overload of work | 1 | 1 |
| Pe5 | Change of job (leaving the team) | 1 | 2 |

## 3.1 Project management risks

### 3.1.1 Risk P1: Bad scheduling of the project/sprints

**Symptom, early warning sign:** not enough time to finish on time.
**Source or reason:** bad scheduling of the times of the project/sprints due to historical data.
**Probability:** 2 medium (on scale 1-3)
**Seriousness:** 3 high (on scale 1-3)
**How to avoid:** think well about what is able to do each member of the team in the scheduled time.
**How to prevent:** reorganize the tasks and workload for each member.
**How to survive:** good relationship with the customer, allowing small changes on the plan.

## 3.2 Technology risks

### 3.2.1 Risk T1: Online tool not available

**Symptom, early warning sign:** delays or no access to the online tool.
**Source or reason:** external problem with the provider.

**Probability:** 2 medium (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** selection of the best tool provider.
**How to prevent:** premium accounts use to have preferences for this situation.
**How to survive:** existing alternative to work (other online platform or offline work).

### 3.2.2 Risk T2: Learning of new technologies

**Symptom, early warning sign:** the speed of the users working is not efficient.
**Source or reason:** the team doesn't know the new technology.
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** provide a solution for the customer with known technologies.
**How to prevent:** give all the necessary information to improve the skills with the technology.
**How to survive:** temporary hiring of a new member who knows the technology.

### 3.2.3 Risk T3: HW problems with the equipment

**Symptom, early warning sign:** disk makes noise, arbitrary reading errors occur more often than before.
**Source or reason:** hard disk is at the end of its lifespan, or hard hit on computer while disk was running.
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** buy a new disk when starting a project.
**How to prevent:** when first symptoms occur, take additional back-ups and change the disk as soon as possible.
**How to survive:** back-ups, and a replacement disk or whole computer.

## 3.3 Customer risks

### 3.3.1 Risk C1: Bad communication with the customer

**Symptom, early warning sign:** the customer doesn't receive the product expected.
**Source or reason:** lack of communication with the client
**Probability:** 3 high (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** define regular meetings and ways of communication.
**How to prevent:** increase the number of meetings with the customer.

**How to survive:** redefine the ways of communication and have a meeting with the customer to review all the work.

### 3.3.2 Risk C2: Unclear requirements

**Symptom, early warning sign:** the customer is not receiving the product expected.
**Source or reason:** bad or lack of communication with the customer. Maybe the customer doesn't really know what he wants.
**Probability:** 3 high (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** define regular meetings and help the customer to decide.
**How to prevent:** stop the project and redefine the requirements.
**How to survive:** meeting to redefine the requirements and the project.

### 3.3.3 Risk C3: Number of requirements increased

**Symptom, early warning sign:** the customer asks for new functionalities.
**Source or reason:** the preferences of the customer have changed-
**Probability:** 2 medium (on scale 1-3)
**Seriousness:** 1 low (on scale 1-3)
**How to avoid:** closed budget and requirements list before starting the project.
**How to prevent:** meeting with the customer to decide if it is possible to increase the number of the requirements.
**How to survive:** re-schedule the work.

## 3.4 Environment risks

### 3.4.1 Risk E1: External attack to own systems

**Symptom, early warning sign:** alarms in the firewall and other systems.
**Source or reason:** external attack (DDoS, for example)
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 3 high (on scale 1-3)
**How to avoid:** define and implement a good security plan
**How to prevent:** isolate the infected device
**How to survive:** backups of all the important data.

### 3.4.2 Risk E2: Internet connection lost

**Symptom, early warning sign:** delays and lack of connectivity to internet.
**Source or reason:** external attack, failure in internal network, failure in network cards.

**Probability:** 1 low (on scale 1-3)
**Seriousness:** 1 low (on scale 1-3)
**How to avoid:** review of systems and devices. Reliable network provider.
**How to prevent:** identify the problem (HW or network provider) and look for an alternative.
**How to survive:** possibility of working in offline mode.

## 3.5 Personnel risks

### 3.5.1 Risk Pe1: Short term absence

**Symptom, early warning sign:** one team member is missing for some days.
**Source or reason:** illness, personal reasons
**Probability:** 2 medium (on scale 1-3)
**Seriousness:** 3 high (on scale 1-3)
**How to avoid:** defined schedule and days when the members are not available.
**How to prevent:** to have access to the data of the member (other one can replace him easily).
**How to survive:** re-distribute the workload between the rest of the member, giving incentives for doing it.

### 3.5.2 Risk Pe2: Long term absence

**Symptom, early warning sign:** one team member is missing for a lot of time.
**Source or reason:** illness, personal reasons.
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 3 high (on scale 1-3)
**How to avoid:** defined schedule and days when the members are not available.
**How to prevent:** to have access to the data of the member (other one can replace him easily).
**How to survive:** re-distribute the workload between the rest of the member, giving incentives for doing it.

### 3.5.3 Risk Pe3: Bad communication within the team

**Symptom, early warning sign:** the team is not synchronized in the work.
**Source or reason:** lack of communication, bad relationships.
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** define ways of communication within the team. Promote good relationships with different activities.
**How to prevent:** meeting of the team to update the information of everybody.

**How to survive:** meeting of the team to solve the problem.

### 3.5.4 Risk Pe4: Task overload

**Symptom, early warning sign:** the team is not accomplishing the deadlines.
**Source or reason:** bad distribution of the work
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 1 low (on scale 1-3)
**How to avoid:** necessary to know the capabilities of each member of the team.
**How to prevent:** redistribute the workload for that member.
**How to survive:** redistribute the excess of workload between the rest of the team.

### 3.5.5 Risk Pe5: Change of job

**Symptom, early warning sign:** one member of the team receive offers from different companies
**Source or reason:** bad conditions in the current job (or worse than the new offered)
**Probability:** 1 low (on scale 1-3)
**Seriousness:** 2 medium (on scale 1-3)
**How to avoid:** good treat to the employees with the best conditions as possible.
**How to prevent:** re-negotiation of the current conditions.
**How to survive:** replace the employee as soon as possible. If not, redistribute the workload between the rest of the team.