

Date: 2017-05-17 (active to 2017-05-19)

Chair: Ludvig Andersson

Participants: Rasmus Lindgren, Markus Pettersson, Anthony Kalcic, Ludvig Andersson

Meeting agenda

Group 10, Parallax

1. Objectives

- Update the priority list.
- Structure RAD
- Discuss the SDD
- Discuss the report.
- Dividing writing assignments.
- Tests and comments (JavaDocs)
- Discuss the game, making sure that we are all on the same page on how things are going to look and work.

2. Reports from the previous meeting

The group has worked together since the last week. Each group member has worked on an individual task from the list of priorities from last meeting. We have also worked together to find a solution to a big memory problem. The result was:

- Anthony has been working to find the reason for our memory leak. The program spikes in memory after a while of running, this happened right after a big change in how we handle rendering.
- Rasmus has been working on the game over screen.
- Ludvig has worked on the score screen, and with this implemented transparency. He has also made a time to score connection. A new royalty-free song has also been added to the program.
- Markus has made the walls now work like solid objects, making Agilion not able to go through them.

Date: 2017-05-17 (active to 2017-05-19)

Chair: Ludvig Andersson

Participants: Rasmus Lindgren, Markus Pettersson, Anthony Kalcic, Ludvig Andersson

3. Discussion items

The list needs to be updated due to a thing being removed, it also has to be updated based on the new goals we decide this meeting. We should structure a list that is updated as it is

We have discussed what the RAD has in it, now we need to structure what is left to be done. Then we can give the things out as assignments with a finish date on it.

The SDD needs lots of work, we need to go through this in stages, we can decide what we can do during the weekend. This will help us plan better next Monday. (2017-05-18).

The report content needs to be defined, we can then start writing on it. We should divide as much work as possible.

Will we use Java-Docs, and to what extent do we comment the code.

We have discussed pressing concerns in the project like the start and end screen, how do we implement these, are we thinking the same and so on.

Date: 2017-05-17 (active to 2017-05-19)
Chair: Ludvig Andersson
Participants: Rasmus Lindgren, Markus Pettersson, Anthony Kalcic, Ludvig Andersson

4. Outcomes and assignments

A new priorities list has been made, it looks like this:

1. Fix the issue with the ship getting stuck to the walls.
2. Fix the spikes that occur while playing.
3. A score function, based on time in the tunnel.
4. Death reset phase, make everything reset
5. Design of death screen
6. Make a start to the level, making it so that you don't just die right away.
7. A well-structured course with obstacles and walls placed at decided places.
8. Implement power-up boxes, giving them a 3D model
9. AI functionality, making them able to track and shoot the enemy at a good level
10. The sound of the game needs to be changed before launch, simple sounds also needs changing
11. Make the missile not go through the course.
12. More power-ups need to be implemented, the simple ones are up next, nuke, extra life and shield.
13. Implement particles, mostly for ship explosion
14. Post-processing effects

The priority list will stay updated here, phases will be documented:

<https://docs.google.com/document/d/1rLWEdmq-qMbeJM56RhFUjK8dNYiBwHbn5xnlRpGWSI/edit?usp=sharing>

We need to have updated these things in the RAD:

- Change the tense to present tense, for example changing it from: "we want to create..." "to we have created or..." , "Galactica is a..."
- Change the introduction, the priority has changed from VR.
- Change pictures, make it focus how it look, and how it should look.
- Update use-cases, make them follow the program structure.
- Updated the Functional requirements, make them follow the new use-cases
- Write "2.3 Non-functional requirements". Can focus on that the game's Usability, reliability, performance. These have been important in the implementation phase.
- Also update the last RAD, to have a type of "2.3 headers". Can simply say that this is nothing we have decided, at least make it have a finished look.
- New use case listing
- New domain model
- Update the class responsibilities, based on the domain model.
- Go through and polish other RAD's (v1 and v2), make sure that information is 100% correct and that is finished to the point that we can use it as one in a chain of updates.

Date: 2017-05-17 (active to 2017-05-19)
Chair: Ludvig Andersson
Participants: Rasmus Lindgren, Markus Pettersson, Anthony Kalcic, Ludvig Andersson

The SDD needs much work, there for we will focus on these points before next meeting.
(2017-05-18):

- Make the introduction describe development platforms, controllers and such.
- Update the list of word, this will include lots of words that we have used during the working phase.
- Write about the different software as much as possible. Focus on the things we have done for Adam to make the structure better. Provide screenshot proof from PMD and STAN.
- Write about vecMath.
- Sequence diagrams
- Refine and extend the text already written on the on point 4.
- Add text about LibGDX data management.
- Write about the reason for user only program, why we only need it and so on. We need to have the headline and write some text under it, even if it is not very important for us. Describe why it's not important for us.
- Remove abbreviations

How we structure the report:

Title: Clear title that summarizes the whole project

Abstract: Short presentation of what the game is, and a summarization of what you can find in the report, more specifically the result.

Introduction: This it what we meant to create.

Theory: This is divided into parts of things a reader needs to know to understand the content of the report. Theories include LibGDX, Google Cardboard, MVC and the processes used in a project.

Method: How we approach the creation of our game, what methods did we use and so on.

Result: Divided into parts, firstly a part that is connected with the method, how did we use the methods, were we successful in using the methods we were told to use and the ones we decided to use in the project?

Secondly, what did this result in, a description of the game and the functionality we created. Thirdly, what we could have done better, what held us back from doing better. Knowledge, time and so on.

Discussion: Why did the game become like this, what could we have done differently and what are the benefits of both what we did and what we could have done differently. End of with what are have learned and if we are happy with the result.

We will write on our own time during the weekend, (2017-05-19 to 2017-05-21) and the assignments given are written in point 5 "Wrap-up".

We will use Java-Docs as much as possible to structure our comments. We will also write a comment on methods related to each other what they are doing. Good method names are sufficient for most methods. If it's still unclear what the method does, then divide the content.

Date: 2017-05-17 (active to 2017-05-19)

Chair: Ludvig Andersson

Participants: Rasmus Lindgren, Markus Pettersson, Anthony Kalcic, Ludvig Andersson

Write clear descriptions if something is meant to be changeable, something that toggles the program, for example, static instance variables.

The creator/writer of a class will both write tests and comments on that class. This will take the least amount of time.

We will keep the screen implementation we currently have. Time does not allow a new implementation at this stage of the project.

Date: 2017-05-17 (active to 2017-05-19)
Chair: Ludvig Andersson
Participants: Rasmus Lindgren, Markus Pettersson, Anthony Kalcic, Ludvig Andersson

5. Wrap up

TL;DR of things that we need to have done and if they have a deadline:

Programming: We have structured the work in a list, this list consists of goals and comments on which group member that is working on it.

<https://docs.google.com/document/d/1rLWEdmq-qMbeJM56RhFUjK8dNYiBwHbn5xnlRpGWsl/edit?usp=sharing>

Work until next meeting:

- Anthony will: Write on the method in the report. Write about vecMath and systems in the SDD.
- Rasmus will: Write a theory text about MVC patter in the report. Update Use Cases. Make title.
- Markus will: Write a theory text about LibGDX in the report.
- Ludvig will: Write a theory text about programming process. Fix the old RAD, refining and finishing them. Write about LibGDX data management in the SDD. Update the word list in the SDD. Change tense in the newest version of the SDD.

Next meeting will be on Monday 08:00-12:00 (2017-05-22) in M1222A