Progress Report 5: 30 April 2012

Thomas Sievert, Martin Säll, Lars Woxberg, Kim Restad & Fredrik Johannesson

KIMportant REST ADditions

thsi08@student.bth.se, martin_svart@hotmail.com, lars.woxberg@gmail.com,
kim.restad@gmail.com, fredrik.johannesson@hotmail.com

Work done this week

This week we have worked on, and finished the following:

- L2 report
- Class diagrams
- Line of code counting script
- Redmine setup
- Debug camera class

We have worked on, but not finished the following:

- GameplayHandler class
- Ghost class
- Path finding
- AI class
- Scene class

The finished tasks were finished according to last week's plan, the Ghost class, path finding, AI class and Scene class were scheduled to be finished this week.

Major design decisions done:

A decision was made to not include Boost++ in the project, since it would require too much time for too little gain.

A decision was made to not make use of a BSP tree as a scene graph, since it is an optimization we don't yet know if we need.

A decision was made to abolish the GameObject abstraction in the Model package, since the player and enemy objects have too little in common. Furthermore, the project has no use of polymorphism in this regard.

A decision was made to extend the Observer pattern so that both IngameScreen and Scene implements the GameEventSubscriber interface, since they need the information.

A decision was made to cancel the Sprite batch component, since it is an optimization we most likely will not need. Our system draws a very limited amount of sprites.

A decision was made to use strategy pattern for camera controllers, since the different types of cameras have very clear, different behaviors.

The GameplayHandler class did not need an aggregation to the Level class, since it can get all the information needed from the LevelHandler.

The LevelHandler's interface was simplified to do less, since it previously could do more than the GameplayHandler needed it to.

A decision was made to cancel the inclusion of deferred rendering, since it would take too much time to research and implement.

Changes to the Work Breakdown Structure

The WBS went through a major overhaul as it was transferred to Redmine. It now more closely follows the ClassOverview diagram presented in the L2 report, since we are confident that those components are small enough to be considered tasks.

Since Redmine and L2 took more time than originally estimated, other tasks have had to be postponed. What should have been completed this week, but has been postponed to next week, includes the following tasks:

- Sprite class
- Menu class
- Table class
- Game screen class
- Pathfinding
- Enemy class
- Scene class

A decision was also made for us to start reporting our work in the span of Friday to Friday, starting with this report, since our goal is to avoid having to work in the weekend. On top of that, it's practically easier to gather everyone to put together a weekly report on Fridays than on Sundays.

Starting next week, we will also introduce testing of components whenever there is time over.

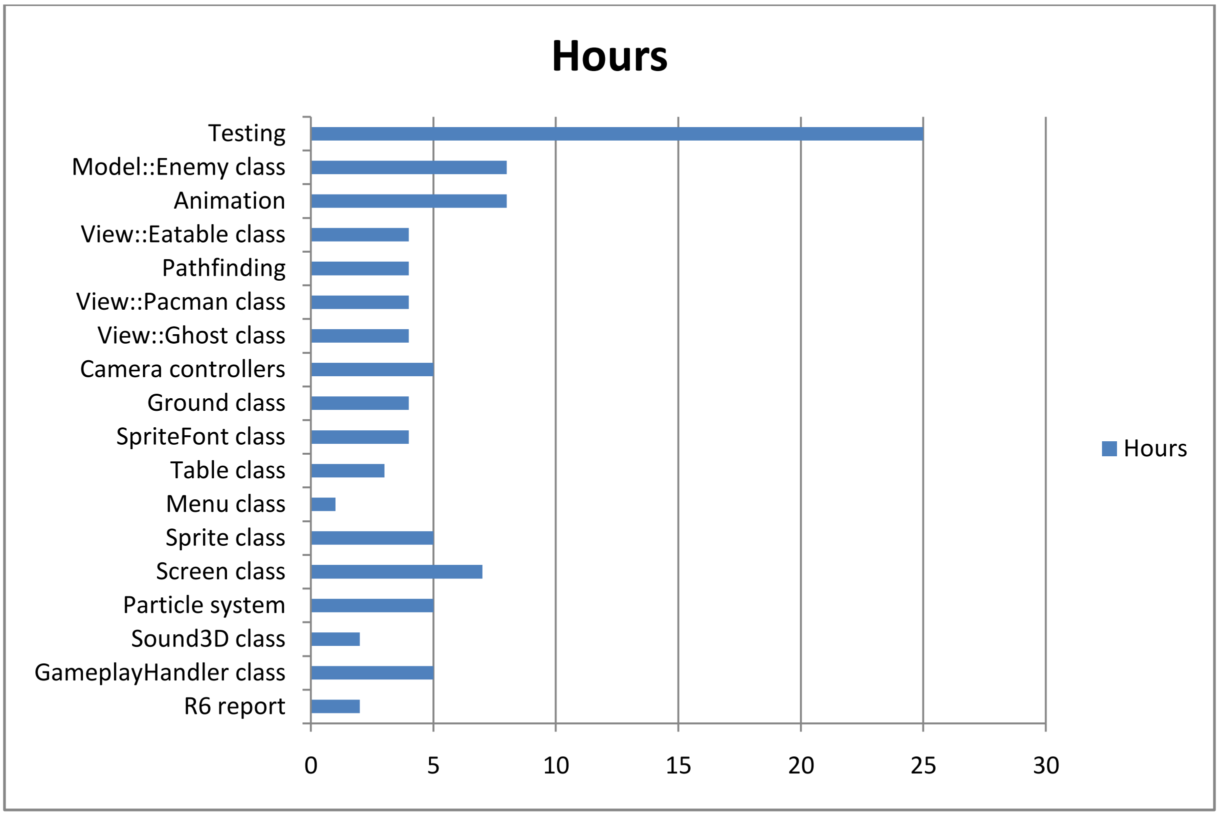
Issues, problems and risks

As always, estimation is a problem. In moving our project to Redmine and restructuring our WBS to closer follow our detailed design we have a better overview of the project than before, but time is still an issue. To counter this, we've cut down our ambitions as far as optimizations are concerned, and started focusing more on just getting a working prototype running. Hopefully these changes will bear fruit.

As for code metrics, we've spent some time putting together a small script counting lines of code (LOC) and estimating the source lines of code (SLOC). Again, thanks to Redmine and L2, we are also able to present a better overview of the project, in terms of task oveview.

Work planned

The work planned for the next week, along with estimated time the task will take, is shown below:



The person in charge of each task is:

- Thomas: *R6 report, Table class, Animation*
- Martin: *Particle system, Camera controllers*
- Lars*: Screen, Sprite, Menu, Sprite Font*
- Kim: *Sound3D, Ground, View::Ghost, View::Pacman, View::Eatable*
- Fredrik: *GameplayHandler class, Pathfinding*

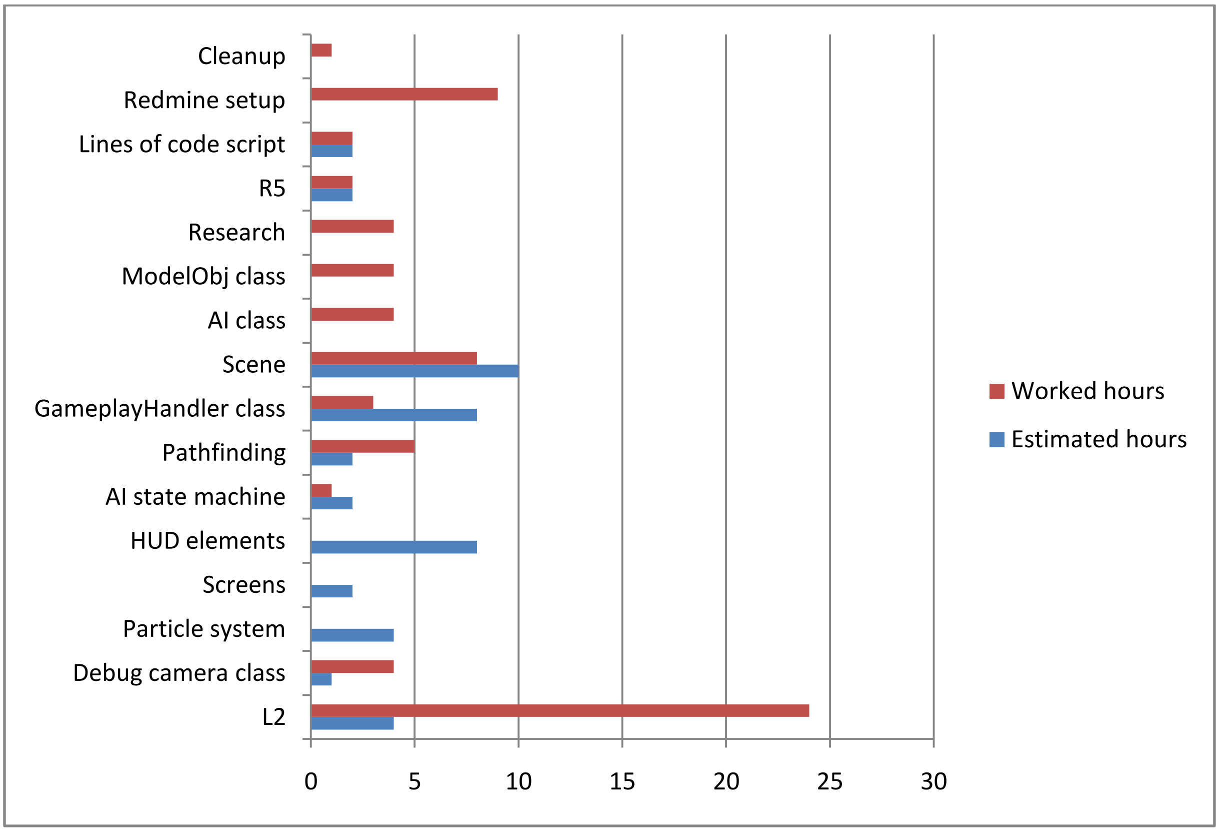
While everyone will still be able to work on any part, this puts someone in charge of every part and will allow us to work on our own as well.

Since we are in the process of moving the tasks of our project to Redmine and that our design and schedule has been given some new priorities, we do not have an actual Gantt chart to show this week.

Summary

Schedule compliance

Below is a graph of the tasks we have worked with the past week, along with the time we estimated and the time we worked. Unless stated below, the tasks are finished.



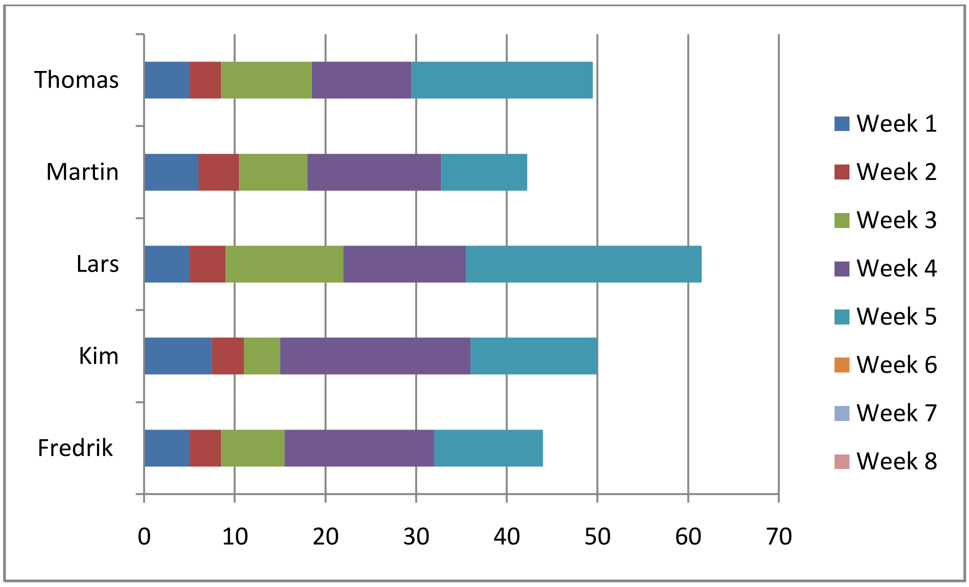
Unfinished tasks (planned over several weeks):

- Particle System
- GameplayHandler class
- AI class
- Scene
- Pathfinding
- HUD elements (this includes Sprites and GUI components such as Menu and Table classes)
- Screens

We are still a bit behind schedule, mostly due to poor planning. It was originally our intention to spread out the L2 work over the course of several weeks, but we discovered that we didn't have such a good overview of the design until this week. As such, it took up a lot of this week's work and other tasks were reprioritized, as can be seen by the difference in estimated/worked time.

Resources spent

The number of hours each person has spent on the project working either in a group or individually is shown below. The total number of hours is a little less than would be expected at this time in the project due to an uncertainty about what the project entailed along with poor time estimations. As we get more comfortable with what we need to do we spend more time on the project, which is evident in the graph.



Product metrics

Current LOC: 4303
Current SLOC: 2552

We currently have 25 unfinished tasks out of a total of 51. This measurement isn't wholly reliable, since new tasks can appear.