

Post Mortem Report

A reflection of the android application project

Group 3

10/26/2013

Table of Contents

1. Introduction.....	3
2. External elements.....	3
2.1 Libraries	3
2.2 Emulators and devices.....	4
3. Process.....	4
3.1 Week by week	4
3.1.1 Week 1.....	5
3.1.2 Week 2.....	5
3.1.3 Week 3.....	5
3.1.4 Week 4.....	6
3.1.5 Week 5.....	6
3.1.6 Week 6.....	7
3.1.7 Week 7.....	7
4. Reflection.....	7
4.1 Our process.....	7
4.2 Responsibilities	7
4.3 Working in the same room.....	8
4.4 Pair programming.....	8
4.5 Testing	8
4.6 LibGDX	9
4.7 Future aspects	9
5. Conclusion	9

1. Introduction

The application developed is called “One more cookie, please!” and is a single-player platform game. The main goal of the game is to get the character, which is portrayed as a squirrel, back home. However, this squirrel is absolutely addicted to cookies and wants to catch as many as he possibly can before getting home. The dilemma is that there’s a storm coming and the squirrel has to hurry to get home before the tornado catches him.

There are several obstacles placed along the way and if the character gets stuck behind one of them, the tornado will eventually catch up with him and the game is lost. To avoid the obstacles the player can make the character jump every time he or she touches the screen. You will also find several cookies placed along the route and if the character succeeds collecting them, the player gets additional points when the game is over.

To make the game harder or easier the player can choose how fast everything in the game is moving. This is done by tilting the phone to the left (everything moves slower) or to the right (everything moves faster). How fast the game moves impacts the total score when the game is over.

2. External elements

2.1 Libraries

LibGDX is a library that works well for developing games, mainly 2D, for a variation of platforms. Since the game was intended to run on android, and the fact that the library was made for android development, was one of the main reasons for choosing it. It was also well documented, which felt essential when creating an application in a completely new environment. Since knowing the game would contain physics, for jumping and colliding with objects as well as animated graphics for the character and the enemy, it was important to have all of that included in the library. The fact that libGDX contains everything needed for both physics and graphics was therefore a huge decision breaker.

2.2 Emulators and devices

An emulator is easily set up with the version of eclipse used for android development. This emulator is however, extremely slow. In the beginning of the project, when the only thing implemented was a jumping box, the emulator was sufficient for testing if the positions were correct and if buttons worked. After implementing the moving camera as well as adding more physical objects to the map, the emulator could no longer handle the game resulting in it crashing. Running the app on a phone is a more efficient way to test it, especially if it's possible to test the application on different devices. This app was mainly tested on two phones: The Sony Ericsson Xperia Arc, which is an older phone with a resolution of 854×480 , and the slightly newer HTC One M7 which has a higher resolution of 1920×1080 . Testing on different resolutions is important to determine if the app will look and work the same on different devices.

3. Process

During the project a somewhat modified version of scrum and agile was used which better suited the group. Documents on Google Drive were created to keep track of the process, which is one of the core elements of agile. Generally this includes post-it notes and a private whiteboard that unfortunately was inaccessible during the project.

Furthermore, it was difficult to approximate how much time a task would take therefore, in the sprint backlog, our user stories were organized after priority, represented with a number (1-5) that showcased a combination of time and difficulty. However, burndown charts, retrospective and velocity were not used, due to lack of time and priority of other matters, and our sprint reviews were quite informal. Since daily meetings were held, daily scrum didn't felt needed. Instead at the beginning of every work session, the group discussed the current workload and decided what to focus on next.

A scrum master was chosen at the start of the project but later on it was decided to try a rotating scrum-master so that everyone could get a feel for the position. However, as the project went further along the position vaguely faded and became more of a role that anyone could take up when they felt needed. Conclusively, the position was not that active during most part of the project.

3.1 Week by week

Weekly meetings and work:

Monday afternoon: 2 hours, Sprint-planning meeting.

Tuesday morning: 3 hours, work.

Wednesday morning: 2 hours, meeting with supervisor and technical instructor to solve issues.

Thursday afternoon: 4 hours, work.

Friday: all day, 5 hours work and 2 hours release.

Total: approximately 18-20 hours/week.

3.1.1 Week 1

After forming the group, a lunch meeting was held to spawn some ideas. A few vague concepts surfaced which later the same week would be defined further when having a more formal meeting. When elaborating the ideas, it was important to have in mind what would be best suited and, more importantly, the most doable in terms of this project. Since the magnitude of the application was somewhat uncertain, we came up with options with more complexity as well as concepts that with certainty would be doable. When the ideas were well thought through in terms of concept, risks etcetera, they were sent in and thereafter the response was awaited.

3.1.2 Week 2

On Tuesday a first meeting was held with the assigned supervisor. Conclusively, it was agreed that the best choice out of the submitted ideas was the sidescroller game, not only because it would be a thrilling game to make but also, and more significantly, it would always be easy to add or remove features depending on the time left on the project.

The next day there was another lunch meeting where a week schedule was set up which determined the workload for each week. Here it became clear that it would be difficult to get the entire group gathered more than a few hours a week since everyone had different courses and activities.

That Thursday a repository on Git was created and some research about how to create an app, what libraries to use and what kind of classes to create was made. During the week documents created from the first two weeks was added to the repository.

3.1.3 Week 3

The week started off with a sprint-planning meeting where three user stories were chosen to be implemented during the week. Furthermore, the library was added into the project. A stable

version was picked from the libGDX website, but later it turned out that the version chosen should have been the nightly build. The stable version did not include all the classes that was needed to render the maps.

This was the first week of coding and the library was making it challenging for the group. The tmx files from Tiled, the map editor used, couldn't be read into the project which was one of the most major issues that occurred during the week. Apart from that, there was, among other issues, some problems with the physics in libGDX. Here the technique of pair programming was to a great assistance. However, because of this, nothing was released at the end of the week since there was no actual code that worked.

3.1.4 Week 4

Because of the problems that occurred the previous week, it was decided to add the same user stories to the sprint backlog as before, not adding anything new. Finally, the library was switched from the stable version of libGDX to the nightly build which enabled rendering the maps. Now the inclusion of physical objects to the character allowed it to jump as the screen was touched.

Testing was also a main focus during the week. A JUnit test was created to test the converter from the physics world's unit meters and the device's unit pixels. The world is the where the game is actually taking place within the application.

For testing the physical components of the game debug rendering was used. Debug rendering draws graphically on the device and one can view how forces are used to give the objects physical states. The project was now making progress.

3.1.5 Week 5

The game, now with its basic functionality, started to evolve when adding several screens beyond the actual game view. A start screen as well as a level section was added to name a few. Obstacles in the world were also implemented, though there were problems with making the physical and graphical objects to represent the same position. During this period pair programming got abandoned as it only would have slowed down the development process.

3.1.6 Week 6

As the last week of adding functionality to the application, as many features as possible were implemented; the gyro-steering to increase or decrease the speed of the game, music and sound effects, cookies for the player to collect was successfully added to the game. Additionally, there was more functionality added to different screens, for example, being able to restart the level or move forward to the next. The obstacle bug from previous week was also fixed.

3.1.7 Week 7

During the last week, finally all of the graphical elements were implemented. The application now started to look like a finished product. However, there were several problems with the size due to the tiles only being displayed as a power of two. A lot of time was spent to resize and sometimes redo important elements. Also, the new background seemed to make the game crash because of its bigger file size than the previous test version. This resolved in changing the background, making it plain to lower the file size, in hope that the issues would dissolve.

Another main focus this week was to fix the last bugs and making sure that the game was working properly. To test the graphical user interaction and its components a manual test was performed.

4. Reflection

4.1 Our process

The usage of a modified scrum was not in retrospect considered to not have been as efficient as initially thought. By not having a non-rotating scrummaster it was hard to have effective meetings and to inform the whole group about general information. Priority lists worked very well and it facilitated the choices for which tasks to implement when facing time limitations.

4.2 Responsibilities

At the beginning of the project responsibilities were handed out to each group member such as “implementing/making graphics”, “knowing how the physics work in detail”, “making sure the code was neat and clean” and so on. But as the project moved along those responsibilities were somewhat neglected as specific, important tasks and problems appeared that needed the group’s attention. This might have been one of the reasons Javadoc and testing did not get the

attention it might have needed. Ultimately, only one group member truly focused on one responsibility, the visible graphics, since knowing it would be extremely time consuming and needed one's attention for it to be finished at the end of the project. If the given responsibilities were stuck with, it most definitely would have improved our working process. However, since this method of guiding the project was unfamiliar to the group, it was difficult to maintain during this short period of time.

4.3 Working in the same room

The majority of the time on this project was spent working together in the same room. This meant that one could easily ask one another for help when needed. Since the group were constantly updated about what everyone was working on it made daily meetings feel abundant. While working in this way it was also possible to easily switch pairs in pair programming, thus, always stay updated on the progress of other team members.

4.4 Pair programming

In our experience, it was easier to program in pairs when there was a problem that needed to be solved. It allowed you to discuss the problem and reach a solution together. This could otherwise be tiresome and stressful to work out on your own; therefore pair programming was more efficient for the bigger, more difficult tasks. For the smaller tasks, like fixing bugs and implementing easy features, one person was usually enough to solve the problems with the support of the rest of the group working in the same room.

4.5 Testing

The methods used for testing were very useful in terms of the aspects of the game, though considerably time consuming. The physics used in the game was a part of the library, making it hard to grasp and so debugging was used to better understand the process. A debug method was testing the physics by drawing shapes and forces active in the physical world which made it easier to see what was happening. Moreover, manual testing worked efficiently when testing if the connections in the application were redirected to the correct action. The JUnit test written to convert meters to pixels was also a great aid when examining if the physical and the graphical

objects positions matched.

4.6 LibGDX

The library chosen worked well and contained everything needed, though however, did cause some problems at the beginning of the project. A libGDX project consists of a core project - which contains all the source code - and then the platform specific projects, in our case, an android project. The challenge here was to get the two projects to communicate both ways to, for example, get access to specific android functionality in our core project. Solving this was quite a challenge and took a lot of work but eventually was figured out.

4.7 Future aspects

If a similar application would be built in the near future, the process of starting up the project would be considerably easier if the same library was to be used. This project have given enough knowledge about android development that all major issues at the start of this project would most presumably not occur and that time would therefore be spent differently. For example, to make the process of implementing the core of the application easier, more time could be spent on pre-production - planning and structuring up the project before building it.

Working together would be a favorable approach to reinstate since it facilitates problem solving when issues occur, meaning, being able to switch to pair-programming when needed.

Nevertheless, there will always be unimaginable problem emerging, thus, to make another application of this kind would certainly be challenging. The major advantage in building another android application would be the experience gathered from this one.

5. Conclusion

The final result of the application is a properly functional game with slightly less functionality than first imagined. For example, there is only one character available instead of several to choose from, there are no power-ups to be collected and there are no enemies you can face along the route in the game.

It is believed that the main reason for the group not having the time to implement these features was due to the problems that occurred early on, for instance the library and tmx files issues previously mentioned at the start of the project. These problems slowed down the process tremendously and vital time was spent fixing the issues instead of moving forward with the development.

Regarding the testing, everything that could be tested is tested. However, as formerly cited, the library made things challenging and if not so, more testing would have been in favored. A lot of manual testing was performed, and though not optimal, it was the best way possible to inspect if methods worked properly. Moreover, the library chosen for the project was well documented and easy to understand thus met our expectations. Problems that occurred during this time is believed to be as a result of the group's inexperience. Conclusively, the library is an excellent choice when working with android development, although testing methods is problematic.

Visually the game looks very pleasing. Though questioning if all that time spent was worth the outcome, our anticipations in making a modern android application were fulfilled. Concerning the working process, as early quoted, the version of scrum which was used had both its pros and cons. To follow scrum more accurately would have been in favor, which unfortunately was discovered afterwards.

In conclusion, the group is very satisfied with the outcome of the project even though it was not alike the initial expectation.