Android Game Development

Final Year Project Plan

Full unit project
2017

Author: Patryk Pawlicki
Supervisor: Peter Robinson

Abstract:

The aim of my final year project is to develop a simple game for Android platform using the Android Software Development Kit which will cope with the variations of hardware configurations and be compatible with different devices and factors such as battery saving will be taken into consideration along with other constraints present when developing phone applications. Mobile applications as well as games are a very big market which is constantly growing with a massive potential for monetization, android as well as IOS are the leading systems for mobile games. It's relatively easy to publish apps on the Android Market making Android App development accessible to anyone interested wishing to dedicate their time. There are a lot of resources available on the web with a big community of android developers. I have always been interested in game development, especially for Android systems. My motivation behind choosing this particular project was mainly due to the fact I would like to get involved with Android Game Development to potentially pursue my future career in app development after graduating from university.

I am an Android phone user myself, who has experience with apps and games available on the Android Market, therefore I understand the industry from the user side and have a bit of insight on what users want and expect from applications they download. Seeing as the market is very densely saturated with app developers, there are a lot of free applications available for download, therefore if an app or game does not satisfy the user they will simply uninstall and install another one. It's very easy to download and install apps and users rarely get attached to using a single application.
I would like to try and get into the realm of Android app development myself and contribute to the community of developers and android game users, hopefully this project will be my gateway.

Timeline:

Milestone:	Description:	Date:
Set up and structure Git Repository for the project	Set up a repository in order to store and update all the documentation and programs for the project	29/09/17
Install and set up Android Studio & Emulator	Install and set up Android Studio and emulator on personal laptop to be able to start working on the proof of concept programs	01/10/17
Proof of concept programs - Hello World Android program.	Complete the Hello World android program and commit to repository	03/10/17
Proof of concept programs - Explosion animation program	Complete the explosion animation program that does not Force Close (Starve user interface android thread) and commit to repository	10/10/17
Research Android User Interfaces further	Conduct further research on Android User Interfaces in order to aid with writing of the report	12/10/17
Proof of concept programs - XML based user interface.	Create an XML based User Interface for Android and commit to repository	15/10/17
Research and brush up on previous knowledge of design patterns	Conduct further research on design patterns as well as brush up the previous knowledge gained in the software engineering course from last year	17/10/17
Report - Design Patterns	Complete and commit a report on design patterns to the repository	20/10/17
Research Android life cycle further	Conduct further research on Android life cycle in order to aid with writing of the report	22/10/17

Reports - Android life cycle	Complete a report on Android Life Cycle and commit to repository	27/10/17
Research animation techniques	Conduct further research on animation techniques to aid with writing the report	29/10/17
Reports - Animation techniques.	Complete a report on Animation techniques and commit to repository	03/11/17
Research worker threads within a GUI	Conduct further research on worker threads within a GUI to aid writing the report	05/11/17
Reports - Worker Threads within a GUI.	Complete a report on Worker Threads and commit to repository	10/11/17
Research GUI design further	Conduct additional research and possibly experiment with different GUI creation techniques to aid the report	13/10/17
Reports -Basic Design of a GUI based mobile game.	Complete a report on Basic Design of a GUI based mobile game and commit to repository	17/11/17
Reports -XML for GUIs.	Complete a report on XML for GUIs and commit to repository	24/11/17
Interim Programs and Reports Deadline	Deadline for all deliverables for the first term, reports as well as proof of concept programs	01/12/17
Presentation Interim Review	Prepare a 10 minute presentation for the Interim Review held 4-8th Dec.	03/12/17
Splash screen for the game	Complete a splash screen for the game	10/01/18
Functional GUI	Have a functional Graphical User Interface	15/01/18
Report - a description of other Android, its history and its internals	Complete the section of the report which covers Android history and its internals	26/01/18
Report - Threads, Workers, Battery Life and other special considerations for mobile applications	Complete a report which covers threads, workers, battery life and other special considerations for mobile applications	09/02/18
Draft Report Ready	Have the full unit draft ready and emailed to supervisor	16/02/18
Changes implemented based on advisor feedback	Implement all the changed suggested by the supervisor	30/03/18
Fully functional game	Have a fully functional finished game which can be played without any major issues	12/03/18
Report - interesting programming techniques and data structures used (or able to be used) on the project	Complete a report which covers interesting programming techniques and data structures used (or able to be used) on the project	19/03/18
Release game on Android Market	Have the game uploaded and available for download from the android market	20/03/18
Final Report and Programs submitted	Complete and submit all of the programs and reports for the final year project	23/03/18

Bibliography:

Beginning Android Games 2nd Edition - Mario Zechner / Robert Green
Covers android game development in a lot of depth, talks about different kinds of games and their characteristics. Found it useful when learning about graphical user interfaces and choosing the appropriate design theme for the game. Also gives some vital insight on techniques used when developing android, different types of I/O, audio as well as game mechanics.

Android Game Programming for Dummies – Derek James
I have found this book very useful due to how clearly and unambiguously it's written, although it mainly focuses on programming in Eclipse with the use of ADT plug-in, it covers the basics of Android programming very well as a whole. It talks about the Android Gaming Industry and what I've found most interesting and useful, how to get the game on the android market and capitalise is through different methods such as making it paid, in game adds as wells as in app purchases.
Gives a good basis for android game development, includes a lot of useful diagrams and illustrations along with tutorials on how to create basic android games and user interfaces which will greatly benefit me with my project.

Official android website, particularly the developer section covering Android Studio
(https://developer.android.com/studio/intro/index.html)
Very useful, covers the basics of installing and setting up Android Studio for my project. Talks about basic tools and shortcuts needed to efficiently use the software, as well provides useful information about the user interface of the application.

Risk Assessment:

It is essential to assess the risks associated with undertaking any project. The project that I will be undertaking spans over a large timescale, largest than I have ever dealt with before, it is important that it is thoroughly planned and that potential risks are taken into consideration since problems are very likely to occur in a project of this magnitude.

One of the main risks associated with projects is not keeping up to the planned timescale. It is very important to take into consideration any problems which could potentially result in the project not meeting the previously planned deadlines.
The risk of not completing tasks on time could be mitigated by allowing extra time for work to be completed; if a problem is encountered we have extra time to resolve it so that the task is still completed on time. There are a lot of factors which can contribute to the tasks running over the timescale; this includes technical problems, illness or perhaps poor and overly optimistic planning.

Failure to complete responsibility is also a risk with a project of this magnitude. There could be various reasons for the work not being completed; laziness and lack of motivation or even lack of time, due to other responsibilities. This can be mitigated by thorough planning and allowing extra time for things to be done.

Running over the timescale is one of the most common issues associated with large projects.

Another major risk associated with a project is not meeting the previously specified requirements. It is important that requirements are well understood and taken into consideration when working on the project. The risk of not meeting the required requirements can easily be mitigated by incorporating the requirements in the plan as milestones as well as communicating with the stakeholders regularly during the project to make sure that they are satisfied with the progress and functionality implemented.

Compatibility is also a very important aspect to consider when working with any IT project. Programming is no different, where we have to make sure the software we are using is compatible, same goes for the hardware or devices we are writing programs for. It is crucial to make sure we don't end up in a situation where some aspects that we have touched upon in our project fail to integrate and are not compatible since that could deem our work worthless. Having to rewrite software can of course result in going over the timescale, but it could even result in the whole project failing if it's not discovered soon enough.
It is likely that more than one type of software will be used for the project so it is important that the software is compatible with each other but also that the hardware that will be using the software being written can run it efficiently. In order to minimise the risk it is essential to do further research to check for compatibility as well as checking the version of the software being used.

When working on anything there is always a risk of our work going missing, fortunately with IT this can be easily mitigated through having backups of our work. The best way to mitigate the risk of our work going missing is by using a repository such as Git or SVN, we can make frequent backups of our work so even if our computer was to go missing, all of the files will be available on our repository. Speaking of repositories, they allow us to go back to previously backed up version of our code, this allows us to pretty much fully eliminate another risk, which would involve us making some changes that ended up breaking our software that we wish to revert, we can easily do that with a repository.