

Major Projects Report

by

Matt Adair
s3204584

Semester 1, 2013
(weeks 01 - 15)

Executive Summary:

This report covers the 15 weeks of semester 1, 2013 spent doing a Major Project. The project proposed was to develop an Android OS game application that would serve as a demonstration of a touch screen based control concept of the way a hand held device would translate touch screen inputs into game object movements. To develop this application it was decided that a purpose built game engine should be coded as opposed to utilising an off-the-shelf library.

The deliverable for this project is an Android packaged .apk file that can be installed onto compatible devices and used to test the game play, mechanics and logic. Initially it was thought that a documented testing period would be included within this project but this proved to be unhelpful due to the feedback received during the early stages of development. The project name is:

Flick Fighter, a demonstration release candidate, ver 6.3

Introduction:

This project was chosen because it would satisfy several requirements that were present; the project had to be of suitable scale to meet University requirements, was within the field of my chosen degree with a major in Computer Science and Programming, was interesting enough to ensure motivation over the fifteen weeks of development and finally, had achievable and definable goals. It is also the first time an attempt has been made by me to develop a computer game.

This project would also allow a rough simulation of a workplace in that the software I chose to develop would have to be researched, planned, designed, implemented, tested and finally released in a way which I assume would be commensurate to a professional small software development house. In keeping with this idea it was discovered that some unforeseen events can also effect software development.

Description:

The Android game I wished to develop was based on a pen and paper game played in childhood. Translating this idea into a touch screen interface game was a challenge in itself and formed the bulk of development time. The game is a turn-based game where the player has to move a fighter object around a maze of walls and shoot any enemy objects that may be present.

For movement the player uses a flick motion to propel the fighter forward as well as propel their laser bolt during the attacking phase. This flick motion was initially thought to be the key to this project in that the aim was to achieve a game play mechanic that felt appropriate and pleasing to the player inasmuch as those words can be utilised for such a description.

It turned out that providing an interface that instructed the player how to play the game and also have the game responses feel authentic, proved to be where a good proportion of development time was spent. Another area which required more time than was initially anticipated was in the development of the logic and coding for the game engine itself.

A game engine is quite a complex area of programming in itself, beyond the scope of this report, suffice to say that the engine for Flick Fighter would need to fulfil the following requirements: single player, turn by turn, several distinct phases within each turn, multiplayer networking capabilities, low CPU usage and playable on a low powered device. The networking aspect could not be developed within the time allowed for this project, however all other major requirements were met.

Context:

This project was considered a possibility for development only because of previous experience gained in this degree program during the Computer Science courses taken. These courses covered all aspects of application development from design, implementation and finally to testing. Several of the Design based courses also provided a useful learning experience when it came to project management and interface design.

Project management is one key area that has been improved by the experience gained from this degree program. Creating a realistic production schedule and adhering to it requires an understanding of the goals involved throughout the project as well as an honesty in assessing how achievable they are within the time allowed. From managing others to managing production, these areas are vital if an idea is to be realised in a form that is useful.

The software was developed using the Eclipse IDE that was introduced in some of the courses taken. These courses also covered the planning and structure of an application utilising the standards compliance adopted in industry. Further to this, the unique specifics of developing for the Android OS were not unknown in that two applications were developed and released previously as part of extra-curriculum activities:

Elevator 17 - <https://play.google.com/store/apps/details?id=com.cityfreqs.elevator17>

IMS5 - <https://play.google.com/store/apps/details?id=com.cityfreqs.ims5>

During the planning phase of the project it was envisioned that it would culminate in a documented testing phase. During earlier builds of the application the demo game was given to several people for testing which revealed that not enough time was spent on the graphical elements and gameplay mechanic feedback. The objects on the screen needed to convey a sense of meaning as to their purpose and how they should interact

with other elements in the game.

Ultimately what this meant is that the underlying logic of the game engine needed to be obfuscated graphically into a playable world that would be fun to use and would make sense to the player. This is the major unforeseen event that took over development time and stopped the planned documented testing phase from taking place as it was felt that meaningful responses would not be generated from the demo in its current form.

Conclusion:

This project saw a change in the way it was thought that a game could be developed for testing. Even though the project followed the Rapid Application Development methodology there were some aspects that required more consideration, the main one being the GUI and the meaning that could be derived from objects on-screen as to their purpose. The other major unforeseen drain on time was the implementation of a suitable collision mechanism.

The initial timeline was planned with several implementation phases to take place that would allow either the game engine to increase in function or for an off-the-shelf library to be used and integrated into the game. Instead these phases were used to refine and improve the core engine and game object classes in an attempt to get the demo into a testable release state.

Object collision is one of the fundamental aspects of a game that has objects interacting with each other. How this collision is detected and then what occurs to the collided objects turned out to be an area that required more time than planned within the initial project timeline. The complexity and fine tuning of these interactions form a vital part in providing the feel for a game and whether it makes sense in the context of the game itself. Without an effective and believable collision mechanism the game would always feel awkward for a player.

In summary, the fifteen weeks were spent mainly coding the engine and object classes in a way that would allow further development on this application either as part of future university activities or as a side project. The experience gained in the complexities of managing and executing a software development cycle proved invaluable in helping to determine possible future paths in the software industry.

Code Lines : 4604

Graphic bitmaps : 38

Audio ogg s: 10

FF_demo6.apk : 579 KB

Appendix:

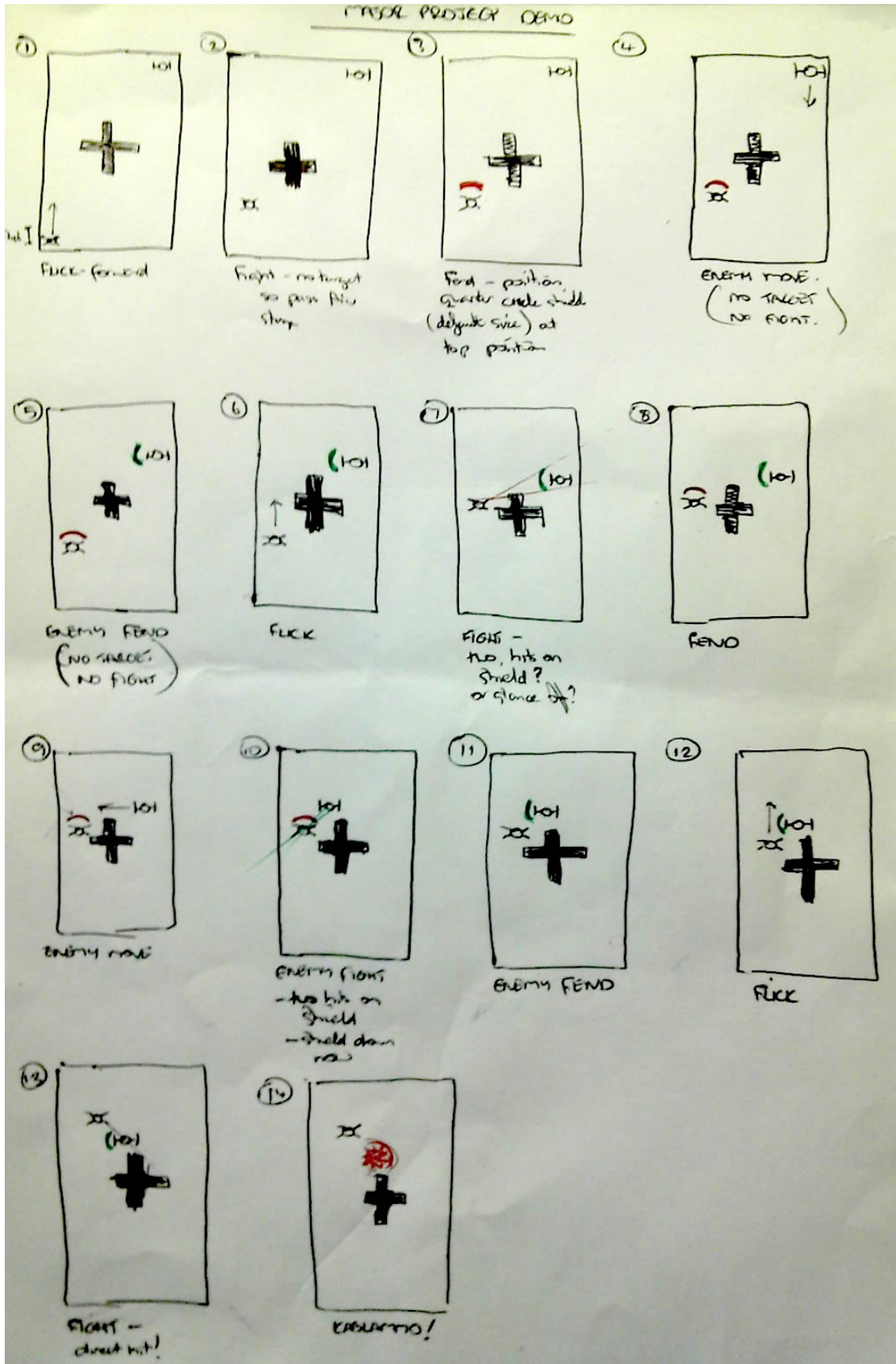
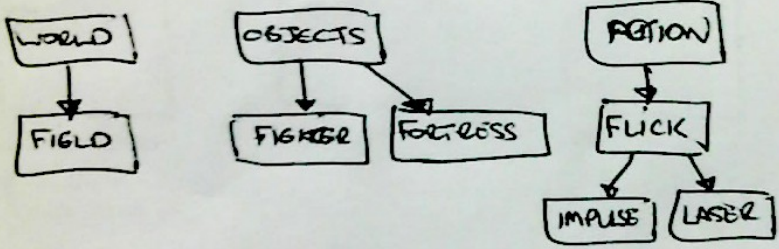
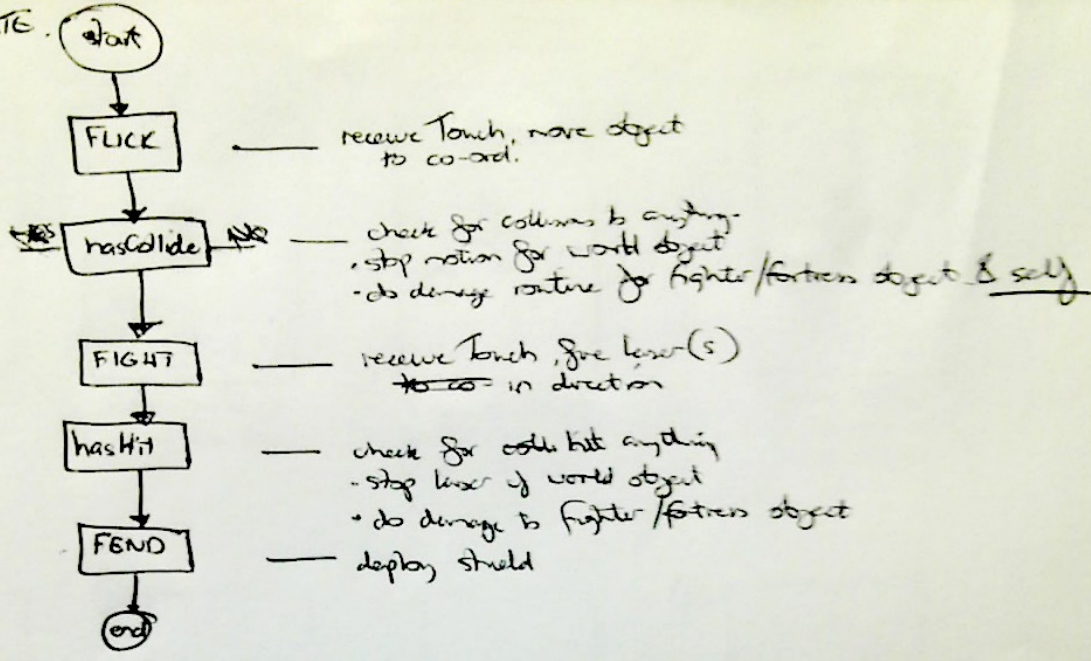


Image: initial game demo proposal, screen concepts.

GAMESTATE for WORLD



MINIMUM

EXTRA

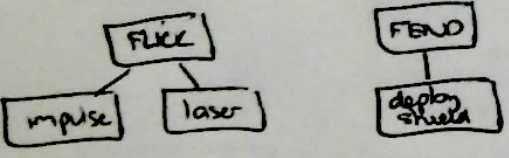
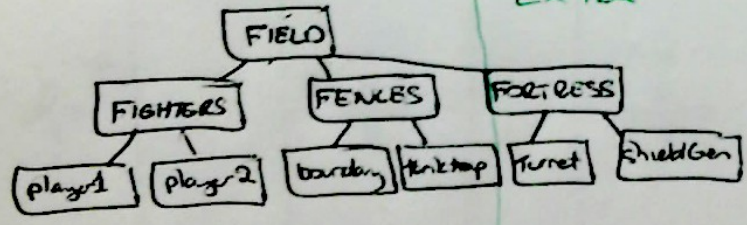


Image: initial proposal for game state with possible extension objects.

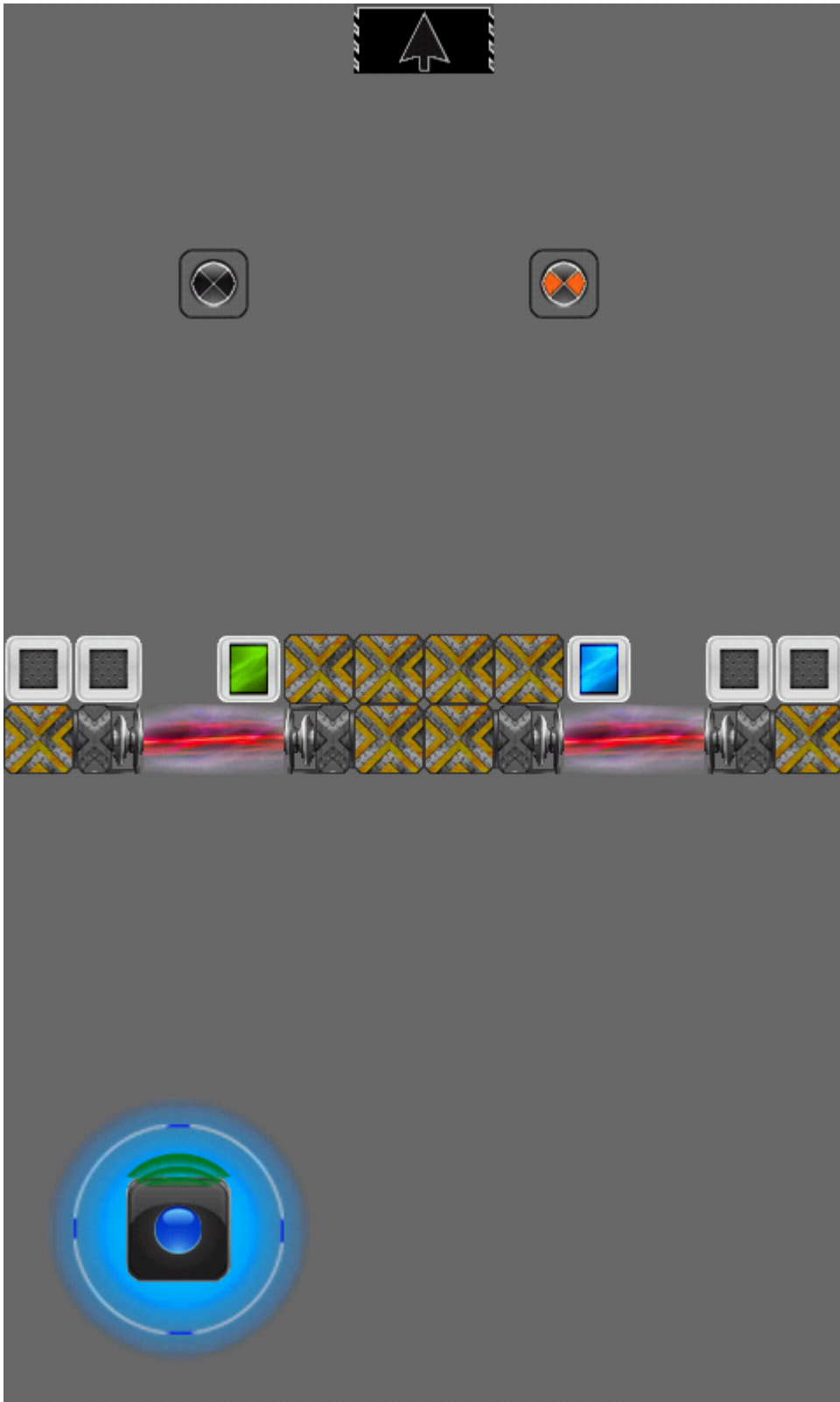


Image: FF_demo 6.3 screenshot showing most game objects and player touch feedback.