Army Builder App

Lewis Jamieson
40273123@live.napier.ac.uk
Edinburgh Napier University - Mobile Applications Development (SET08114)

1 Introduction

The Army Builder app is to be a complementary tool to be used with Games Workshop's tabletop war-game Warhammer 40,000. This app helps users with preparation of an army for a game of Warhammer 40,000. The scope is that the app should be able to store user's army list. It should be able to save it to the system where the user would be able to load it. It is to remind the user what the units' stats are during a game, so that they would not need to bring out the codex for their army each time they forget a unit's stats. It would also be able to connect to Dropbox and save it there as a single file. The idea of an army list was chosen as usually users would make an an army list using a CSV editor or by hand. The app will fix that problem saving users' time. The app was specifically designed with tabletops in mind.

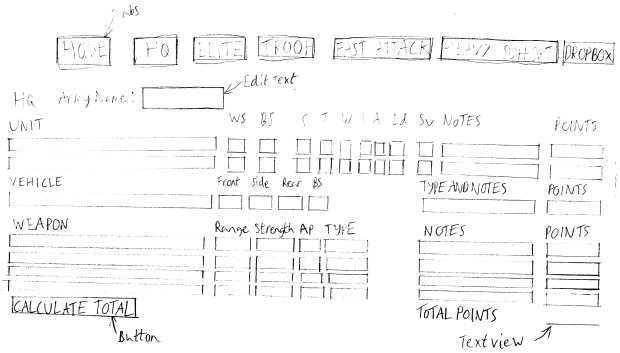

Figure 2: HQ Tab Sketch

2 Software design

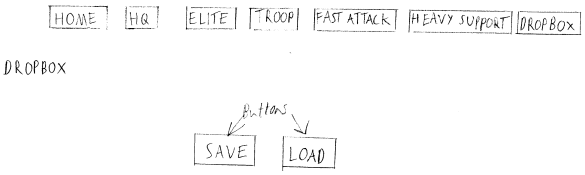

Figure 3: Dropbox Tab Sketch

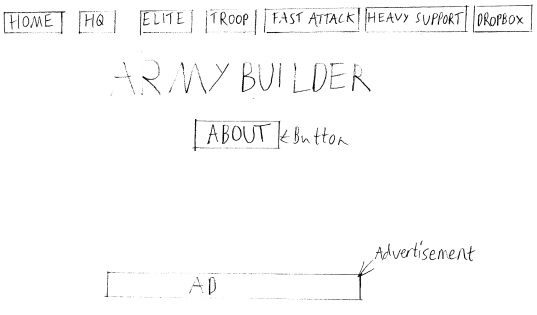

Figure 1: Home Tab Sketch

3 Implementation

The app contains two activities, home, HQ, Elite, Troop, Fast Attack and Heavy Support tabs, apart from home each of these tabs are named after the unit type which can be played in the game. The app icon I made myself with "paint.net".

Home tab (Figure 4) consists of an advertisement [1], an image background [2] and a button which opens a second activity (Figure 5). This activity shows helpful information on how to use the save, load and calculate total buttons and a key for the units, vehicles, walkers and weapons stats. Within the tabs each contain space for units, vehicles and weapons with their respective stats, notes and points cost.The point total for each tab can be calculated using the "CALCULATE TOTAL" button in each tab, also each tab can be individually saved using the "SAVE" and "LOAD" buttons. HQ tab (Figure 6)has space for the name of a player's army, Elite (Figure 7) and Heavy Support have space for Walkers and their stats and point cost. The app uses "WRITE EXTERNAL STORAGE", "INTERNET" and "ACCESS NETWORK STATE" permissions.

3.1 Comparison

The prototype product is similar to the initial concept as it facilitates for the each of the unit types and is able to save the

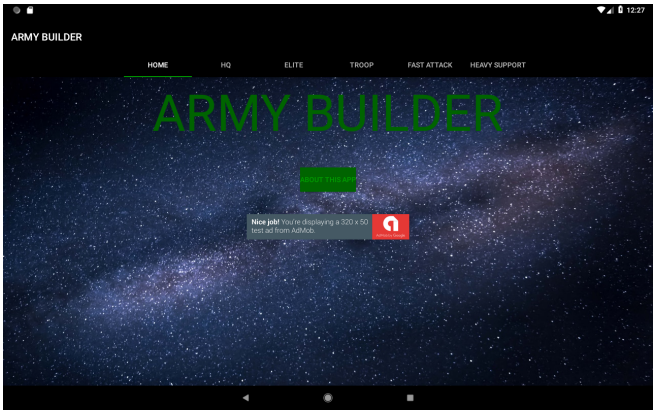

Figure 4: Home Tab

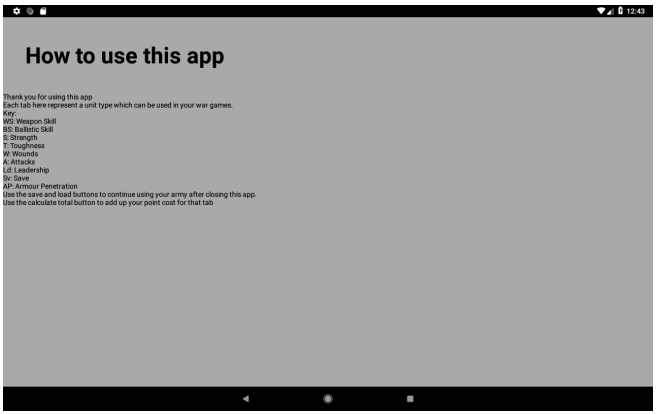

Figure 5: About Activity

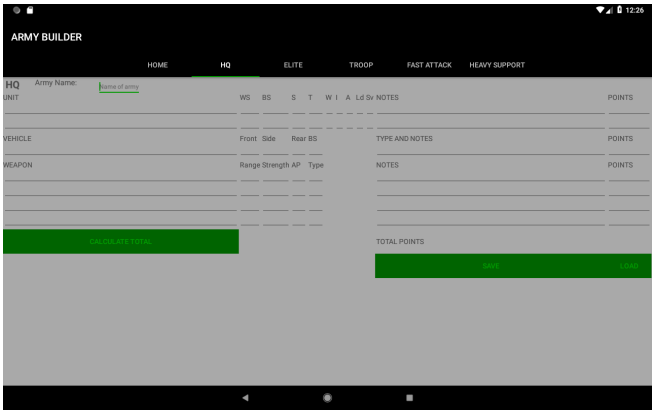

Figure 6: HQ Tab

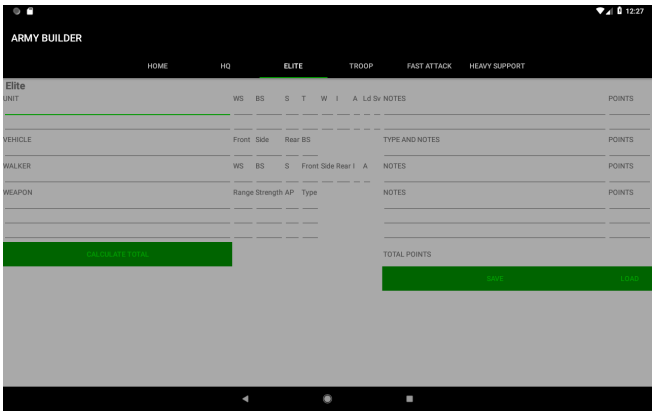

Figure 7: Elite Tab

army list. The Dropbox function (Figure 3) was not implemented in prototype version due to poor time management, if I had planned things out better I would not have had this problem. In the prototype version the file save function was separated to be for each unit type. This was due to not understanding how to transfer data between fragments, but I think it worked out better as the way I set the file data to the edit texts would have been more tedious if I had saved all of the data on one file. There is also an app called BattleScribe which also helps users make an army list. It is able to be configured for multiple tabletop miniature games, where as my app is only for Warhammer 40,000. This app has every possible official unit stored in each configuration file, so users can quickly make an army list. In my app the army creation is slower as each unit, vehicle, weapon and stats have to been manually typed in, but this is an advantage as user could input custom units, vehicles and weapons into my app, where the app is fixed to official units. Users who use BattleScribe would have to wait for the configuration file have to be updated for when Games Workshop makes a new unit for the game, my program does not have that problem as the units are put in manually. BattleScribe has the ability to choose a flexible amount of units, unlike in my app where the user is stuck with a fix amount of units. The app also can have multiple army saved at the same time, my app is limited to one army at a time. It has the function of sharing army lists via Dropbox. This function I was originally going to implement, but BattleScribe uses it to allow compatibility with the Windows, Mac, Linux and iOS versions of the app. BattleScribe has a yearly subscription system and blocks functions behind the pay wall. These functions include removing advertisements, saving units as favourite units, a dice roll probability calculator, quickly viewing the full army list, sharing army lists using NFC technology and customizing unit names and notes. My app doesn't have a paying system and my app does not block the ability to customize unit names and make notes. Both apps have advertisements, but BattleScribe has the ability to disable them. BattleScribe is the better product with its greater number of features, but I think my app is better in the terms of customizing unit details.

3.2 Possible Improvements

Implementing the ability to using Dropbox would be greatly benefited, as it would allow users to download the files in a situation where the user has lost their device or have had to format it. Having the save and load buttons be in a single place would increase user's convenience and save time from selecting all of the buttons in each tab. The ability to customize the number of units would be good as the user would be able to have a larger number of possible armies. Letting the user save multiple army lists and letting them compare the two would allow the user to quickly see which army would be better in a specific game. Having a points total of all of the unit types, would mean so that the user does not need to calculate it themselves and having the calculation being done without the user pressing the button would save time, as the user would have to do less.

4 Personal Evaluation

During this coursework, I have learned how to use fragments and implement advertisements. Fragments are useful for if I want to an app with a lot of object which need to be shown as separating them into tabs make the app run faster than if all objects were in one activity. Advertisements are useful if I want to create an app which I would monetize. The major mistake I made when making this app was time management. In the future I would use a burn down chart, to show myself how much work needs to be do in a certain time frame. When creating the second activity, I had came across a problem where I could not get it to work. The problem was that I did not know that creating a new intent of the second class in a fragment was different from declaring a new intent from an activity. I manage to figure that out and got the second activity to work. I personally feel that I performed well with this task considering how close to the concept it is. It is a bit disappointing that I never got to implement the Dropbox feature, but I feel that I have learned from my mistakes and I will work on my time management.

References

[1] Firebase, "http://firebase.google.com/docs/admob/android/quick-start.html."

[2] Pixelbay, "http://www.pexels.com/photo/photo-of-galaxy-207529.html (cc0 licence)."