Literature review

Charlie Ansell

K00203841

To begin discussing about a library to help developers create companion apps we must first understand what a companion app in terms of video games is, Brandon Landis author of [link 1] talks about exactly what a companion app is, he says that companion apps are applications which “ consume all backend services as if they were the main app upon which their backend is based” this is very true and in my opinion gives a very good overview of what a companion app is and in view of my final project being a library that can help developers create companion apps by helping them collect and transform their player metric data into readable, tangible forms of data.

After reading the short description of what companion apps are from link 1 I believe that anyone can look at my project and get a feel for what I am actually going to be building, all things centre around an application and this application will consume backend services as if they were the main app, in my projects case, the app will consume all of the backend data generated from my library which includes all of the necessary player data from the game that they were playing.

Has something like this been done before? Short answer is no, not in the way that I am proposing, there are applications out there that provide analytics in different ways to different types of users, for example, [link 2] discusses a type of software that in ways has a similar idea to mine but has a different target audience. This is the unity’s answer to “providing developers the tools to analyse player behaviour in their games” with their invention of unity’s data analytics engine, this engine plugs directly into a unity game and from their while a user plays your game the analytics software tracks statistics like player movement , what way the player went, how did they react movement wise to certain scenarios, player deaths, where the players have died, how frequently have they died in a certain level, daily active users, heatmaps. These are just a bunch of things that the unity game analytics software does for you, this information is crucial yes, but it is crucial for developers to use, not the players, with this information developers can create a better gaming experiences for the users using this knowledge of what whey they travelled in a game or where they have died in a level, the user does not need to see this style of information themselves because to them it does not mean a whole lot, players need to see metrics gained from their own gaming experience that can show them how they have performed, how they rank among the best, give them something to actually work towards, which is exactly what [link 3] discusses.

Ben Meddler author of [link 3] discusses how the game data world is quite “fertile” and that data itself is the “new soil”. This statement is very true as there is a lot of different types of data out there to be mined and analysed. Ben goes on to talk about player dossiers which are data driven visual reports comprised of player gameplay data like the achievements that they have earned, the rarity of that achievement. These player dossiers are very nice for users to view and very easy to understand, this is something that I will be looking to incorporate in a way into my project once all the players data has been analysed.

Just how popular are companion apps though, why choose to develop something that might not even be strong in the market, do big games sue these companion applications? Yes.

[Link 4 & 5] looks at 2 of the biggest games of this year which both have decided to incorporate companion apps into them, the first game being red dead redemption 2 which was so big of a game it had sold over 750 million dollars in the first weekend of its launch, so only in 3 days. The developers of this game, Rockstar, decided that because of the games vastness and richly detailed world they will include a companion app to allow the users to further interact with the world around them by using the companion app to view the map around them and be able to read documents that they find in the game all in the comfort of their bed by using their mobile phone. The red dead redemption 2 app on the google play store has over 500,000 total downloads.

This is also true for call of duty black ops 4 which introduced their very own companion app just after release of the game, using the app for this game allows players to create and or manage squads with other players from their phone and be able to manage weapon loadouts and see how friends are stacking up against you. Looking at the companion app on the google play store alone it has amassed over 100,000 total downloads so many players are enjoying the benefit of having a second screen to aid them in their gaming experience.

These figures are proof to me that companion apps are being used in the gaming industry and that they are very popular with users of these games. This means to me that it will be worth-while developing a tool that could assist developers whom are developing in Unity to create something as useful as a companion app, statistics speak for themselves.

Now that we know the usefulness of companion apps and software that could rival the idea of my project I have done some research into the technologies that I will need to utilise to be able to have my idea become a reality.

The first step in being able to build this library is through using [link 6], this link discusses how to build a class library in C# that can be used in unity, the way that they do it is to create a new C# project and make sure that it is set to class library, then we can add a reference to the unityEngine.dll by right clicking on references and browsing to the location of this dll. Then we right click on the properties tab on the project we have created and select unity 3.5 .net subset class base libraries. now we build this and grab the dll file produced and slot it into the assets folder of whatever unity project you want to use it with, then we can reference the dll whenever we wish to use it in our scripts.

Looking at ways then to connect unity with my app and or online database I researched ways that unity allows us to send data over the network and I found [link 7] which talks about unity’s WWW form object. This object is like a form in HTML where it has identifiers and values and we can populate this form with such and be able to send this to a URL of our choosing. I believe this will be the basis of sending the data from Unity to the database and allow us to manipulate the data from unity.

To be able to send a request to a URL we must first have 2 things, firstly, we need some file to be at the end of the URL to accept the transferred form and process it, but, perhaps more importantly than that we need a location that we can house that file. Through researching online I have found this online server hosting facility called 000 webhost [link 8] this allows us to host a lot of files with a pre-existing built in file manager service that they provide, they also provide a database hosting service as well so using this service we can create a cloud database and host files on this cloud server so summarizing we now using this technology will have means of sending a form to an URL and having a file waiting on the other end to process that and do whatever we want with it.

The bottom line after this is that we can now send to a URL, but we need a file to be awaiting the forms request and to process it at that URL, the file can’t just be any file. It must be a file that encases a server-side language powerful enough to process multiple get and post actions. [link9] shows us how we can use PHP as the server-side scripting language to manage these posts and get requests and even connect to our cloud database through its arrange of database connectivity methods. This if used correctly could be the glue that binds all the project together, there will be a steep learning curve for me to use PHP correctly, but I am sure once I get the hang of it then it will all turn out okay.

For the Application I have made my mind up from previous experience to use android studio to make the application for the project, not only am I selecting this IDE based on experience with it I am also selecting it because of how simple it is to use compared to others out there like XCode which is one of many IDE’s that can be used to create applications for apple products. Also, the main reason I want to use android studio for this project is because android studio uses java as its main programming language interface and feel as trough through the years that I have become proficient in my java coding and feel more comfortable using that as a basis for the app than swift or objective C like what Is used for apples design.

With the apps framework chosen I researched what would be the best way to interact with the cloud database would be, obviously abstraction would be a must because if all of the code to connect to a database that stores user information is all in one class then if someone hacked into the phone they would be able to easily get to and manipulate database data or even delete it all completely. From experience again, there are two major HTTP request libraries in Android Studio which can be utilised to send HTTP requests to a URL to get a certain file to process something for the client and echo a response back to them. [link 10] is a very good article that looks at the differences between Volley and Retrofit, both HTTP request libraries available to the public to use within Android Studio

The article, written by Ali Muzaffar, goes on to talk about which is better and gives the pros and cons for each, Ali talks about how Volley suits his style of programming as it gives more freedom to write your own code on top of what the library gives you loads of little freebies like image handling to boot straight from the library itself. “Retrofit on the other hand is more about code abstraction on top of a HTTP client “

Through using these links I have a good idea of how I can go about doing this project and I also feel as though now to me my project is easier for myself to understand as well as others who read this review.

**Links**

1 - <https://intercom.help/aquro/cloud-and-backend-services/miscellaneous/what-are-companion-apps-and-how-do-they-work>

2 - <https://unity3d.com/learn/tutorials/topics/analytics/introduction-unity-analytics>

3 - <http://gamestudies.org/1101/articles/medler>

4 - https://www.rockstargames.com/newswire/article/60648/The-Red-Dead-Redemption-2-Official-Companion-App

5-http://www.pushsquare.com/news/2018/11/call\_of\_duty\_black\_ops\_4\_companion\_app\_released\_for\_ios\_and\_android\_devices

6 - <https://www.tristancartledge.com/setting-up-and-building-a-unity-class-library/>

7 - <https://docs.unity3d.com/Manual/UnityWebRequest-SendingForm.html>

8 - <https://www.000webhost.com/>

9 - https://www.formget.com/php-post-get/

10 - <https://medium.com/@ali.muzaffar/is-retrofit-faster-than-volley-the-answer-may-surprise-you-4379bc589d7c>