

Senior Project Final Paper

Fall 2016

Reddit Gallery Android App

Student: Ryan Cirese

Advisor: Dr. Andy Poe

## **Introduction**

After heading home for summer vacation in May earlier this year I realized I had my senior project coming up next semester. I thought a lot about what I wanted to do as my project a lot during that first month of break, but motivation quickly faded away with the weather getting better and I all but forgot about it until the end of August when I was moving back to Marquette to begin my last semester at Northern.

I played around with a few ideas and eventually settled on making a Gameboy Emulator in C++ and SDL. I started the project and it was going really well, it could open a rom file and identify what it was, but I just didn't feel motivated to move forward with the project. The project just wasn't all that fun to me so I decided to switch. Recently I had worked on a few small android apps, which stemmed from interest after taking the Android Programming class here at Northern, the most notable being one that would display NASA's Astronomy Picture of the Day along with it's information that I had a lot of fun with. So I decided to make an android app, but what app I had a hard time coming up with. After a few days of brainstorming I was browsing the web for ideas on one of my most favorite sites, reddit.com, and it hit me that I should make a reddit app. So I decided to make an gallery app for a website that I visit daily.

## **Overview**

I wanted to make an Android Reddit app that was like a gallery for the image based content on the website allowing fast loading of the image only content in a simple layout. To do this I used Android Studio as it's basically the default IDE android app

development and I was familiar with it already. To hook up with reddit I used their API to send and receive data to and from reddit.

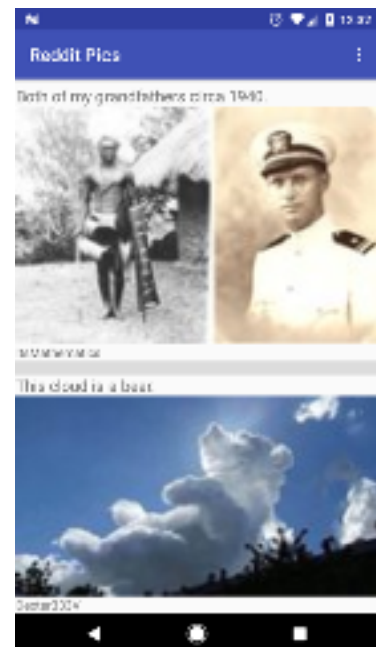
I also used a number of third part libraries in the app to handle to handle certain tasks like making network calls and image loading. I chose to OkHttp3 and Retrofit from Square to handle most network features such as making calls to and from Reddit's API. I used Glide by Bumptech to load image/gif image files and handle caching of the images. These three were the ones used most prominently alongside all of androids own SDK and support libraries. These benefitted my application by making it run fast and with a minimal memory footprint.

## **Planning and Building**

Before I started actually to begin to work on the app I needed to understand Reddit's api and how it worked. I read through most of their documentation, write-ups on some websites, as well as browsing some source code of open sourced reddit clients such as Red Reader to get a better idea of how the api flowed. After I felt I had decent understanding of Reddit's api, I moved on to begin forming the foundation of the app.

The first thing that needed to be done was decide how the main launch screen would look to the user. I decided at first to use a simple list view and load the image, title, and other information in a row item and have it display information like that. An early version of that looked like ->

At first I was happy with the way this functioned but after



adding more and more features I realized that because I wanted to make this image focused and content driven this layout was too empty or slow to deliver content to the user. If the app is image based, that is all the user cares about so that's all that they should be delivered on the surface. Because of this, I decided to change the main screen layout to utilize a RecyclerView, part of Android's support library, to show the content. Using a RecyclerView allowed me to be able to display images in rows and across multiple columns. This layout felt more like viewing a gallery compared to the more simple ListView and this allowed more content to be viewed at once, making it a lot more functional (Figure 1). This eventually evolved into being able to allow the user to choose the amount of columns between two, three, or four based on their viewing preferences. With a RecyclerView, this was simple to do as it has a LayoutManager object that can be manipulated at any time allowing easy changing of the view layout.

Although the app is image focused, not all images on reddit are basic png or jpg files. Reddit also supports gif images and I wanted my app to be able to support these. The image library I'm using, glide handles simple gif files and can load them into the same view type as a standard png or jpg image. There existed a problem however with commonly used sites like imgur and gfycaat whom typically don't use the basic gif filetype, but a video file format such as webm or mp4 for short video clips. To be able to display these I had to use android's VideoView type paired with a MediaPlayer to stream in the video file. After accomplishing full gif compatibility with gif and video types, I realized there was a problem in usability by not being to quickly differentiate between standard static images and gif images. To solve this issue I added a gif tag to the layout item that dynamically changes between light and dark color based on the dominant

color of the upper right corner of the image it is being overlaid onto. This does a pretty good job of informing the user which row items are which data type and overall greatly improved functionality of the app(Figure 2).

After implementing image and gif support in the gallery layout, the app's basic functionality was already completed, but there was still work to be done. Features like allowing the user to change the column count, sort reddit posts, and navigate with familiar android items like a navigation drawer were added in to make usability much better (Figure 3). Examples of these changes are below.

Figure 1

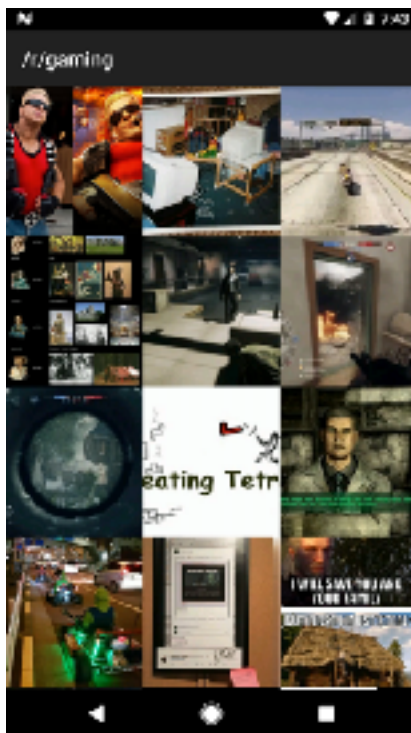


Figure 2

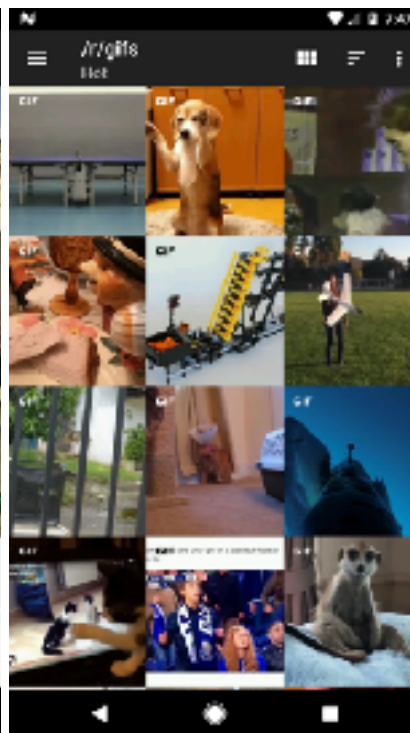
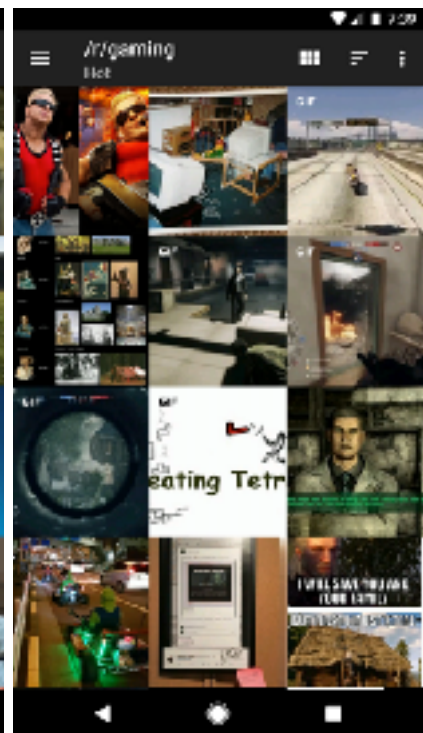


Figure 3



## **Goal**

After the features above were added, I decided that my end goal moving forward would be to make the app as if I were planning to release it. My work cycles became shorter. Planning and implementing a small feature, building and trying it out, debug, polish, start again with the next feature was how I operated.

I created a list of features and one by one I just checked them off the list doing the steps above for each. This really helped me move the project forward faster and more efficiently than I was before as well keeping me motivated through the duration of the project.

## **Problems**

After achieving basic functionality in the app I returned to reading the Reddit api so I could work on features I outlined in the project proposal such as saving a list of subreddits and voting on a post. It was then that I discovered that all Reddit clients making use of Reddit's api are required to use oauth to make valid calls to the api. This was by far the most difficult part of the project for me. I was not at all familiar with what oauth even was, so understanding this alone took up a large chunk of my time. The problem wasn't understanding oauth and how it worked necessarily, but getting it working properly with Retrofit (my networking library) and Reddit's api was.

This was first time working with a restful api as well as retrofit so figuring out alongside the Reddit api oauth requirements how to make calls to the right endpoint with the right headers / post data and how to actually send that call took me a very long time to get working properly. On top of that reddit has two different types of access tokens a user can have. One token is gained when a user is logged in to reddit and the other is a more basic and restricted token that is obtained for use when there is no user logged in but the client still needs to make requests. This further complicated things on top of my shaky understanding, but eventually I worked it out and got a small, functional system put in place that handles everything in the background away from the user.

## **Organization**

The project is organized across a few simple packages that house similar classes. I have activities that handle the main view, logging in, and viewing images or gifs. I separated classes that handle call to and from the reddit and gfycat and their respective model classes into another package. Any adapters needed for lists or other complex views are delegated to their own package, and finally a package to hold miscellaneous utility classes I deemed necessary. This setup worked well for me in that it clearly and effectively laid out what parts of the whole system did what and kept parts of the system that didn't need to know of each other away from each other. I feel this setup was the right choice and it allowed to browse to the right sections of the code I needed to get to quickly as well as add in new things without disrupting functioning parts at all.

## **What I Do Differently**

Starting the project, I had the vision in my head of using a list layout mentioned earlier, but not long after moving forward with that I realized that this wasn't what I wanted. If I could start again this would be the one of the things I would change about the project. Starting with the RecyclerView over the ListView straight away would be better because a RecyclerView can be made to function identically to a ListView and is much more versatile overall. I did learn a lot about how android uses these complex lists with adapters by first implementing a ListView that made implementing a RecyclerView very easy, but it took up a lot of my time only to later go in a different direction from it.

Another thing I would change if starting over is using Glide to load images right from the start instead of Picasso (Another image loading library), which is what I originally used. Picasso doesn't have functionality for gif images and I spent a long time trying to figure how to load gifs while using Picasso until I discovered Glide. Luckily after finding Glide, switching to it from Picasso was very simple. This taught me to research libraries more thoroughly before just implementing the first one that looks promising only to have to go back and change it later.

Finally, the last thing I would change is a general thing and it's that I would do more research before jumping right into writing code for the project. I found that I spent a lot of time backtracking a redoing areas of the project that I could have avoided if I would have researched a little more before starting. In the end I'm happy this happened though as in most all places that I needed to go back and change, I learned much better ways of doing certain things and I feel that it benefitted me as a programmer in general to make these mistakes and learn from them.



## **What I Learned**

During the course of constructing this app, I learned many things that I feel are important and that I could use in my career moving on from Northern. The most important in my eyes is that I learned how to organize and manage a project of this scope. This project is much bigger than any other that I had previously worked on for a class in my time as an undergraduate at Northern. This taught me how to organize my code in a way that made sense, and allowed the most code to be reused as possible without having any one part of the project dependent on another to function. Most of the code I wrote for this project could easily be adapted to work with another with minimal changes, and I feel that this is very important.

A few other things I learned that I think will benefit me greatly is gaining an understanding of how to use REST api and what that is to begin with. Before starting this I had heard the term and had a basic idea of what that was but had never directly used one. Becoming more familiar with android, its SDK and how certain patterns in its design come together. Finally I became a lot better at simply reading documentation for libraries and apis. Before now I never really had to do much of this and I feel that it will benefit me just being more comfortable combing through code / documentation and make me a more efficient programmer.

## **Closing Thoughts**

Overall the project was very fun and satisfying to work and I learned a great amount along the way that will help me moving forward in life. As far as project's difficulty is concerned it was about as difficult as I expected it to be but there were a few

areas that proved to be harder than I thought. At the end of the day the difficulties made it more fun to work on because solving a difficult part felt so great and drove me to push through other problems that arose.

Finally I want to thank everyone at Northern that helped me during my time here and I wish everyone the very best!