As the project has progressed, we have made several new developments, and encountered new issues. Now that we have resolved the team member issues for the most part, the team has made good headway on the project. Phil got the working environment all set up. In doing this, we were able to create our own branches and begin work on the different parts of the project. To start, Gabby and I created a UI. It has basic functionality and can take the user to the different scenes of the project. We still need to scale it to fit on a phone screen however, as right now it runs mainly on a PC resolution.

On my part, I was able to successfully integrate Firebase authentication into the app. The nice thing about Firebase is that it uses Google’s framework. This makes it very easy to set up the different authentication methods, and to view the user’s accounts and disable/delete them. It was admittedly a little tedious to get it all set up inside of Unity though. There was a lot of Google and Android SDK stuff to install. You also had to register your app with Google in order to user Firebase. Then I had to find scripts that would help me get it all set up and connected. The scripts needed to integrate into UI elements, i.e. sign-up/sign-in buttons. However, once I got all that set up, it works great. I’m able to create accounts and sign in/out of them. The way the app works now is that it first takes you to a loading screen. This screen runs the authentication checker script. It will first ensure that the device has the correct dependencies for Firebase, it will then check that it can connect to Firebase, and finally it will check whether the user is signed in or out. If the user is signed out, it will take them directly to the sign in page. There the user can sign into their existing account or create a new one. If the user is already signed in when they open the app, they will automatically be taken to the main menu. That is where the authentication is now.

I plan to expand this using Firestore. This is a part of Firebase that will allow for cloud storage of game data. I plan to use this to store the user’s progress of the planets they’ve discovered. I’m hoping that we will also be able to store the Astrodex information in the database, that way we don’t have a bunch of text sitting in the UI. We would just be able to pull it from Firebase, and all this data doesn’t just sit in the game files. But that’s something I’ll be looking into this week.

The major I’ve run into is that my Unity SDK are all messed up. I was trying to build the Android APK, and I kept getting a bunch of errors about dependencies. I installed a bunch of Android build support files and SDKs, but it won’t build. I’ve spent hours troubleshooting, and what it’s coming down to is I’ll have to just reinstall Unity. I’ve wasted about 25GB of storage on my PC just trying to fix it. Other than that everything seems to be okay on my end.

Sign In/Up UI + Firebase authentication code snippet

