

Report Date: 18/11/2022

To: ematson@purdue.edu, ahsmith@purdue.edu, lee3450@purdue.edu

From: Team Coyote(Sensors & Network)

- Hyemin Lim (freemini2@cau.ac.kr)
- Nayoun Kim (202010766@live.wsu.ac.kr)
- Jaehui Boo (32192075@dankook.ac.kr)
- Hyeongjun Kim (aa980305@cu.ac.kr)

Summary

The python code that changed the data types(datetime to float)[1] of information received and received packets and localization codes were almost completed and the two codes were combined. The hyperbolic curve is used in the localization code, but due to the characteristics of the coordinate values, it was nearly impossible to calculate the intersection point of three hyperbola curves. Therefore the method of getting the intersection point of two hyperbolas and getting the general area of the coyote was suggested. And using the sequence of time each sensors receives a signal, finding general direction of coyote was done before that to increase accuracy.

Due to conversion issue of coordinate information, the resource used to load the map on the Unity platform was changed from Google API to Mapbox SDK. To do this, some research was done to find right version and configuration of Unity. And the errors occurred during the installation of SDK was fixed. After that, the code for changing the view of the map by touching the screen was revised. Also camera bounding code and client side communication code for REST API with Node.js server was written again for the new SDK.[2][3]

With the data received, the singleton pattern code was written to share variable data with Mapbox code. Loading scene was made to solve memory leak related issue, Mapbox code for it is now in progress.

What Coyote Team completed this week:

- Write python code about receive the packets and change data type
- Fix Localization code
- Make one more module (esp32 + RPI = sound3)
- combine python codes in one file
- Set up node js server for transmitting coordinate information to Unity
- Due to the accuracy problem of latitude and longitude, the Unity API to be used was changed from google to mapbox.
- Unity environment setup for mapbox sdk installation
- Wrote Unity camera view with android touch code
- Wrote rest-api, node JS data sending/receiving code in Unity
- Wrote singleton pattern code and connect it with mapbox code in Unity
- Created a loading scene in Unity
- Modified mapbox code to fit the received data in Unity

Things to do by next week

- Localization code experiment
- put the ML model in Raspberry Pi
- Addressing memory leaks in Unity code
- Set up database to store coyote mapping coordinates

Problems or challenges:

- Localization accuracy
- Memory leak in Unity code (A Native Collection has not been disposed, resulting in a memory leak.)

References

- [1] Python. “datetime” python software foundation. Accessed: Nov, 16, 2022 [online]. Available: <https://docs.python.org/3/library/datetime.html>
- [2] mapbox. “Maps SDK for Unity” mapbox docs Guides. Accessed: Nov, 14, 2022 [online]. Available: <https://docs.mapbox.com/unity/maps/guides/>
- [3] Mapbox. “3Build a Unity 3D game Introduction to PocketDroids GO and the Mapbox SDK for UnityCamera, touch code fix” Youtube. Jun, 14, 2018 [online]. Available: <https://www.youtube.com/watch?v=RhG1kfDBhgM>