Report Date: 09/12/2022
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

Since there were some differences between reality and theory, plenty of experiments were conducted. As the previous weekly report said, Δt was the main factor that had to be reduced. Therefore, the experiments were done on Δt. First, there was a point that caused the delay in microphone code. The record method using sound device was worked only per 0.01 seconds.[3] However, since the unit of time that localization is using is microseconds, the record method was revised to be done per 0.001 seconds. Second, Δt was so big because of the microphones' status and performance. It brought about consequences of drawing the ellipse instead of hyperbola. Therefore, Δt was divided into 10. After that, the experiment worked well. Therefore, the contradiction between reality and theory had been solved as much as possible. In the condition, lot's of experiments are done. In the Unity section, this week's work created a list view so that users can see the complete list of coyotes displayed on the map. Animator and animation were created according to the UI, and the function and management code of UIs such as buttons, panels, and list bar was written. When Coyote was selected in the list view, working on a code that automatically moved the camera to the location value of the selected coyote, and created a function that creates a mapping pin at the detected location based on the data received in real-time. As the user touches, scrolls, and pinches the mobile device, the Unity part has completed the code generation and test to move the camera so that the user can check the map where the user wants it. Whenever a coyote is detected in real-time, the camera is automatically moved to the map position. Finally, created all the features the work list had been aiming for, so the unity part completed setting up and bake the environmental lights within the 3D map.

Finally, the whole system experiment was conducted. The experiment followed the direction below.

1) Extracting three timestamp when each three microphones first caught the peak point of sound.
2) Sending the timestamps via LoRaWan.
3) Using MQTT network, the timestamp input values are allocated to the localization code. [4]
4) Sending the results of localization code via websocket.
5) Visualizing the results in Unity Engine.

What Coyote Team completed this week:

- [Unity] Created a Coyote list view, organizing UI-related and animating
- [Unity] wrote code about the map and the camera automatically moves to the location value after selecting Coyote in the list view.
- [Unity] Develop real-time mapping of coyote coordinates
- [Unity] Create and test code that moves the camera according to touch, scroll, and pinch
- [Unity] Move the camera to the mapped location each time the coyote is detected in real-time
- [Unity] Organized unnecessary files and code
- [Unity] Map Environment Lighting Settings and Lighting Bake
- Found the influencing factors that cause the delay of time extract of microphones
- Revised the localization code to apply to the real world environment in visualization system
- Done experiment of the connection of the whole system, from microphones to Unity app.
- Prepared for the final presentation

- Wrote the paper

Things to do by next week

- Write the paper
- Prepare for the final presentation

Problems or challenges

- [Unity] Memory leak occurring in random situations
- There were some differences between reality and theory because of the microphones' status and performance.

References

[1] Jayanam. "Unity UI Panel Animation Tutorial" Youtube. Sep, 09, 2018 [online]. Available: https://www.youtube.com/watch?v=mz9xfDQ4FCk

[2] Unity3D. "Create a list view runtime UI" Unity Documentation. Accessed: Dec 05, 2022 [online]. Available: https://docs.unity3d.com/Manual/UIE-HowTo-CreateRuntimeUI.html

[3] python-sounddevice. "Play and Record Sound with Python". Python-Sounddevice Documentation. Accessed: Dec 05, 2022 [online]. Available: https://python-sounddevice.readthedocs.io/en/0.3.7/

[4] paho-mqtt 1.6.1. "Project description". Python Package Index (PyPI). Accessed: Oct 21, 2021 [online]. Available: https://pypi.org/project/paho-mqtt/#option-functions