

Progress Report

Name: **Oliver Greenwood**
Email: **og314@cam.ac.uk**
Project Title: **Whisperwise: Emergent Communication in Multi-Agent Reinforcement Learning for Complex Tasks**
Supervisor: **Ian Lewis**
DoS: **Ramsey Faragher**

My project aims to evaluate the use of emergent communication. This is done through developing a simulation with multiple agents and comparing a baseline of continuous communication against emergent communication inspired by the paper AI Mother Tongue. The project is on schedule. I have written 2800 lines of code (notably composed of 1500 lines for the simulation and 1000 lines for the MARL controller) and completed four of my five success criteria (developed a simulation, implemented a MARL system, added both communication protocols). My final success criteria involves a streamlined evaluation of my communication protocols. I plan to complete this by 15th February.

Initially, one of my success criteria was to implement a VDN-based MARL system. VDN works on a purely cooperative global reward; however, my agents each had a clear and often unique goal, making individual rewards for every agent necessary. MAPPO allows for this by utilising a centralised critic with decentralised actors, stabilising the learning process whilst preserving individual agent policies.

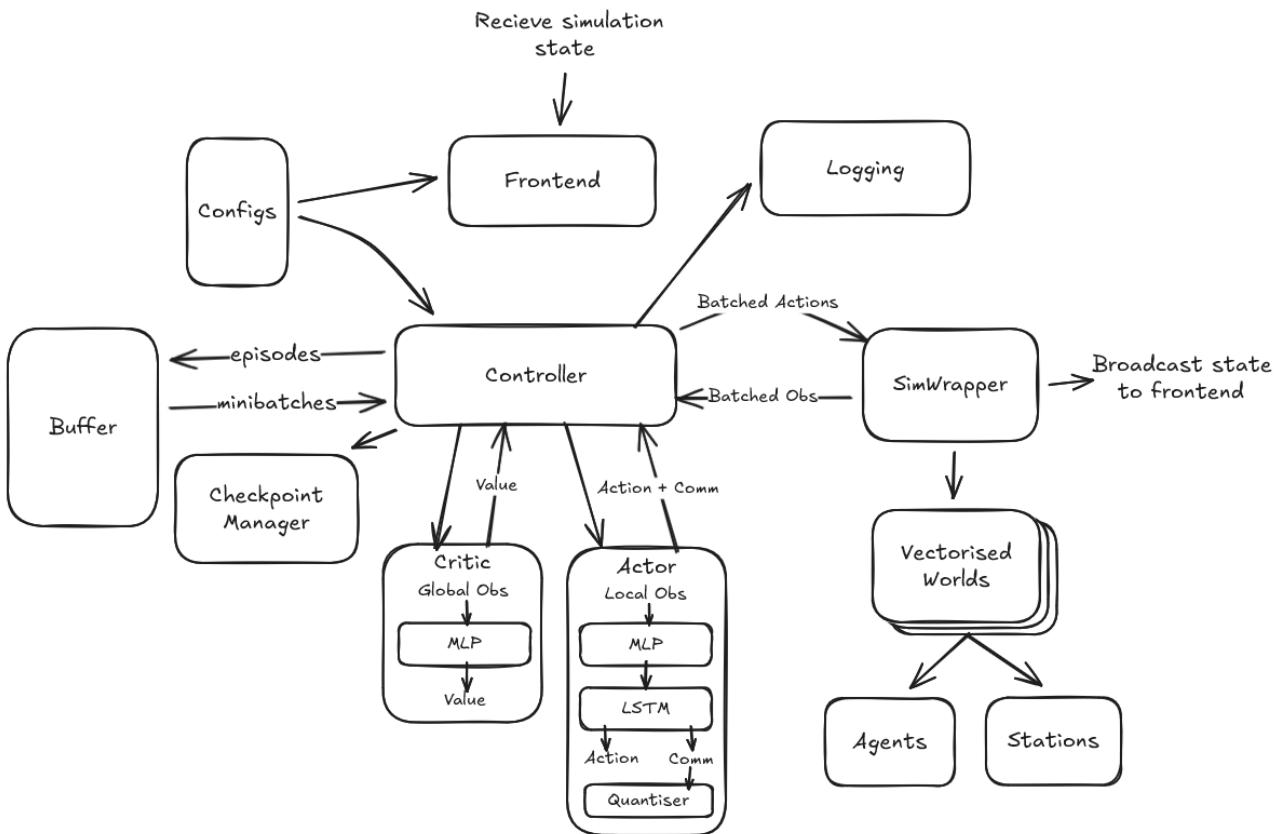


Figure 1: System Diagram

My program consists of multiple logical sections, detailed in the diagram above.

The program is decoupled into a frontend for visualisation and a backend controller. The SimWrapper manages a set of vectorised worlds and broadcasts each world's state to the frontend via a websocket. Each world involves agents with the ability to move in the four cardinal directions and interact with their nearest agent/station. The agents can pick up/drop off specified ingredients to other agents or stations. Their goal is to work together to make and drop off a burger. This task is complex as it involves sequential dependencies. Agents must learn to navigate the grid world to locate specific stations and identify agents holding required ingredients. The SimWrapper batches observations from each agent in the worlds and a global observation from each world and sends them to the controller. The controller then feeds these observations into the actor-critic network, which returns critic values (for training), an action for every agent and a communication vector. When using the continuous protocol, each agent receives all the communication vectors from all other agents in the same world. When using the emergent protocol, each communication vector is quantised first before being broadcast to all other agents. Once a pre-defined number of simulations have been run, the data that we recorded in the buffer is used to update the actor-critic network.

I am currently writing the logging module to evaluate my models. All other sections have been completed. After completing the final section, work on extensions can begin.