COMP3180 Design/Technical User Document

Faizan E Muhammad

47752440

## Overview

This document aims to provide a guide on the design and implementation of the AI training system developed for imitation learning using the ML-agents framework in unity.

## System Architecture

* **Environment:** A racing track that consists of multiple checkpoints and a complete car model
* **Car Agent:** (Attached to the car model) is an AI agent used to interact with the environment.
* **ML-Agents library:** A framework that was utilized for the training and development of the AI agent.
* **Checkpoint System:** Used as a way to help with the rewards needed for the demonstration recording and the training of the AI system.

## Architectural Layers

* **Observation Layer**
  + The agent takes some observation of the environment
* **Decision Layer**
  + Takes a decision based on the observation
* **Action** 
  + The AI takes an action based on the decision
* **Reward**
  + Based on the action the AI receives a reward if it is a positive reward then it continues with that action otherwise it does not.

## System Design Details

* **Car Agent Setup:**
  + **Wheelcolliders:** implemented for realistic car physics and controls.
  + **Ray Perception Sensor:** It has built-in parameters that utilise ray casts to navigate the environment.
  + **Decision Requestor:** Used to change the number of seconds after which a decision is made
* **Data Recording:**
  + **Demonstration Recorder:** 
    - Used to record demonstrations for the agent (only utilised if imitation learning is going to be involved)
    - The recorder stores the data:
      * Number of steps
      * Mean reward
      * Number of episodes
* **Training parameters:** 
  + The training parameters are tweaked in the car agent itself and through the config file which includes
    - **Algorithm used**: behavioural cloning
    - **Learning rate:** To adjust the convergence speed
    - **Batch size:** Set for effective data processing
    - **Number training steps:** set for how long do you want the model training to run.

## Usage Instruction

**Setup:**

* Make sure the unity project is configured with the ML-Agents package installed

**Running the Training :**

* Open the provided project
* Navigate to the Car Agent attached to the car model and ensure all the training parameters are set
* Run the Python virtual environment by running the command
  + venv\Scripts\activate
* Initialize the training by running the command
  + mlagents-learn Config/ImitateDriving.yaml –run-id<run\_name>

**Visualizing the Training:**

* The training and the results of the model's training can be visualized using Tensorboard by running the following command in the virtual environment:
  + Tensorboard –logdir results

**Evaluating the Model:**

* The Model can be evaluated by setting the parameter behaviour type to inference and by setting the model to the previously trained model.