**Automating agent decisions in a virtual environment**

Uicoabă Alexandru

1. **Research questions**

* How can the Cloud be used in the process of learning smart agents to play in a virtual environment?
* Which is the optimal number of learning iterations needed for learning smart agents in a kart game?
* Which Machine Learning algorithm fits the best for learning an agent to drive a kart?

1. **Methodology**

The scope of this paper is to teach a smart agent how to drive a kart. It will be used Unity for creating the environment, Python for learning the agent, and Google Cloud Platform for storing the data. The experiment will be stopped when the agent will be mature enough and the RL will not have such a big impact anymore.

* **Unity scene**

To create an environment in which the agent will learn how to drive the kart, it will be used an already existing scene in unity that contains in a kart and a track. In addition, will be added the kart’s sensors and some other scripts will be needed for sending and collecting data.

* **Input/Output parameters**

The Input data will contain information about the position on the X, Y, and Z axes, the sensors, which will detect if the kart is going to hit the track parapet. Another input parameter will be the direction of the kart. If the kart is moving forward (in the right direction), the agent will get a small award. If the kart is going in the wrong direction (backward), the agent will not receive anything.

The output data will be the direction of the kart and the acceleration.

* **Python algorithm**

The Machine Learning algorithm will be created using python and will learn the kart how to drive. It will take the parameters from the unity environment via Python API and then will be used in the ML algorithm to determine the output parameters which will be the action that the kart will take.

* **Storing the data**

To store and compare the data of each scenario of learning an excel will be created and stored into BigQuery in Google Cloud Platform.

* **Comparing algorithms**

To see the different behaviors of the agent, the RL parameters will be modified a few times. In the end, to see which parameters suit the best for our case, a comparison will be created on Google Cloud Platform using AI Notebooks which will create suggestive diagrams based on the data from the BigQuery.

1. **Experiment implementation**

* **The Unity Environment**

To speed up things, it will be used an already created scene on Unity, in which will be added some new features, like sensors, scripts for collecting data, scripts to connect to Python API and Google Cloud Platform.

* **Sensors**

The kart will have 5 sensors on the front as shown in the image below. These sensors will return False if they detect an object and True otherwise. The role of these is important because this information will be used as input for the Machine Learning algorithm.



* **Driving direction**

To receive a little reward, the agent has to drive forward reported to the track. To do that, in each frame, a parameter will return True if the kart is going forward, and false if the kart is going backward or if it stays still.

* **Machine Learning algorithm**

To determine which ML algorithm and which value of the parameter fits the best, it will be used Python alongside TensorFlow for this task.

* **Control the kart**

The output from the ML will be the direction(left or right) and the acceleration/deceleration of the agent.

* **Store data in BigQuery**

For every iteration in which the kart will complete the game, it will be created a .csv file containing the collected data (input data + output data) will be on the local machine. After the iteration is done, when the Game Over on Unity, a script will automatically upload the data from the .csv file into the BigQuery using a specialized API.

* **Alternative implementation**

To speed up the process of agent learning, the various scenarios in which the parameters of the RL algorithm differ can be used by a few virtual machines on Google Cloud Platform which can run in parallel. These VM will have the same Unity environment, the RL parameters being the only difference.