BahiaRT 2021 Free/Scientific: A Custom OpenAI Gym Environment for Skills Optimization on the 3D Soccer Simulation League*

Gabriel Sousa, Filipe Silva, Jadson Nobre, Rafael Fonseca, Vitor Santos, Wesley Silva, Marco A C Simões

A common challenge faced by every team on the 3D simulation league is the optimization of skills. These can range from physical ones, such as kicking, dribbling and walking, to strategic ones, such as decision making. There are multiple available ways to make these optimizations, either by making or not use of machine learning techniques, but since ML algorithms approaches seems to be getting more and more common at each new day, we started thinking about a way to test and evaluate those algorithms in our team, and that's when we heard of the OpenAI Gym[1].

The OpenAI Gym, as it is written on the front page of its website, is a ”toolkit for developing and comparing reinforcement learning algorithms”. *Gym* has a collection of pre-built environments ready for testing, but it also enables you to create your own. Our idea is to create a custom, abstract environment, built on top of the 3D soccer simulation server so that any team, no matter the language it has been coded upon, can make use of it as easy as possible. For now, the only environment class we're developing is for the kicking skill, but we plan on creating more for other types of skills throughout the next year. Since this is an open-source project, that also means that anyone can contribute to this work by creating environments themselves.

In order to create this custom environment, we had to break its architecture into four different processes:

1) **Agents**: The team agents that will be optimized
2) **Proxy**: A ”man in the middle”, designed to manage the communication between the agents and the server
3) **Server**: The soccer server itself
4) **Gym**: The custom environment's class and attributes

The communication between them is mainly made through TCP sockets, but it can also be made using simple text files. The order of execution is as follows:

1 - Start *rcssserver3d*

The first thing to do is to run the soccer server. There are a few variables that can be changed in order to improve the server speed and learning times, but for now we'll just keep it simple.

2 - Start python script

Every user should create a python file that will instantiate the environment's class they desire, as well as start the proxy right at the beginning. Since the proxy uses threads, it will keep running in the background while the scripts execute the rest of the commands. This way, every object created by it can be accessed through the script and have their variables incorporated into the Gym's calculations.

3 - Start the agent

Finally, the agents should be started. Each user should adapt their code to connect to the proxy port instead of the usual 3100 *rcssserver3d* port. Also, they are also responsible for coding their agents to deal with the received data from Gym, either by a socket or a file. Besides those two things, that are directly related to the user individual code, the rest should be good to go.

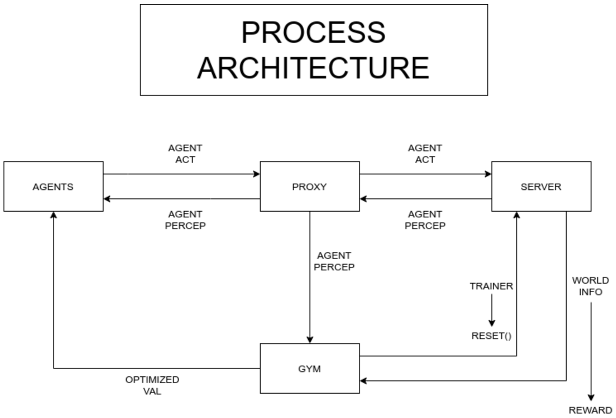

Fig. 1. Custom Environment Processes Architecture

As can be seen in Fig.1, the agents will receive the perceptions from the server and send their actions the same way they would do in a regular simulation environment. The difference will be that the proxy in the middle of that connection should be able to share the perceptions of every agent directly with the Gym, that way it will update itself and do the optimizations without too much interference from the user. The Gym can also send trainer commands to the server, such as changing the *playmode* or beaming a player in any desired coordinate on the field.

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *CoRR*, abs/1606.01540, 2016. _eprint: 1606.01540.

*State of Bahia University (UNEB/DCET-I/ACSO), Salvador, BA, Brazil