Goals

The goal of this Bachelor Thesis was to create and evolve an AI for the Unity game, created in the Project 2 module, using ML Agents and TensorFlow.

It was especially interesting to see how far the so called agents (the AI) would come after a term spent trying to improve them.

Also, it was important to create a variety of agents – agents that are stronger or weaker than others, specialized in different environments or have different roles and abilities such as a warrior, magician or healer agent.

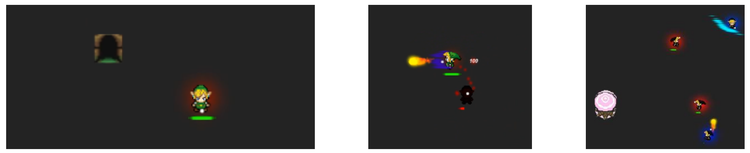
Further, with a variety of agents, it would then be interesting to have a look at usual machine learning problems such as bias, for example by randomizing the layout of the playground (arena) by spawning obstacles (trees), or having various spawn zones – what happens if an AI learns without any obstacles and then suddenly encounters them during the game? What if they are just in different locations? How do the agents perform against past selves, players and other enemies they have never faced?

How it works in a nutshell

Basically, the agents require an agent script that defines what they are able to perceive and what they are able to do. This results in two vectors; an observation and an action vector. Further, the agents receive rewards – either directly defined in that script, or externally from the environment, through events they or another agent trigger trough their actions.
Based on the actions they took and rewards they earn, given a certain observation, the agent tries to find the optimal course of action for each observed state and then updates its policy – in the form of a weight vector.

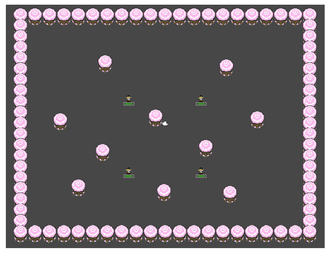
Training Phase

At first, a very basic agent was trained, which would be rewarded for finding an entrance and running towards it. The agent was only aware of its own position and had the ability to either move horizontaly or vertically.



In a further step, the agent was adapted to be able to use two abilities; a fire bolt and a horizontal slash combo, the multi slash. Its target was an immobile training dummy, which would, once defeated, reappear at a random location.

The largest amount of effort was spent on improving the agents – after starting to let agents face themselves using a mechanism called «selfplay», the complexity rose immediately. The agents now had to be aware of their oponent's location, which meant doubling the observation vector's size – and thus also playing around finding the optimal amount of information that would still be trainable, seeing that the bigger the vectors are, the larger the set of posible actions and observed states will become, making it vastly more difficult for the agent to find a beneficial action.



Results

Establishing the environment and starting with the first few trainings was surprisingly simple, however, even though the agents quickly improved thereafter, finetuning them turned out to be quite dificult, especially with the limited computing power. If it were possible to train the agents on 1000 arenas simultaneously instead of the roughly 20 that were possible, the action and observation vector sizes could have been expanded, leading to more «aware», more complex, agents.

Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Reinforcement learning in Unity

Absolvent(en): Sebastian Werthemann
Professor: Prof. Dr. Erik Graf
Experte: Dr. Federico Flueckiger

Bachelor Thesis 2020 Studiengang Informatik