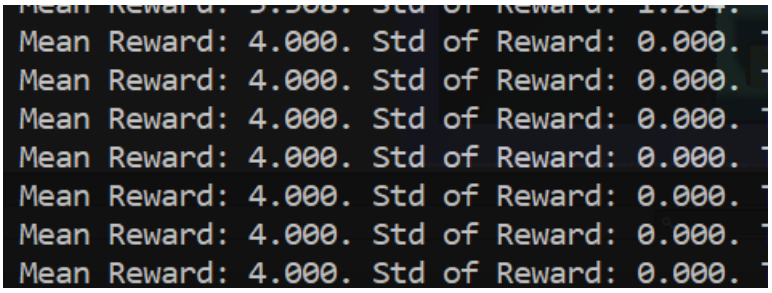
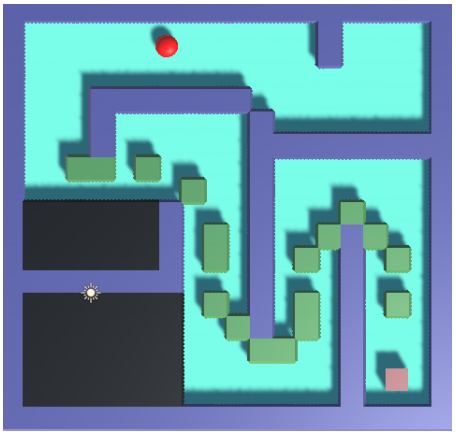
Artificial Intelligence – Project 2

The main goal is to work with Reinforcement Learning with the Unity Asset ML-Agents.

Structure of the maze follows a rather average difficulty after observation. It has not many lost ends and the path is quite well marked with rewards. By closely observing the places where the agent struggles the most to succeed we can deduce what conditions make for a good reward system in order to progress and reinforce fast learning.
For instance, in the very first turn we placed 3 consecutive rewards to prevent errors even if random actions are taken; this way the agent has learned to go through that rather complicated turn after just some iterations crossing the rewards. On the other hand, at the end of the first corridor, the agent took a lot of cycles to learn where to go since the next reward placed after the turn is quite far away and there is a lot of void space it tends to prefer.



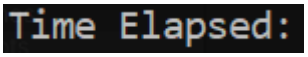
This image is the average view of what we have seen every iteration. We had a very hard time fixing all turns with rewards and modified the maze over 40 times to achieve something. After all the tries we still do not understand what makes the agent take such wrong decisions; could it be our algorithm or either the circuit or even the process of learning itself. Anyways we managed to get some results in the end.



This corresponds to one of the final versions of the circuit. As can be seen, black zones are negative rewards, and so if crossed the agent accounts them as bad options to continue. The turns with most rewards represent the hardest ones (we are not sure why) even though we placed a lot of close and adjacent rewards so that it would learn faster since the difficulty. However, some rewards seem to be useless, and have no effect on the final path.
We must make note that most iterations of this project (by iterations we always mean how many times we forced or resumed the whole learning and played unity), turned into endless loops quite fast. Some of them after a few turns, others just at the beginning. We think the problem came from the values we set up in the configuration file, like randomness and

maximum iterations and stuff. However we managed to solve some issues and achieve a better result.

Following some advice, we decided to use the same tilemap structure to build the maze, since the previous idea we had wasn't working well at all. This hierarchy shows 4 different layers of blocks which constitute the whole maze. The main interest of this lies on the ptr and ntr, which correspond to both positive and negative reward blocks. This can be easily modified using the painting tool provided with the tile map package.
It is also seen the placeholder empty object, which contains various spawn points to be used by the algorithm to make the sphere agent appear at random. After lots of iterations and lost time, we desisted in using those since the whole progress we had made with one single spawn point got taken away by the chaotic learning after adding some more. Thus we decided to go back to the beginning.



Giving some thought to it, we do have the main idea clear, and we understand what the core of the project was. Although, we were confused on how we should achieve the expected goal and the whole process of iterating the maze was so entertaining that we didn't have much time left to spend on extras or some ideas we had to make it look cooler.

Isaac Digón and Marc Ruiz