

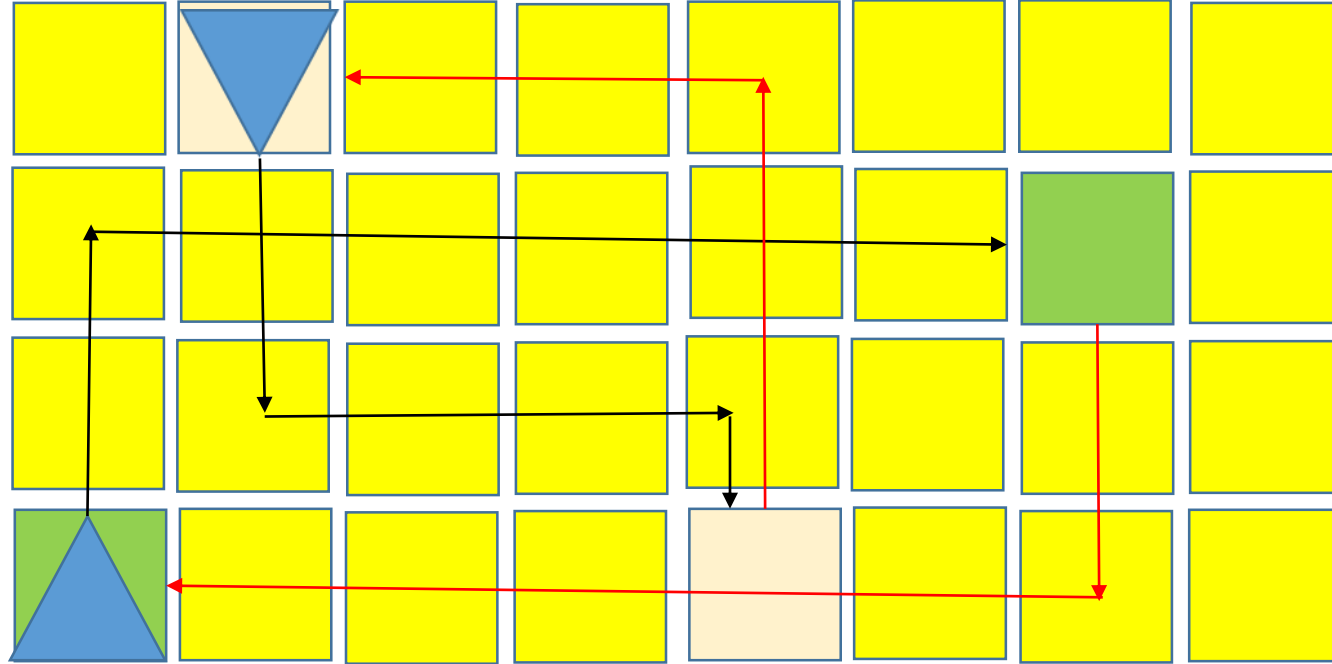
# Robot-Walker

A simple example of a neural network development



Paolo Poli and Alessandro Zanetti

# The goal



Any robot has to walk between the two assigned bases  
avoiding crashes with other robots

# Programs and languages

---

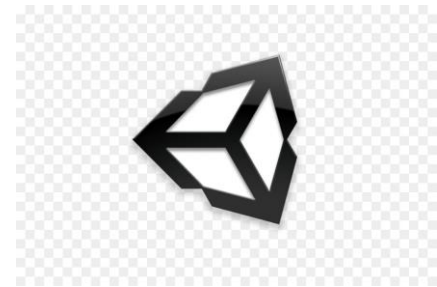
- Python to develop the neural network



# Programs and languages

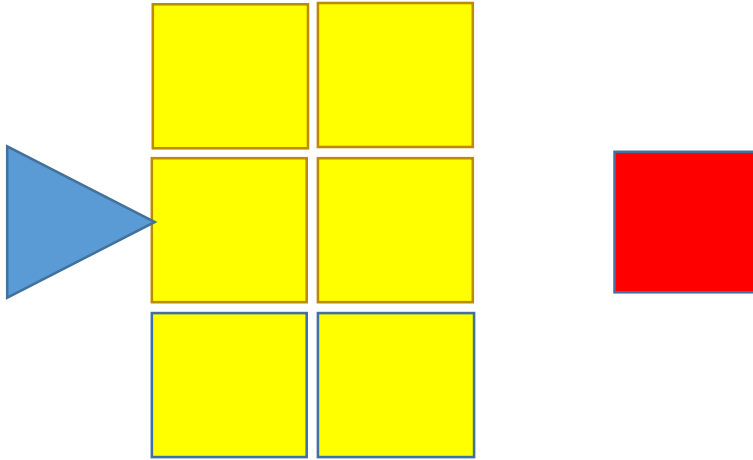
---

- Python to develop the neural network
- Unity 3D to show the dynamic of the simulation

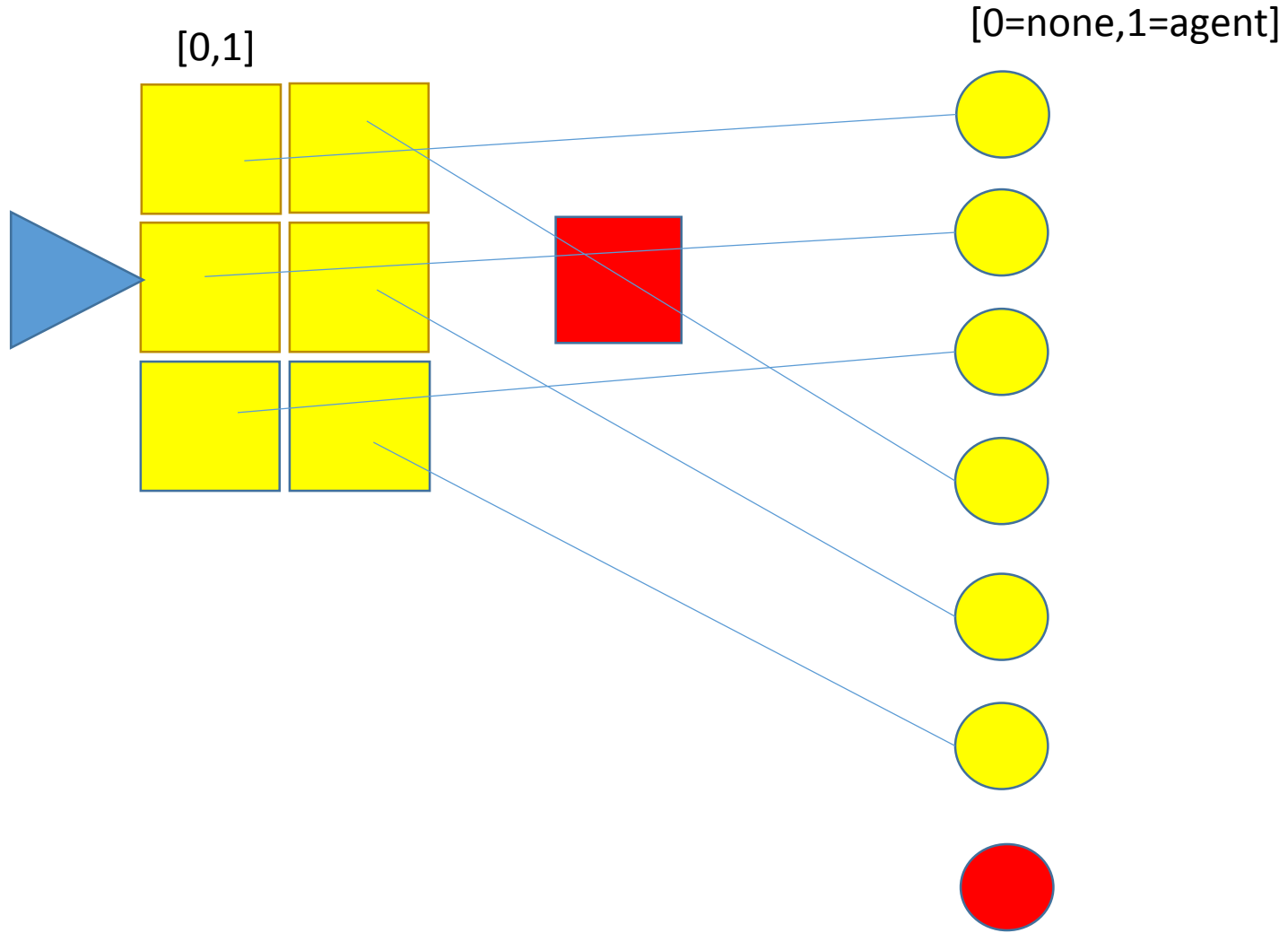


# Neural Network

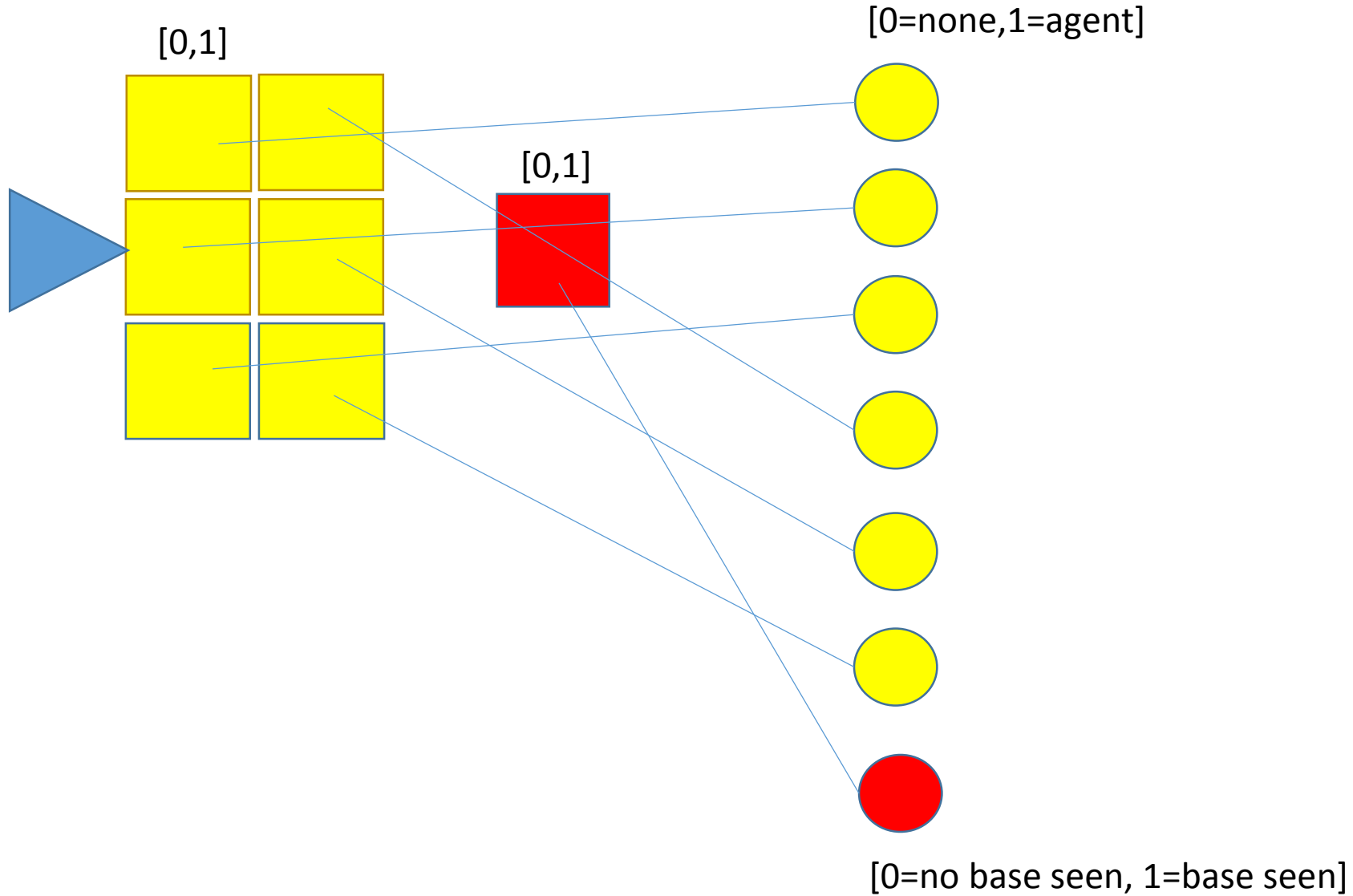
---



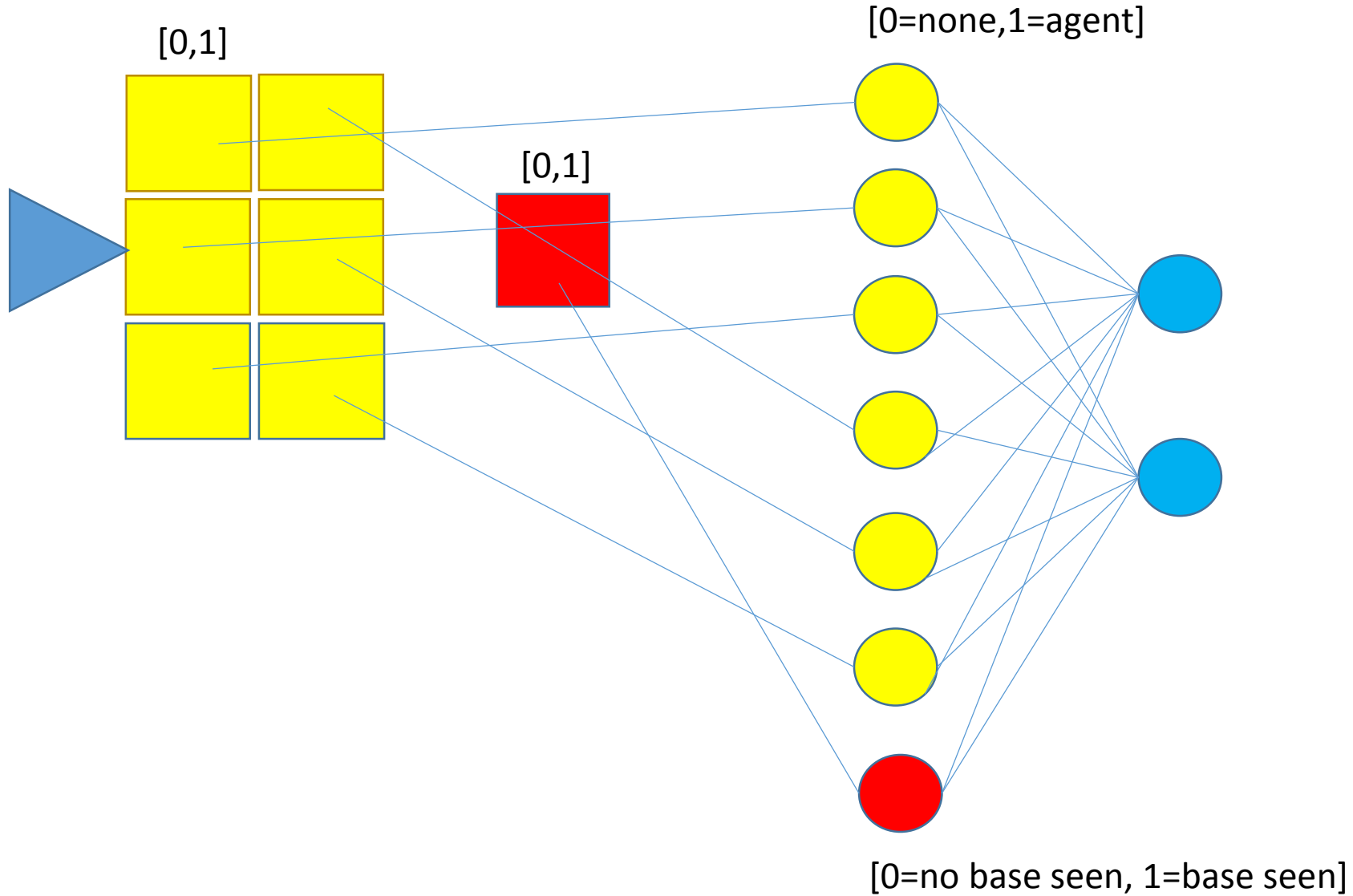
# Neural Network



# Neural Network

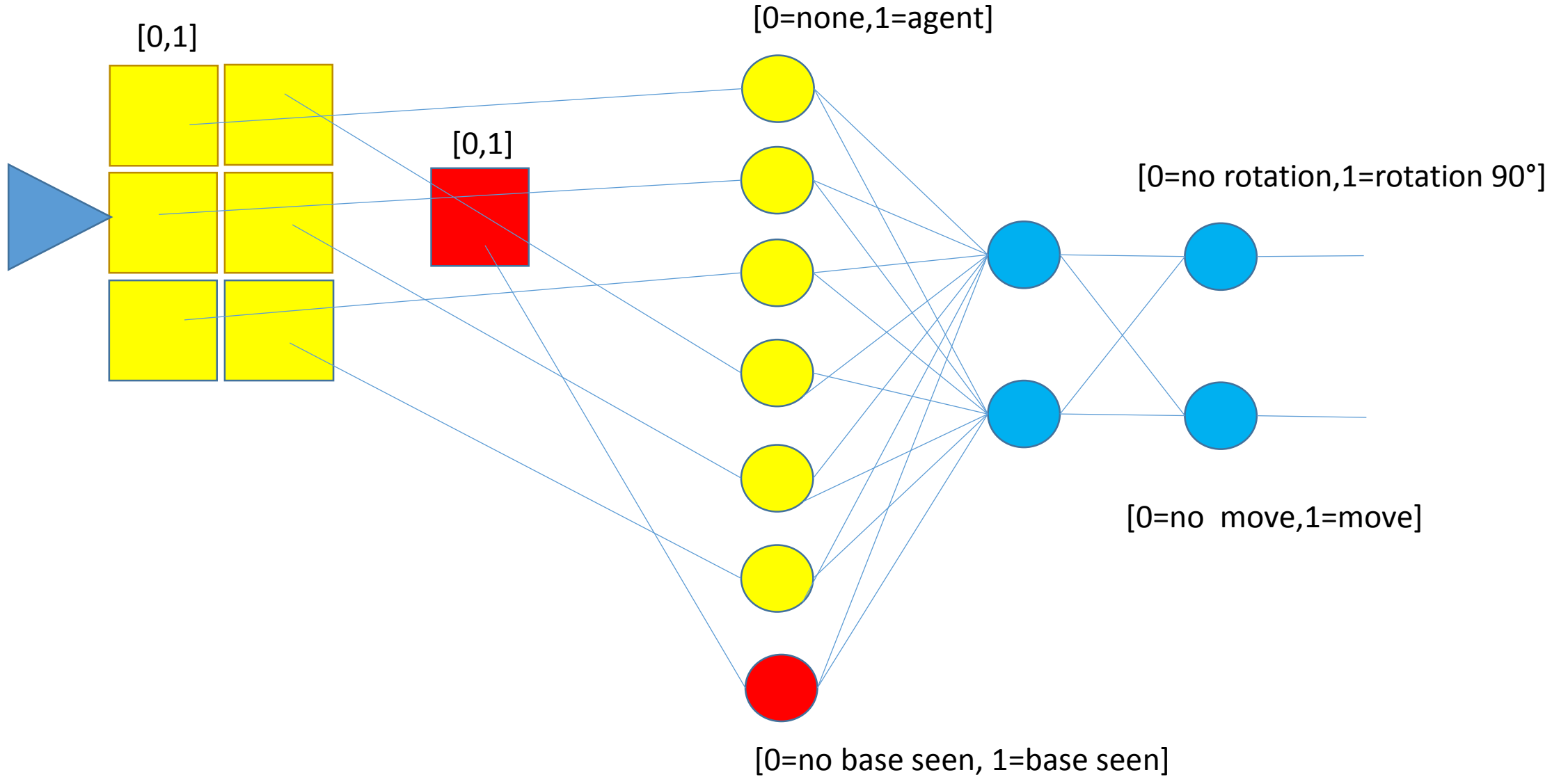


# Neural Network

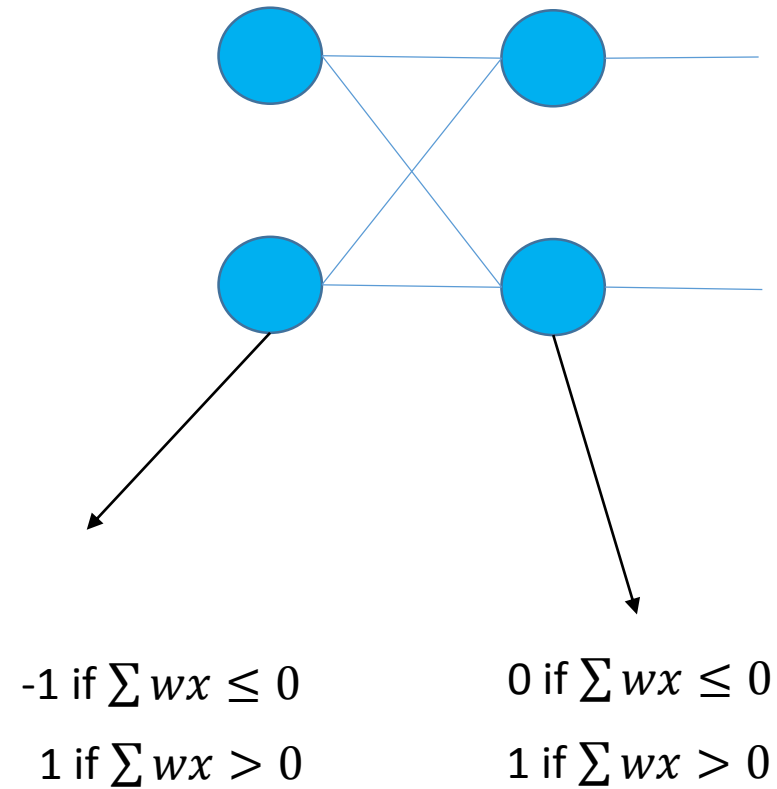




# Neural Network



# Activation functions



Total number of parameters= $14+4=18$

# Neural network training

---

- Greedy method (it's not compulsory to use specific methods in ML literature)

# Neural network training

---

- Naive method (it's not compulsory to use specific methods in ML literature)
  - Two random matrix were used, at the begin of the simulation with value between -1 and 1
-

# Neural network training

---

- Greedy method (it's not compulsory to use specific methods in ML literature)
  - Two random matrix were used, at the begin of the simulation with value between -1 and 1
  - The training works increasing any value of the matrix of a random value between  $-\omega$  and  $\omega$
-

# Neural network training

---

- Greedy method (it's not compulsory to use specific methods in ML literature)
  - Two random matrix were used, at the begin of the simulation with value between -1 and 1
  - The training works increasing any value of the matrix of a random value between  $-\omega$  and  $\omega$
  - The best performed matrix was chosen, any time the result was better of previously results
-

# Cost function

---

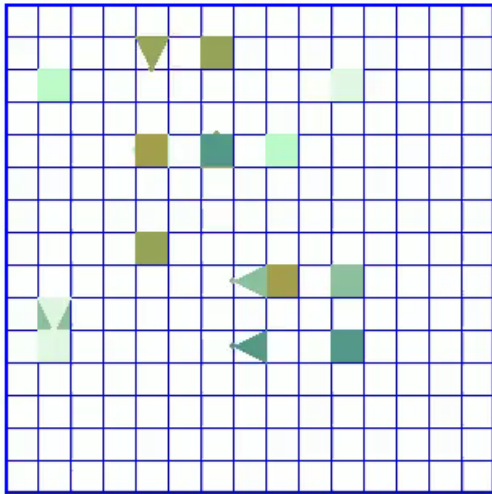
The cost function chosen, is the sum of the number times each robot reaches one of the two assigned bases

---

# Results

The cost function chosen, is the sum of the number times of each robot reaches one of the two assigned bases

Untrained

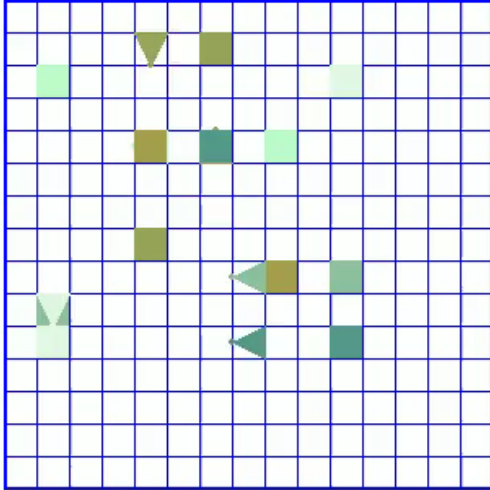




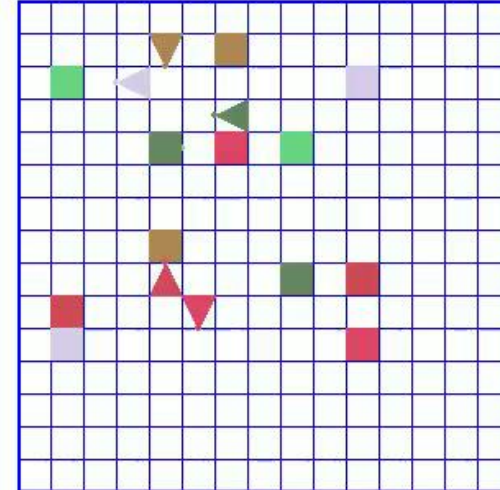
# Results

The cost function chosen, is the sum of the number times of each robot reaches one of the two assigned bases

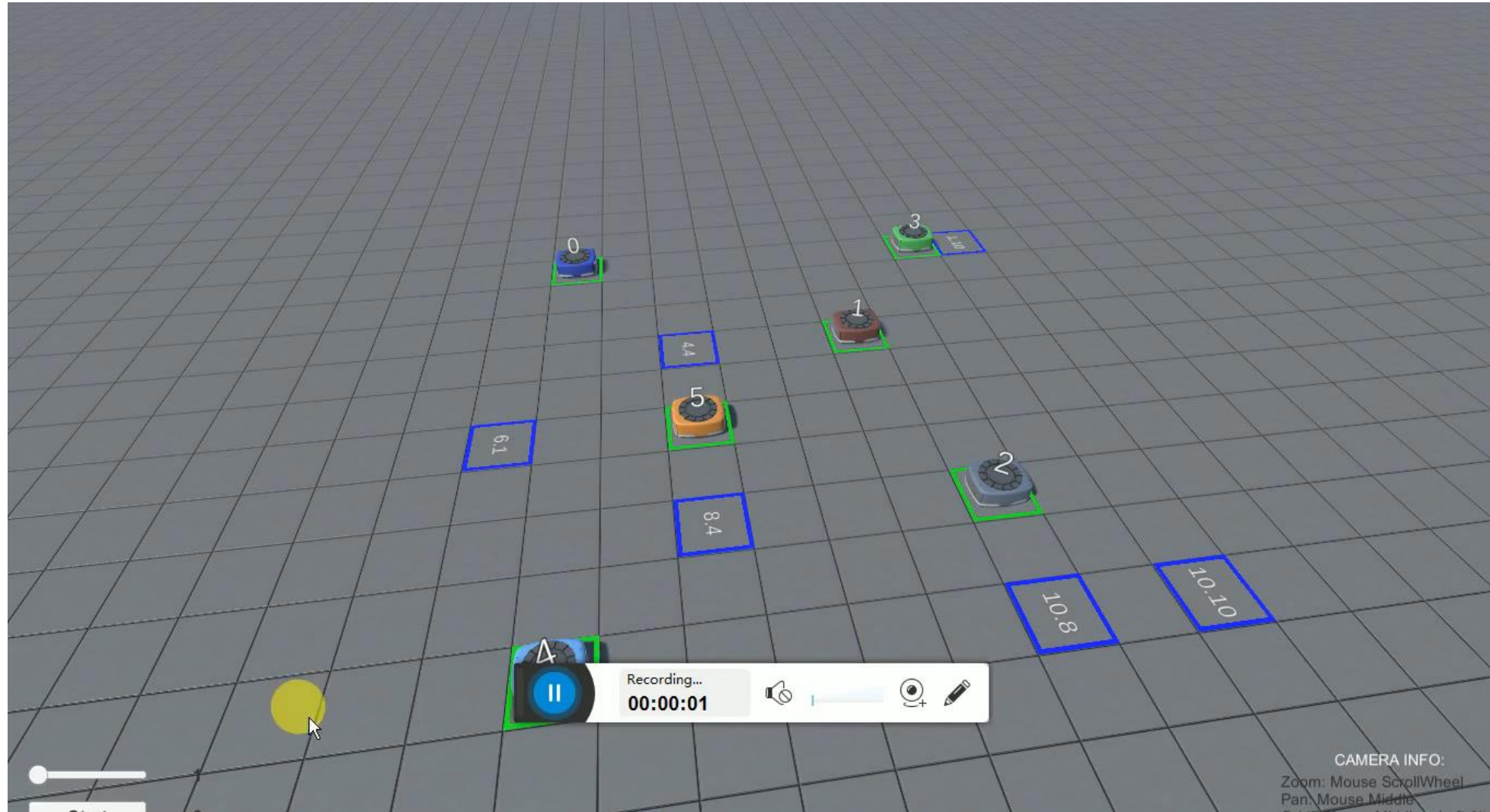
Untrained



trained



# Results in Unity 3D



# Conclusions

---

- An interesting example of problem where a simple neural network was coded has been shown
  - The goal was reached only by a simple laptop and with a normal home pc
  - New things can be added at the problem putting new goals at the problem
-

Thanks!

---

Thanks everyone!

---