

## Some tips: Advanced Machine Learning Assignment

---

1. To help the robot to succeed in the early stages of the training, you can make sure the target is close to the robot at the beginning of an episode. Later, when the robot is better trained, you could relax this requirement (and don't use this during the 50 episodes evaluation). You could modify the `new_episode()` function to position the target close, like in the code snippet below. You also need to provide a `distance` function for this to work.

```
if episode < 1000:
    max_distance = max(50, episode)
else:
    max_distance = 9999

while True:
    robot_pose = [random.randint(FIELD.left, FIELD.right),
                  random.randint(FIELD.top, FIELD.bottom),
                  0] # random.randint(0, 359)
    target_pos = [random.randint(FIELD.left, FIELD.right),
                  random.randint(FIELD.top, FIELD.bottom)]

    d = distance(robot_pose, target_pos)

    if d <= max_distance:
        break
```

2. **Simplify the states and actions:** As discussed in the lecture, the robot really only needs to distinguish two states for a minimal policy: the ball is “in front” of the robot, or it is not. You can define where “in front” is, and how accurate this needs to be. With these two states, you can get by with just two actions: (1) spin on the spot (say, clockwise), and (2) drive forward. Then a (manual, and minimal) policy would be: “if the ball is in front of the robot, drive forward. Otherwise, spin on the spot.”

To create this policy, you would not actually need to use RL at all (since you know the best actions), but it may help to make sure your implementation works as it should. You can also use this policy to train the approach for task 4.