In this challenge, we will use a distance value to calculate fitness rather than the length of time a bot has been alive. This will allow us to train them to not only stay on the platform but encourage them to be moving along the platform rather than “cheating” their survival by learning to simply jump or stand still.

I will provide some clues here, and fully worked solutions as script files your teacher can make available to you if you are having difficulties. Simply, we want to record the distance travelled for each bot. That means the code will need to store their starting location, and then record how far they have travelled.

Think about where these properties need to be added:

public float distanceTravelled;

Vector3 startPosition;

Hint: Which program is responsible for communicating with the DNA?

Next, set a start position within initialise:

startPosition = this.transform.position;

Following this, update the start position:

if(alive)

{

timeAlive += Time.deltaTime;

distanceTravelled = Vector3.Distance(this.transform.position,startPosition);

}

Finally, we need to sort the list based on the distance travelled instead of the timeAlive:

List<GameObject> sortedList = population.OrderBy(o =>o.GetComponent<Brain>().distanceTravelled).ToList();