

## ARTIFICIAL INTELLIGENCE

### Second Practical Assignment

#### ***LADYBUGS' UNDERGROUND LIFE. ANTNEST KEEPING***

**The ladybugs** work for the ants doing nestkeeping tasks. They move eggs to the hatching chamber and transport seeds to the store chamber. They do this tirelessly and following a perfectly established routine:

When they have nothing to do, they just wander through the galleries of the underground nest. When wandering, ladybugs continuously scan their surroundings for eggs or seeds. Priority is given to eggs. They go for

- The closest egg (detection radius: 50)
- If there's no such an egg, a randomly chosen one (detection radius: 180)
- If when going for an egg a closer one is detected (detection radius: 50), this egg becomes the ladybug's target.

If no egg has been detected, they go for

- The closest seed (detection radius: 80)
- If there's no such a seed, a randomly chosen one (detection radius: 125). When a ladybug has decided to target a seed, the apparition of a nearer one does not make it change its mind.
- If when going for a seed or carrying one, an egg is detected the ladybug forgets the seed and targets the egg (detection radius: 25). If it was carrying a seed, the seed is dropped.

Seeds and eggs become detectable when the carrying ants drop them. While held by an ant or another ladybug, seeds and eggs are undetectable<sup>1</sup>

Eggs are always moved to the hatching chamber<sup>2</sup>. while seeds are always moved to the store chamber<sup>3</sup>

**The ants** deliver<sup>4</sup> eggs and seeds. Once their load has been dropped, they exit the scene. They never drop seeds or eggs inside the chambers (hatching or store)

Ants are created off-scene by two ant spawners located at the end of the entry points. When they have done their job they chose an exit point and disappear.

Ladybugs are grateful when the ants place the seeds and the eggs exactly at the centre of a waypoint. The following snippet of code

```
GraphNode node = AstarPath.active.GetNearest(load.transform.position,
                                              NNConstraint.Default).node;
load.transform.position = (Vector3)node.position;
```

moves load to the centre of the nearest waypoint (node). Use it where appropriate.

---

<sup>1</sup> It's just a matter of tags.

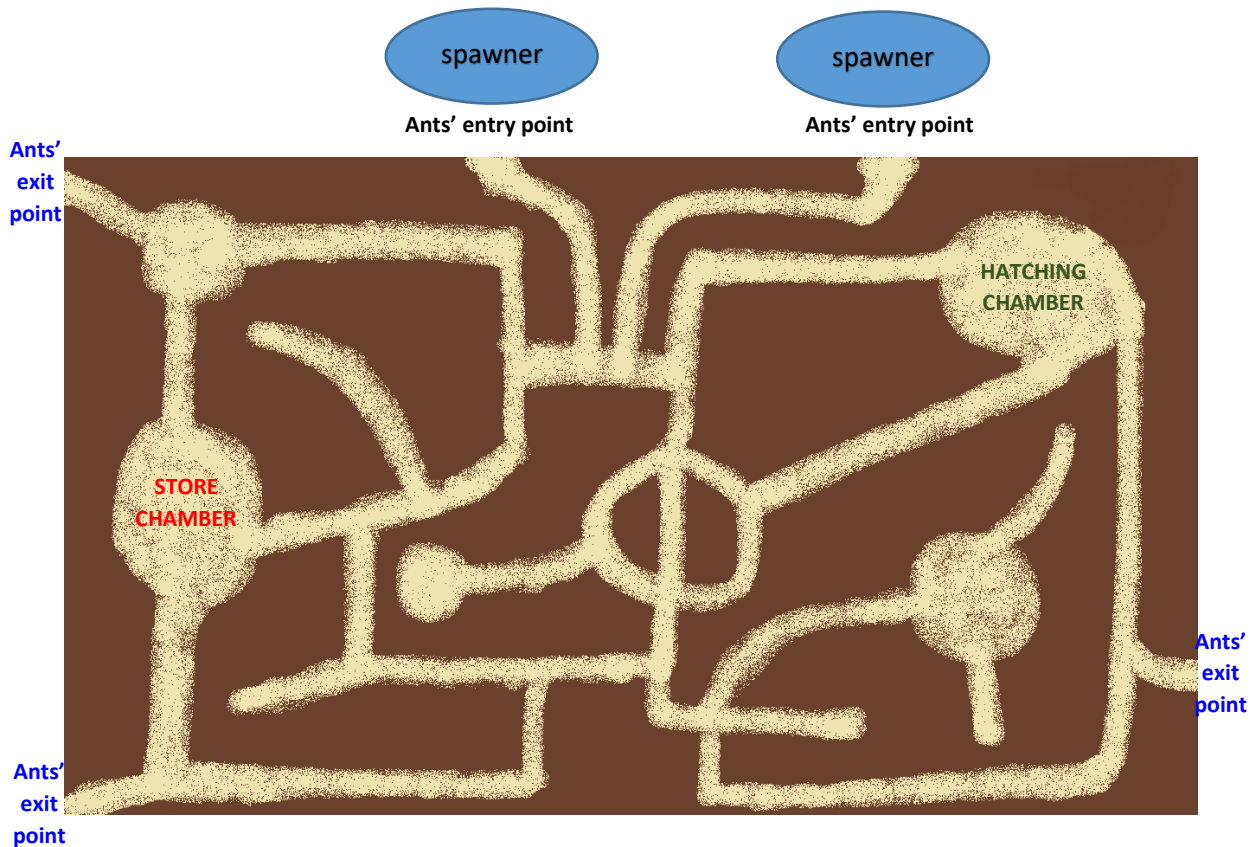
<sup>2</sup> The final location (point) is randomly selected.

<sup>3</sup> Id.

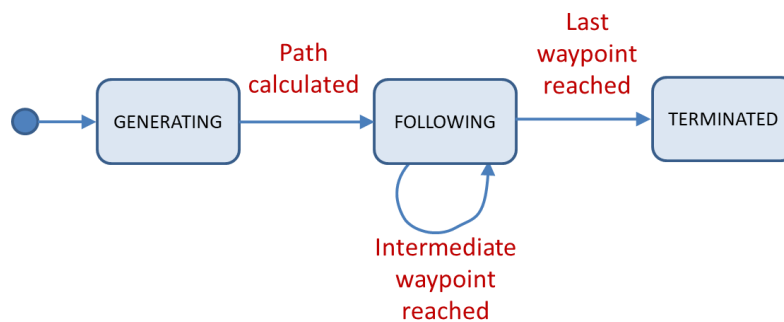
<sup>4</sup> The delivery point is randomly selected (never inside the hatching or store chambers)

### The spawners:

- Create an ant every 15 +/-10 seconds
- The probability of creating an ant carrying a seed is 0.8 (thus it is 0.2 for an ant carrying an egg).



Optionally, you may consider the possibility of constructing a two-parameter<sup>5</sup> FSM-based route execution behaviour



This behaviour is well suited for non-moving targets and substitutes the pair pathFeeder + pathFollowing.

For both ants and ladybugs, you are recommended to set to 2.0f the radius at which a point or an object is considered to be reached.

---

<sup>5</sup> The target to reach and the "waypoint reached radius"