**Team Taco Code Challenge:**

This code challenge involves a simulation of ants in an environment. It is a voxel simulation involving three voxel types: Dirt, Nest, and Food. Ants behave autonomously, and must adhere to a set of rules provided in this document. It is your job to code these rules properly.

**Calls:**

In order to program this, ants must make calls to the environment every tick of FixedUpdate(). To do this, call Environment.Instance.<Method>(this). The only parameter you may pass in is the calling code reference (With the exception of feed). The calls are as follows:

All calls to the environment reduce the health of the ant by one (with the exception of GetCurrentBlock).

Dig: This will cause an ant to destroy the voxel in front of it, decreasing its height by one. It will then cause an ant to have their voxelCarried variable to be set to true. If an ant is already carrying a voxel, this call will fail

Place: This will cause an ant to place a voxel directly in front of it. If the neighbours of the placed voxel are a nest type, the placed voxel will also become a nest type. This will also set the ants voxelCarried variable to false. An ant cannot place a voxel if this variable is not set to true.

Eat: This will check if the ant is currently standing on a food voxel. If so, it will replenish the health of the ant to the max. If the food resource does not have enough to fully replenish the ant, it will replenish the ant with as much food as it has, then turn into dirt.

Feed: This will take health from your current ant, and give it to the nest. Every nest block has a probability of spawning new ants based on how much food it has.

Move: This will cause the ant to move forward in the direction it is facing. Ants cannot climb blocks greater than 1 unit difference from itself

Others: this will return a list of all ants on the same voxel as the current ant.

GetCurrentBlock: This will return the voxel reference to the voxel the ant is currently residing upon. This is the only call which does not reduce the ants health

**Rules:**

-Ants May only make one call to the environment per update.

-You may change the orientation (N, E, S, W) of the ant, but in doing so must also make the ant reflect this change visually.

-Ants are unable to move to a new space that is greater than 1 unit difference than the voxel the ant is currently on (The environment already will prohibit this and notify you if it does. You will also need to handle this in your code however).

-Ants are unable to dig, or place voxels whose height is greater than 1 unit difference to the current voxel. (same as above with being prohibited)

-Only scavenger ants are allowed to feed the nest

-only worker ants are allowed to dig/place

-Ants will ALWAYS prioritize their own health, and replenishing food before death, before anything else

-Ants begin with no knowledge of where the nest blocks are. If an ant has no knowledge of the nest block, it will attempt to find the nest.

-Ants occupying the same voxel may share knowledge about where the nest is

-Scavenger ants will prioritize bringing the nest food

-worker ants will prioritize expanding the nest

-worker ants are supposed to be blue

-scavenger ants are supposed to be red

-when an ant has picked up a voxel (I.e voxelCarried == true), it must show a black block above it.

-The goal is to expand the nest as much as possible, with as few ant deaths as possible

Note: You are allowed to alter the provided code so long as it does not go against the spirit of the challenge (I.e. making it so ants never die, etc.)

If you find the challenge to be too easy, you are encouraged to create and implement your own rules. The provided rules are a floor, not a ceiling.