

CS 2316

Homework 11– Ant Farm

Due: Wednesday, Dec 5th, before 11:55 PM

OPTIONAL EXTRA CREDIT ASSIGNMENT!

Files to submit: 1. antHill.py

This is an INDIVIDUAL assignment!

Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 2316, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
 - *Do not wait until the last minute to do this assignment in case you run into problems.*
 - **Read the entire specifications document before starting this assignment.**
-

Premise

You are in control of a robotic ant hill in an ant farm. Unfortunately, another competing ant hill exists on the far side of the ant farm. You need to find and eat more food than the other ant hill, and protect your ant hill. (Basically, this is simplified logistics, path planning, and strategy problem, and may remind you of mining, shipping, manufacturing, or other similar business problems.)

The "world" is flat (2 dimensional) and made up of a 100 x 100 grid (coordinates range from 0 to 99 in the X and Y axis, 0,0 is the top left, and 99,99 is the bottom right). Each grid space can hold an ant farm, an ant, a wall, or a piece of food. Grid spaces with nothing in them are empty, and ants can move into empty spaces. If an ant moves into a space with food, it eats the food. When an ant eats food, the ant hill it came from spawns (grows) a new ant! This is how your ant hill will gain more ants. The more ants you have, the more food you can eat and area you can explore!

But watch out, the evil red ant hill on the other side of the ant farm is also eating and growing, and they are looking to take you out!

If an ant tries to move into a square that is a "wall" (representing a rock, tree, or other obstacle) it will smash it's head and die! (Hint: Make sure your ants don't move into any walls!)

When an ant moves into a square occupied by another ant the two ants fight and they both die! When ants die, they disappear...note that friendly ants will still fight if they find themselves in the same square! (Hint: Make sure your ants don't move into the same square to keep them from fighting each other!)

If an ant enters a square that contains an ant-hill it destroys the ant hill! If your ant hill is destroyed, the enemy wins! (Hint: Make sure you ants never walk into your ant hill!)

You are a peace loving ant nation, but the red enemy on the other side of the ant-farm has been known to try and destroy your ants and ant hills by sending kamikaze ants to fight/destroy them. You must be prepared to defend yourself! (Sometimes the best defense is a first-strike offense...)

When you start, you have a single ant located next to your ant hill. It will need to go find food to survive!

Application Programmer Interface

You will create a module (antHill.py) that has at least two functions. (You may create as many additional global variables or helper functions as you need.)

Function Name: **initialize**

Parameters:

'hillInfo' - A tuple of information about your ant hill

Return Value:

aTuple containing 2 strings

-Your Name

-Your ant hill's name

The initialize() function will be called by the ant farm server when a game is about to start. (It will only be called once, and it is guaranteed to be called before the "tick" function is called.) Your initialize function should get your module ready to play!

The single "hillInfo" parameter will consist of a tuple of data about your hill. The format is the same as in the "tick" function below. See the description in the "tick" function (below) for a detailed layout of what the hillInfo tuple contains.

Your initialize function must return a tuple with two strings. The first string is your name,

and the second string is the name of your ant-farm nation. For example, you could end your initialize function with the following if you had my name:

```
return ("Jay Summet", "The Green Army!")
```

Note that your ant hill's name may be displayed publicly to the class when revealing the results of the round robin tournament, so keep it clean, and don't reference your own name in your ant hill name if you want to be anonymous to your classmates.

Function Name: **tick**

Parameters:

'hillInfo' - A tuple of information about your ant hill

'antInfo' - A tuple of information about all of your ants.

Return Value:

antMoves – a tuple of values indicating how you want each of your ants to move.

The tick function will be called multiple times. Each time it is called, it will give you two tuples: “hillInfo” and “antInfo”. These tuples contain data about your ant hill and all of your ants, respectively.

The hillInfo tuple: This tuple contains 3 integers.

- The first two are the X and Y location of your ant hill.
- The third is the total number of ants your hill has spawned. Because your hill spawns a new ant each time an ant eats a piece of food, plus you get a “free” ant at the beginning of the game, this number will always be one more than the total number of food items your ants have eaten during the game.

The antInfo tuple: This tuple contains one sub element for each ant you have in play on the board. You may assume that it will have at least one sub-element / ant. (If your total number of ants reaches zero you lost the game.) Each ant sub-element is a tuple with the following information in it:

- Two integers, representing the X and Y location of the ant.
- A tuple that lists information about the 24 squares immediately surrounding the ant, ordered as follows:

23	24	9	10	11
22	8	1	2	12
21	7	Your Ant	3	13
20	6	5	4	14
19	18	17	16	15

Each square will contain only one of the following items:

- "" - Empty String – Representing an empty square
- “FH” - Friendly Anthill (Your ant hill, don't crash into it!)

- “FA” - Friendly Ant (One of your ants, don't crash into it!)
- “W” - A Wall (stick/brush/rock, etc...) Don't crash into it!
- “EA” - Enemy Ant
- “EH” - Enemy ant Hill.
- “F” - FOOD! This is the good stuff, go eat it!

Once you receive the information about where all your ants are and what they can see around them, you need to determine what you want each ant to do! You return these actions as a tuple of “movements”. Each of your ants can either stay where it is (0) or move into one of the eight squares around it (1-8 in the diagram above). You will return a tuple such as (0,1,1,1) to indicate that your first ant will stay where it is, and your next three ants will all move one square forward.

Keep in mind that your ants will take their movements in order, and if they walk into a square occupied by another ant (enemy or friendly) the ants will fight and both will die, leaving the squares empty. So make sure that your ant two doesn't try to move into the space that ant three is currently in, because ant three will not have moved out of the space before ant two moves into it!

Winning The Game

You can win the game in two different ways:

- Eat more food than your opponent. 100 turns after the last food item is consumed, the game will end and the player with the highest total number of food items eaten will be declared the winner. Note that the number of ants you have is not considered (as they may die during the game.) NOTE: If no food items are consumed for a hundred turns, the simulation engine may "end" the game and declare a winner even with food items left on the board.
- Destroy your opponent's ant hill. If your ant hill is destroyed, (even if accidentally by a friendly ant) the game ends and the other player automatically wins. Note that destroying your opponents ant hill can be a faster way to win the game. You should use ants to protect your ant hill from enemy ants!

Grading:

You will earn points as follows for each piece of functionality that works correctly according to the specifications.

Initialize		5
Returns correct information.	5	
tick()		30
Accepts proper input parameters	10	
Returns appropriate antMovement tuple	10	
Finishes execution within 1 second	10	
General Behavior		30
Ants do not destroy their own ant hill	5	
Ants do not destroy friendly ants	5	
Ants do not destroy themselves (walking into walls)	5	
Ants seek and eat food	15	
BetterThan Random		10
Your antHill wins at least 6 of 8 battles against the “Random” hill.		
BetterThanPurposeful		10
Your antHill wins at least 6 of 8 battles against the “Purposeful” hill.		
GoodAsSmart		10
Your antHill wins at least 3 of 8 battles against the “Smart” hill.		
BetterThanSmart		5
Your antHill wins at least 6 of 8 battles against the “Smart” hill.		
ExtraCredit		0-50
Your antHill wins battles against the best other antHill's in the class! Although the exact EC grading mechanism is secret and may change from semester to semester, generally the top 20% of each section will receive around 10 EC points and advance to the “whole class” round robin tournament. The top 2-5 ant Hills will earn 50 points, and the rest will earn points in the 40, 30, 20, or 10 point range depending upon how well they do.		