Greenfoot Two Animals Assignment Lists of Objects

\*I would like to acknowledge a significant amount of help from Jordan Cohen.

Marks for this Greenfoot scenario will be based on the **comments** in your code, and on your verbal explanation / **interview**.

This scenario will use Lists of Objects. One of the animals will always move towards the closest Food object. You should create a getDistance() method to calculate the distance between two objects. The Math.hypot() method will be very useful for this.

The other animal will move to the most recently spawned Food.

Each animal scores points by eating Food objects. The first animal to get 25 points wins the game.

Actors: THREE Actor classes: two different animals, one Fruit ( food for the animals )

Worlds: There will be TWO World classes

AnimalWorld and WinScreen ( You might want two Win Screens – your decision )

AnimalWorld

You will need two score counters (one for each Animal)

Add the two Animals to the World. Their starting position must be somewhat random.

Decide how you will control the spawn rate of the Food objects (relatively slow).

**act() method**

Spawn the Food objects

Check to see which score is the first to reach 25. Move to the appropriate WinScreen.

**addPoints() methods**

There will be two addPoints() methods ( one for each Animal )

Each addPoints() method will be called from the corresponding Animal class

Update each Score label.

Be sure to position the two score labels on different sides at the top

WinScreen ( one or two – your decision )

Choose a different background from the AnimalWorld class

Display a message indicating which Animal won the game.

Actors: Both Animals will START at speed 4. Animal2 will move faster eventually.

Animal1

Initialize the speed to 4 ( constructor is optional )

**act() method**

*if there is at least 1 Food object...*

Add code to determine the most recently spawned Food object

Turn towards that object and write code for the collision

Call the corresponding **addPoints()** method, add TWO points, and remove the Food object

Animal 2

Initialize the speed to 4 ( constructor is optional )

**act() method**

*if there is at least 1 Food object...*

Add code to determine the closest Food object

Turn towards that object and write code for the collision

Call the corresponding **addPoints()** method, add ONE point, and remove the Food object

When the points scored reaches 14 the speed will increase slightly to an amount of your choice.

Food

Does not require any code