

CSE143X Lecture Questions for Friday, 10/23/20

Time (e.g., 12:45)	Question	Answer
19:30-19:41	You mention that younger critters infect more readily but I don't see that in the assignment writeup or CritterModel.java. I only see that hopping gives a 100% chance of infection whereas non-hopping critters have an 80% chance.	You're right...I was remembering incorrectly. It has to do with hopping, not age.
	Is there any time/memory limit for Husky to return from getMove()?	Yes and no. In general, we don't limit the time and memory, but I do disqualify anything that takes a ridiculous amount of time (somewhat subjective).
	<p>It would be nice if there was a way to get the following:</p> <ol style="list-style-type: none"> <li>1) width/height of world</li> <li>2) number of critters (all types)</li> <li>3) number of friendly critters (i.e. Huskies in the case of Husky)</li> <li>4) number of steps elapsed</li> <li>5) total number of steps in game</li> </ol> <p>Is there any way the classes could be changed to allow public access to those with class constants/method calls? #4 is a top priority.</p> <p>Re: Ok, #1 and #5 could be local constants that duplicate the actual values, and #3 and #4 could in theory be calculated (code won't be very nice though)</p>	You are not allowed to modify the simulator, so you can not get access to any of these. Some can be computed.
	I read the spec before class and almost had a panic attack so thank you for explaining it :P	There is a lot of extra stuff for making interesting critters that you'll be able to ignore. The core assignment isn't that complicated. Glad the explanation helped.

	<p>What is CritterInfo in relation to the other classes? I assume each Critter object created has one, so is it a child, parent, sibling (idk.) of Critter?</p> <p>Ok, so this is passed from the simulation, of which we don't really need to pry open and understand, just use the methods?</p> <p>You mentioned the state of the world, so is there one CritterInfo for each or just one CritterInfo for the entire simulation?</p> <p>XD. Interesting. Thanks!</p>	<p>Each time that getMove is called, your critter is passed a CritterInfo object that has information about the state of the world.</p> <p>Yes, the simulator creates these objects and passes them as parameters when it asks each critter for their next move.</p> <p>The simulator constructs a gazillion CritterInfo objects (gazillion is a technical term). :-)</p>
<b>33:05</b>	<p>How come the lion class bounces back and forth? I was expecting them to turn left if they hit a wall in front? OOHOO THEN THE WALLS TO THE RIGHT SO IT TURNS AGAIN</p> <p>I see whats going on. Thank you</p>	<p>Welcome to the world of emergent behavior. The lion will run into a wall, at which time a wall is in front. Then it turns left. That leaves a wall to the right and it turns left again. So it has turned around.</p>
	<p>Did you get a haircut? Looks fresh</p>	<p>Yes, that is so. Thanks.</p>
26:57	<p>Is it typically more favorable in general programming to have no constructor or define a zero parameter constructor?</p>	<p>There isn't a hard and fast rule about this. I tend not to include a constructor if there is nothing to do.</p>
	<p>The mic is still kind of staticky (no idea if that's an actual word).</p>	<p>I'll report it.</p>