

# CSE 142: More Critters!

---

Weds, Nov. 26<sup>th</sup>

**Reading: Section Handout**

# Today's Agenda

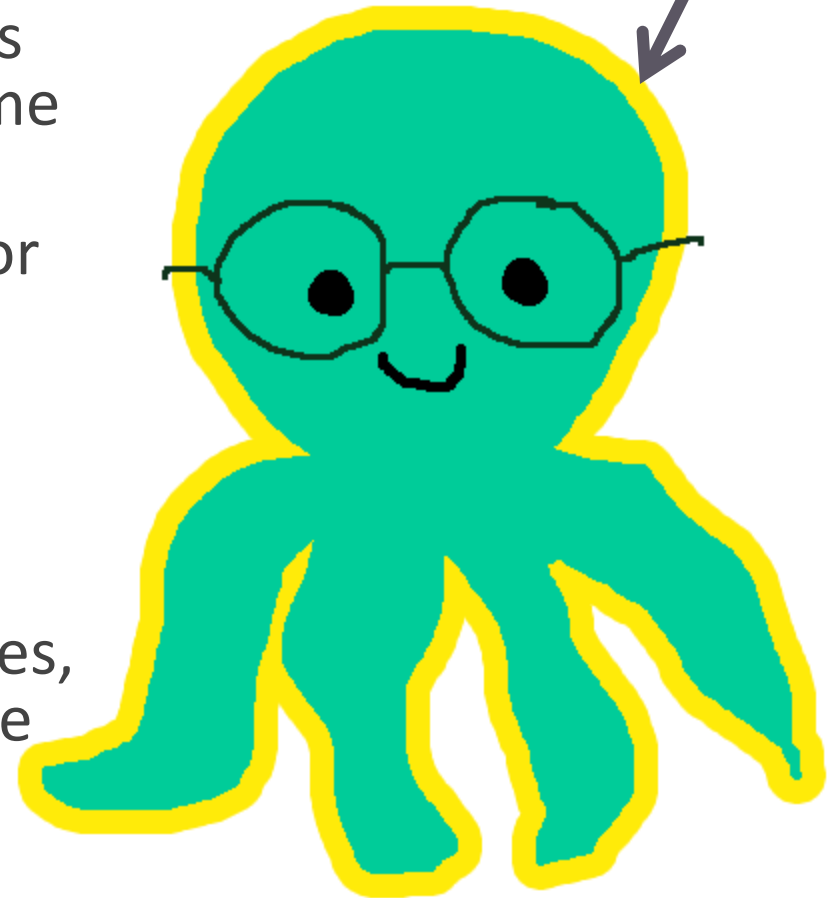
---

- Review ideas behind Critter assignment, answer questions
- Practice writing Critters
  - Critters will also be on the final!

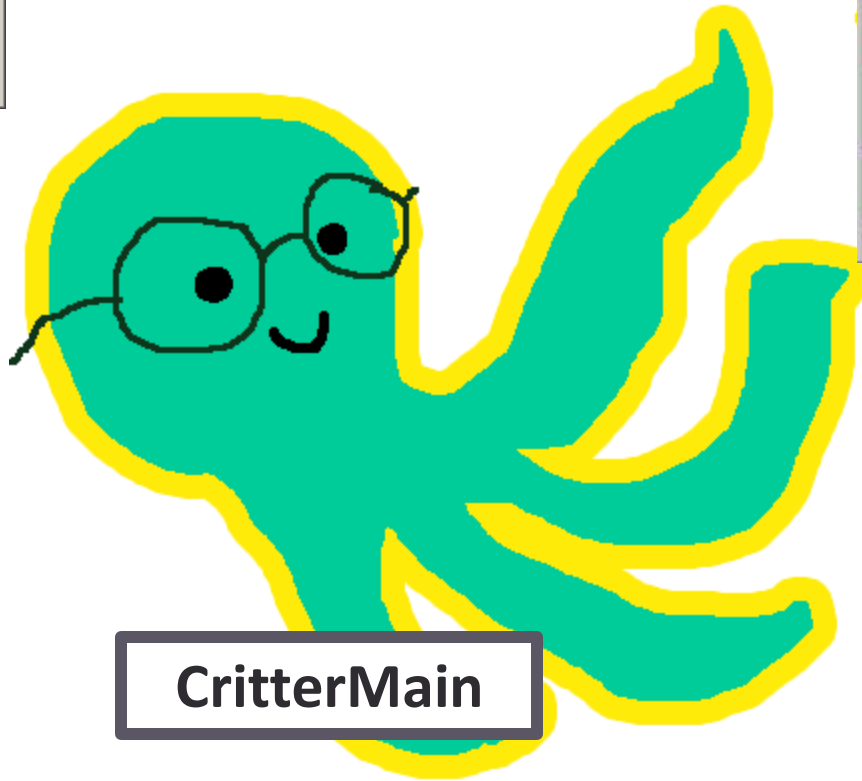
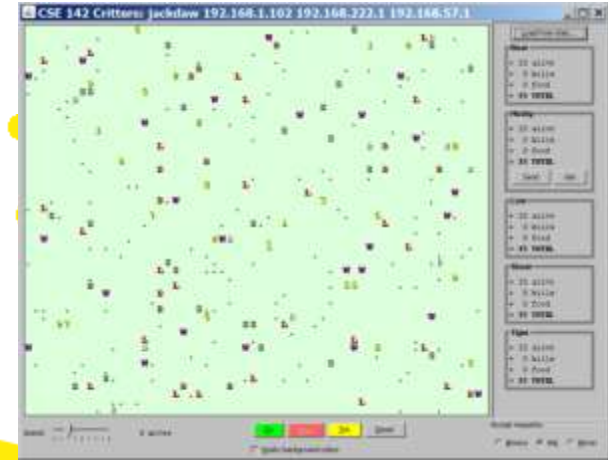
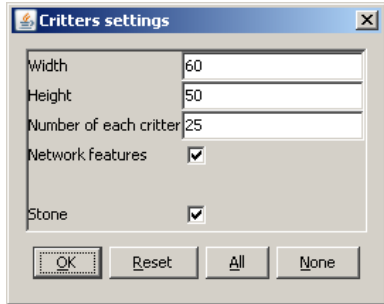
# Role of CritterMain

- Think of CritterMain as **Game Master**
  - Does the behind the scenes work to administer the game
  - Constructs your Critters, places them in the world for you
- **What the Game Master does you do NOT have to implement!!**
  - You create the Critter classes, which are like **players** in the game

CritterMain the Game Master



# Review of CritterMain



**CritterMain**

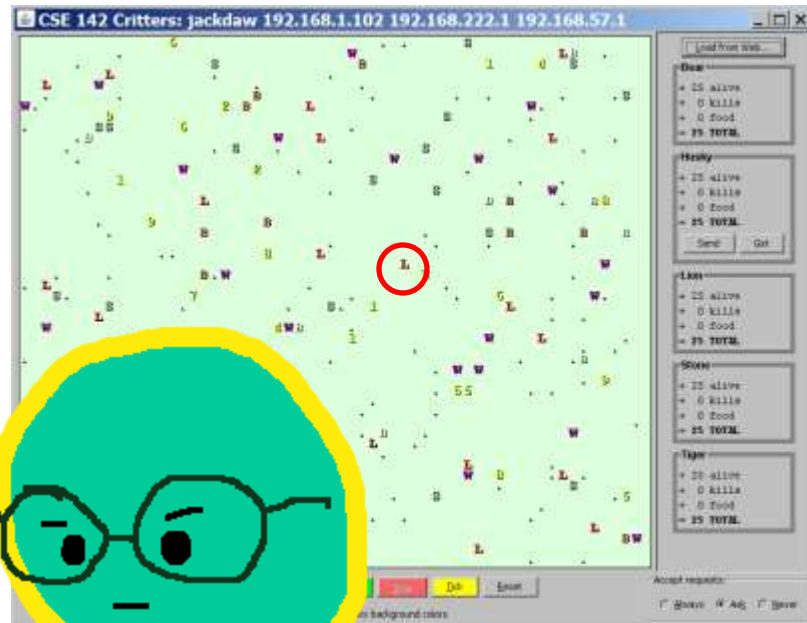
# Role of CritterMain, 2

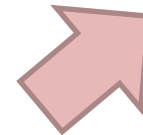
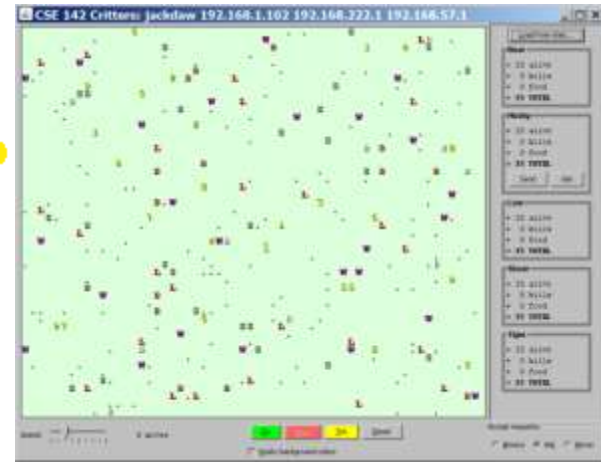
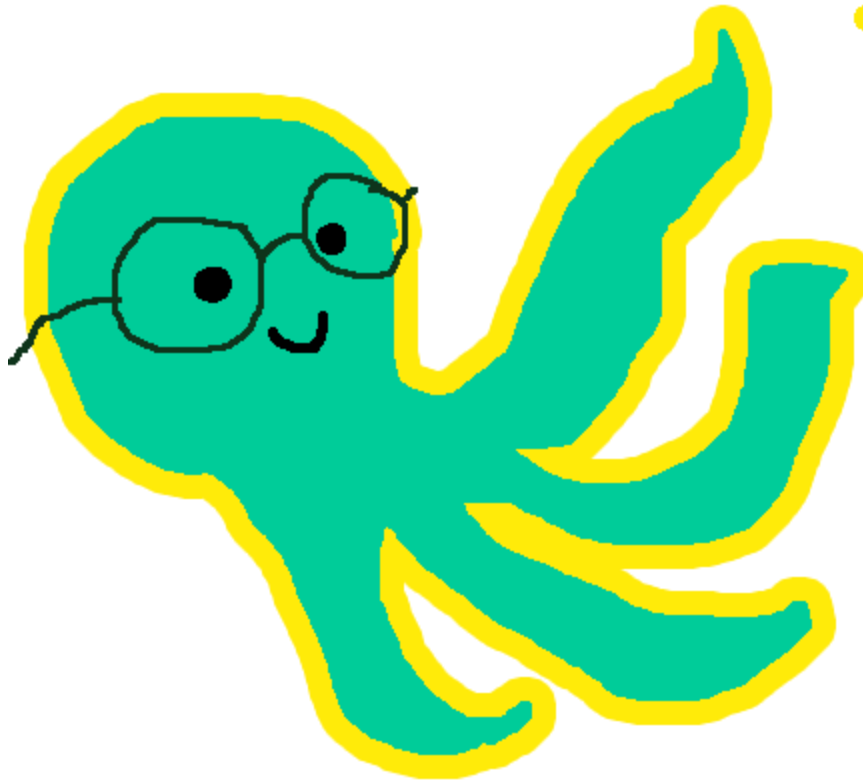
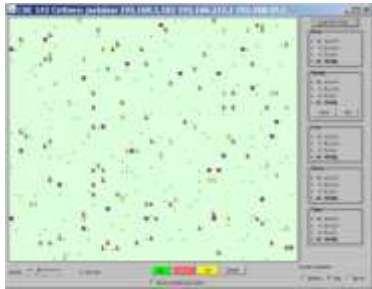
---

- Game Master also runs each “move” of the game
  - Every Critter’s “move” is executed during a **tick**
  - Game = Sequence of ticks
- Moves the critters, runs battles, figures out who’s napping/mating/eating, etc
- Must prompt each Critter:
  - Tell me what you look like (`toString()`, `getColor()`),  
Tell me where you want to go to next (`getMove()`)
  - If on a food, calls `eat()`
  - If in a fight, calls `fight()`

What's your next move?

**I**





# Programming a Critter

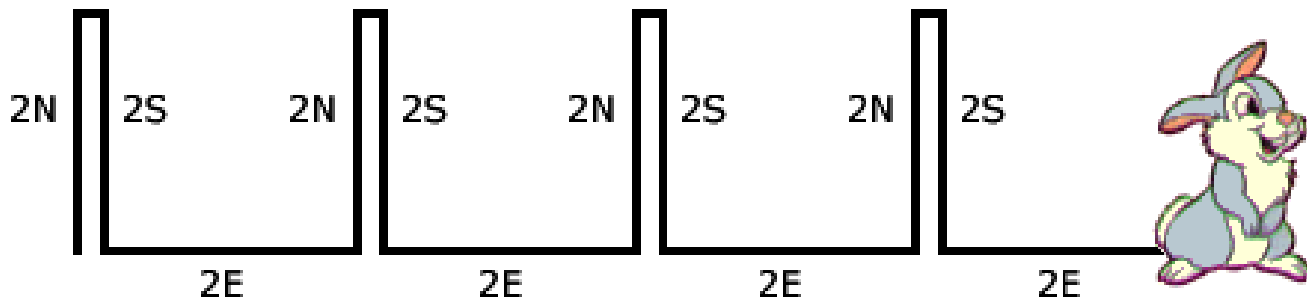
---

- Our job is to program `Critters`: we decide how a Critter **behaves** when `CritterMain` calls its methods
  - How it moves, how it eats, etc.
- Behavior often dependent on things the Critter has to “remember” so we also give `Critters` **state**



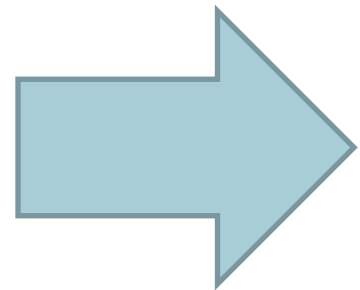
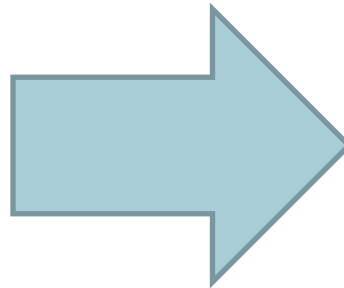
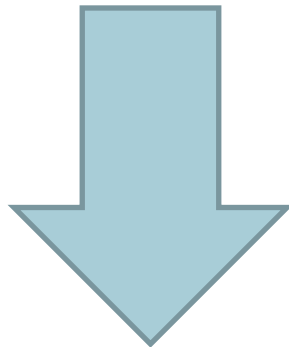
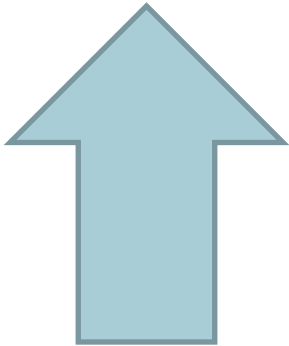
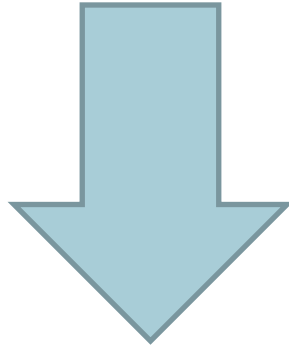
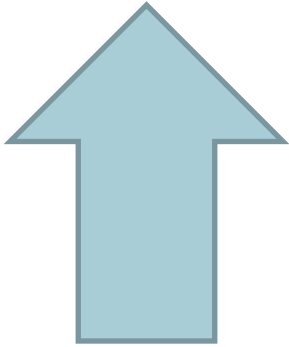
# A Simple Critter: Rabbit

Method	Behavior
Constructor	<code>public Rabbit()</code>
Color	<code>dark gray (Color.DARK_GRAY)</code>
Eating	alternates between <code>true</code> and <code>false</code> ( <code>true, false, true, ...</code> )
Fighting	if opponent is a <code>Lion</code> , then <code>scratch</code> ; otherwise, <code>roar</code>
Movement	<code>2 N, 2 S, 2 E, repeat</code>
<code>toString</code>	<code>"v"</code>



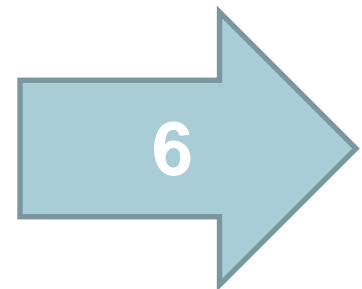
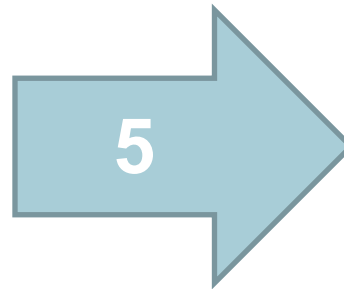
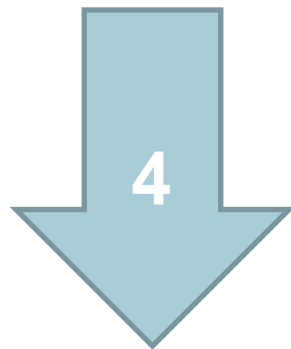
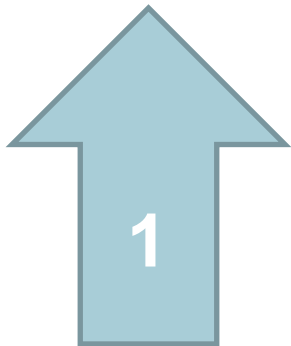
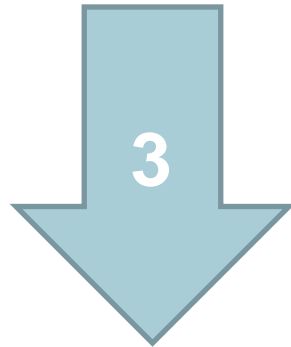
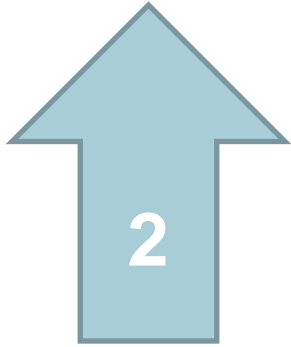
# Rabbit movement

---



# Rabbit movement

---



# A complex Critter: Snake

Method	Behavior
Constructor	<code>public Snake()</code>
Color	<i>(red=20, green=50, blue=128)</i>
Eating	Never eats
Fighting	Randomly choose to roar or pounce
Movement	<b>1 E, 1 S; 2 W, 1 S; 3 E, 1 S; 4 W, 1 S; 5 E, ...</b>
toString	Always returns "S"

