

Lecture 15

Code reading and writing practice.

For a narrated walkthrough, you are encouraged to follow along with the video posted for this portion of lecture.

Code Reading

- Read through the Model and Cell classes in Lecture 15
- Closely read and think through the Model class' methods:
 1. reset
 2. enforceBounds
 3. checkCollisions
- Respond on Gradescope with brief English explanations of what each of these methods is doing before continuing on.

Code Writing

1. Implement the Cell class' tick method as described in the comments of Part 1.
2. Implement the Model class' tick method as described in its comments. After doing so and saving both Cell and Model you should be able to refresh your browser to see the cells moving around.
3. Back in the Cell class, implement the isSick method per its description. After successfully implementing and saving, you should see a single red cell in the simulation upon refresh.
4. Finally, implement the logic of the Cell class' collideWith method per its comments. Upon successful completion, saving, and refresh you should see cells begin to infect one another as they touch in the simulation.

```
tick(): void {  
    // Part #1: Reassign this object's location property the result  
    // of adding this object's location property with its direction  
    // property. Hint, look at the Point class to see its addWith method.  
    this.location = this.location.addWith(this.direction);  
}
```

```
tick(): void {  
    // TODO  
    // #1) Loop through population and call each cell's `tick` method  
    // #2) Call the model's enforceBounds method on all cells  
    // #3) After the loop, call the model's checkCollisions method  
    for (let i = 0; i < this.population.length; i++) {  
        let cell = this.population[i];  
        cell.tick();  
        this.enforceBounds(cell);  
    }  
    this.checkCollisions();  
}
```

```
// Part #2: Define a method named `isSick` with no parameters
// that returns a boolean. It should return true if this object's
// sickness property is greater than 0 and false otherwise.
isSick(): boolean {
    return this.sickness > 0;
}
```

```
// Part #3: Complete the method `collideWith`.
// Its implementation should be such that if this cell is sick
// and the other isn't, then infect the other cell by calling its
// infect method. Be sure the logic is true in the inverse, as well.
collideWith(other: Cell): void {
    if (this.isSick() && !other.isSick()) {
        other.getsSick();
    } else if (other.isSick() && !this.isSick()) {
        this.getsSick();
    }
}
```