Week 2: Pimp My Karel

An introduction to SuperKarel, loops and ifs

Jack Beasley, Colleen Dai and George Younger

February 15, 2017

M-A Hackers

Decomposition

Redundant Karel Repeats Things Redundantly

- This seems repetitive!
- Programmers don't like to repeat themselves and follow a principal called DRY or (Don't Repeat Yourself).
- How could we DRY this code out?

```
public class CollectNewspaperKarel extends Karel {

  public void run() {
    turnLeft();
    turnLeft();
    turnLeft();
  }

}
```

Your first function

We can write a function to turn right!

```
public class CollectNewspaperKarel extends Karel {

  public void run() {
    turnRight();
  }

  public void turnRight() {
    turnLeft();
    turnLeft();
    turnLeft();
  }

}
```

Unlocking Karel's Superpowers

SuperKarel

- Karel is kinda boring...
- Karel can only move(), turnLeft(), pickBeeper() and putBeeper()
- We have to write functions to do EVERYTHING else
- What if souped up Karel?

```
// To power up Karel, this
public void CollectNewspaperKarel extends Karel {
  // Your code here
}

// becomes this:
public void CollectNewspaperKarel extends SuperKarel {
  // Your code here
}
```

What powers did we unlock?

- `turnRight()`!!!!!
- `turnAround()`
- And others, check the Karel reference card for more info

if Statements

Logic

- Any good robot can make decisions for itself
- Karel is no exception!
- Karel can use if statements to decide whether or not to do things

```
if (frontIsClear()) {
  move();
}
```

Karel Sensors

Karel can sense the environment around her and make decisions based on what she sees.

Karel can sense:

- `beepersPresent();`
- `noBeepersPresent();`
- `facingNorth();`
- and many more! Visit a reference card near you for more details

while Loops

While this is true do that

What does this do?

```
while (frontIsClear()) {
  move();
}
```

for Loops

Looping exactly

What if we want to do something exactly 3 times?

```
public class CollectNewspaperKarel extends Karel {

  public void run() {
    turnRight();
  }

  public void turnRight() {
    turnLeft();
    turnLeft();
    turnLeft();
  }

}
```

How would we do this with a while?

```
public class CollectNewspaperKarel extends Karel {

  public void run() {
    turnRight();
  }

  public void turnRight() {
    int i = 0;
    while (i < 3) {
      turnLeft();
      i++;
    }
  }

}
```

Enter the for loop!

```
public class CollectNewspaperKarel extends Karel {

  public void run() {
    turnRight();
  }

  public void turnRight() {
    for (int i = 0; i < 3; i++) {
      turnLeft();
    }
  }

}
```