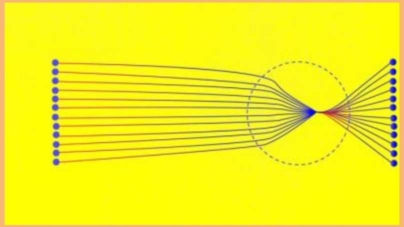
Welcome to Section!!!

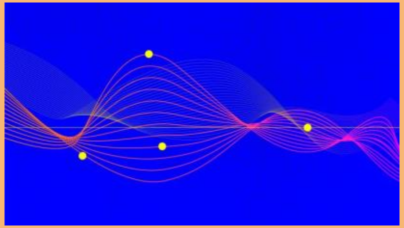
Week One

Introductions

A Little About Francois



Artificial Intelligence: Principles and Techniques
XCS221



Natural Language Processing with Deep Learning
XCS224N

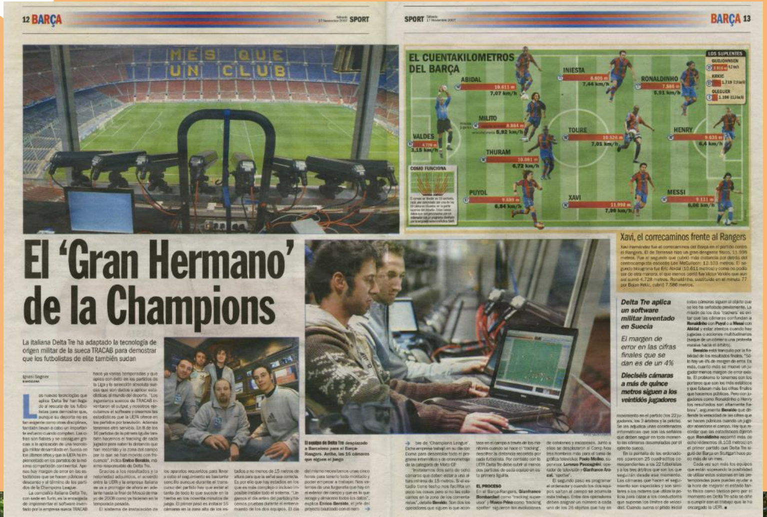
★ My name is François. I'll be your Section Leader for Code in Place

★ I am also a CF for AI classes XCS224N and XCS221 at Stanford.

★ In my free time I like to ride my Elliptigo.

★ I enjoy spending time with my kids and teaching them coding.

A Little About Gian



12 BARÇA · Sábado 17 Noviembre 2007 · SPORT

SPORT · Sábado 17 Noviembre 2007 · BARÇA 13

MES QUE UN CLUB
EL CUENTAKILOMETROS DEL BARÇA
ABIDAL 10.611 m 7,07 km/h
INIESTA 8.809 m 7,44 km/h
RONALDINHO 7.586 m 5,91 km/h
MILITO 8.884 m 5,92 km/h
VALDES 4.728 m 3,15 km/h
TOURE 10.526 m 7,01 km/h
HENRY 9.635 m 6,4 km/h
THURAM 10.081 m 6,72 km/h
PUYOL 9.689 m 6,84 km/h
XAVI 11.998 m 7,99 km/h
MESSI 9.131 m 6,06 km/h
LOS SUPLENTES
GUDJOHNSEN
KRKIC 1.719
OLEGUER 1.106
COMO FUNCIONA

El 'Gran Hermano' de la Champions

La italiana Delta Tre ha adaptado la tecnología de origen militar de la sueca TRACAB para demostrar que los futbolistas de elite también sudan

Ignasi Sagnier
BARCELONA

El equipo de Delta Tre desplazado a Barcelona para el Barça-Rangers. Arriba, las 16 cámaras que siguen el juego

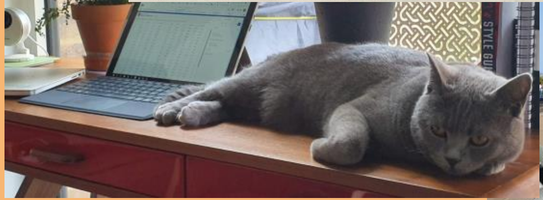
Xavi, el correcaminos frente al Rangers

Xavi Hernández fue el correcaminos del Barça en el partido contra el Rangers. El de Terrassa hizo un gran desgaste físico, 11.998 metros. Fue el segundo que cubrió más distancia por detrás del centrocampista escocés Lee McCulloch: 12.103 metros. El segundo blaugrana fue Eric Abidal (10.611 metros) y como no podía ser de otra manera, el que menos corrió fue Víctor Valdés que aun así sumó 4.728 metros. Ronaldinho, sustituido en el minuto 77 por Bojan Krkic, cubrió 7.586 metros.

Delta Tre aplica un software militar inventado en Suecia

El margen de error en las cifras finales que se dan es de un 4%

Dieciséis cámaras a más de quince metros siguen a los veintidós jugadores



- ★ My name is Gian. I'll be your tutor, and everyone here will be your section-mates!
- ★ I am working in Data Science, leading the Business Intelligence department for a consultancy based in London.
- ★ In my free time I like to cycle, exploring new places/ things and...coding, of course!
- ★ I have a beautiful cat called Martino, who likes to quality control my work!

What About You All?

Go ahead and share:

1. Your name
2. What’s one awesome thing you want to program Karel to do?
3. Why are you excited about learning Python and what are your goals?

Why are we here?

- Live Section != lecture
- **This is your time. We** are here to help answer your questions about section material.
- Finishing problem isn't important.
- Building a community of learners.

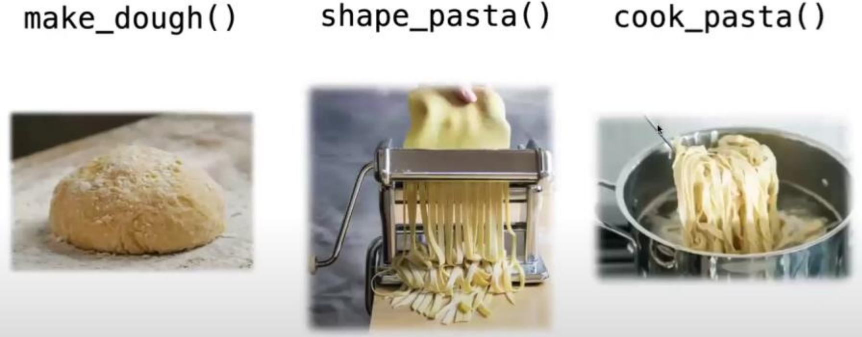
Lecture Review

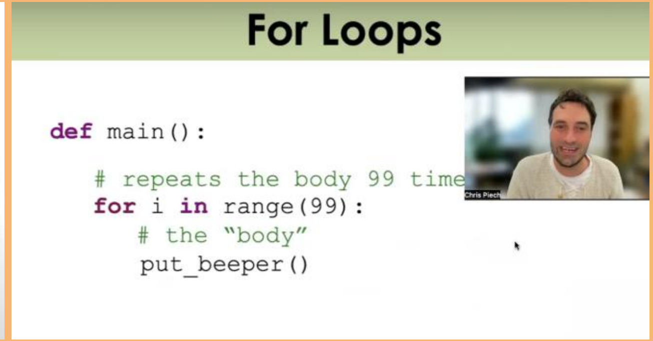
What We've Learned

Before we get into our sample problem for today, let's review a bit. We've learned:

- The basics about Karel, the magnificent and wonderful robot
- Functions, a way of breaking down big problems into smaller chunks
- Control Flow, loops and conditional statements which guide our programs

This is a **LOT** of content, especially if you are newer to Computer Science!




For Loops

```
def main():

    # repeats the body 99 time
    for i in range(99):
        # the "body"
        put_beeper()
```

Let's review and refresh these concepts a bit!

Decomposition: Functions

"Functions should do one thing. They should do it well. They should do it only." *(Robert C Martin, Clean Code)*

”

Define your function like this:

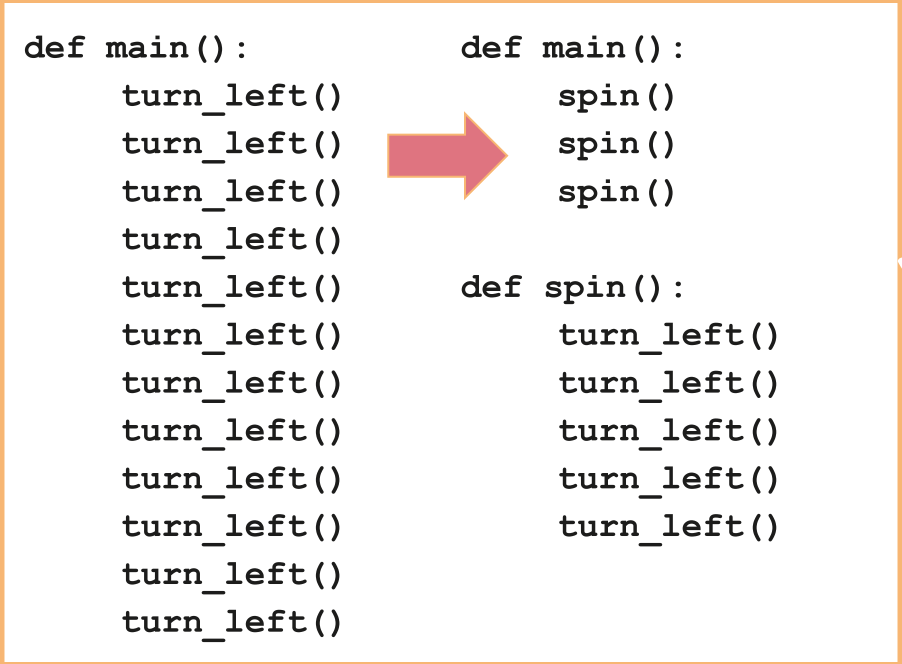
```
    def function_name():
        <write code here>
```

And call it like this:

```
    function_name()
```

Remember:

- More of an art than a science.
- If you repeat things, take a step back.



```
def main():
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
```

```
def main():
    spin()
    spin()
    spin()

def spin():
    turn_left()
    turn_left()
    turn_left()
    turn_left()
    turn_left()
```

Control Flow Overview

Let’s look at some of the more difficult conceptual concepts that we’ve learned recently.

For-loop (definite loop):

- Performs some block of code, a specific amount of times.

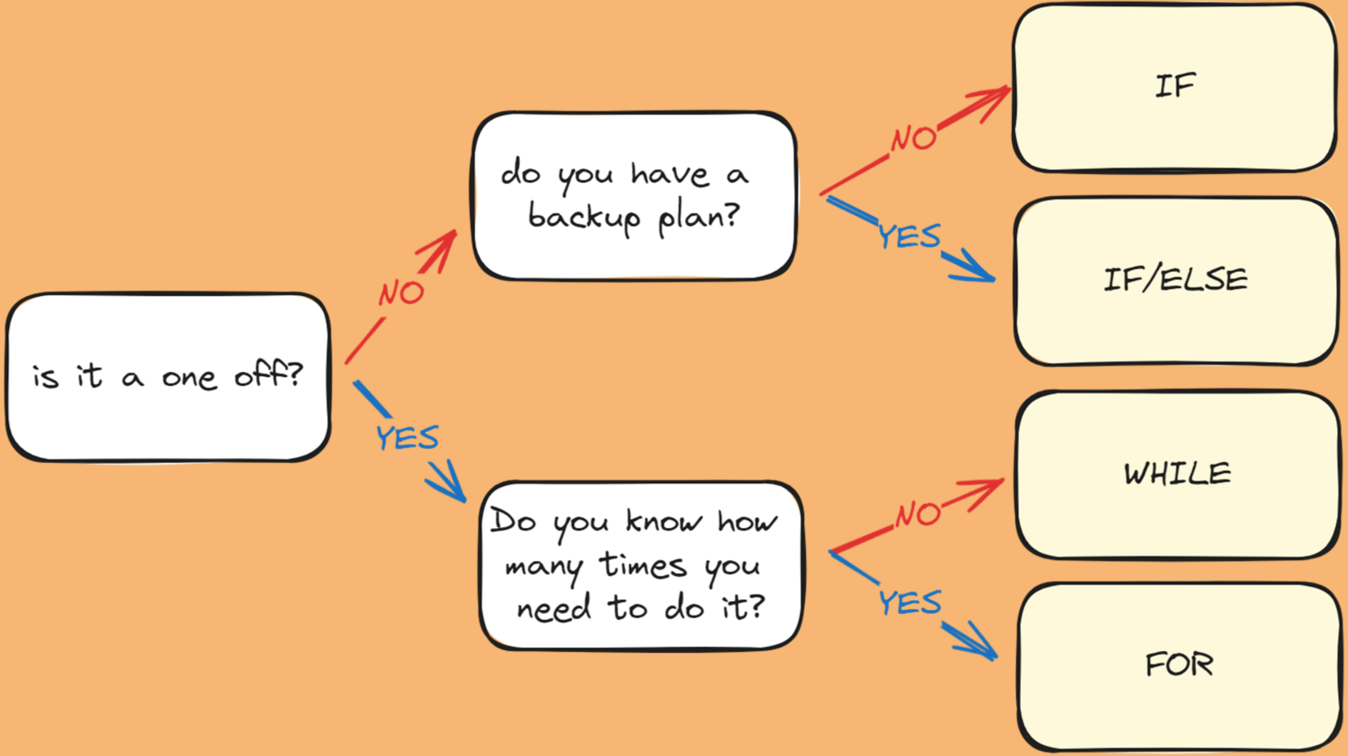
While-loop (indefinite loop):

- Continuously perform a block of code until what’s being tested is evaluated to false.

If-statement:

- Tests for truth. Performs a block of code only when evaluated to true.

Control Flow Overview


is it a one off?
NO
YES
do you have a backup plan?
NO
YES
Do you know how many times you need to do it?
NO
YES
IF
IF/ELSE
WHILE
FOR

If-Statement

An example if-statement that you may see and use with Karel:

```
def safe_move():
        if front_is_clear():
                move()
```

An if-statement runs code inside of it when the associated statement is evaluated to true. We will get into more complex statements later on in the course to add much more flexibility to our if-statements!

While-Loop

An example while-loop that you may see and use with Karel:

```
def move_to_wall():
       while front_is_clear():
              move()
```

This loop is also called an *indefinite loop* because it will run until the associated condition becomes false, which may be never! Who knows? You will, hopefully. Be careful so you don't get stuck in an infinite loop while using this!

For-Loop

An example for-loop that you may see and use with Karel:

```
def turn_right():
       for i in range(3):
              turn_left()
```

This loop is also called a *definite loop* because we know where it ends, when i reaches 3. (Be careful to remember that i begins at 0 when we start our loop!!!)

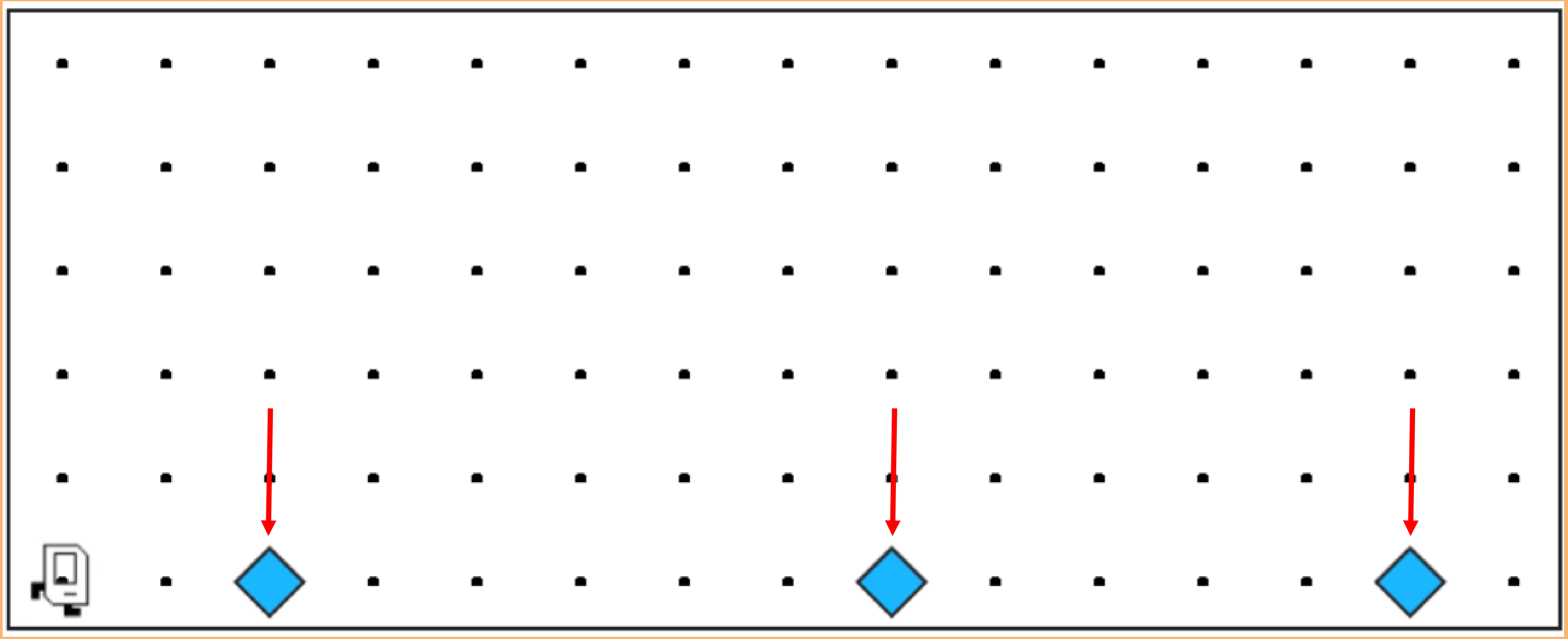
Any Questions?

Section Problem: Hospital Karel

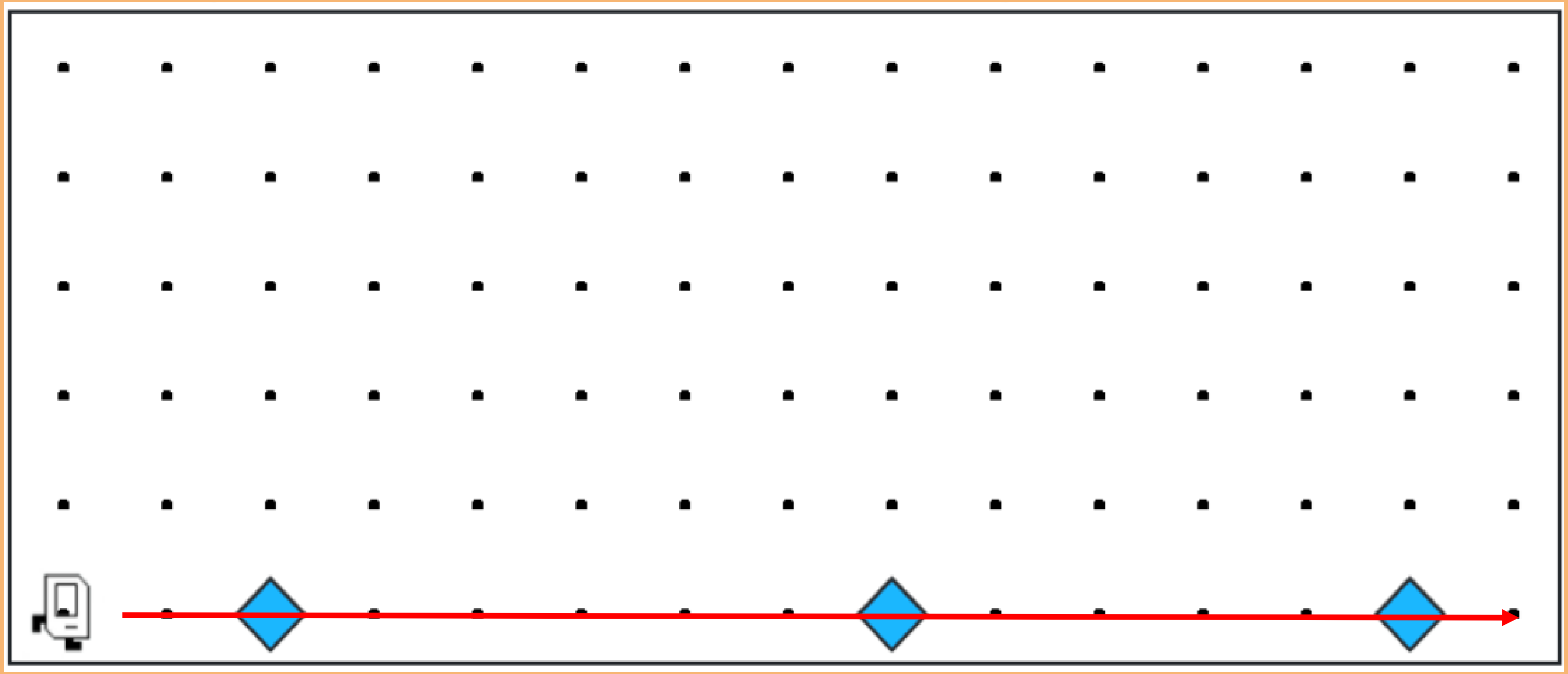
Setting Context

Countries around the world are dispatching hospital-building robots to make sure anyone who gets sick can be treated. They have decided to enlist Karel robots. Your job is to program those robots.

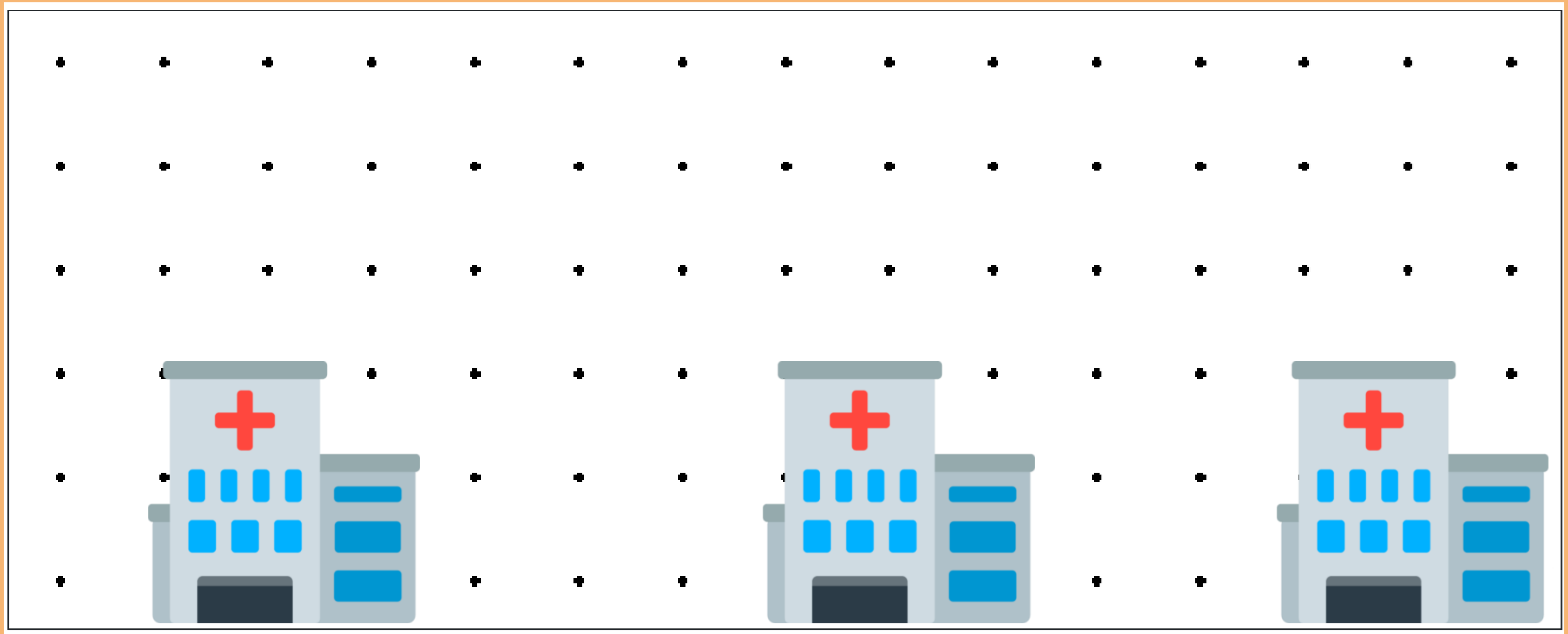
Each beeper in the figure represents a pile of supplies.



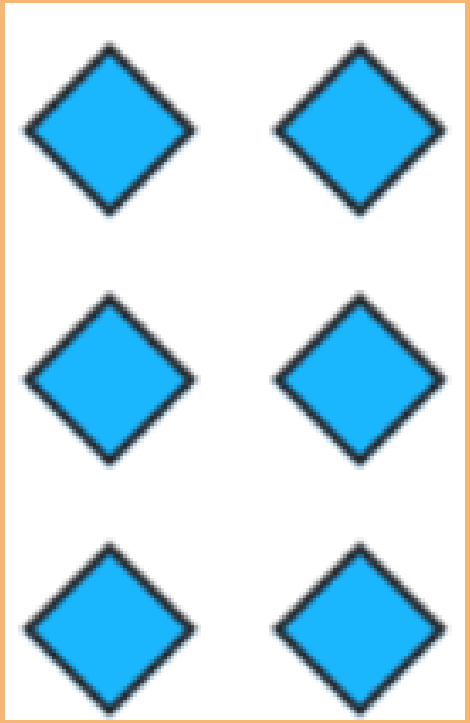
Karel's job is to walk along the row and build a new hospital in the places marked by each beeper.



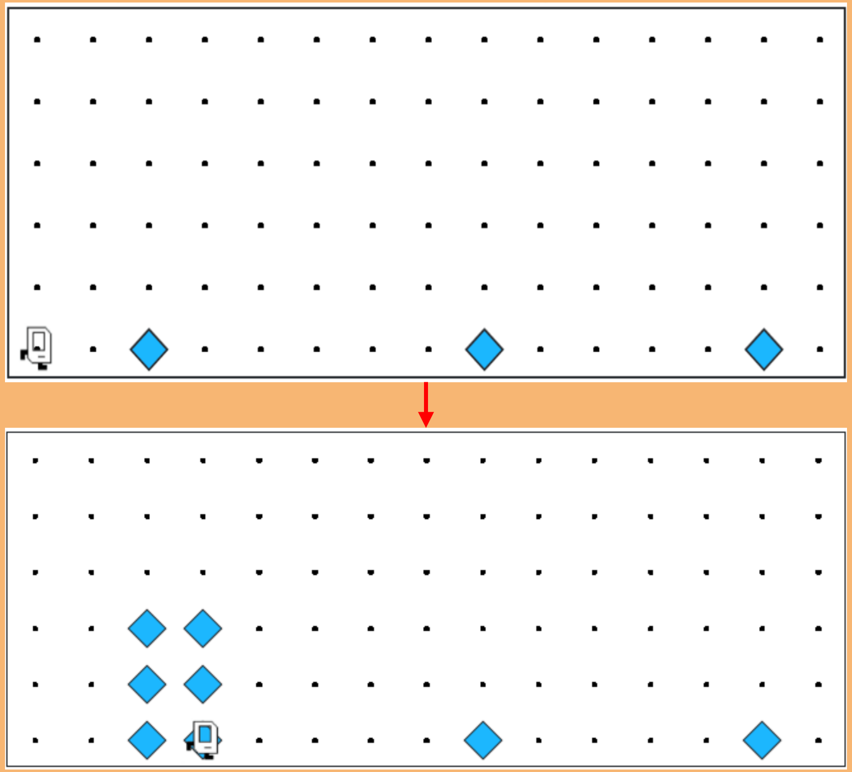
Karel’s job is to walk along the row and build a new hospital in the places marked by each beeper.



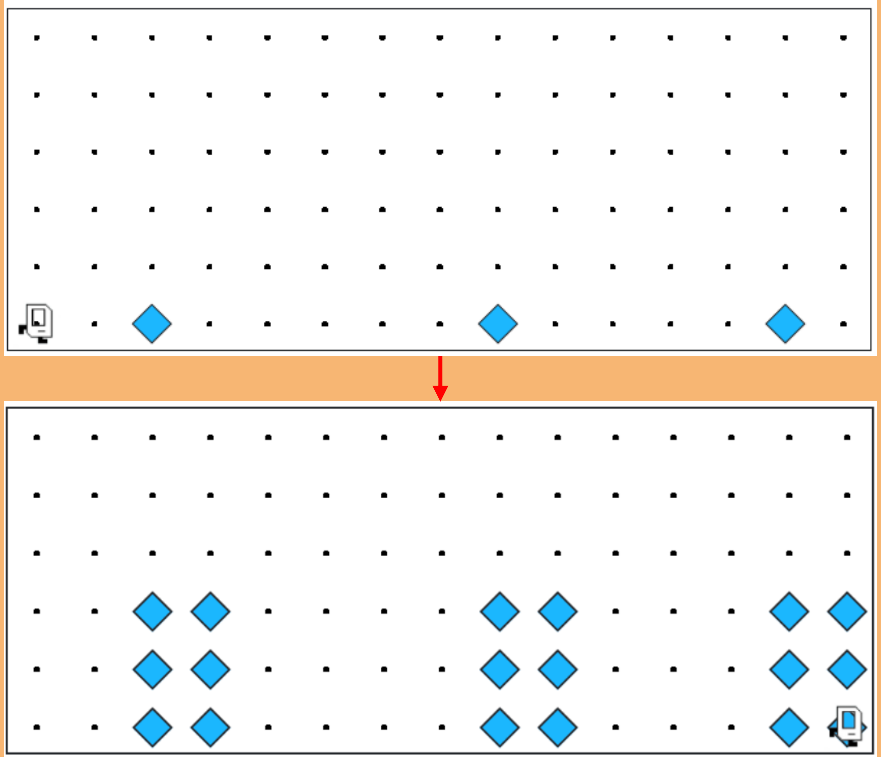
Hospitals look like this: a 3x2 rectangle of beepers!



The new hospital should have their corner at the point at which the pile of supplies was left.



At the end of the run, Karel should be at the end of the row having created a set of hospitals. For the initial conditions shown, the result would look like this:



Notes to Keep in Mind

- Karel starts facing east at (1, 1) with an infinite number of beepers in its beeper bag.
- The beepers indicating the positions at which hospitals should be built will be spaced so that there is room to build the hospitals without overlapping or hitting walls.
- There will be no supplies left on the last column.
- Karel should not run into a wall if it builds a hospital that extends into that final corner.

Questions Before We Begin?

Let's get to work!