Today

- Finish activity recognition example
- Return to binary scanner: using loops to simplify and generalize our code
- while vs do-while

Short Questions?

Quiz

Recognizing Crawling Gestures

Crawling assistant robot:

- If the infant's left hand is far in front and low, then trigger a forward motion
- If the infant's left hand is far to the side and low, then trigger a right turn

Write a program that recognizes whether we are in one of these situations and "triggers" and appropriate robot response

While Loops

```
while(<CONDITION>) {
    <STATEMENTS>
}
```

- If <CONDITION> is true:
  - Execute <STATEMENTS>
  - Repeat

Back to our Binary to Decimal Converter

Problem:

- Prompt the user to enter a binary number of ***arbitrary*** length
- When '\n' is received, stop reading the binary number and report the decimal equivalent
  - The book refers to this last character as a "sentinel" – it is a cue to the code that it is done

Considerations

- Can only receive one of three different characters: ‘0’, ‘1’ and ‘\n’
- When we read a number from left to right and can only see part of the number:
  - What is our best guess for the number so far?
  - How do we handle adding one more digit on the right hand side?

For now: only use a while loop (not a do-while)

Do-While Loops

```
do{
    <STATEMENTS>
} while(<CONDITION>)
```

- Execute <STATEMENTS>
- If <CONDITION> is true, then repeat

Do-While Loops

- How do do-while loops simplify our binary-to-decimal converter?

Summary

- while() and do-while() loops allow us to repeatedly execute a group of statements
- Make sure that the <CONDITION> is false at some point in time (or else, your program will continue forever)
- Often use a counter of some form to determine when to stop (but not always)
- Always check: your first case and your last case (this will cover a large number of bugs)

Wrap Up

Coming up:

- HW 2: due Friday
- HW 3: due next Wednesday
- Exam 1: Sept 15

Next time:

- Methods