

# Research Directions in Block-Based Programming Languages

Franklyn Turbak  
Computer Science Department  
Wellesley College

Science Center Summer Research Seminar  
Tuesday, June 1, 2010

# Programming Language Issues

- o **Syntax** = form of programs
- o **Semantics** = meaning of programs
- o **Pragmatics** = practical aspects of programs

# Example: Absolute Value Function

**Logo:** `to abs :n ifelse :n < 0 [output (0 - :n)] [output :n] end`

**Javascript:** `function abs (n) {if (n < 0) return -n; else return n;}`

**Java:** `public static int abs (int n) {if (n < 0) return -n; else return n;}`

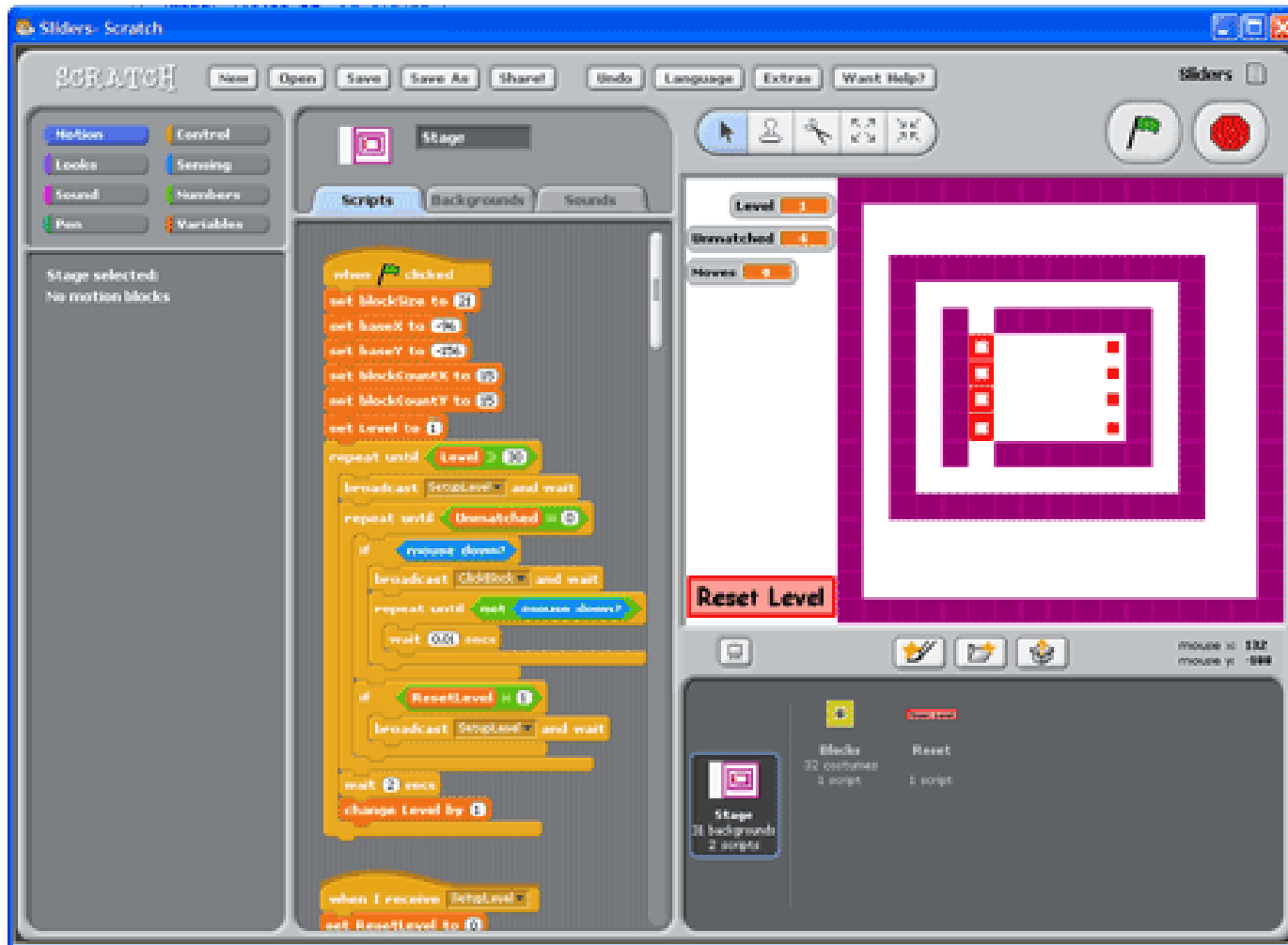
**Python:**

```
def abs(n):  
    if n < 0:  
        return -n  
    else:  
        return n
```

**Scheme:** `(define abs (lambda (n) (if (< n 0) (- n) n)))`

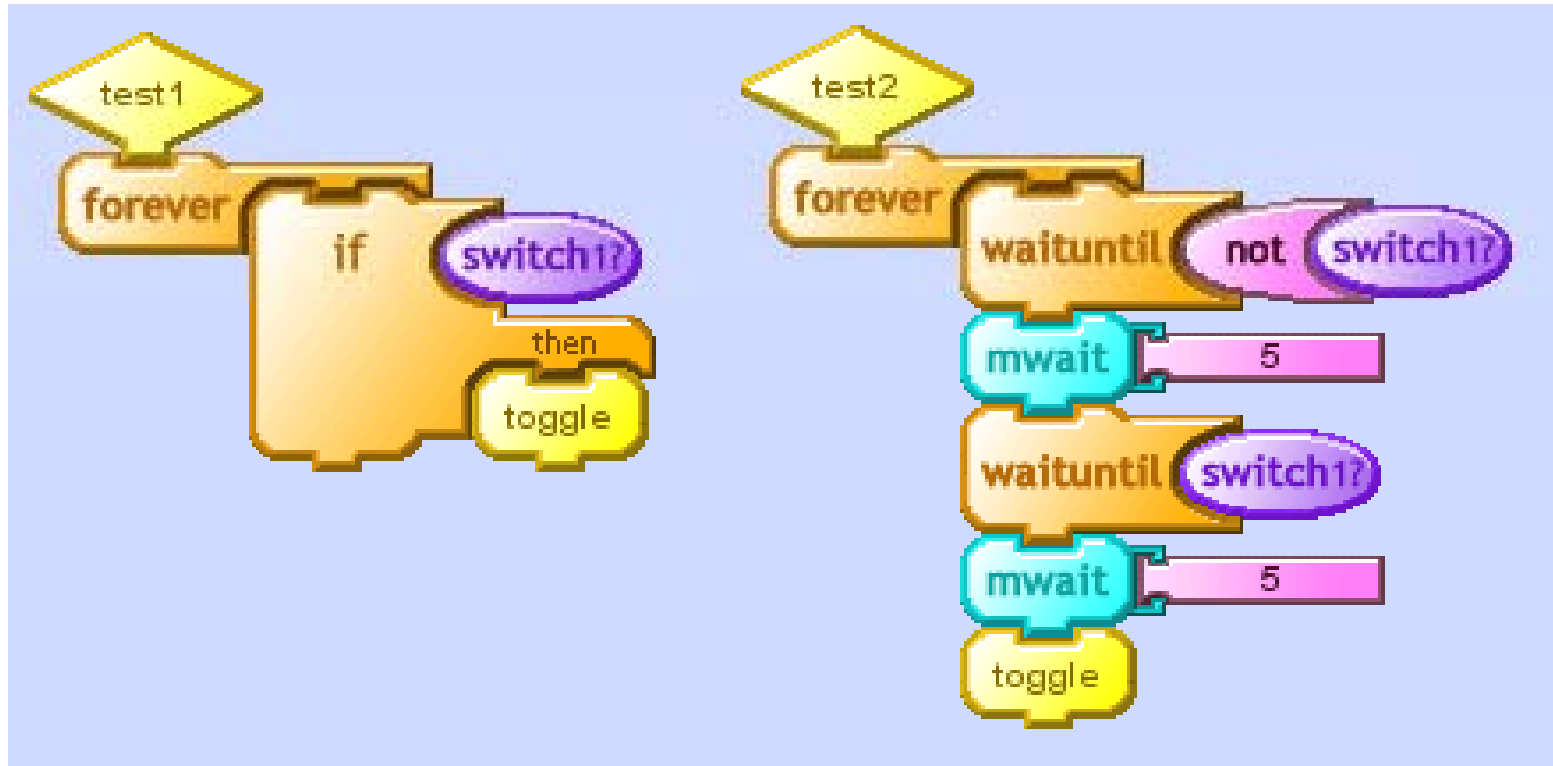
**PostScript:** `/abs {dup 0 lt {0 swap sub} if} def`

# A Popular Block-Based Language: Scratch

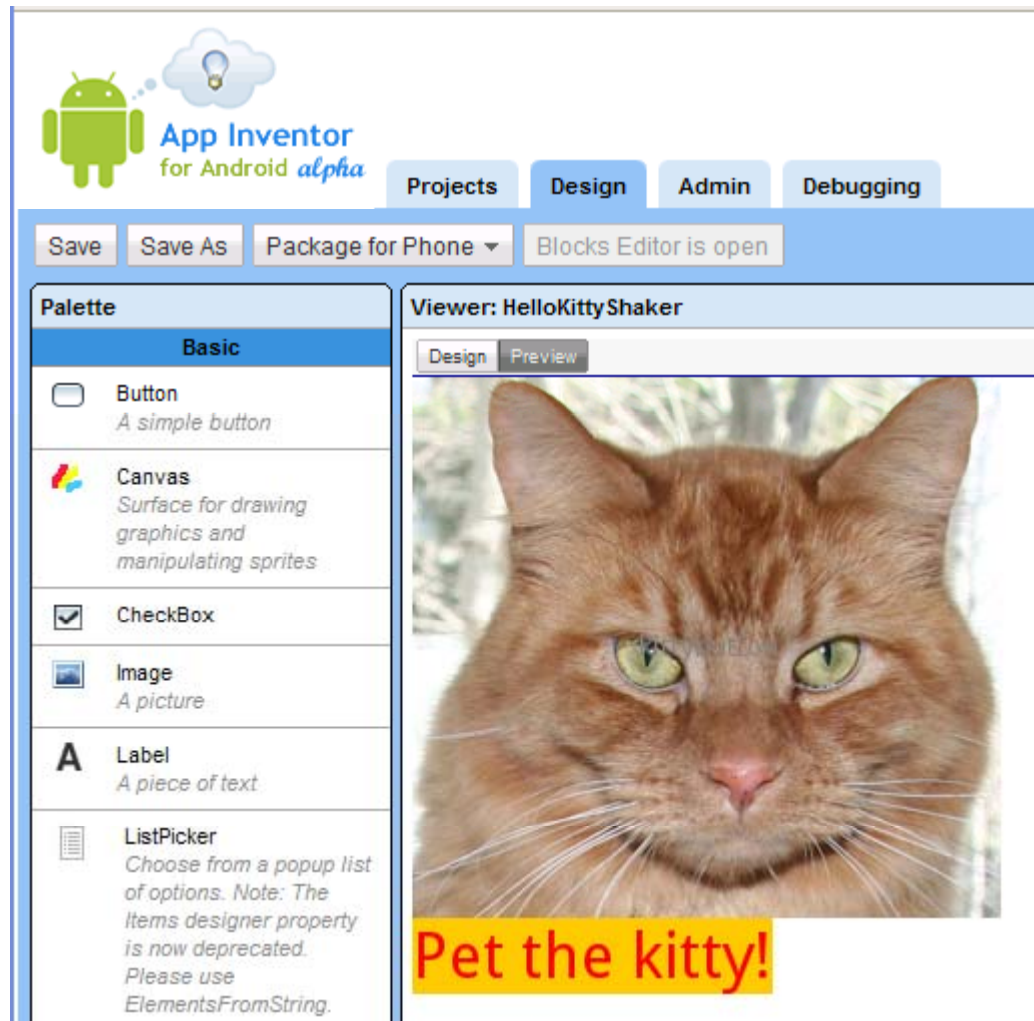


<http://scratch.mit.edu/>

# Another Block-Based Language: PicoBlocks



# App Inventor For Android: Designer Window

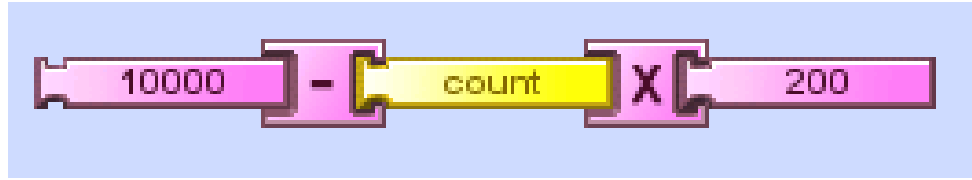


# App Inventor For Android: Blocks Window

The image shows the 'My Definitions' panel on the left and the 'Blocks Window' on the right. The 'My Definitions' panel contains five blue buttons: 'My Definitions', 'KittySound', 'KittyLabel', 'KittyButton', and 'AccelerometerSensor1'. The 'Blocks Window' contains two event-driven code blocks. The first block is triggered by 'KittyButton.Click' and contains two 'do' blocks: 'call KittySound.Play' and 'call KittySound.Vibrate' with a 'milliseconds' parameter set to 'number 123'. The second block is triggered by 'AccelerometerSensor1.Shaking' and contains one 'do' block: 'call KittySound.Play'.

# Operator Precedence Example

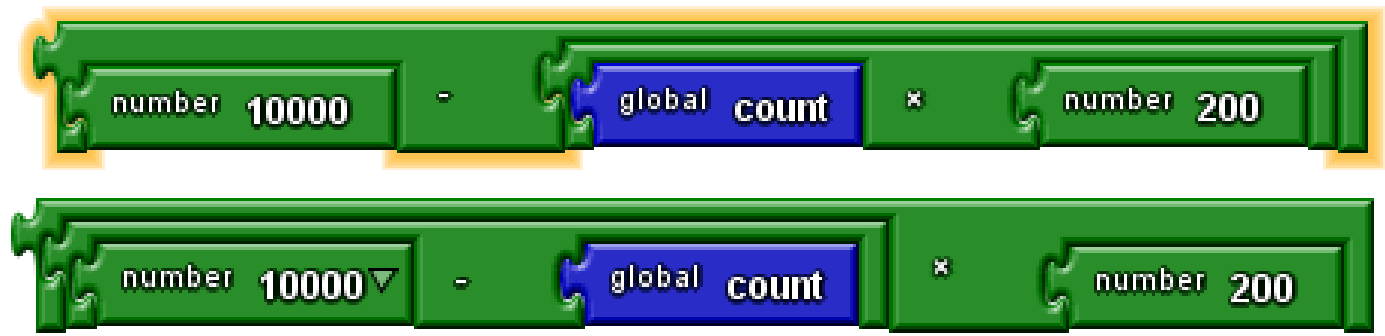
PicoBlocks



10000 - (count \* 200)  
(10000 - count) \* 200

Text (with parens)

AppInventor



PicoBlocks





# Absolute Value Function in PicoBlocks

Definition  
(Text Mode)

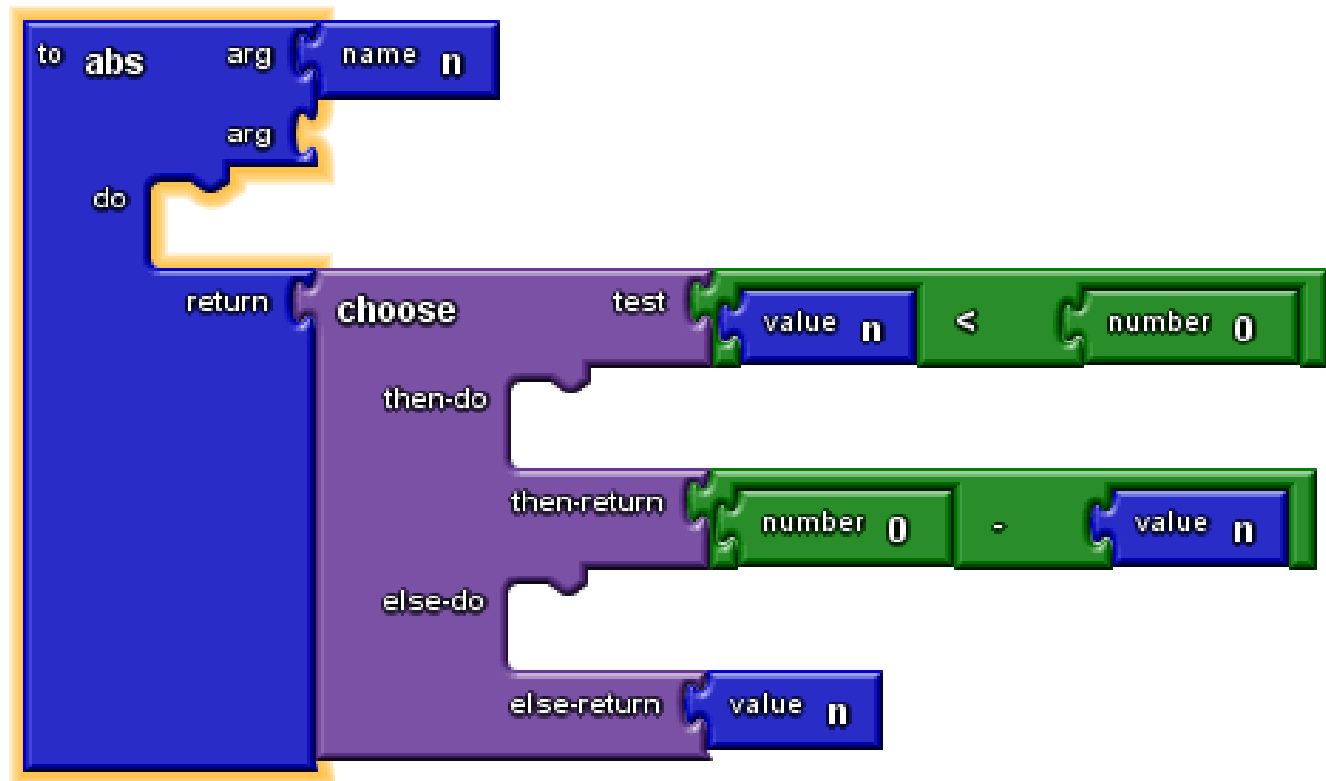
```
block abs :n
  ifelse :n < 0
    [output 0 - :n]
    [output :n]
  end
end
```

Invocation  
(Blocks Mode)



# Absolute Value Function in AppInventor

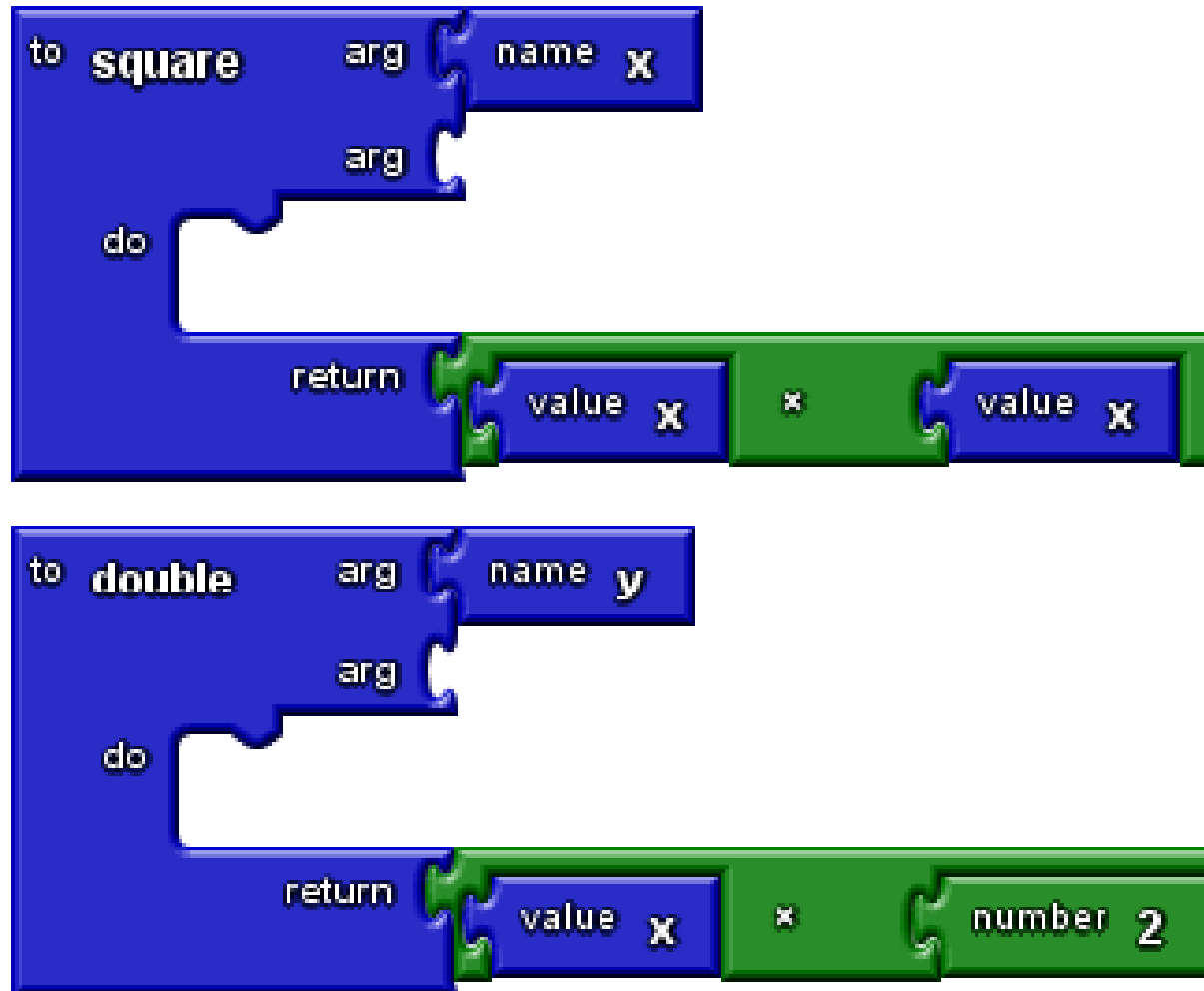
Definition



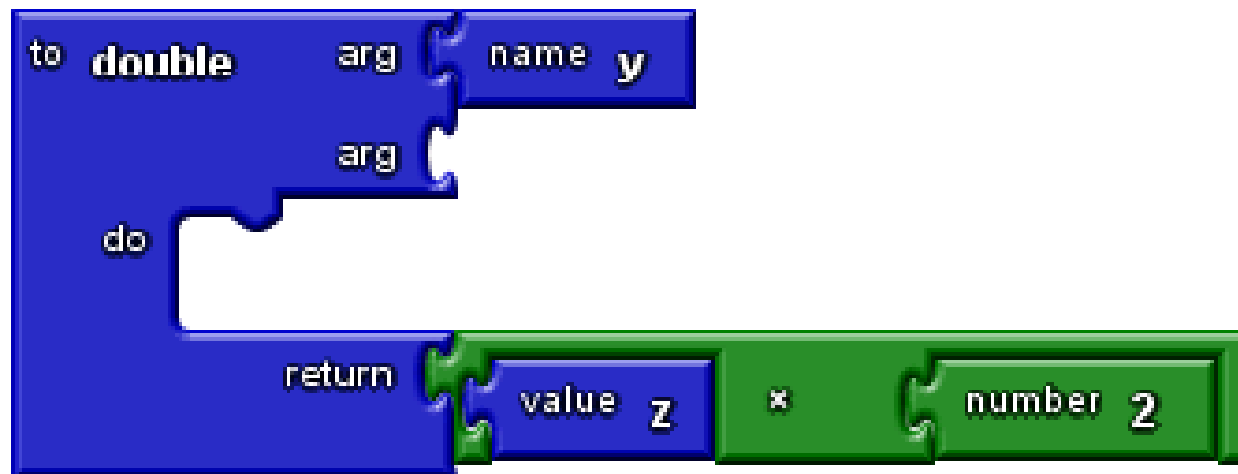
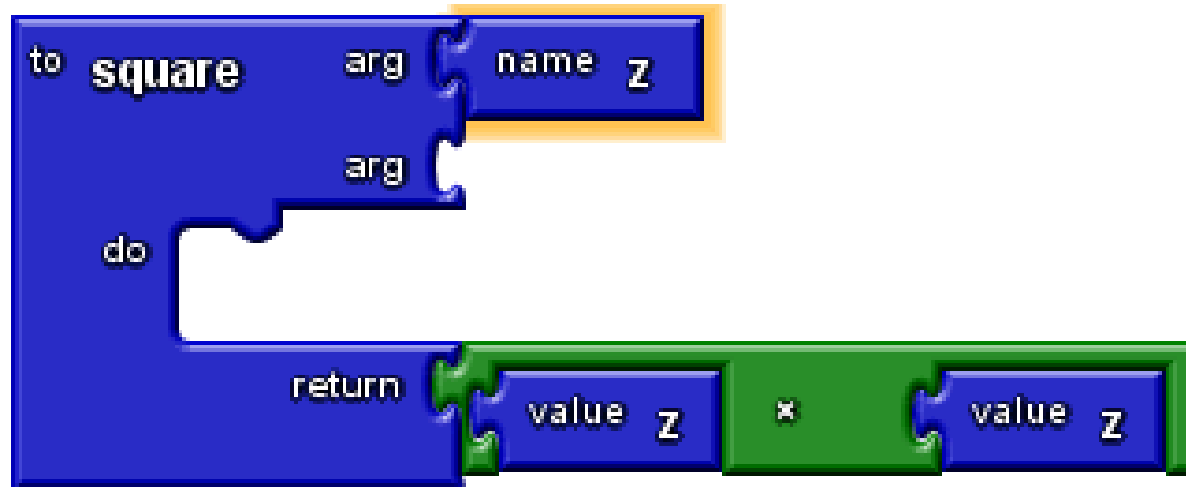
Invocation



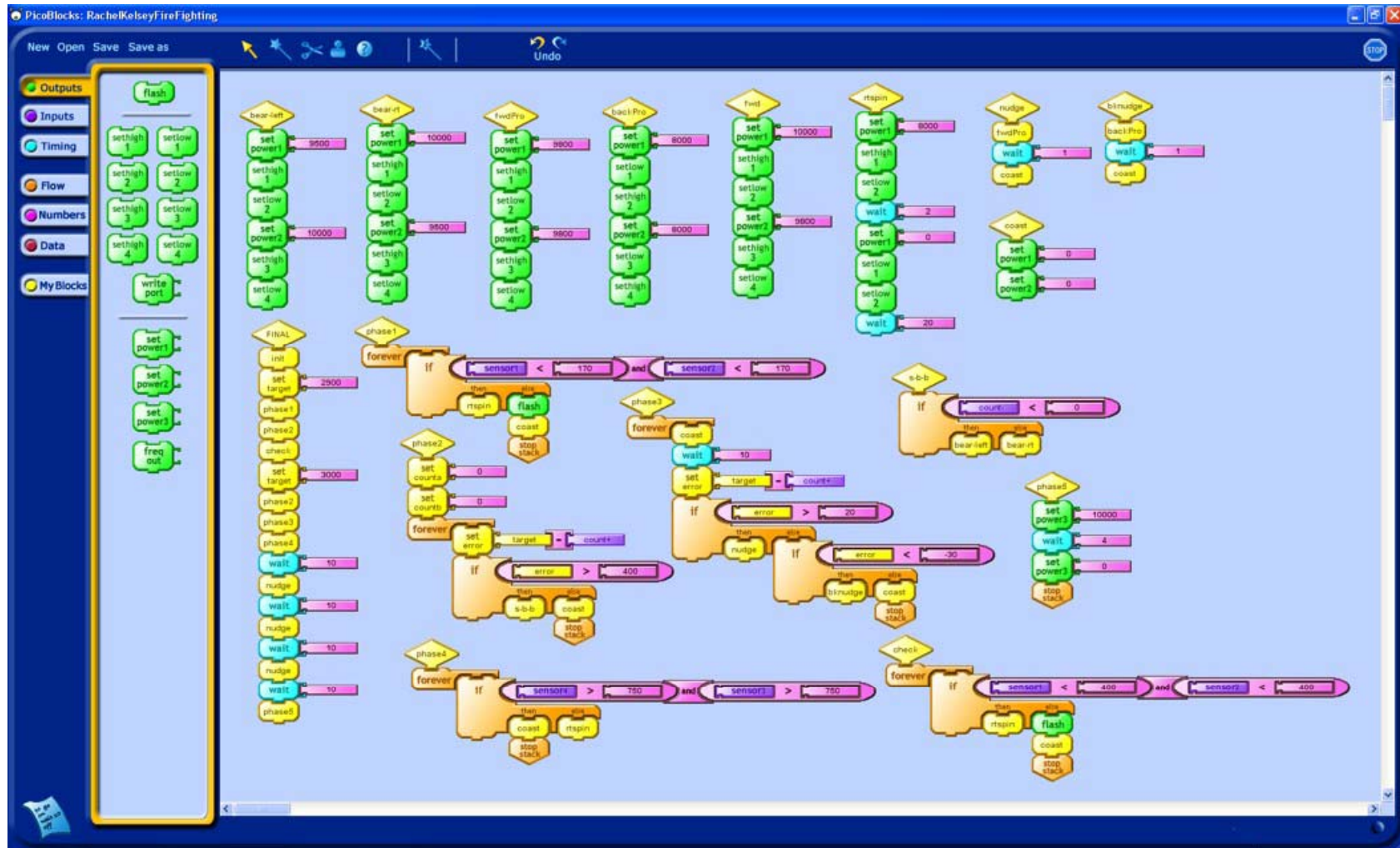
# Parameter Names in AppInventor



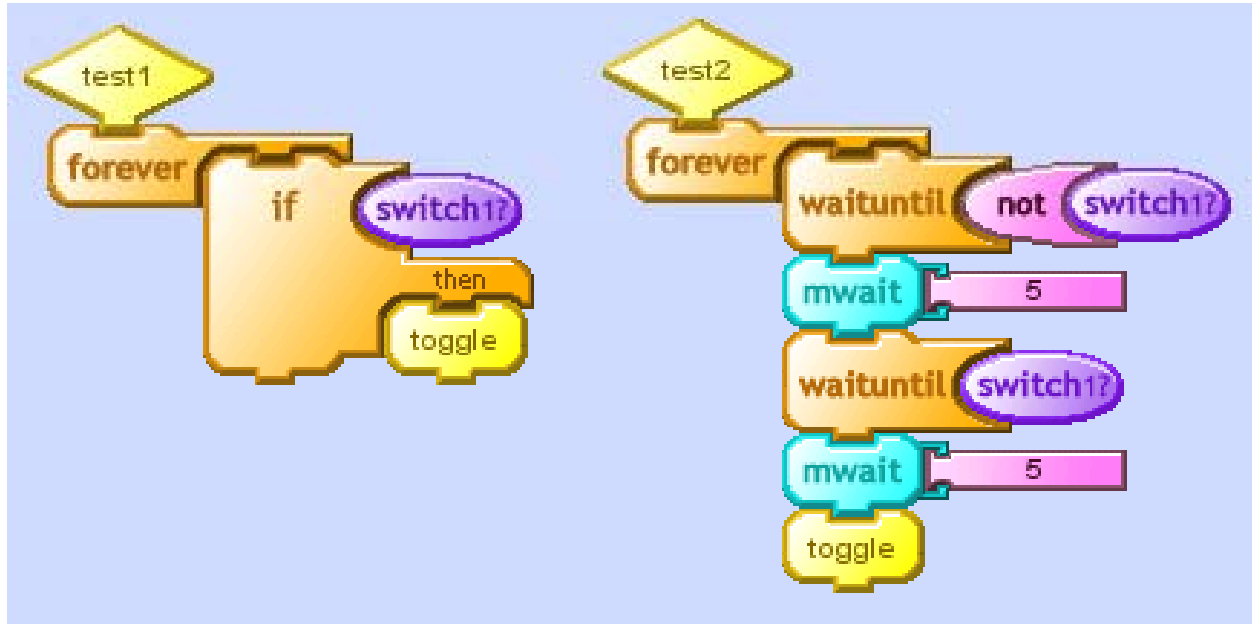
# Parameter Names in AppInventor, Part 2



# Some Problems with Blocks



# Blocks ↔ Text



```
block test1
  forever [
    if switch1?
      [toggle]
  ]
end
```

```
block test2
  forever [
    waituntil not switch1?
    mwait 5
    waituntil switch1?
    mwait 5
    toggle
  ]
end
```

# Goal: Tools for Constructing Blocks Languages

- JavaScriptBlocks

- PythonBlocks

- JavaBlocks

etc.