

# Lecture 22: For Loops

---

CS 51G  
Spring 2018  
Kim Bruce

# Test Programs

- Questions on Test Programs

# Towers of Hanoi

- 3 diamond-tipped needles
- 64 golden disks to move
  - Start on needle one with lower disks larger than upper
  - Move to needle three
  - But can't put big disk on smaller disk
  - Can use 2nd needle to help
  - How many moves?
  - <http://www.cs.pomona.edu/classes/cs051G/demos/Hanoi/Hanoi.grace>

# For loops

- Many while loops similar:

```
var stripeStart: Point := point
var stripeNum: Number := 1
while { stripeNum <= n } do {
  def redStripe: Graphic2D = filledRect.at (stripeStart)
                                size (width @ stripeHeight) on (canvas)
  redStripe.color := red
  stripeStart := stripeStart + (0@(2 * stripeHeight))
  stripeNum := stripeNum + 1
}
```

*Easy to forget some of pieces — want to automate it!*

# For loops

- Many while loops similar:

```
var stripeStart: Point := point
for (1..n) do { stripeNum: Number →
  def redStripe:Graphic2D = filledRect.at (stripeStart)
                                size (width @ stripeHeight) on (canvas)
  redStripe.color := red
  stripeStart := stripeStart + (0@(2 * stripeHeight))
}
```

*Three lines compressed to one!*

*Simpler if do loop for fixed number of times*

*(1..n) is called a “range”*

# Other Examples

- Knitting again:

- <http://www.cs.pomona.edu/classes/cs051G/demos/KnittingFor/KnittingFor.grace>

- Interest:

- <http://www.cs.pomona.edu/classes/cs051G/demos/InterestingFor/InterestingFor.grace>

- Called a range: (m .. n)

- (2 .. 9).reversed gives range in reverse order

# Other Loops

- repeat (n) times { body }
  - Could have used in flag & knitting
- do { body } while { cond }
  - Test after execute body of loop

# Lists

- Another way to hold collections of objects
  - More restricted than recursion as kept as collection indexed by number, e.g., has first, second, third, etc.
  - Often easier to use than recursion
- Must say what kind of element held in list:
  - `var scores: List[[Number]]`
  - `var names: List [[String]]`
  - `var pictures: List [[Graphic2D]]`



# Creating Lists

- Creating lists of strings:
  - `var start: List[String] := emptyList[String]`
  - `var vowels: List[String] := list[String] ["a", "e", "i", "o", "u"]`
- Can also create from ranges:
  - `var scores: List[Number] := list[Number] (1..100)`

# Accessing Elements

- Example:
  - `var vowels: List[String] := list[String] ["a", "e", "i", "o", "u"]`
  - `def first: String = vowels.at(1)`
  - `def last: String = vowels.at(5)`
  - Can use in for loops:
    - `for (vowels) do {letter: String ->  
 print (letter)  
}`

Questions?