

# Curing Recursion Aversion



University  
of Victoria

Katherine Gunion, Todd Milford, Ulrike Stege  
**Computer Science/Education**  
**University of Victoria**

Thanks To Rebeca Dunn-Krahn (Video)  
Terence Nathan (Volunteer)



# Problem

- Looking for interactive ways to teach recursion
- Demonstrate that the concept is not 'too hard' to learn
- Demonstrate given the correct abstraction that young students can learn

# Outline

- Details of Expectations of Understanding
- Participant and Teacher Demographics
- Classroom Dynamics
- Activities
- Sample Methods of Evaluation and Results
- Outcome and Future Directions

# What is recursion?

# What is recursion?

```
to showRecursion :num1 :num2
```

```
end
```

# What is recursion?



```
to showRecursion :num1 :num2
```

```
  print :num1  
  print :num2
```

```
end
```

# What is recursion?



```
to showRecursion :num1 :num2
```

```
  showRecursion :num1 - 1 :num2-1  
  print :num1  
  print :num2
```

```
end
```

# What is recursion?



```
to showRecursion :num1 :num2
  ifelse :num1 < 1[
    print 0
  ][
    showRecursion :num1 - 1 :num2-1
    print :num1
    print :num2
  ]
end
```



# What is recursion?



```
to showRecursion :num1 :num2
  ifelse :num1 < 1[
    print 0
  ] [
    print :num1
    print :num2
    showRecursion :num1 - 1 :num2-1
  ]
end
```

# Example Hanoi Solution

# Example Hanoi Solution



# Example Hanoi Solution



```
doTowers HowManyDisks Source destination Auxillary  
  
if HowManyDisks = 1  
  Move Disk 1 from Source to destination  
else  
  doTowers HowManyDisks - 1 Source Auxillary destination  
  Move Disk HowManyDisks from Source to destination  
  doTowers HowManyDisks - 1 Auxillary destination Source
```

# Related Work



COMPUTER SCIENCE  
*Unplugged*



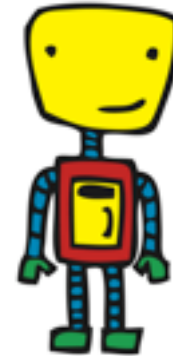
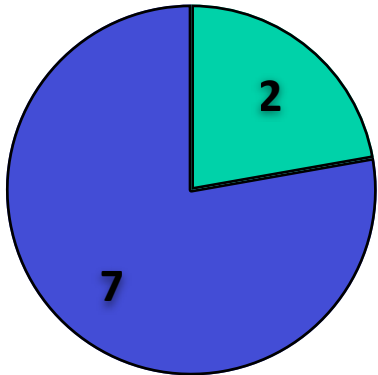
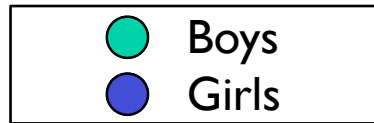
1.



1. <http://ilk.media.mit.edu>

2. <http://csunplugged.org>

# Demographic

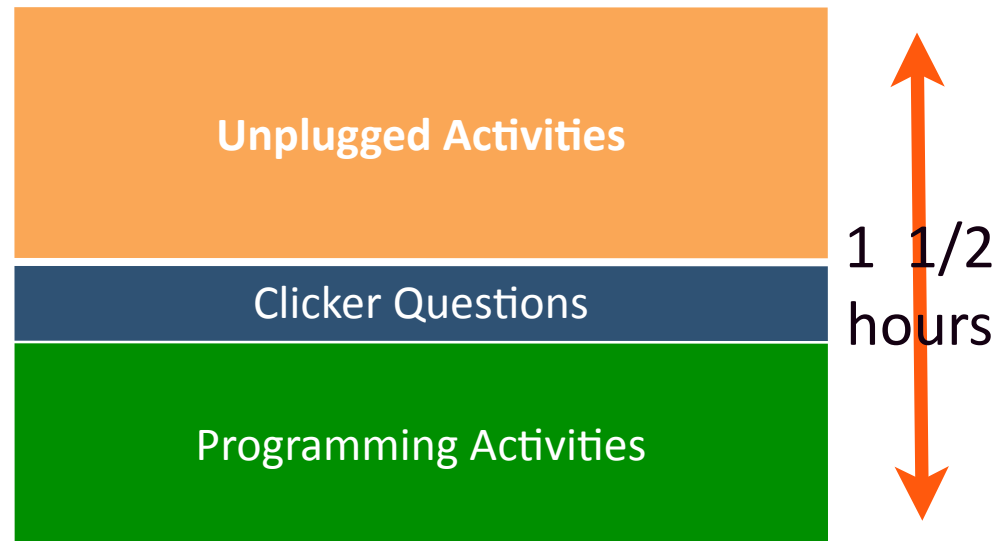


Solving Problems with Algorithms  
Robots and Computers

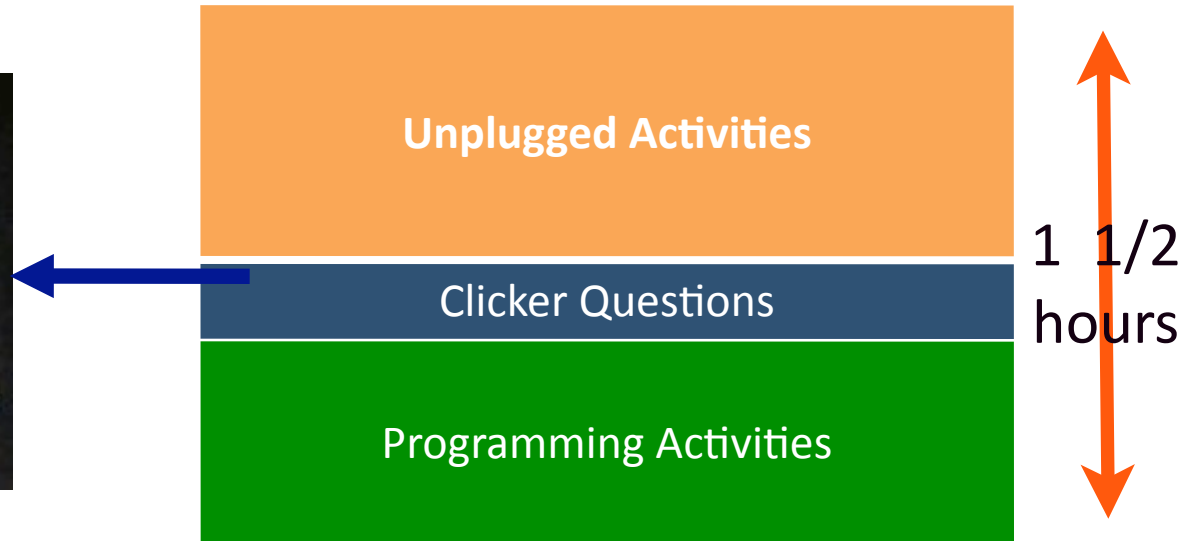


Leaders: Ulrike Stege,  
Katherine Gunion, Terence Nathan

# Overview of Setup



# Overview of Setup







# MicroWorlds vs. Alice



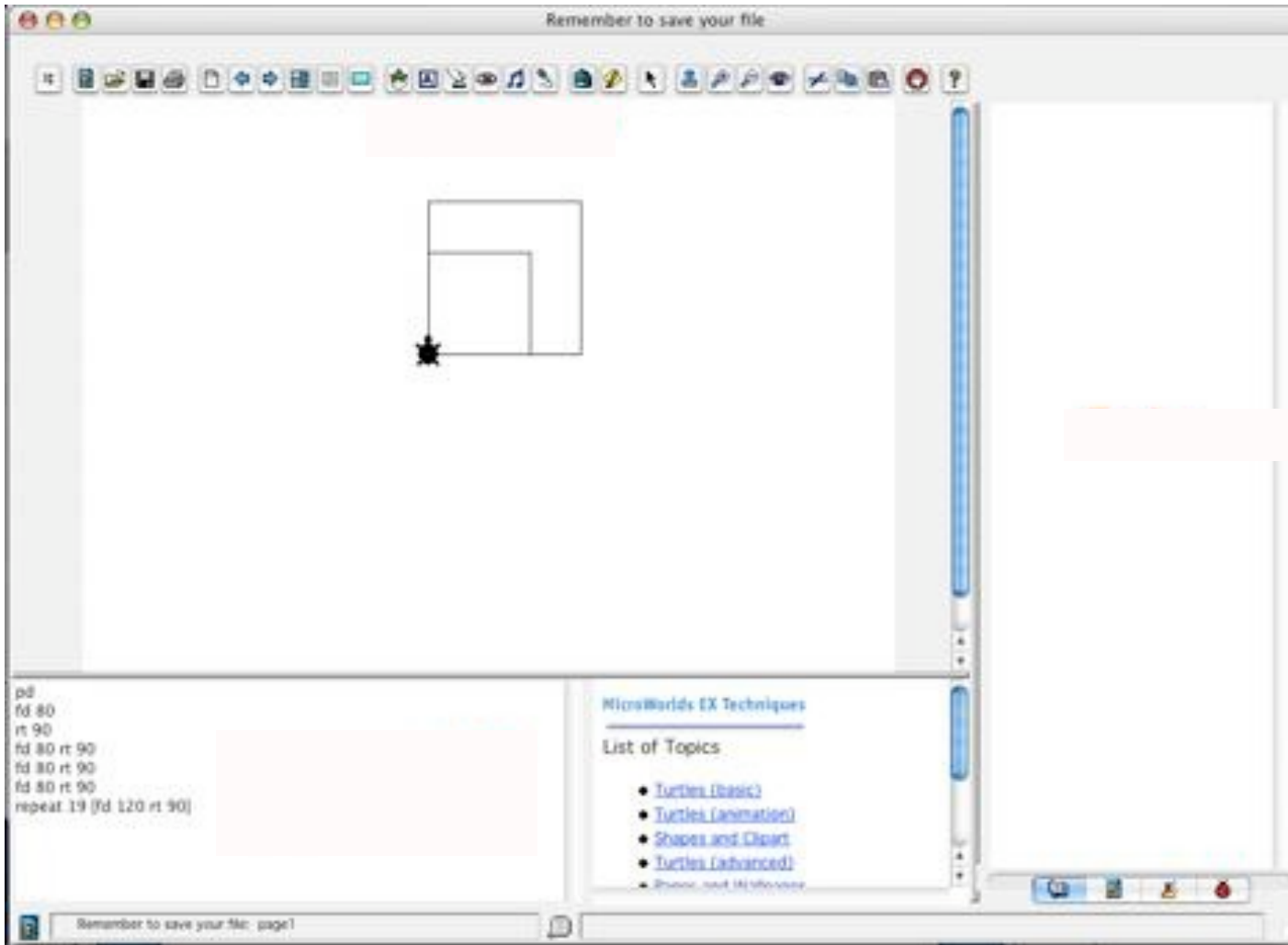
- Two Languages that Support Recursion

| MicroWorlds               | Alice                            |
|---------------------------|----------------------------------|
| Text Based                | Graphical Blocks                 |
| Based on Logo             | Based on Java                    |
| 2 Dimensional             | 3 Dimensional                    |
| Not-Free                  | Free                             |
| Can Create own Characters | Can only use provided characters |

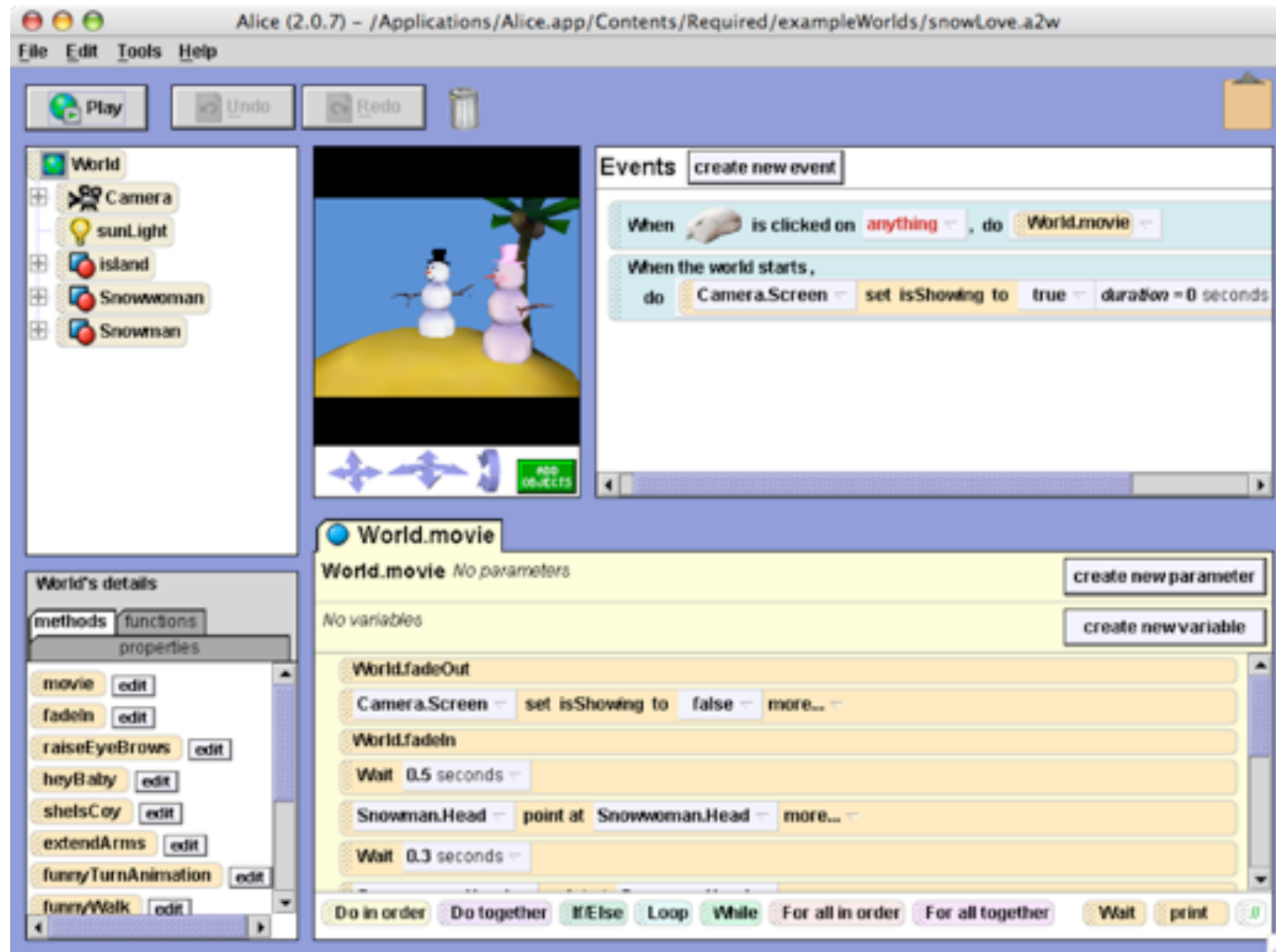
<http://www.microworlds.com/>

<http://www.alice.org/>

# Microworlds Environment



# Alice Programming Environment





# MicroWorlds vs. Alice



- Two Languages that Support Recursion

| MicroWorlds               | Alice                            |
|---------------------------|----------------------------------|
| Text Based                | Graphical Blocks                 |
| Based on Logo             | Based on Java                    |
| 2 Dimensional             | 3 Dimensional                    |
| Not-Free                  | Free                             |
| Can Create own Characters | Can only use provided characters |

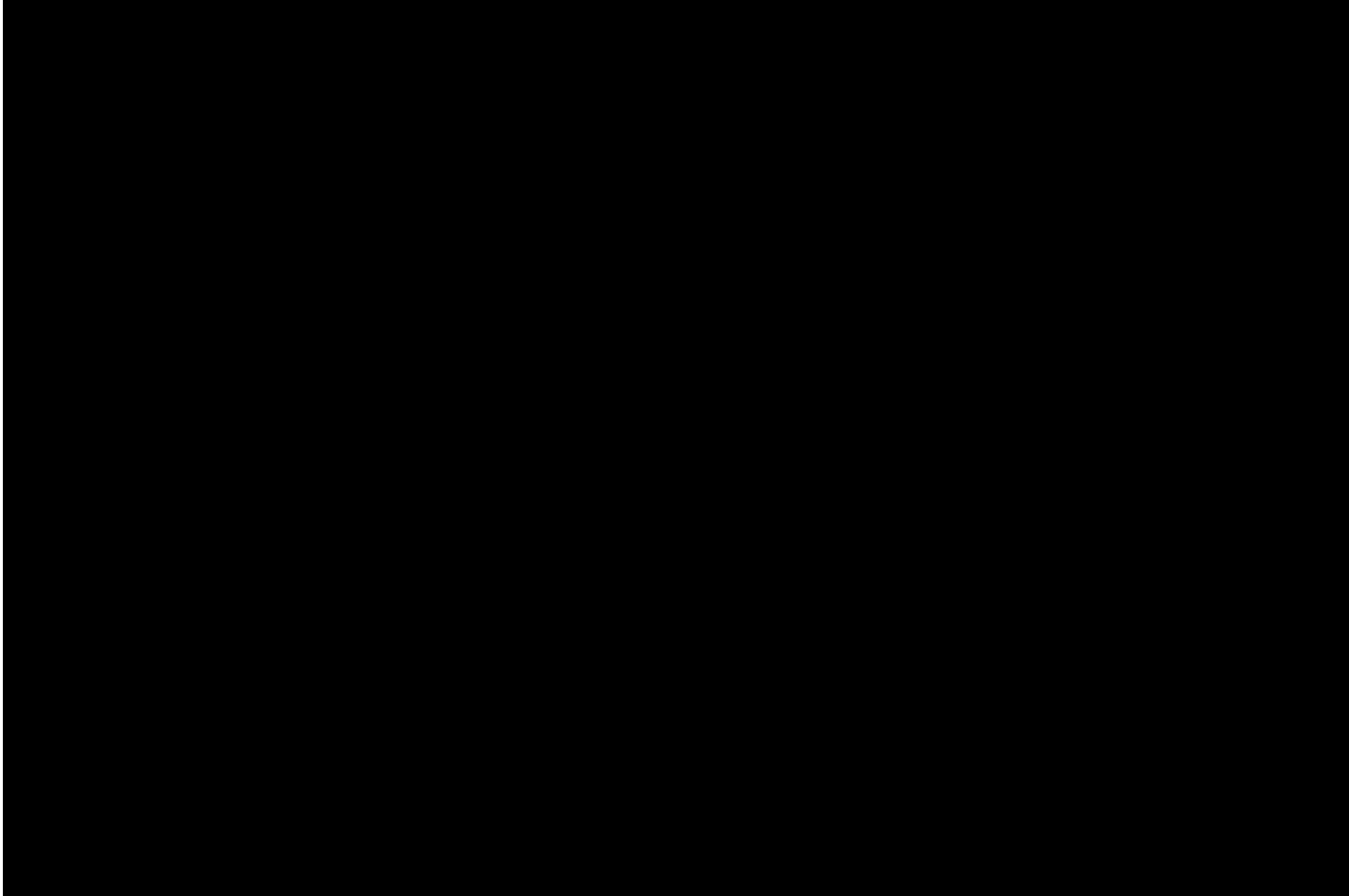
<http://www.microworlds.com/>

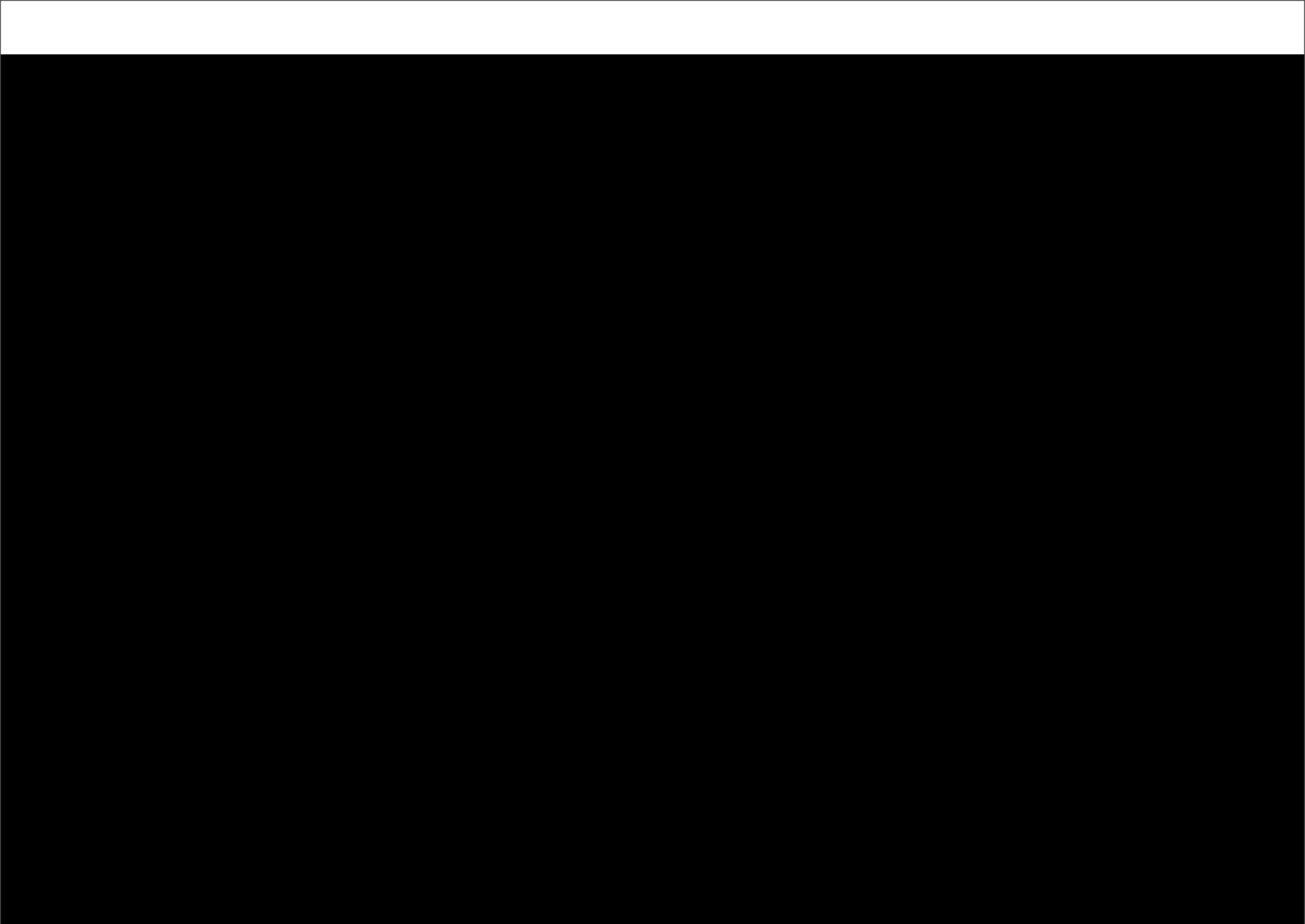
<http://www.alice.org/>

# Weeks and Activities

| Activity                      | Week       |
|-------------------------------|------------|
| Genome Sorting (pre test)     | Week 1     |
| Visual Recursion              | Week 2     |
| Sierpinski Carpet             | Week 3     |
| Line Up                       | Week 4     |
| Animal Chase                  | Week 4     |
| Towers Of Hanoi (unplugged)   | Week 5 & 6 |
| Towers of Hanoi (programming) | Week 5     |
| Match Box Sorting (post test) | Week 7     |

# Video





# Weeks And Activities

| Activity                      | Week       |
|-------------------------------|------------|
| Genome Sorting (pre test)     | Week 1     |
| Visual Recursion              | Week 2     |
| Sierpinski Carpet             | Week 3     |
| Line Up                       | Week 4     |
| Animal Chase                  | Week 4     |
| Towers of Hanoi (unplugged)   | Week 5 & 6 |
| Towers of Hanoi (programming) | Week 5     |
| Match Box Sorting (post test) | Week 7     |



# Weeks And Activities

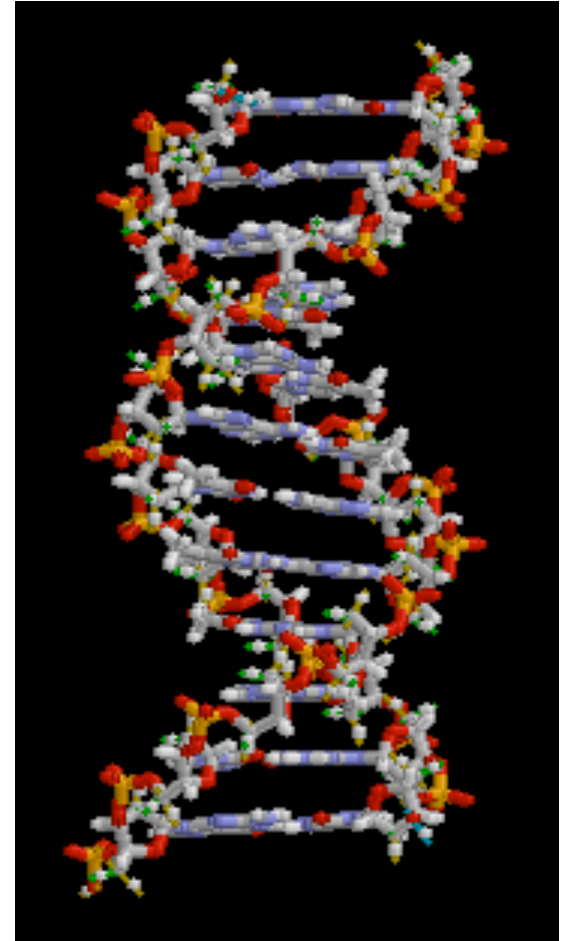
| Activity                      | Week       |
|-------------------------------|------------|
| Genome Sorting (pre test)     | Week 1     |
| Visual Recursion              | Week 2     |
| Sierpinski Carpet             | Week 3     |
| Line Up                       | Week 4     |
| Animal Chase                  | Week 4     |
| Towers of Hanoi (unplugged)   | Week 5 & 6 |
| Towers of Hanoi (programming) | Week 5     |
| Match Box Sorting (post test) | Week 7     |

# Weeks And Activities

| Activity                      | Week       |
|-------------------------------|------------|
| Genome Sorting (pre test)     | Week 1     |
| Visual Recursion              | Week 2     |
| Sierpinski Carpet             | Week 3     |
| Line Up                       | Week 4     |
| Animal Chase                  | Week 4     |
| Towers of Hanoi (unplugged)   | Week 5 & 6 |
| Towers of Hanoi (programming) | Week 5     |
| Match Box Sorting (post test) | Week 7     |

# Genome Sorting (pre test)

Sorting coloured envelopes using  
*comparison based* sorting

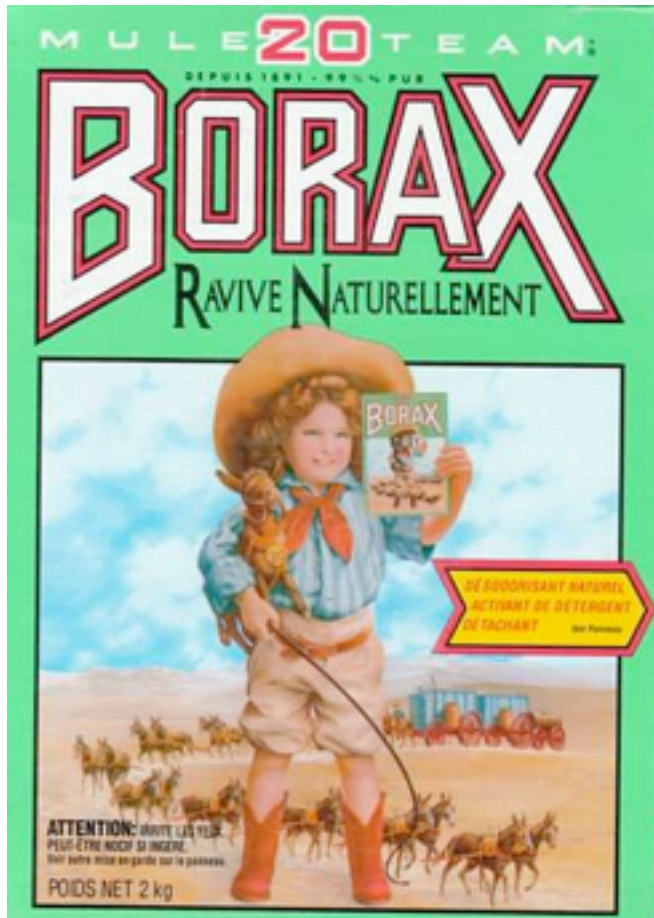


# Students Solutions

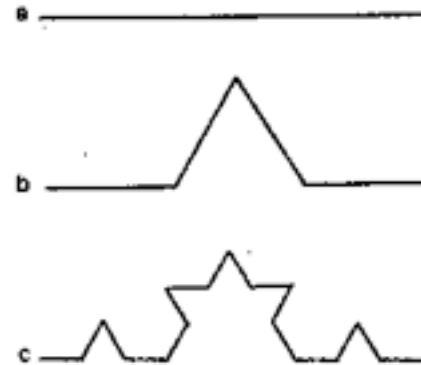
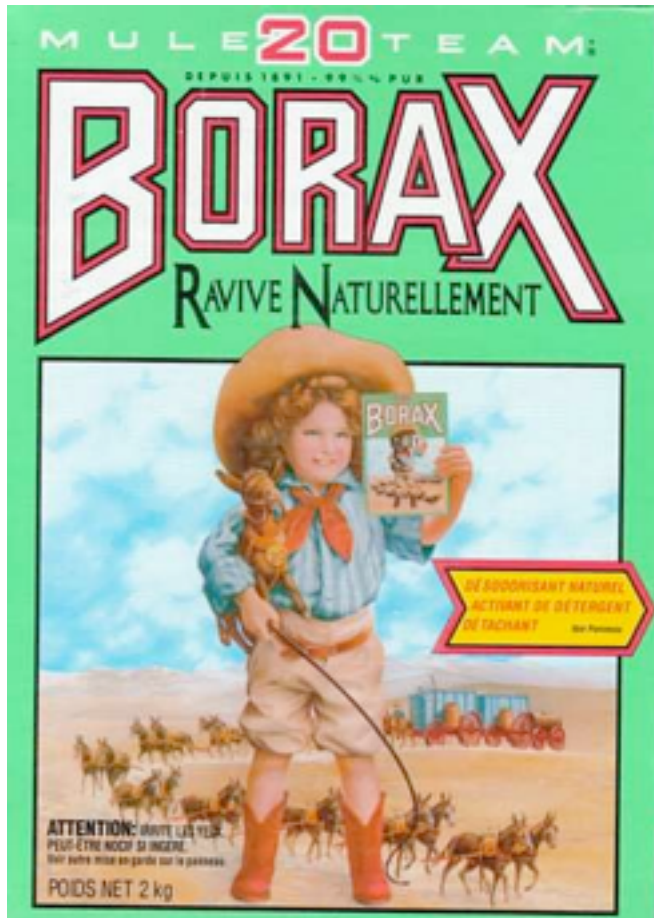
- Initial response: “This is impossible”
- Insertion sort
  - Simplistic & iterative
  - Inefficient
- Simple errors & carelessness
- No groups completed the task
- Students became very frustrated

# Visual Recursion

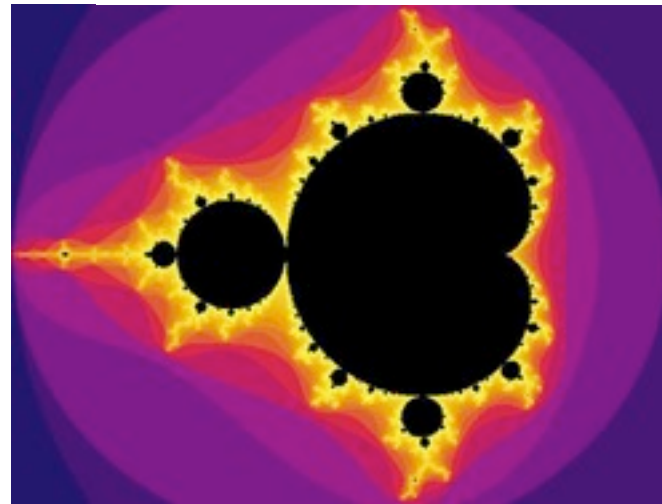
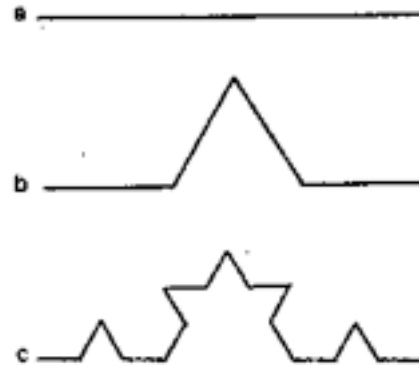
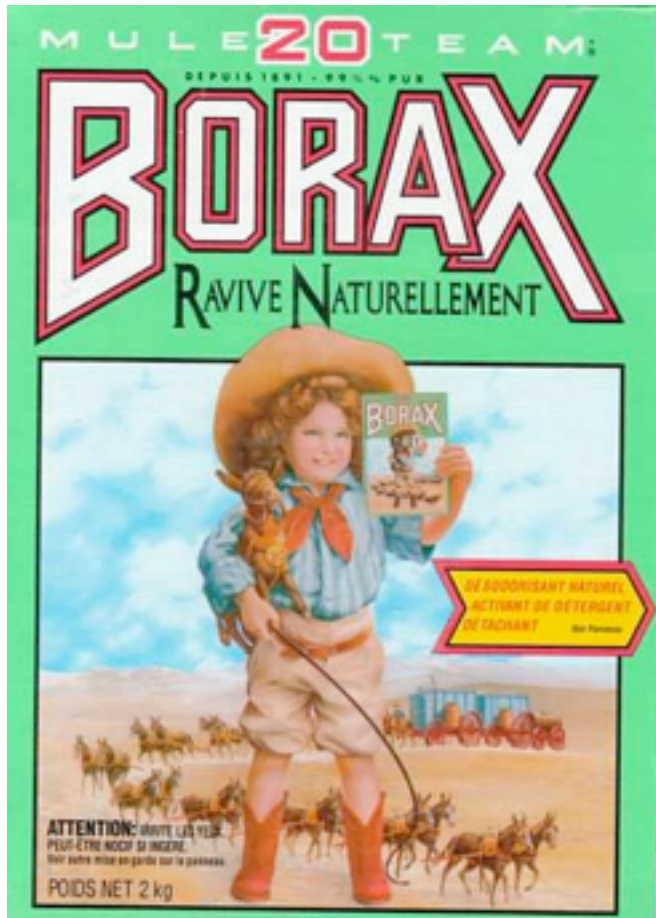
# Visual Recursion



# Visual Recursion



# Visual Recursion





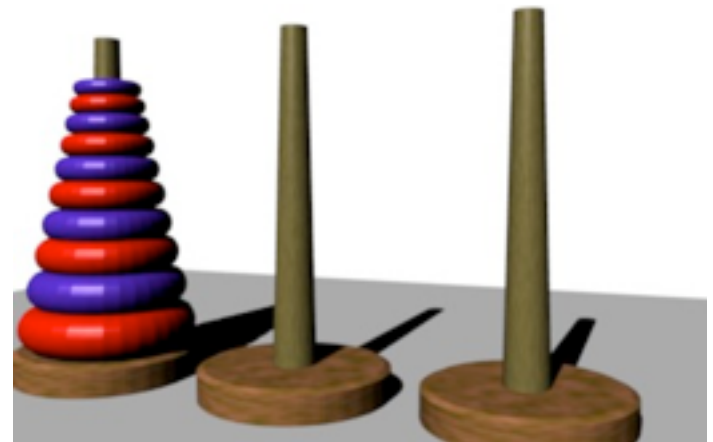
# Student Solutions

- “Picture inside a picture”
- Theme of *infinity* or *forever*
- Related to other aspects of their lives



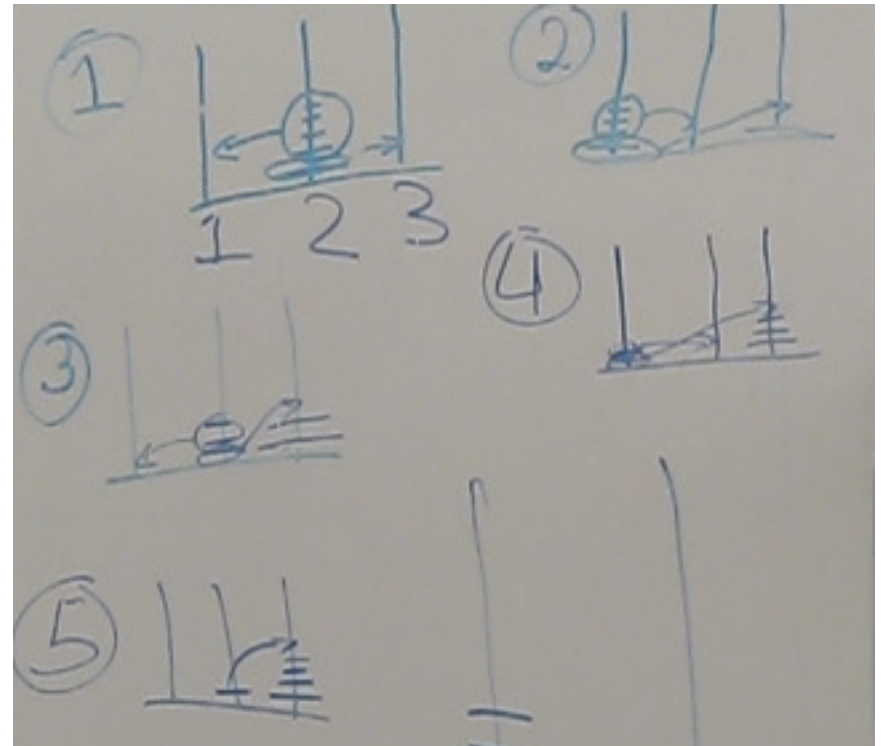
# Towers of Hanoi

- Disks to move from one peg to another, moving one disk at a time
- Larger disks are not allowed on top of smaller ones



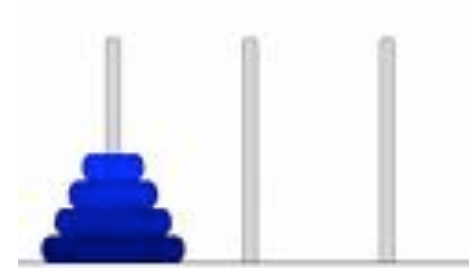
# Student Solutions

- Solved the problem using a reduce and conquer approach
- Even computed number of steps required



# Programming Towers of Hanoi

- 'Move' function
- 'Which' function



Do Towers (How Many Disks) (Source) (Destination) (Auxil)

If *How Many Disks* = 1

Move **Disk 1** From **Source** To **Destination**

Else

Do Towers (How Many Disks -1) (Source) (Auxil) (Destination)

Move How Many Disks From **Source** To **Destination**

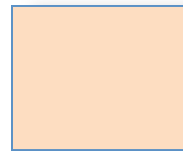
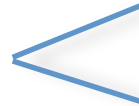
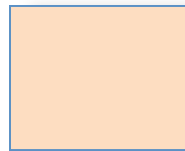
Do Towers (How Many Disks -1) (Auxil) (Destination) (Source)

# A Student's Solution for Towers Of Hanoi in MicroWorlds



```
to doTowers :num :src :dest :aux
ifelse :num = 1
[
move :num :dest
]
[
doTowers :num - 1 :src :aux :dest
move :num :dest
doTowers :num - 1 :aux :dest :src
]
end
```

# Match Box Sorting (post test)



# Student Solutions

- Two piles
- “No reference point”
- Realized they should use recursion
- Designed a ‘quick sort’
- Each took turns on a step

# Clicker Questions

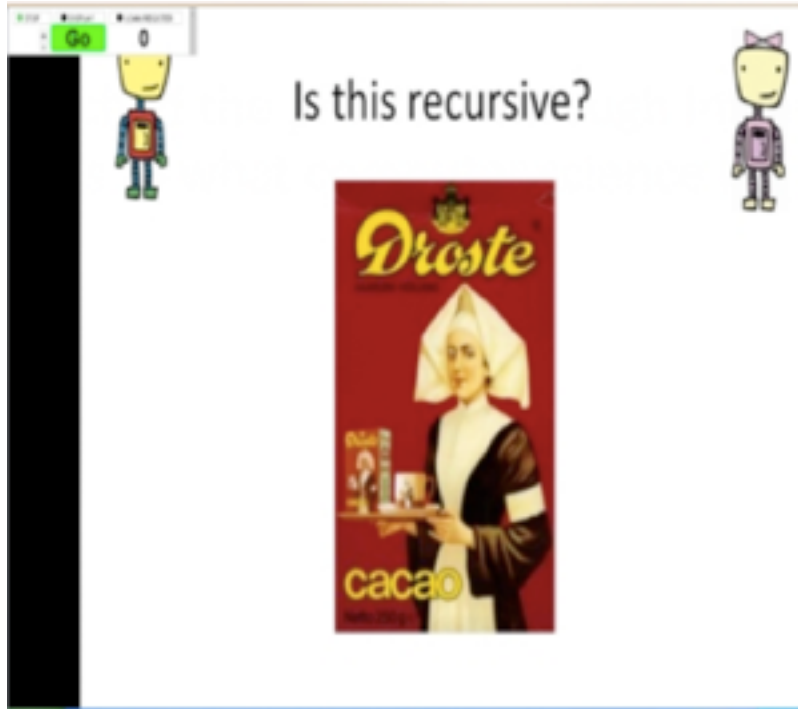




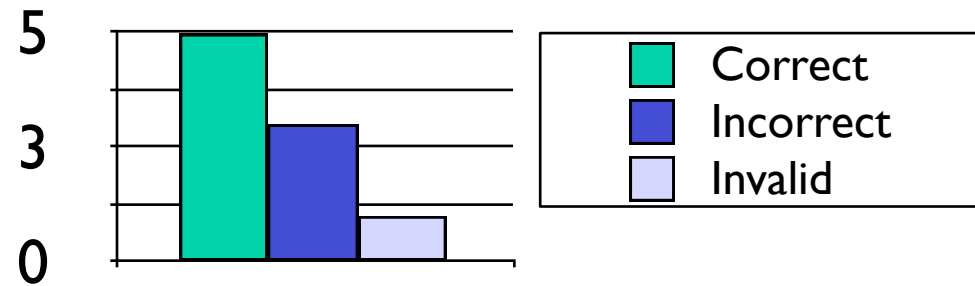
# Clicker Questions



# Is This Image Recursive?



# Is This Image Recursive?



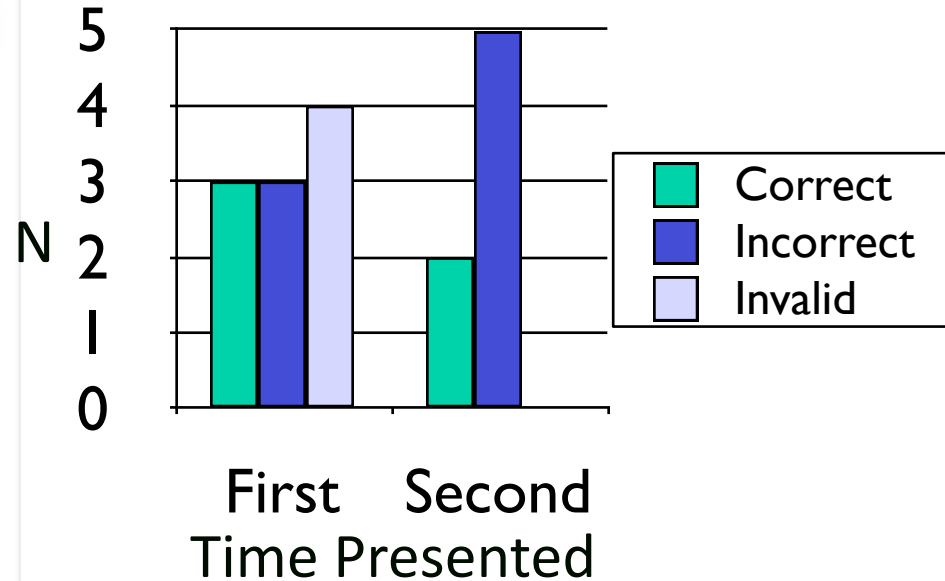
# What is the Output of This Recursive Function?



Clicker 1



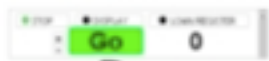
```
To DrawSquares :Num  
Ifelse :Num < 1 [  
  Fill in the Square that you're  
  in with red.  
]  
[  
  Draw a square size :Num  
  Fill it in white  
  DrawSquares :Num - 10  
]
```



# Procedure Recursive?

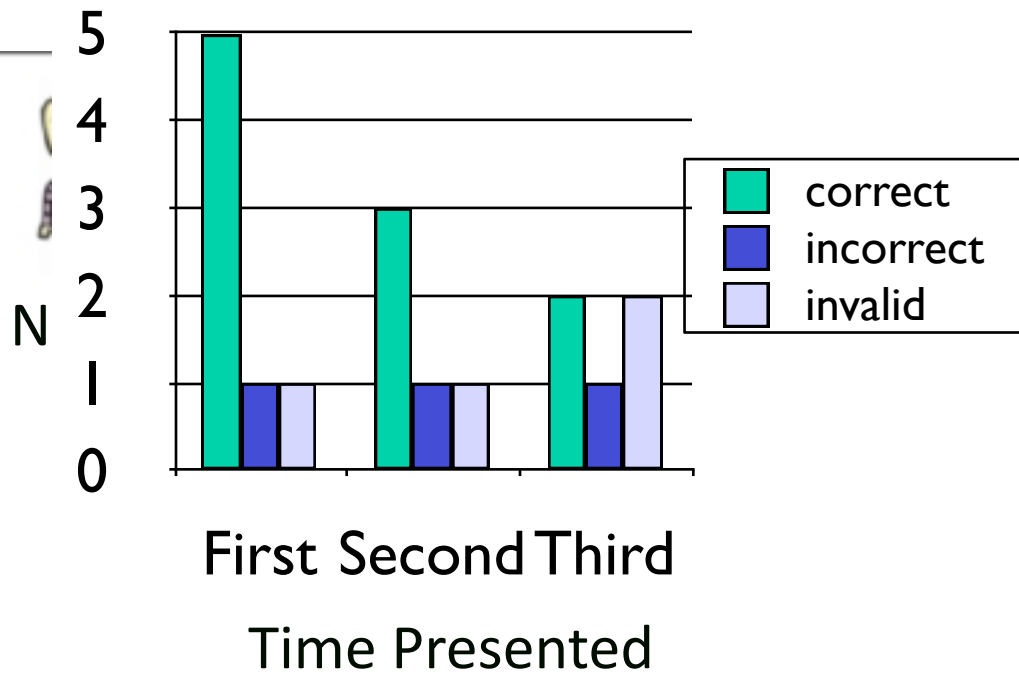


Is Procedure Recursive?



```
to Squares :size
  ifelse( :size < 3
    [drawSquare 2 ]
    [
      drawSquare :size
      Squares (:size - 10)
    ]
  )
end
```

A: Yes  
B: No



Clicker answers became more vague, but following discussion became much clearer and more in depth.

# Limitations

- Clicker results
- Small group
- Skewed representation
- Short study (non-longitudinal)
- Not for 'marks'
- Ethics and public access to the data

# Conclusions

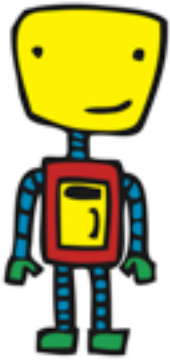
- The students *understood* at different levels
- Students seemed to *enjoy* themselves

| Pre-test  | Post-test   |
|---|---|
| Iterative Solution<br>Too many steps<br>involved<br>Frustrated & limited<br>enjoyment | Used a problem solving<br>strategy<br>Found a solution that<br>was easy to implement<br>Were enthused by the<br>problem |

# Future Work

- Run the program again in the Fall
- Work with other 'Complex Topics'
- Compare to 1<sup>st</sup> and 2<sup>nd</sup> year?
- In 5 years?
- Implemented in the school system?





# Curing Recursion Aversion



University  
of Victoria

Katherine Gunion, Todd Milford, Ulrike Stege  
Computer Science/Education  
University of Victoria

Thanks To Rebeca Dunn-Krahn (Video)  
Terrence Nathan (Teacher)





Try to place all the blocks here in order!!  
Hit REVEAL to reveal their values

Place blocks on the purple/brown squares, to compare them..



Comparisons:

0

Press 'c' to compare