Errors with Errors with Errors with…
Students' Mental Models of Recursively-Defined Objects

Joanna Finkelstein
Pomona College
Claremont, CA 91711
joanna.finkelstein@pomona.edu

Andrea Kimball
Pomona College
Claremont, CA 91711
andrea.kimball@pomona.edu

Jennifer Nguyen
Pomona College
Claremont, CA 91711
jennifer.nguyen@pomona.edu

Christina Tong
Pomona College
Claremont, CA 91711
christina.tong@pomona.edu

1. INTRODUCTION

Recursion is a fundamental and powerful tool in Computer Science, and unfortunately also a concept with which many students struggle. In order to teach recursion more effectively, it is important to understand why students have a hard time understanding it.

A student's *mental model* of recursion is the student's accumulation of ideas about recursion. Existing literature on this subject has shown that the *copies* model allows students to consistently represent recursion accurately, but that the *looping, active, step, magic, return-value,* and other *odd* models do not. These mental models help identify the misconceptions students have about recursion and provide insight on how to change the teaching process.

We have examined common recursion errors in students' efforts to create recursively-defined objects in a lab for the Introduction to Computer Science course (CS51) at Pomona College. Many of their errors resulted from a fragmented understanding of the process used to build a recursive model. We will investigate the application of existing models to students' mental models of recursively-defined objects, i.e. if they apply, which ones are common, and which are viable in this situation.

2. BACKGROUND AND RELATED WORK

Recursion is a programming technique that expresses an operation in terms of itself or an object in terms of objects of the same type.

Existing literature on the topic has identified the various mental models of recursion. The *copies model* is a viable model, in that it always successfully models the recursive process [4]. Non-viable models include the looping model, active model, step model, return value model, magic/syntactic model, algebraic model, and odd models [4]. Most of the existing literature [1,2,3,4] analyses students' responses to recursive problems that are math-based or simple character-based. Our research entails a graphic-based recursive problem, specifically using lines from the Java Objectdraw API. It also requires students to use recursive methods to apply functions to the entire recursively defined object.

3. APPROACH AND UNIQUENESS

3.1 Approach: Recursively-defined Objects

There are many different manifestations of recursion in Computer Science. Past research has looked at how students understand recursion in terms of functions. However, our research takes a different angle, examining instead how students understand and fail to understand objects that are recursively defined. Additionally, we will examine how students understand methods called on these recursively defined objects.

3.2 Evaluation/Experimental Design

Because using graphics to teach recursion allows students to visualize their misconceptions and understand recursion in a different way, we want to know if existing mental models of recursion also apply to students' understanding of these recursively-defined objects. We seek to answer the questions: What model(s) of recursion do students in CS51 apply to create and manipulate these objects? And are those who use the copies model more effective and consistent in implementing recursion than those who apply the other models of recursion?

To answer our research question, we examine 65 students enrolled in the Spring 2014 semester of CS51 Intro to Computer Science at Pomona College. In this Java-based introduction, students learn to use the ObjectDraw API. As part of their recursion lab, the students attempt to draw a recursively-defined object. The recursive object is a colored version of Sierpinski's Gasket, a figure composed of triangles of varying sizes (see Figure 1). Students first create a version without color and then modify it to add color.

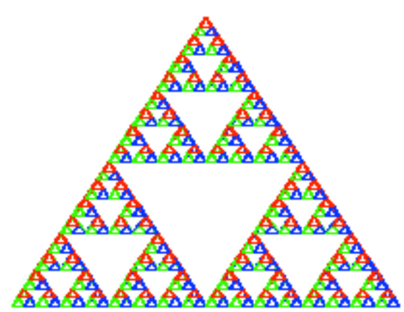

Figure 1

The triangle is created using a controller class, an interface, a recursive class, and a base class. Students must also write methods that can manipulate the object, such as the move and contains methods. During the lab sessions, we examined the correct and incorrect objects that students produced and compared these creations to the existing models.

4. RESULTS AND CONTRIBUTIONS

4.1 Results

Currently, we have data from previous semesters of Introduction to Computer Science courses at Pomona College. We are currently conducting a secondary study and gathering the results from students taking Introduction to Computer Science in Spring 2014. The primary data collection will occur in mid-March. Some of our preliminary results are shown below.

```
// draw the three sides of the first triangle
line1 = new Line(vertex1, vertex2, canvas);
line2 = new Line(vertex2, vertex3, canvas);
line3 = new Line(vertex3, vertex1, canvas);
//make more triangles while bigger than or equal to
//MINSIZE
while (length(line1)>=MINSIZE){
    //get the midpoints
    Location mid1 = midpoint(line1);
    Location mid2 = midpoint(line2);
    Location mid3 = midpoint(line3);
    // create the next smaller triangles
    line1 = new Line(mid1, mid3, canvas);
    line2 = new Line(mid1, mid2, canvas);
    line3 = new Line(mid2, mid3, canvas);
}
```

Figure 2

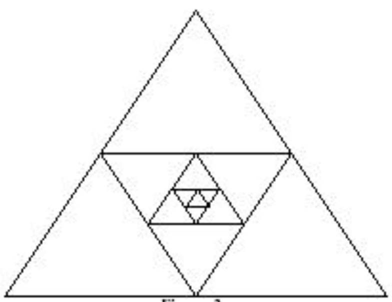

Figure 3

In Figures 2 and 3, this student is using a **looping mental model** in order to create a recursive object incorrectly, thinking of it as a form of iteration that loops until the base case is reached. Instead of regarding the process as a whole, the student is looping through the drawing incorrectly.

In Figure 4, this student is very close to completing the full recursive object, but seems to be exhibiting a **syntactic model**, leading him/her to fundamentally confuse the way that the object is drawn within the recursive syntax. This student likely needs more instruction with recursion.

In Figure 5, this student's error represents a fundamental misunderstanding of how to construct the recursive object, reflecting an **odd mental model** of recursion. This image was the result of syntactic coincidences and guesswork, demonstrating how introductory students often do not comprehend the fundamental structure of recursion.

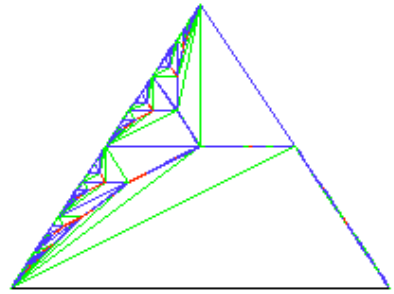

Figure 4

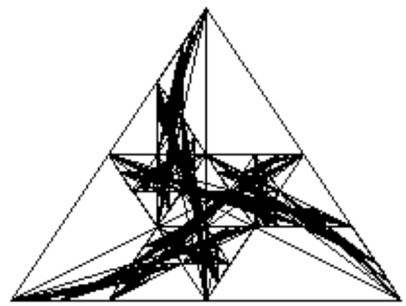

Figure 5

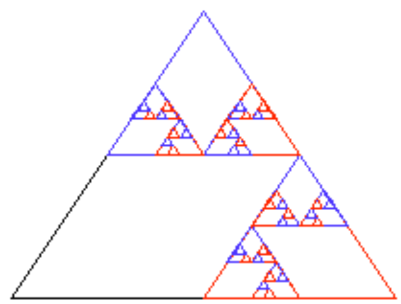

Figure 6

In Figure 6, the student's error was a programming mistake that failed to complete the recursion, likely caused by a **return value mental model** which portrays the steps of recursion as separate iterations rather than a complete process. This mental model led the student to create the recursive image without fully coding the entire process.

4.2 Contributions

The contributions of our work could help improve the teaching of recursion in introductory Computer Science courses at Pomona and at other institutions. By identifying the mental problems that students face while learning recursion, we will be able to take steps towards more effective teaching methods and a better global comprehension of recursion for the next generation of computer scientists.

5. REFERENCES

[1] Ian Douglas Sanders and Vashti C. Galpin. 2007. Students' mental models of recursion at wits. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education* (ITiCSE '07). ACM, New York, NY, USA, 317-317. DOI=10.1145/1268784.1268883 http://doi.acm.org/10.1145/1268784.1268883

[2] Ian Sanders, Vashti Galpin, and Tina Götschi. 2006. Mental models of recursion revisited. In*Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education* (ITICSE '06). ACM, New York, NY, USA, 138-142. DOI=10.1145/1140124.1140162 http://doi.acm.org/10.1145/1140124.1140162

[3] Tamarisk Lurlyn Scholtz and Ian Sanders. 2010. Mental models of recursion: investigating students' understanding of recursion. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (ITiCSE '10). ACM, New York, NY, USA, 103-107. DOI=10.1145/1822090.1822120 http://doi.acm.org/10.1145/1822090.1822120

[4] Tina Götschi, Ian Sanders, and Vashti Galpin. 2003. Mental models of recursion. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education* (SIGCSE '03). ACM, New York, NY, USA, 346-350. DOI=10.1145/611892.612004 http://doi.acm.org/10.1145/611892.61