On Beyond Sudoku: Pencil Puzzles for Introductory Computing

Zack Butler and Ivona Bezakova
Computer Science, RIT
http://www.cs.rit.edu/~pencilpuzzle

Motivation
Giving students interesting context can help them remain engaged with CS concepts. How can we create novel and relevant problems with fresh content that are also easy to understand?

A versatile, fruitful domain
Our assignments cover topics throughout the typical CS 1 and CS 2 curriculum, such as:

- Math expressions (24 puzzle)
- Lists and strings (Blackout Math)
- Nested loops (Easy as ABC)
- Recursion (Jumble)
- Inheritance (KenKen)
- Data structures (Number Tree)
- GUIs (Nurikabe)

See our website for more examples, ideas and even complete assignments to use!

Sample lab assignment: KenKen verifier
[CS 2 Inheritance lab – full version on our website]

In this lab you will write a complete implementation for a KenKen puzzle verifier. Given a puzzle with a potential solution, your program will verify the correctness of each region. [Description of KenKen puzzles, as shown on the sheet next to this poster.]

Problem Solving [paper-based work]:
1. What is the solution to this puzzle? [Puzzle shown on other sheet]
2. Let’s assume that we have a division region whose target is 2 and whose values are stored in an ArrayList<Integer> in the order [2,4,1]. Write pseudocode for a general purpose boolean function, verify, that returns whether any supplied region is correct or not. You can assume the target and values are already accessible.
3. From a design perspective, what state and behavior is common to all regions?
4. Likewise, what state and behavior is unique to each region?
5. Design a UML diagram with an abstract Region class, and two subclasses, AddRegion (for performing addition), and DivideRegion (for performing division).

Implementation:
The puzzle to be verified is specified in a text file that your main program will read in. [File format that students use is given in detail]
[Details of expected program output are given here]
This is the complete UML diagram [with Java syntax]. It includes the private state that we suggest you use. [UML given]
Java Documentation: There are seven classes for this assignment. [Links to javadocs for KenKen (main), Grid, Region ( superclass), AddRegion, MultiplyRegion, SubtractRegion, DivideRegion]

Testing:
In addition to the input file from the top of the writeup, we are also providing a second one. It is a 5x5 puzzle with 9 regions. It uses all 4 of the operations.

Pencil Puzzles
Pencil puzzles are those solved with a deductive, algorithmic process, typically on paper, and with well-defined rules. Many types are widely available and can provide fresh content. Here are two that we have used, see more on other posters!

Blackout Math
Blacken any two squares so that the remaining squares form a correct equality.

Easy as ABC(D)
Put exactly one A, one B, and one C (and in the puzzle at right, one D) in each row and column, such that the letters outside the grid correspond to the first letter encountered along that row or column.

Well-defined algorithms
Both pencil puzzles and open-ended or lateral thinking puzzles can be used to stimulate creative and algorithmic thinking. While both have value, pencil puzzles are designed to be easy to understand and naturally encourage an algorithmic thought process. As such, they may be more suitable for smaller, stand-alone assignments. Many puzzle types are also publicly available making it easy to provide fresh content term after term.

Related good ideas
CS unplugged uses activities without electronic devices to teach algorithms and problem solving. We use paper-based problem solving sessions but students then apply their results to create software implementation of the algorithms developed. Puzzle-based learning uses more open-ended problems to stimulate creative thinking, whereas we focus on existing puzzle types with clear rules so that students can not only develop algorithms but see context for various programming constructs.

Progress, preliminary results
So far we have used pencil-puzzle assignments for over 800 students in CS1, CS2 (including a trailer section), and a problem solving bridge course for incoming MS students.

In our department’s two CS2 trailer sections in fall 2015, one did a traditional inheritance lab and the other the KenKen lab shown here. The student comments for KenKen were overall much more enthusiastic (“It was fun!” This lab has been my favorite of all the ones we’ve had to do so far!”) while the regular lab was perceived as easier and more straightforward.