Programming NIM in Scratch
Lesson Plan

Subject: Math
Topic: Computer Programming and Problem Solving
Grades: 6-8

Stage 1: Desired Results

Summary: This lesson assumes students have some familiarity with the computer programming language Scratch (http://scratch.mit.edu), but the project is suitable for novices to programming.

Common Core State Standards:

- **6.EE.B.5:** Understand solving an equation or inequality as a process of answering a question: which values from a specified set, if any, make the equation or inequality true? Use substitution to determine whether a given number in a specified set makes an equation or inequality true.
- **6.EE.C.9:** Use variables to represent two quantities in a real-world problem that change in relationship to one another; write an equation to express one quantity, thought of as the dependent variable, in terms of the other quantity, thought of as the independent variable. Analyze the relationship between the dependent and independent variables using graphs and tables, and relate these to the equation.
- **MP1:** Make sense of problems and persevere in solving them.
- **MP4:** Model with mathematics.

Transfer Skills

Students will be able to independently use their learning in new situations to…

- analyze a situation and represent it mathematically
- understand and communicate their mathematical reasoning to others and through computer code
- understand games as applications of mathematical problems solving skills

Essential Question

- Can you teach a computer to beat you at a strategy game?

Stage 2: Evidence

Transfer Task: Students will create a computer program using the language Scratch that can play multiple variations of the game NIM.

Other Evidence: Students will complete activities designed to help them:

- understand the math underlying the simple strategy game NIM
- analyze the mathematical patterns within variations of NIM
- represent winning strategies for NIM using mathematical expressions and equations

Stage 3: Learning Plan

Project Overview:

*(**NOTE:** This project assumes students are already familiar with Scratch, its interface, and introductory programming concepts, such as control loops and variables. The Scratch web site contains tutorials and examples if needed to provide this introduction.)*

1. Introduce the project to students. Learn the simplest version of NIM.
2. Develop a table to explore the patterns in NIM. Translate this simple version into a mathematical equation that can be used to express a winning strategy for the game.
3. Teach several simple variations of the basic NIM game. Students work in groups to analyze the games and model them mathematically.
4. Introduce the Scratch project to students as a multi-phase problem. Students will not be expected to successfully complete all phases. Rather, this shows students opportunities for greater depth so that those who solve earlier phases can move on to choose greater challenges and variations:
    a. Simulate (model) the basic game with two human players (i.e. represent the pieces and game play, but the computer merely displays the moves of each player).
    b. Program the computer to play as one of the two players in the game.
    c. Program the computer to win the game.
    d. Program the computer to simulate other variations based on player choices (i.e. different starting position, different move options, different winning condition).
    e. Program the computer to win the game regardless of the variation chosen by the human player.
    f. Program additional modes of NIM (options here are nearly limitless; online searches for NIM will provide many varieties that students can explore, or they can invent their own versions.)
5. Students work in pairs or groups to develop their programs.
6. Conduct a tournament with student programs competing against each other. (This is executed by having the students input the moves selected by one program into the other program as the "human" move.)
7. Student groups review the results and outcomes of the tournament and revise their programs.
8. Student groups present to the whole group an explanation of their algorithms for playing NIM and the mathematical reasoning behind their coding choices.

Assessment:

1. Students develop a portfolio of the math work they used to develop their algorithms and coding strategies.
2. Student groups are assessed on their ability to create a working game that at minimum reaches level "c" above (ability to play and win the basic NIM game). *Though this is mathematically simple, the reasoning and coding required to create this are relatively sophisticated.*
3. Individual students should be able to accurately explain the code for their group's game and also should be able to successfully demonstrate the algorithm/strategy when playing a "live" game between two humans.

Resources/References:

Rules of NIM and variations:

- Freeman, C. (2005). *NIM: Serious Math With a Simple Game*. Prufrock Press.
- http://nrich.maths.org/1209
- http://www.byrdseed.com/lets-play-some-inductive-math-games/

Examples of computer NIM programs to show students some possibilities for their version:

- http://education.jlab.org/nim/
- http://illuminations.nctm.org/Activity.aspx?id=4221
- http://www.archimedes-lab.org/game_nim/nim.html (includes an explanation of the formula, so use after students have done their own analysis)
- http://www.gametheory.net/games/nim.html (a long list of additional online NIM games; I have not vetted all of these, so review before sharing with students)

Scratch and Coding:

- http://scratch.mit.edu
- http://info.scratch.mit.edu/Video_Tutorials
- http://learnscratch.org/
- http://code.org/learn
- http://cs.harvard.edu/malan/scratch/index.php (a more advanced tutorial, aimed at high school or college students, but may be useful for more advanced middle schoolers, or as a reference for the teacher)