Inspiring Curiosity through Computer Science Challenges:
Students Modeling Chemistry using "Scratch" Visual Programming

Lynne N. Cohen
Orange County Public Schools, Florida

Summary

Integration of engineering and computer science in secondary education supports student success and increases the likelihood that students will pursue STEM careers (Gullen & Sheldon, 2014). Computer programming can be linked to many subjects but has specific potential in Chemistry curricula to support both digital and scientific literacy.

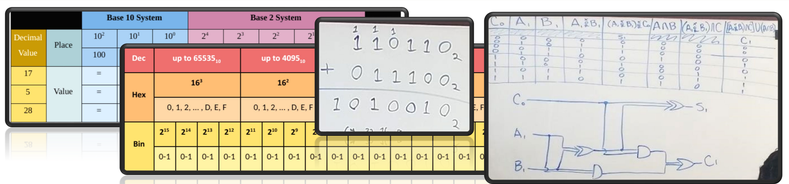
- **Chemistry courses teach problem-solving skills** grounded in practical applications & may directly translate to lucrative engineering & computer science careers (Carr, 2014).
- **Computer science and chemistry share similar skill sets** of *defining* problems, *developing* models, *designing* piece-wise solutions, *interpreting* data, and *procedurally testing* methods and solutions (National Research Council, 2012).
- **Computing is required in modern experimentation** for *modeling* and *calculation,* i.e., MATLAB which uses syntax similar to the Java programming language (Carr, 2014).
- **Computer science challenges are authentic applications** for science modeling, formula use, and unit analysis – all essential to 6-12 chemistry standards.
- **Students report** increased motivation for hands-on computer-based activities involving authentic problems, creative freedom, and development of student expertise (Gullen & Sheldon, 2014).

Research Activities

Module 1: *Making better sensors with carbon nanofiber & 3D printing.*

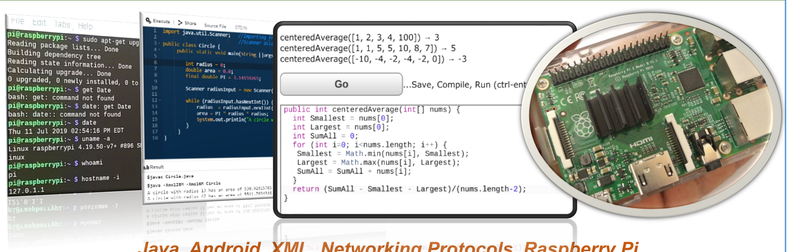
Carbon Nanotubes, Buckypaper, CNT Plates, Composites, Sensor Testing, 3D Printing

Module 2: *Using Digital Design to make Computers better, smarter, and faster.*



Binary, Hexadecimal, Logic Gates, Basys 3 FPGA, Verilog, Deep Learning

Module 3: *Coding is the only International Language... Are you fluent?*



Java, Android, XML, Networking Protocols, Raspberry Pi

Coding Lessons: Overview & Activities Planned

Overview

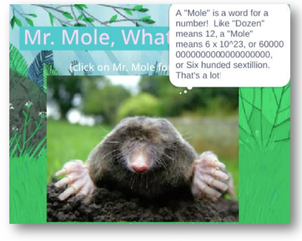
Throughout an introductory chemistry course, students experience a series of programming challenges using the "Scratch" visual block programming language. All challenges involve creation of models and calculators relevant to chemistry standards.

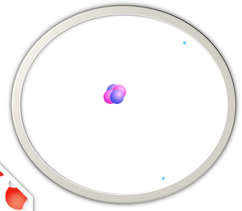
Preview & Programming Blocks of Animated Electron Excitation Model from scratch.mit.edu.

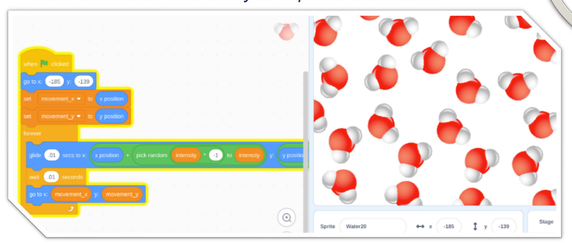
Program by Lynne N. Cohen, 2019. Images from within platform.

Activities in Series

- **Exit Ticket -** *Introduction and/or Practice.* Choose a sprite from the interface & create a three-block program with an event, movement, and text.
- **Model an Atom** – Draw an atom by creating circles representing protons, neutrons and electrons. Animate! *Is your model like Dalton's, Thompson's, Rutherford's, Bohr's, or Quantum? Why?*
- **Modeling Molecular Motion** – Import an image of a water molecule and use movement blocks to motion solid, liquid, or gaseous phases. *Compare your work to others. How can you improve the model?*







Multi-Day Challenge: Mole Conversion Calculator

Day 1 – What is a mole? *Avogadro's Number and mole concept;* ***Create a Scratch with a Mole sprite.***

Day 2 – Conversion Practice *Practice conversions & Unit analysis.*

Day 3 – Converting with Sci Notation *Convert from moles to particles.* ***Add a calculator to scratch*** *to do the conversion for you!*

Input * 6.02E23 = Output

Note: *Explicit instruction for variables is needed on days 3 & 4.*

Day 4 – Moles to Volume *Prove 22.4L per mole at STP with ideal gas law.* ***Add to calculator!*** ***Input * 22.4 = Output***

Day 5 – Molar mass *Reading atomic mass as "grams per mole"; Practice converting moles to grams.* ***Can we add this into our calculator? Challenge: Second input!***

Day 6– Molecular Molar Mass *More complex practice with multiple atoms.* ***Add to calculator!***

Input 1 * Input 2 = Output

Day 7: Final Touches & Demonstration

Lessons Learned

My experiences with the RET program revealed some important concepts for my teaching practice. The variety of fields involved in developing smart devices is incredible, and the field is wide open for intelligent and motivated young people, if they are inspired to pursue STEM careers. The medical and infrastructure possibilities of IoT are breathtaking—but it will be up to young people entering the field to decide if the technology will be used for good or for ill. Of course, in order to live, work, and lead in the future, students will need to know the language of the future. They will need to know more than how to communicate *on a computer*, they will need to be able to communicate ***with*** *a computer* - in code!

Implementation Strategy

Adaptation - Projects can be altered for a variety of topics and subjects. *Guiding points:*

- **Keep it simple.** The first activity (and repeated variations) should require only 2-5 blocks. Intermittent small challenges encourage familiarity and confidence.
- **Build slowly.** Add one or two new concepts each time and consider difficulty.
- **Creativity takes time.** Plan for plenty of time for creative or exploratory tasks.

Gradual Release - Give structured guidance at the start; expect independence at the end.

- **"I do"** - Clarify expectations with a visual image or simplistic sample program.
- **"We do"** - Provide explicit instructions and have available sample code to achieve the *minimum* requirements of task.
- **"You do"** - Encourage exploration and pose challenges through questions: *Does that move? What happens if I click on it? How is this useful as a model?*

Scaffold based on Difficulty – Note the potential difficulty levels of different Scratch tasks. *Based on a 2013 case study of 423 scripted projects* (Maloney et. al., 2008):

- ❖ **Pre-Programming:** Import Images, Draw Sprites, Record Sounds.
- ❖ **Beginner Scripting:** Animate a single sprite, Small stacks of code, Simple user interaction (click to start script), Simple motion.
- ❖ **Intermediate Scripting:** Multiple sprites, Larger stacks of code, Keyboard input, Loops, Simple Conditional (if/then), Simple Broadcasting & When-Receive.
- ❖ **Advanced Scripting*:** Boolean statements (And, Or, Not), Randomization, Number generators, Uncommon math operations (absolute value), Variables ** students unlikely to approach without guidance*

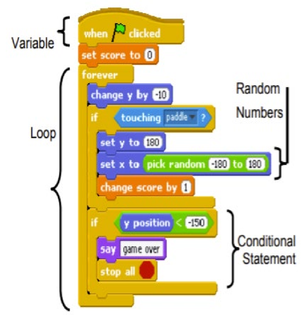


Figure from Maloney et. al., 2008

Acknowledgments

RET Site: COMET Program, College of Engineering and Computer Science, University of Central Florida. This content was developed under National Science Foundation grant EEC-1611019.

Special thanks to James Ebbert & Barbara Barnard-Figaro for help and inspiration.

References

- Carr, K. (2014, July 17). Computer science adds new dimension to study of chemistry. The Stanford Daily. Retrieved from https://www.stanforddaily.com
- Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. Paper presented at the SIGCSE'08 - Proceedings of the 39th ACM Technical Symposium on Computer Science Education.
- Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? Journal of Information Technology Education: Research, 15, 283-303. doi:10.28945/3521
- National Research Council. (2012). A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. doi:10.17226/13165