SRI Education

ETR

C2STEM

Collaborative, Computational STEM Learning Environment

Salem State University

Stanford University

NSF

STEM+C Program

Gautam Biswas & Nicole Hutchins

Dept. of EECS/ISIS

Vanderbilt University, Nashville, TN. USA.

gautam.biswas@vanderbilt.edu

Collaborators

SRI (Kevin McElhaney), Stanford (Dan Schwartz), Salem State (Luke Conlin), Shuchi Grover (Indep consultant), Vanderbilt (Akos Ledeczi)

Evaluator: ETR (**Shannon Campe**)

Team: **Nicole Hutchins, Miklos Maroti, Brian Broll, Naveed Mohammed, Satabdi Basu, Christopher Harris, Carol Tate, Kristen Blair, Doris Chin, Rachel Wolf,**

Supported by NSF Award DRL-1640199.

Motivation

- Computation third pillar (along with theory and experimentation) of science and engineering disciplines (Wing, 2016)
- “Using Mathematics and CT” key science and engineering practices (K-12 Science Education (NRC, 2012); Next Generation Science Standards (NGSS Lead States, 2013); Integrative Science Framework (Honey, et al., 2014))
- Learn to solve problems while “thinking like a computer scientist” (Wing, 2011; Barr & Stephenson, 2011)
- Science &Engineering can be an important context for teaching CT (K-12 CS Framework, 2016)
- CT will help students “become creators, and not just consumers of the next wave of computing innovations” (Schnabel, 2011; Wing, 2006)
- STEM and CT essential skills for 21st century workforce (NRC, 2011)

Our Approach

- Integrate CT with existing K-12 science curricula (middle, high school)
- Goal: Synergistic learning of science and CT concepts

  (NRC 2010; Basu, et al., 2017; Sengupta, et al., 2013; Navlakha & Bar-Joseph, 2011; Brennan and Resnick (2012); García-Peñalvo et al., 2016; Weintrop et al, 2016; Wilensky, Brady & Horn, 2014)

- Accomplish this by developing a computer-based learning environment

  C2STEM: Collaborative, Computational STEM Learning Environment)

  – Learning by Modeling and Simulation (Sengupta, et al, 2013; Hutchins, et al., 2018)

What is Computational Thinking?

- General, analytic approach to:
  - Problem solving
  - System design
  - Understanding human behavior
- Concepts fundamental to computing & computer science
  - Algorithm design & structure
  - Decomposition & Composition
  - Modularity
- Practices central to STEM modeling, reasoning, and problem solving
  - Problem representation
  - Abstraction and decomposition
  - Simulation and prediction
  - Verification

Barr & Stephenson (2011)
Guzdial (2008)
Wing (2006, 2008, 2010)

Distinguishing characteristics of our research

- Emphasis on integrating CT with existing middle & high school science curricula
  - Simple enough for use by science teachers with no programming experience
- Understanding challenges typically faced by students working in such CT-based environments
  - Focus on synergistic integrated learning
- Developing and evaluating an adaptive scaffolding framework based on an assessment of students' modeling strategies and performances
  - Students solve complex, open-ended problems; provide scaffolding that helps them learn and succeed
- Use of multiple assessment types for studying students' science and CT learning and characterizing students' learning processes
  - Analyze students' learning performances and behaviors

Synergistic STEM + CT Learning

- Predicated on the idea – simultaneous learning of two overlapping domains can lead to better learning of concepts and practices in the two domains than when the domains are learned separately
  - Argument against: Conceptual learning of STEM & CT as coupled domains can be difficult (Roman-Gonzalez, et al., 2017)
    - Basu, et al., 2016: converting domain understanding into computational constructs
    - Chi, 2005: individual behaviors intuitive;, but aggregate behaviors (emergent phenomena) are hard to understand
    - Reiner, et al., 2000: biology and ecology – individual entities to polulation-level phenomena
  - Argument for: STEM + CT have similar epistemic origins (Sengupta, et al., 2013; Weintrop, et al., 2016)
    - Harel & Papert (1990): Programming is reflexive with other domains
    - Cooper & Cunningham, 2010; Margolis & Fisher 2002: CT better inculcated when taught in the context of other domains

Synergistic STEM + CT Learning

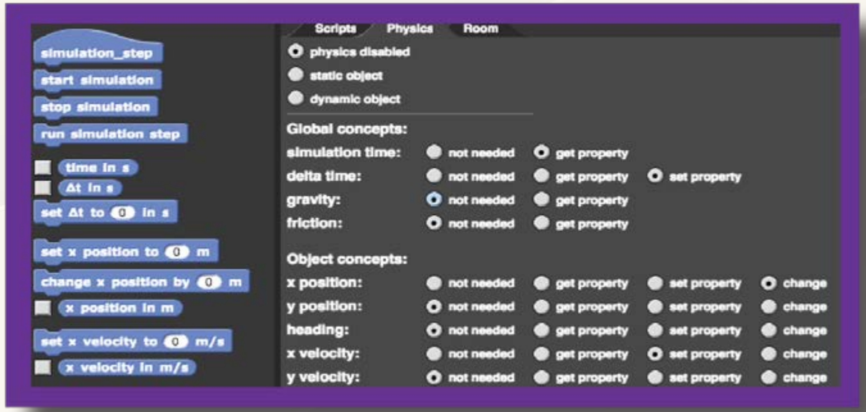
- Actualize using a *learning by modeling* approach – students learn their STEM topics by building and solving problems with computational models
  - Benefits (Sengupta, et al., 2013)
    - Lower learning threshold for science concepts by linking them around intuitive computational mechanisms
      - Create discrete representations of fundamental laws, and develop a deep understanding of the laws using the step by step simulations ; simpler to understand than equation-based representations (Redish & Wilson, 1993)
      - Computational modeling & programming are core scientific practices (Soloway, 1993)
      - Contextualized representations make it easier to learn programming (Papert, 1991)

Previous Work

- Visual programming interfaces + block-structured language constructs
- All of them extend Netlogo – a multi-agent modeling environment
  - CTSiM (Computational Thinking using Simulation and Modeling)
    - (Sengupta, et al. 2013; Basu, Kinnebrew, & Biswas, 2017)
  - ViMap
    - Elementary and middle school classrooms (Sengupta, et al., 2015)
  - CT-STEM
    - Focused on K-12 curricula (Jona, et al., 2014)

C2STEM System Highlights

- Employs novel computational paradigm that combines
  - Visual, block structured programming (based on Netsblox (Broll, et al., 2017)
  - Domain specific modeling languages (DSMLs) (Hasan & Biswas, 2017)
  - "step-by-step" programming (simulation models)
- Low threshold, wide walls, high ceiling
- Coupled multi-level representations to support learning
  - Conceptual modeling & inquiry components offer new forms of exploring & decomposing STEM domain



Design Principles

- Evidence-centered Design
- Domain specific modeling languages (DSMLs)
- Exploratory learning of dynamic processes
- Multiple assessment forms
  - Formative (embedded assessments)
  - Summative assessments
    - Challenge Problems
    - Preparation for Future Learning (PFL)

C2STEM – Instructional Design

- Involve teachers in curriculum development and align with classroom activities
- *Instructional Tasks*
  - Highly scaffolded with the goal of focusing student attention on the learning and application of primary Physics & CT concepts, often one at a time
- *Inquiry Tasks*
  - Students asked to explore environment & find relationships between variables; usually guided by leading question
- *Model Building*
  - Students apply their learned Physics + CT concepts to build computational models of relevant Physics phenomena
- *Formative Assessments*
  - Assess student learning with multiple choice, short answer questions, & small computational model building exercises
- *Challenge Problems*
  - Comprehensive; test students' abilities to put together all concepts & practices to build a computational model that solves a difficult problem

<<Switch to C2STEM demo/hands on activities>>

More details of C2STEM system and curricular modules developed can be obtained from

www.c2stem.org

Contact: nicole.m.hutchins@vanderbilt.edu
gautam.biswas@vanderbilt.edu