Visualizing Computational Thinking Across Math and Science

Charles Xie ✉: charles@intofuture.org
Institute for Future Intelligence 🌐: https://intofuture.org

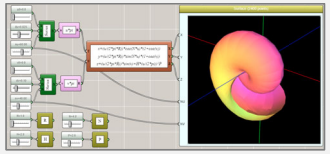
Introduction

Research Hypothesis

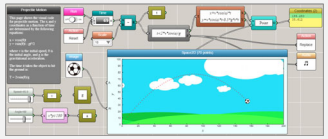
A challenge in teaching real-world computational thinking is that the thought process of solving a concrete problem can easily escalate into a complex mental model consisting of many abstract, intertwined moving parts that are often difficult for students to imagine and think through, preventing them from sorting out a solution and building up self-efficacy. Externalizing such a complicated mental process step by step through drawing representational diagrams piece by piece (i.e., making students' computational thinking visible while they are shaping it) can be cognitively offloading.

Math and Science Application Examples

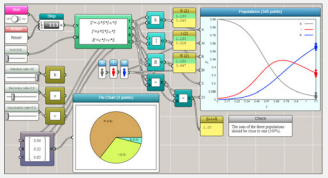
Math: Parametric Surfaces



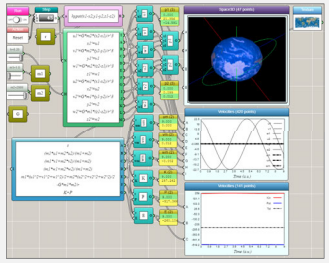
Physics: Projectile Motion



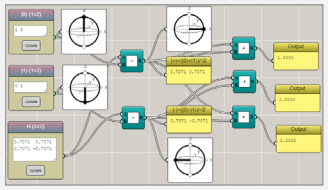
Epidemiology: SIR Model



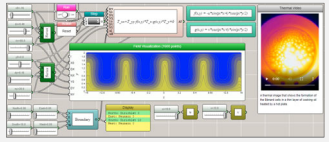
Physics: Satellite Orbits



Quantum Computing



Engineering Computing



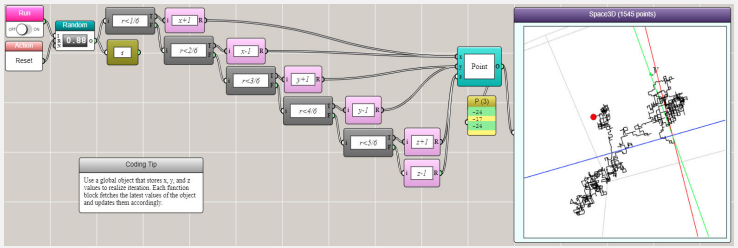
This project is supported by NSF under grant number #2131097. Any opinions, findings, and conclusions or recommendations expressed in this material, however, are those of the authors and do not necessarily reflect the views of NSF.

Line-Based Coding vs. Graph-Based Coding

JavaScript

```
let i = 0
let r
while (i < 5000) {
 r = Math.random()
 if (r < 1/6) {
  x = x + 1
 } else if (r < 2/6) {
  x = x - 1
 } else if (r < 3/6) {
  y = y + 1
 } else if (r < 4/6) {
  y = y - 1
 } else if (r < 5/6) {
  z = z + 1
 } else {
  z = z - 1
 }
 drawPointAt(x, y, z)
 i++
}
```

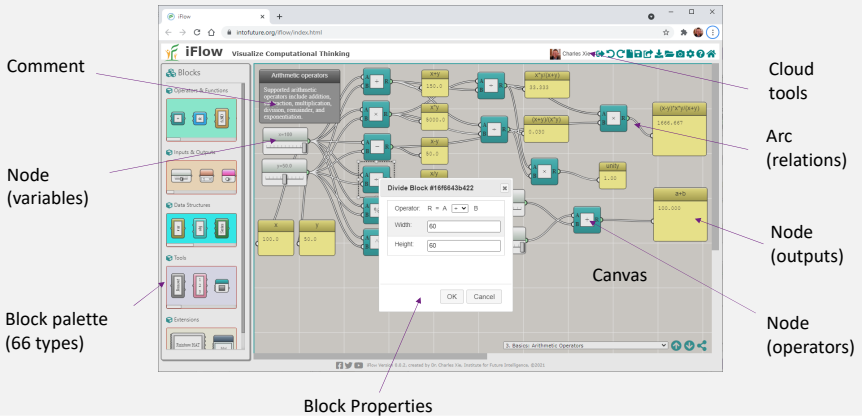
iFlow



An iFlow implementation of the 3D random walk model. Compared with the JavaScript version to the left, an iFlow program visualizes and animates the logic, dataflow, and result of computation in a single workspace (it helps to imagine such a program as a "live" flowchart that can be executed right away to produce results). The if-else branches are displayed as a decision tree, the active connectors are highlighted as white and the inactive ones as gray to indicate the branches in which the data currently flow, and the result is readily rendered in a 3D line plot linked to the decision tree and dynamically updated as the program evolves.

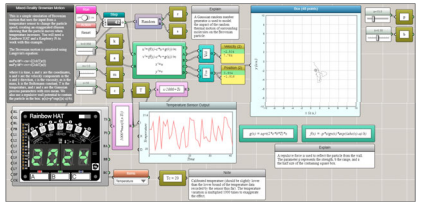
Model relationships, not procedures

User Interface

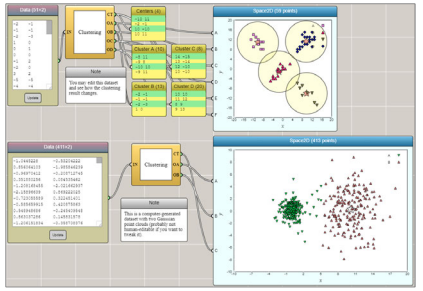


Extensions

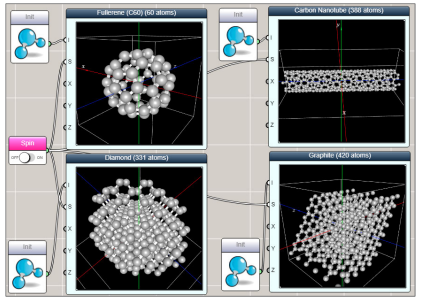
Cyber-Physical Computing



Machine Learning



Cheminformatics



Web Link

https://intofuture.org/iflow.html