Digital Abacus Games

The Digital Abacus is research and development project exploring alternative representation systems for arithmetic and algebra. Our long term vision is to provide powerful, formally correct, declarative programming environments for working with numbers, both for math education and for practical use.

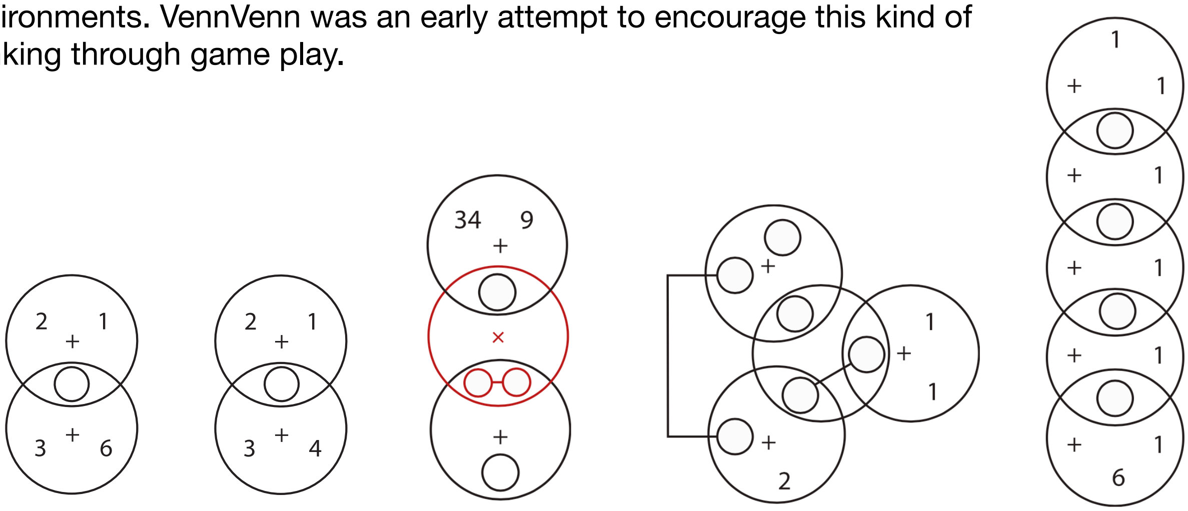
In the *shorter term*, however, we are interested in exploring these systems from a recreational point of view. How might a person first engage with our systems? What kinds of games, puzzles, activities or curricula can be extracted from or inspired by these systems? The current focus of our grant is to investigate these questions.

This document presents the state of these explorations as of Spring 2022.

VennVenn:

VennVenn is a “fill in the blanks” puzzle concept, loosely inspired by Sudoku and KenKen. See this document for instructions, example puzzles and a collection of rules of inference.

The objective of these puzzles is to get the player to “think relationally” about numbers, *eg* to think of 2, 3 and 5 as numbers *related* by addition, rather than thinking of addition as a procedure that takes 2 and 3 as inputs and outputs 5. This is a subtle but important shift in perspective that is at the heart of the Digital Abacus programming environments. VennVenn was an early attempt to encourage this kind of thinking through game play.



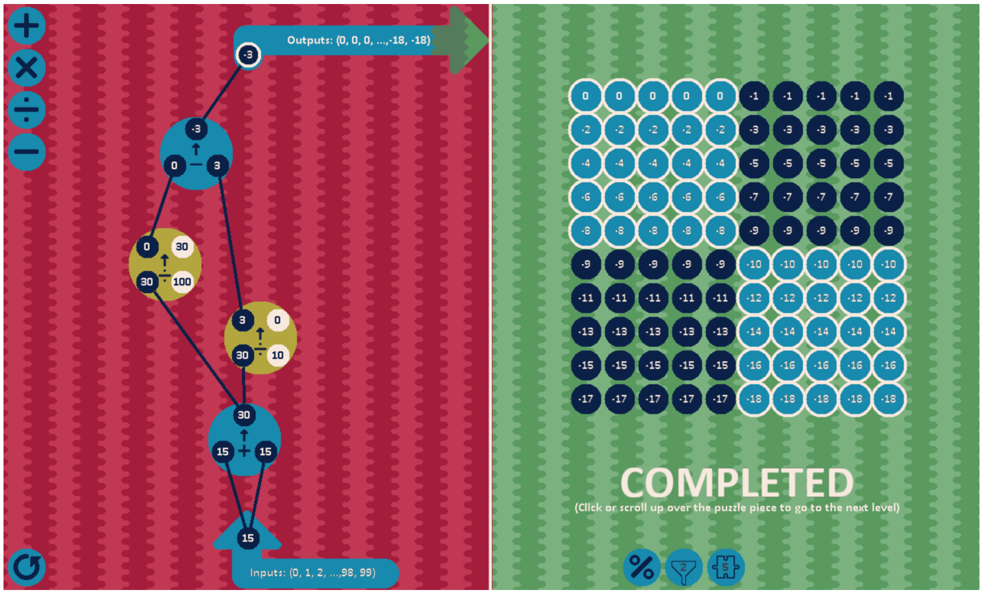
Example VennVenn puzzles.

Le Slo’s Circuits game:

In this game, the player uses a “boxes-and-wires” style programming environment to build arithmetic functions that light up a grid of dots, essentially “programming” different visual patterns.

One of the key concepts explored in this prototype is the “repeater” tool, a component which lets you iterate a function a specified number of times, allowing for rich mathematical exploration.

See this document for links to playable prototypes, and a more in depth presentation of the gameplay and mathematical content.


Outputs: (0, 0, 0, ...,-18, -18)

Inputs: (0, 1, 2, ...,98, 99)

0	0	0	0	0	-1	-1	-1	-1	-1
-2	-2	-2	-2	-2	-3	-3	-3	-3	-3
-4	-4	-4	-4	-4	-5	-5	-5	-5	-5
-6	-6	-6	-6	-6	-7	-7	-7	-7	-7
-8	-8	-8	-8	-8	-9	-9	-9	-9	-9
-9	-9	-9	-9	-9	-10	-10	-10	-10	-10
-11	-11	-11	-11	-11	-12	-12	-12	-12	-12
-13	-13	-13	-13	-13	-14	-14	-14	-14	-14
-15	-15	-15	-15	-15	-16	-16	-16	-16	-16
-17	-17	-17	-17	-17	-18	-18	-18	-18	-18

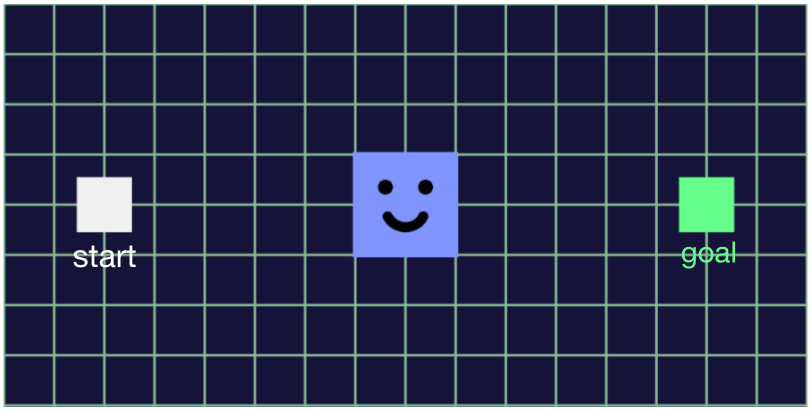
COMPLETED

(Click or scroll up over the puzzle piece to go to the next level)

Screenshot of a completed level in Le Slo’s Circuits game.

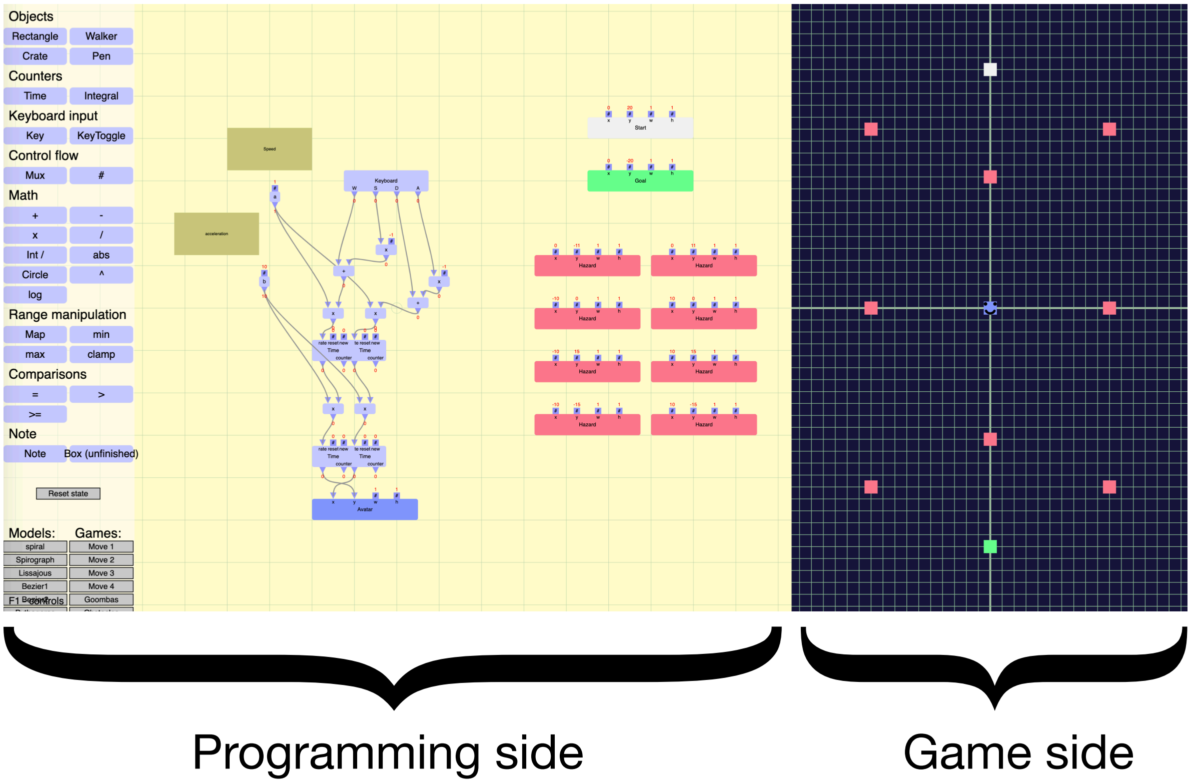
Patrick’s Rectangle game:

This game also uses a “boxes-and-wires” style programming environment, this time for controlling the position and dimensions of rectangles. The player designs control systems that allow a blue rectangle character to navigate a set of obstacles.

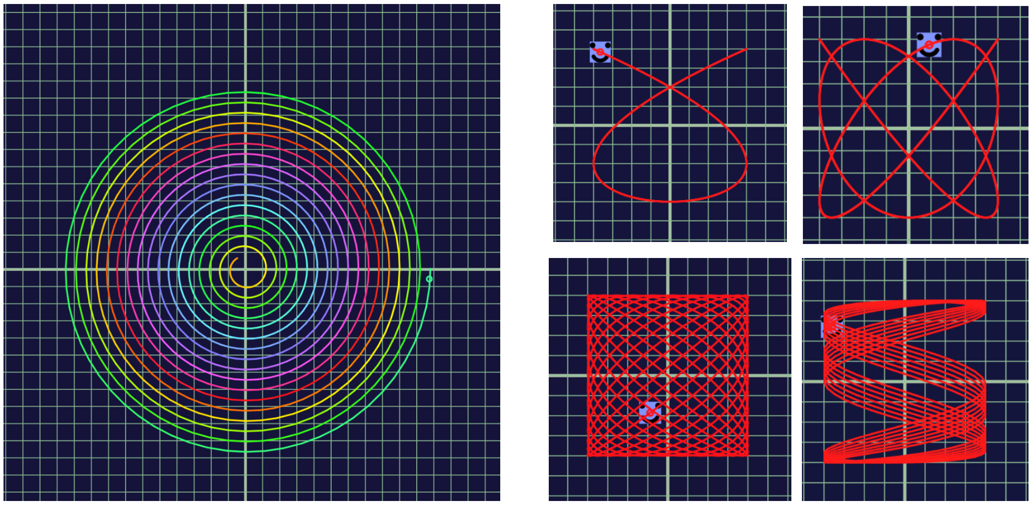


“Main character” from Patrick’s Rectangle game.

See this document for links to playable prototypes and a more in depth presentation of the gameplay and mathematical content.



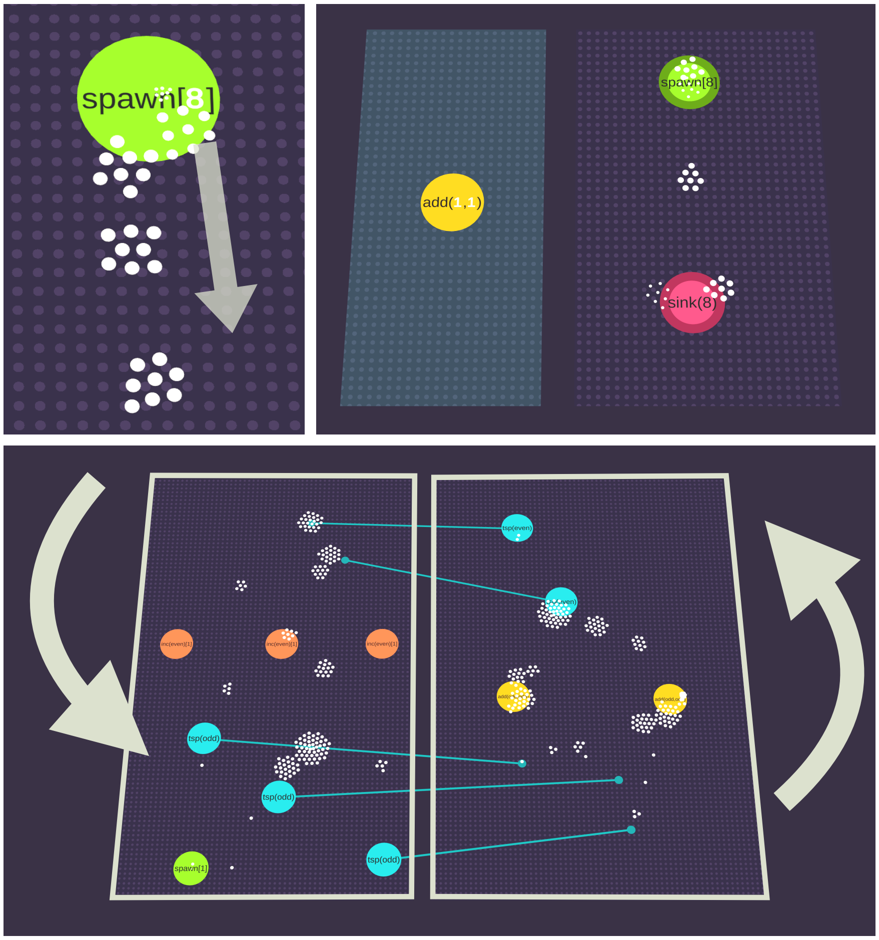
One of the key concepts explored in this prototype is the “time” tool, a component introduces dynamics into the game. Combined with the pen tool and keyboard commands, this game can also function as an open ended programming environment for making graphics and animations.



Programming graphics in Patrick’s Rectangle game.

Polka Dots:

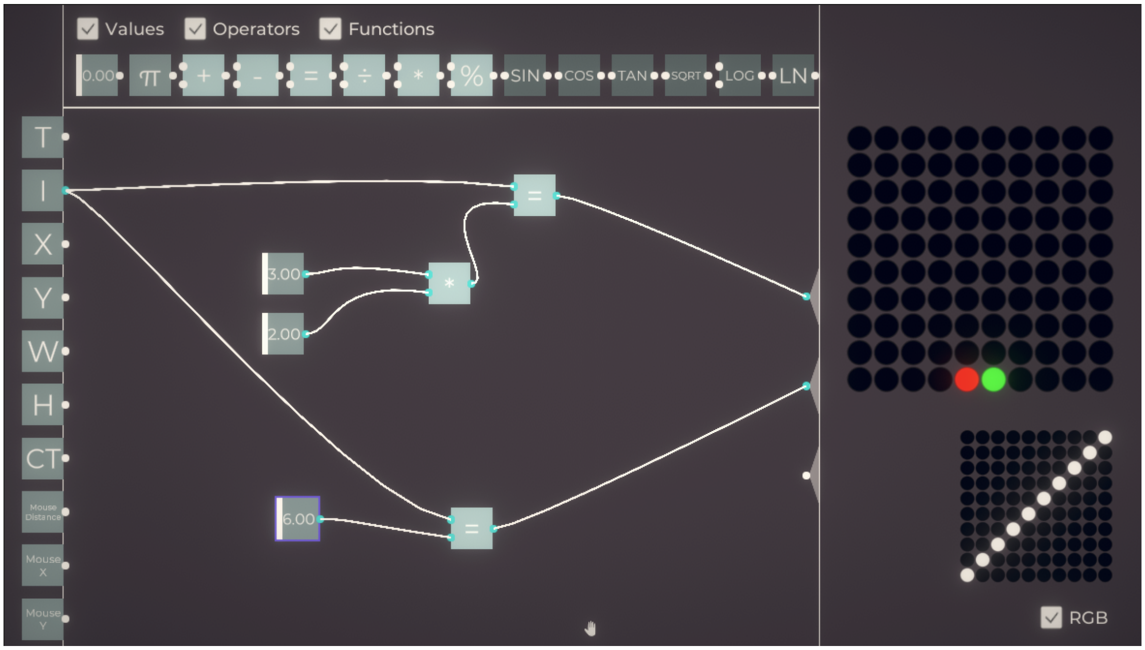
Polka Dots is a factory style puzzle game in which clusters of dots emerge from “Spawners” and disappear into “Sinks”. The player must manage these swarms of dots by arranging components which merge clusters, increase or decrease their size, duplicate, split, transport and filter them based properties like evenness or oddness.



Screenshot of Polka Dots components and levels.

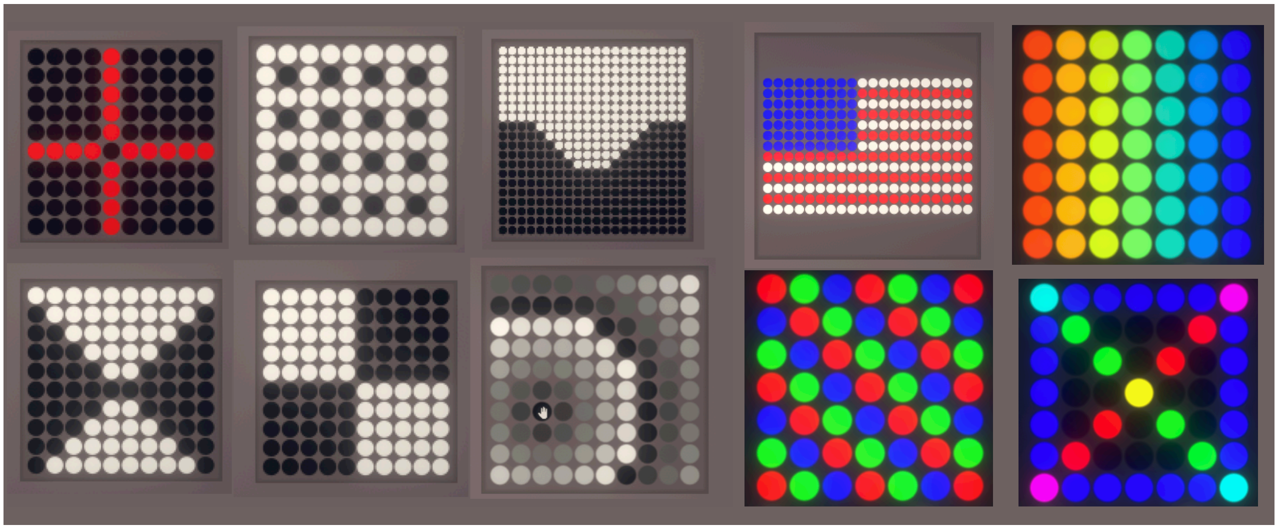
Polka Dots is still in early development. Please contact pdancstep@gmail.com for latest build.

Shader game:

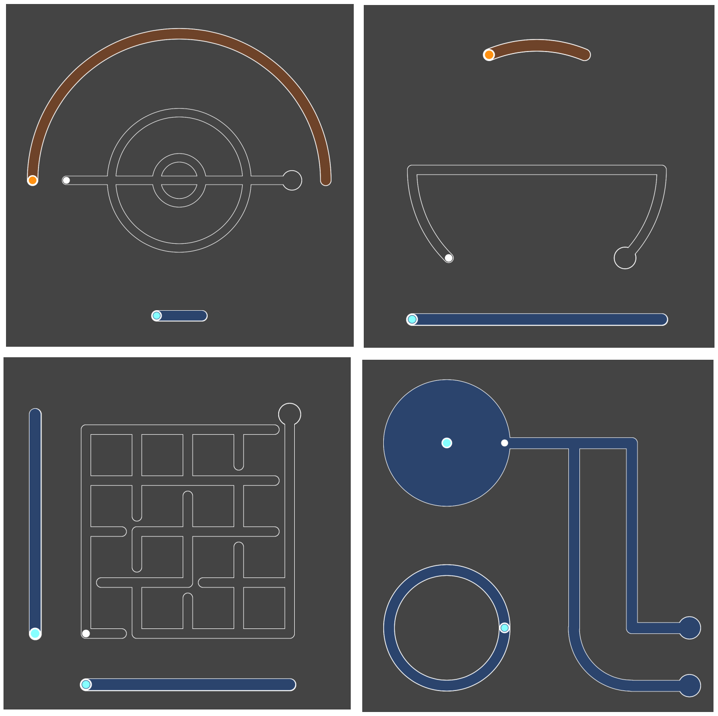


This game uses a node-based system to light up RGB pixels in an array. Each level gives the player an image which they try to reconstruct. The constraints are taken from shader programming, in which every pixel is memoryless and blind to its neighbors. The game progression combines concepts from boolean logic and color theory, and ends in a sandbox mode where players can author their own images and animations.

Shader game is undergoing play testing and polishing and will be publicly available on itch.io at the end of April.



Programming challenges from Shader game



Early prototypes.

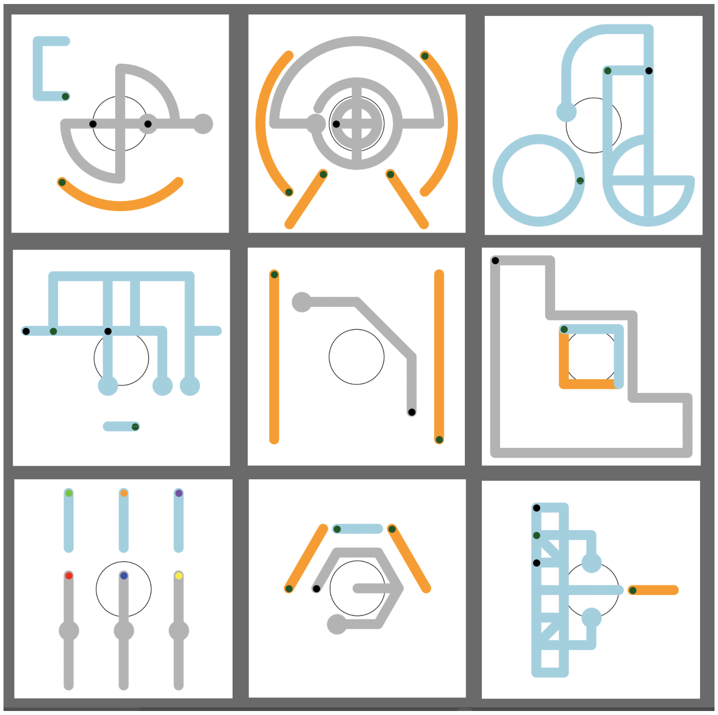
Marble game:

In this geometric puzzle game, players use sliding, scaling and rotating transformations to try to navigate a marble through a maze. Some early prototype levels can be played here:

http://digitalabacus.org/Linkages/0FreePlay

http://digitalabacus.org/Linkages/Game1
http://digitalabacus.org/Linkages/Game2
http://digitalabacus.org/Linkages/Game3
http://digitalabacus.org/Linkages/Game4
http://digitalabacus.org/Linkages/Game5
http://digitalabacus.org/Linkages/Game6

(The last one has no constraints, but hopefully the idea is clear.)



Example levels in our custom level editor.

We've created about 50 levels for this game and worked out most of the basic puzzle types. We do not currently have a standalone build for these puzzles. Please contact pdancstep@gmail.com for access and instructions.

Digital Abacus toolkit

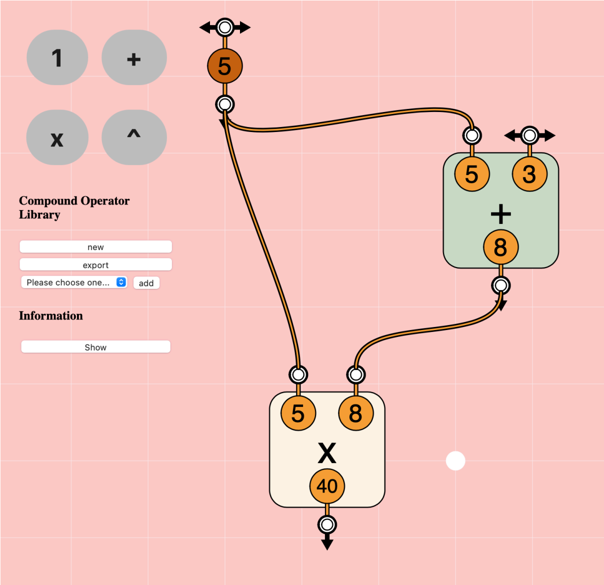
In addition to these games, we are also developing tools for doing algebra. Below are links to two systems currently in development, which are the inspiration for the above games.

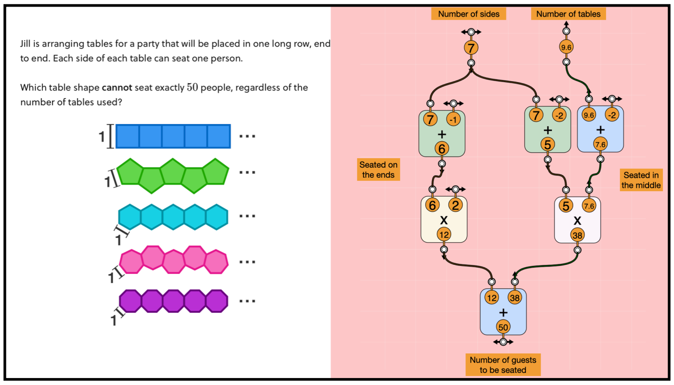
We are still working on instructions and curricular activities for these systems, so they are presented here "as is." Please contact pdancstep@gmail.com for a demonstration of how these systems can be used to solve algebra problems.

Circuits System:

The Circuits System is a node-based numerical programming environment in which all operators are "reversible" – meaning the user can flexibly switch which numbers play the roles of input and output to a given expression.

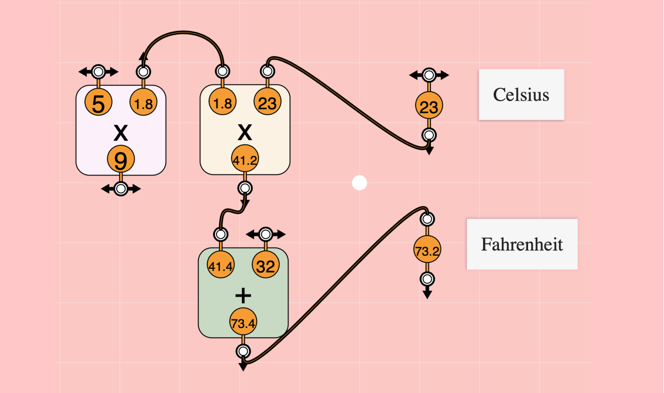
Here's a link to the latest demo. This system is under active development, with several improvements and new features coming soon.





Jill is arranging tables for a party that will be placed in one long row, end to end. Each side of each table can seat one person.

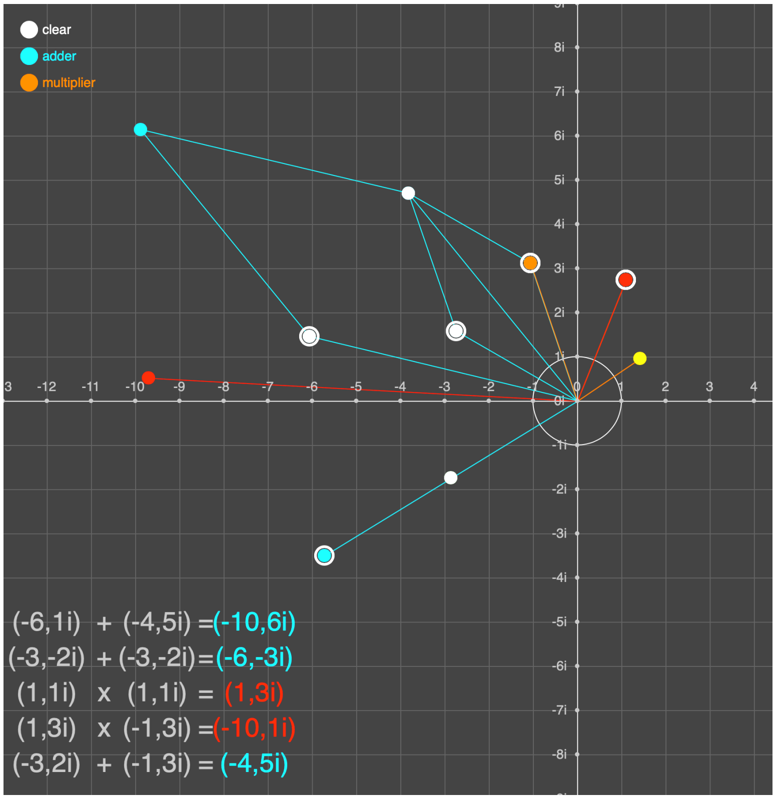
Which table shape **cannot** seat exactly 50 people, regardless of the number of tables used?



Screenshots of the Circuits System, including problem solving (upper right) and modeling (Celsius to Fahrenheit converter, lower right).

Linkages System:

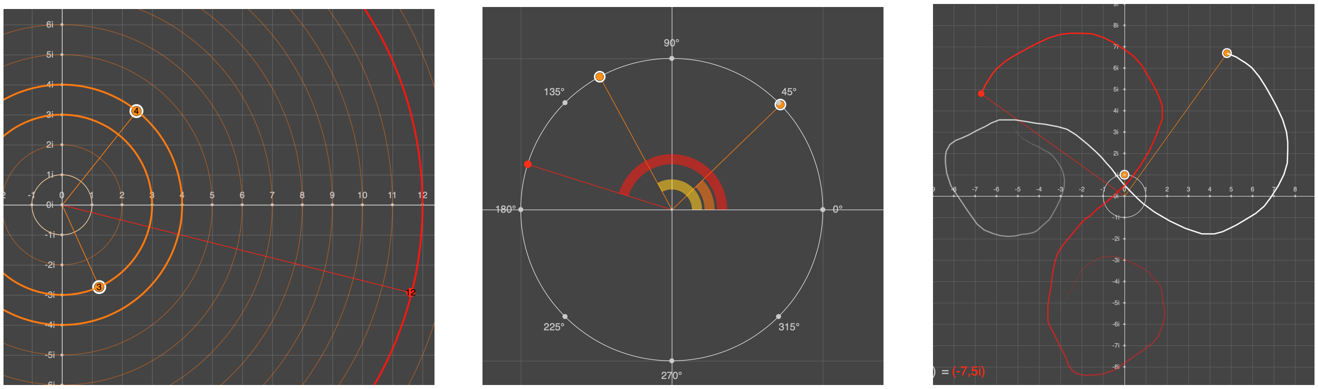
This is a geometric representation system for complex valued functions, also organized around the feature of “reversibility.” This system can be built up from the kinematic ingredients explored in the Marble game.



(-6,1i) + (-4,5i) =(-10,6i)
(-3,-2i) + (-3,-2i)=(-6,-3i)
(1,1i) x (1,1i) = (1,3i)
(1,3i) x (-1,3i) =(-10,1i)
(-3,2i) + (-1,3i) = (-4,5i)

Here’s a link to the latest demo. Some basic instructions:

- Use the buttons on the upper left to place operators on the board
- Press-and-hold two nodes to bind them together.
- Double click a node you cannot control, and you can take control from another (turning an adder into a subtractor etc.)
- Use p/n to see other settings for the Linkages system.



Various modes, visual aids and drawing tools that can be turned on in the Linkages system.