Roundup: Three “programming” board games that will tickle your pseudocode

Enlarge / Programming aspects are all over boardgames these days, even on *Fireball Island*.
Charlie Theel

reader comments

17 with 17 posters participating

Share this story

Welcome to Ars Cardboard, our weekend look at tabletop games! Check out our complete board gaming coverage at cardboard.arstechnica.com.
“Programming” in board games—where you lay down instructions in advance and then execute them in order—has shown up in everything from 1984’s *RoboRally* to 2014’s *Colt Express*. Several of the most interesting games I’ve played this year rely almost entirely upon this mechanic, which makes the games fun to play while also introducing the concept of programming to young players who might have little to no previous exposure to it.

Here are three of my recent favorites.

Lovelace & Babbage

Scott Almes’ *Lovelace & Babbage* (buy on Amazon) is the most interesting programming game I’ve seen in some time, as it feels like a real board game while asking players to do math on the fly. Granted, the “math” is just arithmetic, but the

game actively rewards players who are quick at mental calculation, and the constraints it imposes force everyone to think in different terms about how to get from point A to point B.

The game takes its name from Ada Lovelace (the woman commonly credited with writing the first computer algorithm) and Charles Babbage (who designed some of the first computing machines but never saw them completed in his lifetime). Play as one of those two characters (or choose from two others), trying in each round to get from your starting two-digit number to a different number, using up to five steps. To do this, you can use the allowed operations prescribed on the central board—things like "add/subtract 10" or "multiply/divide by 2" or "reset the number to 55."

As the game progresses, the board gains more operations, but these no longer include round numbers, so you might end up with instructions like "add/subtract 37." (You can't go above 99 or below 0; any such instruction just stops at those figures, rather than returning an error.) You're encouraged to use these harder instructions because you can gain a few extra points for doing so, while the round-number instructions don't give you any such rewards.

Lovelace & Babbage makes this even more challenging in two ways. First, it introduces a timed element, where the clock starts ticking for all other players once the first player has

reached their goal for the round. Second, players can gain additional bonuses for hitting different numerical targets along the way from their start figure to their goals.

Enlarge / The game components—the “instructions” are on the

center board.
In each round, multiple “patron” cards each show one two-digit number and two symbols; the first player to get that number after any intermediate instruction in their code gets to claim one of the two symbols, while the other remains available for another player (except in two-player games). At the end of the game, players score for having acquired the most of each available symbol, for hitting their goals in each round, and for using the more complicated commands available in the game’s second through fourth rounds.

The combination of arithmetic and programming reminded me of my first encounter with coding back in elementary school, using the now-archaic BASIC programming language to write all kinds of useless but very gratifying bits of code. The mere idea of getting a computer to do something because I wrote a series of commands was enough to get me interested in coding, a skill I worked on until my early 30s. *Lovelace & Babbage*’s mechanic, as with all the programming games I mention here, asks players to work in pseudocode, where you don’t need the programming language itself but write out a general set of instructions (an algorithm) that you can later translate into the language of your choice. Games take under a half an hour and, although the box says this one is for ages 14+, you can play with any child who’s able to handle arithmetic with two-digit numbers.

Quirky Circuits

Quirky Circuits (buy on Amazon) is a cooperative programming game in which players alternate placing instruction cards to the table, but the cards themselves are kept face-down until everyone has played at least one card. Using the Adventure Book format from the company’s other games (*Stuffed Fables*, *Comanauts*, and the new *Aftermath*), *Quirky Circuits* features a one-game puzzle on each page. To solve it, players must issue commands to a robot to move it around a room, picking up

certain items while avoiding other obstacles. Instruction cards can be as simple as move forward/backward one to three spaces, turn left/right, or rotate 180 degrees. But certain robots—the game comes with four—have unique functions or default moves, so the deck of cards you use will change each game and may include instructions specific to each robot.

When an instruction card is face-down, all that players can see is what *type* of command it shows—a forward/back move, a rotation or turn, or a pick-up-and-drop action. Players can't communicate about what cards they've played, nor can they tell other players what cards to play. (You can tailor this to the skill levels of younger players, though; I found it useful to ask my daughter out loud to think about what moves she thinks I played, and let her talk through it, without telling her explicitly what cards I'd played.) Once all players have played at least one card and there are at least five instructions played in total, the players flip over all cards and execute the commands in order. After each round, the robot's battery level drops by one. The goal is to finish the robot's task, usually involving picking everything up and returning to the start space, before its battery reaches zero.

The game's core challenge, trying to anticipate what instructions other players will play, focuses on deductive reasoning, which I found more akin to debugging than to

straight programming. The process after each round is very clearly a debugging process, where all players get to discuss what went wrong and how to fix it the next time around. The game’s first few challenges are straightforward and are clearly designed like a tutorial to get players used to the idea of programming moves and to making errors, since you quickly encounter boards where an incorrect move sets you back (for example, “breaking” an object that must then be cleaned up).

https://arstechnica.com/?p=1591793