Online Coding Games Have Their Shortcomings

Brief

Online, apps, gamified coding activities and physical coding toys are valuable tools for getting kids excited about coding. However, they aren't robust enough, on their own, to prepare kids for the rigors of real-world computing. By design, these tools are limited, which means the lessons kids can learn from them are incomplete. The games are an excellent introduction to coding concepts, but they cannot replace curriculum based instruction, as without the greater contexts of science, math, and engineering, gamified coding is just a game.

Coding education professionals recognize that there's tremendous value in teaching children to code, and starting them on the path as early as possible. It's an endeavor that pays dividends throughout their lives and positions them for exciting, high-demand, lucrative careers.

That's why we will forever be indebted to code.org and its venerable "Hour of Code" initiative. This program, which started in 2013, now reaches hundreds of millions of children around the world, demonstrating that anyone can learn to code. It established learning to code as an accessible and universal part of elementary school education in the US.

Because the Hour of Code is free and online, it's accessible to a wide range of students, popularizing Scratch, Roblox, Minecraft, and other browser-based, entry-level programs. These are all effective methods for showing kids that coding can be fun, exciting, and rewarding. However, these programs are only gateways. They can start kids on their journey, but they don't go very far. Eventually, the value peters out if they aren't followed with more in-depth instruction that places the lessons kids are learning in the greater context of a proven curriculum.

As limited as online coding games are they still have value. On top of their role as a gateway into the world of coding, they help refine observational intelligence, and provide a window into computational thinking. Plus they're fun! So by all means get your child involved with the Hour of Code, coding toys, and other online coding opportunities. There's quite a lot they can gain from this exposure.

Why Games Fall Short

The sort of coding gamification that you find through the Hour of Code and related programs, as well as through many popular coding games and toys does an excellent job of sparking the imagination and getting kids exciting about computers and computer programming. Include here are Scratch, MineCraft, Tinker, LittleBits and a myriad of apps.

However, these tools are unable to build Computational Thinking foundations which is the real value of coding for most kids.. Instead, they're just learning to play a game about coding.

As an example, take Minecraft, the popular building and crafting game which lets kids explore a massive virtual world, harvesting raw resources, converting them into more refined materials and

then using these to build massive structures. These structures allow kids to program actions and behaviors, like creating doors that open when players draw near or lights that turn on when night falls in the game.

These are valuable coding lessons because they teach kids logic flow, inputs and outputs, and other basic programming concepts. However, the lessons are trapped in the mechanics of the game. Knowing how to build an automatic trap door in Minecraft doesn't grant you the ability to do that anywhere else. It doesn't teach you real-world programming, only game-world programming. The game teaches valuable concepts but omits any practical knowledge.

This problem extends into nearly every gamified coding activity. Kids will develop skills, but those skills are inextricably tied to the game they're playing or the toy they're using. They become very good at playing a game that someone else created, but they wouldn't be able to create a game of their own. This sort of one-dimensional instruction doesn't supply the needed context for kids to see how the coding skills they're learning in the game can extend into the real world.

Just don't rely on these tools for your kid's entire coding education. If you do you're doing them a disservice. Without real-world, multi-pronged exposure to the science, the language, and the modes of thought that lie behind the games, your kids will be great game-players but not much else.