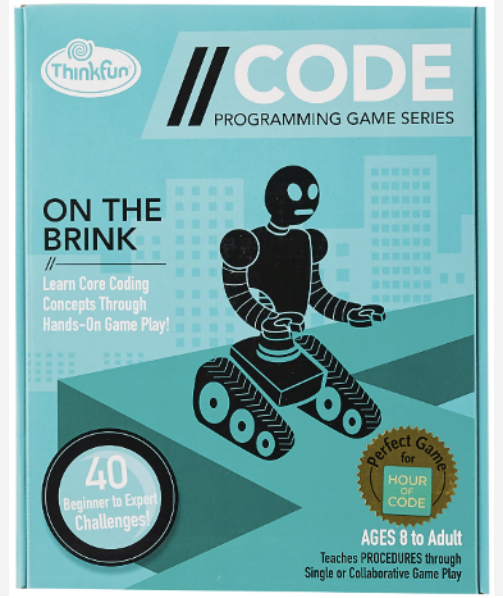
CODE: ON THE BRINK

PROGRAMMING GAME SERIES



A logic-based coding game where players think like programmers, using step-by-step sequencing and problem-solving to navigate tricky paths–without a single line of code!

Overview:

Code On The Brink is a hands-on, logic-based coding game that introduces students to computational thinking and algorithmic problem-solving without requiring a computer. Players use movement tiles to create step-by-step sequences, mimicking real-world programming logic. The game builds foundational skills in sequencing, pattern recognition, and debugging.

Time to Play:

- Approximately 20 - 30 minutes per session
- Can be played as a one-time activity or as part of an extended coding unit

Objective:

Students develop and refine algorithms to navigate a path using predefined movement tiles. The goal is to correctly program the path while reinforcing logical thinking and problem-solving strategies.

PURDUE UNIVERSITY | Department of Computer Science

Learning Goals:

Computational Thinking– Develop skills in sequencing, loops, and conditional logic.

Algorithmic Problem-Solving– Strengthen understanding of structured problem-solving methods.

Debugging & Troubleshooting– Learn how to identify and correct errors in logical sequences.

Pattern Recognition– Improve critical thinking by analyzing movement patterns and optimizing solutions.

Teaching Ideas

- **Unplugged Coding Challenge**– Have students create and test algorithms using movement tiles on different paths.
- **Pair Programming Simulation**– One student writes a sequence, and their partner executes it, identifying errors or inefficiencies.
- **Design Your Own Level**– Students create their own movement challenges for peers to solve.
- **Real-World Coding Connection**– Discuss how step-by-step instructions are essential in programming and robotics.

Suggested Classroom Adaptations

- **Individual Play**– Great for independent practice in logical reasoning.
- **Small Group Collaboration**– Encourage teamwork by having students refine each other's sequences.
- **Coding Relay Race**– Teams work in rounds to complete different levels as quickly and accurately as possible.

Next Steps

- Consider integrating this game into a larger **coding fundamentals unit**, pairing it with introductory block-based programming lessons.
- Encourage students to reflect on debugging techniques and how they relate to real-world programming challenges.

CSTA Standards:

Algorithms & Programming

- **2-AP-11**: Create programs that use variables to store and modify data.
  - *If paired with coding activities, students could explore how movement sequences can be represented as variables in programming.*
- **2-AP-12**: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
  - *Students analyze logical sequences in the game and refine their paths, mirroring debugging in real-world programming.*
- **2-AP-13**: Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.
  - *The game challenges students to break down larger movement paths into smaller, manageable steps, simulating how programmers decompose complex coding problems.*
- **3A-AP-21**: Evaluate algorithms and their efficiency, considering trade-offs, and propose refinements.
  - *Encourages students to optimize their movement sequences and find the most efficient solutions, reflecting real-world algorithm optimization.*

**Connection to Real-World Computing

The On The Brink game models the logical thinking required in real-world programming and robotics. Just like software engineers break down problems into step-by-step sequences, students must carefully plan and adjust their movement algorithms. The game's debugging process mirrors the way coders identify and correct errors in software development. These skills are foundational for careers in computer science, automation, and AI, where precision and structured problem-solving are critical.