ThinkFun®

HACKER™

CYBERSECURITY LOGIC GAME

INSTRUCTION MANUAL

AGES 10 TO ADULT

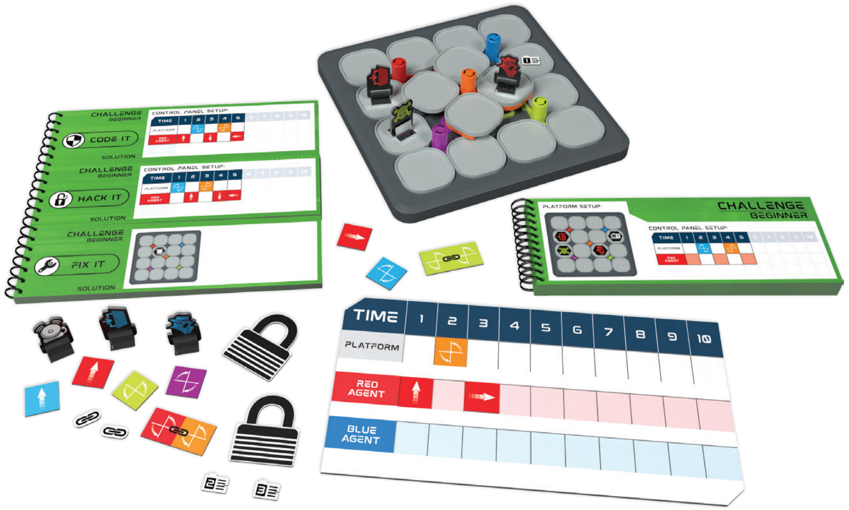
SINGLE OR COLLABORATIVE
UNPLUGGED GAME PLAY

HACKER INSTRUCTIONS

INTRO

Train to become a member of the elite white hat hacker team Oblivion. At their hidden training facility, you will be tasked with finishing forty (3-part) simulations.

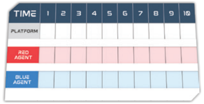
Think like a hacker to complete each simulation and join team Oblivion on their mission to thwart cybercriminals!



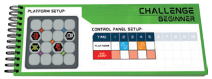
Hacker is a unique programming game with many novel elements. To gain a clear understanding of the three different phases, as well as the new components and goals that are introduced at each level, we encourage you to go to **www.thinkfun.com/learn-coding** for How to Play videos, strategy tips, and more!

INCLUDES:

Game Grid: The training center is a 4x4 grid of 16 tiles. The tiles rest on 5 rotating platforms, 2 that rotate clockwise and 3 that rotate counter-clockwise.

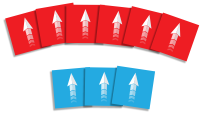


Control Panel: Holds the tiles that control the movements of Red Agent, Blue Agent, and each of the five Revolving Platforms.

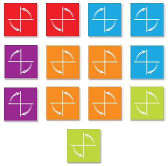


Challenge Booklet: Contains the parameters for completing each of the 40 (3-part) coding simulations.

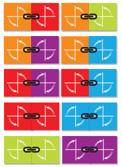
2 Agent Tokens: Red Agent appears in all 40 simulations, and Blue Agent appears in the final 20 simulations. Each Agent moves according to the instructions of their corresponding Movement Tiles.



9 Movement Tiles: Placed in the Control Panel; instruct the Agents to move forward, backward, left, and right around the Game Grid.



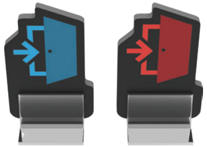
13 Revolving Platform Tiles: Instruct the Revolving Platforms to rotate 90 degrees (in the direction of the white arrows on the Tile).



5 Double-sided Transaction Tiles: Revolving Platform Tiles that cannot be separated during the HACK IT phase.

Transaction Link Token: “Links” together two Movement Tiles during the Intermediate level FIX IT phase.

3 Data File Tokens: Must be picked up in numerical order during the CODE IT phase.



2 Exit Point Tokens: The end locations during the CODE IT phase.

Virus Token: The end location during the HACK IT phase.

Alarm Token: Used in the Beginner and Advanced FIX IT phases. When an Agent touches the Alarm, the program ends safely, foiling an attempted hack.

2 Lock Tokens: Appear in the Expert Challenges only. Restrict both Agents from being in the same row or column at one time.

Solution Booklet*: Divided into three sections to reveal solutions one phase at a time.

*For solutions to Challenges 61–120, flip the book upside down!

HOW TO PLAY:

Each of the 40 simulations consists of three interconnected phases:

Phase 1 - CODE IT: Program the Agents to pick up Data Files and reach their corresponding Exit Points.

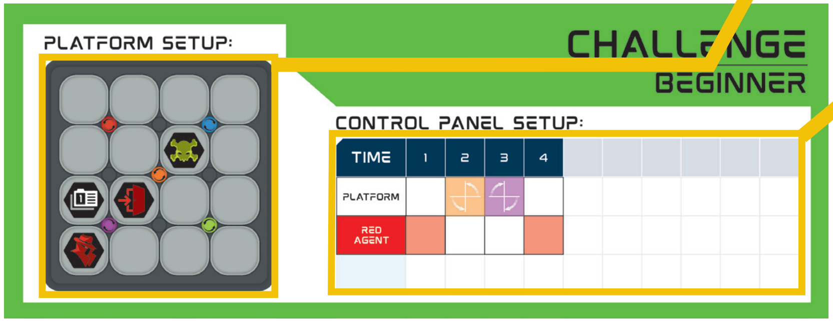
Phase 2 - HACK IT: Analyze the program you wrote in Phase 1, looking for security vulnerabilities. Discover how a hacker could alter your program and infect it with a Virus.

Phase 3 - FIX IT: Use your knowledge of weaknesses in the computer systems to secure your program. Prevent a devastating cyberattack with Alarms, Transactions, and Locks.

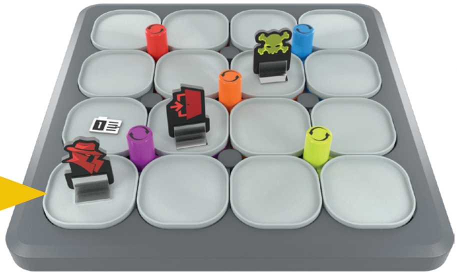
See the Phases demonstrated at **www.thinkfun.com/learn-coding**

SETUP

1. Remove all pieces from the punchboards. Place the Agents, Exit Points, Virus, and Alarm Tokens into the token stands.
2. Select a Challenge from the Challenge Booklet.

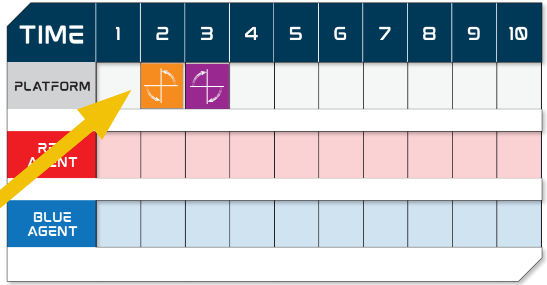


3. Place Tokens on your Game Grid to match the illustration shown under PLATFORM SETUP.



NOTE: Place Data File Tokens flat on the Game Grid.

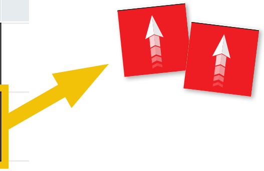
4. Place Revolving Platform Tiles on your Control Panel to match the CONTROL PANEL SETUP.



5. Count the number of solid colored squares (red or blue) shown in the CONTROL PANEL SETUP, and gather the same number of Movement Tiles in the corresponding colors.

CONTROL PANEL SETUP

TIME	1	2	3	4
PLATFORM		🟧	🟪	
RED AGENT	🟥			🟥



UNDERSTANDING THE CONTROL PANEL AND GAME GRID:

Each Challenge shows a Control Panel with between four and ten Time Steps.

During each Time Step, an instruction is given to either an Agent (Red or Blue) or one of the five Revolving Platforms through two types of instruction tiles:

A **Movement Tile** will instruct an Agent to move up, down, right, or left one space.

- Movements are relative to the Game Grid itself, and not the Agents.

A **Revolving Platform Tile** will instruct its corresponding platform to rotate 90 degrees in the direction of the white arrows on the Tile.

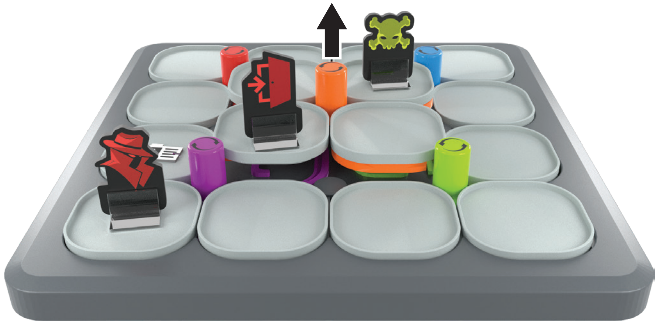
- Any Tokens on a platform will rotate along with the platform.

ROTATING A REVOLVING PLATFORM

See the Platforms rotate at **www.thinkfun.com/learn-coding**

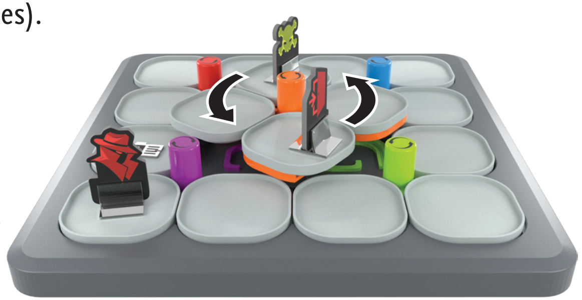
Each of the five Revolving Platforms on the Game Grid is made up of the four tiles surrounding a colored handle.

To rotate a Platform, use the handle to gently lift the platform just above the height of the tiles resting on the Game Grid.

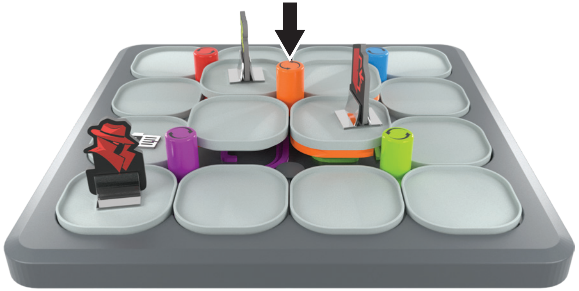


Then, rotate the platform 90 degrees in the direction of the black arrows printed on the handle (these directions match the white arrows on the Revolving Platform Tiles).

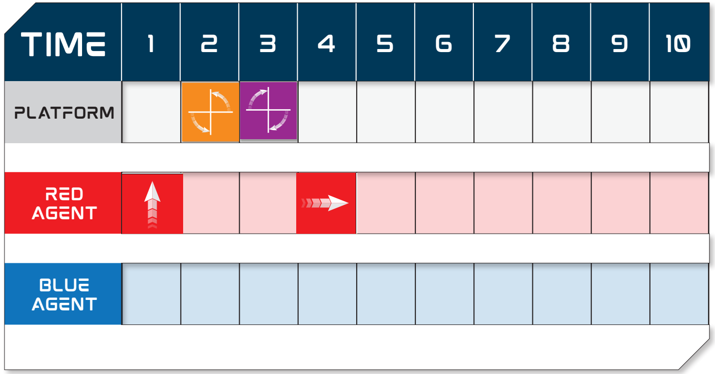
NOTE: Knobs show the direction the platform should rotate.



Lower the Platform so it is once again level with the rest of the Game Grid.

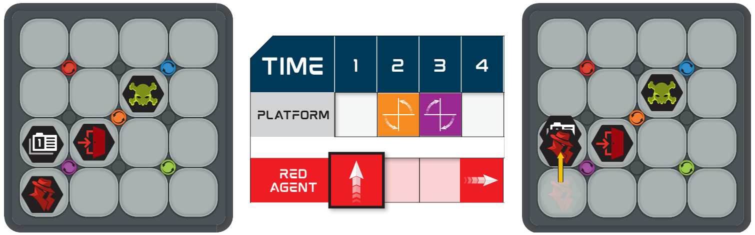


Follow the example on the next page to see how Tokens move around the Game Grid according to the program shown below.



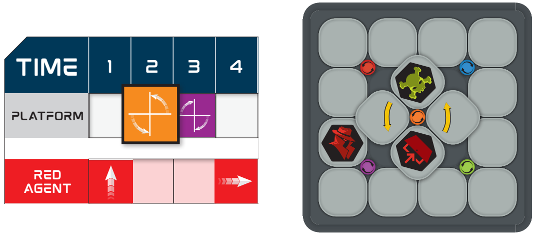
TIME	1	2	3	4	5	6	7	8	9	10
PLATFORM		↺	↻							
RED AGENT	↑			→						
BLUE AGENT										

Starting Position

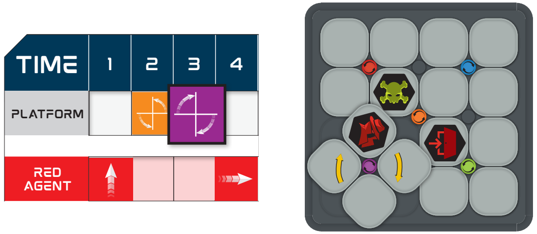


In Time Step 1, ↑ instructs Red Agent to move up one space, picking up Data File #1.

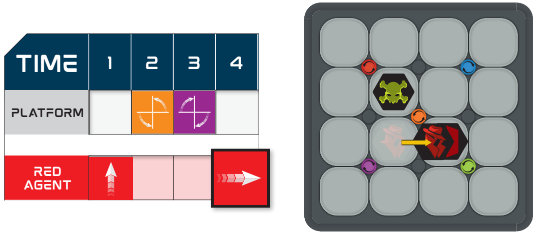
In Time Step 2, 🟧 rotates the center platform and all Tokens on the platform 90 degrees counter-clockwise.



In Time Step 3, 🟪 rotates the bottom left platform (and thereby Red Agent) 90 degrees clockwise.



Finally, in Time Step 4, → instructs Red Agent to move one space right and into the Red Exit Point, completing the program.



BEGINNER CHALLENGES

PHASE 1 - CODE IT:

CODE IT GOAL: Program Red Agent to arrive at the Red Exit Point. Don't let the Agent reach the Virus, or the control systems will be infected!

STEPS TO CODING A PROGRAM:

1. Examine the Game Grid and Control Panel, noting which Platform(s) will rotate, and how many times the Agent will move to a new space.
2. Determine which direction the Agent should move during each time step marked with a solid red square.
   - The Agent must remain on the Game Grid at all times.
   - The Agent must pick up any Data File Tokens in numerical order.
     - The Agent will automatically pick up a Data File when the Agent moves into a space with the lowest-numbered File still on the Grid.
     - An Agent may land on a space with higher-numbered Data Files before lower-numbered Data Files, but the Agent won't pick them up.

When an Agent lands on the same space as the lowest numbered Data File still on the Game Grid, pick up the Data File and add it to the Agent Token as shown.

3. Add Movement Tiles to the Control Panel to program the Agent's path.
   - Point the Movement Tiles in the direction you want the Agent to move (forward, backward, left, or right).

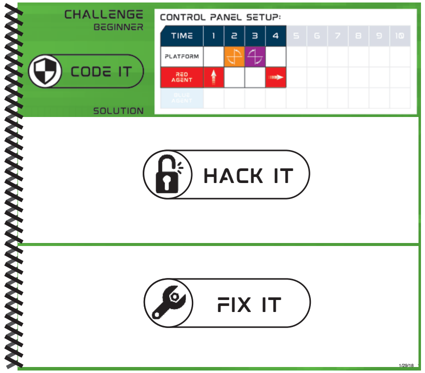
4. Execute your program, moving Red Agent around the Game Grid according to the instructions on the Control Panel. When the Agent picks up the Data Files in numerical order, and lands at the Exit Point after the last instruction – **YOU WIN!**

CHECKING YOUR SOLUTIONS: You can verify your CODE IT solution is correct by carefully executing the program. We recommend double-checking your program against the Solution Booklet because you will need your CODE IT program to solve the next phase, HACK IT.

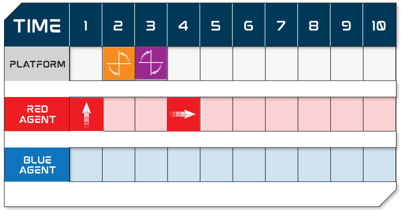
> **NOTE:** The solutions for the CODE IT phase are found within the top third of the Solution Booklet.

Once you have verified the correct solution, **leave your CODE IT program on the CONTROL PANEL.**

Solution Booklet



Control Panel



If your program matches the Solution Booklet, you are ready for Phase 2 - HACK IT.

STOP AND PLAY: You are now ready to play Phase 1 - CODE IT. We recommend you play CODE IT, Challenge 1, then turn to page 13 to learn how to HACK IT.

PHASE 2 - HACK IT:

HACK IT GOAL: Think like a hacker and alter the program from the CODE IT phase so that Red Agent reaches the Virus Token, infecting the control systems. Only by discovering the hacker's plan can you hope to prevent the attack in the final FIX IT phase.

STEPS TO HACKING A PROGRAM:

1. Reset the Game Grid according to the original set-up outlined on the Challenge Card. **Leave your CODE IT program on the Control Panel.**
2. Examine the Game Grid and Control Panel and note where the Virus Token is located in relation to the Exit Point Token.
3. Review the "Hacker Constraints" described on pages 14-15, and determine how a hacker could alter your existing program to bring Red Agent into contact with the Virus Token.
4. Alter your program as if you were a hacker, sliding the instruction tiles from your CODE IT program left and right in their respective rows.

   > **NOTE:** Data Files do not affect a hacked program; it does not matter whether they are picked up or not.

5. Execute the hacked program, moving Red Agent around the Game Grid according to the instructions in the Control Panel. When Red Agent arrives at the Virus Token– **YOU WIN!**

CHECKING YOUR SOLUTIONS: The solutions for the HACK IT phase are found within the middle section of the Solution Booklet. Before moving on to Phase 3 - FIX IT, we recommend that you reveal the HACK IT solution while leaving the CODE IT solution visible as well.

Leave your HACK IT program on the Control Panel.

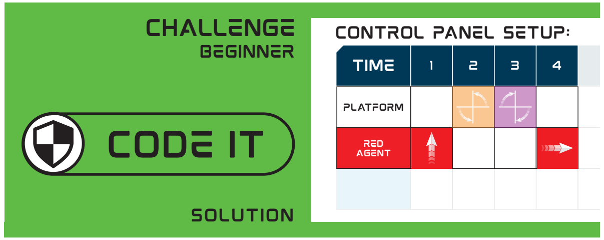
HACKER CONSTRAINTS

Our computer systems prevent hackers from altering programming instructions, but a hacker could potentially change the *timing* of the instructions. By sliding the instructions from your CODE IT program left and right in their respective rows, you will see the damage a hacker could do with this simple power.

IMPORTANT: An Agent may land at the Virus in fewer steps than what was needed to reach the Exit Point. In other words, a HACK IT program may contain fewer tiles than the CODE IT program. In the Solutions for the HACK IT phase, unused Tiles have diagonal lines running through them.

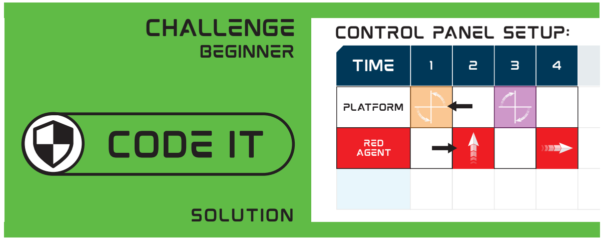
EXAMPLES:

CODE IT program



A Hacker can:

- Slide any of the Tiles left or right within their own rows.

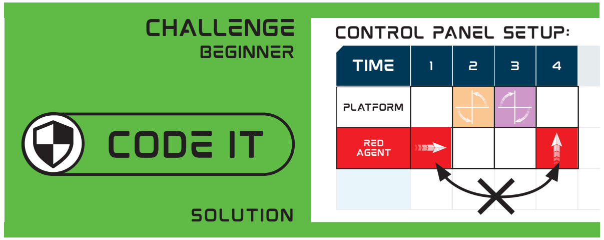


In this example, the 🟧 Instruction is slid one Time Step to the left, and the ↑ Red Agent instruction is slid one Time Step to the right.

A Hacker cannot:

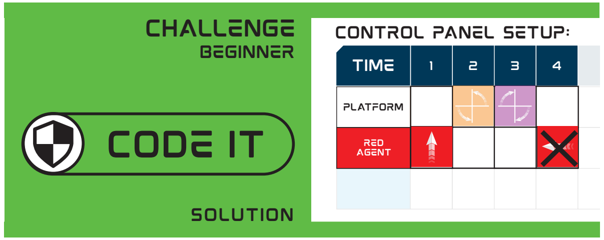
- Alter the *order* of Tiles within a row

If in the Correct Program, a ↑ instruction occurs after a → instruction, a hacker cannot make the ↑ instruction occur first.



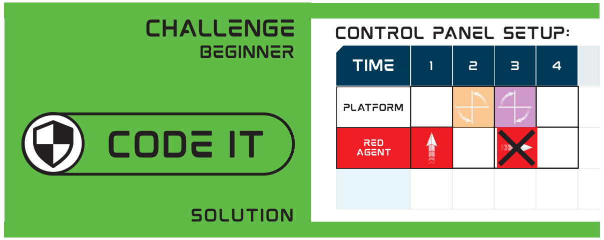
- Alter the *orientation* of the Movement Tiles

If in the Correct Program, the Agent instruction in Time Step 4 is →, a hacker cannot change the instruction to be ←.



- Place more than one Tile in the same time step

A hacker cannot make both a Platform instruction and Agent instruction occur during the same Time Step.



Watch Hacker Constraints explained at **www.thinkfun.com/learn-coding**

STOP AND PLAY: You are now ready to play Phase 2 - HACK IT. We recommend you play HACK IT, Challenge 2, then turn to page 16 to learn how to FIX IT.

PHASE 3 - FIX IT:

FIX IT GOAL: Prevent an attack on your program by placing an Alarm Token in a space that would interrupt the hacked program, safely shutting it down before the Agent can reach the Virus.

STEPS TO FIXING A PROGRAM:

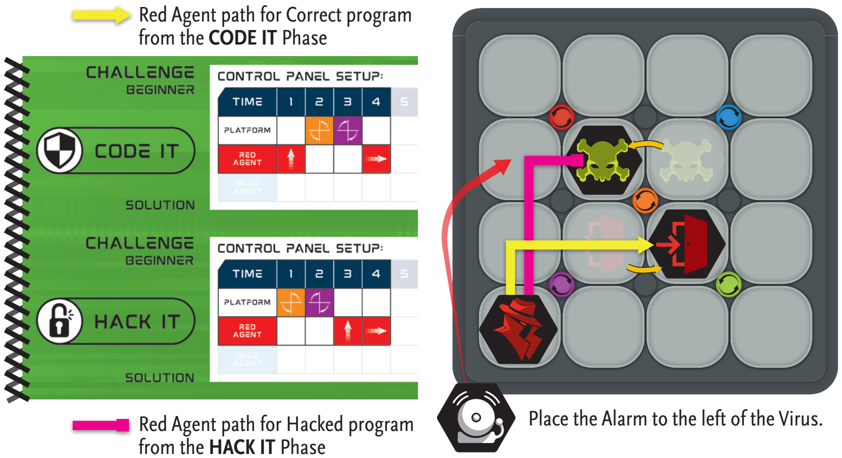
1. Reset the Game Grid according to the original set-up in the Challenge. Leave your HACK IT program on the Control Panel.
2. Examine the Game Grid and Control Panel, noting the empty spaces on the Game Grid and their relationship to the Virus and Exit Tokens.
3. Determine on which of the empty spaces you should place an Alarm Token. When an Agent comes into contact with the Alarm, the Alarm is triggered and the program safely shuts down.
   - When following the hacked program, the Agent should arrive at the Alarm Token before it can reach the Virus Token.
   - When following the original program from the CODE IT phase, the Agent should miss the Alarm completely.

   IMPORTANT: The Alarm is a physical object that rotates along with the Revolving Platforms. The Alarm is triggered when an Agent enters the same space as an Alarm, but this isn't always the same space where the Alarm started.
4. Place the Alarm Token on the Game Grid. If the Alarm foils the hacked program, but does not interfere with the CODE IT program – **YOU WIN!**

TIP: Keep the solutions to the CODE IT and HACK IT Challenges handy! You may want to move your Agent along the CODE IT and HACK IT paths once or twice more in order to determine the best spot for the Alarm.

Watch an Alarm shut down a hacked program at
www.thinkfun.com/learn-coding

TIP: To successfully secure or "FIX" a program, track where the hacked program diverges from the CODE IT program.



Red Agent path for Correct program from the **CODE IT** Phase

Red Agent path for Hacked program from the **HACK IT** Phase

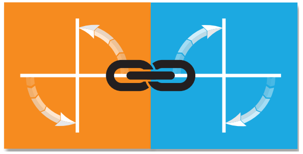
Place the Alarm to the left of the Virus.

STOP AND PLAY: You are now ready to play all of the Beginner Challenges! We recommend you play FIX IT, Challenge 3, followed by the remaining Beginner Challenges.

When you are ready to play the upper level challenges, review pages 17-20. As you progress through the Intermediate, Advanced, and Expert Challenges, new components will be introduced, and the goal of each phase will change slightly.

INTERMEDIATE CHALLENGES

Transaction Tiles & Transaction Links first appear in the Intermediate Challenges. Transaction Links connect two adjacent instruction Tiles in the same row. These two instructions must always occur one after the other (but NOT simultaneously).



Pay close attention! When you see a 🔗 between two Revolving Platform Tiles, use the rectangular Transaction Tiles.

> **IMPORTANT:** Transaction Tiles are doubled-sided, with different rotating platform combinations on each side.

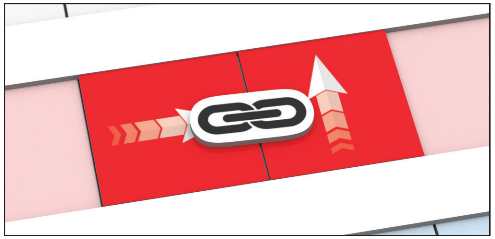
CODE IT GOAL: Program Red Agent to retrieve all Data Files, avoid the Virus, and arrive at the Red Exit Point.

HACK IT GOAL: Alter the CODE IT program to cause the Red Agent to land at the Virus. Remember, two instructions in a Transaction Tile can never be split apart.

FIX IT GOAL: Foil the hack by adding one Transaction Link between two adjacent Movement Tiles in your CODE IT program, linking the moves together.

> **HINT:** Pay close attention to which Movement Tiles you split apart during the HACK IT phase.

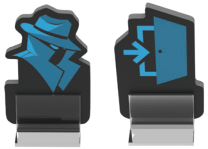
Place the Transaction Link between two Movement Tiles.



> **NOTE:** You will not use the Alarm Token for the Intermediate Challenges.

ADVANCED CHALLENGES

Blue Agent and the **Blue Exit Point Tokens** first appear in the Advanced Challenges.



CODE IT GOAL: Program both Agents to collect Data Files and reach their corresponding Exit Points.

- Either Agent can pick up any Data File, but the Files must be picked up in numerical order (i.e., Red Agent can pick up Data File #1, then Blue Agent can pick up Data File #2)
- Both Agents may be on the same space at the same time.
- Both Agents must avoid the Virus Token.

HACK IT GOAL: Discover what a Hacker could do to destroy both Agents. Alter the CODE IT program so both Agents reach the Virus Token.

> **NOTE:** When one Agent reaches the Virus, it no longer executes any further instructions. However, the program continues executing the other Agent's instructions and any remaining Platform instructions.

FIX IT GOAL: Place an Alarm Token on an empty space on the Game Grid so that at least one Agent triggers the Alarm while executing the hacked program. The Alarm should be triggered before either Agent reaches the Virus.

- The placement of the Alarm must not interfere with the CODE IT program. When executing the CODE IT program, neither Agent should reach the Alarm.

EXPERT CHALLENGES

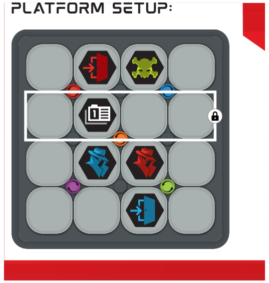
Locks are introduced in the Expert Challenges. A Lock is a kind of alarm that is triggered when both active Agents are together in the same locked row or column.

During Challenge setup, examine the Challenge Card to see which row or column is marked with a Lock. Place the Lock Token near the Game Grid, closest to that row or column.

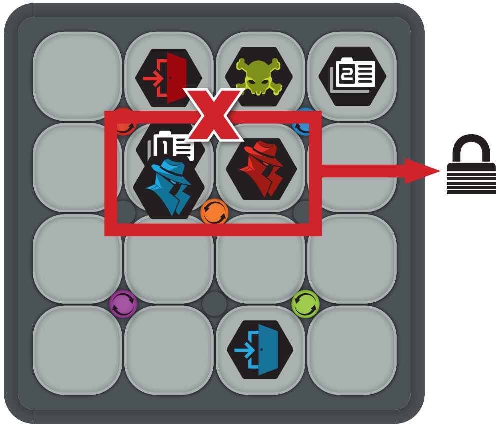
NOTE: One Lock is used during the setup, and the other is used during the FIX IT phase.

EXAMPLE:

Challenge Card



Game Grid



A Lock is on Row 3. If at any point both Agents were in Row 3 at the same time, as shown, the program would shut down.

CODE IT GOAL: Program both Agents to pick up Data Files and reach their corresponding Exit Points. Your program may not cause both Agents to be in the locked row/column at the same time.

HACK IT GOAL: Alter the program so both Agents reach the Virus Token. Your hacked program must avoid having the two Agents in the locked row/column at the same time.

FIX IT GOAL: Place a second Lock on a row/column. The new Lock must trigger the Hacked program to shut down before either Agent reaches the Virus.

- The additional Lock must not interfere with the CODE IT program

TIP: During the CODE IT and HACK IT phases, pay careful attention to every row and column that both Agents occupy simultaneously.

A CLOSER LOOK AT HACKER

ThinkFun's line of unplugged coding games are designed to take players on a grand tour of the big ideas in computer science. In Hacker™, players are introduced to the concept of concurrency.

Programs for computers with multiple processors are usually divided into separate subtasks that can be distributed to the different processors and run at the same time. Subtasks are run in separate "threads," and while the instructions in a single thread are guaranteed to execute in order, it is impossible to predict the exact order instructions from different threads will be executed. Every possible ordering of the instructions from different threads is called an "interleaving." Reasoning about all the interleavings in a program requires clever computational thinking and imagination.

Hacker allows players to reason about interleavings using strategies similar to those used by real programmers. Transactions, first featured in the intermediate challenges, are one way programmers manage the complexity of reasoning about interleavings by ensuring that certain instructions can never be split apart. In addition, in the expert challenges, the locks mimic a strategy used in real concurrency programming, where locks are used to prevent any interleaving that would cause two threads to occupy the same critical section of code at the same time.

Furthermore, the alarm is a metaphor for the way programmers throw an exception when a program enters an unusual state, catching a problem early before it leads to catastrophic failure. Hacker, a Cybersecurity Logic Game, provides a taste of what it is like to analyze a multi-threaded program with a security mindset, thinking through all the ways a program might go wrong.

ABOUT THE INVENTOR

Mark Engelberg is the inventor of Hacker™. To create his growing line of unplugged coding games, which includes Code Master™ and the //CODE Programming Game Series, Mark draws on his experience as a programmer of virtual reality simulations for NASA, as well as his many years of experience as a teacher of computer science and mathematical logic. Mark believes that kids of all ages can and should learn how a computer executes programs - entirely through play!

Mark Engelberg is also one of the challenge developers for ThinkFun's blockbuster game Rush Hour®.

Vesa Timonen contributed to challenge development for Hacker.

Rotating tile transference mechanism developed by Tactrics BV
(EP15182865.4) Used here with permission.

TACTRICS
by Hans Beeftink

VORTEX

NOTES:

ThinkFun's Mission is to Ignite Your Mind!®

ThinkFun® is the world's leader in addictively fun games that stretch and sharpen your mind. From lighting up young minds to creating fun for the whole family, ThinkFun's innovative games and mobile apps make you think while they make you smile.

ThinkFun®

www.ThinkFun.com

MADE IN CHINA, 106. #1920. IN02.