# GridForge — Flashcards

*A Spark & Anvil printable flashcard deck. Quiz yourself on the GridForge cast's curriculum — one card at a time.*

---

## How to use these cards

**Fold** each sheet down the middle line so the answers are hidden. Read the **question** on the left, say your answer out loud, then **unfold** to check the right side. Testing yourself — then checking — is one of the most powerful ways to remember. Get one wrong? That's the best kind of practice: try it again tomorrow.

*Want real flashcards? Cut along the fold line and the rows into cards — question on one side, answer on the other.*



---

# 1. connection

| ✂ Question (fold →) | Answer |
|---|---|
| Watching the opponent's discs lets you: | **spot and block their threats early** — Reading the opponent's developing lines is key defense. |
| A diagonal four-in-a-row goes: | **corner-to-corner through four stacked-and-shifted slots** — Diagonals rise (or fall) one row per column across four columns. |
| You cannot drop a disc into a column that is: | **full** — A full column has no empty slot to accept a disc. |
| Good Connect-4 players think about: | **their threats AND the opponent's at the same time** — You must build your lines while blocking theirs. |
| A vertical four-in-a-row is made by: | **dropping four of your discs into the same column** — Four in one column stacked = a vertical win. |
| Big idea: Connect-4 is a race to: | **connect four while blocking the opponent's four** — Build your line of four first — and stop theirs. |
| Rowan's signature is: | **extending a chain toward the winning connection** — Rowan embodies building the connecting line. |
| If your opponent has three in a row with an open end, you should usually: | **block the open end** — An open three threatens to become four — block it. |
| Three in a row open on BOTH ends is dangerous because: | **the opponent can complete four on either side** — Two open ends = two ways to win; a single block can't stop both. |
| A winning Connect-4 line can be: | **horizontal, vertical, OR diagonal** — Any of the three orientations of four-in-a-row wins. |
| The first player in Connect-4, with perfect play, can: | **force a win (Connect-4 is solved)** — Connect-4 is a solved first-player win with perfect play (kit 04). |
| Because discs stack, the square directly ABOVE a slot only becomes playable: | **after that slot is filled** — You can only reach a higher slot once the ones below are filled. |
| A "threat" in Connect-4 is: | **three in a row that could become four** — A threat is a line one move from completing four. |
| A Connect-4 disc, when dropped into a column, falls to: | **the lowest empty slot** — Gravity pulls the disc to the lowest empty spot in that column. |
| The connection primitive (making a line) appears in Connect-4 as: | **four in a row** — Connect-4's core is building a connected line of four. |
| In Connect-4, you win by making a line of: | **four of your discs in a row** — Four in a row (horizontal, vertical, or diagonal) wins. |
| Because a disc falls to the bottom, you cannot: | **place a disc floating above an empty slot** — Discs can't float; the slot below must be filled first. |
| A column with 5 discs (of 6 slots) has room for: | **one more disc** — A standard column holds 6; five filled leaves one slot. |

| | |
|---|---|
| On your turn in Connect-4 you may: | **drop one disc into any non-full column** — Each turn is a single drop into a column that isn't full. |
| If both players fill the board with no four-in-a-row, the game is a: | **draw** — A full board with no line is a draw. |
| Blocking an open three uses the skill of: | **recognizing the opponent's pattern** — Spotting the three-in-a-row pattern lets you block it. |
| Playing the CENTER column early is strong because it: | **is part of the most possible four-in-a-rows** — The center is involved in more winning lines than any other column. |
| The center column is valuable enough that a common opening is to: | **play the center first** — Perfect play starts in the center. |
| If you can win THIS move or block the opponent's win, you should: | **take your own win** — If you have an immediate winning move, play it (the game ends). |
| Setting up TWO threats at once (a fork) is strong because the opponent: | **can only block one** — A fork gives two ways to win; one block leaves the other open (next kit). |

## 2. tactics

| ✂ Question (fold →) | Answer |
|---|---|
| In Connect-4, creating threats on two different rows/diagonals at once is a: | **fork** — Two threats in different lines = a fork. |
| If you see a chance to fork OR to block a minor threat, and the fork wins, choose: | **the fork** — A winning fork usually outweighs a small defensive move. |
| A fork is powerful because it changes "one threat" into: | **two simultaneous threats** — Doubling the threats overwhelms a single defensive move. |
| A "fork" (double-threat) is a move that creates: | **two winning threats at once** — A fork makes two threats; the opponent can block only one. |
| In Connect-4, a classic fork is three-in-a-row that is: | **open on both ends** — Open on both ends = two ways to make four = a fork. |
| Forks appear in many games because the idea is: | **one move, two threats — general across grid games** — The fork transfers to Gomoku, checkers, chess, and more. |
| Preventing forks is easier when you: | **avoid leaving two of your lines one-move-from-a-threat** — Don't give the opponent squares that create two threats. |
| The best forks threaten wins that are: | **both unstoppable individually next move** — Each threat should independently win if unblocked. |
| A fork is the tactical cousin of the deduction idea: | **forcing the opponent into a no-win choice** — Like a forcing clue, a fork removes all good replies. |
| The fork works in Connect-4, Gomoku, and beyond because all are: | **line-connection games where two threats can't both be blocked** — Connection games share the two-threats-one-block logic. |
| A fork that threatens to win on the next move on two lines is: | **usually decisive (winning)** — Two live winning threats normally decide the game. |
| Big idea: a fork gives the opponent: | **one move but two problems** — One reply can't solve two threats — the fork wins. |
| A fork demonstrates that in grid games, a good position can convert into: | **a forced win** — The right geometry (a fork) forces a win. |
| Twain's signature is: | **creating two threats at once (the fork)** — Twain embodies the double-threat fork. |
| Against a strong forker, the safest play is: | **watch for and block fork-setups early** — Early prevention beats late blocking. |
| Forks reward the player who: | **looks a move ahead to set two threats** — You must plan ahead to create a double threat. |
| To defend against a fork, you should: | **prevent the opponent from setting up two threats** — Stop the fork before it forms; after it forms it's usually too late. |
| If your opponent threatens to make a fork next move, you should: | **block the fork-setup square** — Deny the square that would create the double threat. |

| | |
|---|---|
| Twain would say the secret to forks is: | **planning two threats before you play the move** — Plan the double threat ahead of the move. |
| Setting up a fork often needs a: | **quiet preparing move first** — A preparing move creates the geometry for the fork. |
| A fork with two threats that the opponent CAN answer with one move is: | **not a real fork** — If one move stops both, it isn't a genuine fork. |
| The opponent facing a fork should: | **block the more dangerous threat and accept losing the other** — You can only save one; choose the better block (though a true fork still wins). |
| The fork is an example of the math practice: | **using structure/regularity (MP.8)** — Recognizing the forking structure is repeated-reasoning skill. |
| A fork wins because the opponent gets: | **only one move to stop two threats** — One block leaves the other threat to win next turn. |
| A "double threat" and a "fork" are: | **the same idea** — They are two names for the same tactic. |

## 3. defense

| ✂ Question (fold →) | Answer |
|---|---|
| A good defender blocks: | **the most dangerous threat first** — Address the biggest threat first. |
| If you cannot block both ends of an open three, the correct earlier lesson is to: | **have prevented the open three from forming** — Prevention beforehand is the real defense; once it's an open three you're often lost. |
| If both you and the opponent have a three-in-a-row and it's your move, and you can complete four: | **win by completing your four** — An immediate win takes priority over blocking. |
| Ignoring the opponent's discs usually leads to: | **getting surprised by a completed line** — Not reading threats gets you beaten by a line you didn't see. |
| When the opponent has three-in-a-row with one open end, you must: | **block the open end** — Blocking the open end stops the four. |
| A "threat" is a move away from: | **the opponent (or you) completing a winning line** — A threat is a line one move from winning. |
| An opponent with two separate open threes has essentially: | **a fork (you can't block both)** — Two open threes are two forks worth of threats — usually a loss. |
| If blocking one threat CREATES your own threat, that block is: | **especially good** — A defensive move that also attacks is efficient — best of both. |
| Blocking and threatening are two sides of: | **reading lines on the grid** — Both come from seeing the lines forming on the board. |
| A threat you make forces the opponent to: | **respond (block) instead of building freely** — Your threats seize the initiative by forcing responses. |
| The best time to block a threat is: | **before it becomes unstoppable (early)** — Early blocking prevents forks and open fours. |
| Rowan and Twain together in this kit show that you must: | **build your threats AND read the opponent's** — Good play balances your connection-building with reading their threats. |
| Making threats that force blocks lets you: | **control the flow of the game** — Forcing moves keep the opponent reacting to you. |
| The safest board for you is one where the opponent has: | **no open threats** — No live threats means no immediate danger. |
| Defensive skill improves by: | **practising spotting threats quickly** — Practice sharpens threat recognition. |
| You should track the opponent's threats by: | **scanning rows, columns, and diagonals for their discs** — Check all orientations for developing enemy lines. |
| Reading threats means: | **noticing the opponent's lines before they complete** — You must spot developing lines early to defend. |
| If you must choose which of two threats to | |

| | |
|---|---|
| block and only one can win for the opponent, block: | **the one that would actually win** — Block the genuine winning threat, not a fake one. |
| Blocking too late (after a fork forms) often: | **loses, because two threats remain** — Once a fork exists, one block can't save you. |
| Threats and blocks make grid games a battle of: | **reading and initiative, not just placing discs** — The real contest is seeing lines and forcing responses. |
| An open three (both ends open) is dangerous because: | **one block can't stop both ends** — Two open ends = two win squares; a single block fails. |
| Reading threats is the same core skill as in deduction: | **using clues (the discs) to predict what's coming** — The discs are clues that reveal the opponent's plan. |
| Big idea: to defend, you must: | **see the opponent's threats before they complete them** — Reading threats early is the heart of defense. |
| A move that blocks AND extends your own line demonstrates: | **efficient, dual-purpose play** — One move doing two jobs is efficient. |
| If the opponent can win next move, your first priority is to: | **block that win (unless you can win first)** — Stop an immediate loss unless you have your own immediate win. |

## 4. computer science

| ✂ Question (fold →) | Answer |
|---|---|
| A game with a SMALLER game tree is: | **easier to solve** — Fewer positions = easier to search completely. |
| The bigger the branching factor, the: | **larger and harder-to-search the tree** — More choices per move explode the tree size. |
| Big idea: "solved" means the perfect-play result is: | **known** — A solved game has a known perfect-play outcome. |
| To solve Connect-4, researchers explored the game's: | **game tree of possible move sequences** — Solving searches the tree of all possible plays. |
| Knowing Connect-4's solution, the best opening advice is: | **take the center** — The center is the solution's key first move. |
| Solving a game is different from just playing it well because solving: | **proves the result for all lines, not just one game** — A solution is a proof over the whole tree, not a single strong game. |
| Tic-tac-toe, with perfect play, is a: | **draw** — Perfect tic-tac-toe is a draw — a tiny solved game. |
| Solving a game connects to the CS idea of: | **searching a space of possibilities** — It's a search-and-analysis problem. |
| Solving harder games is limited mainly by: | **computing power vs the tree size** — The tree size can outstrip available computing power. |
| A solved game is proof that games can be: | **analyzed mathematically** — Solving shows games are formal, analyzable objects. |
| A game is "solved" when we know the result assuming: | **perfect play by both sides** — A solved game's outcome is known under optimal play. |
| "Perfect play" means each move is: | **optimal against the opponent's best replies** — Perfect play is optimal versus optimal opposition. |
| Comparing Connect-4 (first-player win) and checkers (draw) shows that solving a game tells you: | **the exact result, which differs by game** — Different games have different solved results. |
| The lesson from solved games is that a game's outcome can be: | **determined in advance under perfect play** — Perfect-play outcomes are fixed facts about the game. |
| Endgame positions (few pieces) are usually solved: | **first, because they have smaller trees** — Fewer pieces mean fewer positions, so endgames are solved before openings. |
| A common way to solve games is to work: | **backward from known end positions** — Retrograde analysis from solved endings is a key method. |
| The winning first move in perfect Connect-4 is to play: | **the center column** — The center is the key first move for the first-player win. |
| A perfect Connect-4 computer going first will: | **win with correct play** — Since it's a solved first-player win, perfect play wins. |

| | |
|---|---|
| Storing exact results for many positions creates a: | **database used to play perfectly** — Position databases let a program play perfectly from stored results. |
| Connect-4, with perfect play, is a: | **first-player win** — Connect-4 is solved as a first-player win (start in the center). |
| If a game is solved as a DRAW, perfect play by both sides yields: | **a draw** — A solved-draw game draws under perfect play. |
| Chess and Go are NOT solved because: | **their game trees are far too large** — Their combinatorial explosion makes solving infeasible so far. |
| Even though Connect-4 is solved, people still play it because: | **playing well is still fun and challenging for humans** — Solved games remain enjoyable human challenges. |
| Connect-4 being a first-player win shows that a game can be: | **unfair with perfect play (first mover advantage)** — Perfect play favors the first player in Connect-4. |
| Gridley's role is to: | **explain how games get solved and how the ideas connect** — Gridley is the family host + CS/transfer mentor. |

## 5. connection

| ✂ Question (fold →) | Answer |
|---|---|
| A stone in the center of the board is strong because it: | **can be part of lines in all directions** — Central stones participate in the most potential lines. |
| A stone placed to extend a line of three toward five creates a: | **growing threat** — Extending toward five builds a threat, like Connect-4 threats. |
| The huge branching factor makes Gomoku: | **harder to fully search than Connect-4** — Free placement explodes the tree, making full search harder. |
| Gomoku is played by placing stones on: | **the intersections of a grid** — Stones go on grid intersections and stay put (no gravity). |
| Unlike Connect-4, Gomoku stones: | **can be placed anywhere empty (no gravity)** — You freely choose any empty intersection. |
| Gomoku rewards players who: | **look ahead to create multiple threats** — Planning multi-threat sequences wins. |
| In Gomoku, you win by making a line of: | **five in a row** — Gomoku (five-in-a-row) needs an unbroken line of five. |
| A key Gomoku skill is spotting when your line has: | **two open ends (a strong threat)** — Open ends make a line a serious threat. |
| A winning Gomoku line can be: | **horizontal, vertical, or diagonal** — Any straight line of five wins. |
| Gomoku is sometimes called: | **Five-in-a-Row** — Gomoku literally means five-in-a-row. |
| The main difference between four-in-a-row and five-in-a-row games is: | **the length of the winning line and the free board** — Gomoku needs five and places freely; Connect-4 needs four and drops. |
| An "open four" (four in a row with both ends open) in Gomoku is: | **an unstoppable threat (wins next move)** — Both ends open means two ways to make five — unstoppable (next kit). |
| The free board makes Gomoku have: | **more possible moves than Connect-4** — Placing anywhere gives a huge branching factor. |
| A stone placed at an empty intersection in Gomoku stays there: | **for the rest of the game (no gravity, no flipping)** — Gomoku stones are permanent once placed — unlike Connect-4 drops or Reversi flips. |
| Because you place freely, forks (double threats) in Gomoku are: | **very common and powerful** — Free placement makes it easy to set two threats — forks abound. |
| Five stones in a row with a gap (X X _ X X) is: | **not yet a win (must be unbroken)** — The five must be consecutive with no gap. |
| The reason competitive Gomoku isn't just "first player always wins" is: | **the added opening rules restore balance** — Opening restrictions neutralize the first-player advantage. |
| The connection primitive transfers from Connect-4 to Gomoku because both ask you to: | **build a straight line before your opponent** — Both are races to complete a straight connected line. |
| Gomoku and Connect-4 share the primitive | |

| | |
|---|---|
| of: | **connecting a line** — Both are connection games — build a line. |
| Because there is no gravity, defending in Gomoku means watching: | **lines in every direction across the whole board** — Threats can form anywhere in any direction. |
| Rowan's connection primitive in Gomoku is: | **extending a line toward five** — Rowan builds the connecting line of five. |
| To defend Gomoku well you must: | **scan all directions for the opponent's growing lines** — Threats can grow in any of the four directions anywhere. |
| Big idea: Gomoku is Connect-4's bigger cousin, needing: | **five in a row on a free (no-gravity) board** — Same connection idea, longer line, free placement. |
| Because free Gomoku favors the first player, competitive versions: | **add opening restrictions to balance it** — Rules like swap/opening restrictions balance the first-move advantage. |
| Free-style Gomoku (no restrictions), with perfect play, is a: | **first-player win** — Unrestricted Gomoku is a solved first-player win — which is why tournaments add rules. |

## 6. tactics

| ✂ Question (fold →) | Answer |
|---|---|
| Big idea: control Gomoku by: | **making forcing open-threes that build to an unstoppable open four or double three** — Forcing sequences that end in a double threat are how you win. |
| Twain's forcing sequences use open threes to: | **force the opponent to keep blocking** — Each open three forces a block, letting Twain keep the initiative. |
| Chaining forcing moves (each an open three) lets you: | **steer the game toward a winning fork** — A sequence of forced blocks can be steered into a final fork. |
| Recognizing open threes/fours is an example of: | **reading structure/patterns (MP.8)** — Spotting these patterns is repeated-reasoning structure. |
| To WIN with forcing sequences you usually aim to end on: | **a double threat (open four or double three)** — The finish is an unstoppable double threat. |
| A "closed four" (four in a row with only ONE open end) is: | **still a threat, but blockable at its single open end** — One open end means one win square — the opponent can block it (unlike an open four). |
| Open threes are the building blocks of: | **forcing sequences and forks** — Chained open threes build toward a winning fork. |
| Open threes and fours are the Gomoku version of: | **threats and forks (like Connect-4)** — They are the same threat/fork ideas specialized to five-in-a-row. |
| If the opponent ignores your open three, you can: | **make an open four and win** — An unblocked open three becomes an open four — a win. |
| A "double three" (two open threes at once) is powerful because it: | **creates two threats the opponent can't both stop** — Two open threes = a fork; the opponent can't block both. |
| To defend against a double three, you needed to: | **prevent the second open three from forming** — Once both open threes exist, you can't block both — prevent earlier. |
| A "four" with both ends open is stronger than a "three" with both ends open because it: | **wins immediately, not just threatens** — An open four wins next move; an open three only threatens to make one. |
| If you make an open three, the opponent should: | **block one end before it becomes an open four** — Blocking an end stops it growing into the unstoppable open four. |
| A "forcing move" is one that: | **makes a threat the opponent must answer** — Forcing moves compel a response, letting you dictate play. |
| The best forcing sequences are ones the opponent: | **must answer every time, ending in a fork** — Every move forces a block, and the last creates a double threat. |
| Forcing sequences show that in Gomoku, tempo (initiative) can be worth: | **more than raw stone count** — Keeping the initiative with forcing moves often decides the game. |
| An "open four" (four in a row, both ends open) is: | **unstoppable — it wins next move** — Both ends open gives two ways to make five; one block can't stop both. |

| | |
|---|---|
| An open four cannot be stopped because blocking one end: | **leaves the other end to complete five** — One block still leaves a second winning square. |
| The difference between an open three and a closed three (one end blocked) is: | **the open three still threatens; the closed one is much weaker** — A blocked end removes one growth direction, weakening the three. |
| Making a threat that ISN'T forcing (opponent can ignore it) is: | **less useful — it doesn't control the game** — Non-forcing moves let the opponent do what they want. |
| An "open three" in Gomoku is three in a row with: | **both ends open (empty)** — An open three has empty squares on both ends, threatening to become an open four. |
| An open four beats a simple four (one end blocked) because: | **the simple four can be blocked at its one open end** — A four with one open end has a single win square that can be blocked; an open four has two. |
| A defender facing a long forcing sequence should look for a block that: | **also creates their own threat (counter-force)** — Blocking with a counter-threat can wrest back the initiative. |
| The pattern _ X X X _ (open three) threatens to become: | **_ X X X X _ or X X X X X (an open/strong four)** — Extending an open three makes a four, ideally an open four. |
| The initiative in Gomoku belongs to the player who: | **keeps making forcing threats** — Whoever forces the responses controls the game. |

# 7. connection

| ✂ Question (fold →) | Answer |
|---|---|
| A Hex chain wins if it touches: | **both of your assigned edges continuously** — The chain must connect your two edges without a break. |
| The proof that Hex has no draws is an example of: | **mathematical reasoning about a game** — It's a topological/mathematical result about the game. |
| Why can't Hex be a draw? | **the two players' connections would have to cross, so exactly one succeeds** — Topology guarantees exactly one player connects their edges. |
| Hex was invented independently by: | **two mathematicians (Piet Hein and John Nash)** — Hex was created by Piet Hein and, separately, John Nash. |
| Hex demonstrates that a game can be provably: | **a first-player win even without knowing the strategy** — The existence of a first-player win is proven non-constructively. |
| A Hex position where your chain reaches both edges means: | **you have won** — Connecting both your edges is the win. |
| Hex is a pure __ game (no capturing, no luck): | **connection** — Hex is a pure connection game of strategy. |
| Because blocking and connecting are the same in Hex, a single move can: | **advance your path AND block the opponent** — Every path-building move also hinders the opponent's crossing path. |
| In Hex, you win by building an unbroken chain of your pieces connecting: | **your two opposite edges of the board** — Each player owns two opposite edges and must connect them with a chain. |
| Blocking your opponent in Hex is the same as: | **trying to connect your own edges (they cross)** — Because the two connections must cross, blocking them IS building your own path. |
| With perfect play, the FIRST player in Hex can: | **force a win** — A strategy-stealing argument proves the first player wins with perfect play. |
| Each player in Hex owns: | **two opposite edges of the board** — You must connect your own two opposite edges; the opponent owns the other pair. |
| Hex is played on a board of: | **hexagons (six-sided cells)** — Hex uses a rhombus-shaped grid of hexagonal cells. |
| The core skill in Hex is: | **seeing paths and connections across the board** — Visualizing connecting paths is the central skill. |
| To reduce the first-player advantage, Hex often uses the: | **swap rule (pie rule)** — The swap rule lets the second player take the first move, balancing the game. |
| The bigger the Hex board, the: | **larger the game tree (harder to solve)** — Larger boards explode the number of positions. |
| A famous fact about Hex is that it: | **can never end in a draw** — In Hex, someone always connects their edges — a draw is impossible. |
| Rowan's connection primitive in Hex is: | **building a path from one edge to the other** — Rowan builds the connecting path — Hex's core. |

| | |
|---|---|
| The connection primitive links Hex to Connect-4 and Gomoku because all three ask you to: | **build a connected structure across the board** — All three are connection games — build a connected line/path. |
| In Hex, playing in the CENTER early is strong because it: | **helps connect toward both edges** — Central play keeps your path options open toward both edges. |
| The strategy-stealing argument shows the first player wins but: | **does NOT give the actual winning moves** — It proves a first-player win exists without revealing the strategy. |
| Hex differs from Connect-4/Gomoku mainly in that its winning shape is: | **an edge-to-edge path, not a fixed-length line** — Hex wins by any connecting path, not a set-length straight line. |
| Because you can't draw, defense in Hex is really: | **offense (connect your own edges to stop theirs)** — You "defend" by building your own crossing connection. |
| Because a draw is impossible, every finished Hex game has: | **exactly one winner** — One player always achieves the crossing connection. |
| Big idea: Hex is a pure connection game where: | **you build a path between your edges, and a draw is impossible** — Connect your two edges; someone always connects, so no draws. |

# 8. tactics

| ✂ Question (fold →) | Answer |
|---|---|
| A bridge is safe because if the opponent plays one of its two cells, you: | **play the other to keep the connection** — The two empty cells give a guaranteed reconnection. |
| A "bridge" in Hex is: | **a safe two-cell connection that can't be cut in one move** — A bridge links two of your stones with two empty cells so the opponent can't cut both. |
| The bridge pattern is popular because it balances: | **speed of connection with safety** — Bridges are both fast and hard to cut. |
| Hex strategy shows that in connection games, SAFE connections matter because: | **a fragile link can be cut and cost the game** — Fragile links are liabilities; safe patterns win. |
| The reason a bridge is better than placing the two stones adjacent is that a bridge: | **covers more ground while staying safe from a single cut** — A bridge reaches farther per move yet keeps a guaranteed reconnection. |
| Blocking the opponent in Hex means: | **building your own path across their intended route** — You block by connecting through where they need to go. |
| Good Hex players think about the WHOLE board because: | **a connection can be routed many ways** — Flexible routing means whole-board awareness pays off. |
| The two big Hex skills are: | **seeing safe connections (bridges) and reading ladders** — Bridges + ladders are core Hex technique. |
| Because there are no draws, the whole game is a fight over: | **who completes their crossing connection first** — Both race to connect; exactly one succeeds. |
| Bridges reduce risk because a single opponent move: | **cannot break the two-cell connection** — One move can't cut both empty cells of a bridge. |
| Since connecting and blocking are the same, a defensive Hex player is really: | **still trying to connect their own edges** — There's no pure defense — you connect to block. |
| An "escape" or "ladder escape" is a stone placed to: | **help a ladder connect instead of failing** — A pre-placed stone lets a ladder reach its goal. |
| Blocking with a bridge is powerful because it: | **safely stops the opponent while advancing you** — A bridge both blocks and builds — efficient. |
| Efficient Hex play means connecting with: | **safe patterns (bridges) rather than fragile single links** — Bridges beat fragile one-cell links. |
| The center of the Hex board is valuable because it: | **offers the most flexible connections toward both edges** — Central stones keep the most path options open. |
| Using bridges lets you connect stones: | **with fewer moves and less risk of being cut** — Bridges connect efficiently and safely. |
| Recognizing bridge patterns is an example of: | **using structure (MP.8)** — Bridges are a structural pattern you learn to see. |

| | |
|---|---|
| If the opponent threatens to cut your connection, a good reply is to: | **reconnect via a bridge or alternate path** — Reroute through a safe pattern to keep connected. |
| Rowan and Twain in this kit combine: | **building connections (paths) with blocking the opponent** — Hex strategy blends connection and blocking (they're the same move). |
| Planning several moves ahead in Hex helps you: | **set up bridges and ladder escapes in advance** — Look-ahead lets you prepare safe connections and escapes. |
| A "ladder" in Hex is a forced sequence where: | **players repeatedly respond along a line** — A ladder is a chain of forced adjacent responses. |
| A strong Hex move often does two things: | **extends your connection and blocks the opponent's** — The best moves connect and block at once. |
| Big idea: Hex strategy is about making: | **safe, efficient connections that also block the opponent** — Use bridges and ladders to connect safely while blocking. |
| A ladder that has no escape usually: | **fails to connect (the defender wins that fight)** — Without an escape, a ladder can be blocked at the edge. |
| The idea of a "safe two-way connection" (bridge) is like the earlier idea of: | **an open-both-ends threat that can't be stopped by one move** — Both rely on having two options so one enemy move can't stop both. |

## 9. territory

| ✂ Question (fold →) | Answer |
|---|---|
| Flank's signature is: | **flanking a line of enemy discs and flipping them** — Flank embodies the outflank-and-flip primitive. |
| The end-of-game disc count is what matters, so you should: | **aim for a strong FINAL position, not a big middle lead** — Play for the final count, not a fleeting mid-game lead. |
| The winner of Reversi is the player with: | **the most discs of their colour at the end** — Most discs of your colour wins. |
| Big idea: Reversi is a game of: | **flipping ownership by outflanking, won by the final disc count** — Outflank to flip; whoever has the most discs at the end wins. |
| A move that flips discs in three directions at once is: | **often strong (big positional effect)** — Multi-directional flips can be powerful if they gain good squares. |
| Reversi teaches that "having more now" can be: | **temporary — ownership can reverse** — Because discs flip, a lead can vanish — plan for the end. |
| The starting position has: | **four discs (two each) in the center** — Othello begins with a 2×2 center of alternating discs. |
| In Reversi/Othello, you place a disc so that it: | **outflanks (traps) a line of enemy discs between it and another of your discs** — A legal move must flank at least one line of opponent discs. |
| A legal Reversi move MUST: | **flip at least one enemy disc** — If a placement flips nothing, it is illegal. |
| Having the MOST discs in the MIDDLE of the game is: | **not necessarily good (they can be flipped back)** — Mid-game disc counts swing wildly; leading early can be a trap. |
| A good Reversi habit is to value: | **stable discs (that can't be flipped back)** — Stable discs (like corners) are permanent gains. |
| Flanking can happen along: | **any direction — horizontal, vertical, or diagonal** — Outflanking works in all eight directions. |
| When you outflank enemy discs, those discs: | **flip to your colour** — All outflanked enemy discs flip to your colour. |
| If you cannot flip any disc, you must: | **pass (skip your turn)** — With no legal move, you pass. |
| A single Reversi move can flip discs in: | **several directions at once** — One placement can outflank multiple lines simultaneously. |
| Reversi is also known as: | **Othello** — Othello is the trademarked name for Reversi. |
| If both players pass in a row (no legal moves), the game: | **ends and discs are counted** — Two consecutive passes end the game; count the discs. |
| The outflank-and-flip primitive is DIFFERENT from connection because it: | **changes ownership of pieces instead of building a line** — Reversi is a territory/flip game, not a connection game. |
| The game ends when: | **neither player can make a legal (flipping) move** — Play stops when no legal moves remain (usually a full board). |

| | |
|---|---|
| Reversi's primitive (flip ownership) contrasts with connection games where pieces: | **stay your colour once placed** — In connection games your pieces don't change owner; in Reversi they can. |
| The two players in Reversi/Othello use discs that are: | **one colour on each side (flipped to show the owner)** — Each disc is light on one face and dark on the other; flipping shows the new owner. |
| Flipping many discs early can BACKFIRE because it: | **gives the opponent more edges/moves to flip them back** — Grabbing discs early can expose them and give the opponent good replies. |
| The core question each move in Reversi is: | **which placement flips the RIGHT discs for a lasting advantage** — Quality of flips (position) beats sheer quantity. |
| Reversi rewards thinking about: | **which flips help you LATER, not just now** — Positional value matters more than the current disc count. |
| A disc on a CORNER is special because it: | **can never be flipped (see next kit)** — Corners can't be outflanked, making them permanent. |

## 10. territory

| ✂ Question (fold →) | Answer |
|---|---|
| A disc in a CORNER of the Othello board: | **can never be flipped** — Corners have no square beyond them, so they can't be outflanked — they're permanent. |
| Why are corners so valuable in Othello? | **they are stable AND help stabilize nearby edge discs** — A corner can't be flipped and lets you build stable edges from it. |
| The "frontier" discs (next to empty squares) tend to be: | **more flippable (less stable)** — Frontier discs border empty squares and can be outflanked. |
| Edges are more stable than the center because edge discs: | **can be flipped from fewer directions** — Fewer flanking directions makes edge discs harder to flip. |
| Corners are permanent because outflanking needs a disc on BOTH sides, and a corner: | **has no outside square to be flanked from** — There's no square beyond a corner, so it can't be trapped. |
| Playing an X-square (diagonally next to an empty corner) is usually: | **a mistake (it can give up the corner)** — X-squares often hand the opponent the corner. |
| Flank (in this kit) shows that the strongest squares to take are: | **the corners (and stable edges)** — Corners anchor stability — the key territory idea. |
| Stability is the Othello version of the general idea: | **gains that can't be undone are the most valuable** — Permanent (stable) gains matter most. |
| Reading stability trains you to value: | **the permanence of a gain, not its size right now** — Permanence over immediate size is the Othello lesson. |
| The center discs at the start are: | **unstable (easily flipped later)** — Early center discs flip readily as the game develops. |
| Once you own a corner, you can often build: | **a chain of stable discs along the edges** — Corners anchor stable edge chains. |
| If you must choose between flipping 6 discs (no corner) or taking a corner flipping 1: | **usually take the corner** — A corner's stability usually outweighs a bigger temporary flip. |
| There are how many corners on a standard Othello board? | **4** — A rectangular board has four corners — the four most valuable, stable squares. |
| Stable discs matter because at the END of the game they: | **are guaranteed to count for you** — Stable discs are locked-in points at game's end. |
| Big idea: in Othello, value: | **stable discs (especially corners) over the temporary disc count** — Corners and stable discs decide the game, not the swinging mid-game count. |
| Why do experts sometimes have FEWER discs mid-game yet win? | **they hold stable squares that will decide the final count** — Stability wins the end even from a mid-game deficit. |
| A disc that can never be flipped for the rest of the game is called: | **stable** — Stable discs are locked in your colour. |

| | |
|---|---|
| A player who owns all four corners usually: | **has a strong, stable position** — Four corners anchor a lot of stable territory. |
| A disc is UNSTABLE if: | **the opponent could still flip it later** — Unstable discs remain vulnerable to future flips. |
| The squares NEXT to a corner (X-squares/C-squares) are risky because playing there can: | **give the opponent access to the corner** — Playing adjacent to an empty corner often lets the opponent take the corner. |
| Corners connect to the deduction habit of: | **securing what can't be taken away before chasing more** — Lock down permanent gains first — like securing certain deductions. |
| Recognizing stable vs flippable discs is an example of: | **using board structure (MP.7)** — Stability is a structural feature you learn to read. |
| Good Othello players judge a position by: | **stable discs, not the current raw count** — Stability predicts the final result better than the mid-game count. |
| The safest place of all is the: | **corner** — Corners are the most stable squares. |
| A common Othello mistake is to: | **grab many discs but hand the opponent a corner** — Chasing disc count while giving up a corner is a classic error. |

# 11. territory

| ✂ Question (fold →) | Answer |
|---|---|
| Good mid-game play often means: | **keeping your discs compact and your options open** — Compact, flexible positions preserve mobility. |
| Having MORE mobility than your opponent is good because it: | **gives you better choices and can force them into bad moves** — More options = better play and can force the opponent into weak moves. |
| The lesson of mobility is that in games, having good OPTIONS can be worth more than: | **immediate material (disc count)** — Flexibility (options) often beats immediate material. |
| Flank (mid-game) shows that owning FEWER discs mid-game can be good if it: | **keeps your mobility high and the opponent's low** — A small, flexible position can starve the opponent of good moves. |
| The disc count in the MIDDLE of an Othello game is: | **a poor predictor of the winner** — Mid-game counts swing; mobility and stability predict better. |
| Mobility connects to the endgame idea of: | **forcing the opponent to move (zugzwang-like)** — Restricting opponent mobility is like forcing them into a bad move. |
| The safest way to convert a mobility edge is to: | **use it to secure corners and stable discs** — Turn flexibility into permanent (stable) gains. |
| "Territory" thinking in Othello means valuing: | **control of good squares over sheer disc count** — Controlling key squares (corners, stable edges) beats raw count. |
| A counterintuitive Othello idea is that having FEWER discs mid-game can: | **increase your mobility (evaporation strategy)** — Keeping few, well-placed discs can maximize your future moves. |
| Big idea: in Othello, manage: | **mobility (your options) and territory (good squares), not just disc count** — Options + key squares win Othello, not the swinging mid-game count. |
| "Mobility" in Othello means: | **how many legal moves you have** — Mobility is the count of legal (flipping) moves available to you. |
| Mobility management is the Othello cousin of: | **tempo/parity in checkers (controlling who moves)** — Both control who is forced to move at bad moments. |
| Reasoning about mobility over many moves is an example of: | **using regularity/structure (MP.8)** — Tracking move-count patterns is repeated-reasoning structure. |
| If you have many legal moves and the opponent has few, you have: | **a mobility advantage** — More legal moves than the opponent is a mobility advantage. |
| A position with only one legal move gives you: | **no choice (forced)** — Single-move positions remove your agency — low mobility. |
| Low mobility is dangerous because it can force you to: | **play an X-square and hand over a corner** — Out of good moves, you may be forced into a corner-losing move. |
| Sacrificing disc count for mobility is worth it when it: | **leads to taking corners and winning the end** — Trading count for mobility pays off if it wins the corners/endgame. |

| | |
|---|---|
| Frontier discs (next to empty squares) affect mobility because they: | **create squares where the opponent can play** — Many frontier discs tend to give the opponent more move options. |
| Mobility and stability together tell you a position is strong when you have: | **high mobility now and stable discs for the end** — The best positions combine present flexibility with future permanence. |
| A player with high mobility can usually: | **wait to take corners on good terms** — Flexibility lets you grab corners when it's most favorable. |
| Forcing the opponent to reduce YOUR flips while keeping THEIR options low is: | **good mobility management** — Managing both sides' mobility is a core Othello skill. |
| A player who ignores mobility and only grabs discs often: | **gets forced into losing a corner** — Greedy disc-grabbing typically backfires via forced corner losses. |
| If the opponent runs OUT of good moves, they may be forced to: | **give up a corner or make a weak move** — Low mobility forces bad moves, often surrendering a corner. |
| An "even" (parity) idea appears in Othello endgames as: | **who is forced to make the last move in a region** — Endgame parity — who moves last in each empty region — matters, echoing Dots-&-Boxes. |
| Experts watch mobility because it predicts: | **who will be forced into bad corner moves later** — Mobility trends forecast future forced errors. |

# 12. territory

| ✂ Question (fold →) | Answer |
|---|---|
| Dots and Boxes is really a battle over: | **who is forced to open chains (parity)** — The core fight is parity — who must open first. |
| Big idea: Dots and Boxes is won by: | **controlling who is forced to open the chains (parity)** — The parity battle over who opens first decides the game. |
| Drawing a line that leaves a box with two sides is: | **usually safe** — A 2-sided box is not yet a gift; drawing to keep boxes at ≤2 sides is safe. |
| When you complete a box, you: | **claim it AND take another turn** — Completing a box earns the box and an extra move. |
| A player who never thinks about parity will often: | **be forced to open the first long chain and lose** — Ignoring parity means getting stuck opening first. |
| When only unsafe moves remain, the position is decided by: | **who was forced to move there (parity)** — Whoever runs out of safe moves first must open — parity decides. |
| A "chain" is: | **a line of boxes that can be taken one after another** — A chain is a run of connected boxes claimable in sequence. |
| On your turn in Dots and Boxes you draw: | **one line (edge) between two adjacent dots** — Each turn is a single horizontal or vertical line between neighbouring dots. |
| The goal in the OPENING/middle of Dots and Boxes is to: | **make safe moves and set up favorable chain parity** — Play safe and steer the chain count/parity in your favor. |
| You should AVOID drawing the THIRD side of a box because it: | **gives the opponent an easy box (the 4th side)** — Drawing the 3rd side lets the opponent complete the box for free. |
| Counting sides before each move helps you avoid: | **accidentally making a 3-sided box (a gift)** — Check side-counts to steer clear of gifting boxes. |
| Oddwin's signature is: | **counting parity and chains (who moves last)** — Oddwin embodies parity/chain reasoning in Dots and Boxes. |
| Once you open a chain for the opponent, they can usually: | **take the whole chain in one turn (each box gives another move)** — Each completed box grants another move, so a chain falls all at once. |
| The primitive in Dots and Boxes is: | **parity / who-moves-last (with a bit of territory)** — It's a parity + territory game, not a connection or flip game. |
| The number of long chains and their parity often decides: | **the winner of a Dots-and-Boxes game** — Chain parity is the key strategic factor. |
| The winner is the player who claims: | **the most boxes** — Most boxes wins. |
| A box needs how many sides to be claimed? | **4** — All four sides complete (claim) a box. |
| If you must give away boxes, a smart idea (next kit) is to: | **sacrifice a small chain to keep control (the double-cross)** — The double-cross sacrifices to force the opponent to open next. |

| | |
|---|---|
| The player forced to open the FIRST long chain is usually: | **at a disadvantage** — Being forced to open first often means losing the chain battle. |
| Dots and Boxes connects to checkers' endgame idea of: | **tempo/parity (who is forced to move)** — Both hinge on forcing the opponent to move/open at the wrong time. |
| A "safe" move is one that draws a side WITHOUT: | **making any box have three sides** — Safe moves don't create a 3-sided box for the opponent. |
| Dots and Boxes rewards players who: | **count sides and plan chain parity** — Careful counting and parity planning win. |
| In Dots and Boxes, you complete a box by drawing: | **its fourth side** — Drawing the 4th side of a box claims it. |
| When all safe moves run out, a player is FORCED to: | **open a box/chain for the opponent** — Eventually someone must give the opponent boxes — who, is decided by parity. |
| The extra-turn-on-completion rule is what makes: | **chains fall all at once** — Because each box grants a move, a whole chain can be taken in one turn. |

## 13. parity

| ✂ Question (fold →) | Answer |
|---|---|
| The player controlling parity uses the endgame technique called the: | **double-cross (next kit)** — The double-cross is how the parity-winner keeps control. |
| Parity is decided by the interaction of: | **the number of chains and whose turn it is** — Chain count + turn order determine who must open. |
| The number of long chains having the "wrong" parity for you usually means: | **you'll be forced to open first** — Wrong parity → you open first → you likely lose the chains. |
| A chain of exactly two boxes is a "short chain"; short chains matter less because they: | **don't support the controlling double-cross** — Only long chains (3+) sustain control via the double-cross; short chains don't. |
| The chain rule connects Dots and Boxes to checkers because both use: | **parity/tempo to force the opponent to move badly** — Both force the opponent into the losing move via parity. |
| Big idea: parity and the chain rule are about: | **controlling who is FORCED to open the chains** — Win the parity battle and the opponent must open first. |
| Counting parity means counting the number of moves/chains and checking if it is: | **odd or even relative to whose turn it is** — You compare odd/even against whose turn it will be. |
| Mastering parity turns Dots and Boxes from a kids' game into a: | **deep strategic contest** — Parity/chain theory gives the game real strategic depth. |
| A player who wins the parity battle will: | **force the opponent to open first** — Winning parity means the opponent must open. |
| Long chains are the key to parity because short chains (1-2 boxes): | **don't allow the controlling double-cross** — The double-cross needs a long chain; short ones can't sustain control. |
| Whoever is NOT forced to open the first long chain can often: | **use the double-cross to keep control** — The controlling player uses the double-cross (next kit) to stay in control. |
| If you realize you'll be forced to open, your best hope is: | **to open the SHORTEST chain (give away the least)** — When forced, minimize the loss by opening the smallest chain. |
| The chain rule (a common version) says: the first player wants the number of long chains to be: | **matched to a parity depending on the board (e.g., related to the number of dots)** — The exact target depends on board size and who is first; the point is to control chain parity. |
| "Parity" in Dots and Boxes is about whether a count is: | **odd or even** — Parity tracks odd/even to predict who is forced to open a chain. |
| Parity reasoning is an example of: | **using odd/even structure in repeated reasoning (MP.8)** — Counting odd/even to force an outcome is MP.8. |
| If getting the right parity requires one more (or fewer) chain, you can: | **join or split chains with your safe moves** — Safe moves can merge or split chains to fix parity. |
| If you count the chains early, you can: | **steer safe moves to reach a winning parity** — Early counting lets you plan safe moves toward good parity. |

| | |
|---|---|
| Parity thinking transfers to other games as: | **counting to control who is forced to act** — Counting-to-force-the-opponent is a general strategic tool. |
| The player who is forced to OPEN the first long chain usually: | **loses the chain battle** — Opening first hands the opponent the chain and control. |
| The "chain rule" is a guideline about: | **trying to make the number of long chains match a target parity** — The chain rule aims for the parity of long chains that favors you. |
| A "long chain" is a chain of: | **three or more boxes** — Long chains (3+) are the ones that decide control via the double-cross. |
| Good Dots-and-Boxes players start counting chains: | **well before the endgame** — Early counting is needed to steer parity in time. |
| The chain rule is a heuristic (guideline), meaning it: | **usually helps but has exceptions** — It's a strong guideline, not an absolute rule. |
| Oddwin's whole method is: | **counting chains + parity to force the opponent to open first** — Oddwin embodies chain/parity counting. |
| The reason parity matters is the extra-turn rule, which makes: | **whole chains fall to whoever the opener gives them to** — Because chains fall all at once, who opens them (parity) is decisive. |

## 14. tactics

| ✂ Question (fold →) | Answer |
|---|---|
| Oddwin's master move is: | **sacrificing two boxes to keep control of the chains** — Oddwin uses the double-cross to stay in control. |
| On the VERY LAST chain, you should: | **take ALL of it (no need to sacrifice)** — There's no next chain to control, so grab every box on the last one. |
| Leaving the last two boxes forces the opponent to: | **take those two and then OPEN the next chain** — The opponent must take the two gift boxes, then open the next chain for you. |
| When you double-cross, the opponent takes the two boxes you left and then MUST: | **open the next chain for you** — Taking the two gift boxes leaves them no safe move but to open the next chain. |
| To set up the double-cross, you first need to have won: | **the parity battle in the midgame** — Midgame parity play sets up the endgame double-cross. |
| The double-cross keeps the opponent in a loop of: | **opening chains for you to control** — Each turn they must open; you double-cross; repeat. |
| The double-cross is the payoff of winning the: | **parity battle (being NOT forced to open first)** — You must be the controller (opponent opens) to use the double-cross. |
| The double-cross is the CLIMAX technique built on: | **parity and the chain rule** — It's the payoff of the parity/chain-rule groundwork. |
| The two sacrificed boxes are called: | **the all-but-two (hard-hearted handback)** — Leaving two boxes for the opponent is the all-but-two / handback move. |
| The double-cross turns Dots and Boxes into a game of: | **endgame control, not greedy grabbing** — Skilled play is about control, not immediate box-grabbing. |
| For a double-cross to net a gain, the chains you control must be: | **big enough that two sacrificed boxes are worth the control** — The controlled chains must outweigh the two-box sacrifice. |
| The double-cross shows that in Dots and Boxes, taking FEWER boxes now can: | **win more boxes overall** — A small sacrifice for control usually nets more boxes. |
| Big idea: the double-cross means: | **leave the last two boxes of a chain to keep the opponent opening chains for you** — Sacrifice two boxes per long chain to retain control and win overall. |
| If you GREEDILY take every box in a chain, you must then: | **open the next chain yourself (losing control)** — Taking all boxes ends your bonus moves, forcing YOU to open next. |
| The "double-cross" (all-but-two) is when, given a long chain, you take all boxes EXCEPT: | **the last two, leaving them for the opponent** — You decline the last two boxes, forcing the opponent to open the next chain. |
| The double-cross only works on chains of length: | **3 or more (long chains)** — You need at least a long chain to leave two and still profit. |
| If you are FORCED to open first (lost | **cannot help you keep control (the opponent controls)** — The double- |

| | |
|---|---|
| parity), the double-cross: | cross is a controller's tool; the forced-opener can't use it to seize control. |
| The double-cross is like the checkers "shot" because both: | **sacrifice a little to gain more** — Both are sacrifices that force a bigger gain. |
| The double-cross is an example of: | **sacrificing a little to control parity (MP.8)** — It's a parity-control sacrifice — repeated-reasoning structure. |
| After you double-cross one chain, the opponent opens the next, and you can: | **double-cross again, keeping control chain after chain** — Repeated double-crosses keep you in control through all the long chains. |
| The double-cross demonstrates that CONTROL can be worth more than: | **immediate boxes** — Control of the chains outweighs a couple of immediate boxes. |
| A player who doesn't know the double-cross will often: | **grab everything and then lose control** — Greedy grabbing loses control and usually the game. |
| The double-cross trades a small loss (two boxes) for: | **control of who opens the next chain (usually a bigger win)** — Giving two boxes buys control, usually netting more boxes overall. |
| Each double-cross costs you 2 boxes but wins you the NEXT chain, which is usually: | **longer than 2 (a net gain)** — Trading 2 boxes for a longer chain nets positive. |
| A player using double-crosses across several chains will typically: | **win the majority of boxes** — Sustained control wins most of the boxes. |

## 15. transfer

| ✂ Question (fold →) | Answer |
|---|---|
| The "fork" (two threats at once) appears in: | **Connect-4 AND Gomoku (both connection games)** — Forks are a connection-game tactic seen across the family. |
| Recognizing that games share primitives is an example of: | **transfer — applying an idea from one context to another** — Transfer is using a learned idea in a new setting. |
| A game where you win by "trapping and flipping" the opponent's pieces most likely uses the primitive of: | **flipping/territory (like Reversi)** — Trap-and-flip is the Reversi territory primitive. |
| Spotting the shared primitive lets you: | **learn a new game faster by reusing what you know** — Transfer speeds learning: recognize the familiar core. |
| Recognizing primitives helps in subjects BEYOND games because it teaches: | **how to transfer a solution method to new problems** — Transfer of methods is a general learning skill. |
| Reversi/Othello's core primitive is: | **flipping ownership (territory)** — Othello is about outflanking to flip discs — a territory/flip game. |
| Gridley's "five games, four ideas" lesson is that different games: | **reuse the same core ideas (transfer)** — The family shares a few primitives that transfer across games. |
| The "who-moves-last" (parity) idea appears in BOTH: | **Dots and Boxes AND checkers endgames** — Parity/tempo shows up in Dots and Boxes and checkers endgames alike. |
| A "safe two-way connection" idea (the bridge in Hex) is like the "open-both-ends" idea in: | **Connect-4/Gomoku (two ways to win)** — Both give two options so one enemy move can't stop both. |
| Corners in Othello and the king row in checkers both represent: | **high-value squares worth fighting for** — Both are key squares that confer lasting advantage. |
| The idea "make a gain that can't be undone" appears in Othello as: | **stable discs / corners** — Permanent (stable) gains are Othello's version of the idea. |
| Dots and Boxes' core primitive is: | **parity / who-moves-last** — Dots and Boxes is a parity game about who is forced to open chains. |
| The four big ideas across GridForge are roughly: | **connection, forks/threats, flipping/territory, and parity** — Connection, threats/forks, flipping/territory, and parity span the family. |
| If a new game rewards "trapping and flipping pieces," think of: | **territory/flipping (Reversi)** — Flip-ownership games call for Othello-style territory thinking. |
| Transfer turns "I learned Connect-4" into: | **"I understand connection games"** — Transfer generalizes one game into a class of games. |
| If a new game rewards "who is forced to move," you should think of: | **parity (Dots and Boxes, checkers tempo)** — Forced-to-move games are parity games — apply parity thinking. |
| Transfer is powerful because ideas learned in one game: | **often work in many others** — A well-learned primitive generalizes across games. |
| The habit of asking "what's the shared idea?" | **faster, more flexible learner** — Seeking shared structure |

| | |
|---|---|
| makes you a: | accelerates and deepens learning. |
| Connect-4, Gomoku, and Hex all share the primitive of: | **connection (building a line/path)** — All three are connection games — build a connected line or path. |
| The fork in Connect-4 and the double-three in Gomoku are: | **the same idea (two unstoppable threats)** — Both create two threats one move can't both stop. |
| If a new game rewards "building a line or path," think of: | **connection (Connect-4, Gomoku, Hex)** — Line/path building is the connection primitive. |
| When you meet a NEW grid game, a smart first question is: | **which known primitive does this use?** — Identify the core primitive to apply what you already know. |
| Big idea: many grid games are built from a few shared: | **primitives — connection, threats/forks, flipping/territory, parity** — Spotting the shared primitive lets you understand a whole family of games. |
| Gridley would say the point of playing five games is to discover: | **the few big ideas they share** — The goal is to see the shared primitives, not memorize five separate tricks. |
| Hex's "blocking = connecting" duality is a special case of: | **the connection primitive** — It's the connection idea, sharpened by Hex's no-draw topology. |

## 16. mastery synthesis

| ✂ Question (fold →) | Answer |
|---|---|
| An "open four" in Gomoku is: | **unstoppable (wins next move)** — Both ends open = two ways to make five = unstoppable. |
| "Parity" in Dots and Boxes decides: | **who is forced to open the first chain** — Parity determines who must open — the key to the game. |
| Hex is special among these games because it: | **can never end in a draw** — Hex's topology guarantees exactly one player connects — no draws. |
| Corners (Othello) and the double-cross (Dots and Boxes) both show that: | **controlling the right thing beats grabbing the most now** — Both reward control/permanence over immediate quantity. |
| A Hex "bridge" is: | **a safe two-cell connection** — A bridge links stones with two empty cells so one move can't cut both. |
| When meeting a new grid game, the smart first question is: | **which shared primitive does it use?** — Identify the primitive to transfer what you know. |
| The fork idea transfers between: | **Connect-4 and Gomoku** — Forks (two threats) appear across the connection games. |
| The "double-cross" means leaving the last __ boxes of a chain: | **two** — Leave the last two boxes to keep control (all-but-two). |
| In Dots and Boxes, completing a box lets you: | **claim it and move again** — A completed box grants an extra move. |
| Connect-4, Gomoku, and Hex share the primitive of: | **connection** — All three are connection games. |
| Hex can never end in a: | **draw** — In Hex, someone always connects — no draws. |
| Connect-4, with perfect play, is a: | **first-player win** — Connect-4 is a solved first-player win (center opening). |
| Dots and Boxes' primitive is: | **parity / who-moves-last** — Dots and Boxes is a parity game. |
| Parity connects Dots and Boxes to: | **checkers endgames (tempo)** — Both use who-moves-last (parity/tempo). |
| A corner in Othello: | **can never be flipped (stable)** — Corners can't be outflanked, so they're permanent. |
| A "fork" wins because the opponent has: | **one move but two threats** — Two threats, one block — the fork wins. |
| Reversi's primitive is: | **flipping ownership (territory)** — Reversi is a flip-ownership territory game. |
| In Reversi/Othello you win by having: | **the most discs at the end** — Most discs of your colour at the end wins. |
| Gomoku is won by: | **five in a row** — Gomoku needs an unbroken five-in-a-row. |
| You should avoid drawing the __ side of a box: | **third** — Drawing the 3rd side gifts the box to the opponent. |

| | |
|---|---|
| Across GridForge, the recurring skills are: | **seeing structure and thinking a move ahead** — Structure-spotting + look-ahead run through every game in the family. |
| A solved game's result is known assuming: | **perfect play by both sides** — Solved = perfect-play outcome known. |
| Connect-4 is won by: | **four in a row** — Four-in-a-row (any orientation) wins Connect-4. |
| "Mobility" in Othello means: | **how many legal moves you have** — Mobility is your number of legal moves. |
| Big idea: GridForge shows that many grid games are built from: | **a few shared primitives — connection, forks, flipping/territory, and parity** — Master the handful of primitives and you master the whole family. |

# Keep going

You practiced **400 cards** from GridForge. Come back to the ones you missed — spaced-out practice makes them stick.

*Spark & Anvil is a 501(c)(3) public charity. Every app, story, and book is free, forever — no ads, no in-app purchases, no tracking.*


## Keep exploring online

| Play it now | Meet the cast | More free books |
|---|---|---|
|  |  |  |
| `https://spark-and-anvil.org/play/gridforge` | `https://spark-and-anvil.org/cast/gridforge` | `https://spark-and-anvil.org/books` |

*Scan a code with any phone camera, or type the address. Every app, story, and book is free, forever — no account, no tracking.*