Magic Is in the Air (Secret Solution)

Alex Walker, Chris Yu

The number of distinct cards in Magic: the Gathering exceeds 25,000. Why did we choose these 14, especially since some of them (like Priest of Gix) don't even have two-word names? This peculiarity is a strong indication that this puzzle contains secret content.

To get started, we consider the following line from the flavortext: ***Can the right combo make a flightless bird into a flier?*** Two of the cards – Zirda, the Dawnwaker and Basalt Monolith – form a well-known combo that can produce an arbitrary amount of colorless mana. This suggests that it might be possible to find the "right combo" for... something. To find this something, we look at more cards.[1] Two stand out: Chocobo Racetrack, which can create a non-flying Bird token, and the enchantment Super State, which grants flying to the creature it enchants. If we can play both, we can answer the question in flavortext in the affirmative.

This is very easy to do if we have infinite resources. To make things more interesting, we assume in this puzzle that we begin with zero resources (except the 14 cards found in the front half). Under this assumption, there is a unique order in which the 14 cards can be played that produces a game state in which we control a once-flightless Bird with flying:

#	Card Name:	Annotation:
1	Black Lotus	Cast Black Lotus and sacrifice it for 💀💀💀.
2	Priest of Gix	Pay 💀💀💀 to cast Priest of Gix. When it enters, add 💀💀💀.
3	Burnt Offering	Pay 💀 to cast Burnt Offering, sacrificing Priest of Gix and adding 🔥🔥🔥. Our mana pool is now 💀💀🔥🔥🔥.
4	Basalt Monolith	Pay 3 (using 💀💀🔥) to cast Basalt Monolith. Tap the Basalt Monolith for 3. Our mana pool is now 3🔥🔥.
5	Zirda, the Dawnwaker	Pay 1🔥🔥 to cast Zirda, leaving 2 in our mana pool. Zirda's ability reduces the untap cost of Basalt Monolith to 1. We pay 1 to untap Basalt Monolith, then tap it for 3. We repeat this process to generate an arbitrary amount of colorless mana.
6	Agility Bobblehead	Pay 3 to cast Agility Bobblehead and tap it for 🌳.
7	Food Chain	Pay 2🌳 to cast Food Chain. (Our mana pool contains only colorless mana now.) Exile Zirda using Food Chain, adding 💧💧💧💧 (which can only be used to cast creature spells).
8	Clever Conjurer	Pay 2💧 to cast Clever Conjurer, then exile it to Food Chain for 💀💀💀💀. Our mana pool consists of 💧💧💧💀💀💀💀 (for creatures) and a large amount of colorless mana.[2]
9	Thoughtcutter Agent	Pay 💧💀 to cast Thoughtcutter Agent, then exile it to Food Chain, adding 🌳🌳🌳. Our mana pool contains 💧💧💀💀💀🌳🌳🌳 (for creatures) and a large amount of colorless mana.
10	Yarok, the Desecrated	Pay 2💀🌳💧 to cast Yarok. Our mana pool contains 💧💀💀🌳🌳 (for creatures) and a large amount of colorless mana.

[1] Click here to view the 14 cards in an image gallery.

[2] Note that we are unable to activate Clever Conjurer's ability due to summoning sickness.

#	Card Name:	Annotation:
11	Corsair Captain	Pay 2💧 to cast Corsair Captain. Since we control Yarok, this creates two Treasure tokens when it enters. Our mana resources are now 💀💀🌳🌳 (for creatures), colorless mana, and two Treasures.
12	Chocobo Racetrack	Sacrifice the two Treasures for 🌳🌳 and pay 3🌳🌳 to cast Chocobo Racetrack.
13	Enchanted Prairie	Play Enchanted Prairie as our land for the turn. When it enters the battlefield, Chocobo Racetrack creates a 2/2 Bird creature token.
14	Super State	Pay 9 to cast Super State, enchanting the Bird, which gains flying.

To finish the puzzle, we simply index into the card names (now in play order) using the indices 10 6 2 13 6 2 7 6 1 13 12 8 9 6 provided with the puzzle. This produces the answer **STUTTGART CARDS**.

Construction Notes:

We managed to trick our editor into approving a Magic: the Gathering puzzle on the condition that the base path didn't require the solver to know how to play the game. Given this constraint, we set out to create a puzzle that would extract two answers: one using an order provided by the puzzle and another using a secret "combo" order which required some game knowledge. This meant that both extractions would have to use the same list of indices (lest we create a red herring on the base path).

Our particular objective – make a non-flying Bird into a flier – was chosen for a few reasons: it resonates with the penguin theme, it has clear mechanical interpretation in Magic, and it's easy to describe in common terms. Unfortunately, most Birds in Magic already have flying (or some conditional form of it). Indeed, when this puzzle was written, there were only 26 birds in Magic that didn't have some form of in-built flying. These include some moas, chickens, and owlbears, but no penguins.[3]

Since we wanted this puzzle to be about penguins (and not owlbears, say), we wanted to avoid using a flightless bird with an obviously-non-penguin name. Our solution was to use the flightless 'Bird' creature token, which was introduced to Magic in the 2025 Final Fantasy crossover set. This token is created by seven cards, one of which is Chocobo Racetrack.

Combo construction was split into two halves: resource generation and pay-off. One issue with resource generation is that the useful cardpool – mana-positive cards and traditional 'combo pieces' – is relatively narrow. We tried several engines before landing on the current engine using Zirda and Basalt Monolith.[4] The ability to generate infinite *colorless* mana and then slowly filter it into colored mana using Food Chain offered a much-needed source of flexibility in construction. Likewise, Chocobo Racetrack was chosen because it allowed one of our cards to be an otherwise-unconstrained land.

The index-based extraction places two constraints on each card name. To satisfy both, we began with the extraction for STUTTGART CARDS, which placed constraints on each individual index. (For example, BASALT MONOLITH needed an index 6 or 15 to extract the T.) For the other extraction, we predetermine an index set and permute the set of cards, hoping that the act of indexing spells out PHOENICOPTERUS. Since the number of permutations (roughly 87 billion) is too large, this step is better done using a graph-based algorithm. Given a list of 14 cards and their proposed indices, one can construct a (bipartite) graph in which an edge is drawn between a card c and an integer $1 \leq n \leq 14$

[3] After writing this puzzle, two "penguins" – Otter-Penguin and Sledding Otter-Penguin – were introduced in a set based on *Avatar: the Last Airbender*.

[4] Some other realizable engines include Black Lotus / Salvaging Station / Skittering Precursor and Animate Dead / Worldgorger Dragon. Many famililar two-card combos are unusable because the order in which the combo pieces are played can't be enforced.

if and only if the n-th letter of PHOENICOPTERUS can be obtained by indexing into c using the n-th index number. The proposed set of indices is then usable if the corresponding graph admits a structure called a ***perfect matching***. These can be detected efficiently.

Once the set of 14 cards and indices were established, we just needed to write everything else. Unlike the previous steps, which took around six weeks, this step was completed in about a day (after some false starts). The main challenge was to write a puzzle with the following properties:

a. It should matter that these 14 cards are Magic cards.

b. Knowledge of Magic should not be necessary to solve this puzzle.

c. To tie things together, the theme of the puzzle should involve flight.

Parts (a) and (b) were satisfied by the step in which partial card names were combined. (The fact that each card name could be roughly split in two was pure coincidence and not a constraint during construction of the secret half.) To satisfy (c), we developed the mechanic in which penguins formed flight paths through a grid of letters.