Constraints as Catalysts: A (De)Construction of Codenames as a Creative Task

Brad Spendlove and Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
brad.spendlove@byu.edu, ventura@cs.byu.edu

Abstract

Constraints are a common feature of creative domains, and the presence of constraints often facilitates creative outcomes. We examine the relationship between constraints and creativity by examining the competitive language game Codenames. We characterize the game space of the Codenames spymaster role by describing a set of successive constraints that give rise to the game. This constraint-centric characterization both demonstrates that the game is successfully designed to allow for creative play and also serves as a basis for a computational analysis of the spymaster task. We consider some of the implications of this characterization generally, and how we can think about both the game and the abstract principles it instantiates from the standpoint of a computationally creative system for playing it and what we might learn about creative search by building such a system.

Introduction

The concepts of constraint and creativity seem to enjoy an intimate relationship: creativity is necessary when constraints are present and constraints are necessary for creativity to be present. In what follows, we illustrate this in the context of a competitive language game, presenting a series of thought experiments while "designing" the game and examining a critical constraint "phase transition" that appears crucial to the game's success both as a game and as a creative exercise. We then consider the task imposed by the game from a computationally creative standpoint.

Competitive language games have recently been suggested as an interesting domain for creativity because they offer a proxy measure for (successful) creativity in the form of winning the game (Spendlove and Ventura 2022). Codenames (Chvátil 2015) is a well-known example of the genre, and we will assume a familiarity with it as we proceed, as it provides an interesting case study for our arguments.

To begin with, we note that constraints play a critical role in making the game an interesting/challenging/fun game, and we will demonstrate that momentarily. First, though, we note the existence of a meta-level creative task, presented by the game,[1] of figuring out the best general strategy given the constraints. The rules of the game (constraints) are designed to force this meta-level strategy to be one that requires new (base-level) creativity to solve each new game. We proceed by iteratively building up various constraints to illustrate the "emergence" of both the game of Codenames itself as well as (the need for) creativity in the context of the game. In a very real sense, these seem to exhibit a primal/dual relationship.

At an abstract level, the unconstrained task is to reveal 9 locations out of 25. That means there are $\binom{25}{9}$ different target states, with each instance of the game requiring the communication of one of them. Without additional rules forbidding this, one can easily accomplish the task by simply pointing to the 9 target locations. This is a solution strategy that wins the game in a single turn and is target-state agnostic (that is, it works equally well for any target state). While there are likely many ways of accomplishing this task, anyone first exposed to the game task is likely to immediately "invent" this solution, which requires only that the players can see and the spymaster can point. It is not at all surprising, and if it is employed, the game is no game at all.

However, we can introduce our first constraint by allowing only verbal communication, because this is a language game. What strategy might then be employed instead? We can invent a simple indexing scheme for the 25 locations and then give a sequence of 9 indices to communicate the targets. This strategy requires that the players understand the indexing system (they have to get together in advance to communicate it, or hope it is "self-evident" enough that it can be picked up on the fly). This is again a solution strategy that wins the game in a single turn and is again target-state agnostic.

We can introduce a further constraint by allowing the spymaster to communicate only a single clue word per turn.[2] A somewhat clever strategy involves constructing a mapping from the $\binom{25}{9}$ possible positions to the integers. Then, the clue word is just the appropriate integer.[3] To construct such a mapping, consider the board as an element of $\mathbb{B}^{25}$. Then the

[1]It is possible that many other or even all other creative tasks also present this same meta-task, though we don't explore that claim here.

[2]This ignores the additionally allowed clue number, but for the current discussion this is unimportant.

[3]For the sake of argument, we consider any integer a single "word".

integer for each board position is given by the binary number that has 1s in the 9 target positions and 0s elsewhere. This requires that the players understand binary numbers and how to convert between base 10 and base 2 representations. This is again a solution strategy that wins the game in a single turn and is again target-state agnostic.

To impose further constraint, we can disallow the binary code strategy by being a stickler about word count or by disallowing numbers.[4] A strategy for meeting this additional level of constraint involves the invention of a diabolical code that maps an ordered list of $\binom{25}{9}$ English words to the $\binom{25}{9}$ integers above, which works as follows. Associate the ith position in the word list with the ith 25-bit binary number that contains exactly 9 1s. This requires that the players have the same ordered list of words (which must be communicated in advance). In addition, if visual aids are disallowed, the players must be able to commit the list to memory. This is yet again a solution strategy that wins the game in a single turn and is also target-state agnostic. It also is beyond human capability.[5] Note that at each additional constraint level, we have produced a winning meta-level strategy: they all immediately win the game for any target state.

The constraints have become restrictive enough now to suggest a couple of potentially interesting Codename-specific questions:

- What is the best trick like this diabolical code that is not beyond human capability?[6]
- Does there exist such a strategy that wins in a single turn and is target-state agnostic?
- If not, does there exist a target-state agnostic strategy of any turn length?
- If so, what is the minimum number of turns required for the optimal target-state agnostic "cheat code"?

To constrain the task even further, we can disallow such cheat codes by requiring that the clue word "must be about the meaning of the [target] words".[7] Given this, we have reached the actual instantiation of the game of Codenames. And, it is interesting to note that it is now unclear that there exists any target-state-agnostic strategy that wins the game in a single turn, even given superhuman ability.[8] A corollary to this is that the game's rules have been designed in just such a way as to make the answer to this meta-level strategy question be that the best strategy now is game-specific.

[4] Numbers are allowed to be used in Codenames in some ways but not in others.

[5] And there's another difficulty—according to Oxford, there are currently $171,476 \approx \binom{25}{6}$ words in use in the English language, which means that there are roughly an order of magnitude fewer (current) English words than we require to implement our diabolical code solution.

[6] That is, what is the best "cheat" we could reasonably operationalize as human players?

[7] cf. Codenames rules.

[8] That is to say, we have not been able to invent such a strategy and therefore leave it as future work and/or an exercise for the creative reader to either produce such a strategy or prove it is not possible to do so.

Assuming this is the case, the natural follow-on question is, what is the game-specific strategy for winning as quickly as possible:

- Is there a game-specific, guaranteed, one-move strategy?

The constraints of the game appear to have transmuted the task, which now requires a new level of game-specific creativity—barring an epiphany that allows us to communicate positions directly, we instead are reduced to trying to communicate semantic relationships among words, by giving a single clue-word. This transmuted task can be represented as the discovery of meaningful relationships amongst a set of words. These relationships are dependent on the persons involved, their experiences, their knowledge and any shared knowledge/experience, and will evolve over time; as a result, there is no correct solution to the task (a hallmark of creativity "problems"), and such relationship artifacts can be novel, valuable, and surprising (more creativity hallmarks).

What is the key to the apparent "phase transition" from a less-constrained game that admits a meta-solution to the more-constrained game that (apparently) admits only game-specific solutions? For game variants with fewer constraints, all target states are in some important sense indistinguishable, so it is possible to find a general solution (which can still require creativity, of course). By contrast, as the constraints are increased to the point where semantic (or other types of) relationships between words become important, these relationship artifacts are no longer indistinguishable, so finding a general solution is nontrivial at best (and may likely be impossible).

Perhaps there is an argument to be made here that the genius of the game designer is in imposing constraints that disallow "cheating"/boring target-state-agnostic meta-solutions, while still maintaining enough flexibility to admit fun/creative target-state-specific solutions.

Clue Graphlets

The artifact created by a Codenames spymaster is more than just a word w (and number k); it is a graphlet of connections between words. The clue word w, drawn from the set of all English words, is the center node in the graphlet. The k word cards the spymaster intends to relate to the clue word are each connected by an edge to the center node. The center clue word and the number of connections k are given to the spymaster's teammates, and their task is to guess which word cards $\{c_1 \dots c_k\}$ fill in the graphlet. Figure 1 shows an illustration of this structure for $k = 4$.

For example, if the spymaster's team's word cards include "Plane" and "Ambulance", a potential clue word that relates to both could be "Vehicle." We can formulate this as a graphlet with "Vehicle" in the center, an edge to "Ambulance" and an edge to "Plane".

There are many ways that two words can be related. For the purposes of playing Codenames, however, we are only concerned with whether a potential clue word will direct the guessers to a given word card or not. Despite the many different forms this relationship could take, in practice it is usually intuitive for humans to determine. For example, "Vehicle" could serve as a clue for "Ambulance", but "Sky" would

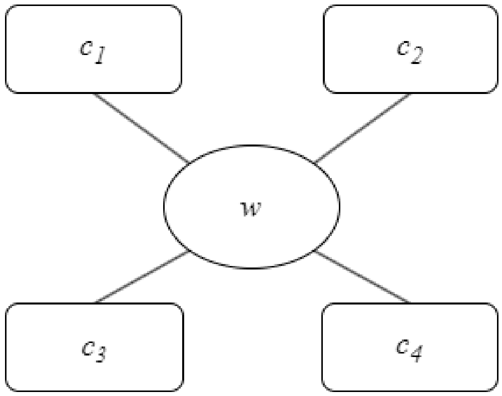

Figure 1: The structure of a Codenames clue graphlet with $k = 4$, consisting of a potential clue w and four word cards $c_1 ... c_4$.

likely not.[9] We can abstract this by considering a function $rel(w, c)$ that takes two words w and c, and returns `True` if w relates to c in this way or `False` otherwise.[10] This function can be used to evaluate the edges of a given graphlet. Call the potential clue word w and the set of word cards under consideration C, with $|C| = k$. If $rel(w, c_i)\ \forall c_i \in C$, then w is a good clue word for C.

The spymaster's job is to search through the set of English words for one that relates so well to k target words that the guesser's job is easy. Although the graphlet is the spymaster's creative artifact, only the clue word w and clue cardinality k are given to the guesser. The better the clue word, the more obvious it makes the graphlet's relationships. Note that this constraining of communication between the spymaster and the guesser(s) suggests an interesting alternative interpretation/understanding of the game—it is a game about a type of co-creativity—the spymaster creates an interesting/useful graphlet and then attempts to help the guesser discover that same graphlet by giving a hint. Viewed through the lens of creativity, the game requires creativity from both the spymaster and the guesser, with the spymaster first being creative and then guiding the guesser(s) to be creative in the same way.

Spymaster Task Search Space

To analyze the computational difficulty of the spymaster's task, we will first reason about the number of potential graphlets that the spymaster must search for a clue pair (w, k). For one move of the game, the board will have at most 9 word cards belonging to the spymaster's team. The spymaster may choose any number of those cards to incorporate into their chosen graphlet. An upper bound on the total number of unique configurations of word cards included (or not) in the graphlet is therefore $(\binom{9}{0} + \binom{9}{1} + \cdots + \binom{9}{9}) = 2^9 = 512$.

For each of these graphlets, the spymaster must consider English words to fill in the center node. Given an estimate of $170,000$ English words in current use, that puts the total number of graphlets to be searched at $2^9 \times 170000 \approx 8.70e7$. These graphlets, however, only consider the relationships between the potential clue word and our team's word cards. If the potential clue also relates to one of the other team's words, a neutral word, or the assassin word, then that graphlet is not a good candidate to be chosen as the spymaster's clue that round. We observe that a single such undesirable relationship disqualifies a graphlet, so any graphlet needs only to be checked individually against the 16 disqualifying word cards (those not belonging to the spymaster's team), increasing the size of the search space to $2^9 \times 16 \times 170000 \approx 1.39e9$.

We can significantly reduce the number of graphlets that must be considered by only searching for graphlets with up to four word cards. According to the Codenames rulebook, choosing a successful clue that relates to four word cards is a difficult accomplishment. A system that could consistently create clues (w, k) with $k = 4$ would perform at a superhuman level. We can also exclude the trivial case of a graphlet with zero connections, as it is irrelevant to the game. This results in a reduction of the size of the search space to $(\binom{9}{1} + \binom{9}{2} + \binom{9}{3} + \binom{9}{4}) \times 16 \times 170000 = 6.94e8$ graphlets.

We can further reduce the size of this search space by observing that graphlet edges must be considered one at a time. Thus, to evaluate a graphlet with $C = \{c_1, c_2, c_3, c_4\}$, the agent must first evaluate the graphlets with $C = \{c_1\}$, $C = \{c_1, c_2\}$, and $C = \{c_1, c_2, c_3\}$. By caching those calculations, the agent does not need to (explicitly) consider graphlets of $k < 4$ separately. This leaves the agent with $\binom{9}{4} \times 16 \times 170000 = 3.43e8$ potential graphlets to search.

Finally, note that a rough estimate of a college-educated person's vocabulary is $30,000$ words (D'Anna, Zechmeister, and Hall 1991). Substituting that for $170,000$ in our calculations results in a search space containing $6.05e7$ graphlets. Therefore, reasonable estimates of an agent's vocabulary do not change the search space by more than an order of magnitude.

Given a vocabulary size and maximum graphlet size (e.g. $30,000$ and $k = 4$ respectively above), the number of graphlets likely cannot be further reduced *a priori*.[11] Further reduction of the search space (at compute time) would require employing search heuristics. We observe that even without employing such heuristics, the number of graphlets through with the spymaster must search is of a magnitude that is potentially computationally tractable. The determin-

[9] Of course, some non-obvious connection could be drawn between "Sky" and "Ambulance", but this would require that the spymaster and guesser both are aware of that relationship, which raises theory of mind questions.

[10] In practice, this function could return a more nuanced value, such as a real number in the range [0,1]. However, we can simplify this by assuming a threshold to convert the real into a Boolean.

[11] Another approach to refining the bound could be characterizing the connectivity of the specific word cards included in the Codenames deck. Cursory examination suggests that the word cards are especially evocative or easy to draw connections between. More rigorous data analysis may be able to identify relevant characteristics of those cards.

ing factor is not the size of the search space of graphlets, but the cost of evaluating them.

The Dual Nature of Human and Computer Spymasters

When designing creative computer systems, we naturally turn toward human performance at the creative task as a gold standard. In the case of the Codenames spymaster, it is clear that human players can execute the spymaster task (and enjoy doing so!) A primary tool in accomplishing this is our powerful language faculty. Although such abilities are by no means easy to implement computationally, we can use them as the basis for computational analysis of the spymaster task.

Whether an agent has a fast or powerful method for determining the relationships between words or not, the computational task is the same. All graphlets in the reduced set described previously are candidates for clues. Therefore, it is instructional to compare search strategies for each agent over the set of graphlets given a function rel that determines whether a relevant relationship exists between two words.

The computational cost for searching for a clue graphlet is the product of the time it takes to search the space of all graphlets and the time it takes to execute rel on each edge in those graphlets. We can therefore reason about the costs of four computational tasks: human search, human rel, computer search, and computer rel.

Human language faculties include the storage of complex networks of semantically related concepts (Collins and Loftus 1975). Given both an understanding of human language faculties and observations of successful human Codenames play, it can be inferred that humans can execute rel with significantly higher speed and accuracy compared to a computational implementation.

Humans and computers employ different search strategies, each drawing on their own strengths (He, Mao, and Boyd-Graber 2022). While it is difficult to reason about the exact human search strategy for the spymaster task, we can assume from many existing examples that the computer search will be faster. Computational search strategies are well-understood, and selecting the best for the task gives us a lower bound on computer search speed.

This leaves us in the curiously complementary situation that *the human spymaster has a fixed, efficient rel function and the computer spymaster has a fixed, efficient search function*.

The skill and creativity exhibited by the human playing the spymaster reside primarily in their ability to effectively navigate the imposingly large search space. The better the human can do that, the better they will perform at the overall task.

This aligns with intuition and observation of human Codenames play. The human spymaster brings a practically immutable set of semantic relationships to the game and can exercise different search strategies in an attempt to maximize performance. As this work primarily concerns computational creativity, we defer further exploration of human spymaster strategies to future work.

With a computer agent's relatively low search cost, the gap between computer and human spymaster performance comes down to the cost of the computer's rel function. Methods for designing efficient implementations of this and related functions are open research questions. Computer models of semantic meaning have been the subject of ongoing study in the fields of natural language processing and machine learning (Otter, Medina, and Kalita 2020).

One approach to isolating and analyzing this function is employing ideal module prototyping (Spendlove and Ventura 2020). This paradigm sees creative system designers replacing flawed computer task modules with human-delegated versions. By so doing, the system's architecture can be tested without the confounding factors of inefficient or incorrect task modules. Once the module-agnostic design is validated, effort can be expended to improve the flawed module with the assurance that any low-quality output is not due to other factors.

Discussion

We have demonstrated that reasoning about Codenames as a set of constraints facilitates a thorough characterization of the game and its creative tasks. It also allows us to rule out any potentially pathological play patterns. The constraints implicit in the game's design delineate a search space of clue graphlets that we have explored in some detail.

The designs of other games may also be deconstructed in a similar way to allow for more explicit characterization of the games' search spaces. Of course, for some types of games, such as abstract strategy, this analysis may be trivial and unnecessary. Language games, however, are a popular genre of game that inherit the complexity and open-endedness of human language. Language games may be a source of many well-defined creative tasks that could be excellent candidates for computational creativity research. Our constraint-centric analysis can serve as a template for analysis of such games and their creative tasks. Future work may generalize this hierarchical constraint characterization to creative tasks in general.

Our analysis of Codenames highlights the relationship between rules and creativity, demonstrating how the introduction of certain constraints can act as an intentional catalyst for creativity. We observe that the addition of specific constraints to expansive or mundane game spaces can unlock the potential for creative gameplay.

We have demonstrated how adding constraints to a trivial word identification game transforms it into the intriguing and creative game of Codenames. These constraints act as focal points that channel players' attention, encouraging them to explore, experiment, and discover creative solutions. Constraints establish boundaries, rules, or objectives that guide players' actions, transforming an otherwise unmanageable or uninspiring space into a captivating and intellectually stimulating environment. An interesting question for future work is if the abstract notion of hierarchical characterization of constraint generalizes to other specific (language) games and/or example tasks.

Another angle for future work could address the possibility that the addition of specific constraints may induce a phase transition from games (or other types of tasks/domains) that do not allow for creativity and those that do. The concept of a phase transition, commonly encountered in physics and other complex systems, refers to a qualitative change in the behavior of a system as the result of external factors or internal conditions. In the context of game design, we have shown an example of how the introduction of specific constraints can lead to a significant shift in the creative dynamics within the game. The obvious follow-on question is whether there is a general principle that elucidates the relationship between constraints and creativity.

References

Chvátil, V. 2015. *Codenames*. Kladno, Czech Republic: Czech Games Edition. Board Game.

Collins, A. M., and Loftus, E. F. 1975. A spreading-activation theory of semantic processing. *Psychological Review* 82(6):407.

D'Anna, C. A.; Zechmeister, E. B.; and Hall, J. W. 1991. Toward a meaningful definition of vocabulary size. *Journal of Reading Behavior* 23(1):109–122.

He, W.; Mao, A.; and Boyd-Graber, J. 2022. Cheater's bowl: Human vs. computer search strategies for open-domain QA. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 3627–3639.

Otter, D. W.; Medina, J. R.; and Kalita, J. K. 2020. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* 32(2):604–624.

Spendlove, B., and Ventura, D. 2020. Humans in the black box: A new paradigm for evaluating the design of creative systems. In *Proceedings of the 11th International Conference on Computational Creativity*, 311–318.

Spendlove, B., and Ventura, D. 2022. Competitive language games as creative tasks with well-defined goals. In *Proceedings of the 13th International Conference on Computational Creativity*, 291–299.