TropeTwist: 
Trope-based Narrative Structure Generation

Alberto Alvarez  
Malmö University, Game Lab  
Malmö, Sweden  
alberto.alvarez@mau.se

Jose Font  
Malmö University, Game Lab  
Malmö, Sweden  
jose.font@mau.se

ABSTRACT
Games are complex, multi-faceted systems that share common elements and underlying narratives, such as the conflict between a hero and a big bad enemy or pursuing a goal that requires overcoming challenges. However, identifying and describing these elements together is non-trivial as they might differ in certain properties and how players might encounter the narratives. Likewise, generating narratives also pose difficulties when encoding, interpreting, and evaluating them. To address this, we present TropeTwist, a trope-based system that can describe narrative structures in games in a more abstract and generic level, allowing the definition of games’ narrative structures and their generation using interconnected tropes, called narrative graphs. To demonstrate the system, we represent the narrative structure of three different games. We use MAP-Elites to generate and evaluate novel quality-diverse narrative graphs encoded as graph grammars, using these three hand-made narrative structures as targets. Both hand-made and generated narrative graphs are evaluated based on their coherence and interestingness, which are improved through evolution.

CCS CONCEPTS
• Applied computing → Computer games  • Theory of computation → Grammars and context-free languages.

KEYWORDS
Authoring Tools, Narrative Generation, Evolutionary Computation, MAP-Elites, Computer Games

ACM Reference Format:
Alberto Alvarez and Jose Font. 2022. TropeTwist: Trope-based Narrative Structure Generation. In FDG ’22: Proceedings of the 17th International Conference on the Foundations of Digital Games (FDG ’22), September 5–8, 2022, Athens, Greece. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3558588.3563271

1 INTRODUCTION
There exists a plethora of games\(^1\), with diverse genres and each containing a different set of gameplay mechanics, audio, level, graphic, and narrative facets. The creation and combination of these facets make game development a hard task, commonly involving a diverse group of developers [12]. Likewise, the generation of these facets in conjunction has been categorized as one of the biggest and most challenging tasks within computational creativity [36, 37]. However, games share common elements and underlying narratives, but it is non-trivial how to identify these, how to define and analyze these games structurally, or what type of common underlying structures exist; pointed out as well by [5, 59].

Among the different facets, narrative stands out in games as it helps to create meaning, make sense of situations, and make games [stories] recognizable [2, 32, 38, 56]. Narrative structures can be used to describe how an experience or story is to be developed as argued by Barthes [11], and to create an abstract representation based on the narrative structure instead of a temporal and partially-ordered sequence of events [51]. Common narrative structures used in many domains are Aristotle’s drama structure, which subdivides a story into exposition, climax, and resolution or Propp’s analysis on the morphology of Russian folktales, which revealed a common structure among them, denoted as Propp’s 31 “narremes” [45].

This paper presents TropeTwist, a preliminary system that uses Tropes [21, 54] extracted fromTvTropes [26, 46] as patterns and fundamental units, which when combined can compose structures further representing other composed tropes. Common narrative structures can be identified and defined using TropeTwist. TropeTwist can define generic aspects of a story, leading to the identification of events, roles, and narrative elements, as well as a novel way to form narratives. As a proof-of-concept, we built, analyzed, and described structurally three game examples shown in figure 1, top row.

We propose graph grammars as indirect encoding of narrative graphs and the use of the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [40] to generate novel variations (shown in figure 1, bottom row) using the proof-of-concept examples as roots. Simultaneously, we propose metrics to evaluate the resulting narrative graphs’ coherence, cohesion, and interestingness. Our preliminary results show that we can produce more interesting structures retaining coherence based on our metrics.

2 RELATED WORK
Propp [45] analyzed Russian folktales identifying their fundamental structure in 31 steps. His work contributed to the identification of core elements, the proposal of actions and events as functions and narrative atoms, and roles that are recurrent within the folktales. Propp emphasized that these 31 functions and their arrangement were the structure and what gave meaning to the story discourse. Barthes [11] proposed three intertwined and progressively integrated levels in narrative work: functions, actions, and narration.

\(^1\)For instance, currently there are more than 68k games in steam https://store.steampowered.com/search/?category1=998.
His work is characterized by the proposal of fundamental narrative units in the function level to better assess and identify structures in a narrative. Furthermore, Baikadi and Cardona-Rivera [10] further discuss these fundamental units as narremes encoding narrative state and how they could be combined to narrative structures. Their work, similar to TropeTwist, proposes a graph structure of interconnected narremes. However, they defined narrative axes like Barthes, where each connection between narremes means a change along a narrative axis. In games, the narrative is usually directed by quests, which Aarseth [1] discusses as a central element in games to make sense of other elements, and which are defined by Yu et al. as a form of structure, dividing the story into achievable rewards and partially ordered set of tasks [62].

Furthermore, the generation of narratives, stories, and quests using a variety of techniques such as planning algorithms [47, 61], grammars [7, 27], or machine learning [53, 58], is a growing and important field within games research and narrative research in general [18, 22, 32, 62]. One typical approach for the generation of content and stories is the use of patterns representing different elements such as level design patterns [4, 56], quest patterns and common quests in games [16, 57], or identifying fundamental units and assembling them based on various pre-conditions [20, 31]. A particular type of pattern is tropes, which are concepts that are recurrently used in transmedia storytelling [21, 46]. Horswill [28] focused on constructing an expressive language that could encode plot tropes as story fragments, composing a database of fragments combined sequentially with a planner. Similarly, Thompson et al. [54] used the idea of tropes as story bits where a system would construct valid stories from users’ defined story bits with pre- and post-conditions. TropeTwist uses the idea of tropes for nodes and patterns in structures and encodes and represents these as a graph. Scheherazade is a system that can capture narrative structures by encoding and annotating narrative texts, which introduced the Story Intention Graph model, a formal and expressive representation of narratives [19].

Moreover, we use graph grammars and grammar recipes to generate structures. This approach is similar to how Dormans and Bakkes [17] generate missions and space using a “key and lock” structural idea. Our approach uses MAP-Elites, a quality-diversity algorithm that uses behavioral dimensions that are orthogonal to the objective function to store diverse individuals in a grid [40]. Evolutionary algorithms are a popular approach in PCG to generate diverse type of content [55], but not as much for narrative content. MAP-Elites have been used to generate content in different game facets such as levels [5, 15], mechanics [14], or enemy behavior [29].

Assessing narratives is a complex and non-trivial task. The goal is to create a narrative that is both syntactically correct (e.g., coherent and consistent) and semantically rich (e.g., novel and interesting) [13, 25, 48]. Perez y Perez and Ortiz [44] proposed a model to evaluate interestingness based on novelty and correct story recount, with emphasis on the story’s opening, closure, and dramatic tensions. Szilas et al. [52] discuss interestingness as a paradox dramatic situation with obstacles and conflicts, albeit applicable to stories as successive events. Yet, to approach subjective measurements such as interestingness, most research turns towards having human evaluation [30, 33] or using such to form human models to be used as surrogate models [35, 50].

3 BUILDING NARRATIVE STRUCTURES WITH TROPES

In storytelling, a trope [21] is a convention or figure of speech that the storyteller assumes to be recognizable by the audience. TvTropes is an online wiki that compiles and describes several
Table 1: Tropes included and used in TropeTwist, extracted from [46].

| Name               | Symbol | Definition                                                                 |
|--------------------|--------|---------------------------------------------------------------------------|
| Hero               | HERO   | A protagonist character.                                                   |
| Five-man band      | SMA    | Group composed by up-to-five archetypical characters.                      |
| The chosen one     | NEO    | Specific hero chosen as the one.                                           |
| Superhero          | SH     | Specific hero with unique abilities.                                      |
| Conflict           | CONF   | Non-specific problem to overcome between characters.                      |
| Enemy              | ENEMY  | A nemesis to the hero.                                                    |
| Empire             | EMP    | Collective enemy with the ambition of conquering the world.                |
| Big bad            | BAD    | Specific enemy, which is the ultimate cause for all the bad.               |
| Dragon             | DRAKE  | Specific enemy, which is the right hand of BAD.                            |
| Plot device        | PLD    | A feature or element that drives the plot forward.                         |
| Chekhov’s gun      | CHK    | PLD relevant to the story                                                  |
| MacGuffin          | MCG    | PLD with irrelevant nature to drive the story.                             |
| May help in quest  | MHQ    | PLD important to resolve a conflict.                                       |

Furthermore, interconnecting tropes can give rise to other tropes and patterns, described in the following section. The nodes and their respective trope and pattern were chosen from a subset of tropes in generic categories such as heroes or plot devices. These categories were inspired and chosen based on tropes from TVTropes, the division by James Harris [26], and previous research such as Propp’s morphology [45] or Greimas’ actantial model [24].

3.2 Trope Patterns

Tropes and interconnected tropes (i.e., subgraphs) give rise to different types of patterns. These patterns can be micro-patterns, encapsulating a single trope node. meso-patterns, often composed by more than one micro-pattern with special meaning, and auxiliary patterns, denoting graph problems. We calculate the relative tropes and patterns’ quality within an NG and use this to assess the general quality of the graph. These qualities are proxies for certain characteristics among the defined patterns that are used to evaluate the graphs, but they do not capture any story quality; especially, since we are only defining structures. When generating narrative graphs from a root (explained in section 4), the quality of a narrative graph becomes relative to the root, henceforth, the “root graph” (RG). In the following descriptions, we will use EG referring to the “evaluated graph” we are calculating the pattern’s quality (the generated individual), and RG to refer to the relative and root graph. When using subscript “pat,” we refer to the current pattern that is evaluated.

For most patterns, we calculate three general qualities (indicated when used) that add to the quality of the pattern. \( G_q(pattern) \) relates to the Generic quality of patterns in EG, which calculates the general occurrence of a pattern within EG compared to its occurrence in RG, calculated in eq. 1. \( R_q(pattern) \) relates to the Repetition quality of patterns, which calculates if a trope is unique in EG \( (R_q(pattern) = 1) \) or its ratio among the same base pattern. Lastly, \( I_q(pattern) \) relates to the Involvement quality of patterns in EG, which calculates the amount of associations a pattern has with structure patterns. Involvement means that the pattern is either source or target in a structure and is calculated as the ratio of structure pattern involvement by the structure pattern count in EG. These three metrics incentivize graphs with similar amount and type of nodes than RG, minimal repetitions, and more involvement.

\[
G_q(pattern) = 1.0 - \frac{|RG_{pat} - EG_{pat}|}{\max(RG_{pat}, EG_{pat})} \quad (1)
\]

3.2.1 Micro-Patterns. Micro-patterns are the fundamental unit in the system, which aims at categorizing different sets of the individual patterns that are shown in table 1. Micro-patterns are single nodes and the basic building block that, when interconnected, allows the detection of meso-patterns.

Structure Pattern (SP) is any type of trope that would give some structural definition to a narrative, whether this being a conflict, specific act, or a part in a dramatic arc (e.g., climax). Currently, the only type of structure trope is the conflict (CONF) trope, which represents the most basic structural interaction. The quality \( S_{p_q} \) is calculated as the equally weighted linear combination of:

\[
S_{p_q} = G_q(SP) + I_q(SP) \quad (2)
\]
Character Pattern (CP): are identified as nodes within the narrative that could be either the player, possible ally or enemy NPCs, or simple enemies. In TropeTwist, it is distinguished between heroes and villain patterns, and these are commonly used as sources or targets (or both) of other patterns, and on a few special occasions to denote a relation to another character. The quality $CP_q$ is calculated per group (heroes and villains), and it is the equally weighted linear combination of:

$$CP_q = G_q(CP) + R_q(CP) + I_q(CP)$$

(3)

Plot Device Pattern (PDP) is described as the element within the narrative that moves it forward, as a goal, object, or dramatic element. The quality $PDP_q$ is calculated as the equally weighted linear combination of:

$$PDP_q = G_q(PDP) + R_q(PDP)$$

(4)

3.2.2 Meso-Patterns. Meso-patterns are the features that emerge in the narrative from dynamically combining micro-patterns and, on some occasions, these with other meso-patterns. They are always composed of more than one pattern denoting some spatial, semantic, or usability relationship within the narrative graph. We identified a subset of Tropes (extracted from TVTropes [46]) that requires or works as the combination between more fundamental units. For instance, the reveal pattern relates to the "Good all along" or "evil all along."

Conflict Pattern (ConfP) is a type of structure pattern composed by a conflict node (Con), a source s node, and a target t node, which are both CPs and usually a hero and a villain or the same character as s and t. For instance, the subgraph HERO $\rightarrow$ CONFLICT $\rightarrow$ EMP, indicates that a hero CP has a conflict with an enemy CP. A conflict node can be used indefinitely to define several ConfP. A ConfP is also either explicit or implicit. Explicit conflicts are explicitly encoded in the graph and directed from s to t passing through the conflict trope. On the other hand, Implicit conflicts relates to the conflicts from t (or derivatives) to s (or derivatives) that are not encoded in the graph. For instance, the previous example is an explicit conflict from HERO to EMP, and at the same, the EMP has an implicit conflict with the HERO. The quality $ConfP_q$ is calculated as the equally weighted linear combination of:

$$ConfP_q = G_q(ConfP) + R_q(ConfP)$$

(5)

Derivative Pattern (DerP) defines a relationship between tropes connected by "entails" connections (\(\rhd\)). Therefore, a DerP contains a list of patterns connected by entails, named derivatives. DerP starts from a root micro-pattern and continue until no more "entail" connections are encountered, effectively establishing a hierarchy from the root derivative to the rest. By design, the patterns within a DerP have a local and temporal order and a causal relationship. For instance, in the subgraph EMP $\rhd$ DRAKE $\rhd$ NEO, engaging with the EMP, entails both the conflict with DRAKE and the appearance of NEO. This means that only by overcoming the DRAKE, NEO will appear - as a new hero or the evolution of another. The quality $DerP_q$ is calculated (eq. 6) based on its $G_q(DerP)$, the ratio of derivatives within the DerP among the total amount of derivatives across all DerPs in EG ($ratio\theta_q$), and the derivatives’ diversity.

$$DerP_q = G_q(DerP) + \frac{\sum l_{\text{DerP}_i} \cdot DerP_{\text{der}, basepat}}{\sum l_{\text{DerP}_i}}$$

(6)

Reveal Pattern (RevP) connects two independent CPs as one, meaning that character A was, in fact, always character B, and vice-versa. This pattern identifies confusion and surprise within an EG, as, for instance, a villain could have been, in fact, “Good All Along”². In practice, a RevP is identified as a villain or hero connected with a unidirectional connection (\(\rhd\)) to another hero or villain. As a consequence, all existing conflicts between them would become fake. $RevP_q$ is calculated based on its $G_q(RevP)$, the number of reveals in EG in relation to characters, and the number of fake conflicts given the specific reveal.

$$RevP_q = G_q(RevP) + \frac{\sum l_{\text{EG}_\text{RevP}}}{\sum l_{\text{EG}_\text{CP}}}$$

(7)

Active Plot Device Pattern (APD) operationalize and integrate PDPs within a narrative since PDPs only describe an abstract goal or target. In practice, an APD is identified as PDPs that have at least one incoming connection, and optionally, one single outgoing connection. These limitations are added to limit the effect of a PDP within a narrative. $APD_q$ is measured based on its $G_q(APD)$, and the APD’s usability, calculated based on the sum of incoming and outgoing connections divided by half of the nodes in EG depicted as $baly_q$, penalizing APDs for not using all their connections.

$$APD_q = G_q(APD) + baly_q$$

(8)

Plot Points (PP) are key events within the EG, identified as discrete moments given some pattern. The derivatives within a DerP, RevP’s source, and PDPs that are APD are considered as plot points. $PP_q$ is measured based on the number of PPs within RG ($G_q(PP)$), and the number of PPs within EG in relation to the number nodes within it ($Balance_q(PP)$).

$$PP_q = G_q(PP) + Balance_q(PP)$$

(9)

Plot Twist (PT) takes advantage of plot points to identify those that could have a bigger impact on the narrative. In practice, PTs consider the source of RevP, derivatives from DerP that are a different micro-pattern than the root of the DerP (except PDPs), and APDs that are connected to other APDs. For instance, in the subgraph: EMP $\rhd$ DRAKE $\rhd$ NEO, given that NEO is a different micro-pattern than root EMP (Hero and Villain, respectively), NEO will be identified as a Plot Twist as it alters the “natural” order in the DerP. $PT_q$ is based on the number of PTs within RG ($G_q(PT)$), the PT’s involvement in EG, and the balance of PTs based on the PPs in EG. Involvement varies depending on the associated pattern to PT. When a PT is associated with a RevP, involvement is calculated as how much the structure changes based on that (i.e., how many fake conflicts are created). When it is related to DerP, involvement

²https://tvtropes.org/pmwiki/pmwiki.php/Main/GoodAllAlong
We use the Constrained MAP-Elites [29], and adapt it to work with
which then would enable the player to get to their objective (MCG).

3.2.3 Auxiliary Patterns. Auxiliary patterns denote problems in
the graph and sub-optimal or impractical nodes and connections
within a graph. They are classified into Nothing, which are nodes
that are not identified as part of a meso-pattern; and Broken Link,
which are outgoing connections from a node that are not used or
do not lead to any pattern.

3.3 Proof-of-Concept
TropeTwist can be used to represent different narrative structures
and parts of games. To test and show TropeTwist’s expressiveness,
we chose to form three different narrative graphs representing
different games shown in figure 1, top row: Zelda: Ocarina of Time
(Zelda:OoT) [43], Zelda: A Link to the Past (Zelda:LttP) [42] - eastern
palace, and Super Mario Bros (SMB) [41]. They represent different
games from different genres (fig. 1.a and 1.b are adventure-dungeon
games, and 1.c is a platformer), and represent different game’s
phases; in the case of fig. 1.a and 1.c, both represent the main
structure of the game, while 1.b, represents a specific area and
sequence of the game.

Figure 1.a represents a simplified overarching narrative structure
from Zelda: OoT. The ocarina of time, given by Zelda to Link, is
defined as a McGuffin (MCG) that, when collected by “young link,”
allows him to go forward in time to “adult link,” the chosen one
(NEO). This, in turn, enables explicit conflicts between hero and
enemy characters, which represents the main loop of the game.
The structure shows two factions, a set of heroes and the BAD.

Figure 1.b represents the structure and plot points from the
eastern palace in Zelda: LttP. All palaces in A Link to the Past follow
a very similar structure and sequence. The HERO’s goal is to get
the “Pendant of Courage” (MCG). However, the MCG derives from
ENEMY and BAD, so the HERO must overcome them to achieve his
goal. The structure shows a causal and linear narrative that could be
used to identify elements that need to appear before others, similar
to the work by Dormans and Bakkes [17].

Figure 1.c represents the overarching narrative structure of SMB.
In SMB, the objective of Mario (HERO) is to rescue Peach (HERO)
from Bowser (BAD). To do this, the player goes through a series of
platform worlds that always end in a “Fake Bowser” (DRAKE). The
player must continue until encountering the “Real Bowser” (BAD),
which then would enable the player to get to their objective (MCG).

4 EVOLVING NARRATIVES WITH GRAPH
GRAMMARS
We use the Constrained MAP-Elites [29], and adapt it to work with
graph grammars, evolve production rules, and adapt the evolution
towards a target similar to [5]. Constrained MAP-Elites adds
feasible-infeasible two populations to each cell, effectively evolving
sub-populations per cell. An individual’s phenotype is a narrative
graph, and its encoding genotype is the production rules of a graph.
A graph grammar is a context-free grammar whose productions
add, remove, and modify nodes and edges of a graph. Our
implementation uses the tropes listed in Table 1 as nodes, and the
three available connection types as edges (→, ↔, ⊴→). Graph grammars
do not apply rules sequentially; instead, every individual does a
random sampling of the rules in their genotype to produce recipes
to generate graphs. Recipes describe the rules’ order and repetition,
and their size is limited by the amount of production rules as mini-
um and the minimum plus five as maximum. Recipes do not have
repetitions within them, i.e., if rule 1 is added at step 2, subsequent
addition would simply add to the number of times that rule will
be applied at step 2. Their size is limited by the number of produc-
tion rules as minimum and up to five more samples as maximum.
Figure 2 shows a sample complete process from an individual’s
genotype (i.e., rules) to the phenotype (i.e., narrative graph).

Individuals move between the feasible and infeasible population
depending on the feasibility constraint. NGs are deemed infeasible if
the nodes are not fully connected or if there exists a conflict pattern
with more than one self-conflict. Infeasible individuals are evaluated
based on how close they are to be fully connected and not having
any inadequate self-conflict. The fitness function assesses NGs that
are deemed feasible based on their coherence (equation 12), which
we use to assess how correct, coherent, and in general, syntactically
correct the narrative graphs are. Coherence aims at maximizing an
equally weighted sum between cohesion and consistency. Cohesion
refers to the link between elements that hold together to form some
group. In our implementation, it focuses on minimizing the number
of auxiliary patterns by calculating the proportion of Nothing and
Broken Link among all patterns in NG. A consistent NG should be
regular and free of contradictions. Thus, we calculate consistency
(eq. 11) as the collective quality of micro-patterns since they are the
building blocks, and conflicts’ goodness based on the number of fake
conflicts. Thus, we aim at maximizing the quality of micro-patterns
and minimizing contradictions created by meso-patterns.

\[
\sum_{\text{conf}(\text{PT})}\frac{\text{len}(\text{ng}_\text{conf})}{\text{len}(\text{ng}_\text{conf}_\text{pat})} = \frac{\text{len}(\text{ng}_\text{conf})}{\text{len}(\text{ng}_\text{conf}_\text{pat})} \tag{11}
\]

Furthermore, MAP-Elites uses behavioral dimensions in a grid
shape to retain and foster diversity throughout generations. We
use the following two dimensions to evaluate the diversity:

- **Step.** Step (eq. 13) calculates the Levenshtein distance [34] be-
tween two narrative graphs, taking into consideration the number
and type of nodes and connections. Step is normalized using step
threshold \( \theta = 11 \) determined through a process of experimenta-
tion, which does not consider steps farther than \( \theta \), avoiding the
generation of too dissimilar graphs.

\[
\text{D}_{\text{step}} = \min(|E_{\text{EG}}, |R_{\text{EG}}|, \theta) \tag{13}
\]

**Interestingness (int).** We aim at measuring the semantic qual-
ity of a narrative graph. A narrative graph can be syntactically
correct and coherent yet lack a good semantic quality and do not
evolve interest for designers or players. Therefore, we leverage plot
point, plot twist, and active plot device patterns to measure the
interestingness of the NGs. The nature of interestingness creates

We conducted a series of experiments to evaluate and analyze how the system could evolve NGs into quality-diverse and valid narrative structures. We evolved the three manually constructed narrative graphs shown in figure 1, top row. They were used as root graphs and explored the possibility space, and the seemingly competing qualities of coherence (i.e., fitness) and interestingness.

Furthermore, in figure 1, bottom row, it is shown three different example elite narrative graphs, generated from their respective root graphs on the top row and with each individual evaluation shown in table 2. The root graphs have a cohesion of 1.0 since none of them have unused nodes or connections and have similar mid-high consistency values because of using generic nodes (e.g., HERO or ENEMY), repeating them, and low involvement in structures by characters. In the case of fig 1.a1, the RevP from HERO to SH creates some fake conflicts, which affect the consistency but also boost the interestingness value of the narrative graph. Both fig 1.b1 and 1.c1, are evaluated similarly with low interestingness; c1 involves a simplistic and linear structure, and b1, while in principle more complex, is also a relatively linear structure with no PTs.

Furthermore, all the exemplar elites have better consistency, coherence, and interestingness than the respective root graph. In figure 1.a2, the graph has been reduced towards a bottleneck, RevP (HERO → SH) is removed, and MCG is added as the objective for SH, which could point towards competition or cooperation to enable NEO. Such a change gives more consistency to the graph while seemingly reducing its interestingness, but this relation and the connection between MCG and NEO increase its interestingness. In figure 1.b2, the narrative has more interaction between characters. In the case of fig 1.a1, the RevP from HERO to SH creates some fake conflicts, which affect the consistency but also boost the interestingness value of the narrative graph. Both fig 1.b1 and 1.c1, are evaluated similarly with low interestingness; c1 involves a simplistic and linear structure, and b1, while in principle more complex, is also a relatively linear structure with no PTs.
the emergence of the BAD. Finally, BAD is no longer connected to EMP and DRAKE; thus, BAD could be its own enemy faction, in this case, complexifying the narrative and creating more challenge.

5 DISCUSSION AND LIMITATIONS

The trope-graph representation in Tropetwist allows for a quick definition of narrative structures. They are, by design, ambiguous, do not encode temporal information besides causal chains, and are, to some extent, generic, which makes structures relatively simple to develop but more complex to interpret. These design decisions make the system encode less rich information than others, such as Scheherazade [19], but allow the structure to be interpreted in multiple ways. For instance, the generated graphs could equally describe different stories, and the interpretation given in this paper is just one of many. Thus, the system effectively shifts the complexity from the structure to the “interpreter.” While the generated structures could already serve as inspiration for users, an interpreter could provide alternative interpretations that could be guided by or learned from users, which is part of our future work.

Furthermore, the metrics proposed and developed here were used to tune and evaluate the graph outputs without humans in the loop. However, they do not stand in or replace human judgment. The metrics are estimated heuristics mainly based on the graph functionality and relation among patterns. Most of them are related to a “root graph,” which is a preliminary step for making Tropetwist interactive and have humans-in-the-loop. We aim to develop a mixed-initiative version of Tropetwist, where metrics depend on the designer’s creation. This would, in turn, allow the designer to steer the MAP-Elites search, generating content adapted to them [6], and for MAP-Elites to assist designers with ideation proposing varied structures.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented Tropetwist, a system that interconnects tropes and trope patterns to describe narrative structures. We demonstrated through three proof-of-concept structures the system’s expressiveness to describe games with diverse genres and mechanics, and different game phases. Further, we illustrated how we could generate novel structures from the three proof-of-concept structures using MAP-Elites, improving them on our metrics.

Tropes could be seen as something to avoid when exploring creativity, mainly due to the possibility of showing unoriginal views by definition. However, a set of combined tropes, patterns, and structures could give rise to novel combinations that express the wanted structure. Similarly, identifying, visualizing, and defining the tropes and patterns and doing “twists” with them; thus, transforming something typical into atypical is the goal with Tropetwist.

The narrative structures show essential aspects of how the story will develop and lead, and important components such as events, conflicts, or roles. However, to further operationalize these structures, it is necessary other systems that make use of them, such as quest [7, 8] or plot [9] generators. Another interesting future work would be to explore the multi-faceted nature of games [36] and combine this type of system with generators that focus on other facets such as level design [4, 49] or game mechanics [14, 23].

Generating novel narrative structures resulted in interesting variations, but the system could not exploit all the advantages of MAP-Elites. Our results point towards difficulties exploring the space, possibly because coherence and interestingness are to some extent competing objectives. Therefore, we aim at extending Tropetwist towards a mixed-initiative co-creative system [60], and with that, evaluate with human participants. Given that our metrics are dependent on the designed graph; then, we could constantly adapt the content generation and have adaptive models, for instance, of interestingness, based on the user’s creation similar to [4, 39].
