The Limitations of a Propp-based Approach to Interactive Drama

Zach Tomaszewski & Kim Binsted
Information and Computer Science Department,
University of Hawaii—Manoa, Honolulu, HI 96822
(ztomasze@hawaii.edu & binsted@hawaii.edu)
http://zach.tomaszewski.name/argax/

Abstract

Eudaemon—a top-down, directed interactive drama system—is based on the functions of Vladimir Propp and a number of existing interactive narrative systems. While proving to be of some success, the Eudaemon system also highlights a number of deficiencies that seem to be inherent to a function-based approach to interactive drama.

Propp's *Morphology of the Folktale*

Vladimir Propp (1968) explored the problem of classifying folktales. Previous attempts at classification had focused on the content of folktales, such as their general category—whether humorous tales, animal tales, or tales of everyday life—or their motifs—such as vampires, witches, or dragons. Instead of looking at the content, Propp examined the formal structure of tales.

Propp's work revealed a number of functions, where a *function* is "an act of a character, defined from the point of view of its significance for the course of the action" (Propp 1968, p.21). Furthermore, these functions are "independent of how or by whom they are fulfilled" (Propp 1968, p.21). For example, whether a tzar gives the hero a horse or the hero slays an evil sorcerer and takes his magical ring, the hero is effectively receiving a Magical Agent from a Donor. This Agent—whether a horse or a magic ring—might then serve to transport the hero to a new kingdom. This illustrates two functions: first, *Receipt of an Agent* (**F**), and then *Spatial Transference* (**G**).

Propp claimed that there are a limited number of such functions, finding only thirty-one of them in one hundred Russian folktales. Though not all functions appear in every tale, when they do occur, they always occur in the same order. Certain pairings and groupings are also evident. For instance, the details of the initial *Villainy* (**A**) are closely tied to those of the *Liquidation of Misfortune* (**K**), which resolves or undoes the villainy. The functions that define interaction with the Donor (**DEF**) usually occur all together.

Propp also outlines a number of character roles that occur in folktales, such as the Hero, Villain, Donor, Princess, and Helper.

Since Propp's functions "constitute the fundamental components of a tale" (Propp 1968, p.21), his work has served as a foundation for automatic story generation and, more recently, interactive narrative systems.

Previous Propp-based Interactive Narrative Systems

Grasbon and Braun (2001; Spierling, et al. 2002) used Propp's morphology in developing their augmented-reality interactive story-telling system, GEIST. GEIST builds stories piece-by-piece from a collection of pre-authored scenes, each of which is associated with a particular Propp function. Each scene also includes other metadata details that affect its selection, such as the scene's duration, required setting, and necessary story context. Besides indirectly affecting which scene is selected based on their location, the user may also directly determine the outcome of certain "polymorphic" functions based on their choices during that scene.

Fairclough's (2005) OPIATE system uses case-based reasoning based on Propp's functions. For its set of cases, OPIATE uses a number of example folktale plots, as translated into their component functions by Propp himself. OPIATE then tries to choose a case (that is, a plot) from this pool, casting characters into various story roles based on their opinion of the player's character (the Hero). If a good enough match cannot be found, OPIATE will combine existing cases to generate a new plot. The player indirectly affects the plot by affecting other characters' opinions of him. The player can also refuse to respond positively to certain functions.

Peinado and Gervás (2004; Peinado, Ancochea & Gervás 2004) have followed Fairclough's approach of using case-based reasoning with Propp functions. They have supplemented this with an ontology and a model of the user's role-playing style, both of which further affect the selection of the next scene. Each scene includes pre-conditions for its selection as well as post-conditions for its completion. A working prototype of their design is expected soon.

Prada, Machado, and Paiva (2000) use Propp's character

roles in their TEATRIX virtual environment but do not use his functions. TEATRIX is intended to aid collaborative story creation among school children, and so it offers users a greater degree of directorial control than a typical interactive drama.

Eudaemon: A New Propp-based Interactive Drama System

The goal of our Eudaemon architecture is to produce a *directed interactive drama*. By *interactive drama*, we mean a system in which the player assumes the role of a character in an interactive narrative. That is, as a character in a virtual story world, the player can manipulate objects and interact with character agents. These interactions of the player then become part of a well-formed story constructed at runtime (rather than completely pre-authored). This story should be adaptive enough that the user can affect its conclusion though their actions. By *directed*, we mean that the system includes a *drama manager* component that has the ability to manipulate the story world in order to influence and further the story. (This is in contrast to an *emergent* approach, where the story is not centrally directed but emerges solely from the player's interaction with autonomous character agents and the story world.)

We designed the Eudaemon system assuming an eventual prototype implementation using Inform, an interactive fiction platform. In such an implementation, the story world—including locations, props, and characters—would be defined using Inform. These world object definitions would include default reaction behaviors for most game events, such as being moved or talked to. The world would be presented to the user as text descriptions. The player would affect the world through typed commands, such as "`examine stone idol`", "`talk to witch`" or "`go north`". A player-entered command and the resulting system response would be a *turn*.

The Eudaemon drama manager is an agent external to the story world. Its basic purpose is to dynamically build a story piece-by-piece from pre-authored story components, which we call *scenes*. Selection of the next scene is determined by the current world state, which may be influenced by the player's actions. This is essentially the approach of Grasbon and Braun's GEIST, though Eudaemon also includes dynamic casting of character roles, similar to Fairclough's OPIATE. Mateas and Stern's *Facade* (Mateas 2002) also uses a similar method of building a story from pre-authored components, though they call those components *beats* and do not explicitly employ the higher-level distinction of functions.

The Eudaemon drama manager includes a story model, a set of assigned roles, and a collection of scenes. The story model is essentially a history of the story so far, recorded

Symbol	Function	Description
α	*Initial Situation*	Introduction to the hero by name or status, enumeration of family members and loved ones, establishment of initial location, etc.
γ	*Interdiction*	The hero is forbidden to do something.
δ	*Violation*	The hero violates the interdiction.
A / a	*Villainy / Lack*	The villain causes harm, injury, or misfortune to a family member. / The hero or a family member lacks something. (**a** serves as an alternative to **A**.)
B	*Mediation*	The hero is made aware of the misfortune or lack.
C	*Counter-action*	The hero agrees to attempt to resolve the misfortune.
↑	*Departure*	The hero leaves home.
D	*First Function of the Donor*	The hero encounters a donor, who greets or otherwise tests the hero.
E	*Hero's Reaction*	The hero reacts to the donor.
F	*Receipt of Magical Agent*	The hero receives some (magical) item, animal, or other assistance from the donor.
G	*Spatial Transference*	The hero is guided or transferred towards the object of his search.
H	*Struggle*	The hero and villain engage in combat or competition.
I	*Victory*	The villain is defeated.
K	*Liquidation of Misfortune or Lack*	The initial villainy is undone, or the initial lack is satisfied.
↓	*Return*	The hero returns home.
W	*Wedding*	The hero marries, ascends the throne, or receives some other reward.

Table 1: Subset of Propp's functions used in the Eudaemon system.

as a series of completed scenes. As every scene is associated with a Propp function, this history can also be viewed abstractly as a series of functions.

Eudaemon uses only a subset of Propp's functions (see *Table 1*). We did not include Propp's description of trebling or moves (a kind of sub-story), so each function occurs only once within a played story. Every story begins with an **α** scene. The other functions required by Eudaemon are **A**/**a**, **D**, **E**, **F**, **K**, and **W**. **H** is required if the story includes **A** rather than **a**. These required functions serve as a series of waypoints in the story. At any point in the story, the drama manager can attempt to activate all those scenes of functions up to and including the next required waypoint function. For example, after an **A** scene, the drama manager can attempt to activate any scenes for functions **B**, **C**, ↑, or **D**, though it favors scenes from earlier functions over later functions.

Eudaemon scenes are not internally interactive. Instead, they serve as kind of cut-scene, advancing the action in a single turn and then returning control to the player. During execution, a scene can assign character roles and make changes in the story world—such as introducing pre-authored locations, moving characters or props to a different location, or overriding the default reaction behaviors of the story world.

Besides its function, each scene also has other *preconditions* that must be met before it can be activated. These may include the player's current location, specific preceding scenes, and the previous casting of certain roles. A scene also has *firing conditions*. That is, it may be possible for a certain scene to occur at the current location and point in the story, but it still requires some action on the part of the user before it actually does so. Therefore, scenes are either *immediate* (occurring immediately upon activation) or *dependent* (requiring some further event before firing). Dependent scenes usually have a lifetime, expressed as a number of turns before expiration. Once a scene expires, it cannot be selected again during that story. A dependent scene may also be *submissive*, which means scenes from other functions may also be active at the same time.

Scene activation occurs in the following way:

1. All scenes for the next function that have satisfied preconditions are *selected*.
2. If one of these selected scenes is immediate, it is fired and scene activation ends. (If more than one scene is immediate, one of them is chosen at random.)
3. If no selected scenes are immediate, all selected dependent scenes are activated.
4. If only submissive dependent scenes have been activated, scenes for the following function are also activated.

In response to each turn of user input, the system:

1. Fires any active dependent scenes that were waiting for the current user action.
2. Allows default behaviors to respond to the user's action.
3. Checks that preconditions are still satisfied for all active dependent scenes.
4. Ages all active dependent scenes and removes any that have expired.
5. Repeats the scene activation sequence if no more scenes are active.

The story world includes a number of pre-authored characters that can be assigned to various story roles as the drama manager requires. Character definitions include a list of roles they can fill. A role is assigned only once, meaning that, once cast, a character will hold that role for the remainder of the story. A character may fill more than one role, but this is the exception rather than the rule. For instance, the Donor may offer to accompany the Hero, thereby becoming the Agent in later scenes. *Table 2* shows the character roles supported by Eudaemon and the function in which a character is first cast into that role. These roles are based on a subset of those described by Propp, with some minor changes.

Role	Cast in...	Notes
Hero	(**α**)	The player's character always fills the role of Hero
Loved One	**α**	Not a required role, but lacking a Loved One may limit which Villainies can occur. In a more complex system than Eudaemon, this role might be filled based on player interaction with characters before the **A**/**a** scene.
Villain	**A**	Stories with an **a** (instead of **A**) scene do not have a Villain.
Dispatcher	**B**	This role only appears in **B** scenes.
Donor	**D**	Should be filled by a character not yet cast in a role.
Agent	**F**	The only Eudaemon role that can be filled by an object instead of a character.

Table 2: Character roles used in the Eudaemon system.

Table 3 below provides a walk-thru of an example story from the game *Wharf*, which produces occult pirate fairytales. This example happens to use all the functions

possible in the Eudaemon system. The table shows how each successive scene is activated and fired. At each point, it also gives an alternate scene that could have been selected instead, given the same world state and story history at that point.

Function	Activation Details	Scene Contents	Example of Alternate Scene
α - Initial Situation	**Precond**: None. Randomly selected from all **α** scenes. **Firing**: Immediate.	Establishes the player's character (*Hero*) as an orphan, adopted by a ship's captain (*Loved One*) and now pursuing a life on the high seas. Sets the player's location on a ship docked at the wharf.	Establishes the player's character (*Hero*) as son of an old, infirm sailor (*Loved One*). Sets the player's location in a boarding house in town.
γ - Interdiction	**Precond**: Location on ship at dock. **Firing**: Immediate.	The captain says "Don't leave the ship! We're about to set sail."	[No alternates at this location.]
δ - Violation	**Precond**: Location on ship at dock; follows **γ**("Don't leave ship"). **Firing**: Dependent on player leaving current location. **Expires**: after 5 turns.	[In this example, the player does not leave the ship. This waiting scene times out, allowing the selection of another scene. As there are no other **δ** scenes at this location, the next possible scene is an **A**]	[No alternates at this location. If player had left the ship, however, it would set up other possible Villainies in town.]
A - Villainy	**Precond**: Location on ship at dock. **Firing**: Immediate.	The ship sets sail, and is then attacked by pirates, lead by the ominous, magic-wielding Black-Souled Jack (*Villain*). Jack kidnaps the captain.	**a - Lack**: The ship sets sail, but encounters a savage storm. The player is shipwrecked on a strange isle (setting up the need to get home.)
B - Mediation	**Precond**: Location on damaged ship at sea; follows **A**("Captain is Kidnapped"). **Firing**: Dependent on still being active at expiration. **Expires**: after 5 turns **Submissive**.	A sailor (*Dispatcher*) exclaims "The captain has been kidnapped! Someone must save her!"	[No alternates at this location.]
C - Counteraction	**Precond**: Location on damaged ship at sea. **Firing**: Dependent on player entering the row boat.	The player bravely sets out in the row boat.	Other **C**s waiting to fire this point might depend on the player attempting to signal for help or rallying the crew to follow her as a temporary captain.
↑ - Departure	**Precond**: Location in rowboat at sea. **Firing**: Immediate.	The hero washes ashore on a strange isle.	The hero is picked up by another ship, which carries her to some new land.
D - Donor	**Precond**: Location on beach of strange isle. **Firing**: Dependent on player leaving the beach.	The player meets a strange old woman (*Donor*) in a little hut, who invites the player in. The old woman possesses a magic mirror.	On another part of the island (either randomly or because the player chose a different direction), the player might meet a magical creature known as the Horax (*Donor).*

E - Reaction	**Precond**: Location in hut on strange isle. **Firing**: Dependent on player pushing the old woman into the fire.	The old woman burns up in the fire, leaving her magic mirror unprotected.	Other **E**s waiting to fire might include asking the old woman about Black-Souled Jack or stealing her magic mirror.
F - Agent	**Precond**: Location in hut on strange isle. **Firing**: Immediate.	The player now possesses the magic mirror.	[No alternates at this location.]
G - Transference	**Precond**: Location in rowboat on strange island; Jack is cast as *Villain.* **Firing**: Immediate.	The player heads out to sea once more to find Black-Souled Jack. After two days, Jack's ship captures her.	Alternately, if the player goes to the north of the island, a pegasus will carry her to Jack's ship. Also, a location-free, submissive dependent scene waits to fire if the player figures out how to use the magic of the mirror to transport herself.
H - Struggle	**Precond**: Same location as *Villain*; Jack cast as *Villain*. **Firing**: Immediate.	The player engages Black-Souled Jack in combat, using the magic mirror to reflect Jack's magic back at him.	[No alternates at this location.]
I - Victory	**Precond**: Location on Jack's ship at sea; follows **H**("Struggle with Jack"). **Firing**: Immediate.	Jack is slain, and, after a moment of consideration, his crew turns to the player as their new leader.	[No alternates at this location.]
K - Liquidation	**Precond**: Location on Jack's ship at sea; follows **A**("Captain is Kidnapped") **Firing**: Immediate.	The captain is released from the hold, and Jack's store of gold is plundered.	[No other **K** for this story's preceding **A.**].
↓ - Return	**Precond**: **α(**Orphan sailor hero). **Firing**: Immediate.	The player returns to the original ship.	[No alternates for this story history.]
W - Reward	**Precond**: **α**(Orphan sailor hero). **Firing**: Immediate.	Player is promoted to First Mate, and given a purse of gold, enough to buy a ship of her own some day!	[No alternates for this story history.]

Table3: An example story from the interactive drama *Wharf* built on the Eudaemon system.

We have also toyed with a few modifications and short-cuts to Eudaemon. One improvement is to make scenes more template-like. For example, a generic, location-free *Mediation* (**B**) scene could follow any *Villainy* (**A**), casting any available character as Dispatcher and simply reiterating a summary specified by the preceding **A** scene. Or a *Struggle* (**H**) scene could dynamically change to handle either different Villains or different Agent objects. We have also explored bundling groups of dependent scenes for the same function with the same prerequisites, as well as combining related functions (such as **D**, **E**, and **F**) into a single larger function. While these changes help ease the authorial load or simplify the system, they do not significantly change its basic operation.

A slightly more drastic change is to attempt to make scenes internally interactive. For example, right now the *Villainy* scene in *Table 3* is immediate—Jack attacks the ship and kidnaps the captain. This could be broken down into two parts—Jack attacks the ship, followed a number of dependent scenes that determine which actual villainy results. For instance, depending on what the player does during the battle, Jack might instead sink the ship (marooning the hero) or imprison the hero's soul. But this change is not a fundamental difference in the Eudaemon architecture. It can be achieved simply be dividing *Villainy*

(**A**) into *Villany-Intro* (an immediate scene) and *Villany-Result* (a bundle of dependent scenes).

Successes

The Eudaemon architecture does achieve some of our goals for an interactive drama system. It does indeed provide interactive drama, wherein the user takes on the role of a character and, through their actions within the story world, determines the outcome of the story. The drama manager exerts strong, top-down control of the story, yet it is only loosely coupled to the world. The plot is determined procedurally at run-time. That is, it is not a fully pre-authored branching tree.

Limitations

However, Eudaemon also suffers a number of limitations. First of all, we find Propp's functions too constraining for the kind of interactive drama we seek. They specify what *must* occur at each point of the story. Most of Propp's functions detail events that are inflicted upon the hero. As Propp points out when discussing the sphere of action of the different roles, the hero only exerts influence in **C↑** (when he agrees to set out on his quest) and **E** (when he responds to the donor).

In the few functions where a choice seems to be offered, it is generally only a binary choice. The hero character can comply with the *Interdiction* (**γ**), or violate it. He can respond positively (successfully) or negatively (unsuccessfully) to the Donor. Furthermore, these choices are actually illusory. According to the morphology, an *Interdiction* must be violated. The hero must succeed in his interactions with the Donor in order to receive the agent and advance the story. Fairclough's OPIATE suffers from this: once the system has generated a plot that fits the current world state, the player must agree to the binary choices presented to her. If she disagrees three times, the entire story is put on hold, and a new one is generated instead.

In an attempt to circumvent this limitation, the Eudaemon system uses bundles of multiple dependent scenes with the same preconditions for a single function. This means that, though the player cannot fail to react successfully to a function and advance the story, the details of *how* they react are still up to them. (See the alternate scenes for functions **C** and **E** in *Table 3* for examples of this.)

At the expense of story coherence, Eudaemon also allows some functions to time-out. For example, if an *Interdiction* (**γ**) is not *Violated* (**δ**) after a time, the drama manager pretends the *Interdiction* never happened and carries on with the story. However, each time this is done lessens the well-formedness of the story as experienced by the player. Events occur that are unnecessary to either earlier or later events.

A second limitation is that, due to using static scenes as the atoms of story construction, the player's interactions are generally not part of the content of the resulting tale. Specifically, the default behaviors of the world are never included as part of the drama manager's story model. They serve only to provide verisimilitude between scenes and to influence the selection of the next scene. Dependent scenes—such as used for functions **C** and **E—**are the only mechanism by which a user's actions can be explicitly made part of the system's story representation.

This brings us to the question of user agency. In the Eudaemon system, the player has a high degree of local agency (Mateas 2002). They can freely wander the world and interact with its objects and characters, seeing the immediate effects of their actions. However, while these local, world-level actions also affect what scene occurs next (a global, story-level result), the player usually has no awareness of why this should be so. Frequently, it would make no difference to the player's experience if the next scene was simply selected randomly. (See the alternate scene for function **D** in *Table 3* as an example of this.) Poor global agency is a third limitation of Eudaemon.

In considering these limitations, we've identified four degrees of story-level agency. In Eudaemon, the degree of user agency is usually dependent on the nature of the scenes that can be activated for a particular function. A *hoop-jumping* function has only a single dependent scene associated with it, meaning the user must determine the single action required to advance the story. In this case, we believe it is better to just make the scene immediate, but this can lead to long stretches of non-interactive story generation. The example story in *Table 3* demonstrates this: there is no further chance for user input after function **G**.

A *characterizing* function has a number of dependent scenes, but all result in the same next scene. For example, all the alternate scenes for function **E** in *Table 3* still result in the player receiving the magic mirror. The player's choice can customize the current function—whether they violently or gently deal with the old woman—but it does not otherwise impact the outcome of the story.

A *content-shifting* function also offers a choice through one or more dependent scenes, but the choice can lead to different following scenes. These following scenes will still fill the same following function, however. For example, how the user responds within the *Counteraction* (**C)** in *Table 3* could lead to different Donors in other locations besides the strange isle. But the user cannot avoid encountering a Donor after the *Counteraction*.

Finally, a *structure-shifting* function offers choices that can result in a different following function, such as **C** being followed by a **G** scene. Because Eudaemon follows Propp faithfully, it contains no such structure-shifting functions.

As a fourth limitation, the Eudaemon system has a high authorial load. The system only offers interaction capabilities to the degree that there are multiple possible scenes to choose from at any given point in the story. Given sixteen functions and a number of possible locations, this means the number of scenes required can be quite high. The burden is on the author to ensure the quality of the interaction. The majority of the scenes must be written so they can appear in multiple different stories, regardless of earlier events. The player must always be capable of activating scenes for the next function.

Finally, any system based on Propp does not scale well. Propp's functions describe very short, simple—though very well-formed—tales. Yet functions only work well for very specific genres, not for general fiction (Chatman 1978). Determining the functions for genres other than the Russian folktale, though seemingly simple at first, proves more difficult in practice. And generating longer tales with more functions increases the authorial load.

Conclusion

Due to its perceived limitations, we chose not to implement our Eudaemon design at this time. Furthermore, we believe these limitations are inherent to a Propp-based approach to interactive drama. Essentially, Propp describes what *must* occur within a function. Such a system can still allow for an interesting form of interactive drama—within the tight bounds of the Russian folktale structure, the player can influence what kinds of villains and donors they meet, how they react to them, and where. But the player must still successfully receive the magical agents and liquidate the original misfortunes, as the functions require. They are limited to the characterizing and content-shifting degrees of agency and cannot explore other *kinds* of stories. The great strength of interactive drama should be the ability to explore the ramifications of one's choices as a player. But, in a Propp-based system, players find they are always successful heroes in the end. And any attempts at broadening the range of interactivity in a Eudaemon-like system only proportionately increases the authorial load.

Instead of attempting to loosen or expand Propp's functions to gain greater user agency, we have decided to take Crawford's (2004) advice: rather than starting with story and trying to add interaction, start with interaction and structure it into a story. Though still aiming to build stories piece-by-piece at run-time, we are now exploring the technique of table-top role-playing games of building adventures from interactive encounters. We are hoping to dynamically combine these interactive atoms based on the more general story structure and rules of causal necessity described by Aristotle and Gustav Freytag.

References

Chatman, Seymour. *Story and Discourse: Narrative Structure in Fiction and Film*. Ithaca, NY: Cornell University Press, 1978.

Crawford, Chris. *Chris Crawford on Interactive Storytelling*. Berkeley, CA: New Riders, 2004.

Fairclough, Chris R. "Story Games and the OPIATE System: Using Case-Based Planning for Structuring Plots with an Expert Story Director Agent and Enacting them in a Socially Simulated Game World." PhD thesis. University of Dublin, Trinity College, 2005. <http://www.cs.tcd.ie/publications/tech-reports/reports.05/TCD-CS-2005-59.pdf>

Grasbon, Dieter, and Norbert Braun. "A Morpholgical Approach to Interactive Storytelling." Proceedings of *cast01//Living in Mixed Realities*, 2001. <http://netzspannung.org/version1/extensions/cast01-proceedings/pdf/by_name/Grasbon.pdf >

Mateas, Michael. "Interactive Drama, Art, and Artificial Intelligence." PhD Thesis. Technical Report CMU-CS-02-206. Carnegie Mellon University, Pittsburgh, PA, 2002.<http://www.cs.cmu.edu/afs/cs.cmu.edu/misc/mosaic/common/omega/Web/Groups/oz/papers/CMU-CS-02-206.pdf>

Prada, Rui, Isabel Machado, and Ana Paiva, "TEATRIX: Virtual Environment for Story Creation." Proceedings of *5th International Conference on Intelligent Tutoring Systems*, 2000. <http://gaips.inesc-id.pt/teatrix/papers/teatrix-its2000.pdf>

Peinado, F., M. Ancochea, and P. Gervás. "Automated Control of Interactions in Virtual Spaces: A Useful Task for Exploratory Creativity". Proceedings of *7th European Conference on Case Based Reasoning*. CERSA, 2004. 191-202. <http://www.fdi.ucm.es/profesor/fpeinado/publications/2004-peinado-automated.pdf>

Peinado, F., and P. Gervás. "Transferring Game Mastering Laws to Interactive Digital Storytelling". Proceedings of *2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE'04*. Lecture Notes in Computer Science, 3105. Eds. S. Gobel, et al. Springer, 2004. 48-54. <http://www.fdi.ucm.es/profesor/fpeinado/publications/2004-peinado-transferring.pdf>

Propp, Vladimir. *Morphology of the Folktale.* Translated: Laurance Scott. Ed. Louis A. Wagner. 2nd ed. Austin, TX: University of Texas Press, 1968.

Spierling, Ulrike, Dieter Grasbon, Norbert Braun, and Ido Iurgel. "Setting the Scene: Playing the Digital Director in Interactive Storytelling and Creation." Computer & Graphics 26 (2002): 31–44.