Dynamic quest generation in Micro Missions

Yana Malysheva
RIT
Rochester, NY
yana@yanamal.com

***Abstract*—Most video game stories rely on the story designer anticipating and authoring every choice a player can make with respect to the story, and the consequences of these choices. With this method, a player's sense of agency with respect to the story depends on whether that player's desired story-related actions were anticipated by the authors at the time of the development of the game. Some video games instead create an environment for emergent stories by defining detailed simulations of characters and the world they inhabit. But there are not a lot of video games that actively attempt to direct story while making it dynamic and responsive to player actions.**

Micro Missions is a mini-game/adventure game with a quest generation system that seeks to increase player agency by reacting to player actions dynamically, at runtime. This system uses a taxonomy of quest archetypes to select and populate quests on demand based on the current world state and current story-related goals. The creation and analysis of the quest generation system is facilitated by the architecture of Micro Missions: all gameplay is separated into modular interactions with discrete outputs, which are organized into strict cause-and-effect trees(or more precisely, directed acyclical graphs).

***Index Terms*—Interactive storytelling, Procedural generation, Quests**

I. Motivation

There are two major motivators that informed the design of this system: reducing discrepancy between story world and game world, and enabling runtime flexibility and adaptiveness to player actions. There are also other important motivations for developing a dynamic storytelling system, such as increasing replayability and player agency, but they were not as helpful in guiding the development of this particular system.

Both of these important motivations imply a need for providing computer-readable metadata about story and story elements. In order to be able to adapt to player actions in a way that's consistent with the game world, the quest generator must somehow "understand" how story elements fit together and with the world. We need to encode narrative elements in a computer-readable way.

II. Scope

The quest generator's goal is to decide, at runtime, what the next quest given to the player should be, based on the current state of the game and story. Creating the story setup itself - who is the bad guy, what is the conflict, what is the resolution - is not within the scope of the quest generator itself.

Using the analogy of a pen-and-paper RPG session, the quest generator is the Game Master (or even just part of the Game Master), not the module writer. Although the "module writer" could also be procedural, this module writer would be a separate part of the storytelling system. In the current implementation, describing the story in a way that is usable by the quest generator is the burden of the human story designer.

III. Design and Encoding

The Quest Archetypes encoding is based on the Quest as a basic unit of storytelling. The story is told to the player through a sequence of quests, each of which is generated out of a quest archetype. Each archetype is an abstraction of a familiar type of quest present in most quest-based games, like a fetch quest or a "prove your worth" quest.

These archetypes describe how to implement a specific quest (who says what and in what order, what does the player need to do and to whom), while abstracting out all specifics - who, what, where - into variables. Variables can also be used to introduce variations into a particular instance of a quest, so that not all quests derived from the same archetype play the same. Given an archetype and a set of variables to fill in, the generator can create a particular instance of the quest.

This part of the quest archetype system - using variables to determine some specifics of a quest at runtime - is similar to what games like Skyrim[1] and its Radiant Story system do. But just having generic quests with lots of variables would not allow the quest generator to decide, intelligently, how to choose and populate these quests in order to progress the story. For this purpose, the quest archetypes also store important metadata that defines the purpose and context of a particular archetype, what it's good for and what it might cause to happen.

This metadata is described in terms of states, which represent both the prerequisites for a quest of a particular type and the intended results of completing such a quest. For example, the prerequisite to a "rescue a princess" type of quest might be that there is a princess in trouble, and the intended results would be to make the princess in question safe again, and perhaps to gain her favor.

Note that the intention is from the point of view of the player character, or in some cases another character. It is not necessarily the storyteller's intention. It is also not necessarily the same as the actual outcome of the quest: perhaps in trying to rescue the princess, the hero will be too late, or accidentally

978-1-4673-1358-2/12/$31.00 ©2012 IEEE

kill her himself. Twists and reversals that differ from the “intended” outcome can and should be introduced intentionally by the system.

States are not limited to describing the state of the story and world as the player perceives it. They can also describe hidden storytelling meta-goals, goals of other characters in the story, and in general any effect a quest might have on the world and story.

States, like archetypes, have variables. As part of the archetype definition, state variables are mapped to the archetype variables: if the quest is to rescue Princess X, then Princess X (not Princess Y) must first be in need of rescue.

These states describe most of the metadata needed to pick and populate archetypes. Any implementation of the quest archetype encoding will use this information to choose an archetype and populate it with actors based on desired storytelling goals.

IV. Implementation: Micro Missions

Micro Missions is an abstract action-adventure game where every single thing you do is a mini- or micro-game. These mini-games are connected through shared items and characters. Story is conveyed through short cut-scenes. Characters usually speak in symbols, not words. This abstract, symbolic framework lends itself well to a story generation system.

The story in Micro Missions is always the same. You are a young adventurer who has been asked by his mother to pick up some Milk and Bread while you’re out adventuring. Your goal in the game becomes to track down these groceries by doing various favors for other inhabitants of the world in return for information on the location of the items you seek. Upon completion of the favor, the inhabitant will either give the requested item to the player character or direct him to someone else.

At the beginning of the adventure, the player must ask the wise bartender for advice on who can help him get milk and bread. From the bartender, he might get a tip that the Bandit knows something about milk. If the player then goes and talks to him, the Bandit might ask the player to spy on his rival, Robin Hood, in exchange for information about the milk. Once the spying quest is completed, the Bandit will either give the milk to the player, or refer him to another character in the game, who would in turn ask the player for a favor in exchange for further information.

Although the basic premise and structure of the story are the same every time, each playthrough can be quite different, depending on who the player runs into and what that character needs help with.

In this implementation, most of the states that drive the quest selection are the goals and needs of these non-player characters. These goals and needs are based on relationships between the non-player characters. If you are someone’s rival, your goal might be to spy on them. If you are someone’s boss, you may have a goal to make your minion do some work for you. The quest generator picks a quest archetype that satisfies some of these goals, and populates it with the characters and items that would complete that specific goal to create the next quest for the player. When the quest is don, the quest-giver decides randomly whether to give the food item to the player or to direct him to a new character, chosen randomly from the characters with whom the original quest-giver has a positive relationship.

V. Evaluation

The Micro Missions quest system succeeds in generating understandable, fun-to-play quests. Players did not have trouble figuring out the general quest structure and the story progression. This is a relatively simple implementation, but it does, in large part, succeed in the original goals of creating a dynamic quest system.

It does create a story that is more integrated with the structure of the world than normal RPG stories, since the individual pieces of the story are rooted in the way the world functions. If a character were no longer alive, a quest would never be generated where you need to talk to him. If two characters became enemies, there would no longer be any quests where they help each other.

The story definitely has runtime flexibility, since all of the details are generated at runtime. The responsiveness to player actions is somewhat limited, but it is there. The player can choose to ignore a particular character and still complete the main story by simply asking the bartender for more leads. The player’s actions - who he chooses to ignore or help - actually also affect the relationships between the characters, strengthening them when assisted and weakening over time when ignored.

But there are still some problems with the implementation. Although the abstract nature of Micro Missions made the quest system easier to implement, it also made the game too confusing for many players. The game as a whole is very fast-paced, and suffers from a lack of immediate feedback, so it is often hard for the player to understand what’s happening and why. This lack of polish distracted a lot of players from the quest system, and made it hard to follow or care about the story. Only a few players actually finished the story without getting stuck or distracted elsewhere in the game, and most players didn’t seem to care a whole lot about why or how a certain quest was given to them. It was also hard for players to understand how they were affecting the world.

The quest generation system itself is rather small-scale and local. The quest generator can only make local, greedy decisions based on the current state and the possible next states. It does not look ahead or take into account any kind of bigger picture. This obviously limits the kind of story it can be used to generate, and the story design (just keep asking different people until you get your groceries) reflects this.

VI. Future Implementations

The ability to plan ahead and take a bigger picture view of the story into account is the next thing to accomplish. Luckily, the quests and states encoding lends itself very well to AI planning algorithms. The major building blocks in planning are states, such as "NPC X does not have item Y", and actions, such as "get item Y" and "give item Y to NPC X". Actions have prerequisites and outcomes. If we're using first-order logic, actions and states are generic, and take variables such as "NPC X" and "item Y". A planner's job is to create a sequence of actions, given an initial state and a goal state.

A quest planning agent would be able to come up with a series of quests for the player to go through in order to complete a story, given a description of the initial conflict and a way to resolve it. Classical planners come up with a static plan before execution starts. Online planners are more complex but can react dynamically to unexpected situations. I would like to make a continuous planner, which monitors the changing world state continuously and decides on the next action without necessarily figuring out the whole plan first. This is representative of game stories: the player characters do not, and often cannot, come up with an exact plan to save the world before they start acting.

The burden of the designer in such a system would be coming up with the initial state, goal state, possible actions(quest archetypes) and state descriptions. The initial state and goal state define the basic storyline, and would include important things like the real nature of the conflict, relevant history of the world, what is hidden or apparent to the player at the beginning of the game, and how this conflict can be resolved. The set of quest archetypes will define the space of possibilities for what can happen during the story. For example, if there are no archetypes that can introduce twists or misdirections, the story will be quite straightforward and boring. If there are no variations how a particular archetype can be implemented in the game world, the questing can feel stale and repetitive.

The set of states, and how well the archetypes use these states to describe what they do, will limit how much information the generator has to make its decision about what to do next.

So, another future project is coming up with a good general set of archetypes and storytelling-related meta-states. I want to analyze existing video game stories from the point of view of, what kind of quests, archetypes, states would generate such a story.

Analyzing video game stories, as well as stories from other media, can also provide insight into techniques and structures that make for interesting, engaging stories. I would also like to incorporate elements of existing analyses of storytelling, such as the Hero's Journey and Propp's Morphology of the Folk Tale.

Acknowledgments

I'd like to thank my capstone advisor, Prof. David Schwartz, for helping formulate and iterate upon the capstone research topic which originally led to this system. I'd also like to thank all the game lab regulars, my family, and anyone else who's ever listened to me rant about quests generators and provided valuable feedback on both the content and the way I presented it.

References

[1] Skyrim. Bethesda Game Studios. Bethesda Softworks. 2011.

[2] Ryan, Marie-Laure. Peeling the onion: layers of interactivity in digital narrative text. http://users.frii.com/mlryan/onion.htm. 2005

[3] Mateas, Matthew. Narrative intelligence: an introduction to the NI Symposium. In M. Mateas and P. Sengers (Eds.), Working notes of the Narrative Intelligence Symposium, AAAI Fall Symposium Series. Menlo Park: Calif.: AAAI Press, 1999.

[4] Mateas, Matthew. Narrative intelligence. John Benjamins Publishing Co. Amsterdam, The Netherlands, 2002.

[5] Howard, Jeff. Quests: design, theory, and history in games and narrative. A.K. Peters, LTD, 2008.

[6] Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28 – 1 December, 2011. Proceedings.

978-1-4673-1358-2/12/$31.00 ©2012 IEEE