Story Curve Driven Mission Generator

Wang JiaoJian WANGJIAOJIAN@UNI-BONN.DE
Olana Missura OLANA.MISSURA@UNI-BONN.DE
University of Bonn, Germany

Abstract

We present a novel approach to automatic interactive mission generation for computer games that emphasize the story aspect of a mission and aims to ensure coherence and interestingness of the story. The development of the stories in the generated missions are influenced by both the desired story curve and the inputs from the player, utilizing her preferences. We are still implementing our approach into a usable mission generator, such that it can be evaluated by game players.

1. Introduction

Procedural content generation is a very popular technique in game industry. It has been used to generate terrains, vegetation, dungeons, etc (Hendrikx et al., 2013). A lot of research on procedural content generation involve computational creativity and player preferences. One example is generating missions for computer games, with existing approaches based on level design (Dormans, 2010) or player modeling (Zook et al., 2012). We aim to create an automatic interactive mission generator for computer games which focus on the story aspect of the mission. Its primary function is to generate a story line that the player's quests are based upon.

In computer games, players become active participants instead of passive audiences of the stories and are capable of influencing the development of the stories through their actions. Therefore, an interactive mission generator has to take player actions into consideration. Generating stories for game missions is closely related to computational storytelling or, more precisely, interactive storytelling (Sgouros, 1999; Yu & Riedl, 2012). It is crucial that the automatically generated stories in missions are interesting to players. In order to generate interesting stories, the mission generator needs to learn the preferences of each player. Then the player preferences can be used to guide the story development of the generated missions. There are also different approaches on making the generated story interesting, e.g. sequential recommendation (Yu & Riedl, 2012) and player preference modeling (Sharma et al., 2007).

Here we propose a new approach to learn the player preference based on the concept of story curve. This concept is independent of actual events in the story and, therefore, flexible enough to accommodate different story settings. Our approach models the story curve from story examples that a player finds interesting, and use that model to manage the drama structure while generating new stories. It iteratively presents a player with a mission and receives an explicit feedback. The feedback will be used to improve the story curve model. We are currently implementing our approach into a usable tool for computer games. It will be evaluated by game players once it is finished.

2. Story curve driven approach

The approach we proposed uses a multi-agent system and automated planning(Riedl & Young, 2005; Brenner, 2010) to generate events for a new story, and manages the drama structure of the new story based on the concept of story curve. The multi-agent system consists of a world environment and multiple intelligent agents. Agents can interact with each other and the world. Each agent possesses its own agenda, as well as an access to a plan necessary to pursue it. Plans are often made by automated planning algorithms which include causal links, such as partial order planning (Weld, 1994) and Graphplan (Blum & Furst, 1995). A new story is generated as a sequence of actions performed by these agents as they pursuit their agenda. The temporal and causal coherence of the generated story is guaranteed by actions generated by automated planning. The development of the story is adjustable through drama management, which manipulates the outcome of actions or forces a re-plan for alternative actions. Drama management in our approach is guided by story curves, which in turn are learned from stories that a player finds interesting.Then we generate new stories following the learned story curve model. As a result the generated stories will possess the

Proceedings of the Constructive Machine Learning workshop @ ICML 2015.

```
(define (:domain fantasy-story)
  (:objects warrior dragon village cave)
  (:predicates
    (Place ?x) (Actor ?x) (At ?x ?y))
  (:action Move :parameters (?a ?s ?d)
    :precondition (and
      (Actor ?a) (At ?a ?s) (Place ?d))
    :effect (and
    (not (At ?a ?s)) (At ?a ?d)))
  (:init
    (Actor warrior) (Actor dragon)
    (Place village) (Place cave)
    (At warrior village) (At dragon cave)))
```

Figure 1. A plan domain for a fantasy story setting.

similar dramatic structure as the training examples. There have been studies in generating interesting stories by reproducing certain narrative structure patterns (Prez y Prez & Sharples, 2001). Hence, we hypothesize that the generated stories with desired story curve will also be interesting to the player. All components of our approach are described in details in the following sections.

2.1. Multi-agent planning

A story can be considered as a sequence of connected events in a story world, where each event consists of an action and its outcome. Multi-agent planning (Riedl & Young, 2005; Brenner, 2010) is a popular method to generate such event sequences. It is a combination of a multi-agent system and automated planning. The world environment and multiple intelligent agents in a multi-agent system correspond to the world and characters in a story. The agents use automated planning algorithm to decide which actions they are going to take. A story setting includes a world where the story occurs and everything in the story, e.g. locations, items, individuals and possible actions. In multi-agent planning system, a story setting is defined as a planning domain using Planning Domain Definition Language (Mcdermott, 1997) or other similar planning domain languages, so that it can be understood by automated planning algorithms. An example of a planning domain for a fantasy story setting is presented in Fig. 1.

At the beginning of the mission story generation, each intelligent agent is assigned a goal; different goals result in different story lines. At each time step, each agent checks if there is a valid plan to achieve its goal. If there is, the agent takes one possible action from the plan. The story curve is used here to decide how agents choose their actions, as well as the outcomes of the actions. The action and the outcome are then recorded as an event for the generated story. If there is no valid plan, the agent uses an automated planning algorithm to make a new plan. A planning task $\mathcal{P} = \{\mathcal{A}, \mathcal{I}, \mathcal{G}\}$ consists a set of action schemata $\mathcal{A}$, a initial state $\mathcal{I}$ and a goal state $\mathcal{G}$. A plan a_1^n is a sequence of n actions instantiated from schemata $\mathcal{A}$. The task is to find a plan a_1^n which starts from the initial state $\mathcal{I}$ and ends in the goal state $\mathcal{G}$. The generating procedure repeats at each time step until the mission ending criteria are met. A mission is considered over when either the player has achieved his goal, thus mission accomplished, or there is no valid plan available for the player to reach his goal state, thus mission failed.

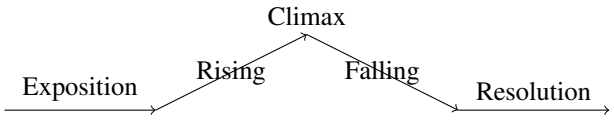

Figure 2. "Freytag's pyramid", a drama structure example.

2.2. Story curve

Story curve is the dramatic structure of a story line. It outlines the development of drama over time inside a story. The earliest example is the dramatic theory in Poetics (Aristotle, 2008), a work from the Greek philosopher Aristotle. Another example of this concept is Gustav Freytag's analysis of ancient Greek and Shakespearean drama (Freytag, 2011). He proposed a dramatic structure that consists of five parts: exposition, rising, climax, falling and resolution. An example of his theory of dramatic structure is illustrated in Fig. 2.

2.3. Story curve model

Given a story, its events and world states at every time step are observable information, while story curve is hidden and unobservable. In order to support interactive story telling, the player actions also have to be taken into account. Our task here is to model the relationship between the hidden and the observable information. According to the concept of a story curve, the unobservable story curve $s \in \mathcal{S}$ influences the observations and the next story curve. Intuitively, the player's actions would influence both hidden states and observations. We use the story states $y \in \mathcal{Y}$ as observations, which abstract the observable information in the story. The player states $x \in \mathcal{X}$ represent the actions made by the player and are used as input. We model their relationship as an Input-Output Hidden Markov Model (Bengio & Frasconi, 1995a;b), the model structure is illustrated in Fig. 3. The state space description of our model is defined as:

$$\begin{aligned} s_t &= f(s_{t-1}, x_t)\,, \\ y_t &= g(s_t, x_t)\,, \end{aligned} \tag{1}$$

where f is the state transition function and g is the output function.

Clearly, our story curve model is a Mealy machine (Mealy, 1955). Thus, the story development is influenced by both

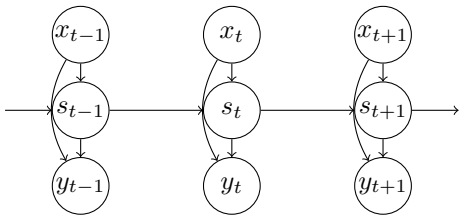

Figure 3. Input-output HMM: x are player states (input), s are story curve states (hidden states) and y are proposed story states (output).

the desired story curve and the player inputs. It is important to note that the story states generated here are only proposed ones. This is because the actual actions are generated by automated planning algorithm and therefore are subject to temporal and causal constraints. Once we have the proposed story state, we approximate it through manipulating action outcomes and re-planning, which may result in the generated story state differing from the proposed one.

2.4. Story and player states

The story curve itself is independent of story settings, but the story and the player actions are not. The observable information in a story includes the world states and all actions from arbitrary number of agents/characters. All of them are dependent on the story settings. The case of player actions is even more complicated. They are dependent on the settings of both the story and the game. For our story curve model to be also independent, we need to abstract the observable information in a story into story states, and the player actions into player states.

We achieve this abstraction by focusing on the results of the player actions with respect to the protagonist's progress with regard to her ultimate objective. Although certain types of story do not involve an ultimate goal for the protagonist or doesn't even have a protagonist at all, most stories do. It is especially true in case of computer game missions, most of which are objective oriented. Because of that, the protagonist's progress can be used as a measurement of the development of the story in game missions. At each time step t, we want to measure the distance between the current world state and the ultimate objective state. This measurement is approximated by the amount of the actions required for the player to reach the goal. Planning algorithms such as Graphplan can be used here to obtain the list of required actions.

2.5. Training

In order to learn the preferences of a player, we need to train our story curve model so that it could represent the desired story curve more accurately. There are two network mappings in our work: $\mathcal{N}_{ij}(t, \theta_j)$ and $\mathcal{O}_i(t, \vartheta_i)$, where θ_i and ϑ_i are vectors of adjustable parameters/weights. The story curve network mapping $\mathcal{N}_{ij}(t, \theta_j) = \mathrm{P}(s_t = i | s_{t-1} = j, x_t)$ calculates the distribution of the current story curve state given last story curve state and current player state. The story network mapping $\mathcal{O}_i(t, \vartheta_i) = \mathrm{P}(y_t | s_t = i, x_t)$ predicts the current story state distribution given current story curve state and player state. The goal of training task is to find the maximum likelihood estimates of parameters (Bengio & Frasconi, 1995a;b). The training data $\mathcal{D}$ is a set of stories that a player finds interesting. It can be represented as a set of P pairs of player/story state sequences of length T_p:

$$\mathcal{D} = \{x_1^{T_p}(p), y_1^{T_p}(p); p = 1 \dots P\} \tag{2}$$

$\mathcal{D}$ is referred to as the incomplete data in training due to the story curve being unavailable. Let Θ denote the vector of parameter collection of all θ_i and ϑ_i. Then the likelihood function is given by:

$$\mathrm{L}(\Theta; \mathcal{D}) = \prod_{p=1}^{P} \mathrm{P}(y_1^{T_p}(p) | x_1^{T_p}(p); \Theta) \tag{3}$$

As stated in the original Input-Output Hidden Markov Model paper, expectation-maximization (Dempster et al., 1977) algorithm can be used to train Input-Output HMM. Expectation-maximization is an iterative approach to maximum likelihood estimation. Each iteration consists two steps: an expectation step and a maximization step. At each iteration $k = 1, 2, \dots$. The expectation step computes the expected value of the log-likelihood function $\log \mathrm{L}(\Theta; \mathcal{D}_c)$ with respect to the current estimate of parameters Θ^k. Here, the $\mathcal{D}_c = \mathcal{D} \cup s_1^{T_p}(p)$ is the data set including story curve, therefore it is referred to as complete data. The current estimate of parameters Θ^k is computed in the maximization step from the previous iteration $t - 1$. This expectation of log-likelihood function is defined as:

$$Q(\Theta; \Theta^k) = E_{\mathcal{S}}[\log \mathrm{L}(\Theta; \mathcal{D}_c) | \mathcal{D}, \Theta^k] \tag{4}$$

The maximization step updates the current estimate of parameters Θ^k with the one that maximize the expected value of likelihood function from the expectation step. The updated estimate of parameters Θ^{k+1} will be used in the expectation step in the next iteration $k + 1$. This update of parameters-estimation is defined as:

$$\Theta^{k+1} = \arg\max_{\Theta} Q(\Theta; \Theta^k) \tag{5}$$

Expectation-maximization algorithm iterates until a local maximum of the likelihood function is found. A variety of heuristics can be used to help escaping the local maximum, e.g. random start.

References

Aristotle. *Poetics*. Book Jungle, 2008.

Bengio, Yoshua and Frasconi, Paolo. An input output HMM architecture. In *NIPS'95*, pp. 427–434. MIT Press, 1995a.

Bengio, Yoshua and Frasconi, Paolo. Input/output HMMs for sequence processing. *IEEE Transactions on Neural Networks*, 7:1231–1249, 1995b.

Blum, Avrim L. and Furst, Merrick L. Fast planning through planning graph analysis. *ARTIFICIAL INTELLIGENCE*, 90(1):1636–1642, 1995.

Brenner, Michael. Dynamic plot generation by continual multiagent planning. In *AAMAS'10*, pp. 1529–1530. International Foundation for AAMAS, 2010. ISBN 978-0-9826571-1-9.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

Dormans, Joris. Adventures in level design: Generating missions and spaces for action adventure games. In *PCGames'10*, pp. 1:1–1:8, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0023-0. doi: 10.1145/1814256.1814257.

Freytag, Gustav. *Die Technik Des Dramas*. Nabu Press, 2011.

Hendrikx, Mark, Meijer, Sebastiaan, Van Der Velden, Joeri, and Iosup, Alexandru. Procedural content generation for games: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(1):1:1–1:22, February 2013. ISSN 1551-6857. doi: 10.1145/2422956.2422957.

Mcdermott, D. Pddl - the planning domain definition language, 1997.

Mealy, George H. A method for synthesizing sequential circuits. *Bell System Technical Journal, The*, 34 (5):1045–1079, Sept 1955. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1955.tb03788.x.

Prez y Prez, R. and Sharples, M. Mexica: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13(2):119–139, 2001.

Riedl, Mark O. and Young, R. Michael. An objective character believability evaluation procedure for multiagent story generation systems. In *IVA'05*, pp. 278–291. Springer, 2005.

Sgouros, Nikitas M. Dynamic generation, management and resolution of interactive plots. *Artif. Intell.*, 107 (1):29–62, January 1999. ISSN 0004-3702. doi: 10.1016/S0004-3702(98)00106-4.

Sharma, Manu, Ontañón, Santiago, Strong, Christina, Mehta, Manish, and Ram, Ashwin. Towards player preference modeling for drama management in interactive stories. In *FLAIRS'07*. AAAI Press, 2007.

Weld, Daniel S. An introduction to least commitment planning. *AI Magazine*, 1994.

Yu, Hong and Riedl, Mark O. A sequential recommendation approach for interactive personalized story generation. In *AAMAS'12*, 2012.

Zook, Alexander, Lee-Urban, Stephen, Drinkwater, Michael R., and Riedl, Mark O. Skill-based mission generation: A data-driven temporal player modeling approach. In *PCGames'12*, pp. 6:1–6:8, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1447-3. doi: 10.1145/2538528.2538534.