

Creating Characters for Dynamic Stories in Interactive Games

David Shaw, Nick Barnes and Alan Blair

Computer Science and Software Engineering Department

University of Melbourne

2nd October 2001

Abstract

The possession of strong characters and plot is an important element in computer games. In this paper, we discuss how dynamic stories cannot be satisfied using techniques adapted from standard entertainment media such as film. We propose that semi-autonomous characters are a requirement for creating dynamic non-linear stories, and propose a framework for building such characters and how they can show personality.

1 Introduction

Stories have been a popular form of entertainment for millenia. People have been drawn to a well told story, from the words of Homer's *Iliad* through to the modern computer generated films of today. It is therefore natural for stories to be a vital component in computer games, one of the newest media that has the potential for providing drama. While there will always be a market for 'twitch' action games (such as *Quake*) and puzzle games (like *Tetris*) for which stories are not required, most games benefit from having a well crafted plot. At the most basic level, adding a story gives a sense of purpose to the action in the game, linking otherwise unrelated tasks together to form a meaningful whole (this can be applied to practically any games that is 'level-based', such as most RTS¹ and FPS² games.) In some types of games

¹RTS: Real Time Strategy

²FPS: First Person Shooter

(particularly adventure and RPGs³) advancing the story is a vital component of its appeal. For many games, the possession of a good plot is (or at least should be) a high priority.

Many factors contribute to the making of an enthralling story, but an essential component is the depth of its characters. A story can be thought to consist purely as the interaction of the characters between each other and their environment. For classic pieces of drama (such as the plays of Shakespeare), observe the complexities of the characters; most have a strong sense of believability that audiences can relate to, and have multi-dimensional personalities that are plausible and yet deeply intriguing. For the construction of any story-based game, having interesting, immersive and compelling characters is a critical element.

In this paper, we argue that although it is feasible to create good drama via traditional methods (using the linear style of existing media), computers have the potential to create dynamic, non-linear stories that provide a deeper sense of immersion within the game world. We further propose that semi-autonomous characters are a necessary component in creating dramatically compelling non-linear story lines, and show the preliminary stages required in achieving such characters (taken from a system currently under development).

2 The need for an alternative to linear storytelling

In order to craft a story for a game, the most common and intuitive approach taken is to apply the ideas and techniques developed for the established traditional forms of story-telling. For text based games, methods applicable for novels are easily applied, but with the audio-visual mastery of computers today, the silver screen is the standard source for inspiration. By using film as its guide, the story for a game can be written essentially the same way as a script for a screenplay.

There are a number of advantages to applying the existing techniques of film to games. One is that as the same creation methods are being applied, the same type and level of drama as that which graces the traditional media can be achieved (provided the talent of those involved in creating the story is equivalent). Another is that traditional scripts are reasonably easy to write, edit, understand and apply, and people are comfortable working with them.

However, there is a major disadvantage to using film as the approach for

³RPG: Role Playing Game

scripting stories in a game. Film is purely linear, their narratives will never change. This is necessary because it is a passive medium, to experience the story all that is done is to watch from beginning to end. The viewer is put in the role of an independent observer, detached from the events shown on the screen. The same can be said about novels and traditional theatre such as Shakespeare. No matter how deeply moving a version of *Hamlet* is, it is not the viewer but rather another person (Hamlet) who lives out the story. The power of the story comes from the rich description of the narrative and the characters.

Computer games have the same audio-visual abilities of film, but with a crucial additional feature - interactivity. Unlike film, input from the player is used to affect the output of the game. The player can tell their character when and where to move, and which way to jump. This level of interaction greatly adds to the immersive effect of the medium, where the player projects themselves into the character which they control. The player is no longer witnessing the exploits of a character in some game world, to some extent it is *them* who is having the adventure. The character may be considerably less developed than Hamlet (as are practically all game characters), but the immersive nature of the game is such that the empathy can be much higher (as any misfortune that occurs to the character directly affects the player). This immersion with the game world is a powerful effect that can immensely add to the appeal of a game.

Herein lies the problem involved with applying a story to a game. In film, the director has complete control over the path of the story and the actions of the characters. In games, some control must be transferred to the player (otherwise there would be no game). Most games give the player a fair amount of control of their character's actions for the 'gameplay' sections of the game⁴, but very little to no control over the direction of the story⁵, in an attempt to strike a balance between story and gameplay.

While a viable technique for adding plot to games, this approach sacrifices immersion for story. The level of immersion in a game is proportional to the amount of control given to the player. For the player to be truly immersed in the game world, they need to be given the option to choose their actions (and for them to genuinely affect the story). They need the freedom to make their own path and roles. This cannot be done with the linear film script techniques, as this 'free' approach is inherently non-linear. This means that a dynamic non-linear story cannot rely on using a traditional pre-written

⁴For example, the 'battles' in an RPG, or the 'missions' in an RTS.

⁵Story elements are usually advanced in 'cutscenes' (often pre-rendered CGI movies or video clips), where the player has no control over what the characters do.

script to ensure compelling drama. New techniques need to be developed.

If games are to realise their potential as a non-linear dynamic story-telling medium, then the following need to be realised:

1. *Interactivity*: The ability for a player to take a wide variety of actions that can influence the outcome of the story.
2. *Well-defined characters*: Computer controlled characters that can act with a definable personality in a diverse range of situations and to a wide variety of interactions with other characters (both computer and human controlled).
3. *Dramatic plot generation*: A way to combine the motives and actions of the player and the computer controlled characters to create a dramatic story.

The remainder of this paper examines what is required to achieve the second point; how to develop well-defined characters in a non-linear story.

3 Providing freedom for characters

As the plot of a non-linear story can be influenced by the actions of a human player (whose decisions cannot be fully anticipated at design), the characters in this story need to be able to adapt to changing situations in their world. It would seem desirable to have characters that act autonomously, but full autonomy (in the strict sense of the word) would imply each character's AI be a contained unit, with the input based on a model of the characters 'senses' (i.e. sight, hearing, touch, smell and taste). While this method is required for truly independent life-like characters, there are some problems with this approach:

1. As the game world is purely computer modelled, there is no need to manufacture an accurate model of reality. The world can be structured to speed up computation time by giving more direct access of the data to the characters.
2. To accurately process every character's cognitive processes as a self-contained unit would add an extra layer of complexity that is not required when creating a game.
3. It might be desirable when creating dynamic plot-lines to manipulate the actions of characters in order to create interesting situations (i.e.

force the characters to entertain the player if the pace of the story is languishing). This desire for compelling drama breaks the ideals of autonomy, which is acceptable as the story is a higher priority than the purity of the implementation.

The possession of full autonomy is not a prerequisite for a realistic character in games. What is needed is a ‘semi-autonomous’ character, one which appears autonomous to the game player, but is not truly autonomous in implementation.

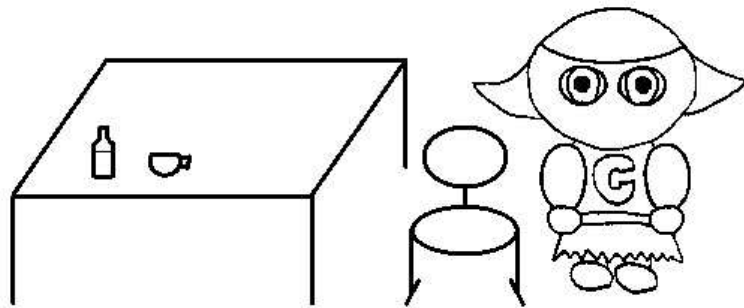


Figure 1: A simple example world; a table, a chair, a bottle of water, a cup and Claire (a computer controlled character).

To explain a method of how this could be done, consider the simple game world shown in figure 1. This world consists of nothing more than a room containing four items (a table, a chair, a bottle of water and a cup), and a computer controlled character (called Claire).

In order for Claire to interact with the items in the room, they would need to be represented as separate entities. In an object-oriented code design, it is logical to represent each item as an object containing information about the attributes of that item. These attributes include simple quantitative terms such as mass and size dimensions, but should extend to links to the actions that could be taken with the object. For example a character can drink from the cup or the bottle, can sit on the chair or the table, and can place smaller objects on top of the table. As there are practically infinite actions that a person can perform with an object (such as smashing the bottle or wearing the cup as a hat), some limit on which to implement would have to be decided

upon.⁶

As Claire is a character, she should also be represented as an object containing the attributes that describe her. As well as physical attributes, a representation of her personality and her current moods and drives would also be required. These personality attributes would be used in an internal decision making algorithm to choose between alternative courses of action, and could be implemented as fractional values to be used as weights in a fuzzy control system. The moods and drives are the more variable attributes such as ‘hunger’, ‘thirst’, ‘fear’, ‘boredom’ or ‘loneliness’. These would also be best represented as fractions, and their values would be modified as the character chooses actions (e.g. hunger drops if the character eats). The choice of personality traits and drives would depend on the nature of the game.

In order to coordinate the actions of characters such as Claire, it would be convenient to have a high-level controller to handle some of the decision making process. This ‘world’ controller in this scenario would be a class containing a reference to all the objects in the game world (i.e. the four items and Claire). To see how this world controller can help the process, here is a high-level description of how Claire can decide a simple course of action. For this example, assume that Claire is ‘thirsty’, i.e. her thirst drive is high (and no other drives are competing for her attention).

- The world controller ‘asks’ Claire what decision she wants to take.
- Claire checks her current mood, drives and her personality attributes. As her thirst is the strongest drive at the moment, she decides to choose an action that would reduce her thirst. Claire does not know what actions are available to achieve that as she has no direct access to that information, so she sends a reply back to the world controller to reply with a list of all actions involving items that can make her less thirsty.
- The world controller looks at each item’s actions to find those that affect thirst. For this to work efficiently, some optimisations are needed in the algorithm to reduce checks on items that are irrelevant (such as the table and chair in this example). Also required is an estimate of what the effect of each action would be, as an attribute of either the item or the action (or both).

In this case, there are two actions available, ‘drink from the cup’ and ‘drink from the bottle’. The world controller would send these to Claire to decide upon.

⁶It would be best to limit the actions to only those that seem sensible in the context of the game. This would help with the complexity of the algorithms required to select actions.

- Claire now has two actions to choose from. The decision process would factor in her personality and the levels of her other drives, but would probably work best with an element of randomness included as well. In this example, lets assume that it is more ‘polite’ to drink from a cup than a bottle, so this is the action chosen by Claire.

A further complication that was avoided in the previous description is that the chosen action (‘drink from cup’) has a hidden level of complexity involved in its execution. There are a number of smaller steps involved in performing this action:

- Walk up to cup.
- Pick up cup.
- Lift cup to mouth.
- Actually drink from cup (at this stage the amount of water in the cup should be reduced, as well as the level of thirst of the character).
- Put down cup (this step is technically optional, but if omitted then the character is still holding the cup).

Although it might seem that a massive amount of scripting is required to code every action with every object, notice that replacing ‘cup’ in the above steps with ‘bottle’ converts the action to ‘drink from bottle’. So this list of steps can be written as a generic ‘drink from x ’ procedure, where x is any valid drinking vessel.

This is a simplification of what would happen in a more complex game world, as there would more likely be multiple drives that warrant attention from a character. In this case it would be best to pass all high valued drives to the world controller, and attempt to find actions that solve as many problems as possible (i.e. if a character is ‘cold’ and ‘thirsty’, drink a hot beverage instead of water). Another problem is dealing with scenarios where actions may not be immediately possible due to the nature of the current situation (such as Claire already holding an item, or the chair blocking the path to the cup). Some level of logic processing needs to be done to choose secondary actions (such as dropping the item or moving the chair) for this to be a successful algorithm in all situations. However, the general process would still be the same as that described here.

4 Showing personality

The above example shows how the game world and characters could be structured to allow the appearance of autonomous decision making (such as ‘I am thirsty, so I will drink from this cup’), but the decision chosen did not show any appreciable depth of personality. However, this process is applicable for more personality-based situations, as shown in the following example (see figure 2).



Figure 2: Another simple example world; two characters (Adam and Beth) and a puddle.

In this example, there are two characters (called Adam and Beth). Adam is walking toward Beth, when he trips and falls into a large puddle. This event can be used as an opportunity to show an aspect of Beth’s personality, via the following process.

The world controller detects Adam’s misfortune of falling in the puddle as an event that Beth should acknowledge. For this example, the controller could pass the following options to Beth to choose from:

1. Ignore Adam’s fall (if Beth is in the middle of an important action)
2. Sympathise with Adam (and help him out of the puddle).
3. Laugh at Adam (and insult his clumsiness).

Many more responses could be added (and probably should in a real application), but this is sufficient for this example.

If Beth is very busy with a task that cannot be interrupted, then Adam can be ignored. If Beth is not busy, then she will choose between the two

options based on her personality (is she the sort of person who enjoys other people's misfortunes or a more kind-hearted soul?), her relationship with Adam (is he a friend, enemy or stranger?), and the possible consequences of her action (how does she want Adam to react, and how important is his reaction to her?). The choice she makes will affect their relationship (i.e. Adam becomes more friendly if she helps him, more hostile if she insults him). With these kinds of personality-based decisions as part of the gameplay, meaningful interactions between characters can be added interactively, rather than relying on the pre-written cutscene elements used in existing games.

5 Conclusion

This has been a brief overview of what needs to be done to achieve non-linear stories in games, and some methods to approach development of the semi-autonomous characters needed to reach this goal. There are many issues not covered (or merely touched upon) in this paper that are also pertinent. These include:

- *Player interactivity:* The player's character needs be able to interact with the computer controlled characters and influence the direction of the story. An obvious way to help achieve this is to allow the player character the same interactions with other objects as the computer controlled characters (i.e. allow the player instead of the computer deciding to choose to 'drink from the cup'), which suggests a menu-based approach. However, for these types of games it is usual to design the interface in such a way that the player *becomes* the character, rather than merely directs the character (i.e. the interface controls the character at a very low level, such as 'move forward' rather than the more abstract higher level of 'drink from the cup'). In this case, some additional work needs to be done to achieve the same goal but still providing the immersion used in existing games.
- *Motive and plot generation:* Only mundane actions were considered in the actions above. For a dynamic story to be present in a game, the characters would need to act on more abstract motives, such as the need for adventure, desire for power or revenge, trying to rescue a loved one etc.
- *Complexity of a full-scale game world:* For a game world the size of a standard RPG there would be a multitude of characters and thousands of items to interact with. To make this run at a reasonable speed,

heavy consideration is needed to the optimisation of the decision making process. One simple optimisation would be to only consider actions with objects within a certain distance of a character (so if a character is thirsty they will not consider drinking from a cup in another house, for example).

While more work is needed in the modelling of reactive characters in games, it is worth considering the methods touched upon here when implementing a story within gameplay. Games have the potential to be a totally different form of story-telling from the traditional linear methods. Hopefully within a decade games can be used not merely as an extension of existing media, but as a story-telling medium in its own right.

A Related Work

There are a number of projects on creating individualism and autonomy in characters in both academia and industry. The following is a short list of some of these projects.

The Oz Project[1] (developers of ‘Edge of Intention’⁷) has developed a language called *Hap* [3, 4], to allow artists to individually script the personality of a character (by writing rules unique to the character to express their personality traits). This approach leads to the development of a distinctive character as specified by the vision of the artist, and is convenient for simulations containing a few characters with richly defined personalities.

An alternative approach is taken by the Virtual Theater Project [5], which involves creating dramatic pieces containing characters modelled in software. Instead of using individual scripting for each character, a personality is defined parametrically (i.e. the level of each personality trait is defined by a number), in a similar fashion to that discussed (briefly) in this paper.

Some attempts to give individualism to characters have been made in games featuring artificial life, for example in the *Creatures*TM[6] and *Petz*TM[7] series. Their goal is to model organisms’ behaviour at a low level, usually through the application of neural networks. While these can be used to create believable characters, it is hard to predict exactly what personality will arise due to the need for training. Also, sophisticated artificial life techniques tend to be computationally expensive, which restricts the depth and number of the characters allowed in a scenario.

⁷The ‘Edge of Intention’ simulation features a group of four ‘woggles’, sphere shaped creatures who can communicate through body language (and in later versions natural language), with one woggle human controlled.[2]

References

- [1] Michael Mateas. An Oz-Centric Review of Interactive Drama and Believable Agents. In *Lecture Notes in Artificial Intelligence*, volume 1600 of *Lecture Notes in Artificial Intelligence*, pages 297–329. Springer-Verlag, 1999.
- [2] Joseph Bates. The Role of Emotion in Believable Agents. *Communications of the ACM*, 37(7):122–125, April 1994.
- [3] A. Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, Computer Science Department, Carnegie Mellon University, May 1997.
- [4] W. Scott Neal Reilly. *Believable Social and Emotional Agents*. PhD thesis, Computer Science Department, Carnegie Mellon University, May 1996.
- [5] Daniel Rousseau and Barbara Hayes-Roth. A Social-Psychological Model for Synthetic Actors. Technical Report KSL 97-07, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, September 1997.
- [6] Eric Goodwin. How a Biochemical Metabolic Model can Contribute to Intelligent Lifelike Behaviour. In *1999 AAAI Symposium on Computer Games and Artificial Intelligence*, pages 51–54. AAAI, 1999.
- [7] Andrew Stern. Interactive fiction: the story is just beginning. *IEEE Intelligent Systems*, pages 16–18, November/December 1998.
- [8] Mark Barret. Irreconcilable Differences: Game vs. Story. *GameDev.net*, 1997.
- [9] Selmer Bringsjord. Is It Possible to Build Dramatically Compelling Interactive Digital Entertainment (in the form, e.g., of computer games)? *GameStudies*, July 2001.
- [10] Markku Eskelinen. The Gaming Situation. *GameStudies*, July 2001.
- [11] John Funge. AI for Games and Animation: A Cognitive Modelling Approach. *Gamasutra*, December 6 1999.
- [12] Randy Littlejohn. The Need to Adapt the Tools of Drama to Interactive Storytelling. *Gamasutra*, September 14 2001.

- [13] Pascal Luban. Turning a Linear Story into a Game: The Missing Link between Fiction and Interactive Entertainment. *Gamasutra*, June 15 2001.