Façade - An interactive drama

Ole Rehmsen

Saarland University, 66123 Saarbrcken, Germany

Abstract. Many modern games suffer from a trade-off between the degree of interactivity and the dramatical quality of the background story. This paper is a review of Façade, an interactive drama by Michael Mateas and Andrew Stern, designed to overcome this shortcoming. Through natural language input and physical actions, the player has to find his way through a domestic quarrel of a couple he is friends with. His actions have and major influence on the outcome of the story. This is achieved by structuring the story as a set of beats, short situations that determine the agents goals and behaviors. In any run through the game, only a small subset of the available beats will be used. The selection is based both on the users actions and the desired story tension arc. This leads to a user experience, which is both highly sensitive to the players actions and coherently forms an exciting story.

1 Introduction

1.1 Status quo

During the last couple of years, game development has evolved rapidly. New hard- and software enabled a whole new dimension of immersion, featuring a stunning level of graphical detail as well as physical realism that was unthinkable only a few years ago.

Other aspects of game development, however, have experienced almost no improvements during the last decade. While many developers strive for non-linear story-lines, giving the player the opportunity to make choices and experience their effects, this often leads to an unpleasant trade off between the depth and dramatic quality of the story and the amount of freedom for the player. The more choices are left to the player, the harder it is to ensure that events are believable and suspense is kept up during the game. In addition, to simplify storytelling, many games are designed to have different streams of action all ending up at the same point again. This means that the user has the choice to do something in several different ways, but without his choices having a long term effect on story or gameplay.

Another aspect that has not seen much improvement recently is the dialog system. Since the early days of monkey island, virtually nothing changed: The player can participate in a conversation with one person at a time, selecting predefined sentences from a list of usually about five options, to find out information he might need to solve quests or find interesting items. In most games, he simply has to exhaust all possibilities of a dialog tree, while the order in which the phrases are selected does not matter.

These shortcomings all to often make conversations a dull, unwelcome matter that keeps the player from participating in the actual main part of the game: Action. Looking at the list of typical quests in today's computer games, one finds that it is dominated by action: Go there, Shoot this guy, operate that lever, repair that tool, find this item. This is very different from most people's everyday life, which is dominated by interactions with other humans. Feelings and individuality, while being absolutely central to human beings, are simply not part of most computer games.

1.2 Enter Façade

Façade by Michael Mateas and Andrew Stern is designed to overcome these shortcomings. An interactive drama, it is meant to resolve the trade-off between immersion and interaction. Different from today's games, the player can enter free text in natural language at any time, and life-like agents will react immediately in a believable way. At any one time, the player can perform several actions like walking around in the room, padding another character on the back and looking around. These actions will have a lasting impact on both the world and the characters surrounding him. The same phrase might be picked up totally different by the characters in different situations. Chances missed might be lost for ever.

While offering the player this degree of freedom, Façade makes sure that the story develops in a dramatic way, with the tension following classical patterns like the Aristotelian arc. Ideally, the mechanics of how the story develops should not be obvious to the player.

1.3 Structure

The rest of this paper is organized as follows: Section 2 is about aspects of the demo of Façade that are not part of the framework, such as the story and the 3D world. In section 3, I will give an overview of the architecture of the framework and introduce the concept of beats. Section 4 is about believable agents in Façade including (joint) goals and behaviors. Section 5 is about the ways the player can interact with agents and environment. Section 6 explains drama management in Façade, especially how beats are selected to form an exciting story. Section 7 concludes the paper.

2 The Demo

To demonstrate their ideas, Mateas and Stern have developed both a framework for designing interactive dramas and a 20 minutes demo of such a drama. Both the framework and the demo are referred to as Façade. While in most cases it will be clear from the context, I will point out what is meant in cases where it is ambiguous.

2.1 The Story

Façade is the story of Grace and Trip, a married couple in their thirties the player has been friends with for many years. In the beginning, the player is invited over to their apartment for drinks, where he becomes quickly aware that Grace and Trips' marriage is in danger of falling apart. The players actions decide, if they break up, who is moving out and whether there friendship survives.

2.2 Choice of Story

This story has been selected, because it allows for many different ordering of the events. It is not central whether the conversation is first about the decoration and then about Grace's parents or the other way around. This leaves enough room for the players actions influencing the way the story develops.

If one used a story where everything has to happen in a very specific order to work out, this would greatly cripple the user's freedom of action and their effects. If there is only one ordering that yields a coherent story development, the player's actions do not matter, they cannot influence the story.

2.3 3D World

The demo of Façade features a non-realistic 3D world implemented with OpenGL in C++. The world consists of Trip and Grace's apartment, that is, the living room, a corridor and the kitchen. Most action takes place in the living room.

3 Architectorial overview

The Framework of Façade is supposed to allow for easy development of interactive dramas.

3.1 Languages

Many special purpose languages have been specified and implemented to make authoring quicker and code more readable:

A Behavioral Language (ABL) Based on Hap and used to specify the agents behavior. ABL is being compiled to Java. More on ABL in the section 4.

Natural Language Understanding (NLU) Template language A forward-chaining template language that allows the author to specify how to infer meaning from the text the user inputs. It maps natural language phrases into an intermediate representation called discourse acts. More on this in section 5.1. NLUTP is being compiled into Jess, which is a Java implementation of the CLIPS Rule Language.

Reaction Decider language Another forward-chaining template language. Maps discourse acts on the agents reaction. Also compiles to Jess.

Beat Sequencing language Is used to specify how to manage the dramatic development of the story. More on beats and beat selection in section 6. Compiles to Java.

3.2 Threads

Due to modular design and autonomous agents, the Façade framework has several threads running concurrently, reading and writing global story state.

- one for the drama manager
- one for each non-player agent
- one for the player agent

In the Demo of Façade, there is also a thread rendering the 3D world. This, however, is not part of the framework as explained in section 2.3.

3.3 Beats

The story of Façade is made of beats. In drama theory, a beat is the smallest unit of dramatic action that moves a story forward. In Façade, it can be understood as a situation, lasting about one minute. Each beat influences the actions and behavior of the agents and defines a new set of rules for interaction among them. Façade consists of hundreds of these beats, with only a small fraction being used in any one run through the game. This allows for truly diverse story developments with changing, interesting endings. The authors claim that Façade can be played up to seven times before the player gains the feeling of having exhausted all possibilities. Examples for possible beats are the "Fixing drinks"-beat or when Grace's or Trip's parents are calling.

In Façade, beats consist mainly of preconditions, effects, one or more beat goals and a set of rules that influence Grace and Trips behavior.

The preconditions must be fulfilled for the beat to run. For example, the "Decoration Beat" will not be triggered unless the player makes some reference to decoration, either verbally or by for example standing next to and looking at the wedding picture.

The effects determine the changes the completion of a beat has on the global story state. The tension for example might rise as the effect of the completion of the "Second Honeymoon Beat". The effects are also analyzed by the drama manager at beat selection time in order to find the beat best fitting to the desired tension arc. This will be explained in more detail in section 6.

The beat goals specify the agent's goals during this beat. More on goals and behaviors in section 4.3.

In addition to the aforementioned properties, beats may be assigned a particular priority and a weight influencing the likeliness that it is selected by the drama manager (See section 6).

4 Believable Agents

The main goal of Façade is to provide a very dense experience, yet allowing the user to largely influence the development of the story. Actually, allowing the user to influence the story adds to the impression of a very life-like and immersive world. One major role in achieving this degree of realism and atmosphere play believable agents. The characters of Façade are autonomous agents that maintain there own state, have an individual personality and pursuit there own set of goals. At the same time, they remain responsive to other agents and the players actions.

4.1 Working Memory

Agents interface the world through the so-called working memory. If something happens in the world, a *Working Memory Entry (WME)* is set accordingly. Working Memory Entries are like Java variables: They have a specific type and contain a value. In ABL, agents can check if a certain WME exists and which value it has. There are also language constructs for continuously monitoring working memory and notifying once a certain variable is set (a mechanism similar to event handling).

4.2 Motor System

The Façade Agents have a motor system attached to them, that performs physical actions like walking somewhere, moving the lips or gazing in certain directions. The concrete motor system itself is independent from the framework. As mentioned before, in the demo the motor system is an avatar in the 3D world, however, it could just as well be any other kind of output, or example a text-based UI or even robots. The motor system is interfaced in ABL using the *act* keyword.

4.3 Goals and Behaviors

The agents actions are modeled and coded as goals and behaviors. A goal might be something like the plan to let a knocking guest in. Goals are imposed on the agents by the current beat - for more details see section 3.3.

Behaviors are means of achieving goals: One behavior to reach the goal to let the guest in would be to yell “Come in!” in direction of the door. Often, their are multiple alternative behaviors to reach a certain goal. An alternative behavior to let the guest in might be to walk to the door and open. Each Agent has a *Behavior Library* containing all the behaviors he can perform.

A behavior can consist of actions (cf. section 4.2), mental actions and subgoals. Mental actions are pure Java computations, for example for managing working memory. Subgoals allow for nested behaviors. In the “going to the door and opening”-example, a subgoal might be to go to the door, and the other one to open it.

There are sequential and parallel behaviors. In sequential behaviors, the subgoals and actions are performed one after the other, in the order they are called in the behaviors body. An agent would first walk to the door, with all subgaols this might include, and not open it until the walking to the door behavior was successful. Parallel behaviors, on the other hand, execute all subgoals in parallel. An example would be a greeting behavior, where the agent verbally greets a guest as he is walking towards him to shake hands.

In addition to (mental) actions and subgoals, behaviors can be assigned preconditions and specificity numbers. Preconditions must be fulfilled before a particular behavior can be chosen to reach a goal. For example, one might want the yelling behavior only if the agent is sufficiently far away from the door. Specificity numbers define how special a certain behavior is. High specificity numbers mean very special, low ones mean very general solution. You might for example define a very general behavior, that always works, like walking to the door and opening. To make the experience more credible and flexible, you also have other behaviors like yelling, that can be done instead, however only makes sense in specific situations - for example if you are not standing right next to the door.

One very important feature of Façade's agents is that they can do multiple things at the same time. A characters could, for example, walk across the room, open his arms and greet a guest verbally all at the same time. The currently active behaviors are organized in the *Active Behavior Tree (ABT)*. Each node represents either a goal or a behavior. Each goal node has one child, which is a behavior node. This node represents the behavior currently tried to reach the goal. Subgoals are children of behavior goals.

A behavior can succeed or fail. Success and Failure are propagated through the tree. Normally, if one subgoal of a behavior fails, the whole behavior will fail. In parallel behaviors, however, one can specify how many of its subgoals have to succeed before the behavior succeeds. There is also a *with(persistent)* construct, that will repeat a certain behavior without caring about success or failure. A behavior can also be failed using timeouts. If all behaviors that can be used to reach a particular goal have been tried and failed, the goal will fail.

4.4 Synchronization

One very important aspect of believable agents is that, while being autonomous, they have to somehow interact with each other, react on each other. If Trip makes fun of Grace, then Grace should react angry or hurt and not act uninvolved. In Façade, this is achieved by a built-in synchronization mechanism, *Joint Goal and Behaviors*. This is also one major improvement in comparison to ABL's ancestor, Hap.

A Behavior can be modified with the keyword *joint*, meaning that this behavior cannot just be executed, but has to wait until all other team members agree to participate in this behavior. The set of team members for a particular joint behavior is specified at the very beginning of the behaviors body, via the *teammembers* keyword. If an agent wants to start a joint behavior, he broadcasts a request to all team members of this behavior and waits until he receives

responses. The team members who are asked to join a behavior check if they have a matching behavior in their library (which should be the case) and send an okay. Each agent might have his own version of the joint behavior. If, for example, the joint behavior is a discussion about the marriage, then grace might have a version where she complains about problems whereas Trip has a version where he tries to interrupt and becalm Grace.

If something unexpected happens, one of the agents might want to quit the joint behavior. If the player, for example, confesses to Grace that he loves her while she and Trip are discussing problems in their marriage, reacting on this might be a high priority goal that should precede further discussions on marriage problems. In such a situation, it would be unacceptable for Grace to quit the conversation while Trip keeps talking as if nothing happened. In Façade, if one of the agents wants to abort a joint behavior, he broadcasts this intention and quits only after he or she has received a confirmation to quit from all team members. The joint behavior is then aborted. Instead of aborting a joint behavior, it can also be suspended, so that it can be resumed after the high priority goal succeeded or failed. It will depend on the particular situation which of these options you want to use to achieve a realistic user experience.

Joint goals and behaviors can also be arbitrarily nested, meaning that joint behaviors may again have joint subgoals that enforce their own synchronization. In addition, one agent might in parallel be involved in two different joint behaviors with a different set of team members.

As always in concurrent programming, quite some effort has to be made to ensure that synchronization works properly in all situations. The Façade framework guarantees coordinated entry and exit for joint goals and behaviors, even if multiple team members want to abort and suspend a joint behavior at the same time or similar conflicts. This makes authoring in ABL much easier than it would be in pure Java or another multi-purpose language.

5 Interaction

We will now turn to another great strength of Façade, namely the very large degree of freedom for the player. The player can walk and look around in the world, pick things up, perform physical actions like padding the agents on the back and of course enter any text he or she wants at any time. More importantly, the player cannot only do all this, but the framework even provides mechanisms that can turn the users actions into something meaningful for the AI. This allows the author to define agent reactions to all kinds of behavior. If you walk to a picture on the wall and look at it for example, this may trigger Grace to start a conversation on her attempts to decorate the apartment. Or Trip and Grace might interpret the position you are standing at relative to each of them as an indicator of you taking sides for one of them. In the following, I will explain how interpretation of the player's actions works.

5.1 Natural Language Processing

The primary interaction tool of Façade is the natural language input. While you are to some extend able to convey non-verbal messages, the possibilities to send out whatever verbal message you want at whatever time plays the biggest role in interacting with the agents in the world.

Since Natural Language Understanding is a very complex problem that has not been fully solved up to day, Stern and Mateas have concentrated on a broad but shallow approach. This means that the agents are able to react to a relatively large variety of different statements, however the understanding is somewhat perfunctory and details are ignored.

The Natural Language Processing strategy of Façade consists of two steps:

1. Mapping the surface text to a set of *Discourse Acts*
2. Deciding on appropriate reactions on a Discourse Act

The first step is implemented in the Natural Language Understanding Template language, a forward-chaining template language. Forward chaining means looking at the input and trying to find from a set of conditional rules (if . . . then . . .) a rule whose condition is fulfilled. This rule is then used to infer some knowledge about the meaning. There are about 30 parameterized discourse acts in Façade, including (Harsh) criticism, Agreement, Flirt, Provocation etc. If nothing matches, there is a *Miscellaneous* discourse act that will eventually lead to a very general reaction. The mapping is total, meaning that whatever you say, it will always map to some discourse act. This comes at the expense of being overly permissive: Even ungrammatical utterances will be at least mapped to miscellaneous.

The seconds step of the NLP is written in the Reaction Decider language. This also is a forward-chaining template language, however now mapping discourse acts obtained in step one unto appropriate reactions. One very cool feature of Façade is, that the same discourse act might be reacted on very differently in different situations. A good example to demonstrate this is when Trip and Grace argue about the decoration. Grace is convinced that her decorating was not successful, and in this situation, she will actually appreciate criticism and even be disappointed if you compliment her for her design choices. In other situations however, she will be hurt if you criticize her for what she has done. Situations are modeled in Façade via Beats (see section 3.3). These situation-by-situation semantics are implemented via Beat-specific Reaction Decision rules: The agents first try to find an appropriate reaction on a given discourse act in their beat-specific reaction rules. Only if they do not find a fitting rule, meaning the player performed a discourse act that has no particular meaning in the current situation, will the agent fall back to his normal rules. Imagine in the midst of a discussion on what to drink, the player confesses he loves Trip. In the drinking situation, such discourse acts have no specific meaning, thus the agents will fall back to their general reactions, which in this case might be to act baffled or shocked by the player's sudden advance and his disrespect for social rules towards married couples.

Non-verbal interactions, like padding someone on his or her back, are treated similarly as the verbal interactions: They too are mapped to discourse acts and then fed in the reaction decision rules.

5.2 Interaction handlers

Once a reaction is chosen, it has to somehow be mixed into the current performance, to give the impression of a coherent, believable transition from one goal to another. This is where interaction handlers come in. They usually abort the current beat goal and create a new goal with a high priority that is responsible for reacting on the players interaction. After the reaction has finished, the original beat goal might be restarted if necessary. Interaction handler might even abort a whole beat and mix-in a transition to another, more appropriate beat. To allow modification of the Active Behavior Tree, ABL supports reflections: All nodes are wrapped in special working memory entries.

5.3 Local Agency

Local Agency is a concept that includes all immediate effects of the players interactions. On your wish to drink a Martini, for example, you will notice Grace's reaction of irritation, because she does not approve of drinking, and Trip's enthusiasm because you chose such a classic, classy drink. The logic of local agency is implemented in beats.

5.4 Global Agency

Besides the immediate and obvious responses, there are also long term effects of the players actions. If the player for example sides with one of the two, this will have an effect on both Trip and Grace long after the causing beat has finished. This so called Global Agency is implemented in the drama manager.

6 The Drama Manager

The purpose of the drama manager is to control the overall development of the story. It is responsible for selecting which beat should be used next. The beats are for this purpose organized as a *bag of beats*, which is nothing else but a set. The selection of the most appropriate beat works as follows:

6.1 Beat Selection

Evaluating Preconditions At first, all preconditions of all beats are evaluated. The beats whose preconditions are fulfilled form the *Satisfied* set of the beats.

Priority Next, the beats priority is evaluated: All beats that are not in the highest priority tier are removed. This results in the *Highest Priority* set.

Determining the appropriate tension Next, the effects of the highest priority beats on the story tension are examined. The beats are scored in how good they match the desired story tension, where low values denote a small error from the optimal tension.

Static Weighting As a last step, the remaining beats' score is multiplied with a static, beat-specific weight. The default weight is 1.0. By specifying a weight of, for example, 0.8, the author can make the selection of a particular beat less likely.

Selection We now have a set of weighted, scored, beats in the highest priority tier. The weighted scores determine the probability distribution of these beats. The upcoming beat is now selected randomly using this distribution.

6.2 Problems in beat selection

There are two problems that may occur with beat selection:

1. There is no beat whose preconditions are all satisfied. In this case, the authors should add new beats or alter existing beats to make sure there is always a satisfied beat.
2. The error between the available beats' tensions and the desired tensions exceeds a certain threshold. This situation occurs if the players actions do not match the intended story. Again, this information can be used during development to indicate that more beats are needed to accommodate for this kind of player interaction.

7 Conclusion

We have seen that today's video games, while offering superior graphics, often lack interactivity beyond action. They also suffer from a trade-off between story-depth and player's freedom of actions. We have seen that Façade, an interactive marriage drama, is able to resolve this trade-off to a large extend. The player can walk around, enter natural language via keyboard and perform certain physical actions like picking things up and padding the agents on their back. All of these actions have both immediate and long-term effects on the story's agents, Trip and Grace. This is achieved through splitting the story up into a large number of beats, situations that may or may not occur in a run through the game. In Façade, beats significantly alter the goals and behaviors of the autonomous agents. Through joint goals and behaviors, agents cannot only react on the player, but also on each other. While the demo of Façade uses a non-realistic 3D world to visualize the agents actions, the framework could just as well be connected to a text interface or even robots. Which beats are selected depends both on player interaction and on the desired tension slope. The latter

is implemented in the so-called drama manager, who with a non-trivial selection algorithm decides, which of the possible beats best fits a predefined curve.

To make authoring of an interactive drama using the Façade framework as easy as possible, Mateas and Stern have specified and implemented several different special purpose language that compile to Java or Jess. Using powerful features such as joint goals and behaviors, however, remains a tedious, time consuming task. This is an issue, which could and should be addressed in future work.

References

1. Mateas, M., Stern, A.: Façade: An Experiment in Building a Fully-Realized Interactive Drama (2003)
2. Mateas M., Stern A.: A Behavior Language: Joint Action and Behavioral Idioms (2004)