

Scrivere Façade: un caso studio nell'autorialità procedurale

di Michael Mateas | Andrew Stern.

(estratto da: Pat Harrigan e Noah Wardrip-Fruin (a cura di), *Second Person*, Cambridge (MA), MIT Press, 2007, pp. 183 - 207).

(traduzione di Antonio Pizzo)

1. Introduzione

La capacità di gestire procedure è l'essenza del computer in quanto strumento di rappresentazione – vale a dire la possibilità di gestire molteplici processi automatici a cui gli osservatori possano attribuire un significato. I computer, naturalmente, già aiutano la produzione di immagini, contribuiscono la comunicazione tra le persone tramite la mediazione di segnali a lunga distanza, controllano dispositivi elettromeccanici, sopportano la conservazione e l'interconnessione di grandi quantità di informazioni alle quale l'essere umano può accedere rapidamente. Gli utenti possono attivare queste diverse capacità del computer in molti modi, come la manipolazione di immagini o la creazione di pagine Web, senza che debbano necessariamente pensare in modo procedurale. Ma è proprio la capacità del computer di trasformarsi in queste diverse macchine con scopi specifici che mette in evidenza la natura procedurale del computer. Queste macchine specifiche (per esempio, gli strumenti, i *tools*) sono fatte di processi computazionali; in altre parole, rappresentano la capacità del computer di impegnarsi in processi diversi grazie ai quali di trasformarsi in diverse macchine diverse.

Per sfruttare al massimo la natura rappresentazionale del computer, bisogna avere la competenza di un **autore procedurale**, cioè, essere artisti e scrittori in grado di pensare e lavorare all'interno di strutture di calcolo. Con il termine competenza procedurale, intendiamo la capacità di leggere e scrivere i processi, impegnarsi in rappresentazioni procedurali ed estetiche, comprendere l'interazione tra le pratiche innatamente culturali della produzione di senso umana e i processi mediati dalla tecnologia. Questa alfabetizzazione procedurale è importante anche per i professionisti dei nuovi media, che in genere non scrivono molto codice, così che possano partecipare con successo in team di lavoro interdisciplinari, e per comprendere il ventaglio di possibilità che si apre per le opere digitali. Molti artisti si trovano, in un modo o nell'altro, a dover programmare, specialmente per le opere interattive che, necessariamente, si basano su input e risposte condizionali, e quindi sulla definizione di un processo. Nei casi più estremi, in cui si sviluppano nuove modalità di espressione computazionale, gli autori devono essere molto abili nell'uso di linguaggi di programmazione di base, al fine per

elaborare nuovi linguaggi e strumenti specializzati per la nuova modalità di rappresentazione.

In questo capitolo, proponiamo un caso di studio, utilizzando il **dramma interattivo *Façade***, che appartiene a questo ultimo tipo di autorialità procedurale. *Façade* rappresenta un nuovo modo di rappresentazione computazionale, **teatro interattivo**, che unisce il piacere dell'interazione *live* con personaggi credibili, tipico del gioco, con il piacere di partecipare e di influenzare uno sviluppo drammatico a lungo termine, ben compiuto, tipico della narrativa. Come autori procedurali, abbiamo affrontato diversi compiti di progettazione e di programmazione: decostruire un racconto drammatico in una gerarchia di storie e segmenti di comportamento, progettare un'architettura di intelligenza artificiale, cioè un insieme di linguaggi specifici all'interno dell'architettura, che rispondono (integrandole) momento per momento alle azioni dello user, così da comporre, con quei segmenti, una performance drammatica in real-time. Infine, utilizzando questo sistema, abbiamo dovuto scrivere una storia avvincente e coinvolgente.

Questo saggio sottolinea l'importanza dell'autorialità procedurale, descrive gli obiettivi di progettazione di *Façade* e come questi obiettivi possono essere raggiunti solo attraverso un approccio altamente procedurale alla narrativa interattiva. In ultimo, descrive l'architettura di *Façade*, l'organizzazione dei contenuti, e la nostra esperienza di produzione di contenuti in questo tipo di sistema.

2. Proceduralità

Janet Murray ha individuato quattro proprietà essenziali del computer come mezzo di rappresentazione: i computer sono di **natura procedurale, partecipativa, enciclopedica e spaziale** (Murray 1998). La caratteristica *procedurale*, ovviamente, si riferisce alla natura meccanica dei computer, che rappresentano processi causali complessi, e, in effetti, possiamo chiedere al computer di rappresentare qualsiasi processo arbitrario. La natura *partecipativa* si riferisce alla qualità interattiva dei computer, i quali possono dinamicamente rispondere ai segnali esterni e sono pensati in modo da rispondere con segnali ai quali la gente può attribuire un qualche significato (cioè, la risposta non arbitraria). La natura *enciclopedica* si riferisce alla capacità di stoccaggio di dati dei grandi calcolatori digitali, e alla capacità di organizzare, recuperare, e indicizzare i materiali stoccati. La natura *spaziale* si riferisce alla capacità dei calcolatori digitali di rappresentare lo spazio, che sia lo spazio fisico della realtà virtuale e dei videogiochi, o lo spazio astratto delle reti di informazione.

Diverse comunità di professionisti tendono a considerare diverse proprietà come basilari. Di seguito indichiamo alcuni esempi del prevalere di varie

proprietà. Per il *Demoscene*, una comunità (ma anche una subcultura) in gran parte dedicata alle gare, con gruppi e singoli artisti in competizione gli uni contro gli altri per la propria eccellenza tecnica e artistica (Wikipedia 2005), è centrale la *proceduralità*: l'obiettivo è quello di generare proceduralmente un'esperienza audiovisiva più ricca possibile, utilizzando però il minimo di contenuti memorizzati in precedenza. La *partecipazione* è privilegiata in quei precetti creativi in cui ci si concentra su *agency*, controllo, e collaborazione autoriale, ed è stata adottata da diverse comunità dedicate allo *user-interface design*, all'arte interattiva, e al Digital Marketing. Le opere che si basano sull'uso di database privilegiano la natura *enciclopedica*, a volte considerando tutta la *new media art* come metaforicamente legata alla manipolazione e al ri-sequenziamento degli archivi di dati. La *spazialità* è privilegiata da quelle comunità legate alla realtà virtuale, al game design, e all'ipertesto.

Sebbene tutte queste proprietà possono giocare un qualche ruolo nei diversi media computazionali, **la proceduralità resta la proprietà essenziale** e definitiva dei media computazionali, poiché senza di essa le altre proprietà non potrebbero esistere.

Qualsiasi sistema partecipativo richiede di indicare le azioni potenziali che potrebbero avvenire in risposta a uno eventuale stimolo. Progettare una possibile azione, vuol dire definire una macchina o un algoritmo che possa attualizzare/definire eventi solo in potenza (che potrebbero accadere) secondo le diverse contingenze. In altre parole, i sistemi partecipativi sono proceduralità. Ma non è vero il contrario, ci possono essere sistemi procedurali che non sono partecipativi, ma che eseguono processi fissi senza accettare input. Molti sistemi di arte generativa, come *Aaron* (McCorduck 1991) (<http://www.kurzweilcyberart.com>), sono procedurali senza essere partecipativi.

Anche i sistemi enciclopedici dipendono dalla proceduralità. Infatti i grandi archivi da dati sarebbero inutili se fossero privati della capacità di eseguire operazioni sui dati, e cioè di essere in grado di accedere, riorganizzare, ricercare, modificare, indicizzare, e così via. Senza le competenze procedurali dei motori di ricerca sul web, quest'ultimo, letteralmente, non potrebbe esistere nella sua grandezza e pervasività attuale. Non ci sarebbe alcun motivo di creare una nuova pagina Web, se non avessimo, ad esempio, la possibilità di linkare la pagina a quelle già pubblicate, o se gli altri non fossero in grado di trovare e visualizzare la nostra pagina. E ancora una volta, non è vero il contrario. I processi possono creare esperienze molto elaborate da un *kernel* (il cuore del sistema operativo) molto ridotto; in altre parole la capacità procedurale può svilupparsi anche su un limitato volume di dati: questa capacità è, del resto, l'ispirazione per il *Demoscene*.

La spazialità è chiaramente una proprietà derivata, si tratta di un'illusione creata da una particolare rappresentazione prodotta da un processo. Le

rappresentazioni grafiche spaziali utilizzano modelli procedurali per calcolare e aggiornare dinamicamente lo spazio visualizzato. D'altro canto, quegli ambienti interattivi che creano il senso dello spazio permettendo una navigazione individuale, e che magari non utilizzano per nulla rappresentazioni grafiche 2D o 3D (si pensi ai vecchi ambienti testuali multiutente organizzati per *room*), dipendono dalla partecipazione, che a sua volta dipende dalla proceduralità.

Sia chiaro che qui non stiamo tentando di giocare "a chi è il più forte" tra le varie qualità rappresentazionali del computer; piuttosto vogliamo eliminare le gravi confusioni che possono sorgere nella teoria e nella pratica dei nuovi media a causa di un fraintendimento della fondamentale importanza della proceduralità. Senza una profonda comprensione del rapporto tra ciò che risiede nello schermo e al di là di esso, gli studiosi non possono comprendere in profondità la funzione dei nuovi media, mentre i professionisti, che vivono nella casa-prigione degli strumenti "*art-friendly*", non possono sfruttare il vero potere rappresentativo del calcolo come medium.

Se gli studiosi di nuovi media non comprendono la proceduralità, se non sono sanno come il codice funziona in quanto strumento espressivo, si troveranno costretti a trattare il funzionamento delle opere che studiano, come una sorta di scatola nera, perdendo il rapporto cruciale tra l'autorialità, il codice, e la risposta del pubblico. **Il codice è un tipo di scrittura.** Così come gli studiosi di letteratura non si sognerebbero mai di leggere solo le glosse tradotte dell'opera e mai l'opera completa in lingua originale, allo stesso modo gli studiosi dei media dovrebbero leggere il codice, non solo al semplice livello di operazioni primitive e controllo di flusso, ma a livello di retorica, estetica e poetica procedurale codificate nel lavoro.

I professionisti dei nuovi media senza un'alfabetizzazione procedurale sono limitati alla produzione di quei sistemi interattivi che gli sono permessi nei limiti segnati dagli attuali strumenti di *authoring*. Fino ad oggi, tali strumenti tendono ad avere un orientamento enciclopedico e, in mancanza di un sostegno significativo all'autorialità procedurale (ad esempio, di programmazione), il compito dell'autore consiste solo nel mettere insieme diverse risorse media (video, audio, testo, immagine, ecc) e nella comporle, nello spazio e nel tempo, all'interno del quadro procedurale permesso dagli strumenti (ad esempio, *linking*). Questo approccio limita fundamentalmente le dimensioni e la complessità dei lavori new media. Per le opere interattive, questo problema è particolarmente grave, in quanto costringe l'autore a pre-specificare e scrivere in modo esplicito le risposte a tutte le possibili situazioni interattive.

2.1 Proceduralità e contenuti

Per descrivere il rapporto tra il calcolo e le risorse multimediali, Chris Crawford ha introdotto il termine **intensità di processo** (Crawford 1987). L'intensità del processo è il "*crunch per bit*", il rapporto tra la computazione e la mole di risorse multimedia che vengono manipolate dal sistema. Se un gioco (o qualsiasi altro software interattivo) come attività primaria si limita ad attivare la riproduzione di contenuti multimediali in risposta all'interazione, allora diremo che si tratta di un processo a bassa intensità. Il codice sta lavorando poco - in sostanza non fa altro che spalare bit dal supporto di memoria (CD-ROM, hard drive, ...) verso lo schermo e gli altoparlanti. Se un gioco (o qualsiasi altro software interattivo) manipola e fonde risorse multimediali, allora diremo che il suo processo aumenta di intensità. Se le immagini e i suoni sono generati algoritmicamente, e quindi non si utilizzano risorse prodotte offline (registrate), allora diremo che hanno una massima intensità di processo.

L'intensità del processo permette direttamente la ricchezza di interattività. Se diminuisce l'intensità di processo, l'autore deve produrre un maggior numero di risorse offline (ad esempio, porzioni di testo predefinite, animazioni e video) per rispondere alle diverse interazioni possibili. Il numero di risorse offline necessarie per mantenere un determinato livello di interattività aumenta in modo esponenziale quando si riduce l'intensità processo. Quindi, in generale, se diminuisce l'intensità nel processo, diminuisce la ricchezza di interattività.

Anche se i giochi, all'interno dei prodotti digitali, hanno un'intensità relativamente elevata di processo, i giochi contemporanei stanno mettendo alla prova i limiti dell'authoring causati da un eccesso di dipendenza da risorse statiche e non-procedurali. I giochi contemporanei come *Il Signore degli Anelli* (Electronic Arts) attualmente contengono più file multimediali di linee di codice (Mateas 2005). Anche i giochi *open-world* come il *Grand Theft Auto*, lodati per i loro ambienti simulati e procedurali, fanno ancora molto uso di risorse statiche per ogni veicolo, per ogni tipo di persona, per ogni palazzo, arma, e così via.

Inoltre, gli sviluppatori a una recente conferenza di "Game Developers" hanno espresso la preoccupazione che l'hardware delle console di prossima generazione non farà che aggravare questa crisi di contenuti (Taylor 2005). La richiesta di una grafica sempre più dettagliata per invogliare i consumatori ad acquistare le console di nuova generazione, significa che le risorse diventano sempre più costose da produrre, e quindi che ci sarà bisogno di team di lavoro sempre più numerosi, così che i giochi diventeranno più costosi. I consumatori reclamano più attività di gioco, e ciò vale a dire giochi più grandi, il che richiede più risorse media da produrre: tutto ciò conduce a un circolo incrementale che molti ormai considerano insostenibile.

L'insufficiente dose di proceduralità non solo sta creando una crisi nella produzione di giochi tradizionali, ma ha anche impedito lo sviluppo di progetti sull'arte interattiva e di intrattenimento, come ad esempio le narrazioni con una forte dose di interazione. Unire processi di alta intensità, le tecniche di Intelligenza Artificiale con le questioni che chiamano in causa la narrativa interattiva, è stato uno degli obiettivi di ricerca fondamentali per il nostro dramma interattivo *Façade*.

3. I contenuti procedurali nell'Interactive Drama *Façade*

3.1 Un caso di studio per contenuti procedurali.

Spinti dalla nostra convinzione che la ricerca sui metodi di authoring altamente procedurali permetterà la nascita di forme di arte o intrattenimento finora sconosciute, ci siamo impegnati nello sviluppo del dramma interattivo *Façade*. Il sogno di un teatro interattivo, forse meglio immaginato dall'holodeck di *Star Trek*, è presentato per la prima volta in un contesto accademico da Brenda Laurel nel suo *Computers as Theatre* (Laurel 1991), prevede giocatori che interagiscono con personaggi affascinanti e psicologicamente complessi, e che, proprio grazie a queste interazioni, influenzano e modificano in modo significativo una trama che evolve dinamicamente. Potendo contare su circa un decennio di ricerca del progetto Oz alla Carnegie Mellon (Bates 1992; Loyall 1997), e convinti che un dramma interattivo pienamente sviluppato non fosse stato ancora costruito, ci siamo imbarcati per cinque anni nello sviluppo di un sistema di authoring procedurale per la creazione di personaggi credibili, di conversazioni in linguaggio naturale, e di una trama dinamica; il tutto integrato in una breve ma completa esperienza di gioco. Disponibile al pubblico nel luglio del 2005, *Façade* è stato scaricato, fino a questo momento, da oltre 150.000 giocatori in tutto il mondo, e ha ricevuto un ampio successo di critica (Montfort 2005).

I videogiochi più godibili tendono ad essere altamente procedurali nella loro costituzione, in quanto tra i vari metodi di implementazioni, la proceduralità offre il massimo grado di dinamismo e reattività - caratteristiche molto piacevoli per i giocatori. I migliori videogiochi procedurali sono capaci di far provare ai giocatori esperienze con un alto grado di *agency*, cioè, forniscono molte opportunità di azione e di feedback immediati. Con *Façade*, abbiamo voluto creare un dramma interattivo che fornisce il livello *agency* immediata, momento per momento, cioè, proprio quel tipo di ***agency locale*** che troviamo nei video giochi. Ma a differenza di giochi, vogliamo che il giocatore faccia esperienza anche di una ***agency globale***, cioè, una influenza più a lungo termine su l'arco complessivo della narrativa, su come gli argomenti si sviluppano, su quello che i personaggi "sentono" per il giocatore nel corso del tempo, e su come la storia si conclude.

Come i giochi moderni, *Façade* è ambientato in un ambiente simulato in tempo reale con un'animazione 3D e contenuti sonori; offre al giocatore un'interfaccia in prima persona, continua e diretta, con una navigazione non vincolata e la possibilità di manipolare alcuni oggetti. Così come un dramma, in particolare i drammi basati sulle relazioni personali come *Chi ha paura di Virginia Woolf?* (Albee 1962), *Façade* usa il linguaggio naturale spontaneo e la gestualità emotiva come le primarie modalità di espressione per tutti i personaggi, tra cui lo stesso giocatore. La storia narrata non tratta di salvare il mondo, o mettere in salvo principesse combattendo mostri, bensì riguarda il coinvolgimento emotivo e i rapporti umani: in particolare si concentra sulla dissoluzione di un matrimonio. C'è l'unità di tempo e di spazio - tutta l'azione si svolge in un appartamento - e la struttura complessiva dell'evento è modulata per conformarsi a un arco drammatico delineato secondo il principio di tensione aristotelica; al suo interno, quindi, si possono riconoscere: evento scatenante, aumento di tensione, crisi, climax e conclusione. Il tutto in modo indipendente da ciò che avviene in dettaglio in ognuna delle sessioni di gioco.

Inoltre, le scelte operate a livello della storia, in *Façade*, sono progettate in modo da non essere percepite come palesi snodi della ramificazione. Noi crediamo che quando un giocatore si trova di fronte a punti di scelta troppo evidenti, costituiti da un numero limitato di possibilità (ad esempio, un menu di tre diverse possibili cose da dire), il senso di agency diminuisce, e il giocatore si sente su un binario fisso, costretto a fare ciò che il progettista ha deciso. Invece, in *Façade*, la progressione della storia cambia in risposta alle tante piccole azioni svolte dal giocatore durante l'esperienza. In seguito, in questo capitolo, descriveremo in dettaglio i contenuti procedurali di *Façade*, e come il gioco raggiunge gli obiettivi dai quali siamo partiti.

3.2 I problemi dei contenuti con poca o nulla proceduralità.

Gli autori che affrontano le storie interattive hanno sempre dovuto affrontare un annoso enigma: come si può includere l'interazione in una storia ben strutturata, e nello stesso tempo mantenere una trama soddisfacente e ben fatta, nel momento in cui viene vissuta dal lettore / giocatore? Storicamente, i progetti di storie interattive poco e per nulla procedurali sono stati costretti a operare un compromesso tra questi due obiettivi. In genere, la "storia interattiva" che ne risulta è dotata di una trama ben formata, ma può essere solo minimamente influenzata dal lettore / giocatore, come si è visto nei plot da narrativa lineare della maggior parte dei giochi e di alcuni testi avventure di narrativa interattiva (Interactive Fiction).

In alternativa, il bilanciamento nella progettazione può essere indirizzato nella direzione opposta, con conseguenti esperienze interattive che possono variare notevolmente a seconda dell'azione giocatore, ma che però mancano del grado

di coerenza, ritmo e attenzione che rappresentano una qualità fondamentale nelle storie ben costruite. L'approccio enciclopedico e non-procedurale alla progettazione, laddove l'autore crea un gran numero di segmenti di storia statica (risorse) sequenziati da un sistema semplice, conduce inevitabilmente questo tipo di compensazione nella progettazione. L'autore può scegliere di porre vincoli minimi su come ordinare i segmenti di storia, così da permettere il sequenziamento locale dei segmenti a seconda dell'interazione locale del giocatore. Ma poi le sequenze prodotte mancheranno di quella coerenza tipica degli archi drammatici ben formati. Le trame frammentate, o trame fortemente diluite con pezzi di azione disorganizzati o non utili, sono comuni nella narrativa ipertestuale, così come in alcune Interactive Fictions, e di conseguenza non appaiono storie corrette e ben formate.

Nell'ambito di un approccio al design enciclopedico, l'unico modo per aumentare l'interattività è creare una straordinaria quantità di contenuti con la "forza bruta". Questa strategia si è rivelata impraticabile: anche i migliori libri "Choose Your Own Adventure" o i loro equivalenti digitali, dove la trama può variare in modo significativo in risposta alle scelte del lettore ed essere ben formata, si trovano necessariamente ad offrire una serie di brevi, insoddisfacenti e infrequenti scelte binarie, al fine di evitare un'esplosione combinatoria di esplicite scelte drammatiche predefinite (già pronte). In questo tipo di approccio, è chiaro il carattere limitato e scomodo di un approccio enciclopedico e non-procedurale.

Il compromesso enciclopedico tra la coerenza e l'esplosione combinatoria, visto a livello della trama, si riflette anche al livello più specifico del dialogo dei personaggi. La combinazione tra bassa coerenza nel sequenziamento e processi semplificati, appare in modo lampante nei **ChatterBots**, in cui le linee di dialogo sono selezionate all'interno di un grande repository, in risposta ad ogni interazione (input) del giocatore, facendo un uso limitato o nullo del contesto (il topic della conversazione), e dipendendo principalmente da una semplice regola stimolo / risposta. D'altro canto i sistemi che prediligono una alta coerenza nella sequenza di dialoghi, del tipo "Choose Your Own Adventure", sono accomunati dalla struttura dei dialoghi ad albero, in cui un autore deve rappresentare in modo esplicito e statico il contesto del discorso progettando e realizzando tutti i percorsi possibili del dialogo, con una conseguente esplosione combinatoria (tipiche delle strutture ad albero).

Sulla base di queste frustranti limitazioni negli approcci precedenti alla storia interattiva, i due tipi di agency, livello locale e livello globale, sono comunemente considerate incompatibili.

3.3 Il design di storie procedurali

La nostra soluzione in *Façade* per questo annoso problema è di riconsiderare le interazioni del giocatore all'interno di una storia in termini astratti di *giochi di società* [qui gli autori intendono quei giochi – ad esempio, da tavolo – che coinvolgono più persone che interagiscono tra di loro sulla base di regole e con l'ausilio di supporti specifici, *N.d.T.*]. I giochi sono procedurali per propria natura, e raggiungono un elevato grado di variabilità di eventi e agency del giocatore, come noi desideriamo; la sfida si sposta sul come progettare e strutturare un tipo di giochi che riflettano i particolari significati che vogliamo per la nostra storia, e come eseguire drammaticamente giochi che siano tanto coerenti e mirati, quanto le narrazioni ben ritmate.

Inoltre, per essere compatibile con la naturale procedurale e simulativa dei giochi, la granularità dei segmenti di storia (gli elementi fissi precostituiti) deve essere insolitamente piccola, nell'ordine di singole e ricombinabili espressioni facciali, di gesti e di battute, e non a livello dei brani con più frasi e delle lunghe clip video/grafiche. Come descritto in dettaglio più avanti, i segmenti di contenuti in *Façade* sono organizzati in molteplici e miscelabili livelli gerarchici, organizzati in sequenza da procedure scritte in numerosi linguaggi di authoring componibili tra di loro.

Ad un livello più alto, i giochi di società astratti di *Façade* sono organizzati intorno ad un "punteggio" numerico, come ad esempio l'affinità tra un personaggio e il giocatore. Tuttavia, a differenza dei videogiochi tradizionali in cui vi è una connessione abbastanza diretta tra interazione del giocatore (ad esempio, premendo un pulsante fare fuoco con una pistola) e lo stato punteggio (ad esempio, una diminuzione della salute di un mostro), l'attività ludica in *Façade* ha diversi livelli di astrazione, separando così le interazioni atomiche del giocatore dai cambiamenti nel "punteggio" della vita sociale. Invece di saltare sugli ostacoli o sparare, in *Façade*, i giocatori "sparano" una serie di atti discorsivi in linguaggio naturale, come *lode*, *critica*, *flirt*, *provocazione* (vedi tabella 30.1). Mentre questi atti discorsivi produrranno reazioni immediate nei personaggi, potrebbero essere necessari degli atti discorsivi relativi alla struttura specifica del contesto narrativo per influenzare il punteggio sociale gioco. Inoltre, il punteggio non è direttamente comunicato al giocatore tramite numeri o cursori, ma piuttosto attraverso una ricca e teatrale performance drammatica.

Il giocatore, in qualità di amico, è invitato per un drink dai due protagonisti, Grace e Trip, in un momento critico e cruciale del loro difficile matrimonio. Il giocatore diventa inconsapevolmente una sorta di antagonista, costretto da Grace e Trip in gioco psicologico del tipo "Head Games" [giochi mentali] (Berna 1964). Durante la prima parte della storia, Grace e Trip interpretano tutti gli atti discorsivi del giocatore nei termini di un *gioco di affinità* a somma zero, e in questo modo i due litiganti si formeranno un'opinione riguardo la

posizione del giocatore (chi dei due sta ottenendo il supporto del giocatore). Allo stesso tempo, ha luogo un *hot-button game*, in cui il giocatore può attivare argomenti incendiari come il sesso o il divorzio, progredendo attraverso i livelli per ottenere informazioni sui personaggi e il background; e qualora si spinga troppo in là su un argomento, si possono verificare inversioni di affinità. La seconda parte del racconto è organizzato intorno al “gioco terapia”, dove il giocatore sta potenzialmente aumentando (volutamente o non) il grado di coscienza di sé di ciascun personaggio rispetto ai propri problemi (all'interno del sistema questa parte è rappresentata come una serie di contatori). Inoltre, il sistema tiene traccia del livello globale di tensione nella storia, che è influenzato dalle mosse del giocatore nei diversi momenti d'interazione sociale. Ogni cambiamento in ogni stato del gioco viene reso da Grace e Trip in maniera emotivamente espressiva e drammatica. Nel complesso, poiché i loro atteggiamenti, i livelli di consapevolezza di sé, e la tensione generale progrediscono regolarmente, l'esperienza assume la forma e l'estetica di un dramma domestico più o meno unitario.

In quanto la granularità dei segmenti atomici di storia (ad esempio, il dialogo, l'espressività emotiva e i comportamenti gestuali) diventa molto piccola, e le procedure per ordinarli e riunirli in una performance narrativa coerente diventano fondamentali per la concretizzazione dell'esperienza del giocatore, l'attività dell'autore si sposta da quella di uno scrittore di prosa a uno scrittore di procedure, vale a dire un un programmatore.

3.4 Ricchezza mediante una coerente amalgama.

Per eseguire drammaticamente le varie interazioni sociali di *Façade* come narrazioni coerenti, mirate, e ben ritmate, è necessario un principio organizzativo che si distacchi dai vincoli delle tradizionali strutture narrative a ramificazione, per evitare l'esplosione combinatoria che scaturirebbe da una complessa catena di eventi causali (Crawford 1989). Il nostro approccio a questo problema in *Façade* è duplice: in primo luogo, si divide il racconto in *più fronti di progressione*, spesso causalmente indipendenti, e solo occasionalmente interdipendenti. In secondo luogo, costruiamo una varietà di *sequencer* (sequenziatori) narrativi per organizzare queste narrazioni progressive multiple. Questi sequencer procedurali, che descriveremo in seguito, operano in parallelo e possono amalgamare coerentemente le attività tra di loro.

L'architettura di *Façade* e la struttura dei contenuti sono le due facce della stessa medaglia, e saranno descritte in tandem; nel corso del nostro discorso, spiegheremo come è ottenuta questa amalgama coerente.

3.4.1 Architettura e Content Framework

Il sistema di *Façade* è costituito da diversi sottosistemi procedurali che operano simultaneamente e comunicano tra loro (Mateas e Stern 2000, 2003a, 2003b, 2004a, 2004b). Ognuno di essi è descritto brevemente qui di seguito.

La performance dinamica momento-per-momento dei personaggi Grace e Trip - il modo in cui svolgono il loro dialogo, il modo in cui esprimono le emozioni, come seguono il giocatore nello spazio e utilizzano gli oggetti - sono scritti come una vasta collezione di *comportamenti* (behaviors), che sono brevi procedure reattive che rappresentano numerosi goal (obiettivi) e sub-goal per i personaggi, disposti secondo un'ampia struttura ad albero, gerarchica, e che cambia dinamicamente. Questi comportamenti sono scritti in un linguaggio di programmazione reattiva chiamato "**A Behavior Language**" (ABL), sviluppato nell'ambito del progetto di *Façade*, che gestisce le interrelazioni di comportamenti sia paralleli sia in sequenza come successo o fallimento di sub-goal, la priorità, i conflitti, le precondizioni e le condizioni specifiche del contesto.

I sequencer narrativi per le interazioni sociali sono anche scritti in ABL, approfittando della capacità di ABL di produrre meta-comportamenti che modificano lo stato di *runtime* (tempo di esecuzione) di altri comportamenti.

Il sequencer narrativo di più alto livello, un sottosistema chiamato il **drama manager** (gestore dramma), mette in sequenza i **beats** drammatici secondo le specifiche scritte in un linguaggio *ad-hoc* creato per la gestione del dramma. I beats in *Façade* sono grandi gruppi di comportamenti organizzati attorno ad un tema particolare, come descritto nella prossima sezione.

Un altro sottosistema è un set di regole per la **comprensione e l'interpretazione del linguaggio naturale (NL)** e degli input gestuali del giocatore. Tali norme sono scritte in un linguaggio personalizzato implementato con Jess, un linguaggio con regole di concatenazione in sequenze di avanzamento. Quando il giocatore intraprende in dialogo, queste regole di NL interpretano uno o più significati (gli atti discorsivi di cui sopra). Una seconda serie di norme chiamate **proponenti di reazioni** (*reaction proposers*) interpretano ulteriormente questi atti in una modalità che tiene conto del contesto specifico, come ad esempio accordo, disaccordo, alleanza, o provocazione, e invia questa interpretazione al gestore dei comportamenti e al drama manager affinché reagiscano.

Il sottosistema finale è un **motore di animazione** sviluppato per l'occasione che esegue l'azione del personaggio, l'espressione emotiva e il dialogo parlato, secondo i modi di una riproduzione procedurale in real-time e non fotorealistico, così come la musica e il suono. Il motore di animazione è guidato dai comportamenti scritti in ABL; il motore recepisce anche le

informazioni sulla posizione e sulle azioni di ogni personaggio, al fine di decidere quali comportamenti usare.

Tabella 30.1 Atti discorsivi in *Façade*

1. agree	21. pacify
2. disagree	22. criticize
3. generalExclamation	23. oppose
4. positiveExclamation	24. greet
5. negativeExclamation	25. goodbye
6. express <emotion>	26. getAttention
7. maybeUnsure	27. inappropriateObscene
8. dontUnderstand	28. explain
9. thank	<explainAdviceDescriptor>
10. apologize	29. advice
11. referTo <character>	<explainAdviceDescriptor>
<object> <topic>	30. explainRelationship
<theme>	<character1>
12. physicallyFavor <object>	<relationshipDescriptor>
13. praise	<character 2>
14. hugComfort	31. leaveApartment
15. flirt	32. leaveForKitchen
16. kiss	33. uncooperativeNotSpeaking
17. showConcern	34. uncooperativeNotMoving
18. howAreYou	35. uncooperativeFidgety
19. areYouOkay	36. systemDoesntUnderstand
20. showSupport	

3.4.2 Beats, Goals Beat e Beat Mix-ins

La prima messa in sequenza di segmenti narrativi in *Façade* avviene all'interno dei *beat*, per i quali ci siamo ispirati alla più piccola unità di azione drammatica come descritta nelle teorie di scrittura drammatica (McKee 1997). Tuttavia, **i beat di *Façade* finiscono per essere strutture più grandi dei beat canonici della scrittura drammatica.** In quest'ultima, un beat tende a consistere in poche righe di dialogo che trasmettono una singola coppia narrativa di azione / reazione.

Ad esempio, nella scena di *Casablanca* in cui Rick affronta Ilsa sulle ragioni del suo ritorno, lo scambio che segue è un singolo beat:: RICK: "Perché sei tornata? Per dirmi perché mi abbandonasti alla stazione ferroviaria?" ILSA: "Sì".

Un beat in *Façade*, tuttavia, può andare da **10 a 100 join dialogue behaviors (JDBs** -brani di dialogo e comportamento condivisi da più personaggi), scritti

in ABL. Ogni beat in sé è un sequencer narrativo, responsabile per il sequenziamento di un sottoinsieme dei suoi JDBs in risposta all'interazione del giocatore. **In ogni momento è attivo un solo beat.** Un JDB, l'unità atomica di azione drammatica in *Façade*, (e più vicino alla definizione canonica di beat nella scrittura drammatica), consiste in uno scambio drammatico, ben coordinato, da 1 a 5 battute di dialogo tra Grace e Trip; e dura normalmente pochi secondi. Di solito i JDBs sono costituiti da 50 a 200 linee di codice ABL. Gli JDB di un Beat sono organizzati intorno ad un comune obiettivo narrativo, come un breve conflitto su un argomento, l'ossessione di Grace per arredare l'appartamento, o la rivelazione di un segreto importante, come il tentativo Trip di costringere Grace a vivere la loro seconda luna di miele in Italia. Ogni JDB è in grado di cambiare uno o più valori nello stato della storia, così come il valore nel gioco di affinità, o uno dei qualsiasi contatori nella progressiva autorivelazione del gioco di terapia, o il livello complessivo di tensione nella storia. All'interno dei beat, i sequencer narrativi implementano il gioco di affinità: l'argomento di un beat è un'istanza particolare del gioco di affinità.

Ogni beat può essere visto come un gruppo di contenuti procedurali, in particolare JDBs, che sono dinamicamente messi in sequenza dalla logica specifica di ogni beat. Il drama manager è, a sua volta, una sacca di contenuti procedurali, in particolare di beat, che sono messi in sequenza dinamicamente secondo la logica generale del drama manager, in quanto influenzato dalle precondizioni, i pesi, le priorità, ecc specificati per ogni beat. La logica necessaria per mettere in sequenza le singole battute del dialogo è più dettagliata e complessa di quanto si possa facilmente descrivere nelle annotazioni dichiarative a livello del drama manager, è proprio per questo i nostri beat finiscono per essere più grandi dei beat tradizionali della scrittura drammatica.

La sequenza dettagliata e il coordinamento delle singole linee di dialogo sono più facilmente espresse in ABL piuttosto che nel linguaggio di descrizione dei beat, e infatti cambia in tal misura da contesto a contesto all'interno del dramma da rendere non fattibile un generico processo decisionale per il sequenziamento battute di dialogo (almeno, non senza un più profonda rappresentazione della conoscenza, un profondo ragionamento circa le relazioni umane, includendo anche qualche ragionamento base di senso comune, ecc.) Quindi, cerchiamo di spingere questa dettagliata logica nel sequencer narrativo personalizzato, scritto in ABL, che vive all'interno di ogni beat, lasciando che il gestore dramma (drama manager) metta in ordine i grossi blocchi di contenuto narrativo le cui interrelazioni sono abbastanza semplici da essere gestite da un più generico processo decisionale operante a questo livello.

Vi sono due usi tipici di JDBs all'interno dei beats: come ***beat goals*** e ***beat mix-ins***. Un beat è costituito da una sequenza canonica di obiettivi narrativi chiamati *beat goals*. La sequenza tipica canonica consiste di un **goal**

transition-in che fornisce una transizione narrativa all'interno del beat (ad esempio, portando un nuovo argomento, magari connesso al tema precedente), diversi **body goal** completano il beat (nei beat del gioco di affinità, i body goal stabiliscono il tema specifico del conflitto tra Grace e Trip e costringono il giocatore a scegliere da che parte stare), un **goal di attesa** in cui Grace e Trip attendono che il giocatore risponda al head game stabilito dal beat, e una **transition-out** di default che conduce fuori dal beat in caso che non ci sia interazione da parte del giocatore. In generale, i goal transition-out rivelano informazioni e comunicano in che modo le azioni del giocatore all'interno del beat hanno cambiato la dinamica delle affinità.

La sequenza canonica beat goal in un beat definisce il modo in cui il beat sarebbe proseguito nel caso di assenza di interazione. Oltre ai beat goal, esiste un insieme di meta comportamenti chiamato **handler**, che attende per le interpretazioni specifiche del dialogo del giocatore (atti del discorso), e **modifica la sequenza canonica** di conseguenza, in genere utilizzando **mix beat-ins**. Cioè, la logica dell'**handler** implementa il sequencer narrativo personalizzato per il beat. I JDBs dei **beat-mix in** sono beat-reazioni specifiche utilizzate per rispondere alle azioni di gioco e ricollegare l'interazione alla sequenza canonica. Gli **Handlers** sono potenzialmente responsabili sia di aggiungere, rimuovere, e riordinare i beat goal futuri, come pure di introdurre **mix beat-ins** nella sequenza canonica. Manipolando in questo modo la sequenza logica narrativa e i beat goals, si evita di dover manualmente dispiegare la sequenza logica all'interno degli stessi beat goal JDBs, evitando così il problema del dialogo ad albero menzionato in precedenza.

Per *Façade*, un'esperienza che dura circa 20 minuti e richiede diversi replay per vedere tutti i contenuti disponibili (un'unica sessione di gioco esegue al massimo il 25% del contenuto totale disponibile), abbiamo scritto circa 2.500 JDBs. Circa il 66% di questi 2.500 sono i **beat goal** e **beat mix-ins**, organizzati in 27 beat distinti, di cui circa 15 sono incontrati dal giocatore in ognuna delle possibili sessioni di gioco (vedere la sezione del drama manager).

3.4.3 Global Mix-in di progressione

Un altro tipo di sequencer narrativo, che opera in parallelo, e può mescolarsi con, i beat goal e i beat mix-in, sono **global mix-ins**. (In seguito descriviamo come realizziamo una amalgama coerente.) Ogni categoria di global mix-ins ha tre livelli, andando sempre più a fondo nell'argomento; l'avanzamento dei livelli è causato dall'interazione del giocatore (ad esempio, il giocatore nomina o discute un argomento specifico). Ciascun livello della progressione è costruito da uno o più JDBs, proprio come i beat goal o i **beat mix-ins**. Sono focalizzati su argomenti satellite (come il matrimonio, il divorzio, il sesso e la terapia), o sugli oggetti (come i mobili, le bevande, la loro fotografia di

matrimonio, il toro di ottone, o la vista dalla finestra), o come reazioni generiche (lode, critica, flirt, opposizioni, e simili). Inoltre, ci sono una varietà di global mix-ins di deviazione e recupero per rispondere agli input del giocatore quando sono troppo confusi o inadeguati. In totale, vi sono circa 20 casi di questo tipo di sequencer narrativi in *Façade*, di cui circa il 33% dei circa 2.500 JDBs totale.

3.4.4 Drama Management (Sequencing Beat)

La sequenza narrativa più grezza in *Façade* si verifica nel drama manager, o *beat sequencer*, come si vede nella tabella 30.2.

I nomi dei 27 beat in *Façade*

PlayerArrives	PlayerFollowsTripToKitchen
TripGreetsPlayer	TripReturnsFromKitchen
PlayerEntersTripGetsGrace	TripReenactsProposal
GraceGreetsPlayer	BlowupCrisis
ArgueOverRedecorating	PostCrisis
ExplainDatingAnniversary	TherapyGame
ArgueOverItalyVacation	RevelationsBuildup
FightOverFixingDrinks	Revelations
PhoneCallFromParents	EndingNoRevelations
TransitionToTension2	EndingSelfRevelationsOnly
GraceStormsToKitchen	EndingRelationshipRevelationsOnly
PlayerFollowsGraceToKitchen	EndingBothNotFullySelfAware
GraceReturnsFromKitchen	EndingBothSelfAware
TripStormsToKitchen	

Il drama manager resta dormiente per la maggior parte del tempo, ed è attivo solo quando il beat attuale termina o si annulla (per decisione dello stesso beat, o di un globale mix-in). È al livello del sequenziamento dei beat, dove si gestisce la dipendenza causale tra gli eventi più importanti - cioè, in cui sono prese le decisioni di trama, ad alto livello.

Nel linguaggio di sequenziamento dei beat, l'autore annota ogni beat con determinate conoscenze che consistono in precondizioni, pesi, test del peso, priorità, test di priorità, valori della storia - nel caso di *Façade*, il livello generale di tensione. Dato un insieme di beat rappresentato nel linguaggio dei beat, come ad esempio i ventisette elencati nella tabella 30.2, il beat sequencer seleziona il beat successivo da eseguire. Il beat non ancora utilizzato, le cui condizioni sono soddisfatte e il cui valor di tensione nella storia coincide con la traiettoria più breve all'interno di un arco drammatico disegnato dall'autore (in *Façade*, un arco di tensione aristotelico) è quello prescelto; anche i pesi e le priorità influenzano la decisione (Mateas e Stern 2003b).

Il sequenziamento dei beat è ulteriormente discusso nella sezione sulla mescolanza coerente, così come in quella su fallimenti e successi.

3.4.5 Comportamenti Mix-in automatici a lungo termine

Comportamenti automatici a lungo termine, come servire i drink e berli nel corso del tempo, o il giocare compulsivamente con un giocattolo (la sfera dei saggi consigli, ad esempio), durano più a lungo dei sessanta secondi dei beat dei dieci secondi dei global mix-in. Anche se forse svolgono un ruolo narrativo minore, di tanto in tanto introdotti in un JDB nel beat in corso (che comprende solo l'1% dei JDBs di *Façade*), essi contribuiscono molto a dare una apparenza di intelligenza ai personaggi, facendoli eseguire per esteso, e coerentemente una serie di azioni di basso livello, in background, nel corso di molti minuti, oltre i confini dei diversi beat. Eseguendo simultaneamente, e per intero, comportamenti autonomi e comportamenti comuni, i personaggi di *Façade* sono un ibrido tra i diversi approcci al coordinamento di agenti (ai due estremi troviamo il modello "una sola mente", e all'altro quello "molte menti"), diventando di fatto un agente "multi-mente" (Mateas e Stern 2004a).

3.5 Strategie per una mescolanza coerente

Poiché i global mix-in per le interazioni da *hot-button* sono scelti tra gli i beat goal / ins-mix per il gioco di affinità, che operano in parallelo con il drama manager nel far progredire la tensione generale della storia, sono necessarie diverse strategie per mantenere la coerenza, sia in termini di gestione del discorso e sia di flusso narrativo.

In primo luogo, le progressioni global mix-in sono scritte per essere causalmente indipendenti da ogni flusso narrativo di beats. Per esempio, mentre cavillano sulla loro seconda luna di miele in Italia, o discutono su quale tipo di drink Trip dovrebbe servire (beat di gioco di affinità, scelto dal drama manager), è conveniente inserire un dialogo su, per esempio, il sesso, o la fotografia del matrimonio (hot button game mix-in innescato dal fatto che il giocatore può aver menzionato questi argomenti). Ogni dialogo mix-in è scritto e recitato con la voce come se fossero argomenti un po' tangenziali che vengono gettati nel flusso della conversazione ("Oh, quella foto, sì, è veramente...").

A livello del discorso, esistono meccanismi per gestire agevolmente tali interruzioni. Se durante un beat goal (ad esempio, i ricordi di Trip circa il cibo in Italia) viene attivato un global mix-in (ad esempio, il giocatore prende il toro d'ottone regalo di una amante di Trip - e quindi ciò è interpretato come un riferimento a questo oggetto), l'attuale beat goal sull'Italia s'interrompere immediatamente nel mezzo della sua esecuzione, e global mix-in "il toro di

ottone” sarà messo in esecuzione, a qualsiasi livello al quale quel gioco hot-button sia già arrivato. Al momento dell’interruzione, se l’attuale beat goal “Italia” non aveva ancora superato il suo punto essenziale, che è un punto nei JDBs di un beat goal definito dall’autore, esso dovrà essere ripetuto quando il global mix-in (intanto entrato in esecuzione) sarà completato. Un breve dialogo alternato e non interrompibile è scritto per ognuno di questi beat goal a tale scopo. Inoltre, ogni beat goal ha un JDB di ristabilimento, che viene eseguito se si torna al beat specifico da un global mix-in (“Allora, stavo dicendo, riguardo l’Italia ...”). I mix-ins per se stessi possono essere interrotti da altri mix-in, ma in tal caso, non sono ripetuti come accade per i beat goal.

A parte poche eccezioni, gli stessi beat di gioco di affinità sono stati progettati per essere causalmente indipendenti l’uno dall’altro. Per esempio, in termini di mantenimento della coerenza, non importa in quale ordine Grace e Trip discutono dell’Italia, dei loro genitori, dell’arredamento, dei drink, o della data del loro anniversario. Quando i beat vengono messi in sequenza, questo consente al drama manager di preferire l’inserimento di qualsiasi beat che sia in relazione agli argomenti passati introdotti dal giocatore. Allo stesso modo, hot button mix-ins possono essere convenientemente attivati in qualsiasi ordine, in quasi ogni beat, in qualsiasi momento.

Tuttavia, c’è stato un importante sforzo autoriale per rendere il tono di ciascun beat goal / mix-in e global mix -in relazione agli altri durante l’esecuzione. La maggior parte dei JDBs sono scritti con tre a cinque alternative per esprimere il loro contenuto narrativo secondo le diverse combinazioni di affinità di giocatore e livello di tensione. Questi includono variazioni nella scelta delle parole, nella qualità della voce, di emozione, del gesto, e le variazioni appropriate delle informazioni rivelate. Possedendo il tono di hot-button, i global mix-in e gli beat goal / mix-ins di gioco di affinità coincidono sempre tra essi; e i giocatori spesso li percepiscono in relazione causale, anche se non lo sono. Inoltre, per ogni tono, la maggior parte dei JDBs sono scritti con due, tre o quattro alternative di dialogo, equivalenti per la funzionalità narrativa, ma che contribuiscono a creare un senso di freschezza e di non artificialità nei personaggi nelle sessioni del dramma.

4. Esempio dettagliato di creazione procedurale di contenuti in *Façade*

Per rendere concreta la nostra discussione sulla creazione di narrativa e dialogo all’interno di una cornice procedurale, descriveremo il processo di creazione di uno specifico beat nella storia del dramma interattivo *Façade*. Creare un beat in *Façade* comporta la combinazione di design dell’interazione, di scrittura del dialogo, e di programmazione, che riassumiamo qui di seguito.

4.1 Progettare la struttura centrale di un beat in *Façade*

Il nostro esempio sarà il beat “FightOverFixingDrinks”, in cui Trip e Grace discutono su che tipo di drink servire al giocatore, costruito in modo da rivelare alcune delle tensioni sottostanti tra loro, e sviluppare ulteriormente i personaggi. Nel momento in cui in cui questo beat si può verificare, la prima metà del dramma, la coppia Grace e Trip, il cui matrimonio ha raggiunto il punto di rottura, sta facendo del proprio meglio per comportarsi come se tutto fosse a posto. Nell specifico, in questo beat Trip mostra la propria fissazione per il drink, come un modo per vantarsi di quanto la loro coppia sia benestante e colta. Grace, tuttavia, incoraggiata dalla presenza del giocatore, mira a contrastare Trip con un tentativo di attaccarlo sul piano del suo materialismo e la sua finta sofisticazione. Grace e Trip sfideranno il giocatore e a prendere posizione su queste differenze.

Per prima cosa definiamo una struttura relativamente semplice per il beat, a cui possiamo aggiungere ulteriore ricchezza nel corso del lavoro. Per questo beat abbiamo progettato come struttura di base, la seguente sequenza di beat goal:

- transizione-in sul beat- Trip porta in primo piano l'idea dei drink.
- Trip fa un suggerimento iniziale, con spavalderia; Grace reagisce inizialmente al quel vantarsi. Attendono qualche secondo una risposta giocatore, nel caso ci sia.
- Grace contrattacca con il proprio suggerimento, basato su ciò che il giocatore ha detto, contrastando Trip; Trip resiste. Attendono qualche secondo un'altra risposta del giocatore, nel caso ci sia.
- transizione-out dal beat - sia Trip sia Grace reagiscono alla decisione del giocatore, e Trip inizia a servire i drink.

È importante notare che ogni beat goal qui descritto è relativamente breve, per esempio non più di dieci secondi ciascuno, idealmente 5 secondi o meno. Questa dimensione minima della granularità dei beat goal permette ad altri beat goal di essere mescolati più facilmente all'interno della sequenza (come di seguito descritto). Se un beat goal è durato più di dieci secondi, bisognerebbe dividerlo in multiple parti come piccoli beat goal.

4.2 Reattività aggiunge ricchezza

A seguire, descriviamo i requisiti aggiuntivi di reattività per questo Beat, che aggiungeranno ulteriore ricchezza all'interazione. Tali requisiti includono:

- In qualsiasi momento durante il beat, il giocatore dovrebbe essere in grado di interrompere quello che Grace e Trip stanno dicendo e ottenere una

risposta immediata di qualche tipo. Qualunque sia il dialogo che è stato interrotto dovrebbe poter essere nuovamente ripreso in seguito, in modo credibile, se necessario.

- In qualsiasi momento durante il beat, il giocatore dovrebbe essere in grado di introdurre altri argomenti o fare azioni che non sono direttamente collegate al tema della servire un drink, e anche in questo caso ottenere una risposta da Grace e Trip, come descritto in precedenza. Questi global mix-in includono il procedere delle risposte ad argomenti tangenziali come il divorzio, il sesso, o la terapia, o sugli oggetti, come i mobili, le loro foto del matrimonio, o il toro di ottone, o reazioni generiche di lode, la critica, corteggiamenti, le opposizioni, e simili. Dopo la risposta, Grace e Trip dovrebbe tornare a far progredire lo stesso beat originale, in modo coerente.
- Ogni volta *dopo* il beat, nel momento in cui l'azione è entrata in un altro beat, il giocatore dovrebbe essere in grado di fare riferimento a quanto accaduto in precedenza nel corso di questo beat e ottenere una risposta di qualche tipo. Noi chiamiamo questo un post-beat mix-in.

A sostegno di queste esigenze la reattività, si aggiungeranno le seguenti caratteristiche specifiche per la struttura del beat:

- Punti nodali: ciascun beat goal deve essere annotato con un punto nodale, per sapere quanto in fondo in un beat goal il giocatore deve essere andato per evitare di doverlo ripetere, qualora fosse interrotto per eseguire qualche altro mix-in.
- Ripetere il dialogo: ogni beat goal ha bisogno di una variazione del dialogo nel caso in cui il beat goal debba essere ripetuto, qualora fosse interrotto, al fine di effettuare un mix-in.
- Ristabilire il dialogo: ogni beat goal ha bisogno di una battuta di prefazione che può ristabilire il suo contesto, nel caso in cui il beat goal precedente fosse un global mix-in e il beat goal corrente sta tornando a ciò di cui stava parlando in precedenza. Questi sono usati come prefissi alla ripetizione di dialogo.
- Dialogo locale deviato: ogni beat goal ha bisogno di un piccolo insieme di dialoghi locali di diversione, da utilizzare nel caso in cui il giocatore interrompa il beat goal con una parola molto generica, per la quale non c'è un global-mix-in adeguato. Questi sono essenzialmente local mix-ins.

4.3 Prestazioni in una varietà di contesti che aggiunge ricchezza

In aggiunta ai requisiti di reattività descritti finora, vogliamo che questo beat possa essere utilizzato in diversi contesti. Ad esempio, il dialogo specifico, e forse la sua struttura, dovrebbe variare se il beat viene eseguito all'inizio del

dramma, quando la tensione è ancora bassa, rispetto a un po' più avanti, quando la tensione è aumentata. (Una volta che la tensione ha raggiunto un livello molto elevato, in quanto gli autori, abbiamo deciso che Trip non sarà in vena di servire a nessuno un drink, e quindi questo beat non potrà più aver luogo.)

Inoltre, il beat dovrebbe variare il dialogo specifico, e forse la sua struttura, se il giocatore si è schierato con Grace, o con Trip, o ha mantenuto una posizione neutrale, indipendentemente dal livello di tensione. In realtà, se cambia l'affinità del giocatore durante il beat, il beat dovrebbe usare la sua variazione di dialogo in modo appropriato.

Infine, questo beat, per sua natura, può essere eseguito una seconda volta, se è passato del tempo dalla prima volta che è stato eseguito. Cioè, se il giocatore desidera che Trip gli serva un secondo drink, ciò dovrebbe essere possibile. Ci deve essere sufficiente dialogo interno e variazione di struttura al fine di evitare che si ripeta lo stesso dialogo una seconda volta, con conseguente perdita di credibilità.

Per sostenere una tale varietà di contesto, si aggiungeranno le seguenti caratteristiche specifiche alla struttura dei nostri beat:

- Ogni beat goal sarà scritto con variazioni di dialogo per ogni combinazione di livello di tensione (bassa o media) e di ogni valore di affinità giocatore (neutro, schierandosi-con-Grace, schierandosi-con-Trip), per un totale di $2 \times 3 = 6$ varianti.
- Quando il beat si sta svolgendo al secondo (medio) livello di tensione, creeremo diverse transition-out beat goal (finali), per il beat, in cui Grace rivela ad alta voce uno dei sconvolgenti segreti legati all'alcol di Trip in *Façade*, come un segreto avversione per il gusto del liquore, il suo lavoro segreto in un college come umile barista, o come si intrufola regolarmente in un bar sport proletario in fondo alla strada. Tutti questi saranno divisi tra le variazioni della struttura tensione/affinità.

Soddisfare i requisiti elencati in questa sezione e in quella precedente contribuisce a creare agency per il giocatore, perché consentono al giocatore di essere causa dell'attivazione del beat che sta giocando a suo piacimento. Questi requisiti inoltre aumentano la credibilità drammatica, perché ha senso che si richieda di servire un drink in ogni momento, almeno fino a quando il livello di tensione del dramma non diventi troppo grande. Senza l'ausilio di tali requisiti, la tempistica e la struttura del discorso e il dramma totale possono sembrare arbitrariamente e innaturalmente costretti, riducendo in modo significativo la agency e la credibilità; cioè, i problemi di cui sopra con lo status quo dei prodotti commerciali e non commerciali di narrative interattive.

4.4 Il dialogo con alternative aggiunge ricchezza

Idealmente ogni riga di dialogo ha diverse varianti. Ad esempio, da tre a cinque alternative, tutte con lo stesso significato drammatico, ma con diversi fraseggi e scelta delle parole. Mentre solo una delle alternative sarà ascoltata per ogni dialogo della sessione, il giocatore avrà l'opportunità di notare questa variante la prossima volta che gioca con *Façade* e si troverà di nuovo in questo beat, o se capita che questo beat è ripetuto una seconda volta nella stessa sessione.

4.5 Il Comportamento Parallelo aggiunge ricchezza

Grace e Trip sono elementi fondamentali per la verosimiglianza e la credibilità drammatica, quindi devono recitare comportamenti espressivi e paralleli come parte dei loro beat goal:

- Quando Grace e Trip recitano il loro dialogo, dovrebbero esprimere il loro stato d'animo anche attraverso l'espressione del viso, con lo sguardo e i gesti. Il dialogo specifico che recitano durante il beat influirà sui loro stati d'animo, naturalmente, ma l'umore generale potrà essere influenzato anche da qualsiasi altro degli eventi che sono accaduti prima di questo beat, così come da qualsiasi mix-ins che si verifica durante il beat. Ad esempio, se entra in gioco un global mix-in riguardo il divorzio nel corso di questo beat, ciò potrà inasprire lo stato d'animo di Trip, anche se aveva cominciato allegramente preparando dei drink. Inoltre, mentre un personaggio parla, tutti gli altri dovrebbero reagire dinamicamente. È per questo che l'autore deve scrivere coppie di comportamenti/dialogo comuni per ciascun carattere; il comportamento deve essere scritto per i personaggi che non parlano in modo da controllare la maniera in cui reagiscono al dialogo recitato dal personaggio parlante.

I personaggi, mentre dicono le loro battute, dovrebbero tendere a seguire il giocatore ovunque vada nella stanza. Ciò significa che, in generale, il dialogo dovrebbe essere scritto in modo da non dipendere da dove il personaggio si trovi.

Ci sono beats cui il dialogo è dipendente da un luogo preciso della stanza; ad esempio, il dialogo in *ArgueOverItalyVacation* richiede che Trip stia vicino alla foto dell'Italia accanto al bar e un gesto verso di essa (o da dietro il banco, come caso speciale, dal momento che la foto è vicina al bar). Al contrario, il beat *FightOverFixingDrinks*, è uno dei tipi più comuni di quei beat che dovrebbero essere eseguibili in qualsiasi punto della stanza.

In qualsiasi momento durante il beat, Trip potrebbe decidere di andare dietro il bancone e preparare i bicchieri mentre parla, in previsione di preparare i drink. Come variazione di dialogo alternato questa differenza di tempistica,

sarà rilevante per le successive attuazioni di questo beat in questa sessione o nelle successive. Per questa ragione il dialogo del beat deve essere scritto in modo che appaia credibile in entrambe le situazioni, laddove Trip sia dietro il bancone o in un altro punto della stanza.

4.6 Semplificazioni / Astrazioni per ridurre la complessità

Ci sono alcuni aspetti di questo progetto che possono essere semplificati e / o astratti per ridurre la complessità della sua implementazione, pur mantenendo un sufficiente livello di agency e di credibilità per il giocatore.

- Semplificare la mappatura delle espressioni giocatore/azioni sui diversi significati, riducendo il numero di reazioni che l'autore deve prevedere nella storia. Idealmente, vorremmo creare una reazione diversa (più il dialogo alternato) per ogni atto discorsivo che il giocatore possa esprimere, in ogni diverso contesto di un beat. Tuttavia, ci sono decine di atti discorsivi possibili (cfr. tabella 30.1), e, potenzialmente, altrettanti contesti all'interno di un beat quanti sono i possibili goal. Per esempio, all'incirca da cinque a dieci per ogni beat. Cercare di permutarle tutte si tradurrebbe, per l'autore, nella scrittura di centinaia di reazioni per ogni beat. Invece, per semplificare, abbiamo raggruppato insieme atti discorsivi in relazione gli uni con gli altri in maniere che corrispondono a contesti specifici. Ad esempio, se Trip offre un Martini e spera in una risposta positiva del giocatore, diversi atti discorsivi simili possono essere raggruppati ed essere letti come "accordo con Trip" in questo contesto, come ad esempio: accordo ("yes"), positiveExclamation ("sweet!"), grazie ("thanks"), esprimere felicità ("that makes me happy"), o il gesto di abbracciare. Anche se questo richiede la creazione di una mappatura personalizzata per i contesti del discorso, riduce comunque il lavoro rispetto alla scrittura di decine di reazioni individuali all'interno di ogni contesto.

Generalmente, ogni beat definisce un contesto di discorso, anche se ci possono essere molteplici contesti distinti all'interno di un solo beat. I contesti di discorso più piccoli sono associati con i singoli obiettivi del beat, anche se i beat possono avere contesti per i sub-beat che si prolungano nel tempo (vale a dire per diversi beat). Poiché non possiamo raggruppare gli stessi atti discorsivi insieme, in generale, ogni beat avrà bisogno di una mappatura personalizzata a partire dall'atto discorsivo fino al significato specifico del beat. Per esempio, anche se "thank" e "positiveExclamation" hanno entrambe lo stesso significato specifico (essere d'accordo) nel beat FightOverFixingDrinks, in altri contesti del dramma possono avere significati distinti e diversi per un beat specifico.

Ridurre la dipendenza causale. In precedenza, abbiamo definito come obiettivo di progettazione, quello di permettere reazioni ad argomenti "tangenziali" da poter essere *mix in* (introdotte) in qualsiasi momento (vale a dire i global mix-

in, descritti in precedenza). Ad esempio, dopo che Trip offre un Martini, se il giocatore menziona il divorzio, Trip deve rispondere sull'argomento divorzio, e, si auspica, dopo poco riproporre il Martini. Ma potrebbe essere possibile che il riferimento al divorzio, o qualsiasi altra cosa, possa cambiare tanto la situazione che non avrebbe più senso continuare a proporre Martini, o qualsiasi cosa si stesse facendo e dicendo prima. Per poter affrontare/trattare questa evenienza, cerchiamo di progettare la narrazione e scrivere gli specifici dialoghi in modo da ridurre tali dipendenze causali. La battuta di Trip che risponde al divorzio, mentre rivela sottilmente una certa tensione o emozioni nascoste, ce lo mostra anche mentre cerca di nascondere sotto il tappeto, il che gli consente di tornare a ciò che stava dicendo in modo credibile. Ad esempio, quando si torna a proporre un Martini, l'umore Trip può farsi più scuro, intervenendo sulle sue espressioni facciali e sul linguaggio del corpo da quel punto in avanti, ma senza richiedere necessariamente che si cambi significativamente la struttura di *FightOverFixingDrinks*. Questa strategia, ovviamente, ha come conseguenza una riduzione della agency globale. Sebbene il giocatore abbia causato localmente una risposta immediata (una reazione alla menzione del divorzio), cioè agency locale, egli causa meno effetti narrativi sul lungo termine (ad esempio, un cambiamento significativo nel modo in cui si discute di drink da quel momento in poi). Come autori cerchiamo di supplire a tale riduzione della agency ritardando l'effetto narrativo causato dall'introduzione dell'argomento "divorzio", rimandando a un momento successivo la risposta più adatta, quando sarà più facile a farlo; per esempio, in un beat (come ad esempio il beat "BlowupCrisis") che riepiloga esplicitamente le cose provocatorie che il giocatore ha detto in precedenza.

Unire i contesti insieme, quando possibile. In precedenza, abbiamo dichiarato quale obiettivo di progettazione che ogni beat goal dovrà essere scritto con variazioni di dialogo per ogni combinazione di livello di tensione (bassa o media) e per ogni valore di affinità giocatore (neutro, schierandosi-con-Grace, schierandosi-con-Trip), per un totale di $2 \times 3 = 6$ varianti. Tuttavia, alcuni di questi contesti sono abbastanza simili da poter essere uniti insieme. In particolare, nel caso del beat in cui Trip offre un drink, in quanto autori potremmo immaginare che Trip agirà con livelli simili di millanteria sia se ha affinità con il giocatore, o se l'affinità è neutrale, mentre agirà in modo diverso se è Grace ad avere affinità con il giocatore. Inoltre, come autori potremmo decidere che, una volta che la tensione è aumentata ad un livello "medio", non abbia alcun senso che l'affinità giocatore sia "neutrale". Se, nel momento in cui la tensione sale a medio termine, il giocatore ha ancora un'affinità neutrale, noi spingeremo l'affinità verso Trip o Grace. Ognuna di queste semplificazioni serve a rimuovere un contesto dalla lista, riducendo il totale a quattro, e quindi riducendo così gli oneri di authoring per la "*FightOverFixingDrinks*" del 33%.

Scrivere il dialogo per consentire brevi momenti di ininterrompibilità, riducendo la necessità di ripetere il dialogo in caso di interruzione. Come descritto in precedenza, ciascun beat goal dovrebbe avere una variazione di dialogo nel caso in cui il beat goal fosse interrotto da un mix-in e dovesse essere ripetuto. Tuttavia, è possibile eliminare la necessità di ripetere un dialogo per un beat goal se siamo in grado di scrivere il dialogo del beat goal così da comunicare velocemente l'essenza del suo significato nei primi secondi di svolgimento, e dichiarare (annotare) questo primo breve lasso di tempo come non interrompibile. Cioè, se il giocatore parla durante i primi secondi di un tale beat goal, la risposta di Trip e Grace è ritardata fino a quando l'essenza del beat goal è raggiunta – si tratta di un ritardo nella reazione di pochi secondi, che è il limite accettabile per la credibilità. Se l'essenza del significato del beat goal viene comunicata in quei pochi secondi, siamo in grado di interrompere il beat goal, al fine di eseguire un mix-in in risposta all'interruzione, senza doverci preoccupare di ripetere in seguito il beat goal interrotto. Ciò richiede di scrivere il dialogo in modo che la quantità minima dei contenuti necessari per la progressione narrativa del beat (il minimo di informazioni affinché il beat abbia senso) sia comunicata all'incirca all'inizio del beat goal, e facendo in modo che il resto del dialogo del beat goal serva ad aggiungere ricchezza, colore e dettagli che integrano i contenuti di base. Come regola generale, l'autore deve evitare battute di dialogo lunghe e complesse. Deve, al contrario, rompere il dialogo in più battute, così che possa essere interrotto al limite della battuta (una battuta completamente interrompibile può naturalmente essere interrotta in qualsiasi punto).

4.7 Authoring di un battito Goal

Ora che il nostro progetto è a posto, siamo pronti per la scrittura dei beat goal. Per non dilungarci troppo, mostreremo solo i dettagli di due beat goal del beat "FightOverFixingDrinks". Per la precisione, il secondo e terzo beat goal dalla struttura di base che abbiamo mostrato prima, che qui chiamiamo "TripSuggest" e "GraceCounterSuggest":

TripSuggest: Trip fa una proposta iniziale, vantandosi; Grace inizialmente reagisce al vantarsi. Attendono qualche secondo per una risposta giocatore, nel caso ci sia. (La risposta di Grace e Trip al giocatore avviene nel beat goal successivo, "GraceCounterSuggest".)

GraceCounterSuggest: Trip risponde al giocatore, e Grace fa una controproposta con i propri suggerimenti, sulla base di quando ha detto il giocatore, attaccando Trip; Trip resiste. Attendono qualche secondo per un'altra risposta giocatore, nel caso ci sia. (La risposta di Grace e Trip al giocatore avviene nel beat goal successivo.)

Scriveremo il dialogo in varie fasi, partendo solo con “TripSuggest”, e aggiungendo ricchezza nel corso del lavoro. In ognuna delle fasi di scrittura, il ***grassetto corsivo*** indica i cambiamenti rispetto alla fase precedente. Per ragioni di brevità, mostreremo uno pseudocodice tra parentesi angolari, e non l’effettivo codice ABL. “T:” e “G:” denotano dialogo parlato rispettivamente da Trip e Grace. Laddove la parola “giocatore” appare nel dialogo, sostituisce il vero nome del giocatore, per esempio, “Brenda”.

4.7.1 L’impalcatura

Ecco alcune battute di base di dialogo per “TripSuggest” che possono fungere da impalcature per il processo di authoring.

“TripSuggest”

T: (allegro) Cosa gradisci?

T: (allegro) Che ne diresti di un Martini?

T: (vantandosi) Sono un vero esperto in questo, almeno e quello che mi dicono tutti.

G: (un po’ infastidita) Oh Dio, Trip, per favore ... Non strafare con la preparazione drink.

4.7.2 Ininterrompibilità, punto essenziale, e ristabilire il dialogo

Come descritto in precedenza, innanzitutto stabiliamo che questo beat goal sia non interrompibile, poi settiamo il punto della sua interrompibilità, e poi definiamo il suo punto essenziale pochi secondi dopo. Inoltre, prefissiamo una battuta di ristabilimento del dialogo, che dovrà essere utilizzata se il contesto del beat deve essere ristabilito a causa di un’interruzione di un gloobal mix-in.

“TripSuggest”

<set uninteruptible>

<if reestablish> T: (allegro) Dunque! Un drink!

T: (allegro) Cosa gradisci?

T: (allegro) Che ne diresti di un Martini?

<set interruptible>

<set gist point>

T: (vantarsi) Sono un vero esperto in questo, almeno e quello che mi dicono tutti.

G: (un po' infastidita) Oh Dio, Trip, per favore ... Non strafare con la preparazione drink.

Ciò significa che se il giocatore parla all'inizio del beat goal, Trip non smette di parlare fino a quando ha dice: " Che ne diresti di un Martini?". Solo in seguito si avrà la reazione (qualunque essa sia), e le ultime due battute non saranno dette.

4.7.3 Reattività perfonalizzata

Il beat goal "TripSuggest" che stiamo scrivendo è il secondo beat goal in questo beat. Cosa succederebbe se in precedenza, durante il primo beat goal, quello di transizione (transition-in), il giocatore ha chiesto un drink specifico: ad esempio, ha detto: "Vorrei una birra "? Dovremmo avere delle variazioni nel dialogo "TripSuggest" per reagire a.

"TripSuggest"

<set un interruptible>

<if reestablish> T: (allegro) Dunque! Un drink!!

<if nothing suggested so far>

T: (allegro) Cosa gradisci?

T: (allegro) Che ne diresti di un Martini?

<set interruptible>

<set gist point>

T: (vantarsi) Sono un vero esperto in questo, almeno e quello che mi dicono tutti.

<if a fancy drink was just requested>

T: (eccitato) Sicuro, ottimo, prendo anche io lo stesso!

<if a boring drink was just requested>

T: (un po' deluso) Oh, ma cerchiamo di divertirci stasera!

G: (un po' infastidita) Oh Dio, Trip, per favore ... Non strafare con la preparazione drink.

4.7.4 Mappare le espressioni del giocatore / Azioni per alcune reazioni

Come descritto in precedenza per reagire al giocatore nel nostro beat goal "TripSuggest", raggrupperemo insieme atti discorsivi simili, e li mapperemo in un ridotto set di reazioni. Qui riconduciamo molti degli atti discorsivi nella tabella 30.1 a solo cinque tipi di reazione:

- AgreeTrip: è d'accordo ("sì"), esclamazione positiva ("che caro"), grazie ("grazie"), esprimere felicità ("questo mi rende felice"), abbraccio.
- DisagreeTrip: non sono d'accordo ("No"), esclamazione negativa ("Che sfigato"), esprimono rabbia ("Sono incazzato"), di esprimere triste ("io sono improvvisamente depresso").
- SpecificFancyRequest: riferimento a <fancyDrink> ("che ne dici di un Cosmopolitan?", o "hai per caso uno Scotch lì?").
- SpecificBoringRequest: riferimento a <nonFancyDrink> ("solo acqua per me," o "un Coors Lite, ce l'hai").
- NonAnswer: timeout (molti secondi di silenzio dal giocatore), riferimento a drink ("un drink mi suona bene"), ridere ("ah ah"), forse ("Credo di sì"), non capire ("Non so cosa sia un Martini"), scusa ("sorry"), d'accordo, ecc. Verso Grace ("Grace, sicuro", "grazie Grace", ecc.)

Si noti che i rimanenti atti discorsivi dalla tabella 30.1 che non sono gestiti da questa mappatura locale saranno comunque gestiti dal contesto globale. Ad esempio, se il giocatore fa un riferimento al sesso durante TripSuggest, dato che nulla nella mappatura di TripSuggest riesce a gestire un simile riferimento, ciò suggerisce una non reazione. Il contesto globale, d'altro canto, suggerisce una reazione global mix-in, che, non essendoci alcun contesto specifico che possa suggerire una particolare reazione, sarà quella selezionata ed eseguita. Il global mix-in verrà inserito tra TripSuggest e GraceCounterSuggest; GraceCounterSuggest dunque sarà eseguito preceduto dalla sua battuta di ristabilimento dell'argomento. (Si noti che qualche altro contesto in questo o in un altro beat potrebbe avere effettivamente un beat di risposta specifica all'argomento "sesso", per cui non ci sarebbe bisogno dell'intervento di un global mix-in.)

I tipi di reazione qui elencati sono implementati come variazioni di dialogo all'inizio del prossimo beat goal, "GraceCounterSuggest":

"GraceCounterSuggest"

<set uninterruptible>

<if reestablish> T: A proposito di quei drink

<if AgreeTrip> T: (eccitato) Buona scelta!

<if DisagreeTrip> T: (sgomento) Che cosa? Oh, ho pensato che ti sarebbe piaciuto...

<if SpecificFancyRequest> T: (eccitato) Oh yeah, questo è ancora meglio!

<if SpecificBoringRequest> T: (deluso) Oh, ma cerchiamo di festeggiare stasera ...!

<if NonAnswer> T: (nervoso) Uh, beh, la sai una cosa, adesso io ti vado a fare un Martini.

<if current drink suggestion is fancy> G: No. No, *Player*, forse hai voglia di un succo, o acqua minerale?

<if current drink suggestion is not fancy, e.g., a beer>

G: Trip, non tutti condividono tutti la tua infatuazione per i cocktail.

G: *giocatore*, preferisci solo una birra, giusto?

<set interruptible>

<set gist point>

T: (sbigottito, con il respiro) Oh, ma dai...

4.7.5 Performance fisica

Finora, il nostro esempio di scrittura di beat goal si è concentrato sulla creazione e sulla logica del dialogo. Oltre a questo, come accennato in precedenza, l'autore deve specificare le performance fisiche. Ciò include: decidere dove debbano collocarsi i personaggi in relazione alla posizione del giocatore (messa in scena); quanto vicini debbano essere al giocatore (spesso in relazione al criterio di affinità); i cambiamenti di umore (che influenzano l'espressione del viso e la posizione del corpo); gli eventuali gesti dei personaggi, e come possano essere coordinati con il dialogo; l'espressione facciale di base e le espressioni momentanee (shock, sorpresa, ecc.); e così via. Oltre a partecipare alla logica del dialogo, ogni JDB precisa le direzioni procedurali su come il personaggio debba rendere le sue battute linee specifiche. Ciò che segue è un esempio di codice ABL per un unico JDB: in questo caso il JDB per la "TripSuggest", dove il giocatore non ha effettuato alcuna richiesta di drink fantasiosi.

```

// ## if no specific fancy drink request (but includes if we
had gotten a specific non fancy request)
joint parallel behavior
bFAskDrinkT1NTPA_TripSuggest_BodyStuff() { teammates Trip
Grace; precondition { StoryMemory
(BeatFAskDrinkT1WME bGotFancySpecificRequest == false
curDrinkIdea :: drink) }
    with (ignore_failure, property isStagingGoal true)
        spawngoal StagingConverse (40, player,
eConverseType_offCenterShared, eWalkType_normal);
    with (persistent, team_effect_only)
        subgoal TryToKeepFacingSprite(0, player);
    with (priority_modifier 1, ignore_failure,
team_effect_only)
        subgoal SetPerformanceInfo(40, 2,
eHeadEmphType_nodStrong,
    eArmsEmphType_atSide, startWith, -1, GazeType_normal,
player,
    eFEBase_serious);
    with (priority_modifier 2)
        subgoal SetMood(eMood_happy,
eMoodStrength_barely, -1, 0);
    subgoal
bFAskDrinkT1NTPA_TripSuggest_BodyStuff_NoSpReq_seq(drink);
        subgoal SetLetBeatGoalFinishFlag(true);
}
sequential behavior
bFAskDrinkT1NTPA_TripSuggest_BodyStuff_NoSpReq_seq(int drink)
{int doOptionalPreface;// optional prefacemantal_act {
doOptionalPreface = randGen.nextInt(2); }
    with (ignore_failure)
        subgoal

bFAskDrinkT1NTPA_TripSuggest_BodyStuff_NoSpReq_dial
        (doOptionalPreface);// possible second
preface
    with (ignore_failure)
        subgoal
bFAskDrinkT1NTPA_TripSuggest_BodyStuff_NoSpReq_seq2(drink);
        // the drinksuggestion
    with (ignore_failure)
        subgoal DoFullExpressionBase(70,
eFEBase_pleasant);
    with (ignore_failure)
        subgoal
bFAskDrinkT1NTPA_TripSuggest_BodyStuff_NoSpReq_dia3(drink);
}

```

Lo scopo di mostrare un frammento di codice non è quello di analizzare il codice nei minimi dettagli, ma piuttosto quello di sottolineare alcune caratteristiche del codice ABL per JDBs:

- La prima cosa da notare è che si tratta di codice. Non si tratta di una struttura statica; è, invece, una piccola macchina dinamica che sa come eseguire delle battute particolari; le può eseguire in qualsiasi posto nella stanza, anche se il giocatore si aggira nello spazio. Può eseguire quelle battute anche se il giocatore è impegnato in altri comportamenti fisici a lungo termine (ad esempio, Trip gira con la sua sfera della saggezza), che potrebbero, quindi, richiedere la sostituzione o la soppressione di alcuni movimenti fisici. Questo non è una clip o una rappresentazione pre-scritta, ma è piuttosto un comportamento che regola dinamicamente la performance.
- Il comportamento comune parallelo (questo è quello per Trip) si sincronizza automaticamente con un comportamento associato in Grace, permettendo loro di coordinare strettamente le loro prestazioni, anche se ciascuno di loro sono simultaneamente impegnati in comportamenti paralleli e non sincronizzati. (Non mostriamo qui la parte che riguarda Grace, di questi comportamenti comuni)
- Con il comportamento parallelo, l'autore specifica un gruppo di azioni che dovrebbe avvenire nello stesso momento. Nel nostro caso specifico ciò include: il personaggio inizia la messa in scena al centro dello spazio così come lo vede il giocatore e condivide questa posizione con un altro personaggio, tenta di mantenersi di fronte al giocatore mentre questi va in giro; i personaggi recitano le loro battute utilizzando gesti del capo e delle braccia per dare enfasi al dialogo; sono di un umore mediamente felice (ciò si combinerà con una espressione facciale di base seria), recitano una certa sequenza di battute che inizia non interrompibile (la non interrompibilità sarà poi disattivata quando l'essenza sarà raggiunta - e ciò avviene nel dettaglio dei comportamenti di dialogo che noi non mostriamo qui)

A livello di beat goal, la scrittura di *Façade* combina le qualità sia di uno scrittore, sia di un regista, in cui la logica del dialogo e le modalità di prestazione sono espresse in modo procedurale.

4.7.6 Dialogo Variazione di tensione di livello, il giocatore Affinity, e Alternate-Dialogo

Per completare il nostro esempio di scrittura, abbiamo bisogno di elencare interamente il dialogo delle restanti delle permutazioni: i due livelli di tensione (basso, medio) e le tre affinità del giocatore (neutro, Trip, Grace). Ciò va fatto per i due beat goal ("TripSuggest" e "GraceCounterSuggest"), comprendendo la variazione di dialogo alternato all'interno di ogni dialogo.

Come descritto in precedenza, l'affinità neutra e l'affinità con Trip sono stati combinate in una sola, riducendo così il numero totale di permutazioni per questo beat da $2 \times 3 = 6$, a $2 \times 2 = 4$: TensionLow-AffinityNeutralTrip, TensionLow-AffinityGrace, TensionMedium - AffinityNeutralTrip, e TensionMedium-AffinityGrace.

5. Valutare *Façade*

In questa sezione, cerchiamo di descrivere il grado di agency raggiunto in *Façade*, così come i fallimenti e i successi in termini di design, interfaccia e l'architettura del sistema.

5.1 Descrivere l'Agency

La creazione di agency per il giocatore è stato il nostro obiettivo primario nella progettazione di *Façade*, reso possibile dal nostro approccio di scrittura fortemente procedurale.

5.1.1 Agency locale

Quando le azioni del giocatore causano reazioni immediate, relative al contesto e significative per l'intero sistema, noi le indichiamo come agency locale. Inoltre, maggiore è la gamma di azioni che il giocatore può fare - cioè, quanto più espressivo è l'interfaccia, - allora tanto è più ricca l'agency locale (di nuovo, se le risposte sono significative).

Façade offre ai giocatori un'interfaccia aperta, continua e in linguaggio naturale, così come azioni fisiche e gesti quali l'esplorazione, prendere oggetti, abbracciare e baciare. I milioni potenziali input del giocatore, utilizzando centinaia di regole del soprannominato NL, vengono mappati su una o più dei circa 30 atti discorsivi parametrizzati (DA), quali: lode, esclamazione, riferirsi a un argomento, e dare spiegazioni. Un secondo gruppo di norme chiamato "proponenti di reazione" interpreta questi atti discorsivi (DA) in un modo specifico al contesto, come, ad esempio, accordo, disaccordo, alleanza, o provocazione.

Idealmente, ci sarebbero risposte immediate, significative, contestualizzate disponibili in qualsiasi momento per tutti gli atti discorsivi. Nell'attuale stato di implementazione di *Façade*, a nostro avviso, questo ideale si raggiunge circa

il 25% delle volte, laddove il giocatore ha un grado soddisfacente di controllo in tempo reale sullo stato emotivo di Trip e Grace, sulle loro affinità, sugli argomenti che si stanno discutendo, su quali informazioni vengono rivelate, e sull'attuale livello di tensione. Ma più spesso, circa il 40% delle volte, si raggiunge solo un ideale parziale. La mappatura / interpretazione dagli atti discorsivi alle reazioni è più grossolana, e le risposte sono più generiche e / o comunque non immediate. Inoltre, circa il 25% delle volte si riscontra una reattività meno profonda, e circa il 10% delle volte vi è poca o nessuna reattività. Questi diversi livelli di agency locale sono talvolta raggruppati in cluster temporali, ma hanno anche la possibilità di variare momento per momento.

Ci sono due ragioni principali per questi diversi livelli di agency locale. In primo luogo, dal punto di vista di design, in certi momenti dell'esperienza complessiva, si rende necessario canalizzare le possibili direzioni della narrazione in direzioni preferite da un punto di vista autoriale, al fine di garantire il ritmo drammatico e la progressione. In secondo luogo, e accade più spesso, la mancanza di un'agency locale è causata dal modo in cui il contenuto narrativo è stato scritto (vedere la sezione fallimenti, più avanti).

5.1.2 Global Agency

Il giocatore ha global agency, quando la forma globale dell'esperienza è determinata dalle azioni del giocatore. In *Façade*, questo significherebbe che la fine della storia, e i particolari dell'arco narrativo che portano a quel finale, sono determinati in modo regolare e continuo da quello che il giocatore fa; e che alla fine dell'esperienza, il giocatore ha capito come le sue azioni hanno determinato quello specifico svolgimento.

Façade tenta di raggiungere un'agency globale in diversi modi. In primo luogo, il sequenziamento dei beat (vale a dire, di alto livello narrativo) può essere influenzato dagli argomenti ai quali il giocatore fa riferimento; la sequenza può variare entro il numero di permutazioni permesso dalle precondizioni dei beat, e dai requisiti dell'arco drammatico di tensione. Anche con solo ventisette beat, tecnicamente ci sono migliaia di diverse possibili sequenze di beat. Tuttavia, dal momento che la maggior parte dei beat sono causalmente indipendenti, il numero di sequenze di beat significative non è elevato.

Più significative delle variazioni di sequenze di beat (il "che cosa" accade), sono le variazioni all'interno dei beat e delle progressioni global mix-in (il "come" accade). All'interno della affinità, degli *hot-button*, e dei giochi di terapia, nel corso dell'interazione, sono possibili una varietà di modelli e di dinamiche. Infatti, questi modelli sono monitorati dal sistema e rimarcati nella

ricapitolazione drammatica del beat BlowupCrisis a metà del dramma , e anche nel beat RevelationsBuildup al culmine del dramma. Un calcolo del “punteggio” finale delle varie interazioni sociali viene utilizzato per determinare quale dei cinque beat finali deve essere inserito nella sequenza. In questi beat finali possiamo vedere Grace Trip che rivelano uno o più importanti segreti e quindi decidere di rompere e lasciarsi, oppure accorgerci che entrambi hanno troppa paura di fare qualcosa, o ancora che mediante un processo di consapevolezza su sé e su di loro come coppia, decidono di restare insieme.

5.2 Fallimenti e successi di *Façade*

In questa sezione cerchiamo di valutare i nostri risultati nella creazione del dramma interattivo *Façade*, i cui obiettivi progettuali sono stati fortemente plasmati dal nostro approccio procedurale centrato sui contenuti in fase di implementazione.

5.2.1 Agency

Durante la produzione di *Façade*, all’interno del nostro “limitato” sforzo di authoring (al di là della costruzione dell’architettura, *Façade* ha richiesto circa 3 anni-uomo di solo authoring, che è più di quanto richieda un tipico progetto di ricerca o realizzazione artistica, ma molto meno di quanto è necessario in un tipico progetto di videogioco commerciale). Abbiamo fatto un compromesso per sostenere un significativo grado di un’agency locale, a scapito della agency globale. Tenendo presente che il tempo necessario per progettare e scrivere le JDBs è notevole, abbiamo creato solo ventisette beat. Di conseguenza abbiamo raggiunto un grado di agency globale di gran lunga inferiore a quanto inizialmente sperato. Come risultato, riteniamo di non aver sfruttato appieno la potenza del drama manager.

Inoltre, poiché la specifica di ogni Dialogo comportamento comune (JDB) - il dialogo parlato, le indicazioni sceniche, l’emozione, e la performance gestuale - richiede una grande quantità di authoring e non viene generato automaticamente da un qualche strumento di più alto livello per la creazione di comportamenti, ci siamo dovuti limitare alle permutazioni fatte a mano, di contenuti mixabili. *Façade* non genera le frasi - anche se genera sequenze.

5.2.2 Feedback

Una delle maggiori sfide che abbiamo incontrato, e in cui crediamo *Façade* mostri le sue insufficienze, è di comunicare sempre chiaramente lo stato

dell'interazione sociale al giocatore. Nei giochi tradizionali, è semplice dire ai giocatori lo stato della gara: si può visualizzare un punteggio numerico, o mostrare fisicamente il personaggio in un livello superiore, o visualizzare l'attuale configurazione dei pezzi del gioco. Ma quando il "gioco" sta avvenendo apparentemente nelle teste dei personaggi, e se intendiamo mantenere un'estetica teatrale e performativa (e non mostrare i sentimenti interni attraverso statistiche e barre di scorrimento, come in *The Sims*), allora la sfida è difficile. Secondo la nostra valutazione, *Façade* riesce a comunicare meglio lo stato della semplice affinità e dei *giochi hot-button* più che il più complesso gioco di terapia.

5.2.3 Interfaccia

Un'altra grande sfida è gestire le aspettative del giocatore, sollecitate dall'interfaccia *open-ended* in linguaggio naturale. Abbiamo anticipato i fallimenti nella comprensione del linguaggio naturale, che, allo stato attuale della valutazione informale di *Façade*, si verificano in media circa il 30% delle volte. Questo compromesso è stato intenzionale, dato che abbiamo voluto capire meglio i nuovi piaceri che linguaggio naturale è in grado di offrire quando riesce nel suo scopo; il che in *Façade* accade circa il 70% delle volte, parzialmente o completamente.

5.2.4 Architettura del sistema

Nella nostra stima, un successo di *Façade* è l'integrazione dei *beat goal /mix-in*, *global mix-in* e del sequenziatore narrativo denominato *drama manager*, con un'interfaccia espressiva in linguaggio naturale, la comprensione del linguaggio naturale in un contesto specifico, e l'animazione in tempo reale di personaggi espressivi. Crediamo che l'effetto complessivo rappresenti un qualche progresso verso i nostri originali obiettivi di progettazione, che prevedevano la creazione di un senso di immediatezza, la presenza e vitalità nei personaggi necessari per il dramma teatrale.

Come si evince dal nostro esempio di *authoring*, vi è ancora significativo sforzo di *authoring* di un dramma interattivo all'interno della nostra architettura. La nostra architettura ora rende possibile la scrittura di teatro interattivo, ma non necessariamente facile (era estremamente ingombrante o impossibile attraverso le tradizionali macchine a stati finiti, alberi di dialogo, e gli approcci delle storie ad albero).

Non è chiaro se mai ci saranno gli strumenti per la creazione di drammi interattivi che non siano di programmazione. In ogni caso noi crediamo che ciò richieda fundamentalmente un'autorialità procedurale. Tuttavia, i linguaggi che abbiamo sviluppato per strutturare il dialogo e l'utilizzo di ABL,

all'interno di *Façade*, possono servire come una specifica per la creazione di uno strumento di livello più elevato che faciliti l'autoring di altre esperienze dello stesso tipo. Anche durante la creazione di *Façade*, siamo stati in grado di formalizzare la struttura generale del beat in un modello di codice ABL che si potrebbe copiare e modificare per la creazione di nuovi beat. Un ovvio passo successivo è quello di far entrare questi linguaggi come strutture di prima classe nel linguaggio ABL, o forse in un linguaggio di livello superiore che si trovi sulla parte superiore di ABL.

In generale, il nostro approccio per l'architettura di sistemi di dramma interattivi non è quello di costruire uno strumento generico, un *one-size-fits-all*, che cerchi di nascondere la natura fondamentale procedurale del mezzo, ma piuttosto di creare nuovi linguaggi per l'autorialità procedurale, costruire nuove esperienze con quei linguaggi, e spingere le risorse computazionali e le cose imparate dalla scrittura di precedenti esperienze in costrutti di prima classe per le lingue del futuro.

5.2.5 Design

Alcuni aspetti del progetto del nostro dramma, aiutano a fare di *Façade* una esperienza interattiva piacevole, mentre altri aspetti no. È meglio avere due personaggi non giocatori strettamente coordinati i quali possono mantenere credibile l'azione drammatica mentre accade, nel caso in cui il giocatore smetta di interagire o si comporti in modo da non cooperare. In realtà, il rapido ritmo del dialogo nella performance di Grace e Trip, scoraggia l'immissione di lunghe battute da parte del giocatore. Da progetto, Grace e Trip sono egocentrici. Ciò gli permette, occasionalmente, di ignorare, senza perdere la credibilità, le azioni del giocatore non riconosciute o non gestibili. Create di una narrativa libera e scarsamente unitaria, concedere una maggiore agency locale, ma a condizione di una minore agency globale. Tuttavia, la ricchezza nella variazione dei contenuti, e (almeno) un moderato grado di agency globale, incoraggiano il replay.

L'enorme campo del dramma, un matrimonio che sta cadendo a pezzi, probabilmente inficia il successo dell'esperienza complessiva, in quanto incoraggia troppo le aspettative dei giocatori riguardo l'intelligenza dei personaggi, la complessità psicologica, e le competenze linguistiche. Come previsto, il sistema non riesce a capire, né possiede reazioni scritte, per molte delle ragionevoli espressioni del giocatore. Il dominio di grandi dimensioni spesso richiede potenzialmente la mappatura di milioni di testi su pochi atti discorsivi, il che può risultare impacciato o troppo grezzo per il giocatore. Inoltre, la continua interazione in tempo reale, contro l'interazione discreta (turn-taking) e / o in tempo non reale, ha aggiunto una grande complessità e di scrittura.

6. Conclusione

In questo capitolo, abbiamo sostenuto che l'autorialità procedurale è necessaria per sfruttare appieno le possibilità di rappresentazione del computer come mezzo espressivo. La proceduralità è un supporto di base per tutta la creazione digitale. Mentre l'alfabetizzazione procedurale non è necessaria per creare lavori digitali, i nuovi professionisti dei media, senza questa alfabetizzazione procedurale si limitano alla produzione di quelle opere interattive che possono essere prodotte entro i limiti posti dagli attuali strumenti di authoring. Abbiamo creato un caso sull'importanza dell'autorialità procedurale, descrivendo gli obiettivi di progettazione di un caso di studio, il dramma interattivo *Façade*. Abbiamo spiegato come questi obiettivi possano essere raggiunti solo attraverso un approccio altamente procedurale alla narrativa interattiva. Sulla base della nostra esperienza, sia per l'architettura sia per l'authoring di *Façade*, abbiamo scoperto che l'autorialità procedurale è essenziale per consentire un genere di arte interattiva e di intrattenimento ancora da venire.

Bibliografia

- Albee, Edward (1962). *Who's Afraid of Virginia Woolf?* New York: Signet.
- Bates, Joseph (1992). "Virtual Reality, Art, and Entertainment." *Presence: The Journal of Teleoperators and Virtual Environments* 1 (1992): 133-;138.
- Berne, Eric (1964). *Games People Play*. New York: Grove Press.
- Crawford, Chris (1987). "Process Intensity." *The Journal of Computer Game Development* 2 (1987).
- Crawford, Chris (1989). "Indirection." *The Journal of Computer Game Design* 3 (1989).
- Laurel, Brenda (1991). *Computers as Theatre*. Reading, MA: Addison-Wesley.
- Loyall, Bryan (1997). *Believable Agents*. Ph.D. Thesis. Tech report CMU-CS-97-123. Carnegie Mellon University.
- Mateas, Michael (2005). "Fever-Addled Impressions of GDC." Grand Text Auto.
- Mateas, Michael, and Andrew Stern (2000). "Towards Integrating Plot and Character for Interactive Drama." Working Notes of the Social Intelligent Agents: The Human in the Loop Symposium. AAAI Fall Symposium Series. Menlo Park, CA: AAAI Press.

--- (2003a). "Façade: An Experiment in Building a Fully-Realized Interactive Drama." Game Developers Conference (GDC) 2003. San Jose, CA, USA.

--- (2003b). "Integrating Plot, Character and Natural Language Processing in the Interactive Drama Façade." Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) 2003. Darmstadt, Germany.

--- (2004a). "A Behavior Language: Joint Action and Behavioral Idioms." *Lifelike Characters: Tools, Affective Functions and Applications*, edited by Helmut Predinger and Mitsuru Ishioka. Berlin: Springer.

--- (2004b). "Natural Language Understanding in Façade: Surface Text Processing." Proceedings of Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) 2004. Darmstadt, Germany.

McCorduck, Pamela (1991). *Aaron's Code: Meta-art, Artificial Intelligence, and the Work of Harold Cohen*. New York: W. H. Freeman and Co.

McKee, Robert (1997). *Story: Substance, Structure, Style, and the Principles of Screenwriting*. New York: HarperCollins.

Montfort, Nick (2005). "Finally, the Curtain Opens on Façade." *Grand Text Auto*.

Murray, Janet (1998). *Hamlet on the Holodeck*. Cambridge, MA: MIT Press.

Taylor, Alice (2005). "Burn the House Down." *Wonderland*.

Wikipedia (2005). "Demoscene."

References: Games

Façade, a One-Act Interactive Drama. Michael Mateas and Andrew Stern; Procedural Arts. 2005.