

JOSÉ DANIEL RAMOS WEY

INTERFACE

Um Sistema de Animação Interativa
de Rostos Humanos

Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do título
de Mestre em Engenharia.

São Paulo
1999

JOSÉ DANIEL RAMOS WEY

INTERFACE

Um sistema de Animação Interativa
de Rostos Humanos

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de Mestre em
Engenharia.

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Marcelo Knorich Zuffo

São Paulo - 1999

Wey, José Daniel Ramos

InterFace: Um Sistema de Animação Interativa de Rostos Humanos. São Paulo, 1999, 112p.

Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo.
Departamento de Engenharia Eletrônica

1. Computação Gráfica 2. Animação Interativa. I. Universidade de São Paulo.
Escola Politécnica. Departamento de Engenharia Eletrônica.

À Deus,
à minha esposa Raquel
e aos meus pais

AGRADECIMENTOS:

A Deus, por todas as graças recebidas na minha vida.

À minha esposa Raquel, por todo o apoio, carinho e incentivo.

Aos meus pais Luis Otávio e Maria Celeste, e aos meus irmãos André, Bia e Malu, pela força e união.

Aos meus sogros Orlando e Benedita, e ao Orlandinho, pelo carinho e incentivo.

Ao meu orientador, Prof. Marcelo Knörich Zuffo, pelo apoio e compreensão no desenvolvimento deste trabalho.

Ao Prof. Dr. Ken Perlin, meus agradecimentos pelo incentivo e apoio essenciais para a conclusão deste trabalho.

A todos os amigos do projeto Aria, que tanto contribuíram para que o desenvolvimento do sistema InterFace.

Resumo

A reprodução de rostos humanos é um enorme campo de estudo para artistas e pesquisadores. Os seres humanos possuem uma grande familiaridade com o rosto, sendo capazes de identificar uma única face entre milhares de outras e de perceber sutilezas que são difíceis de reproduzir. Para os pesquisadores de computação gráfica, a criação de rostos humanos em três dimensões que sejam convincentes é um grande desafio.

Com o advento de métodos para a síntese de imagens por computador de alta qualidade (*rendering*), a criação e animação de expressões faciais de maneira realista tornou-se uma necessidade. Sistemas interativos de computação gráfica como a realidade virtual também têm no uso de expressões faciais um grande aliado para a criação de personagens virtuais mais naturais.

Nosso trabalho visa explorar novas técnicas de criação de expressões faciais emocionalmente significativas, e métodos de animação entre as expressões que permitem um alto grau de realismo. Afim de explorar a fundo estas técnicas, criamos um sistema interativo de animação facial, denominado **InterFace**. O sistema, implementado nas linguagens Java e VRML (*Virtual Reality Modeling Language*), permite a criação de diferentes expressões faciais de maneira simples e interativa. O sistema também possibilita a animação entre estas expressões utilizando funções matemáticas como seno, cosseno, *splines* e ruído simulado e a junção de diferentes animações utilizando uma abordagem baseada em técnicas de composição digital de imagens.

Abstract

The reproduction of human faces is a large research field. Humans have a remarkable familiarity with facial expressions, been capable of recognize one face among thousands and being able to detect very subtle changes in facial expressions, which is hard to reproduce. For computer graphics researchers, the creation of believable three dimension human faces has been a challenge for many years.

With the advent of new techniques for rendering high quality images in computers, the creation and animation of believable facial expressions became a major issue. Interactive techniques in Computer graphics, like virtual reality, also incorporated facial expressions to human models in order to provide more realism.

Our work explore new techniques to the creation of realistic facial expressions that are emotionally meaningful. In order to evaluate these techniques, we created an interactive facial animation system, called **InterFace**. The system is implemented with Java and VRML (Virtual Reality Modeling Language) languages. **InterFace** allows the creation of facial expressions easily and interactively, and also provide animation among expressions using mathematical functions like sine, cosine, splines and simulated noise. To compose the animation, we adopted a layered approach that is similar to techniques used in digital image composition.

Sumário

Capítulo I - Introdução.....	1
1.1. As Expressões Faciais.....	1
1.2. Motivações.....	2
1.3. Relevância.....	3
1.3.1. Realidade Virtual.....	4
1.3.2. Agentes de Interface Gráfica.....	4
1.3.3. Minimização na transferência de dados.....	5
1.3.4. Atores Virtuais.....	5
1.4. Objetivo.....	6
1.4.1. O Sistema InterFace.....	6
1.5. Estrutura da Dissertação.....	7
Capítulo II - Histórico de Técnicas de Animação Facial em Computação Gráfica.....	9
2.1. Primeiras pesquisas em rostos humanos.....	9
2.2. Primeiras Pesquisas de Animação Facial em Computador.....	10
2.2.1. O Trabalho Pioneiro de Frederick Parke.....	10
2.2.2. A Parametrização do rosto.....	12
2.2.3. Modelamento de Expressões Faciais por Músculos.....	13
2.2.4. Animação Facial.....	14
2.3. O Estado da Arte em Animação Facial.....	17
2.4. O Sistema InterFace.....	17
2.5. Conclusão.....	18
Capítulo III - Conceitos do Sistema InterFace.....	19
3.1. Visão Geral.....	19
3.2. Módulo de Criação de Expressões.....	21
3.2.1. A Biblioteca Básica de Expressões.....	22
3.2.2. Combinação de Expressões.....	26
3.3. Módulo de Animação Facial.....	28
3.3.1. Ações.....	29
3.3.2. Grupos de Ações.....	30
3.4. Inteligência do Ator Virtual.....	34
3.5. Conclusão.....	35
Capítulo IV - Implementação do Sistema InterFace.....	36
4.1. Requisitos do Sistema.....	36
4.2. Decisões de Projeto.....	37
4.2.1. Representação da Geometria.....	37
4.2.2. Linguagens de Programação.....	38
4.2.3. Projeto do Sistema Interface.....	41
4.3. Módulo de Criação de Expressões.....	42
4.3.1. Descrição do Ator Virtual em Três Dimensões.....	42
4.3.2. Descrição de Expressões da Biblioteca Básica.....	43
4.3.2.1. As Micro-Expressões.....	44
4.3.2.2. As Expressões da Biblioteca Básica.....	47
4.3.3. Criação de Expressões.....	47
4.3.4. Salvando as Expressões.....	49
4.4. Modulo de Animação Facial.....	50
4.4.1. Ações.....	51

4.4.1.1. Curvas	53
4.4.1.2. Modificadores de Curvas	54
4.4.1.3. Envelopes	55
4.4.1.4. Exemplo	57
4.4.2. Grupos de Ações	59
4.4.2.1. Ordem e Transparência	60
4.5. Composição de Eventos	61
4.6. A Inteligência do Ator Virtual	62
4.6.1. O Usuário como Inteligência do Ator Virtual	62
4.6.2. Conexão Com Outros Sistemas.....	63
4.7. Animações em VRML	64
4.8. Conclusões	65
Capítulo V - Análise De Resultados	66
5.1. Metodologia Para Análise Dos Resultados	66
5.1.1. Análise Conceitual	66
5.1.2. Análise da Implementação do Sistema.....	67
5.2. Análise do Módulo de Criação de Expressões.....	67
5.2.1. Análise Conceitual	67
5.2.2. Análise do Sistema.....	69
5.3. Análise do Módulo de Animação Facial.....	72
5.3.1. Análise Conceitual	72
5.3.2. Análise da Implementação	74
5.4. Conclusão.....	77
Capítulo VI - Conclusões e Projetos Futuros.....	78
6.1. Resultados do Sistema InterFace	78
6.1.1. Contribuições Relevantes.....	78
6.1.1.1. Biblioteca Básica de Expressões	79
6.1.1.2. Extrapolação de Expressões	79
6.1.1.3. Ações e Grupos de Ações.....	80
6.1.2. Implementação do Sistema	81
6.1.3. O Projeto Aria	81
6.1.4. Publicações.....	84
6.2. Trabalhos Futuros e Áreas de Pesquisa.....	85
6.2.1. Aprimoramentos ao Sistema InterFace	85
6.2.2. Projetos de Pesquisa Futuros Relacionados ao Sistema InterFace.....	87
6.3. Considerações Finais.....	88
Bibliografia 89	
Bibliografia Adicional Recomendada.....	92
Sítios Web Recomendados	92
Apêndice I - Sintaxe dos Arquivos do Sistema InterFace	93
I.1. Nós de Controle da Cena VRML.....	93
I.2. Arquivo de Descrição de Ações.....	94
Apêndice II - Sintaxe do Protocolo de Comunicação InterFace Protocol (IFP).....	97

Índice de Figuras

Figura 2.1: Exemplo de animação por interpolação linear	11
Figura 3.1: Modelo abstrato de três camadas do sistema InterFace	20
Figura 3.2: Exemplo de Soma de Expressões.....	27
Figura 3.3: Exemplo de Extrapolação	28
Figura 4.1: Diagrama esquemático das expressões da Biblioteca Básica	44
Figura 4.2: Expressões criadas no módulo de Criação de Expressões	49
Figura 4.3: Diagrama esquemático do módulo de Animação Facial.....	51
Figura 4.4: Curva de Ruído não determinístico de Perlin.	54
Figura 4.5: Envelope que envolve uma micro-ação	56
Figura 4.6: Envelope que envolve uma ação	57
Figura 4.7: Execução da ação “espirrar”	59
Figura 4.8: Interface para escolha de ações	63
Figura 5.1: Composição da expressão facial “assobio”	69
Figura 5.2: A interface com o usuário do módulo de Criação de Expressões	70
Figura 5.3: A interface com o usuário do módulo de Animação Facial	75
Figura 6.1: A primeira versão do projeto Aria	82
Figura 6.2: O projeto Aria II.....	83
Figura 6.3: O projeto Aria III	84

Índice de Tabelas

Tabela 2.1: Equação da interpolação linear de vértices.....	11
Tabela 3.1: A Biblioteca Básica de Expressões	25
Tabela 3.2: Pseudo-código da composição dos Grupos de Ação.....	32
Tabela 4.1: Código VRML para a descrição do ator virtual Giggio	46
Tabela 4.2: Código exemplo de uma ação.....	52
Tabela 4.3: Curvas disponíveis no sistema InterFace.....	53
Tabela 4.4: Exemplo do modificador de curva Trigged_by.....	55
Tabela 4.5: Código da ação Espirrar	57
Tabela I.1: Nós de controle em VRML do sistema InterFace	94
Tabela I.2: Sintaxe na descrição de ações, Grupos de Ações e Micro-ações	95
Tabela I.3: Exemplo de um arquivo de descrição de ações	96

Glossário de Termos

AMA	<i>Abstract Muscle Actions</i> (ações abstratas de músculos). Ver [Tha88].
CGI	<i>Common Gateway Interface</i> (Interface comum para saída de dados): é uma interface padrão para execução de programas em um servidor <i>Web</i> .
EAI	<i>External Authoring Interface</i> (Interface de Autoria Externa): Norma ISO para permitir a comunicação entre uma cena em VRML e outros aplicativos
FACS	<i>Facial Animation Coding System</i> (sistema de codificação de animação facial). Ver [Par96].
GUI	<i>Graphical User Interface</i> (Interface gráfica com o usuário): é o modo como o usuário interage com o computador através de recursos gráficos.
Navegador	Aplicativo que permite o acesso às informações da <i>World Wide Web</i> ou navegar em ambientes de três dimensões criados em VRML.
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i> (Protocolo de Controle de Transmissão / Protocolo Internet): O protocolo de comunicações utilizado na rede Internet.
UA	Unidades de Ação. Ver [Par96].
VRML	<i>Virtual Reality Modeling Language</i> (linguagem de modelamento para realidade virtual): É uma linguagem utilizada para a descrição de cenas em três dimensões.
VVIEBB	Vetor de Valores de Intensidade para Expressões da Biblioteca Básica
WWW	<i>World Wide Web</i> (Teia de alcance mundial): é uma interface padrão para acesso a informações em forma de hipertexto através da rede Internet.

Capítulo I

Introdução

Os seres humanos utilizam diversos meios para se comunicar. Quando duas pessoas estão conversando, elas trocam informações basicamente através de palavras. Porém, outros meios de troca de informações também são utilizados de forma intuitiva, como a movimentação das mãos, a entonação da voz e as expressões do rosto. Mesmo em uma conversa telefônica, a entonação da voz tem uma grande importância sobre a mensagem que está sendo transmitida. Uma mesma frase dita com cinismo é entendida de maneira diferente que quando dita de maneira mais formal. Em meios onde apenas as palavras podem ser usadas para transmitir informações, foram criados outros meios para dar “entonação” à mensagem. Um exemplo disto é o uso de “*emoticons*” (sinais que indicam sorriso ou expressões de tristeza) em mensagens enviadas por correio eletrônico. A esta forma de transmissão de informação (entonação de voz, expressões faciais, movimentação das mãos e do corpo e assim por diante) damos o nome de **linguagem corporal**.

Nesta dissertação de mestrado utilizaremos técnicas de computação gráfica e animação computadorizada para a incorporação de linguagem corporal, particularmente expressões faciais, a um ator virtual.

1.1. As Expressões Faciais

Um dos meios básicos de linguagem corporal é o uso de expressões faciais. De fato, o rosto humano transmite múltiplas informações sobre uma pessoa simultaneamente,

tais como emoção, humor, atitude, idade, sexo e raça [Tha94]. Estas informações tem um caráter universal, ou seja, mesmo em diferentes culturas e em diferentes épocas, expressões faciais que explicitam tristeza, alegria, dor e outras expressões sempre foram as mesmas [Dar72].

Os seres humanos possuem uma familiaridade natural com o rosto. Esta familiarização inicia-se desde o momento em que nascemos, e é aprimorada ao longo de toda a vida. Somos capazes de identificar uma única face entre milhares de outras. Pequenos movimentos, mesmo que muito sutis, não nos passam despercebidos e podem modificar toda a informação que está sendo transmitida. Por exemplo, quando uma pessoa está mentindo, sua expressão facial normalmente permite que percebamos este fato, ainda que a diferença para a expressão facial de alguém que transmita a mesma informação, porém falando a verdade, seja mínima. Assim sendo, não é incorreto dizer que, de certa maneira, todas as pessoas são especialistas em rostos humanos.

1.2. Motivações

A modelagem e a animação de rostos humanos no computador é uma das áreas mais pesquisadas em Computação Gráfica. Ao modelar um rosto humano em três dimensões, temos que imprimir-lhe todas as suas sutis características. Além disso, os movimentos faciais mais simples não passam despercebidos em uma animação. Por exemplo, uma piscada de olho um pouco mais demorada que normal, um olhar mais demorado ou um movimento labial levemente fora de sincronismo com a voz do personagem são facilmente notados. Por isso, a representação convincente de rostos humanos em Computação Gráfica é um grande desafio de pesquisa.

Ao optar pela pesquisa nesta área, decidimos pela construção de um sistema de animação facial, afim de que pudéssemos implementar e validar experimentalmente os conceitos sobre animação facial computadorizada estudados na literatura e propostos por nós.

O candidato tem atuado na área de pesquisa em Computação Gráfica desde 1992 no Laboratório de Sistemas Integráveis da Escola Politécnica da USP. Ao longo deste período, o candidato participou do desenvolvimento de pesquisas em animação procedural de personagens virtuais em conjunto com o Media Research Laboratory, da New York University [Per95] e do desenvolvimento dos projetos Aria [Aria97] e Aria II. Participou recentemente da produção do projeto Aria III, onde o sistema **InterFace** foi utilizado para a animação de personagens humanos. O projeto Aria será descrito em detalhes no capítulo VI.

1.3. Relevância

Nesta seção descreveremos a possível aplicação de nossa pesquisa em diversas áreas da Computação Gráfica. A pesquisa na produção de técnicas de modelagem e animação interativa de expressões faciais em rostos humanos é de grande utilidade para diversas aplicações de Computação Gráfica. O uso de rostos humanos com expressões convincentes, de maneira geral, permite um maior grau de realismo e interação com os usuários de computadores. Conseqüentemente, possuindo uma grande aplicabilidade em áreas como a Realidade Virtual, interfaces de usuário, comunicação de dados e atores virtuais.

1.3.1. Realidade Virtual

Aplicações de Realidade Virtual, como telepresença e imersão em ambientes virtuais, utiliza personagens humanos modelados em computador como o seu representante no mundo virtual. O uso de animação facial torna a comunicação entre o homem e os personagens mais fácil, uma vez que estamos utilizando um canal a mais de comunicação: a emoção transmitida pelas expressões faciais.

Um exemplo de sistema de realidade virtual que utiliza largamente a animação facial é o projeto Aria [Aria97], onde um ator virtual canta um trecho de uma ópera regida pelo usuário. A animação facial tem um papel importante para transmitir as expressões do ator virtual, bem como sincronizar a voz com o movimento dos lábios.

1.3.2. Agentes de Interface Gráfica

Uma evolução natural na área de interfaces gráficas com o usuário (GUIs) é a utilização de atores virtuais, uma vez que a comunicação entre seres humanos é muito mais natural do que entre homem e máquina. Ao utilizar-mos um ator virtual como o meio de comunicação entre nós e os computadores, passamos a esperar um comportamento humano do computador. Em muitos casos é necessário apenas modelar o rosto de um personagem, pois a maior parte da comunicação se dá através da face, além do fato do modelamento de expressões de linguagem de corpo (como movimentos das mãos) ser bem mais custoso em termos de processamento computacional.

As técnicas de animação facial são importantes para agentes de interface gráfica, seja para sincronizar o movimento labial com a voz sintetizada ou para transmitir emoções como por exemplo um sorriso quando o usuário liga o computador ou vergonha

quando acontece um erro de aplicação. Para uma melhor descrição de agentes de interface gráfica, veja em [Nic95].

1.3.3. Minimização na transferência de dados

Várias técnicas de compressão de dados estão sendo utilizadas para viabilizar sistemas de vídeo-conferência em canais de largura de banda estreita, como linhas telefônicas comuns. Uma das abordagens é a utilização de detecção e síntese de expressões faciais. Demitri Terzopoulos e Keith Waters utilizam um sistema de captura de expressões faciais através de vídeo e reconstrução através de simulação de músculos, descrito em [Par96]. Apesar do processamento para captura de expressões e síntese ser alto, a descrição das expressões pode ser facilmente comprimida e transmitida utilizando-se um canal de comunicações de baixa capacidade.

1.3.4. Atores Virtuais

A tecnologia de atores virtuais (também chamada de personagens virtuais) é uma área recente na computação gráfica. O objetivo é criar personagens de computação gráfica em três dimensões com capacidade para realizar ações, como andar, falar e correr, além de tomar decisões, combinando técnicas de animação com algoritmos de inteligência artificial. Para oferecer maior realismo ao personagem, o uso de animação facial é essencial, pois permite que demonstremos suas emoções através do rosto, além de sincronizar o movimento labial com a voz. Um dos trabalhos mais relevantes na área de atores virtuais é o sistema Improv [Per96a], desenvolvido pela New York University.

1.4. Objetivo

Ao longo da história da computação gráfica, vários sistemas de animação facial foram propostos [Par72] [Wat87] [Tha89] [Wan94] [Per97], incorporando técnicas capazes de produzir expressões convincentes em personagens criados no computador.

Esta dissertação tem por objetivo pesquisar e desenvolver técnicas de animação facial que serão incorporadas em um sistema interativo e em tempo real. Este sistema de animação facial é denominado **InterFace**.

1.4.1. O Sistema **InterFace**

InterFace é um sistema de animação facial interativo e em tempo real. Através deste sistema, pretendemos incorporar técnicas de criação e animação de expressões faciais que permitam a um ator virtual expressar-se de maneira natural e interativa, minimizando o esforço de modelagem e animação do profissional de computação gráfica.

Para que pudéssemos melhor abordar o problema da criação e animação de expressões faciais em um rosto humano de forma natural, optamos por dividir o sistema **InterFace** em dois módulos. O primeiro módulo, denominado **Módulo de Criação de Expressões**, oferece recursos para a criação de expressões faciais. O segundo módulo, denominado **Módulo de Animação Facial**, utiliza as expressões modeladas e salvas pelo módulo de Criação de Expressões para animar dinamicamente o rosto do ator virtual.

O módulo de Criação de Expressões faciais utiliza o conceito de soma de expressões. Diferentes expressões previamente modeladas podem ser somadas ou subtraídas em um único rosto, resultando em uma expressão distinta. Cada expressão

criada com este método é gerada por uma combinação linear das expressões inicialmente disponíveis no sistema.

Para minimizar o esforço do animador, desenvolvemos um conjunto mínimo de expressões que devem ser modeladas previamente. A combinação destas expressões permite a criação de uma ampla gama de expressões faciais emocionalmente significativas. A este conjunto demos o nome de **Biblioteca Básica de Expressões**.

O módulo de Animação Facial utiliza as expressões criadas pelo módulo de Criação de Expressões para animar o rosto do ator virtual. Podemos associar a cada expressão uma função matemática que determina a intensidade desta expressão com o tempo. As **Ações** são criadas utilizando estas funções para variar a intensidade das expressões. Alguns exemplos de ações são respirar, piscar ou mover os olhos, espirrar e conversar. Ações de um mesmo tipo, como movimentos dos olhos, movimentos labiais e emoções, são reunidas em **Grupos de Ações** afim de auxiliar o desempenho do sistema.

Cada Grupo de Ações pode ter uma única ação ativa em um determinado instante de tempo. Para compor a expressão final que o personagem realiza em um dado instante, utilizamos um método para a composição dos Grupos de Ações inspirado em técnicas conhecidas de composição digital de imagens utilizando transparência. Estes conceitos serão descritos detalhadamente no capítulo III.

1.5. Estrutura da Dissertação

Neste capítulo de introdução foram apresentados a motivação, a relevância e os objetivos da nossa pesquisa, bem como a estrutura desta dissertação.

No capítulo II apresentaremos um pequeno histórico das técnicas de animação facial em Computação Gráfica. Através deste histórico, mostraremos as características de

outros sistemas de animação facial que serviram de inspiração para o desenvolvimento do sistema **InterFace**.

No capítulo III apresentaremos uma visão geral e conceitual do sistema **InterFace**. Descreveremos o modelo em camadas do sistema, a estrutura básica dos módulos de Criação de Expressões e Animação Facial e os conceitos utilizados em nossa pesquisa, como a biblioteca básica de expressões, a combinação de múltiplas expressões em um único rosto, o agrupamento de ações e a composição dos Grupos de Ações utilizando técnicas de composição de imagens digitais.

No capítulo IV descreveremos detalhadamente a implementação do sistema **InterFace**. Apresentaremos os requisitos para a implementação do sistema, os algoritmos que implementam os conceitos apresentados no capítulo III, bem como as linguagens de descrição de expressões, de ações e o protocolo de comunicação entre o sistema **InterFace** e programas externos.

No capítulo V apresentaremos uma análise dos resultados obtidos na implementação do sistema **InterFace**. Verificaremos a validade dos conceitos do sistema e analisaremos se a implementação cumpriu os objetivos inicialmente propostos.

No capítulo VI apresentaremos as conclusões da nossa dissertação, os projetos correlatos e as perspectivas de nossa pesquisa em projetos futuros.

Ao final da dissertação apresentaremos a lista de referências bibliográficas, a literatura, sítios *Web* recomendados e os anexos com informações adicionais do sistema.

Capítulo II

Histórico de Técnicas de Animação Facial em Computação Gráfica

Neste capítulo faremos uma revisão do estado da arte na área de animação facial em computação gráfica, apresentando os principais estudos sobre o rosto humano e explorando as técnicas utilizadas em alguns sistemas de animação facial. O objetivo desta revisão é posicionar historicamente o sistema **InterFace** em relação a outros trabalhos na área.

2.1. Primeiras pesquisas em rostos humanos

As primeiras publicações significativas na área de rostos humanos foram feitas por Duchenne [Duc62] e por Charles Darwin [Dar72]. Duchenne investigou a importância que alguns músculos da face exercem nas expressões, utilizando uma técnica simples que consiste em aplicar uma pequena corrente elétrica em determinadas áreas do rosto, fazendo com que os músculos se contraíam. O seu trabalho se destaca pelo uso de fotografias, uma inovação para a sua época. Os resultados desta pesquisa foram publicados em 1862 [Duc62].

Em 1872, Darwin publicou o livro *The Expression of the Emotions in Man and Animals* [Dar72]. Este livro é um estudo sobre a universalidade das expressões faciais. Mesmo em diferentes épocas, culturas e raças, expressões faciais que representam dor, alegria, tristeza, seriedade ou força são sempre semelhantes. Darwin demonstrou também que expressões de ataque, medo e dor são semelhantes para os homens e os animais.

Apesar desta publicação nunca ter obtido o reconhecimento do livro *The Origin of Species*, durante todo o século seguinte as pesquisas sobre os rostos humanos foram basicamente refinamentos das teorias propostas por Darwin.

Em 1977, o trabalho de Duchenne foi revisto por Paul Ekman e Wallace Friesen. Eles propuseram um sistema chamado *Facial Action Coding System (FACS)* (descrito em [Par96]), no qual foram classificadas as ações dos músculos da face. Estas ações receberam o nome de Unidades de Ação (UA). Cada unidade de ação apresenta o conjunto de músculos que devem ser relaxados ou contraídos para resultar em um pequeno movimento em uma parte do rosto. Por exemplo, fechar os olhos (UA número 43) é o resultado da relaxação do músculo *Levator Palpebrae Superioris*. Para piscar os olhos (UA 45), relaxa-se o músculo *Levator Palpebrae* e depois contrai-se o *Orbicularis Oculi* e *Pars Palpebralis*. Os autores demonstraram que com a combinação de sessenta e seis unidades de ação, é possível criar uma ampla gama de expressões faciais.

2.2. Primeiras Pesquisas de Animação Facial em Computador

Apresentaremos nesta seção o desenvolvimento das primeiras pesquisas em Computação Gráfica envolvendo o rosto humano.

2.2.1. O Trabalho Pioneiro de Frederick Parke

A primeira pesquisa em animação facial por computador e uma das primeiras pesquisas significativas na área de Computação Gráfica foi publicada em 1972, por Frederick Parke [Par72]. Seu trabalho consistiu em modelar e animar um rosto humano no computador.

Para modelar o rosto, Parke utilizou a técnica de fotometria. Esta técnica consiste em marcar em um rosto humano uma série de pontos de referência, e fotografá-lo em diferentes pontos de vista. Medindo-se a distância entre estes pontos e utilizando equações de transformação de vistas, obtém-se uma malha de pontos de descrevem o rosto em três dimensões.

Para animar este rosto, Parke utilizou a técnica de interpolação linear. Movendo-se determinados vértices do rosto, obtém-se uma nova expressão. A animação consiste em interpolar a posição dos vértices no rosto original até a posição dos vértices na nova expressão, utilizando a seguinte equação:

$$\boxed{[P_r] = [P_o] + ([P_E] - [P_o])t}$$

Tabela 2.1: Equação da interpolação linear de vértices

Onde P_R é a posição resultante dos pontos, P_o é a posição original e P_E é a posição dos pontos da expressão final. A variável t limita-se a faixa de 0 a 1.



Figura 2.1: Exemplo de Animação por Interpolação Linear

Apesar de ser uma técnica de animação simples, este algoritmo ainda hoje é largamente utilizado pela maioria dos sistemas de computação gráfica.

2.2.2. A Parametrização do rosto

A técnica de interpolação linear obtém resultados aceitáveis em modelos simples e não muito expressivos. Porém, conforme a complexidade ou a expressividade do personagem aumenta, a interpolação linear torna-se impraticável, pois para cada expressão nova que é criada, o animador tem que trabalhar com todo o rosto do personagem. Por consequência, o número de expressões faciais a serem modeladas em animações complexas torna-se muito grande, pois é preciso modelar todos os movimentos labiais e expressões emocionais dos personagens.

Para simplificar a criação de expressões, Parke desenvolveu modelos paramétricos do rosto humano [Par82]. Esta técnica consiste em dividir o rosto em determinadas áreas e permitir que o animador, ao controlar os parâmetros que exercem alguma função sobre estas áreas, possa modelar rapidamente novas expressões. A animação entre as expressões é obtida alterando-se os valores dos parâmetros.

Para a criação de expressões, Parke escolheu os seguintes parâmetros:

- movimento das pálpebras (piscar os olhos);
- movimento de rotação das sobrancelhas;
- movimento vertical das sobrancelhas;
- rotação da mandíbula;
- largura da boca;
- expressões da boca;
- posicionamento do lábio superior;
- posicionamento dos cantos da boca;

- movimento das pupilas dos olhos.

Com estes parâmetros é possível criar muitas expressões faciais. O sistema **InterFace** aplica este conceito de forma simplificada, através da Biblioteca Básica de Expressões, descrita no capítulo III desta dissertação.

Em 1985 Bergeron, Lachapelle e Langlois aprimoraram o trabalho de Parke na animação *Tony de Peltrie* [Ber85]. A animação exigia que o personagem transmitisse seu sentimento basicamente através das expressões faciais. Para que isso fosse possível, os pesquisadores combinaram as técnicas de interpolação linear e parametrização.

A partir de uma expressão facial neutra (chamada de E_0), outras expressões foram criadas movendo-se determinados vértices do rosto (chamadas de E_1 , E_2 , etc.). Foram computadas as distâncias entre os vértices do rosto na expressão neutra e nas outras expressões ($E_n - E_0$) para cinco áreas específicas do rosto (as pálpebras esquerda e direita, as pupilas esquerda e direita e o restante da face, incluindo a boca). Novas expressões foram construídas somando e subtraindo estes vetores de distância num único rosto. Através desta técnica, denominada soma de vetores dos vértices, foi possível combinar várias expressões em um mesmo rosto simultaneamente.

2.2.3. Modelamento de Expressões Faciais por Músculos

A partir de 1987, a pesquisa em animação facial por computador tomou um novo rumo, com a introdução de simulação dos músculos da face. Dois trabalhos deram origem quase que simultaneamente à este campo de pesquisa.

Magenat-Thalmann [Tha88] desenvolveu um modelo de simulação de músculos bastante simples, chamado de “pseudomúsculo”, por basear-se em um modelo empírico e

não na simulação física. O modelo utilizado por Thalmann simula a ação dos músculos através de técnicas de deformação livre de formas (*freeform deformations*). Para a criação de expressões foi utilizado um método chamado de **Ação Abstrata de Músculos** (*Abstract Muscle Action*, ou AMA), semelhante ao sistema FACS. As expressões são divididas em pequenas ações (as AMAs) e para cada ação é especificado o movimento (contração ou relaxamento) que os pseudomúsculos devem fazer. As AMAs são chamadas de parâmetros de expressões de baixo nível. Combinando-se as AMAs, obtêm-se expressões de um nível mais alto.

Em 1987, Keith Waters [Wat87] desenvolveu um sistema de simulação de músculos baseando-se em modelos físicos. O modelo de Waters incluía dois tipos de músculos: músculos lineares, para tração, e músculos tipo esfíncer, para compressão. Foram atribuídos valores de "massa" dos vértices dos polígonos que compõe o rosto e entre as arestas a ação dos músculos foi simulada por molas. Este modelo de molas e massa aproxima-se razoavelmente bem da ação que os músculos tem sobre a face. Cada músculo tem uma determinada zona de influência. A influência de um determinado músculo diminui proporcionalmente à distância radial do ponto onde o músculo está fixado. Para criar as expressões foram utilizados os parâmetros do sistema FACS.

2.2.4. Animação Facial

Paralelo ao desenvolvimento de técnicas para a criação de expressões faciais, os pesquisadores de computação gráfica aprimoraram também métodos para animar o rosto humano.

A maneira mais simples de animar um rosto no computador é interpolar linearmente a posição dos vértices de uma expressão para outra (ver tabela 2.1). Além da

interpolação linear, outras funções podem ser utilizadas para animar os vértices de uma expressão para outra. Por exemplo, funções tipo *splines* podem ser utilizadas quando se deseja uma interpolação mais suave. Ainda hoje, o método da interpolação é o mais utilizado para animar rostos no computador.

A utilização de curvas para interpolação de expressões é um processo demorado e trabalhoso. Para simplificar este trabalho, foram criadas algumas técnicas de animação para tornar o processo mais automático. Frederick Parke [Par82] desenvolveu uma linguagem bastante simples para animação facial: o animador especifica a expressão inicial, a expressão final, o número dos quadros inicial e final e a função utilizada para a interpolação.

Nadia Magnenat-Thalmann desenvolveu uma técnica semelhante a de Parke em [Tha88]. A linguagem utiliza os mesmos princípios da linguagem desenvolvida por Parke, com uma diferença: várias expressões podem ser combinadas em um único rosto. Isto se deve ao método de criação de expressões, através de AMAs (*abstract muscle actions*), onde uma expressão é uma combinação de vários AMAs simultaneamente.

Em 1991, Nadia Thalmann e Kalra [Kal91] aprimoraram a técnica de animação facial de [Tha88], através do conceito de animação em camadas. O processo de animação foi decomposto em cinco camadas. As camadas inferiores determinam **como** a animação é feita, enquanto que as camadas superiores determinam **o que** deve ser feito. As camadas são as seguintes:

- **Camada 1: pseudomúsculos** - esta é a camada mais básica do sistema. Aqui estão especificadas as ações básicas que os músculos exercem sobre o rosto (as AMAs);

- **Camada 2: movimentos perceptíveis** - nesta camada são especificadas as ações mais básicas de animação. Aqui são especificadas a AMA que será utilizada, a intensidade e o número do quadro de animação;
- **Camada 3: fonemas e expressões** - esta camada trabalha com um nível mais alto de abstração de animação. Nela são especificadas quais AMAs deverão ser ativadas para realizar uma determinada expressão ou fonema;
- **Camada 4: palavras e expressões** - esta camada do sistema combina os fonemas e expressões básicas para a criação de palavras e expressões mais complexas, como emoções;
- **Camada 5: sincronização de emoções, palavras e movimentos dos olhos** - esta é camada de mais alto nível. Aqui é possível especificar as emoções e as palavras que serão utilizadas ao longo da animação.

O trabalho do animador neste sistema se assemelha com um diretor de uma cena. Basta especificar as frases e as expressões para que o ator sintético desenvolva a cena.

O trabalho de Ken Perlin em animação procedural de atores virtuais [Per94][Per95][Per96a][Per96b] foi adaptado para resolver o problema da animação de rostos humanos. Em [Per97], Perlin utilizou um sistema de ações em camadas para animar um rosto humano de duas dimensões interativamente no computador.

O método de Perlin facilita o processo de criação de expressões das camadas 1 e 2 do modelo de Kalra. Certas ações como piscar os olhos ou movimentar levemente o rosto simulando a respiração são feitas automaticamente pelo sistema, através da utilização de funções de ruído como curvas de interpolação. Para a camada 2 do modelo de Kalra, Perlin utilizou um outro sistema em camadas, para permitir que várias emoções pudessem

ser simuladas simultaneamente (como, por exemplo, falar e chorar ao mesmo tempo). Para compor a animação entre estas camadas foram utilizadas técnicas semelhantes a composição de imagens com transparência [Por84].

2.3. O Estado da Arte em Animação Facial

As pesquisas atuais na área de animação facial tem sido direcionadas para a obtenção de modelos tridimensionais de rostos humanos foto-realistas.

Três trabalhos recentes ([Gue98][Bla99][Pig98]) apresentaram novas técnicas de obtenção de modelos de rostos humanos através de fotometria. Em todos estes trabalhos, o modelo de três dimensões do rosto humano foi obtido pela extração de parâmetros como a distância entre elementos da face, como nariz, boca, olhos e traços do rosto, a partir de imagens de vídeo. Estes parâmetros foram utilizados para ajustar modelos tridimensionais pré-existentes. A imagem do vídeo também foi utilizada para mapeamento de textura, resultando em modelos tridimensionais de rostos humanos foto-realistas.

No trabalho de Pighin et al. ([Pig98]), foi utilizada uma técnica de criação de expressões faciais também baseada em aquisição de vídeo. A partir destas expressões, expressões faciais foram criadas utilizando-se a mesma técnica empregada no sistema **InterFace**: a composição de múltiplas expressões em um único rosto. Descreveremos esta técnica no capítulo III desta dissertação.

2.4. O Sistema InterFace

O sistema **InterFace** baseia-se principalmente nas pesquisas de Parke [Par72][Par82], Kalra [Kal91] e Perlin [Per95][Per96a][Per96b][Per97]. O módulo de

Criação de Expressões Faciais utiliza a combinação de diferentes expressões em um único rosto; o módulo de Animação Facial baseia-se num modelo de camadas de ações de Kalra simplificado, utilizando também os conceitos de animação procedural desenvolvidos por Perlin [Per97] adaptados para rostos em três dimensões. O capítulo III desta dissertação descreve em detalhes os conceitos do sistema **InterFace**.

2.5. Conclusão

É importante notar que, apesar de todo o avanço que os sistemas de animação facial tiveram desde 1972 até hoje, até recentemente a maioria absoluta dos programas comerciais de computação gráfica utilizavam métodos semelhantes ao sistema desenvolvido por Parke em 1972 [Par72], ou seja, modelos de polígonos e animação por interpolação. Outros métodos para representação de superfície, como *B-Splines*, não se mostraram simples ou rápidos o bastante para substituírem os polígonos. A animação através de simulação de músculos, além de ser muito cara computacionalmente, não obteve a mesma qualidade obtida por um animador que utiliza técnicas tradicionais de modelamento de expressões.

Por isso, acreditamos que a área de modelagem e animação de rostos humanos por computador ainda tem muitos desafios pela frente.

Capítulo III

Conceitos do Sistema InterFace

Neste capítulo apresentaremos detalhadamente a concepção do sistema **InterFace**. Descreveremos os dois módulos do sistema, **Criação de Expressões** e **Animação Facial**, e os conceitos envolvidos em cada módulo: a biblioteca básica de expressões, a combinação de expressões e composição de grupos de ações com transparência.

3.1. Visão Geral

InterFace é um sistema de animação facial por computação gráfica. Este sistema foi desenvolvido com o objetivo de aprimorar a pesquisa em técnicas de criação e animação de expressões faciais. O sistema é interativo e opera em tempo real, ou seja, o ator virtual responde aos comandos do usuário imediatamente, da mesma forma que agiria um ator real. Por esta característica, as principais aplicações do sistema **InterFace** são na área de atores virtuais, animação computacional de alto desempenho, realidade virtual e agentes de interface gráfica.

O sistema **InterFace** utiliza uma visão simplificada do modelo de Kalra [Kal91] para descrever um ator virtual. No capítulo II desta dissertação, apresentamos o modelo de um sistema de animação facial em camadas de Kalra. Este modelo adequa-se melhor à implementação de um sistema do que à sua descrição abstrata.

O sistema **InterFace** utilizou um modelo abstrato simples de três camadas, apresentado a seguir:

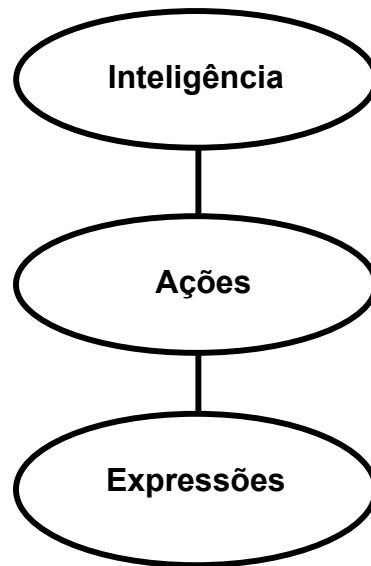


Figura 3.1: Modelo abstrato de três camadas do sistema *InterFace*

A primeira camada, **Expressões**, contém as informações sobre as expressões faciais que o ator virtual conhece. Internamente a esta camada estão tanto as expressões faciais modeladas quanto expressões da Biblioteca Básica, conforme descreveremos no módulo de Criação de Expressões. Esta camada recebe como entrada as expressões que o personagem deve executar, com um respectivo valor de intensidade de cada expressão. Como saída, esta camada modifica o rosto do ator virtual, rotacionando, transladando, escalando ou movendo os vértices dos objetos que compõem este rosto para produzir a expressão resultante da combinação das expressões e valores de intensidade dados pela entrada.

A segunda camada, **Ações**, contém as ações definidas para o ator virtual. Ações são pequenas animações que representam desde emoções (como sorrir ou chorar), movimentos dos olhos ou labiais para a sincronização com a voz, até ações não intencionais como respirar e piscar. As ações similares são reunidas em grupos, para facilitar a composição de diferentes ações simultaneamente. Esta camada recebe como entrada qual ação ou quais ações devem ser executadas e como saída envia as expressões

com seus respectivos valores de intensidade resultantes da combinação destas ações para a camada de Expressões.

Na terceira camada, **Inteligência**, são escolhidas quais ações o ator virtual irá realizar. Na saída desta camada, portanto, são enviadas as ações escolhidas para a camada de Ações. Esta camada pode ser implementada de diversas maneiras. Podemos ter desde um usuário escolhendo as ações através de um menu, um roteiro de ações a serem executadas com o tempo, um programa de inteligência artificial que escolhe as ações através de algum método heurístico de tomada de decisões ou até a comunicação com outros meios externos, como sistemas de música e de síntese de voz que determinam qual movimento labial e expressão facial o personagem deve realizar. Em particular, esta última implementação para a camada inteligência foi utilizada no projeto ARIA [Aria97].

O sistema **InterFace** é dividido em dois módulos. O primeiro módulo é responsável pela criação de expressões faciais e o segundo módulo pela animação facial. O módulo de Criação de Expressões serve como suporte para o módulo de Animação Facial, pois as ações do personagem são construídas a partir das expressões criadas pelo módulo de Criação de Expressões. No módulo de Animação Facial é feita a simulação do ator virtual, implementando as três camadas do nosso modelo do sistema.

3.2. Módulo de Criação de Expressões

O objetivo do módulo de Criação de Expressões é permitir a criação de expressões faciais de forma simples e intuitiva. Existem diversos sistemas comerciais de computação gráfica de alta qualidade para a criação de modelos em três dimensões. Porém, não conhecemos nenhum sistema de Computação Gráfica que tenha recursos voltados exclusivamente para a criação de expressões faciais. Normalmente, este processo é

trabalhoso: cada expressão facial tem que ser criada individualmente, trabalhando-se diretamente sobre os objetos que compõe o rosto.

O sistema **InterFace** propõe um método mais simples e fácil para a criação de expressões faciais complexas: a combinação de expressões simples previamente criadas em um único rosto. Para entender este processo, apresentaremos dois conceitos:

- **Expressão Neutra:** é a expressão básica do personagem. Quando não utilizamos nenhuma expressão em particular, assumimos que o personagem está na expressão neutra.
- **Biblioteca Básica de Expressões:** é um conjunto de expressões simples que são combinadas para gerar expressões faciais complexas.

O sistema de criação de expressões pode ser descrito a grosso modo da seguinte maneira: o personagem é modelado na expressão neutra. A partir da expressão neutra, são modeladas as expressões da biblioteca básica. Combinando-se as expressões da biblioteca básica, é possível construir expressões faciais mais complexas.

3.2.1. A Biblioteca Básica de Expressões





A Biblioteca Básica de Expressões (ou Biblioteca Básica) é um conjunto de expressões faciais simples que, ao serem combinadas, permitem a criação de uma ampla gama de expressões faciais emocionalmente significativas.




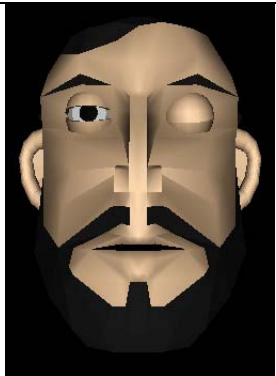




Um dos principais desafios do Módulo Criação de Expressões é definir o conjunto mais apropriado para as expressões faciais que formam a Biblioteca Básica. O número de expressões que compõe a biblioteca básica não deve ser nem muito grande, para evitar

um trabalho complexo de modelamento, nem muito pequeno, de modo a limitar o número de expressões que pode ser criada por combinação.

As expressões que utilizaremos em nossa biblioteca básica foram adaptadas do trabalho desenvolvido por Perlin em [Per97]. Perlin utilizou este conjunto de expressões em um modelo de rosto de duas dimensões obtendo bons resultados. A implementação do sistema **InterFace** demonstrou que o mesmo conjunto pode ser aplicado com sucesso em rostos humanos de três dimensões. No sistema de Perlin e no InterFace, as expressões da Biblioteca Básica foram testadas apenas em personagens com aparências caricaturais (*cartoons*). Porém, nada impede que este conjunto seja também utilizado em personagens com maior grau de realismo, como modelos de rostos humanos obtidos por *scanner* ou fotometria.

As expressões que compõe a Biblioteca Básica são:

	1. Movimento labial em forma de "Ahh"		2. Movimento labial em forma de "Ohh"
	3. Movimento labial em forma de "Dúvida" - lado esquerdo		4. Movimento labial em forma de "Dúvida" - lado direito

	5. Movimento labial em forma de "Sorriso" - lado esquerdo		6. Movimento labial em forma de "Sorriso" - lado direito
	7. Piscar o olho esquerdo		8. Piscar o olho direito
	9. Rotação da sobrancelha esquerda		10. Rotação da sobrancelha direita
	11. Movimento da pálpebra inferior esquerda		12. Movimento da pálpebra inferior direita






	<p>13. Movimento vertical dos olhos</p>		<p>14. Movimento horizontal dos olhos</p>
	<p>15. Rotação da cabeça em torno do eixo X</p>		<p>16. Rotação da cabeça em torno do eixo Y</p>
	<p>17. Rotação da cabeça em torno do eixo Z</p>		

Tabela 3.1: A Biblioteca Básica de Expressões

O conceito da Biblioteca Básica de Expressões é uma implementação da parametrização do rosto proposta por Frederick Parke [Par82]. Apesar das expressões da Biblioteca Básica formarem um conjunto mais reduzido que o número de parâmetros propostos por Parke, demonstramos com o sistema **InterFace** que este conjunto é suficiente para permitir a criação de uma ampla gama de expressões emocionalmente significativas, como a sincronização do movimento labial com a voz do personagem e expressões complexas como desconfiança, medo, susto ou alegria, além de permitir a animação dos olhos e da cabeça.

As expressões da Biblioteca Básica trabalham com todas as áreas emocionalmente significativas do rosto: os olhos, as pálpebras superior e inferior, a boca e regiões próximas, além de movimentos completos da cabeça. As expressões de 3 a 12 da Biblioteca Básica são modeladas em apenas um lado do rosto. Isto permite a criação de expressões não simétricas lateralmente. De acordo com [Per97], muitas expressões tornam-se mais naturais quando não há simetria entre os dois lados do rosto.

3.2.2. Combinação de Expressões

No sistema **InterFace**, cada expressão da Biblioteca Básica é armazenada como um vetor de diferenças entre esta expressão e a expressão neutra. Ou seja, quando o rosto está na expressão neutra e passa para uma determinada expressão da biblioteca básica, os objetos que compõe o rosto podem ter sido rotacionados, transladados, escalados ou podem ter alguns vértices mudados de posição. O sistema **InterFace** armazena cada expressão da Biblioteca Básica como este conjunto de modificações a partir da expressão neutra. A este conjunto é dado o nome de **Vetor de Diferenças** de uma expressão da Biblioteca Básica para a expressão neutra.

Para criar novas expressões, o sistema **InterFace** faz a soma ponderada dos vetores de diferenças que compõe cada expressão da biblioteca básica. Cada expressão da Biblioteca Básica tem um valor de intensidade na composição de uma nova expressão. Se uma expressão da Biblioteca Básica tiver o valor de intensidade igual a 1 e as demais expressões da biblioteca básica tiverem valor de intensidade zero, então a expressão resultante será idêntica àquela expressão da Biblioteca Básica.

No exemplo abaixo, mostramos a combinação de três expressões diferentes, sendo a primeira o resultado de uma soma entre as expressões 9 e 10 da Biblioteca Básica, a

segunda é a expressão 1 da Biblioteca Básica e a terceira é a expressão 2 da Biblioteca Básica. O resultado é apresentado na imagem da direita.

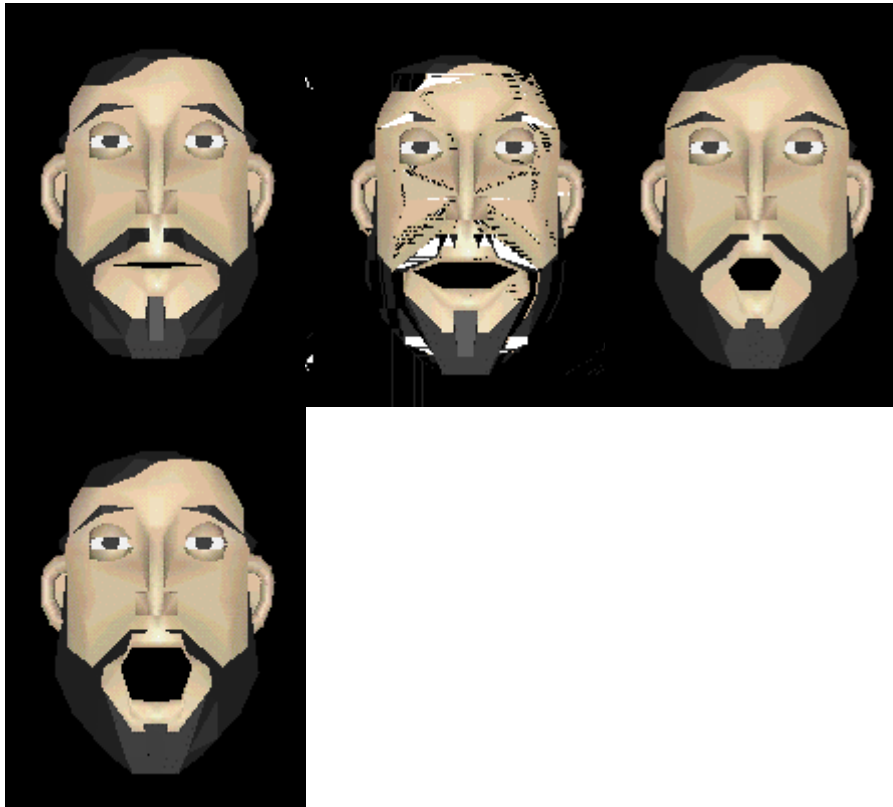


Figura 3.2: Exemplo de Soma de Expressões

O valor da intensidade das expressões da Biblioteca Básica não está limitado à faixa $[0,1]$. Podemos utilizar qualquer valor real. Denominamos **Extrapolção de uma Expressão** o uso de valores de intensidade superiores a 1 ou inferiores a 0 para uma expressão. O uso da extrapolção amplia a capacidade de criar expressões com a Biblioteca Básica. Por exemplo, ao colocar o valor de peso -1 para as expressões "sorriso esquerdo" e "sorriso direito" da Biblioteca Básica, obtemos uma expressão triste. Nas imagens abaixo, mostramos as expressões da Biblioteca Básica números 5 e 6 (sorriso direito e esquerdo), ambas com valores de intensidade +1 (imagem esquerda) e -1 (imagem direita).

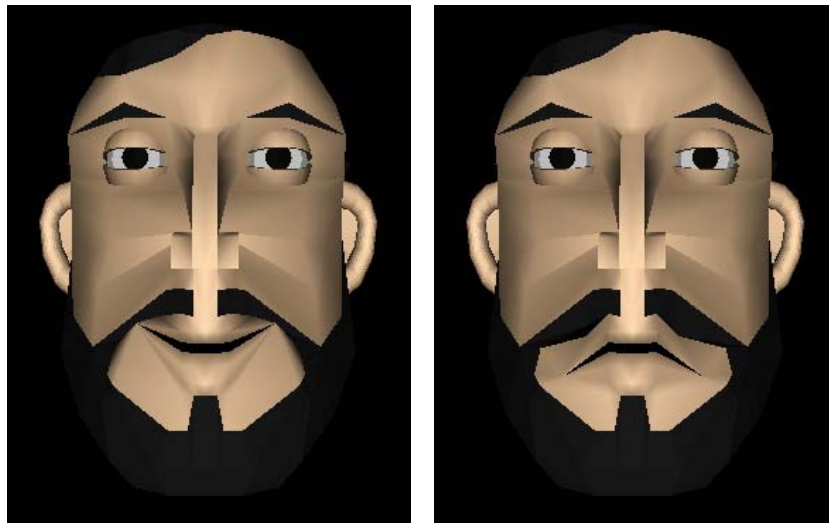


Figura 3.3: Exemplo de Extrapolação

Apesar podermos usar qualquer valor real para o valor de intensidade de uma expressão, os valores úteis raramente excedem a faixa de $[-3,+3]$.

Por se tratar de uma simples soma de vetores, a ordem com que as expressões são combinadas não é importante. A soma de um mesmo conjunto de pesos para os Vetores de Diferença resultará em uma única expressão, independentemente da ordem em que for executada.

3.3. Módulo de Animação Facial

Conforme descrevemos anteriormente, o sistema **InterFace** utiliza um modelo de três camadas para descrever um ator virtual: Expressões, Ações e Inteligência. No módulo de animação facial vemos as três camadas operando simultaneamente, simulando o ator virtual.

A animação facial é feita através da composição de ações, divididas em grupos e utilizando transparência. Nos próximos itens analisaremos os conceitos de ações, grupos de ações e composição de ações com transparência.

3.3.1. Ações

Ações são pequenas animações que o personagem pode executar. Exemplos de ações são piscar os olhos, falar um fonema ou uma frase, cantar, chorar, dormir e espirrar. O conjunto de todas as ações é o que a terceira camada (inteligência) utiliza para controlar o ator virtual.

As ações são construídas baseando-se nas expressões modeladas no módulo de Criação de Expressões. Cada expressão criada pode ser associada a uma função matemática que modifica a sua intensidade com o tempo. O conceito de intensidade de uma expressão é o mesmo utilizado no módulo de Criação de Expressões, porém neste módulo é aplicado à expressão criada como um todo, e não à cada expressão da Biblioteca Básica individualmente.

As funções matemáticas disponíveis no sistema **InterFace** são as seguintes: seno, cosseno, interpolação linear, *splines*, impulso ou função ruído de Perlin [Per85][Per89]. Em particular, a função ruído de Perlin é uma ferramenta importante para a criação de ações mais realistas, pois as características desta função - contínua, não-determinística, porém controlável - podem ser usadas para modelar acuradamente ações como piscar de olhos ou movimentos sutis da cabeça simulando a respiração.

Para facilitar a construção das ações, o sistema **InterFace** permite que a função que determina o valor da intensidade de uma expressão possa ser modulada por outra função denominada **envelope**. O objetivo do envelope é facilitar a criação de ações, permitindo que diversas expressões podem ser combinadas em uma ação com uma transição suave entre elas. A função de envelope será apresentada no capítulo IV desta dissertação.

As ações podem utilizar uma ou mais expressões moduladas por funções de intensidade. Se várias expressões forem utilizadas simultaneamente, o sistema **InterFace** soma as expressões utilizando o mesmo algoritmo para a soma de expressões do módulo de criação de expressões. Assim, a ordem com que as expressões são compostas não é importante. No capítulo IV descreveremos com detalhes o algoritmo utilizado juntamente com exemplos de ações.

A ação como um todo também pode ser envolvida por um envelope. O objetivo deste envelope é permitir que as transições entre o término de uma ação e o início de outra ou a interrupção de uma ação durante a sua execução sejam feitas de forma suave.

3.3.2. Grupos de Ações

Por se tratar de um sistema da animação facial em tempo real, é importante otimizar o processamento das ações. Uma maneira simples de fazer isto é evitar que diversas ações sejam executadas simultaneamente. Por outro lado, desejamos que o ator virtual tenha liberdade para se expressar de forma natural. A solução encontrada para este caso foi utilizar grupos de ações.

O principal conceito dos Grupos de Ações consiste em permitir que uma única ação esteja ativa em um grupo em um dado instante. Por exemplo, um personagem normalmente não ri enquanto chora, não espirra enquanto ri e não fala duas frases simultaneamente.. Portanto, os grupos de ações reúnem em si ações que *normalmente* são mutuamente exclusivas.

Para que o número de grupos de ação não se torne muito grande, o sistema provê um mecanismo para que duas ações de um mesmo grupo possam ser executadas simultaneamente, o que pode ser desejável em determinados casos.

Os grupos de ações escolhidos para o nosso sistema são os seguintes:

1. Ações não intencionais, como respirar ou piscar os olhos;
2. Emoções, como chorar, sorrir ou dormir;
3. Ações para sincronismo de movimentos labiais;
4. Movimentos dos olhos;
5. Movimentos de cabeça, como "sim" ou "não";
6. Todas as ações de todos os grupos anteriores.

Na escolha dos grupos procurou-se quanto possível a maior independência entre a área em que as ações atuam. Por exemplo, um grupo trata apenas de movimentos labiais, enquanto outro trata de movimentos dos olhos e outro de movimentos da cabeça. Esta escolha foi feita para que haja maior controle sobre a ação de cada grupo na composição de expressão final do ator virtual.

O último grupo, que contém todas as ações dos grupos anteriores, é gerado automaticamente pelo sistema **InterFace**. O objetivo da existência deste grupo é permitir que duas ações de um mesmo grupo estejam ativas simultaneamente, caso haja necessidade. Como o sistema não permite duas ações ativas em um único grupo, a solução foi utilizar um outro grupo que contém todas as ações. Assim, duas ações que pertencem a um mesmo grupo podem ser executadas simultaneamente, uma no grupo de ações ao qual ela pertence originalmente e a outra neste último grupo de ações. Obviamente, como há apenas um grupo que contém todas as ações, o caso de executar duas ações de um mesmo grupo simultaneamente só pode ocorrer para um único grupo a um dado instante. Porém, como este é um evento raro, a probabilidade de que ocorra para dois grupos de ação ao mesmo tempo é muito pequena.

Cada grupo de ação contém um valor de transparência (também chamado de *alpha* ou α). Definimos a transparência de um grupo como sendo a sua influência no processo de composição dos Grupos de Ações. Se um grupo tem o valor de transparência igual a 1, a ação executada naquele grupo não exerce nenhuma influência para o ator. Da mesma maneira, quando α é igual a zero, a ação está exercendo sua influência total na composição dos Grupos de Ações. Os valores de α estão limitados entre a faixa de 0 a 1 inclusive.

A composição entre as ações executadas nos grupos segue um algoritmo semelhante à composição digital de imagens com transparência [Por84]. Este algoritmo pode ser descrito pelo seguinte pseudo-código:

```

Enquanto o sistema estiver rodando {
  Para cada grupo (i) de ações {
    Executa-se a ação ativa no grupo, resultando em um
    vetor de intensidades de expressões modeladas;

    Este vetor é transformado em um vetor de valores de
    intensidade das expressões da Biblioteca Básica
    (VVIEBB);

    Somamos com o VVIEBB acumulado pelos grupos
    anteriores através da seguinte expressão:
     $VVIEBB_i = VVIEBB_i * \alpha_i + VVIEBB_{i-1} * (1 - \alpha_i)$ ;

  }
  O vetor VVIEBB gera a expressão atual do personagem;
}

```

Tabela 3.2: Pseudo-código da composição dos Grupos de Ação

Note que o vetor de valores de intensidade das expressões gerado pelas ações só leva em consideração as expressões da Biblioteca Básica que realmente foram utilizadas

pela ação em questão. As expressões da Biblioteca Básica que não são utilizadas pela ação são consideradas "transparentes" para o sistema, ou seja, não exercem nenhuma influência na composição daquela ação.

A cada Grupo de Ações que é processado, o vetor com os valores de intensidade das expressões modeladas resulta em um vetor com os valores de intensidade da biblioteca básica. Conforme explicamos anteriormente, uma única expressão modelada pode conter várias expressões da Biblioteca Básica. Para transformarmos o vetor de expressões modeladas em valores de intensidade da Biblioteca Básica, utilizamos para cada expressão modelada a soma dos valores de intensidade que a compõe multiplicado pelo valor da intensidade da expressão criada naquele instante. Assim, por se tratar de uma soma simples, a ordem com que os valores de intensidade de expressões modeladas é calculada não modifica o vetor dos valores de intensidade da Biblioteca Básica.

Como podemos observar no pseudo-código, o valor de transparência e a ordem com que os Grupos de Ações são acumulados são importantes para a composição da expressão resultante. Uma alteração na ordem com que os grupos são acumulados pode resultar em uma expressão final diferente. Os últimos grupos a serem processados tem maior prioridade que os grupos precedentes. Assim, se um grupo de maior prioridade tiver índice de transparência α igual a zero (ou seja, for totalmente opaco) e uma das ações deste grupo exercer um certo valor de intensidade sobre uma expressão da Biblioteca Básica, este valor irá sobreescrever qualquer intensidade para aquela expressão exercido por grupos de menor prioridade.

O algoritmo para composição dos Grupos de Ações é repetido a cada quadro de animação que é gerado pelo sistema. Idealmente, a composição dos Grupos de Ações devem ser calculados de 25 a 30 vezes por segundo.

3.4. Inteligência do Ator Virtual

Em nosso modelo de três camadas utilizado para descrever o ator virtual, a terceira camada corresponde à inteligência do personagem. Conforme descrevemos acima, há vários métodos de implementar esta camada. No sistema **InterFace** implementamos dois métodos.

No primeiro método, o usuário do sistema **InterFace** age como a inteligência do ator virtual. O sistema apresenta um menu com todos os grupos de ações e as ações disponíveis para cada grupo. O usuário pode selecionar a ação que o ator realizará a cada instante, ou interromper uma ação que o ator estiver executando. O valor de transparência de cada grupo e a ordem com que os grupos são acumulados podem ser modificados em tempo de execução.

Este método, apesar de ser prático para testar a funcionalidade do sistema, requer a intervenção direta do usuário. Não podemos afirmar que o ator virtual é realmente um *Ator Virtual*, pois sua inteligência está limitada a ação do usuário. Para contornar esta limitação, o sistema **InterFace** permite que o ator virtual realize ações enviadas por outros sistemas.

Qualquer programa pode enviar para o **InterFace** as ações que deseja que o ator virtual execute. Um programa de inteligência artificial, por exemplo, pode processar as ações que o ator virtual deve realizar com o tempo e enviá-las para serem executadas no **InterFace**. O projeto Aria [[Aria97](#)] utiliza esta abordagem: para fazer o ator virtual "cantar", o programa de som gera as ações que o personagem deve executar com o tempo, desde as expressões labiais para imitar a voz até as expressões faciais como força ou tristeza, que são combinadas no **InterFace** através da composição de Grupos de Ações.

A transmissão das ações que o ator virtual deve executar é feita pela rede de computadores através de um sistema cliente/servidor implementado sobre TCP/IP, possibilitando que, potencialmente, qualquer máquina conectada à Internet possa requerer que o personagem realize determinada ação.

3.5. Conclusão

Neste capítulo apresentamos os conceitos do sistema **InterFace**. Estes conceitos foram desenvolvidos baseados em nossa pesquisa na literatura existente na área. No capítulo IV, apresentaremos a implementação destes conceitos na produção do sistema **InterFace**.

Capítulo IV

Implementação do Sistema InterFace

Nos capítulos anteriores apresentamos os conceitos que envolvem o sistema **InterFace**, como a biblioteca básica de expressões e o agrupamento de ações. Há várias maneiras de se implementar estes conceitos. Podemos utilizar, por exemplo, objetos descritos por polígonos ou por *splines*, ou até mesmo rostos de duas dimensões; podemos implementar o método de agrupamento de ações tanto em sistemas de animação não interativos, voltados à criação de imagens foto-realísticas, quanto em sistemas interativos de realidade virtual.

Neste capítulo descreveremos o projeto do sistema **InterFace**, seu funcionamento interno e implementação e as soluções adotadas para atender aos requisitos propostos.

4.1. Requisitos do Sistema

Ao projetar o sistema **InterFace**, especificamos que o mesmo deveria atender aos seguintes requisitos:

- Sistema intuitivo, fácil de usar. O tempo necessário para o usuário familiarizar-se com o sistema deve ser pequeno;
- Interativo; Entendemos por interatividade a resposta imediata aos comandos enviados ao sistema. No sistema InterFace, interatividade consiste no ator virtual responder aos comandos do usuário ou de outro sistema imediatamente;

- Operação em tempo real em computadores pessoais, com pequena capacidade de processamento. Entendemos por operação em tempo real a obtenção de movimentos contínuos e suaves. Quantitativamente, esperamos o sistema tenha um desempenho capaz de gerar no mínimo 8 quadros por segundo;
- Expansível; Desejamos que o sistema **InterFace** possa incorporar novos recursos facilmente. Desejamos também que o sistema **InterFace** possa receber comandos de outros sistemas.

4.2. Decisões de Projeto

Nesta seção apresentaremos as decisões iniciais do projeto para atender aos requisitos de tempo real e interatividade em computadores pessoais com pequena capacidade de processamento.

4.2.1. Representação da Geometria

Existem vários métodos de representação de objetos geométricos em três dimensões no computador. Podemos dividir estes métodos em dois grandes grupos:

- **Representação Volumétrica**, que utiliza elementos básicos em três dimensões (os *voxels*);
- **Representação por Superfície**, que utiliza polígonos, equações ou curvas para representar a superfície do objeto.

A representação volumétrica é mais utilizada para visualização científica de dados, principalmente na área de medicina e meteorologia. A representação por superfície é mais usada nas áreas de computação gráfica interativa, realidade virtual e fotorealismo.

Existem vários métodos de representação de superfície geométricas [Par96]. A representação por **Superfícies Implícitas** utiliza funções em R^3 . A **Representação Paramétrica** utiliza três funções de duas variáveis paramétricas, sendo uma função para cada dimensão espacial. Normalmente estas funções são baseadas em polinômios quadráticos ou cúbicos. Alguns exemplos de representação paramétrica são as *B-splines*, curvas de Bézier, nonuniform rational B-spline (NURBS) e *B-splines* hierárquicas.

O método mais utilizado de representação geométrica de superfícies é o **Poligonal**, que representa a superfície através de um conjunto de polígonos.

Alguns sistemas de animação facial (como [Wan94] e [Tha91]) utilizam representação da geometria por *splines*. Em nosso sistema, utilizamos a representação por superfície poligonal através de triângulos. A representação por polígonos apresenta as seguintes vantagens sobre outros métodos de representação por superfícies:

- Existem várias técnicas de otimização de *hardware* e *software* para a projeção de triângulos na geração de imagens bidimensionais;
- Praticamente todos os *softwares* comerciais de modelamento permitem a descrição da geometria em polígonos, ou permitem transformar modelos descritos por *splines* em polígonos.
- É o método padrão de descrição de geometria na linguagem VRML.

4.2.2. Linguagens de Programação

Para implementar o sistema **InterFace**, utilizamos duas linguagens de programação: VRML [VRML96] e Java [Java95].

A linguagem **VRML** é um padrão aberto de objetos utilizado na descrição de cenas em três dimensões. A partir da versão 2.0, a linguagem VRML evoluiu muito, permitindo a descrição de cenas complexas em três dimensões, com animação de objetos, sensores de posição, de visibilidade e de proximidade, além de permitir a integração com outros programas através de um sistema de eventos. Através de um protocolo de comunicação com navegadores VRML [EAI97], é possível criar um programa em qualquer linguagem de programação que modifique parâmetros ou objetos da cena, desde que estes parâmetros ou objetos estejam preparados para receber e enviar eventos.

A linguagem Java, desenvolvida pela SUN Microsystems em 1992, possui várias características interessantes como linguagem para o desenvolvimento de sistemas complexos. Algumas destas características são apresentadas a seguir:

- Por ser uma linguagem orientada a objeto, a incorporação de novos recursos ao sistema pode ser feita facilmente, atendendo assim ao requisito de expansibilidade;
- O formato do arquivo executável, chamado de *byte-code*, é totalmente portátil entre diferentes plataformas. É possível executar um mesmo programa, sem a necessidade de recompilação, em praticamente qualquer sistema operacional;
- Vários conceitos são derivados do C++. Porém, algumas fontes comuns de complexidade nos programas, como ponteiros de memória e heranças múltiplas, foram eliminadas;

- A linguagem Java permite desenvolver muito facilmente programas que utilizem processamento paralelo, comunicação em rede e interface gráfica com o usuário;
- Vários navegadores *Web* possuem suporte à execução de aplicativos Java, facilitando a distribuição de programas pela Internet;
- Possui uma implementação do protocolo EAI (*External Authoring Interface*) para interface com cenas em VRML [EAI97].

Além de serem linguagens de descrição de cena e de programação poderosas, existem ainda outras vantagens da utilização de Java e VRML para o desenvolvimento de aplicações em três dimensões, especialmente comparando-se com outras linguagens e bibliotecas:

- Podemos separar a modelagem da cena, com a descrição de objetos, luzes e cores, do programa que gerencia o sistema;
- É possível criar em VRML objetos com comportamento, eliminando a necessidade de gerenciar este comportamento diretamente no sistema. Por exemplo, é possível criar em VRML uma casa onde ao clicar sobre a porta, esta se abre e depois de alguns segundos se fecha. Esta complexidade é eliminada da parte de programação do sistema, permitindo que tais parâmetros de comportamento possam ser facilmente criados ou modificados;
- O programa em Java é executado independentemente da cena em VRML; O programa em Java não precisa gerenciar a síntese (“*render*”) da cena, a taxa de quadros por segundo, a movimentação do observador e outros parâmetros comuns de cenas em três dimensões;

- A linguagem de programação Java, por ser orientada a objeto, atende ao requisito de permitir a fácil incorporação de novos recursos ao sistema.

Devido a estas vantagens apresentadas acima, optamos pela utilização das linguagens VRML e Java para o desenvolvimento do sistema **InterFace**.

4.2.3. Projeto do Sistema Interface

O sistema **InterFace** foi projetado para ter grande facilidade de uso, principalmente por usuários sem experiência em programação de computadores, porém com conhecimentos básicos em Computação Gráfica.

Na criação de animações em Computação Gráfica, normalmente são empregadas duas etapas: a criação dos elementos que compõe a cena, como personagens, objetos, luzes e câmeras, e a animação destes elementos. Olhando especificamente para a animação facial, estas duas etapas resumem-se a modelar o rosto humano e suas expressões e criar a animação entre estas.

O sistema **InterFace** oferece ferramentas para a criação de expressão de expressões faciais e animação interativa e procedural do rosto. Não há no sistema ferramentas para a modelagem em três dimensões do rosto humano, uma vez que existem diversos sistemas comerciais de alta qualidade para este fim.

Optamos por dividir o sistema **InterFace** em dois módulos, afim de facilitar a criação de expressões e de ações, uma vez que as ações são baseadas nas expressões previamente criadas. Separando os módulos, acreditamos que o trabalho do animador torna-se mais organizado e simples. Estes módulos são:

- Módulo para a **Criação de Expressões**, onde as expressões faciais básicas podem ser modeladas;
- Módulo de **Animação Facial**, onde é feita a simulação completa da animação do rosto do ator virtual.

Nos próximos itens descreveremos em detalhes os módulos.

4.3. Módulo de Criação de Expressões

Conforme descrevemos no capítulo III desta dissertação, no módulo de Criação de Expressões o usuário combina expressões da **Biblioteca Básica de Expressões** para criar novas expressões mais complexas.

Para a implementação deste módulo, temos de definir meios para resolver os seguintes problemas:

- Descrever o ator virtual em três dimensões;
- Descrever as expressões da Biblioteca Básica;
- Compor estas expressões em um único rosto;
- Armazenar as expressões criadas.

4.3.1. Descrição do Ator Virtual em Três Dimensões

O sistema **InterFace** utiliza a linguagem VRML para descrever a cena em três dimensões. Através do VRML é possível definir objetos através de primitivas (como esferas, cubos e cilindros) ou por faces (triângulos ou quadriláteros). O ator virtual deve ser modelado por algum *software* apropriado e exportado para VRML.

O rosto do ator virtual é normalmente composto de diversos objetos em três dimensões, como a face, os olhos, a boca e o cabelo. Estes objetos podem ser hierarquizados, para que as transformações de rotação, translação e escala sejam transmitidas dos objetos “pais” aos “filhos”.

O sistema **InterFace** exige que cada ator virtual da cena tenha um nome. O nome do ator não precisa estar referenciado em nenhum objeto ou conjunto de objetos da cena em particular. Este nome servirá apenas como referência para o conjunto de Expressões da Biblioteca Básica deste ator virtual.

4.3.2. Descrição de Expressões da Biblioteca Básica

Uma expressão da Biblioteca Básica pode ser entendida como transformações sobre um ou mais objetos que compõe o rosto do ator virtual. Há diversas maneiras para descrever estas transformações. A solução adotada no sistema **InterFace** é descrita pelo diagrama de blocos abaixo:

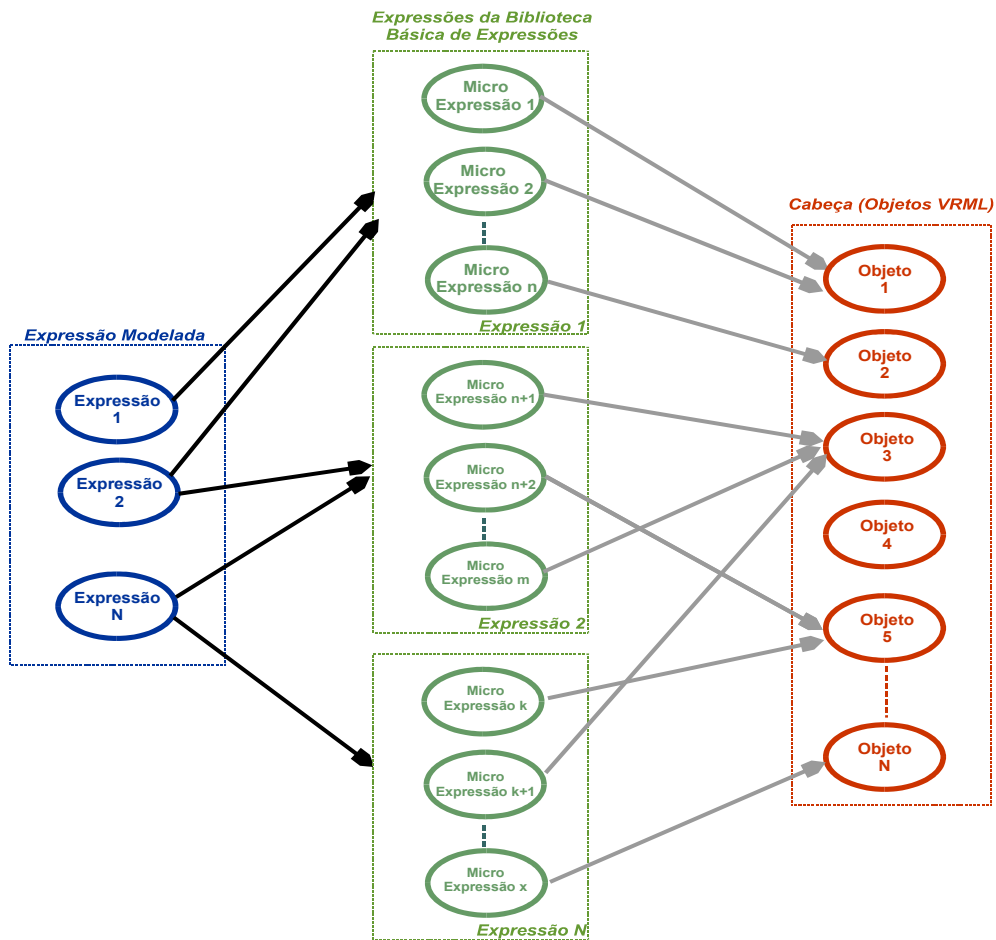


Figura 4.1: Diagrama esquemático das expressões da Biblioteca Básica

As expressões da Biblioteca Básica são compostas por **Micro-expressões**. Cada micro-expressão exerce pelo menos uma transformação sobre um único objeto. As expressões modeladas (também chamadas de expressões criadas) utilizam uma ou mais expressões da Biblioteca Básica. Explicaremos com detalhes este processo nos próximos itens.

4.3.2.1. As Micro-Expressões

Conforme descrevemos acima, a cabeça de um ator virtual é normalmente composta por diversos objetos, como os olhos, a face, a boca, os dentes e o cabelo. As **Micro-expressões** são transformações que operam sobre um único objeto. Na figura 4.1,

utilizamos as flechas cinzas para exemplificar que cada micro-expressão age sobre um único objeto do rosto, porém várias micro-expressões podem agir em um mesmo objeto. Exemplos de como descrever as micro-expressões no sistema **InterFace** podem ser encontrados no apêndice I desta dissertação.

As transformações que uma micro-expressão pode exercer sobre um objeto são:

- Rotação sobre o eixo X
- Rotação sobre o eixo Y
- Rotação sobre o eixo Z
- Rotação sobre um eixo arbitrário
- Translação
- Escala nos eixos X,Y ou Z;
- *Morph*- transformação de objetos através do deslocamento de vértices.

O sistema disponibiliza quatro meios para descrever rotações de objetos. Isto permite um maior controle do animador sobre o movimento exato que o ator virtual irá fazer. Para compor a rotação da cabeça, recomendamos que o modelo seja criado com três nós distintos, hierarquizados, na seguinte ordem: rotação no eixo Y, rotação no eixo X, rotação no eixo Z. Isto permite que o sistema **InterFace** possa executar as rotações da cabeça independentemente para cada eixo, além de permitir que cada eixo de rotação tenha seu respectivo centro.

Para exemplificar o uso da hierarquia de rotações, mostramos abaixo o código VRML do ator virtual Giggio, utilizado no projeto Aria II:

```
DEF giggioRotationY Transform {
```



```
rotation 0 1 0 0
children [
  DEF giggioRotationX Transform {
    rotation 1 0 0 0
    children [
      DEF giggioRotationZ Transform {
        center 0 -5 0
        rotation 0 0 1 0
        children [
          DEF giggio Transform {
            Children [
              DEF HEAD Transform {
                # Aqui entra a definição dos
                # objetos que compõe a cabeça
              }
            ]
          }
        ]
      }
    ]
  }
]
}
```

Tabela 4.1: Código VRML para a descrição do ator virtual Giggio

Notamos que a rotação no eixo Y tem precedência sobre a rotação no eixo X, e esta sobre a rotação no eixo Z. Esta ordem foi escolhida de forma que o movimento de rotação em torno do eixo Y possa ser corretamente executado independentemente do valor da rotação em torno do eixo X e Z, ou seja, independentemente do valor de rotação no eixo X e Z, a rotação em torno do eixo Y sempre ocorrerá com este eixo orientado “para cima”. Desta forma, o movimento de rotação da cabeça é natural, pois normalmente não ocorrem rotações de cabeça em torno do eixo Y quando o mesmo não está orientado “para cima”.

4.3.2.2. As Expressões da Biblioteca Básica

As expressões da Biblioteca Básica são compostas por uma ou mais micro-expressões. Isto permite descrever uma expressão da Biblioteca Básica que opere simultaneamente em diversos objetos.

Por exemplo, a expressão da Biblioteca Básica número 14 (movimento horizontal dos olhos) é composta de duas micro-expressões: uma exerce uma rotação em torno do eixo Y para o olho esquerdo e outra exerce a mesma rotação para o olho direito.

O sistema **InterFace** descreve as expressões da Biblioteca Básica na própria descrição das micro-expressões. A sintaxe das micro-expressões encontra-se no apêndice I desta dissertação.

4.3.3. Criação de Expressões

O sistema **InterFace** permite a criação de expressões faciais complexas através da combinação ponderada das expressões disponíveis pela Biblioteca Básica de Expressões. Cada expressão da Biblioteca Básica de Expressões é armazenada no sistema como uma série de vetores de distância entre a expressão neutra e as transformações nos objetos que compõe a expressão da Biblioteca Básica.

Digamos que a expressão número 7 da Biblioteca Básica (piscar o olho esquerdo) consista em executar uma rotação na pálpebra esquerda de 45 graus em torno do eixo X. O sistema **InterFace** irá armazenar esta expressão como um vetor de distância de 45 unidades na rotação do eixo X para o objeto pálpebra esquerda. Assim, quando esta expressão é exercida com valor 1, a pálpebra esquerda será rotacionada em 45 graus no eixo X. Ao executar esta expressão com valor 0.5, a pálpebra esquerda será rotacionada

em 22.5 graus. Da mesma forma, ao executar esta expressão com valor 1.5, a pálpebra será rotacionada em 67.5 graus, e com valor -1 a pálpebra rotacionará -45 graus.

Este conceito torna-se interessante quando duas expressões da Biblioteca atuam sobre os mesmos objetos. As seis primeiras expressões da Biblioteca Básica atuam sobre a boca. Ao serem combinadas, seus vetores de distância somam-se, permitindo a construção interativa de diversas expressões labiais diferentes. Utilizando apenas estas expressões, foi possível construir um sistema de sincronização de movimentos labiais com a voz (conhecido como *lip-sync*) para o ator virtual Giggio. As expressões labiais para *lip-sync* foram construídas baseando-se no trabalho de [Bla89], que definiu um conjunto de 7 expressões suficientes para descrever os fonemas em um personagem de animação (*cartoon*).

Usando a Biblioteca Básica, construímos diversas expressões faciais interessantes, mostradas a seguir:



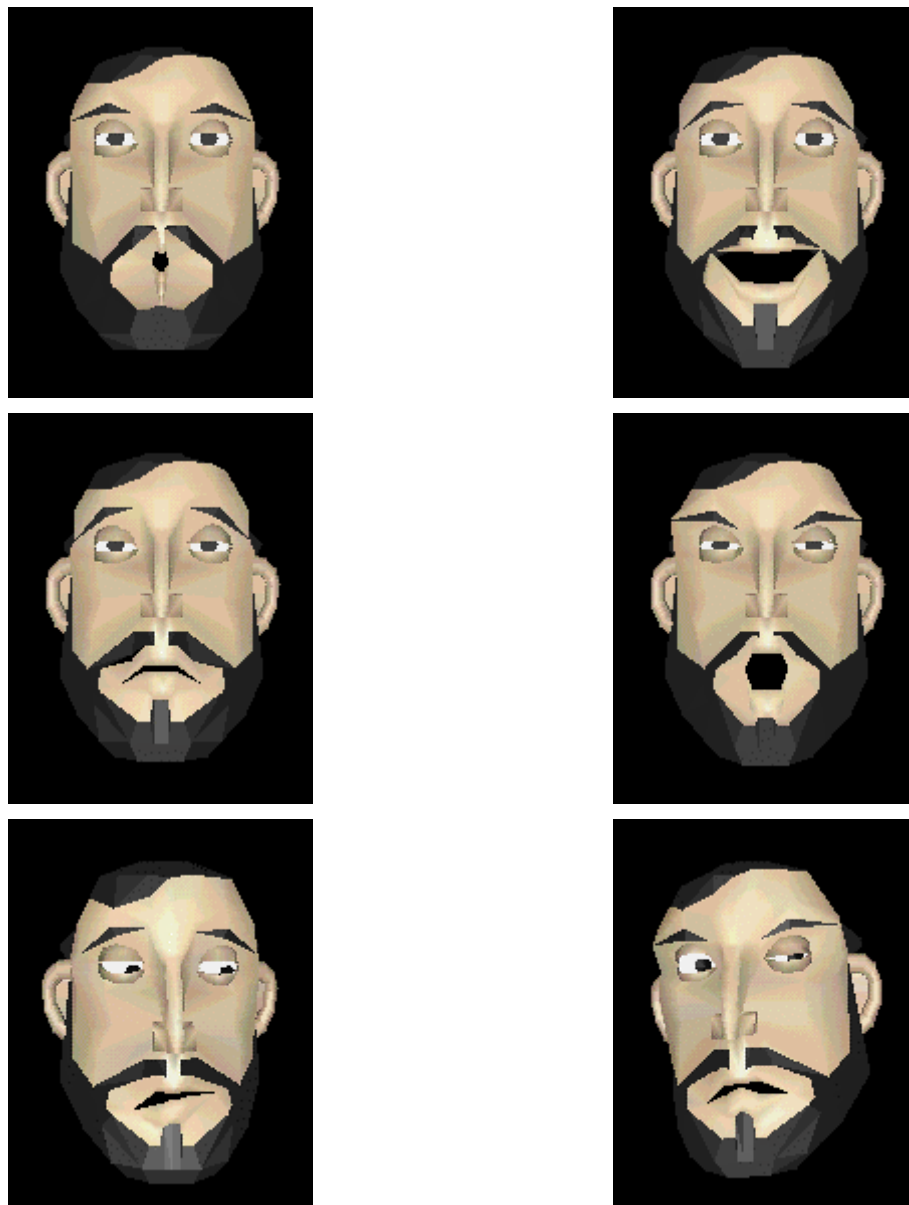


Figura 4.2: Expressões criadas no módulo de Criação de Expressões

4.3.4. Salvando as Expressões

A linguagem de programação Java possui uma série de recursos de segurança, afim de permitir que qualquer programa disponível na *World Wide Web* possa ser executado seguramente em um navegador *Web*. Um destes recursos de segurança é evitar que um aplicativo Java grave dados no disco do usuário.

A solução adotada no sistema **InterFace** para salvar as expressões criadas foi criar uma conexão em TCP/IP entre o aplicativo executado no navegador *Web* e o servidor para enviar os dados. O método escolhido foi codificar os dados através do método POST do protocolo HTTP [HTTP99], enviando-os a um programa em CGI no servidor *Web* que se encarrega de gravar estes dados no servidor.

Com esta solução, permitimos que qualquer pessoa conectada à Internet possa criar expressões e gravá-las no sistema.

4.4. Modulo de Animação Facial

Nesta seção apresentaremos o módulo de Animação Facial, onde o ator virtual executa as **Ações**. Conforme aludimos no capítulo III, ações são pequenas animações construídas utilizando-se as expressões criadas no módulo de Criação de Expressões.

As ações são reunidas em **Grupos de Ações**. Um Grupo de Ações reúne ações de um mesmo tipo, como emoções, movimentos labiais e movimentos dos olhos. Cada Grupo de Ações pode ter apenas uma ação ativa a um dado instante. Este mecanismo limita o número de ações a serem executadas simultaneamente, além de auxiliar a programação na camada de Inteligência do Ator Virtual, pois não é necessário preocupar-se com a execução simultânea de duas ações mutuamente exclusivas. Para combinar os Grupos de Ações, utilizamos um algoritmo semelhante ao utilizado em programas de sobreposição digital de imagens com transparência.

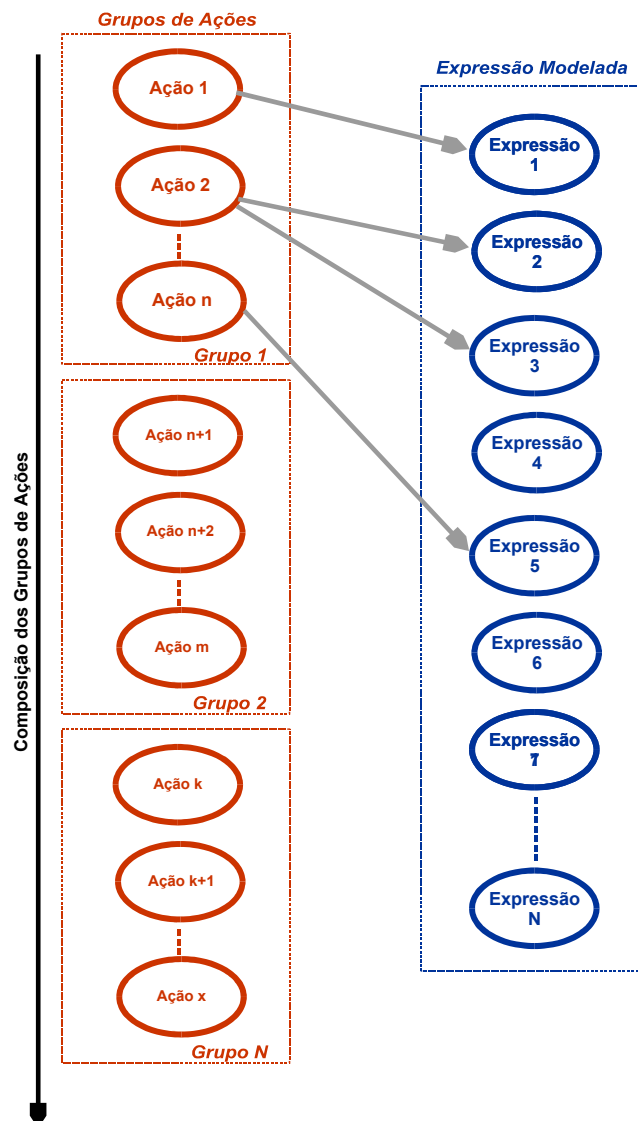


Figura 4.3: Diagrama esquemático do módulo de Animação Facial

Após a composição dos Grupos de Ações, o sistema **InterFace** transforma os valores de intensidade das expressões da Biblioteca Básica em eventos a serem enviados à cena em VRML. A todo este processo é dado o nome de **Composição de Eventos**.

4.4.1. Ações

Ações são criadas combinando-se as expressões da Biblioteca Básica, através da aplicação de valores variáveis com o tempo sobre estas expressões. Na figura 4.3, utilizamos as flechas cinzas para mostrar que uma ação pode utilizar uma ou mais

expressões modeladas, e que diversas ações podem utilizar as mesmas expressões modeladas.

A variação de valores de intensidade das expressões modeladas é feita através de funções matemáticas. No sistema **InterFace**, chamamos as funções de curvas, utilizando assim a nomenclatura comum dos sistemas de animação de Computação Gráfica.

Da mesma forma que as expressões da Biblioteca Básica são compostas de micro-expressões, as ações são compostas de **Micro-ações**. Cada micro-ação descreve a curva que atua sobre uma expressão modelada. Para facilitar a descrição da ação, cada curva é envolvida por um **Envelope**, afim de limitar seu tempo de atuação. A própria ação como um todo é envolvida por um envelope, afim dar a ela um tempo máximo de duração e de tornar a transição entre diversas ações mais suave.

Um exemplo simples de ação seria rotacionar a cabeça no movimento “Não”. A ação é descrita da seguinte forma:

Action nao 0.5 50 0.5 rotY 0 0 0 0 Sin 1.5

Tabela 4.2: Código exemplo de uma ação.

A ação é envolvida por um envelope cujo tempo de subida é de 0.5 segundos, o tempo de duração é de 50 segundos e o tempo de descida de 0.5 segundos. Ela é composta de uma única micro-ação, que modifica a intensidade de uma expressão criada no módulo de Criação de Expressões chamada **rotY**. Esta expressão nada mais é do que a expressão número 16 da Biblioteca Básica (rotação da cabeça em torno do eixo Y) com valor de intensidade 1. Nesta micro-ação, o valor da intensidade da expressão **rotY** varia segundo uma curva seno de período de 1.5 segundos. A sintaxe completa para a descrição de ações no sistema **InterFace** está no anexo I desta dissertação. Os conceitos de curvas, envelopes e Grupos de Ações estão descritos em detalhes nos próximos itens.

4.4.1.1. Curvas

O sistema **InterFace** disponibiliza diversas curvas para a construção de micro-ações:

Constante	
Descrição:	Um valor real constante
Parâmetros:	Valor.
Default:	1.0
Seno	
Descrição:	Função trigonométrica seno
Parâmetros:	Período, amplitude, <i>offset</i> no eixo X.
Default:	1.0, 1.0, 0.0
Coseno	
Descrição:	Função trigonométrica coseno
Parâmetros:	Período, amplitude, <i>offset</i> no eixo X.
Default:	1.0, 1.0, 0.0
Interpolação Linear	
Descrição:	Interpolação linear através de retas entre uma série de pontos X, Y.
Parâmetros:	$X_1, Y_1, X_2, Y_2, \dots, X_N, Y_N$ onde cada par X_i, Y_i é ponto da curva
Default:	Não há parâmetros default para esta curva
Spline	
Descrição:	Spline de Perlin, definida em [Per96a].
Parâmetros:	$X_1, Y_1, X_2, Y_2, \dots, X_N, Y_N$ onde cada par X_i, Y_i é ponto da curva
Default:	Não há parâmetros default para esta curva
Ruído	
Descrição:	Ruído de Perlin, definida em [Per85]. Esta curva é não-determinística, uma vez que seus valores não são conhecidos previamente, porém é controlável pelo período e um valor máximo para a amplitude. Além disso, a curva é contínua em C^2 .
Parâmetros:	Período, amplitude, <i>offset</i>
Default:	1.0, 1.0, 0.0
Impulso	
Descrição:	Impulso gerado em intervalos não-determinísticos, porém controláveis. Esta curva gera um impulso de amplitude 1.0 cada vez que uma curva de ruído Perlin ultrapassa um certo valor de referência.
Parâmetros:	Período (do ruído), amplitude (do ruído), valor de referência, , duração do impulso
Default:	1.0, 1.0, 0.0, 0.0

Tabela 4.3: Curvas disponíveis no sistema **InterFace**

A curva ruído de Perlin é importante para a criação de ações naturais. Piscar os olhos ou respirar são exemplos de ações que se beneficiam muito do uso de ruído, pois são ações que não ocorrem em intervalos de tempo exatos.

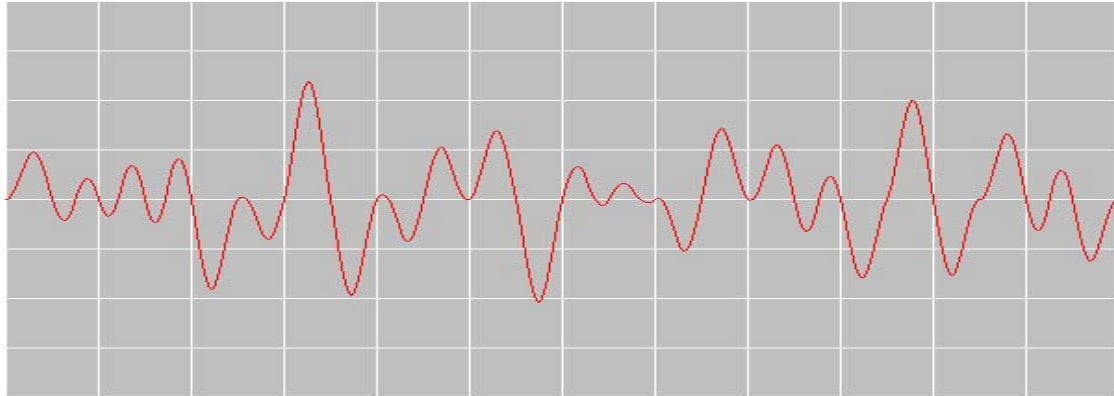


Figura 4.4: Curva de Ruído não determinístico de Perlin.

A *Spline* de Perlin possui uma característica interessante: a curva passa por todos os pontos que a compõe (a contrário de outros tipos de *splines*, como os NURBS). Assim, o animador obtém um maior controle sobre o movimento que compõe a ação, pois especifica todos os pontos-chave da ação precisamente.

A implementação de novas curvas no sistema **InterFace** é muito simples. Utilizando o princípio das linguagens orientadas a objeto, uma curva é uma classe abstrata, contendo um construtor que aceita um único parâmetro (um vetor contendo os dados da curva) e um método chamado *getValue* (`double time`), que retorna o valor da curva no tempo. A implementação de novas curvas consiste simplesmente na criação de uma nova classe que implementa a classe *Curve*.

4.4.1.2. Modificadores de Curvas

Modificadores são curvas que modificam a curva de uma micro-ação. O envelope que envolve uma micro-ação pode ser entendido como um modificador. O sistema

InterFace permite a implementação de outros modificadores de curvas. No momento, o único modificador implementado é o **Triggered_by**.

O modificador `Triggered_by` faz com que um micro-ação seja executada cada vez que uma curva (chamada de *Trigger*) passa de um valor negativo para positivo.

Digamos que a ação “respirar” exija que o ator virtual execute em intervalos não determinísticos a micro-ação de rotacionar levemente a cabeça em torno do eixo X. Uma forma de descrever esta micro-ação é apresentada abaixo:

```
rotx 0 1 3 1 sin 8 0.1 triggered_by impulso 1 1 0.15
```

Tabela 4.4: Exemplo do modificador de curva `Triggered_by`

Esta micro-ação será “disparada” somente quando o valor da curva `Impulso` for maior que zero, ou seja, somente quando ocorrer o impulso.

A implementação de novos modificadores, como multiplicadores, somadores ou saturadores de curvas, pode ser feita de maneira simples, devido à construção do sistema utilizando os princípios de orientação a objeto.

4.4.1.3. Envelopes

Conforme dissemos acima, a curva de uma micro-ação pode ser envolvida por um envelope. Isto facilita o controle que a curva irá exercer sobre a expressão modelada.

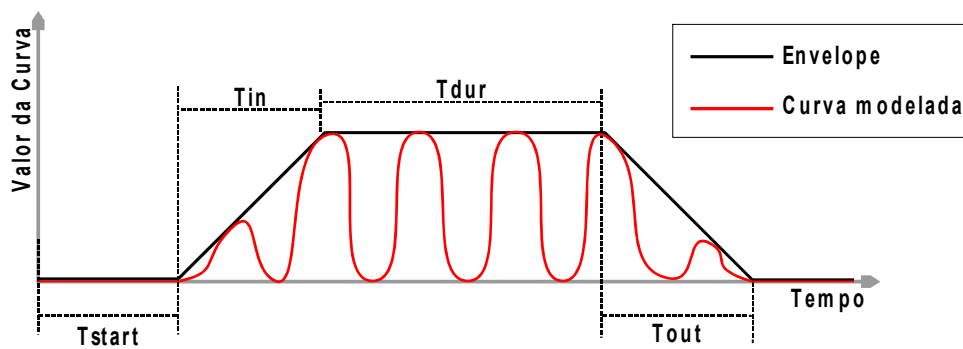


Figura 4.5: Envelope que envolve uma micro-ação

Durante o tempo de início T_{start} , o valor da expressão é nulo, independentemente do valor da curva. Durante o tempo T_{in} , o valor da curva é modulada por uma interpolação linear entre 0 e 1. Durante o tempo T_{dur} , o valor da curva não é modificado. Durante o tempo T_{out} , o valor da curva é modulada por uma interpolação linear entre 1 e 0. Após o tempo $T_{start}+T_{in}+T_{dur}+T_{out}$, o valor da curva será nulo.

A própria ação como um todo também é envolvida por um envelope. Este envelope contém apenas os tempos T_{in} , T_{dur} e T_{out} , uma vez que a quando a ação é disparada, é desejável que esta inicie a sua execução imediatamente, e portanto não faz sentido haver um tempo de início. Durante o tempo T_{in} , **todas as micro-ações** são moduladas por uma interpolação linear de 0 a 1. Da mesma forma, durante o tempo T_{out} , **todas as micro-ações** são moduladas por uma interpolação linear de 1 a 0.

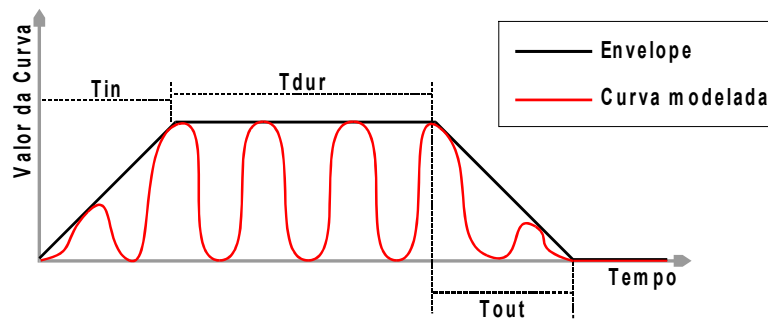


Figura 4.6: Envelope que envolve uma ação

4.4.1.4. Exemplo

Para entendermos melhor as ações, analisemos a ação “Espirrar”:

```

Action Espirro 0.1 9.6 0.3
  rotx 0 0 10 0 spline 0 0 3.5 1 4 -2 5 -2 7 0
  bocejo 0 3 0.5 0.5 constante 1
  pisca 3 0 2.5 2 constante 1
  assobio 3.5 0.5 1 2 constante 1
  roty 7 0.2 1 0.2 ruído 0.3 1
  rotz 7 0.2 1 0.2 ruído 0.3 0.6
  
```

Tabela 4.5: Código da ação Espirrar

A ação tem a duração total de 10 segundos, que é a soma dos tempos de subida, de duração e de descida. O tempo T_{in} da ação é de 0.1 segundos, o tempo de duração é de 9.6 segundos e o tempo T_{out} é de 0.3 segundos.

A expressão “rotx” atua durante grande parte da ação. No módulo de Criação de Expressões, modelamos “rotx” como uma rotação da cabeça de cerca de 30 graus para cima. Assim, a cabeça irá inclinar-se para cima durante os 3.5 primeiros segundos da ação, depois irá descer rapidamente em 0.5 segundo para uma inclinação de cerca de 60 graus para baixo (o momento do espirro), permanecerá por 1 segundo nesta posição e voltará gradativamente, durante 2 segundos, para a posição inicial, na qual ficará até o final da ação.

A expressão “bocejo” foi criada combinando-se as expressões 1 e 2 da Biblioteca Básica. O ator irá abrir gradativamente a boca num “preparativo” para o espirro durante os 3 primeiros segundos. Permanecerá assim durante 0.5 segundos e voltará ao estado inicial em 0.5 segundos. Após o quarto segundo da ação, esta micro-ação não exercerá mais influência.

A expressão “piscar” fecha ambos os olhos. Esta micro-ação que utiliza a expressão “piscar” tem um tempo de início de 3 segundos, ou seja, durante os 3 primeiros segundos, esta micro-ação não exercerá nenhuma influência na ação como um todo. Após este tempo, o ator fechará os olhos imediatamente e permanecerá assim durante 2.5 segundos, abrindo gradativamente nos próximos 2 segundos.

A expressão “assobio” também é uma combinação das expressões 1 e 2 da Biblioteca Básica. Ela começa a exercer influência na ação no momento do espirro, aos 3.5 segundos.

Por fim, as expressões “roty” e “rotz” agem nos 3 últimos segundos da ação, fazendo com que o ator balance um pouco a cabeça após o espirro.



Figura 4.7: Execução da ação “espirrar”

4.4.2. Grupos de Ações

As ações são reunidas em **Grupos de Ações**. Cada Grupo de Ações reúne em si ações semelhantes, que normalmente não são executadas simultaneamente. Por exemplo, as ações “Sim” e “Não” (movimento vertical e horizontal da cabeça, respectivamente) devem estar reunidas em um mesmo grupo.

Cada Grupo de Ação pode ter uma ou nenhuma ação **ativa** em um determinado instante. Quando uma ação torna-se ativa, esta inicia a sua execução e permanece ativa até o seu término ou se for interrompida pela ativação de outra ação do mesmo grupo ou por sua desativação.

Para que a transição entre ações seja suave, quando uma ação é desativada, ela imediatamente passa para o tempo de descida (T_{out}). Se a ação ativa foi desativada pela

ativação de uma outra ação do mesmo grupo, a transição entre as ações será suave, pois enquanto a ação desativada está diminuindo sua intensidade no tempo de descida, a ação ativada está iniciando sua influência, no tempo de subida. Isto garante a naturalidade na cena, sem interrupções bruscas.

Apresentaremos nos próximos itens a forma como os Grupos de Ações são compostos utilizando-se os conceitos de ordem de prioridade e transparência.

4.4.2.1. Ordem e Transparência

Conforme descrevemos no capítulo III, os Grupos de Ações possuem um índice de transparência e a ordem com que os Grupos são compostos pode levar a resultados finais diferentes.

A prioridade dos Grupos de Ações é crescente: os Grupos de Ações de maior índice exercem maior influência sobre o ator virtual que os Grupos de Ações de menor índice.

Cada Grupo de Ações possui um índice de transparência, denominado α . Se um Grupo de Ações tem um índice de transparência de 0.5, todos os valores de intensidade das expressões geradas pela ação ativa e eventualmente pelas ações que estão sendo desativadas naquele grupo são multiplicadas por 0.5. Isso é útil para a composição ponderada entre os Grupos de Ações.

Para exemplificar, digamos que uma ação ativa no grupo de índice 3 determinou que a expressão número 8 da Biblioteca Básica tenha um valor de 1.0. Se uma ação de um Grupo superior determinar que a mesma expressão tem uma intensidade de 0.5 e este grupo tiver um valor de α igual a 1.0, então na composição final, esta expressão terá

intensidade de 0.5 (caso nenhum grupo mais prioritário venha a modificar este valor). Caso o valor de α seja igual a 0.5, então a expressão terá intensidade de 0.75.

A ordem dos Grupos foi escolhida de modo a obedecer a prioridade das ações do ator virtual. Por exemplo, o Grupo de Ações não intencionais tem a menor prioridade. Assim, a ação “dormir” tem maior prioridade que a ação “piscar os olhos”, pois o ator não pode abrir os olhos enquanto dorme; da mesma forma, o grupo de emoções tem prioridade menor que o grupo de movimentos labiais, afim de permitir que o ator fale enquanto sorri ou chora.

4.5. Composição de Eventos

Em nossa implementação do algoritmo de composição dos Grupos de Ações (apresentado no capítulo III), o sistema inicialmente calcula e compõe as ações dos Grupos de Ações seguindo a ordem de prioridade destes Grupos e levando em conta o valor de transparência para cada grupo. O resultado deste acúmulo dos Grupos de Ações é transformado em valores de intensidade das expressões da Biblioteca Básica, e estes em valores de intensidade das micro-expressões, que por fim são enviadas como eventos para a cena em VRML. A este processo, desde a composição dos Grupos de Ações até o envio dos eventos, é dado o nome de **Composição de Eventos**.

O número de vezes com que a Composição de Eventos é executada deve ser o mais próximo possível do número de quadros gerados pelo navegador VRML no mesmo intervalo de tempo. Poderíamos gerar um número muito maior de Composição de Eventos do que o número de quadros gerados pela cena em VRML, uma vez que o navegador VRML ignora os eventos que não consegue executar, ao invés de guardá-los numa fila. Porém, isso faria com que o sistema **InterFace** executasse um processamento

maior que o necessário, utilizando inutilmente o processamento do computador, que poderia estar sendo utilizado em outras funções – como, por exemplo, gerar mais quadros da cena em VRML.

Para que o número de Composição de Eventos fosse próximo do número de quadros gerados pela cena em VRML, utilizamos um sistema de controle na Composição de Eventos, que consiste em inserir uma pequena pausa a cada composição completa. Durante esta pausa, o processador fica disponível para processar outros aplicativos. O intervalo de tempo desta pausa é controlado de acordo com o número de quadros da cena em VRML. Se o número de Composições de Eventos por segundo for maior que o número de quadros por segundo, o sistema aumenta o intervalo de tempo da pausa. Caso contrário, o sistema diminui este tempo.

4.6. A Inteligência do Ator Virtual

Como explicamos no capítulo III, na implementação do sistema **InterFace** há duas opções para a camada de inteligência do ator virtual: o usuário pode atuar como a inteligência do ator, ou outro sistema pode controlar o ator virtual através de um protocolo de comunicação.

4.6.1. O Usuário como Inteligência do Ator Virtual

O usuário pode agir como a inteligência do ator virtual, escolhendo as ações que o ator irá executar. A interface gráfica do sistema tem a disposição inicialmente um botão para cada ator virtual da cena. Ao clicar em um botão, o sistema abre uma janela contendo todos os Grupos de Ações com seus respectivos índices de transparência e a

ordem de composição destes Grupos para o ator escolhido. O usuário pode ativar interativamente a ação que ator irá executar, modificar a ordem dos grupos e modificar os índices de transparência.

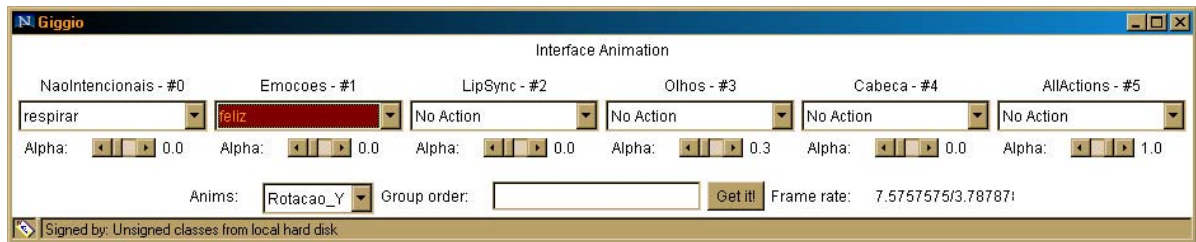


Figura 4.8: Interface para escolha de ações

4.6.2. Conexão Com Outros Sistemas

Uma das maneiras de expandir o sistema **InterFace** é permitir a conexão com outros sistemas. Desta forma, podemos criar diferentes implementações da camada de inteligência do ator virtual externamente, e utilizar o sistema **InterFace** para a implementação das outras camadas (Ações e Expressões).

O meio escolhido para conectar o sistema **InterFace** a outros sistemas é através de comunicação via rede. Na implementação do sistema **InterFace**, criamos um servidor em TCP/IP que potencialmente pode receber comandos de qualquer computador conectado à Internet. Este servidor recebe comandos através de um protocolo de comunicação bem definido, chamado de InterFace Protocol. Atualmente, o InterFace Protocol está na versão 1.0 e é denominado **IFP/1.0**.

Este protocolo é simples. O computador que conecta-se ao sistema **InterFace** (chamado de **Cliente**) inicialmente deve autenticar-se para solicitar outros serviços. Isto evita que usuários não autorizados enviem comandos ao sistema **InterFace**. Após a

autenticação, o cliente pode solicitar informações sobre a cena e os atores, como o nome dos atores, o nome dos Grupos de Ações, o nome e número das ações de cada Grupo, além do nome das expressões da Biblioteca modeladas de cada ator. Com base nestas informações, o cliente pode solicitar que o ator ative uma determinada ação de um determinado grupo, ou solicitar que o ator desative uma determinada ação ou a ação ativa (se houver) de um determinado grupo. O cliente pode ainda solicitar que o ator sorteie e execute uma ação de um determinado grupo. Por fim, o cliente pode solicitar o fechamento da conexão com o servidor.

Devido às especificações de segurança para a execução de aplicativos em Java no navegador *Web*, o sistema **InterFace** só permite o uso do servidor TCP/IP quando a máquina que o executa é a mesma onde as classes do sistema estão localizadas. Ou seja, se o sistema **InterFace** for carregado via um servidor *Web* remoto, a comunicação com outros aplicativos pelo protocolo IFP/1.0 será desabilitada.

A especificação completa do protocolo IFP/1.0 encontra-se no anexo II desta dissertação.

4.7. Animações em VRML

Aproveitando os recursos de animação da linguagem VRML, o sistema **InterFace** permite o ator virtual execute animações criadas diretamente em VRML por algum *software* de modelagem e animação comercial. O sistema **InterFace** controla a execução destas animações, permitindo que as mesmas possam ser executadas através do menu de opções ou por algum programa externo através do protocolo **IFP/1.0**.

Este é um recurso importante para a simulação de atores virtuais completos, uma vez que o sistema **InterFace** atua apenas na animação facial do ator. As animações do

corpo como andar, pular ou gesticular podem ser criadas diretamente no VRML e disparadas pelo sistema **InterFace**.

4.8. Conclusões

Há diversas maneiras de implementar um sistema de animação facial que utilize os conceitos apresentados no capítulo III desta dissertação. Para a implementação do sistema **InterFace**, optamos por criar um sistema interativo, em tempo real, intuitivo e expansível. Utilizando as linguagens VRML e Java, permitimos que o sistema **InterFace** pudesse ser executado em navegadores *Web*, disponibilizando-o através da rede mundial de computadores Internet.

No capítulo VI, verificaremos se esta implementação cumpriu os objetivos propostos e se os conceitos apresentados no capítulo III realmente puderam ser verificados na prática.

Capítulo V

Análise De Resultados

Neste capítulo iremos analisar o sistema **InterFace**, procurando verificar se o sistema proposto e desenvolvido cumpriu suas expectativas nos requisitos estabelecidos inicialmente. Analisaremos separadamente os conceitos do sistema e a sua implementação para cada um dos módulos. Apresentaremos também os comentários obtidos de usuários pela rede Internet, uma vez que o sistema está disponível para ser testado diretamente pela *World Wide Web*.

5.1. Metodologia Para Análise Dos Resultados

Para obter uma análise mais completa do sistema, estudaremos independentemente os dois módulos: o módulo de **Criação de Expressões** e o módulo de **Animação Facial**. Em cada um dos módulos, serão feitas duas análises: a análise conceitual e a análise do sistema.

5.1.1. Análise Conceitual

O objetivo da análise conceitual é validar os conceitos que envolvem o sistema, como a Biblioteca Básica de Expressões ou os Grupos de Ações. Estes conceitos foram apresentados no capítulo III desta dissertação.

5.1.2. Análise da Implementação do Sistema

Através da análise da implementação do sistema **InterFace**, pretendemos verificar se todos os requisitos propostos para o sistema, apresentados no início do capítulo IV desta dissertação, foram atendidos satisfatoriamente. Faremos também uma análise junto aos usuários, que apontarão os pontos positivos e negativos do sistema **InterFace**.

5.2. Análise do Módulo de Criação de Expressões

Nesta seção analisaremos o primeiro módulo do sistema: o módulo de **Criação de Expressões**.

5.2.1. Análise Conceitual

O módulo de Criação de Expressões utiliza os seguintes conceitos:

1. **Biblioteca Básica de Expressões:** um pequeno conjunto de expressões que ao serem combinadas permitem a criação de uma ampla gama de expressões emocionalmente significativas.
2. **Combinação de Expressões:** as expressões faciais são criadas através da soma e subtração ponderada de expressões da Biblioteca Básica. Estas expressões podem ser extrapoladas, ou seja, podemos utilizar valores de intensidade superiores a 1 e inferiores a 0.

A implementação do sistema **InterFace** nos permitiu demonstrar que o conjunto de expressões da Biblioteca Básica é eficiente na criação de novas expressões faciais. Através da combinação das expressões da Biblioteca Básica, conseguimos criar as expressões faciais desejadas, desde expressões de emoção, como tristeza, alegria, susto

ou medo, expressões labiais para sincronização de voz, movimento dos olhos para todas as direções e todos os movimentos de rotação da cabeça. Apresentamos algumas expressões criadas neste módulo no capítulo IV desta dissertação.

O uso de extrapolação nos valores de intensidade das expressões da Biblioteca Básica abriu novas possibilidades para a criação de muitas expressões faciais. Por exemplo, valores negativos pequenos nas expressões 7 e 8 da Biblioteca Básica (piscar os olhos) fizeram com que os olhos do ator ficassem mais abertos, ajudando a criar uma expressão de susto. Valores negativos das expressões 9 e 10 (rotação das sobrancelhas) foram essenciais na criação da expressão de raiva. Valores superiores a 1 nas expressões 1 e 2 (movimento labial em forma de “ahh” e “ohh”) auxiliaram nas expressões de sincronização de voz no projeto Aria, onde o ator virtual “canta” uma ópera.

Com a extrapolação, algumas expressões da Biblioteca Básica puderam ser usadas de formas não planejadas inicialmente. Um valor negativo para a expressão 1 (“ahh”) somado a um valor positivo, também extrapolado, da expressão 2 (“ohh”) criaram a expressão de “assobio”.

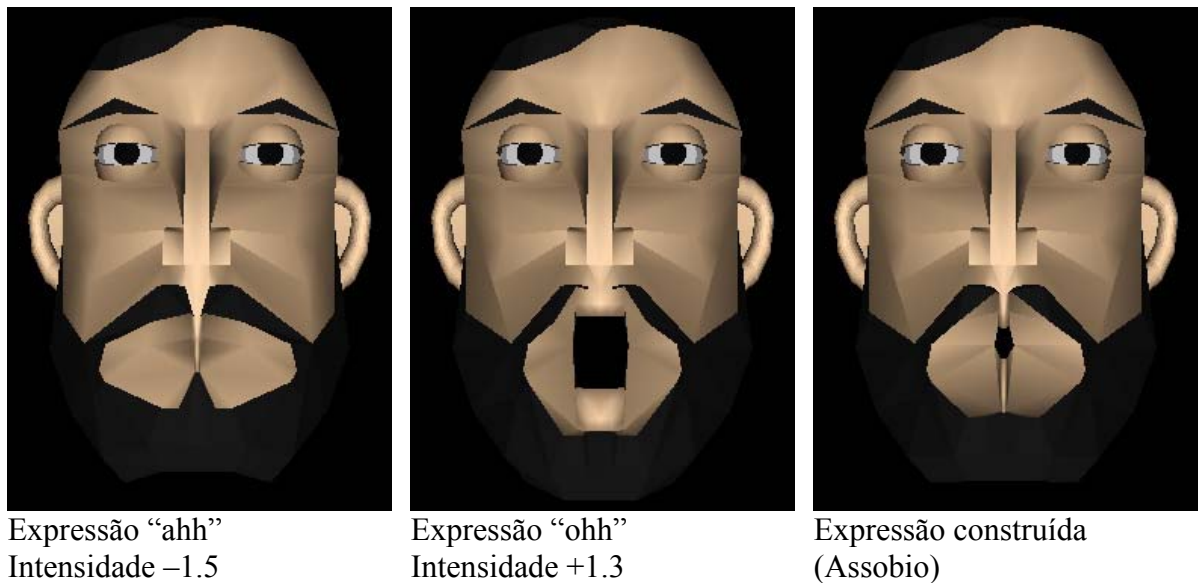


Figura 5.1: Composição da expressão facial “assobio”

5.2.2. Análise do Sistema

Os requisitos de implementação do sistema **InterFace** são:

- Intuitividade;
- Interatividade;
- Operação em tempo real ;
- Expansibilidade.

Em nossa análise, este módulo apresentou-se fácil de usar. Não enfrentamos nenhuma dificuldade na criação de expressões. A interface gráfica com o usuário é simples e intuitiva, baseando-se em um conjunto de barras de rolagem (*sliders*), que controlam o valor da intensidade de cada expressão da Biblioteca Básica. Os valores máximos das barras de rolagem são de -5 a +5. Apesar de que raramente estes valores extremos são utilizados, foi necessário abrir esta grande faixa pois em algumas expressões atingimos valores próximos ao limite. Além das barras de rolagem, temos um

menu com as expressões já criadas, um botão para salvar a expressão atual e um botão para que o ator volte à expressão neutra. Todas as opções referentes ao ator encontram-se em uma única janela, facilitando o acesso às opções.

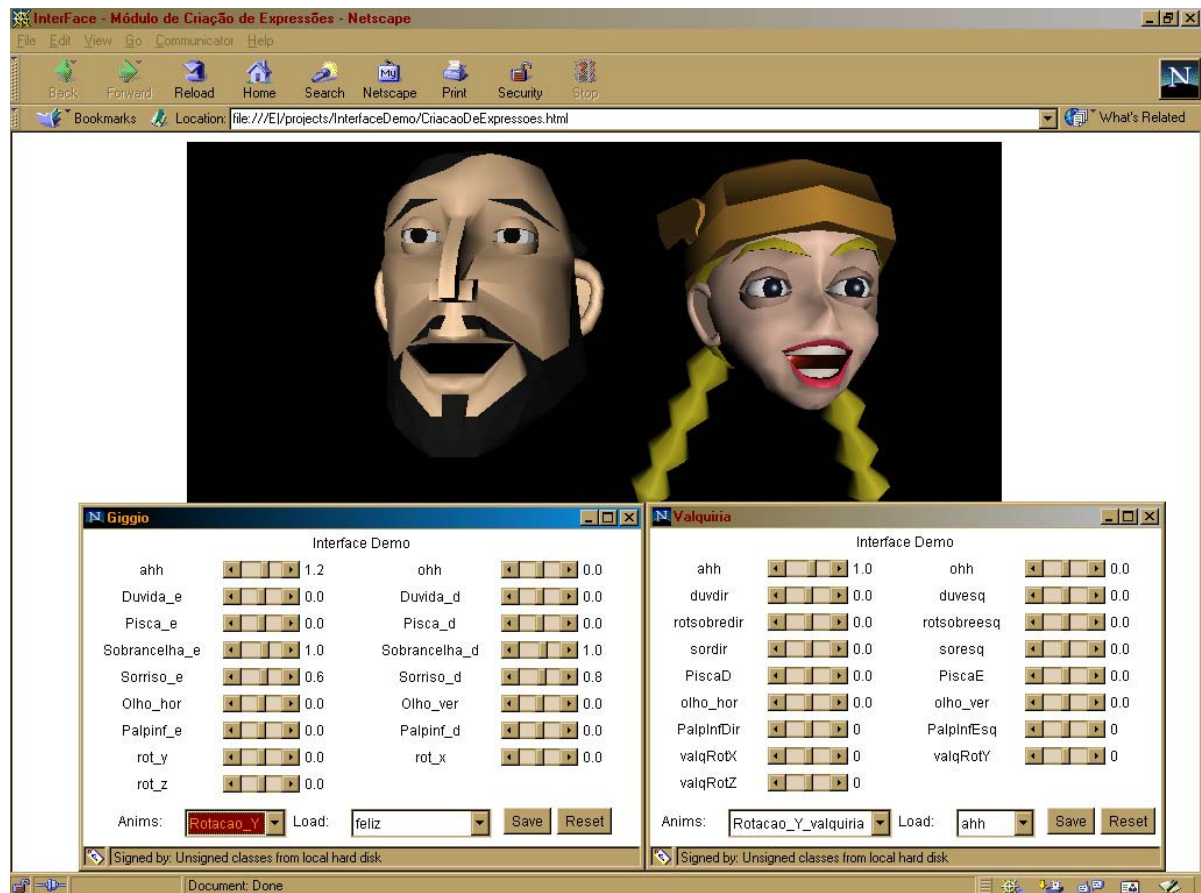


Figura 5.2: A interface com o usuário do módulo de Criação de Expressões

O teste dos requisitos de interatividade e tempo real é simples, consistindo apenas em verificar se a modificação do valor de intensidade de uma expressão da Biblioteca Básica é apresentada imediatamente no ator virtual. Como o cálculo de uma única expressão da Biblioteca Básica envolve uma quantidade de processamento relativamente pequena, obtivemos interatividade em todos os computadores que testamos o sistema. O computador de menor capacidade de processamento utilizado no teste foi um Pentium

133 MHz, com uma placa de vídeo simples (sem recursos de aceleração para o cálculo de imagens em três dimensões).

O módulo de Criação de Expressões atende ao requisito de expansibilidade por ter sido criado em uma linguagem orientada a objeto, permitindo a fácil incorporação de novos recursos. O módulo de Criação de Expressões não necessita do recurso de conectar-se a outros sistemas, vez que o objetivo deste módulo é apenas criar as expressões que serão usadas pelo módulo de Animação Facial.

O sistema foi disponibilizado na rede Internet para que usuários de todo o mundo pudessem testar e emitir suas opiniões. Através destes testes, obtivemos os seguintes comentários:

Aspectos positivos:

- Aprendizado rápido;
- Facilidade na criação de expressões;
- As expressões já armazenadas facilitam a criação de novas expressões.

Aspectos negativos:

- O sistema não opera com todos os navegadores;
- O número de barras de rolagem é grande;
- Nos atores não foram modeladas as línguas.

Sendo que o sistema **InterFace** foi implementado como uma prova de conceito, não foi possível projetá-lo para operar em todos os navegadores. Tal mudança seria trabalhosa sem grandes ganhos para a pesquisa.

5.3. Análise do Módulo de Animação Facial

Nesta seção analisaremos o segundo módulo do sistema **InterFace**: Animação Facial.

5.3.1. Análise Conceitual

O módulo de Animação Facial utiliza os seguintes conceitos:

1. Ações
2. Agrupamento de Ações
3. Composição dos Grupos de Ações com transparência.

Conforme descrevemos no capítulo III desta dissertação, as ações são pequenas animações que o ator virtual sabe fazer. É através das ações que o ator virtual pode expressar-se.

As ações criadas no sistema **InterFace** permitiram ao ator virtual uma grande liberdade de expressão e atuação. Pudemos criar ações em todas as áreas expressivamente significativas da animação facial: emoções, movimentos da cabeça e dos olhos, movimentos labiais e ações não-intencionais. Em particular, as ações não-intencionais, como respirar e piscar os olhos, foram essenciais para dar a impressão de que o ator virtual é um personagem vivo. O uso de ruído não determinístico foi importante para tornar estas ações mais naturais.

Um problema enfrentado nas ações é a falta de sincronismo ao modificar os valores de intensidade das expressões da Biblioteca Básica. Como os eventos que compõe as modificações da Biblioteca Básica são enviados à cena em VRML seqüencialmente, podemos perceber uma pequena ausência de sincronismo na ação. Por exemplo, a ação de

pisar os olhos, com pequena frequência, ocorre fora de sincronismo: um olho pisca frações de segundos antes do outro. Quando ocorre, este evento é facilmente perceptível. Pelo fato da sincronização perfeita de eventos em VRML ser uma tarefa complexa, este problema não pode ser resolvido nesta primeira versão do sistema. No capítulo VI apresentaremos as alternativas para a sua solução.

O objetivo dos Grupos de Ações é auxiliar o desempenho do sistema, limitando o número de ações a serem executadas simultaneamente. Uma das nossas preocupações era que, ao limitar o número de ações a serem executadas simultaneamente, as possibilidades de atuação do ator virtual também ficassem limitadas. Mas a escolha dos grupos de ações permitiu que de fato não houvesse necessidade do ator realizar duas ações de um mesmo grupo simultaneamente em praticamente nenhum caso. O último grupo de ações – que contém todas as ações dos grupos anteriores – praticamente nunca foi usado. Assim, o conceito de grupos de ações demonstrou ser uma boa solução para auxiliar o desempenho do sistema sem sacrificar a atuação do ator virtual.

Para compor a expressão final do ator, utilizamos o algoritmo de composição com transparência descrito no capítulo III desta dissertação. A implementação do sistema **InterFace** demonstrou que a transparência nos grupos praticamente não é utilizada, por dois motivos:

1. As ações são criadas para serem executadas com sua intensidade total. Ao executar uma ação em um Grupo com valor de transparência diferente de zero (totalmente opaco), o resultado normalmente não apresenta-se natural;
2. Os grupos de ação foram escolhidos em uma determinada ordem, de forma a permitir que a sobreposição totalmente opaca de um grupo sobre o outro resulte em uma expressão natural. Por exemplo, o grupo de emoções tem

menor prioridade, pois é natural que as ações de sincronização labial e movimento dos olhos sobreponha ações como sorrir ou chorar.

A utilização de transparência nos grupos de ação será remodelada na próxima versão do sistema **InterFace**. No capítulo VI desta dissertação descreveremos esta e outras melhorias planejadas para o sistema.

5.3.2. Análise da Implementação

Os requisitos estabelecidos para a implementação do sistema **InterFace** são:

- Intuitividade;
- Interatividade;
- Operação em tempo real;
- Expansibilidade.

O módulo de Animação Facial não é totalmente intuitivo, principalmente para quem não conhece os conceitos do sistema **InterFace**. Ao abrir a janela com as opções do ator, a interface de usuário apresenta para cada Grupo de Ações um menu com as ações deste grupo e uma barra de rolagem, na qual o usuário controla o valor da transparência. Há também um menu com as animações em VRML do ator e um campo para entrar a ordem de prioridade dos grupos. Por fim, no campo inferior esquerdo são mostradas informações sobre o desempenho do sistema: o número de quadros gerados por segundo na cena em VRML e o número de Composição de Eventos por segundo.

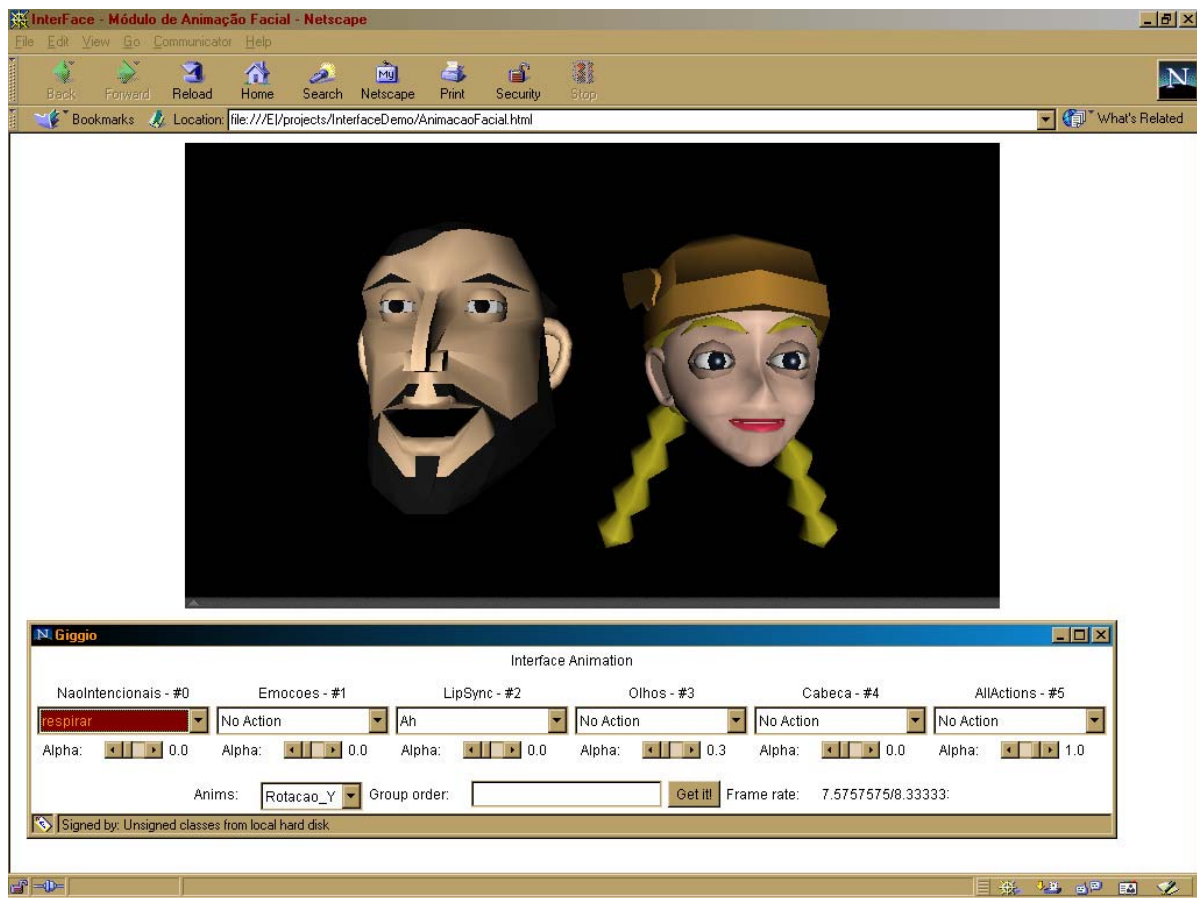


Figura 5.3: A interface com o usuário do módulo de Animação Facial

Como vemos, esta interface com o usuário é um pouco complexa. Porém, através dela o usuário tem controle total sobre a inteligência do ator virtual.

Em nossos testes, o sistema demonstrou ser interativo. Em todas as máquinas testadas, não foi notado nenhum atraso perceptível entre a escolha da ação e a sua execução. Nos projetos Aria II e Aria III, o sistema **InterFace** demonstrou ser interativo também quando é controlado remotamente por outro sistema.

Apesar da execução interativa, o módulo de Animação Facial do sistema **InterFace** não pode operar em tempo real em todas as configurações de máquinas testadas. Diversos fatores podem alterar o desempenho, como o número de polígonos da cena, a quantidade de atores virtuais e a complexidade na descrição das expressões da

Biblioteca Básica. Porém, pela nossa observação, o principal fator limitador é o envio dos eventos do aplicativo Java para a cena em VRML.

A sobrecarga de eventos VRML tornou impossível uma análise quantitativa do desempenho. Mesmo com um número de quadros gerados por segundo na cena em VRML e o número de Composições de Eventos por segundo constantes, notamos que quando o número de eventos são enviados em quantidade à cena em VRML, o sistema apresenta pequenas interrupções no processamento destes eventos, criando um efeito indesejável no ator virtual que denominamos espasmo. Acreditamos que o problema esteja no fato que o navegador VRML ignora alguns eventos se a quantidade de eventos enviados simultaneamente for muito grande. Como os números que podemos mensurar são a quantidade de quadros por segundo gerados pelo navegador VRML e a quantidade de Composição de Eventos por um determinado intervalo de tempo, não foi possível mensurar quantos eventos estão sendo ignorados e qual o intervalo de tempo gasto quando os eventos são ignorados. Por uma observação visual qualitativa, a única configuração de máquina testada na qual o atraso no processamento de eventos não é visivelmente perceptível foi um computador com um processador Pentium II 450 MHz e placa de vídeo com acelerador gráfico para três dimensões Riva TNT. No capítulo VI desta dissertação apresentaremos as possíveis soluções para aumentar o desempenho em tempo real do sistema **InterFace**.

O sistema **InterFace** permite uma integração simples com outros sistemas, através de um esquema cliente/servidor que utiliza um protocolo de comunicação bem definido, o IFP/1.0 (InterFace Protocol, versão 1.0). O sistema **InterFace** foi utilizado com sucesso no projetos Aria II e Aria III, no qual um programa de síntese de voz atuou como a camada de inteligência do ator virtual, enviando para o sistema **InterFace** as ações para a

sincronização de voz e emoções do ator virtual. No anexo II desta dissertação, apresentamos a sintaxe completa do protocolo IFP/1.0.

5.4. Conclusão

A implementação do sistema **InterFace** foi bem sucedida em praticamente todos os requisitos propostos no Capítulo IV. Consideramos que os conceitos apresentados no Capítulo III desta dissertação foram verificados como válidos por esta implementação.

O principal problema detectado é que, sob determinadas condições, ocorrem espasmos no ator virtual devido à pausas esporádicas do processamento das ações, especialmente em computadores de capacidade de processamento não muito alta e quando a quantidade de eventos enviada simultaneamente para a cena em VRML é grande. No capítulo VI desta dissertação, apresentaremos algumas possíveis soluções para este problema.

Capítulo VI

Conclusões e Projetos Futuros

Neste capítulo apresentaremos os resultados obtidos a partir do desenvolvimento do sistema **InterFace**: a pesquisa, o projeto, a implementação, as publicações relativas ao sistema e aplicações relevantes.

Apresentaremos também os projetos futuros e as principais áreas de pesquisa relacionadas com a tecnologia implementada em nosso sistema.

*6.1. Resultados do Sistema **InterFace***

Nesta seção apresentaremos os resultados obtidos pela sistema **InterFace**. Apresentaremos os conceitos que consideramos aprimorados pela nossa pesquisa, a implementação e aplicações do sistema **InterFace** e por fim apresentaremos as publicações realizadas durante o desenvolvimento desta dissertação.

6.1.1. Contribuições Relevantes

A primeira contribuição relevante de nosso trabalho foi introduzir esta área de pesquisa no Laboratório de Sistemas Integráveis da Escola Politécnica da USP (LSI), envolvendo a cooperação internacional do Prof. Dr. Ken Perlin, do Media Reseach Laboratory da New York University.

Para realizar a implementação do sistema **InterFace**, pesquisamos a literatura, selecionando e aprimorando os conceitos mais relevantes para os nossos objetivos,

resultando em um sistema de animação facial tridimensional interativo e em tempo real. Nos próximos parágrafos, apresentaremos estes conceitos que acreditamos serem importantes para uma implementação eficiente de um sistema de animação facial interativo.

6.1.1.1. Biblioteca Básica de Expressões

O conceito da Biblioteca Básica de Expressões, conforme apresentamos no capítulo III desta dissertação, é um conjunto pequeno de expressões simples, facilmente modeláveis, que permite ao animador criar expressões faciais complexas.

Apesar de um conceito semelhante já ter sido apresentado no trabalho de Parke em 1982 [Par82], um conjunto bem definido de expressões só foi proposto por Perlin em 1997 [Per97]. Em seu sistema, Perlin demonstrou que as expressões da Biblioteca Básica de Expressões são eficientes para um ator virtual de duas dimensões.

Em nosso trabalho, pudemos demonstrar que um conjunto levemente expandido das expressões de Perlin é suficiente para a criação de expressões faciais complexas em um ator virtual de três dimensões.

6.1.1.2. Extrapolação de Expressões

O conceito de extrapolação de expressões da Biblioteca Básica de Expressões, apresentado no capítulo III desta dissertação, foi muito importante para a criação de expressões faciais complexas. De fato, não seria possível criar a maioria das expressões utilizadas em nossas demonstrações sem o uso de extrapolação.

Assim, através do nosso sistema, pudemos demonstrar que para modelos de três dimensões, o uso de extrapolação de expressões da Biblioteca Básica de Expressões é uma técnica promissora para a criação de expressões faciais de forma simples e rápida.

6.1.1.3. Ações e Grupos de Ações

O principal conceito do módulo de Animação Facial de nosso sistema são as ações. Ações são pequenas animações que o ator virtual pode realizar. As ações são construídas pela variação, através de funções matemáticas, do valores de intensidade das expressões criadas pelo módulo de Criação de Expressões.

As ações são reunidas em Grupos de Ações de uma mesma categoria. Em cada Grupo de Ações pode haver uma única ação ativa a um dado instante.

O agrupamento das ações permitiu aumentar o desempenho do sistema e facilitou a integração com outros sistemas, uma vez que não é necessário desativar a ação ativa de um grupo para ativar outra ação deste mesmo grupo. Os Grupos de Ações foram construídos de forma que raramente houve a necessidade que duas ações de um mesmo grupo ficassem ativas ao mesmo tempo. Mas para permitir que este raro evento seja possível de ser realizado, o sistema cria automaticamente o último Grupo de Ações, que contém todas as ações de todos os grupos anteriores.

A composição dos Grupos utilizou um algoritmo inspirado em programas de composição digital de imagens. Cada Grupo contém um valor para a transparência, e a composição das ações ativas de cada grupo leva em conta este valor para gerar a expressão resultante da composição a cada instante.

A utilização de Grupos de Ações e a composição com transparência permitiu que o ator virtual pudesse ser facilmente animado. Esta composição torna simples a criação

de animações que seriam trabalhosas por métodos tradicionais de animação. Um exemplo de composição de ações que resulta numa animação complexa é falar e expressar uma emoção simultaneamente.

6.1.2. Implementação do Sistema

Para provar a validade dos conceitos que propusemos, realizamos a implementação do sistema **InterFace**. Planejamos o sistema **InterFace** para ser um sistema interativo, operando em tempo real, fácil de usar e expansível, o que se provou de fato a partir da análise realizada no capítulo V.

A implementação do sistema utilizou as linguagens VRML e Java. Esta escolha nos permitiu disponibilizar o sistema **InterFace** diretamente na rede Internet, através de navegadores *Web*. Usuários de todas as partes do mundo puderam utilizar e avaliar o sistema diretamente em seus navegadores *Web*, sem a necessidade de instalação de outros *softwares*.

6.1.3. O Projeto Aria

O sistema **InterFace** foi utilizado em duas versões do projeto Aria, desenvolvido no Laboratório de Sistemas Integráveis da Escola Politécnica da USP. O projeto Aria [Aria97] consiste na interação de atores virtuais com o usuário, implementando uma apresentação de uma ária de ópera. O usuário age como um maestro, controlando o ritmo da música que está sendo executada e encenada pelo o ator virtual através de uma batuta que contém um rastreador de posição.

Inicialmente, o projeto Aria foi apresentado no congresso Siggraph em 1996. Nesta primeira implementação, o sistema utilizado para simular os atores virtuais foi o

Improv, desenvolvido por Ken Perlin em conjunto com o LSI [Per96a]. A animação facial do ator virtual neste estágio foi rudimentar, limitando-se a aplicar valores de intensidade a expressões previamente modeladas em um sistema comercial.

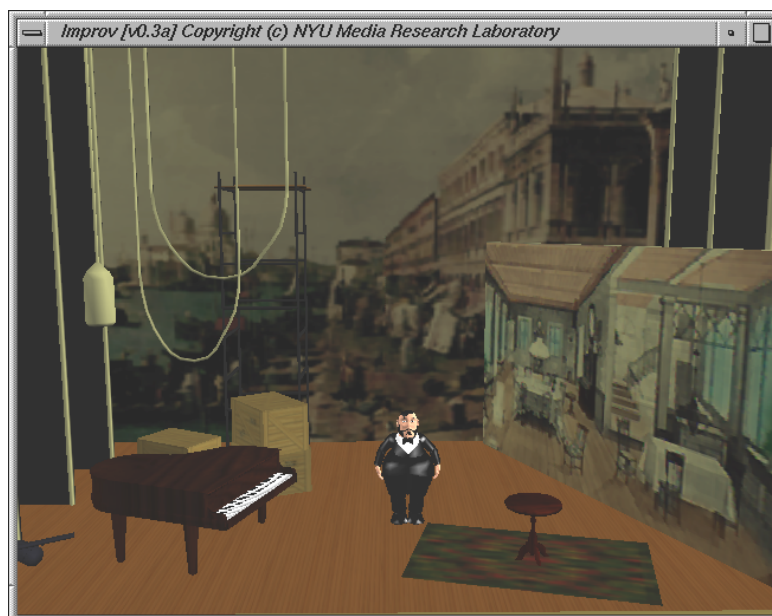


Figura 6.1: A primeira versão do projeto Aria

Esta primeira implementação do projeto Aria inspirou-nos a aprofundar nossa pesquisa na área de animação facial. Em sua segunda versão, o projeto Aria utilizou o sistema **InterFace** como base para a simulação dos atores virtuais. Fizemos uma simulação com dois atores virtuais, respectivamente uma soprano e um tenor, operando em tempo real em um computador Pentium PRO 350mHz. O ganho de desempenho do sistema **InterFace** em relação a primeira versão do Aria foi significativo, uma vez que a primeira versão requeria uma Estação de Trabalho Gráfica de alto processamento. O projeto Aria II foi apresentado na feira Comdex em São Paulo, em setembro de 1997.

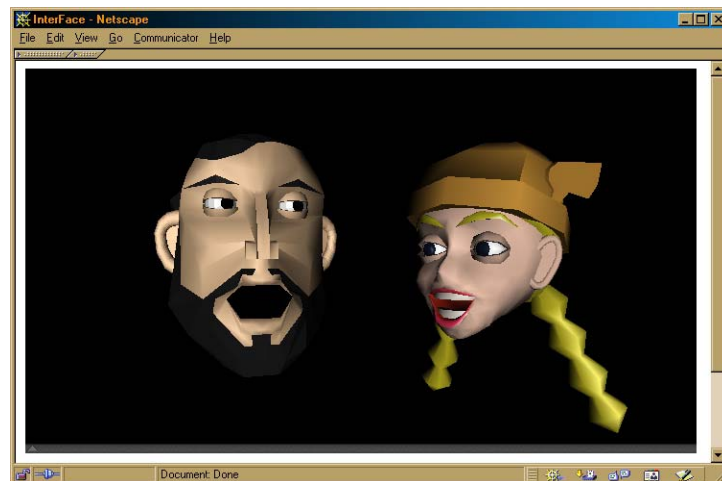


Figura 6.2: O projeto Aria II

Para o projeto Aria III, fizemos uma reformulação no sistema **InterFace**, afim de suportar atores virtuais completos. Para lidar com a animação do corpo do ator, implementamos o suporte a animações criadas em VRML. O projeto Aria III foi exposto no Centro Itaú Cultural, em São Paulo, de agosto a outubro de 1999.



Figura 6.3: O projeto Aria III

6.1.4. Publicações

O sistema **InterFace** foi tema de um artigo publicado em congresso internacional. O artigo *InterFace – A Real Time Facial Animation System*, de autoria de José Daniel Ramos Wey e João Antonio Zuffo, foi apresentado no congresso Sibgrapi (Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens) em 1998 [Wey98]. O sistema **InterFace** também foi descrito no artigo *Some Experiences Implementing Virtual Worlds: The Aria Project*, apresentado em 1997 no congresso IFIP [Aria97].

O artigo *Improv: A System for Scripting Interactive Actors in Virtual Worlds*, apresentado por Ken Perlin no congresso Siggraph em 1996, contém uma referência ao trabalho de animação facial implementado pelo candidato na primeira versão do projeto Aria.

6.2. Trabalhos Futuros e Áreas de Pesquisa

Nesta seção apresentaremos os nossos planos futuros com relação ao sistema **InterFace**. Inicialmente mostraremos os aprimoramentos planejados para o sistema em si e posteriormente apresentaremos os projetos futuros de pesquisa do Laboratório de Sistemas Integráveis que estão relacionados ao sistema **InterFace**.

6.2.1. Aprimoramentos ao Sistema InterFace

Conforme foi discutido no capítulo V desta dissertação, o sistema InterFace foi implementado como prova de conceito. O sistema mostrou-se estável e foi utilizado com sucesso em outros projetos no Laboratório de Sistemas Integráveis.

Os problemas encontrados na implementação atual do sistema **InterFace** são:

- A falta de sincronismo ao modificar os valores de intensidade das expressões da Biblioteca Básica;
- O sistema não pode operar em tempo real em todas as configurações testadas;
- Falta de portabilidade no sistema entre diversos navegadores *Web* e VRML.

Estes problemas estão relacionados com a escolha das plataformas de desenvolvimento do sistema. A integração entre as linguagens Java e VRML permite o desenvolvimento rápido e fácil de aplicações em três dimensões, sendo ideal para a construção do sistema como prova de conceito. Porém, o desempenho de aplicações em Java nos navegadores *Web* ainda é bastante inferior se comparado com outras linguagens, como ANSI 'C' ou C++. Além disso, a falta de aderência estrita dos navegadores VRML ao padrão ISO de comunicação de cenas VRML com outros programas, chamado de

External Authoring Interface ou EAI [EAI97], não permite uma portabilidade fácil entre diversas plataformas.

A solução será a portabilidade do sistema para outra plataforma de desenvolvimento. Uma escolha interessante é utilizar a classe de programação em três dimensões da linguagem Java, chamada de Java3D. Uma aplicação desenvolvida em Java3D tem a vantagem de manter compatibilidade entre diversas plataformas, e não apresenta uma perda de desempenho significativa com relação a outras linguagens, uma vez que há diversas máquinas virtuais Java de alto desempenho disponíveis para aplicativos Java (programas que não são executados diretamente no navegador *Web*).

Uma possível expansão para o sistema **InterFace** é procurar um melhor aproveitamento no uso da transparência na composição dos Grupos de Ações. Conforme apresentamos no Capítulo V, a transparência dos Grupos de Ações praticamente não é utilizada no sistema atual. Pretendemos implementar outros algoritmos utilizados para a composição de imagens com transparência na composição dos Grupos de Ações afim de obter uma composição mais natural entre os grupos. Outro recurso interessante seria permitir ao sistema **InterFace** controlar a transparência dos Grupos de Ações. Assim, quando a ação ativa de um grupo estivesse terminando sua execução, o grupo tornar-se-ia mais transparente, afim de que a combinação com as ações de outros grupos ficasse mais suave. Similarmente, quando uma ação torna-se ativa num grupo, este tornar-se-ia automaticamente mais opaco.

6.2.2. Projetos de Pesquisa Futuros Relacionados ao Sistema InterFace

Conforme apresentamos no capítulo I desta dissertação, diversas áreas de pesquisa podem beneficia-se de um sistema de Animação Facial por computador.

Uma das áreas importantes de pesquisa recentemente no campo da Computação Gráfica é a simulação de atores virtuais inteligentes. Várias pesquisas recentes tem atingido resultados interessantes, como o trabalho de Ken Perlin na área de atores virtuais humanos [Per95][Per96a][Per96b] e de Demetri Terzopoulos, na área de simulação comportamental de animais [Ter98][Ter99].

O sistema **InterFace** pode ser a base para a construção de um sistema de simulação de atores humanos. Conceitos semelhantes ao utilizado neste trabalho para animação facial, como a composição de Grupos de Ações, podem ser utilizados na simulação completa de um ator virtual, bastando que as ações controlem, além das expressões faciais, o movimento de outros objetos que compõe o corpo do ator virtual, como os braços, mãos e pernas.

Uma área de pesquisa interessante é a simulação comportamental de muitos atores virtuais. Pretendemos desenvolver e testar modelos para a simulação de um grande número de atores virtuais em diversas situações. Diversas áreas podem beneficiar-se deste projeto interdisciplinar. Por exemplo, arquitetos poderão testar se um edifício permite a vazão de uma grande multidão em pânico por causa de um incêndio, que é uma simulação impossível de ser realizada na prática.

6.3. Considerações Finais

O trabalho proposto nesta dissertação foi a construção de sistema de animação facial chamado **InterFace**. Fizemos uma pesquisa com a literatura relevante da área afim de reunir as informações necessárias para a construção do nosso sistema. Para a implementação, optamos por construir um sistema que fosse interativo, operasse em tempo real, fosse fácil de usar e expansível. Para atingir este objetivo e baseando-se em nossa pesquisa inicial, implementamos conceitos importantes para o sistema, como a Biblioteca Básica de Expressões, os Grupos de Ações e a composição de grupos de ações com transparência. Na implementação utilizamos as linguagens de programação Java e VRML.

O sistema **InterFace** está disponível pela rede Internet na *World Wide Web*, através do endereço <http://webtec.cebinet.com.br/interface/>.

Bibliografia

- [Aria97] Ruschioni et al, *Some Experiences Implementing Virtual Worlds: The Aria Project*, IFIP 1997, Florianópolis, SC, Setembro 1997.
- [Ber85] P. Bergeron, P. Lachapelle, *Controlling facial expressions and body movements*, Siggraph '85 tutorials, San Francisco, CA, Julho 1985.
- [Bla89] Preston Blair, *How To Draw Film Cartoons*, Walter Foster Publishing, 1989.
- [Bla99] Volker Blanz, Thomas Vetter, *A Morphable Model For The Synthesis Of 3D Faces*, Siggraph 1999 Computer Graphics Proceedings, Los Angeles, CA, Agosto 1999.
- [Dar72] Charles Darwin, *The Expression of the Emotions in Man and Animals*, New York, Philosophical Library - 1955
- [Duc62] G.B. Duchenne, *De la physionomie humaine ou analyse electrophysiologique de l'expression des passions applicable à la pratique des arts plastiques*; Citado em [Par96].
- [EAI97] VRML External Authoring Interface:
<http://www.Web3d.org/WorkingGroups/vrml-eai/>.
- [Gue98] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, Fredrick Pighin, *Making Faces*, Siggraph 1998 Computer Graphics Proceedings, pgs. 55-66, Orlando, FL, Julho 1998.
- [HTTP99] Hypertext Transfer Protocol Specification:
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>.
- [Java95] Java Language *Web Site*: <http://www.java.sun.com/>.
- [Kal91] P. Kalra, Nadia Magnenat-Thalmann, Daniel Thalmann, *SMILE - A Multilayered Facial Animation System*, IFIP WG 5.10, Tokyo 1991.
- [Nic95] Nicholas Negroponte, *A Vida Digital*, Companhia das Letras, 2ª edição, 1996.
- [Par72] Frederick I. Parke, *Computer generated animation of faces*, Master Thesis, University of Utah, Junho 1972.
- [Par82] F. I. Parke, *Parameterized models for facial animation*, IEEE Computer Graphics and Applications, Novembro 1992.
- [Par96] Frederic I. Parke, Keith Waters, *Computer Facial Animation*, Published by A. K. Peters, 1996.

- [Per85] Ken Perlin, *An Image Synthesizer*, Siggraph 1985 Computer Graphics Proceedings, pgs. 287-296, San Francisco, CA, Julho 1985.
- [Per89] Ken Perlin, Eric M. Hoffert, *Hypertexture*, Siggraph 1989 Computer Graphics Proceedings, pgs. 253-262 Boston, MA, Julho 1989.
- [Per94] Ken Perlin, *Danse Interactif*, Siggraph '94 Electronic Theater, Orlando, FL, Agosto 1994.
- [Per95] Ken Perlin, *Real Time Responsible Animation with Personality*, IEEE Transactions on Visualization and Computer Graphics, 1995.
- [Per96a] Ken Perlin, Athomas Goldberg, *Improv: A System for Scripting Interactive Actors in Virtual Worlds*, Siggraph 1996 proceedings, New Orleans, LO, Agosto 1996.
- [Per96b] Ken Perlin, *Life-like, Believable Communication Agents*, Siggraph 1996 Course Notes #25, New Orleans, LO, Agosto 1996.
- [Per97] Ken Perlin, *Layered Compositing of Facial Expression*, Siggraph 1997 technical sketches, Los Angeles, CA, 1997.
- [Pig98] Fredrick Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, David H. Salesin, *Synthesizing Realistic Facial Expressions from Photographs*, Siggraph 1998 Computer Graphics Proceedings, pgs. 75-84, Orlando, FL, Julho 1998.
- [Por84] Thomas Porter, Tom Duff, *Compositing Digital Images*, SIGGRAPH 1984 Conference Proceedings, págs. 253-259, Julho 1984.
- [Ter98] Radek Grzeszczuk, Demetri Terzopoulos, Geoffrey Hinton, *NeuroAnimator: Fast Neural Network Emulation and Control of Physics-Based Models*, Siggraph 1998 Computer Graphics Proceedings, pgs. 9-20 Orlando, FL, Julho 1998.
- [Ter99] John Funge, Xiaoyuan Tu, Demetri Terzopoulos, *Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters*, Siggraph 1999 Computer Graphics Proceedings, pgs. 29-38, Los Angeles, CA, Agosto 1999.
- [Tha88] Nadia Magnenat-Thalmann, N. E. Primeau e Daniel Thalmann, *Abstract Muscle Actions Procedures for Human Face Animation*, The Visual Computer, maio 1988.
- [Tha89] Nadia Magnenat-Thalmann, H. minh, M. de Angelis and Daniel Thalmann, *Design, Transformation and Animation of Human Faces*, The Visual Computer, 1989.
- [Tha91] Nadia Magnenat-Thalmann, Daniel Thalmann, *Complex Models for Animating Synthetic Actors*, IEEE Computer Graphics and Applications,

Setembro 1991.

- [Tha94] Nadia Magnenat-Thalmann, Prem Kalra, *Modeling of Vascular Expressions in Facial Animation*, IEEE Computer Graphics and Applications, Setembro 1994.
- [VRML96] The Virtual Reality Modeling Language Consortium *Web Site*: <http://www.vrml.org/>.
- [Wan94] Carol L. Y. Wang, David R. Forsey: *Langwidere: A New Facial Animation System*, IEEE Computer Graphics and Applications, Setembro 1994.
- [Wat87] K. Waters, *A muscle model for animating three-dimensional facial expressions*, Siggraph 1987 Computer Graphics Proceedings, Anaheim, CA, Julho 1987
- [Wey98] José Daniel Ramos Wey, João Antonio Zuffo, *InterFace – A Real Time Facial Animation System*, Sibgrapi 1998 Proceedings, Outubro 1998

Bibliografia Adicional Recomendada

J. D. Foley, A. van Dam, Steven K. Feiner, John F. Hughes, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Company, 1990

Frederic Parke, *State of the Art in Facial Animation*, Siggraph 1990 Course Notes, curso nº 26, ACM, New York, NY, agosto 1990.

Stephen R. Davis, *Learn Java Now*, Microsoft Press, 1996

M. Hughes, C. Hughes, M. Shoffner, M. Winslow, *Java Network Programming*, Manning Publications Co., 1997

David Flanagan, *Java in a Nutshell*, Second Edition, O'Reilly, 1997

Ellen Adams, Donald Doherty, *Moving Worlds – the VRML 2.0 Specification*, Prima Publishing, 1996

Sítios Web Recomendados

LSI: <http://www.lsi.usp.br/>

MRL – NYU: <http://www.mrl.nyu.edu/>

Facial Animation Resources: <http://mambo.ucsc.edu/psl/fan.html>

Keith Waters: <http://www.crl.research.digital.com/projects/facial/facial.html>

Demitri Terzopoulos: <http://www.cs.toronto.edu/~dt/>

Web3D: <http://www.Web3d.org/>

Java3D: <http://www.java.sun.com/products/java-media/3D/>

Cosmo Software: <http://www.cosmosoftware.com/>

Netscape: <http://www.netscape.com/>

Apêndice I

Sintaxe dos Arquivos do Sistema InterFace

1.1. Nós de Controle da Cena VRML

Utilizando-se da facilidade de criação de protótipos de nós em VRML, o sistema **InterFace** implementou alguns nós para o controle da cena e dos atores virtuais:

Nó: InterfaceInfo	
Objetivo:	Fornecer as informações básicas da cena ao sistema: os nomes dos atores e das câmeras
Protótipo:	<pre>PROTO InterfaceInfo [exposedField MFString actorNames[""] exposedField MFString cameras [""]]</pre>
Exemplo:	<pre>DEF Interface InterfaceInfo { actorNames ["Giggio", "Valquiria"] cameras ["MyCamera"] }</pre>
Nó: ActorData	
Objetivo:	Fornecer as informações sobre um ator: os nomes das micro-expressões, suas animações em VRML e uma animação que ele irá executar como default
Protótipo:	<pre>PROTO ActorData [exposedField MFString microExpressionNames [""] exposedField MFString animations [""] exposedField SFString animationDefault ""]</pre>
Exemplo:	<pre>DEF Giggio ActorData { MicroExpressionNames ["A100", "A101", "A102", "A103", "A106", "A107", "A104", "A105", "A113", "A112", "A116", "A117", "A114", "A115", "A121", "A118", "A119", "A124", "A125", "A126", "A127", "A128", "A129", "A130", "A131", "A132"] animations ["Rotacao_Y", "Rotacao_X", "Rotacao_Z"] }</pre>

Nó: MicroExpression	
Objetivo:	descrever a microexpressao quando esta estiver com valor de intensidade igual a 1.
Protótipo:	<pre>PROTO MicroExpression [exposedField SFString expressionName "" exposedField SFString objectName "" exposedField SFVec3f translation -1000 -1000 -1000 exposedField SFFloat rotationX -1000 exposedField SFFloat rotationY -1000 exposedField SFFloat rotationZ -1000 exposedField SFRotation rotation -1000 -1000 -1000 -1000 exposedField SFVec3f scale -1000 -1000 -1000 exposedField MFVec3f morph [] exposedField MFInt32 index []]</pre>
Exemplo:	<pre>DEF A130 MicroExpression { ExpressionName "roty" ObjectName "giggio" rotationY 0.7853 }</pre>

Tabela I.1: Nós de controle em VRML do sistema *InterFace*

1.2. Arquivo de Descrição de Ações

Um único arquivo é usado para descrever as ações e os Grupos de Ações. A sintaxe utilizada é bastante simples: Descreve-se inicialmente os dados do grupo, e posteriormente a descrição das ações daquele grupo. A descrição de um Grupo de Ações termina quando inicia-se a descrição de um novo grupo ou com o fim do arquivo. Linhas começadas por “#” são consideradas como comentários. Linhas em branco são ignoradas.

Descrição de Grupos de Ações	
Objetivo:	Descrever os dados do grupo: nome, valor inicial de transparência e nome de uma ação a ser ativada inicialmente
Protótipo:	ActionGroup nome [alpha] [defaultAction]
Exemplo:	ActionGroup NaoIntencionais 0 respirar

Descrição de Ações	
Objetivo:	Descrever a ação: o nome, os tempos de subida, duração e descida
Protótipo:	Action nome tin tdur tout
Exemplo:	Action Espirro 0.1 9.6 0.3
Descrição de Micro-Ações	
Objetivo:	Descrever os dados da micro-ação: o nome da expressão utilizada, os tempos de início, subida, duração e descida e a função matemática que modela os valores de intensidade
Protótipo:	ExpName tstart tin tdur tout função parâmetros
Exemplo:	rotx 0 0 10 0 spline 0 0 3.5 1 4 -2 5 -2 7 0

Tabela I.2: Sintaxe na descrição de ações, Grupos de Ações e Micro-ações

As funções que podem ser utilizadas nas micro-ações e seus respectivos parâmetros estão listados no capítulo VI desta dissertação (item 4.3.1.1).

Exemplo:

```
# Arquivo de exemplo para descrição de ações
# Este arquivo contém dois grupos de ações (NaoIntencionais e Emocoes)
# com duas ações no primeiro grupo e cinco ações no segundo grupo

ActionGroup NaoIntencionais 0 respirar
  Action pisca 0 0 0
    pisca 0 0 0 0 impulso 0.9 1 0.15 0.1

  Action respirar 0 0 0
    rotx 0 1 3 1 sin 8 0.1 triggered_by impulso 1 1 0.15
    pisca 0 0 0 0 impulso 0.9 1 0.15 0.1
    rotz 0 0.2 1 0.2 ruido 1 0.4 triggered_by impulso 1 1 0.15
    olho_hor 0 0.5 3 0.5 ruido 2 3 triggered_by impulso 1 1 0.25

ActionGroup Emocoes 0
  Action Espirro 0.1 9.6 0.3
    rotx 0 0 10 0 spline 0 0 3.5 1 4 -2 5 -2 7 0
    bocejo 0 3 0.5 0.5 constante 1
    pisca 3 0 2.5 2 constante 1
    assobio 3.5 0.5 1 2 constante 1
    roty 7 0.2 1 0.2 ruido 0.3 1
    rotz 7 0.2 1 0.2 ruido 0.3 0.6

  Action triste 1 5 1
    triste 0 0 0 0 ruido 1 0.7 1
```

```
Action feliz 1 5 1
  feliz 0 0 0 0 ruido 1 0.7 1

Action MedoBarata 1 5 1
  medo 0 0 0 0 ruido 1 0.7 1

Action assobio 1 10 1
  assobio 0 0 0 0 ruido 1 0.6 1
  rotz 0 0 0 0 ruido 1 1.4
```

Tabela I.3: *Exemplo de um arquivo de descrição de ações*

Apêndice II

Sintaxe do Protocolo de Comunicação InterFace Protocol (IFP)

INTERFACE - PROTOCOLO DE COMUNICAÇÃO

Autor: José Daniel Wey <wey@lsi.usp.br>
Versão Draft 1.1 - 13/agosto/1999

1) INTRODUÇÃO

O protocolo de comunicação com o InterFace permitirá que atores virtuais execute comandos enviados remotamente por qualquer máquina via TCP/IP. O InterFace agirá como "servidor de atores virtuais", permitindo que outros programas utilizem os recursos de atores virtuais providos pelo sistema, ou configurem o sistema InterFace remotamente.

Ao longo deste documento, as palavras InterFace, sistema ou sistema InterFace serão utilizadas para designar o servidor de atores virtuais. Os programas que se conectarem ao InterFace são chamados de clientes.

2) DEFAULTS

O sistema InterFace disponibiliza um servidor TCP/IP. A porta padrão do servidor é 2571.

Cada pedido enviado pelo cliente deve terminar com uma linha em branco. A resposta do servidor também terminará por uma linha em branco.

Os pedidos podem ser feitos em maiúsculas ou minúsculas (o sistema não faz diferença entre os comandos "GET ACTORS IFP/1.0", "Get actors IFP/1.0" ou "GeT AcToRs IFP/1.0"). Porém, isto não é válido para variáveis como o nome do ator ou o nome das ações. Caso o nome do ator seja "Giggio" e o cliente enviar um pedido como "ACTOR giggio ExecuteAction respirar", o sistema devolverá uma resposta do tipo "IFP/1.0 402 Actor not found".

Ao longo do documento, os comandos do protocolo marcados entre "menor que" e "maior que" (" $<$ " e " $>$ ") devem ser substituídos pelo valor da variável. Por exemplo, no comando "GET actionFile <ActorName>", o campo <ActorName> deve ser substituído pelo nome do ator. Os comandos marcados entre chaves ("[" e "]") são opcionais.

Todo pedido do cliente ao servidor termina por IFP/1.0.

3) CÓDIGOS DE STATUS DO PROTOCOLO IFP/1.0

Código - Status

200 - OK
300 - Not Implemented Yet
400 - Bad Request
401 - Unauthorized
402 - Actor not found
403 - ActionGroup not found
404 - Action not found
405 - Animation not found

3) PROTOCOLO

Os clientes terão a sua disposição uma série de comandos para controlar o sistema InterFace, divididos em quatro classes ou tipos: AUTH, GET, ACTOR e QUIT.

3.1) Classe AUTH

A classe AUTH permite ao cliente autenticar-se junto ao sistema InterFace. Enquanto um cliente não estiver autenticado, o InterFace devolverá sempre a mesma resposta a qualquer pedido:

IFP/1.0 401 Unauthorized

Obs.: A classe AUTH não está implementada na versão atual do InterFace. Todos os clientes estão autorizados a conectar-se ao InterFace.

Os comandos da classe AUTH são:

- AUTH login senha IFP/1.0

O cliente envia seu login e sua senha criptografada por DES (utilizando o comando crypt() do ANSI C, por exemplo).

Caso o usuário esteja cadastrado e a senha correta, o sistema irá devolver a seguinte informação

IFP/1.0 200 OK

Caso contrário, o sistema devolverá

IFP/1.0 401 Unauthorized

3.2) Classe GET

Na classe GET o cliente irá obter as configurações atuais do sistema InterFace (número de atores, nome dos atores, arquivos com as ações, etc.).

Os comandos da classe GET são os seguintes:

- GET Actors IFP/1.0

O sistema devolverá a lista de atores, da seguinte forma:

```
IFP/1.0 200 OK
NumActors: <número de atores>
ActorNames: <nomes dos atores separados por espaços>
```

Exemplo:

```
cliente> GET actors IFP/1.0
cliente>
InterFace> IFP/1.0 200 OK
InterFace> Number of Actors: 2
InterFace> Actor Names: Giggio Valquiria
InterFace>
```

- GET Actions <ActorName> IFP/1.0

O sistema irá devolver a lista de ações e grupos de ações para o ator <ActorName>.

Exemplo:

```
cliente> GET Actions Giggio IFP/1.0
cliente>
InterFace> IFP/1.0 200 OK
InterFace> Number of ActionGroups: 2
InterFace> ActionGroup: naoIntencionais
InterFace> Number of Actions: 2
InterFace> Actions: pisca respirar
InterFace> ActionGroup: Emocoes
InterFace> Number of Actions: 4
InterFace> Actions: triste feliz medo alegre
InterFace>
```

- GET Expressions <ActorName> IFP/1.0

O sistema irá devolver a lista de expressões do ator <ActorName>.

Exemplo:

```
cliente> GET ExpressionFile Giggio IFP/1.0
cliente>
InterFace> IFP/1.0 200 OK
InterFace> Number of Expressions: 4
InterFace> Expressions: ahh ohh Duvida_e Duvida_d
InterFace>
```

- GET Animations <ActorName> IFP/1.0

O sistema irá devolver a lista de animações do ator <ActorName>.

Exemplo:

```
cliente> GET Animations Giggio IFP/1.0
cliente>
InterFace> IFP/1.0 200 OK
InterFace> Number of Animations: 5
InterFace> Animations: Danca Canta Sapateia Pula Ajoelha
InterFace>
```

3.3) Classe ACTOR

Através da classe ACTOR, o cliente poderá solicitar que o ator execute uma ação,

Os comandos da classe ACTOR são os seguintes:

```
- ACTOR <ActorName> ExecuteAction <ActionGroup> <ActionName>
[Intensity] <intensityValue> [TimeStart] <timeStartValue>
[TimeIn] <timeInValue> [TimeDur] <timeDurValue>
[TimeOut] <timeOutValue> IFP/1.0
```

Solicita ao InterFace que o ator <ActorName> execute a ação <ActionName> do grupo <ActionGroup>. Opcionalmente, podem ser especificados a intensidade, o tempo para o início da ação, o tempo de "fade in", o tempo de duração e o tempo de "fade out" da ação - sobrescrevendo os valores default da ação.

O sistema irá devolver um ID único para aquela ação.

Exemplo:

```
cliente> ACTOR Giggio ExecuteAction naoIntencionais respirar
IFP/1.0
cliente>
InterFace> IFP/1.0 200 OK
InterFace> Action naoIntencionais respirar ActionId: 102022
InterFace> Action 102022 started.
InterFace>
```

Quando a ação terminar de ser executada, o sistema enviará o seguinte

comando: Action <ActionId> endend.

```
- ACTOR <ActorName> StopAction <ActionId> IFP/1.0
```

Solicita que o ator pare a execução da ação <ActionId>

```
- ACTOR <ActorName> StopAction <ActionGroup> <ActionName> IFP/1.0
```

Solicita que o ator pare a execução da ação <ActionName> do grupo <ActionGroup>

```
- ACTOR <ActorName> StopActionGroup <ActionGroup> IFP/1.0
```

Solicita que o ator pare a execução das ações ativas do

grupo <ActionGroup>

- ACTOR <ActorName> ExecuteActionFromActionGroup <ActionGroup>
IFP/1.0

Solicita ao sistema que sorteie uma ação do grupo <ActionGroup> e execute-a. O sistema irá devolver o nome da ação sendo executada e um ID - da mesma maneira que o comando ExecuteAction.

- ACTOR <ActorName> ExecuteAnimation <AnimationName>
[TimeScale] <scaleNumber> IFP/1.0

Solicita ao sistema que execute a animação <AnimationName>. A duração da animação pode ser escalada pelo fator <scaleNumber>. Um fator de 1.0 irá executar a ação no tempo determinado no arquivo VRML; um fator de 2.0 irá executar a ação duas vezes mais lentamente (o tempo de duração será duas vezes maior); um fator de 0.5 irá executar a ação duas vezes mais rápido.

3.4) Classe QUIT;

Solicita ao InterFace que feche a conexão. O único comando desta classe é o seguinte:

- QUIT IFP/1.0

O sistema irá devolver o comando OK e fechar a conexão.