

MAC499 - Trabalho de Formatura Supervisionado

Sistema de Animação Facial Tridimensional e Síntese de Voz

Aluna: Andréa Britto Mattos

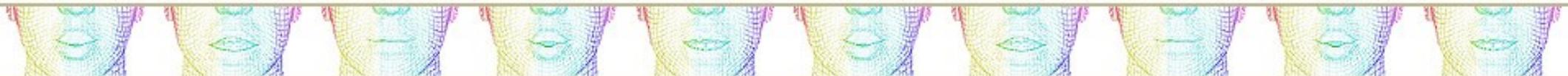
Orientador: Roberto Marcondes Cesar Jr.

O Projeto Avator

- Versão de 2007:



Funcionamento interno



- Haptek Player:
 - Gráficos
 - Áudio
- J-Alice:
 - Inteligência Artificial
- OpenCV:
 - Visão Computacional

Problemas



- Modularização e documentação
- Ambiente muito restrito:
 - Plataforma: Windows
 - IDE: Visual Studio
 - Biblioteca *MFC (Microsoft Foundation Classes)*
 - Não disponível na versão gratuita da IDE
- Culpa: Haptেক

Proposta



- Sistema próprio de animação e síntese de voz:
 - Animação realista
 - Síntese em português
 - Multi-plataforma
 - Nenhuma IDE em particular
 - Mais expressões faciais
 - Documentação
 - Pudessem ser usadas em outras aplicações

Primeira Parte



Módulo de Animação e Síntese de Voz

Morph Target



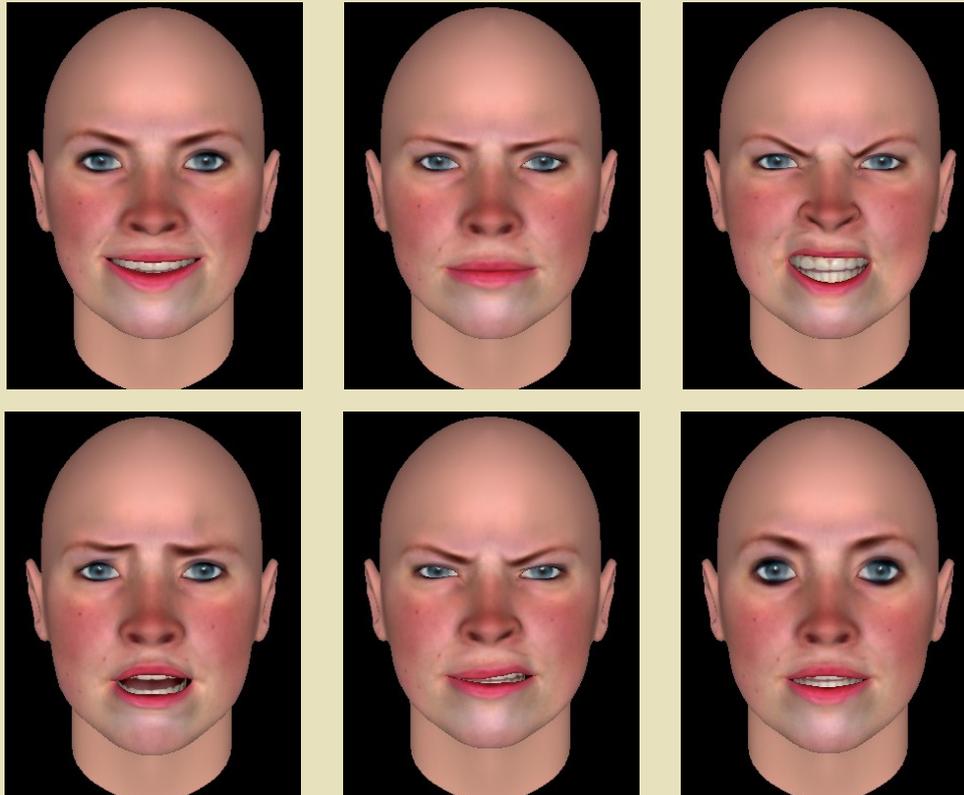
- Conjunto fixo de quadros-chaves
- Interpolação



Quadros-Chaves: Expressões Faciais



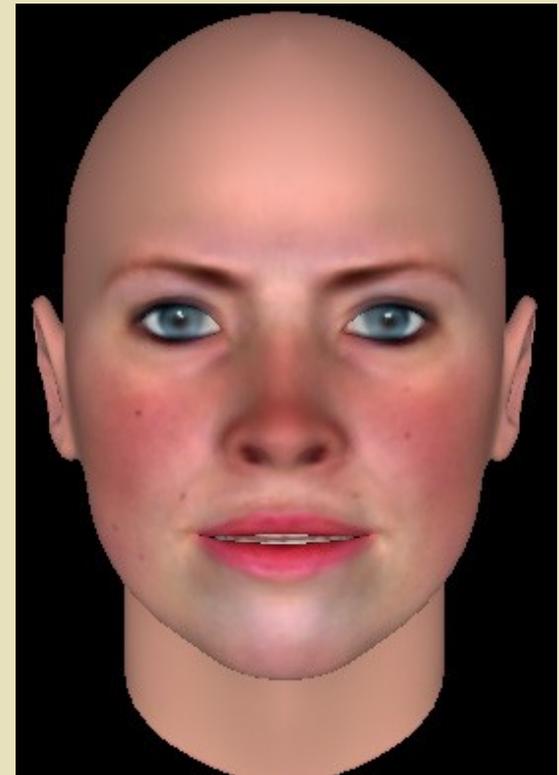
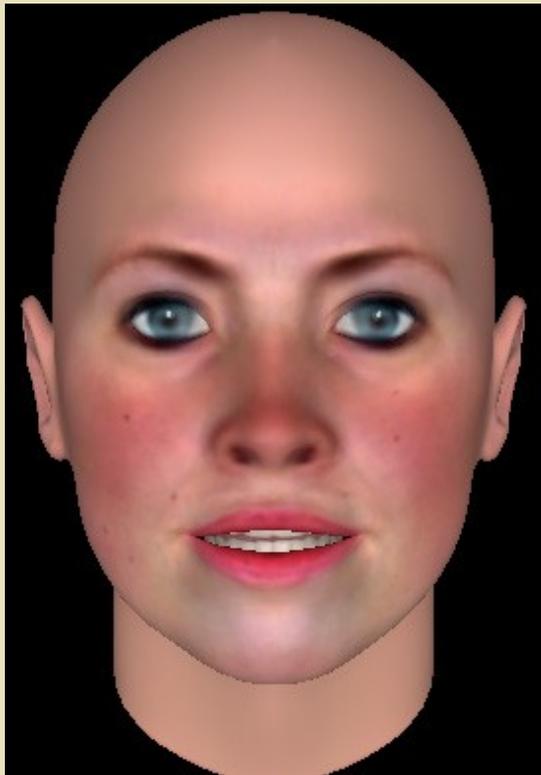
- Modelo de Paul Ekman
 - Seis expressões universais:
 - Alegria, tristeza, raiva, medo, nojo e surpresa.



Quadros-Chaves: Expressões Faciais



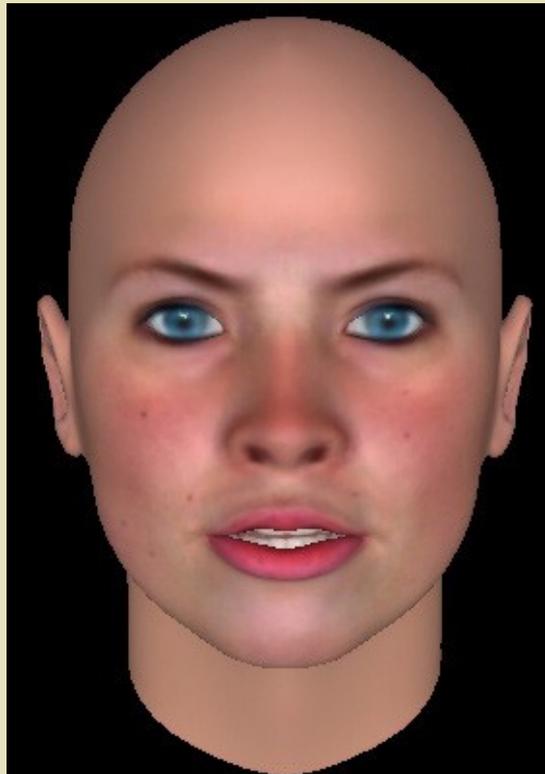
Surpresa + Tristeza = Preocupação:



Quadros-Chaves: Visemas



- Correspondente visual de um fonema
- Pode corresponder a mais de um fonema:
 - Jato e chato:



Desafios em Animação Facial



1. Modelo fiel
2. Animação do modelo
3. Sincronização labial

1. Modelagem: FaceGen



- Modelos realistas
- Expressões faciais
- Visemas
- Movimentos não-verbais
- Exportação do resultado

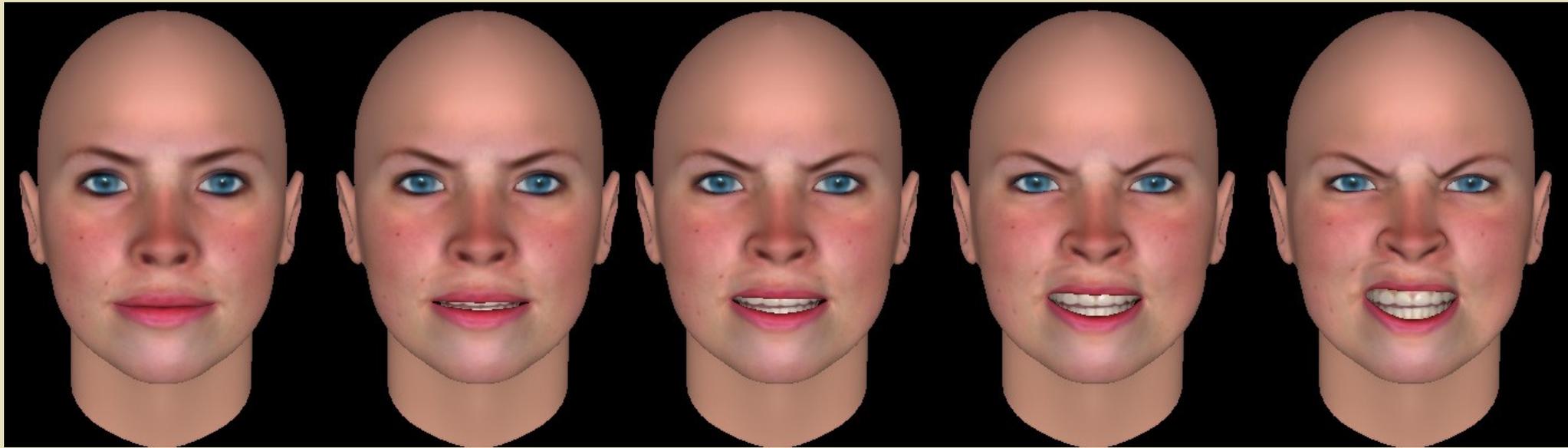
2. Animação: OGRE3D



- *Pose Animation:*
 - Pose:
 - Valores que definem a deformação da malha
 - Não contém a posição espacial de cada vértice da face
 - Influência:
 - Quantidade de deformação da pose
 - Valor entre 0 e 1

2. Animação: Ogre3D

- Raiva:



0%

25%

50%

75%

100%

3. Síntese de Voz: eSpeak



- Na versão anterior:
 - SAPI (*Speech Application Programming Interface*)
 - Instalar uma voz em português
- eSpeak:
 - Programa de TTS (*Text to Speech*)
 - Suporta o português
 - Regras para extrair fonemas de um texto

Importação dos modelos para o Ogre



- O Ogre suporta apenas arquivos binários no formato **mesh**
- Conversão:
 - I. modelo => **mesh.xml**
 - II. mesh.xml => mesh
- Problema:
 - Vários arquivos diferentes, um para cada pose
 - O Ogre espera um único modelo
- Solução:
 - Modificar diretamente os arquivos mesh.xml

O formato mesh.xml

```
<mesh>
  <submeshes>
    <faces count="número_de_faces">
      ...
    </faces>
    <geometry vertexcount="número_de_vértices">
      <vertex>
        ...
      </vertex>
    </geometry>
  </submeshes>
  <poses>
    <pose target="submesh" index="índice_da_submalha" name="nome_da_pose">
      ...
    </pose>
  </poses>
</mesh>
```

○ formato mesh.xml

- Definição de um vértice:

```
<vertex>  
  <position x="30.242" y="70.9967" z="29.62" />  
  <normal x="0.839101" y="-0.19281" z="0.508659" />  
  <texcoord u="0.444124" v="0.673521" />  
</vertex>
```

- Definição de uma pose:

```
<poseoffset index="índice_do_vértice" x="0.138" y="-3.268" z="-2.259" />
```

○ script de conversão

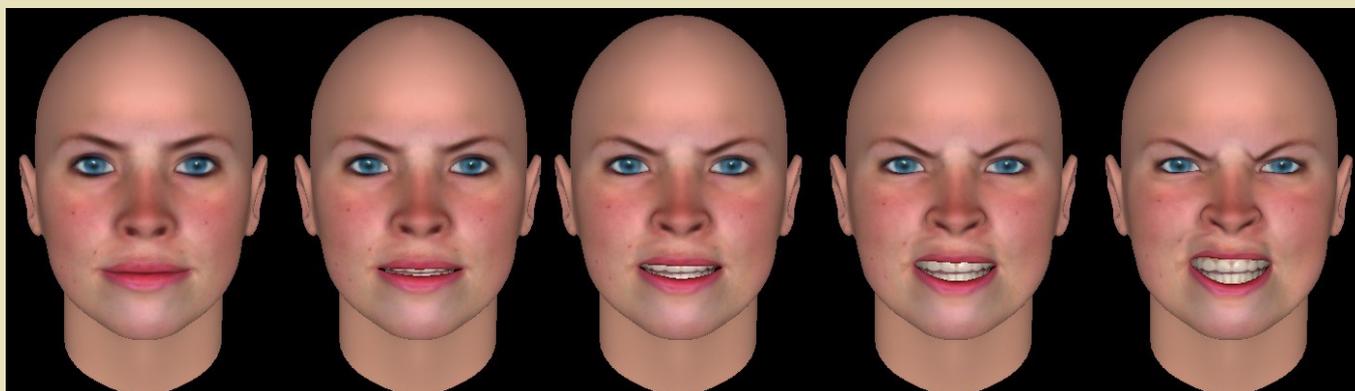


- Subtração entre cada vértice da pose com o vértice correspondente na face neutra
- O valor resultante é o deslocamento que o vértice da face neutra deve sofrer na pose
- Os modelos do FaceGen tem o mesmo número de vértices
 - É viável subtrair ponto a ponto

Implementação do módulo de Animação



- Animação suave:



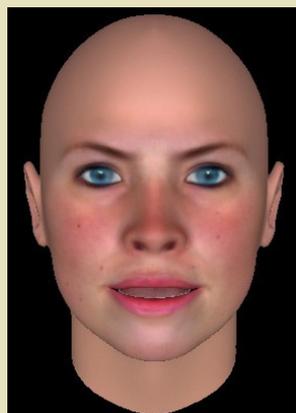
- Atualizar uma pose repetidamente, incrementando o parâmetro da influência
- A velocidade depende do incremento

Implementação do módulo de Síntese de Voz

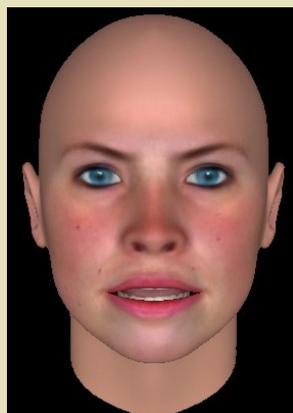


- Saber a hora exata da pronúncia de cada fonema
- *Callback*:
 - Chamada automaticamente pelo eSpeak, quando um buffer de áudio é produzido
 - Pode retornar eventos de fonema
 - O fonema é mapeado para o visema que deve ser exibido
 - Mapeamento em portuguêsês

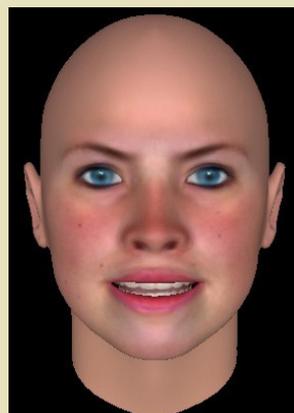
Mapeamento de Fonemas para Visemas



Visema A



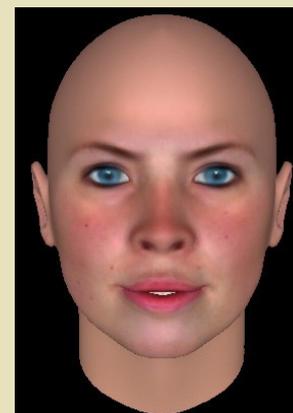
Visema E



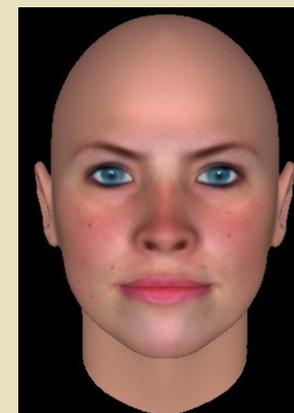
Visema I



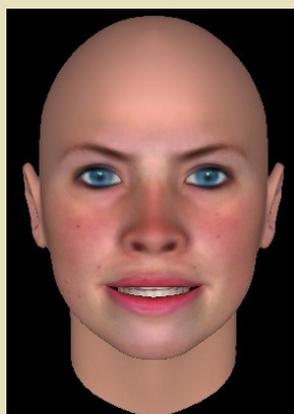
Visema O



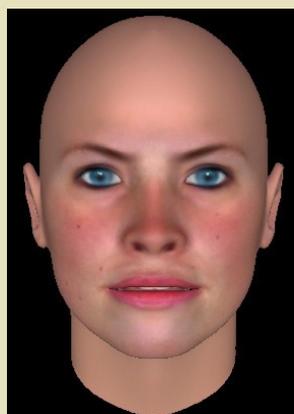
Visema U



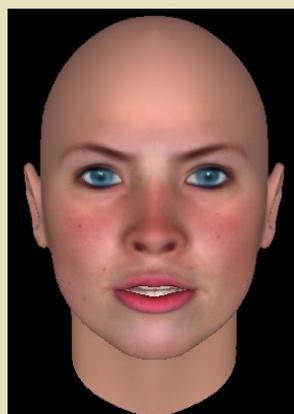
Visema BMP



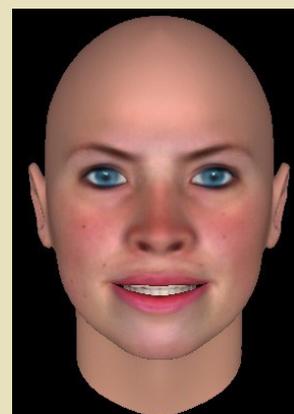
Visema DSTZ



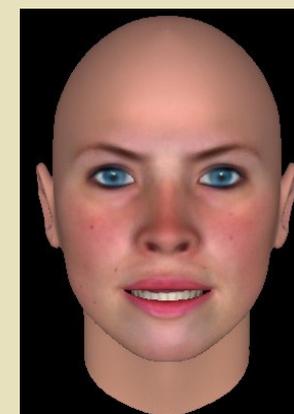
Visema FV



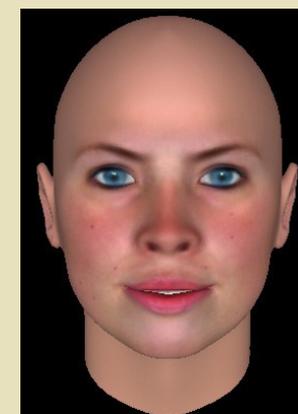
Visema GK



Visema JX



Visema LN



Visema R

Segunda Parte



Refatoração do Avator

Visão Computacional



- 2007:
 - Detecção de movimento
 - Subtração quadro a quadro
 - Não exigia interlocutor humano
- 2008:
 - Detecção de face
 - Movimento dos olhos conforme a posição do usuário

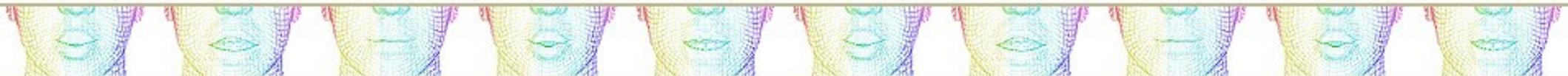
Inteligência Artificial



- AIML (*Artificial Intelligence Markup Language*):
 - Criação de *chatterbots*

```
<pattern>QUAL O SEU NOME?</pattern>  
<template>Meu nome é Ana.</template>
```
- Não houve tempo
 - Implementações antigas, mal documentadas, instalação complexa
 - Buscar outras alternativas

Resultados



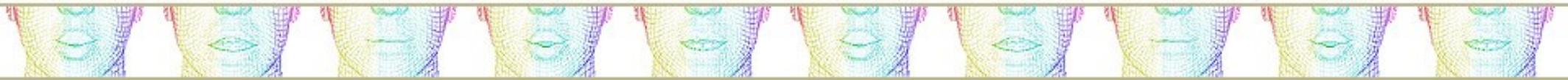
- Pontos negativos:
 - Não houve tempo de implementar a IA
 - O software não pôde rodar no Windows
- Pontos positivos:
 - IDE livre
 - Documentação
 - Mais expressões faciais, modelos, vozes
 - Linux

Demonstração



Funcionamento do software

Conclusão



- Página do Projeto:

www.linux.ime.usp.br/~dedea/mac499

- Dúvidas?