

# IMPLEMENTAÇÃO DE VISUALIZAÇÃO DE PERSONAGEM NÃO JOGADOR NA MOBILE 3D GAME ENGINE (M3GE)

Nome:

Claudio José Estácio

Orientador:

Prof. Dr. Paulo Cesar Rodacki Gomes

# Roteiro de Apresentação

1. Introdução
  - Objetivo do Trabalho
2. Fundamentação Teórica
3. Desenvolvimento do Trabalho
  - Requisitos
  - Especificação
  - Implementação
  - Resultados
4. Conclusão
  - Extensões

# 1. Introdução

- Jogos eletrônicos são entretenimento;
- Jogos para consoles e PCs se modernizaram nos últimos anos;
- O mesmo espera-se para os jogos em celulares;
- Evolução dos jogos 2D para 3D graças a evolução dos celulares;

# 1. Introdução

- Motores de jogos;
- Mobile 3D Game Engine (M3GE);
- Personagem Não Jogador (PNJ);
- Arquivo Quake 2's Model (MD2);

# 1. Introdução

## Objetivo do Trabalho

- Agregar PNJ na M3GE;
- Implementar suporte a importação e visualização de modelos 3D;
- Utilizar o formato de arquivo MD2;

# 2. Fundamentação Teórica

## Jogos em Dispositivos Móveis

- Grande avanço dos aparelhos celulares.
- Soluções:
  - Sun J2ME
  - Qualcomm BREW
  - Mophun

# 2. Fundamentação Teórica

## Motores de Jogos 3D

- Coração de qualquer jogo eletrônico;
- Reusabilidade;
- O motor implementa a visualização 3D dos jogos;
- Utilizando o motor o desenvolvedor fica abstraído das funcionalidades, preocupando-se assim com a lógica e o enredo do jogo;

# 2. Fundamentação Teórica

## Motores de Jogos 3D



# 2. Fundamentação Teórica

## M3GE

- Protótipo de motor de jogos para dispositivos móveis com suporte a M3G;
- Funcionalidade implementadas:
  - Carregar e desenhar um ambiente 3D;
  - Criação de câmeras;
  - Tratamento de eventos;
  - Movimentação de personagens no cenário;
  - Tratamento de colisão.

## 2. Fundamentação Teórica

### Personagem Não Jogador (PNJ)

- Personagem presente no jogo;
- Não é controlado pelo jogador;
- Mas que se relaciona de alguma forma com ele;

## 2. Fundamentação Teórica

### Personagem Não Jogador (PNJ)

- Por exemplo, num jogo de primeira pessoa existem na cena personagens que são inimigos do jogador e tem-se como objetivo acertar tiros neles;
- Diversas aparências:
- Humanos, figuras mitológicas, mutantes, robôs, entre outros.

## 2. Fundamentação Teórica

### Modelo Animado MD2

- Formato Popular;
- Surgiu em novembro de 1997;
- Utilizado no jogo Quake 2 produzido pela ID Software;
- Modelo geométrico formado por triângulos;

# 2. Fundamentação Teórica

## Modelo Animado MD2

- Animação quadro a quadro;
- Formato binário;
- Divido em duas partes:
  - Cabeçalho
  - Dados
- Texturas (skins)

## 2. Fundamentação Teórica

### Modelo Animado MD2



knight.png



knight.md2

# 2. Fundamentação Teórica

## Trabalhos Correlatos

### ■ wGEN

- Jogos 2D;
- Pioneiro na utilização de J2ME.

### ■ mOGE

- Jogos 3D;
- Utilizada malha de triângulos;
- Utilizada a árvore do grafo de cena.

# 3. Desenvolvimento do Trabalho

## Requisitos

- importar modelos 3D;
- inserir personagem ao grafo de cena;
- visualizar personagem na cena;
- utilizar a especificação M3G.

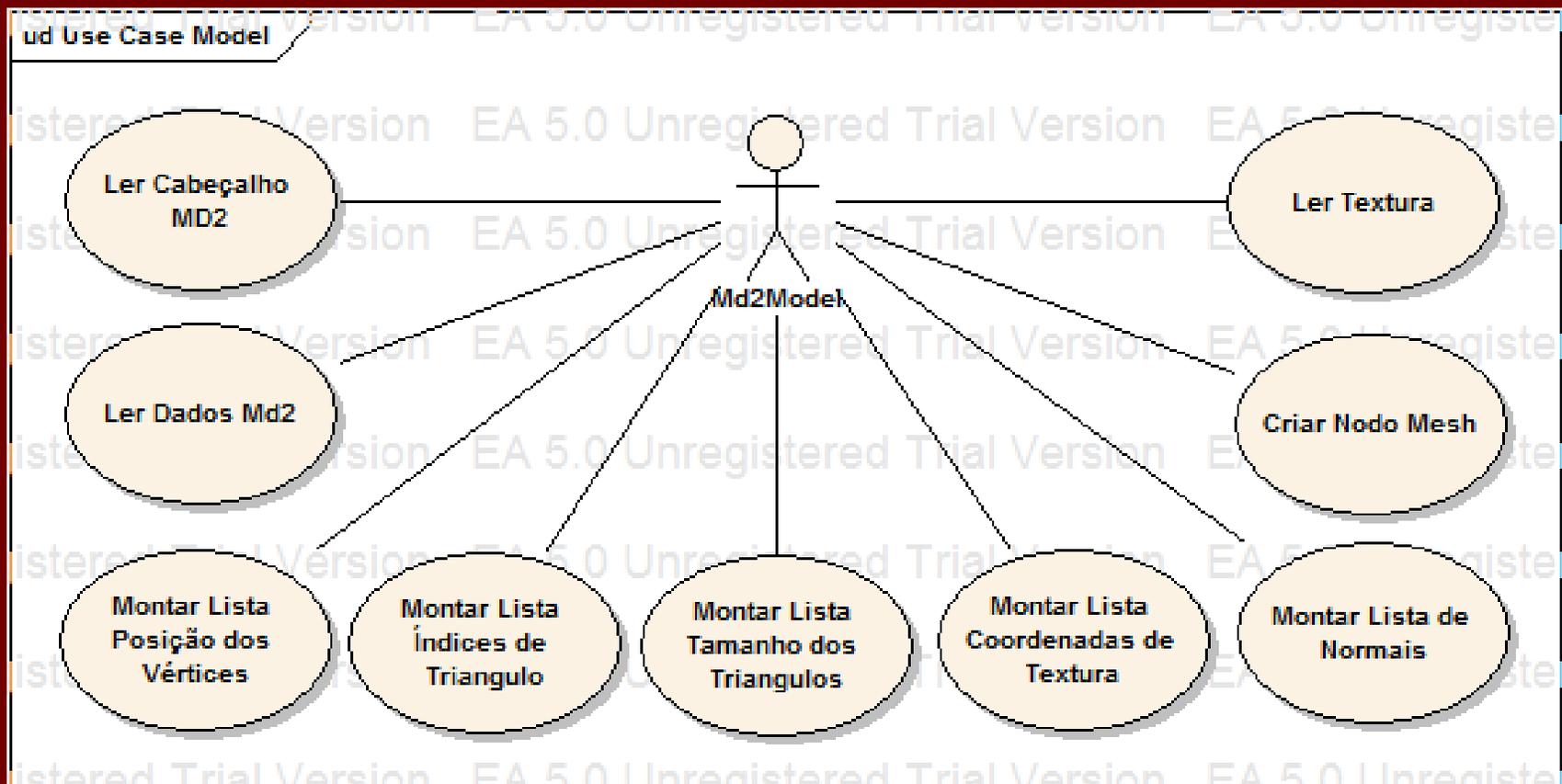
# 3. Desenvolvimento do Trabalho

## Especificação

- Orientação a Objetos;
- Diagramas da UML:
  - diagrama de caso de uso;
  - diagrama de classes;
  - diagrama de seqüência.
- Ferramenta Enterprise Architect.

# 3. Desenvolvimento do Trabalho

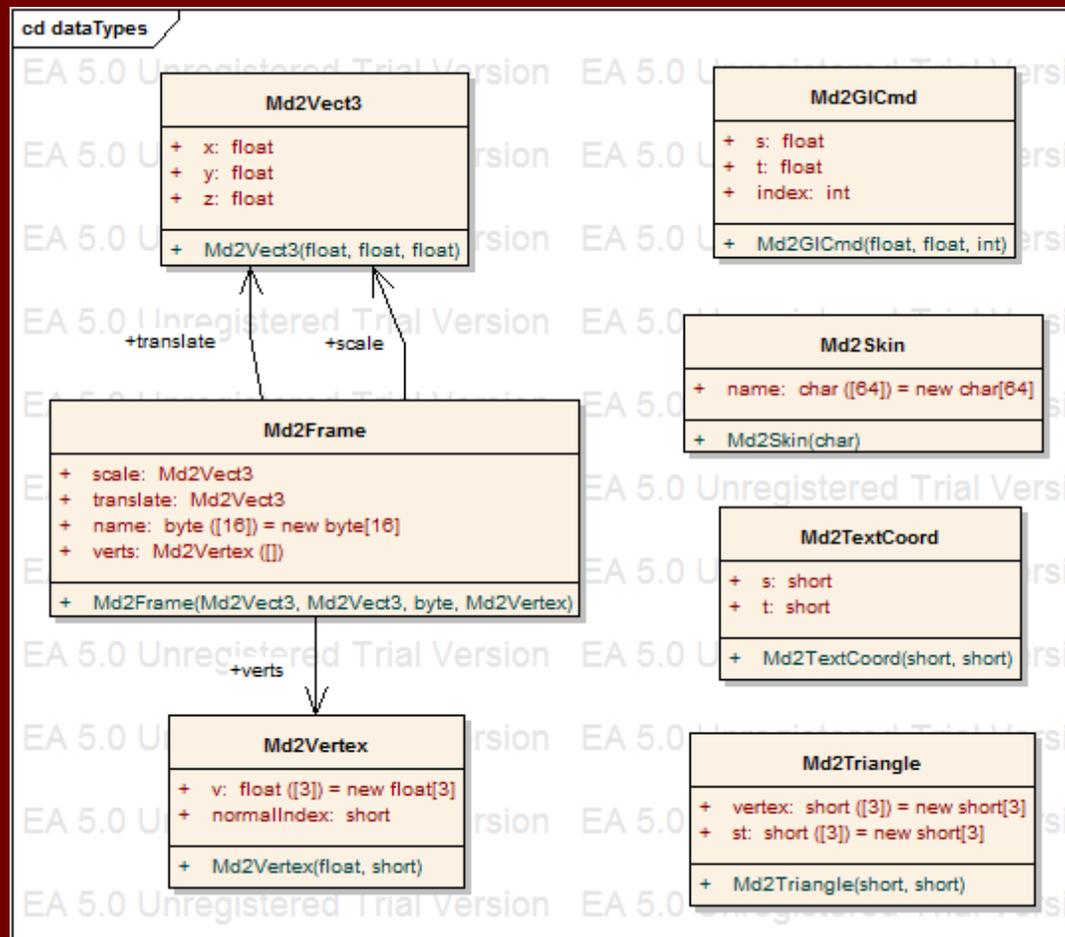
## Especificação





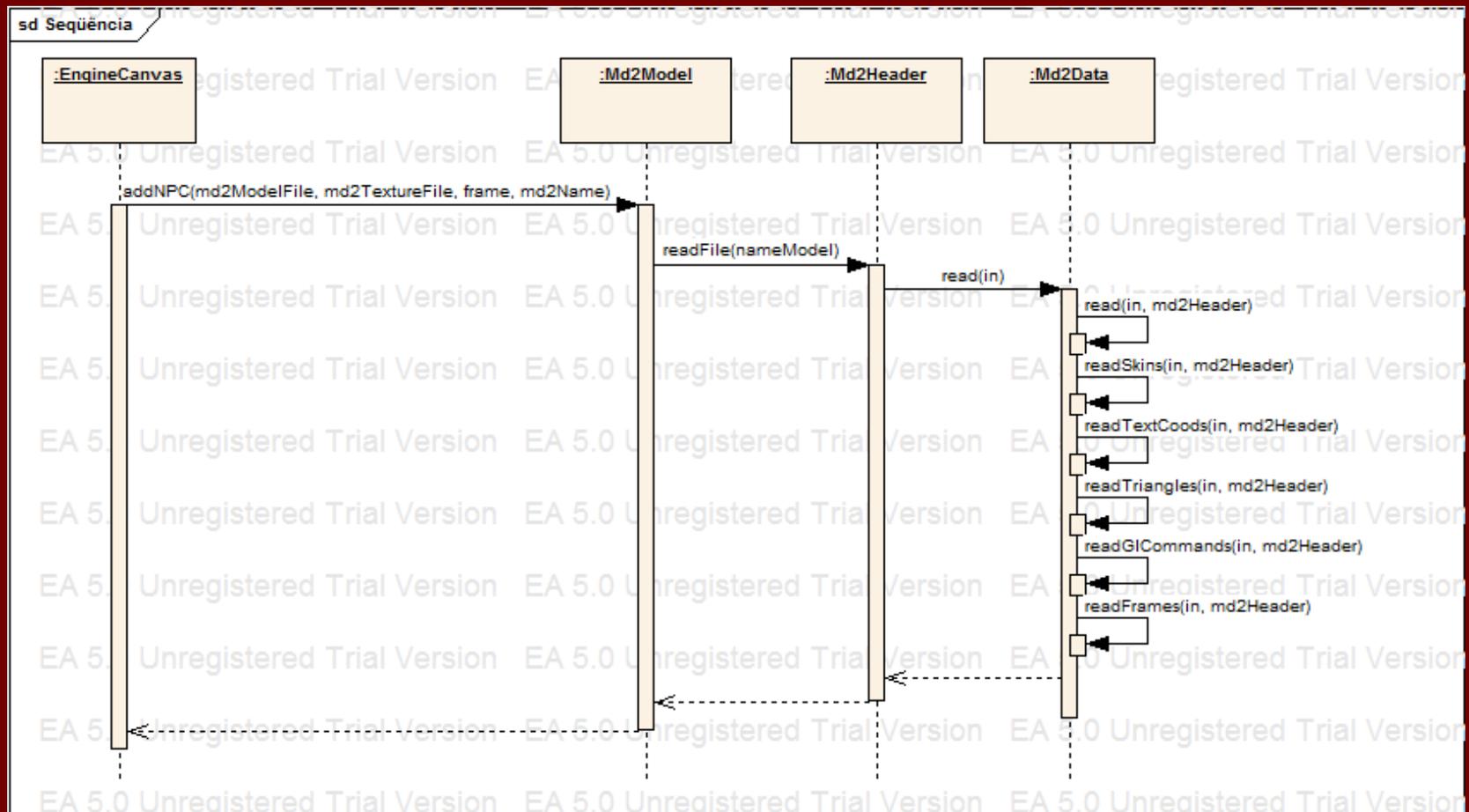
# 3. Desenvolvimento do Trabalho

## Especificação



# 3. Desenvolvimento do Trabalho

## Especificação



# 4. Implementação

## Tecnologias Utilizadas

- Eclipse;
- J2ME;
- Sun Java Wireless Toolkit;
- M3G;
- EclipseME;
- AC3D.

# 4. Implementação

## Leitura do Arquivo Binário

- Classe `DataInputStream` que implementa a classe `DataInput` da J2ME;
- Quando é feita a leitura de mais de um byte, esta classe acaba invertendo sua seqüência;
- Utilização da classe `MD2LittleEndianDataInputStream`;

# 4. Implementação

## Armazenamento da Estrutura do Modelo MD2

- Classes do pacote datatypes:
  - md2Vect3;
  - md2TextCoord;
  - Md2Triangle;
  - Md2Vertex;
  - Md2Frame;
  - Md2GICmd.

# 4. Implementação

## Armazenamento da Estrutura do Modelo MD2

- Duas classes para tratar o arquivo MD2:
  - md2Header
  - md2Data
- A classe Md2Header é responsável pela leitura do cabeçalho;
- A classe md2Data é responsável por fazer a leitura de todo o restante dos dados presentes no arquivo MD2.

# 4. Implementação

## Vetor Normal

- Lista de vetores já calculados;
- Arquivo MD2 traz apenas uma lista de índices para esses vetores;
- A classe Md2Normal que armazena a lista de vetores normais.

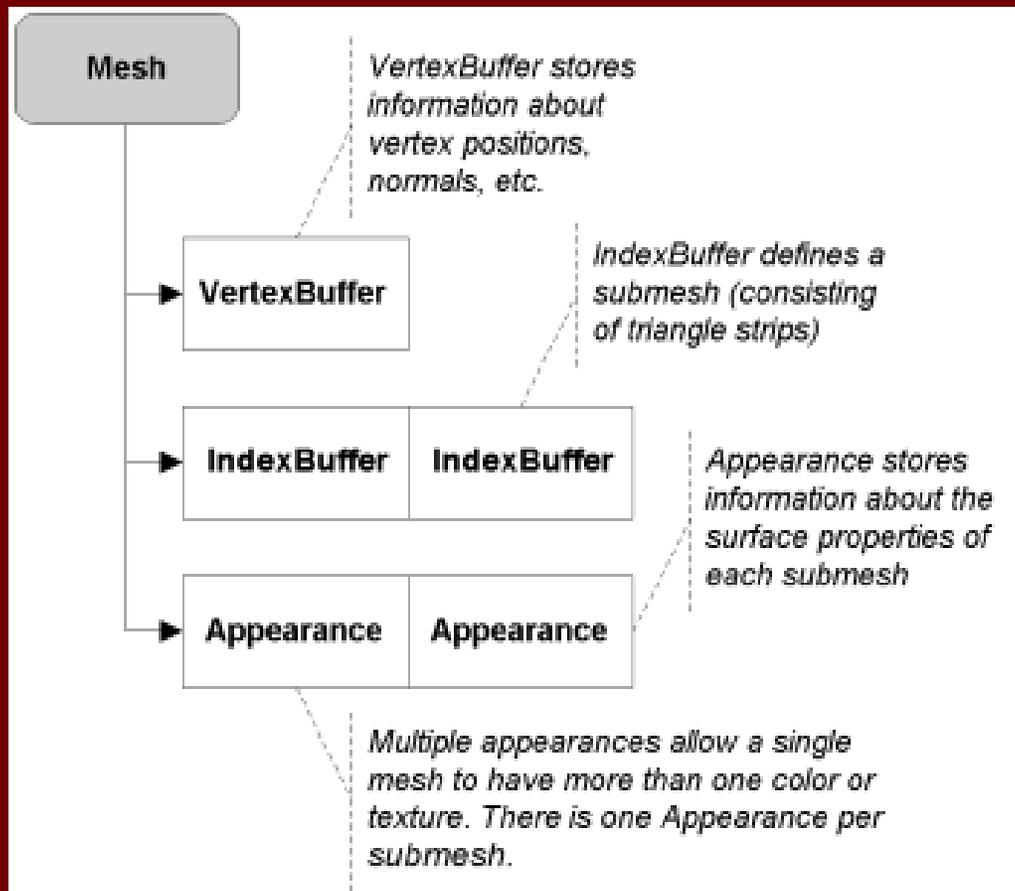
# 4. Implementação

## Visualização do Modelo

- M3GE já tem funções capazes de desenhar o objeto;
- Basta anexar ao grafo de cena;
- Nodo Mesh;

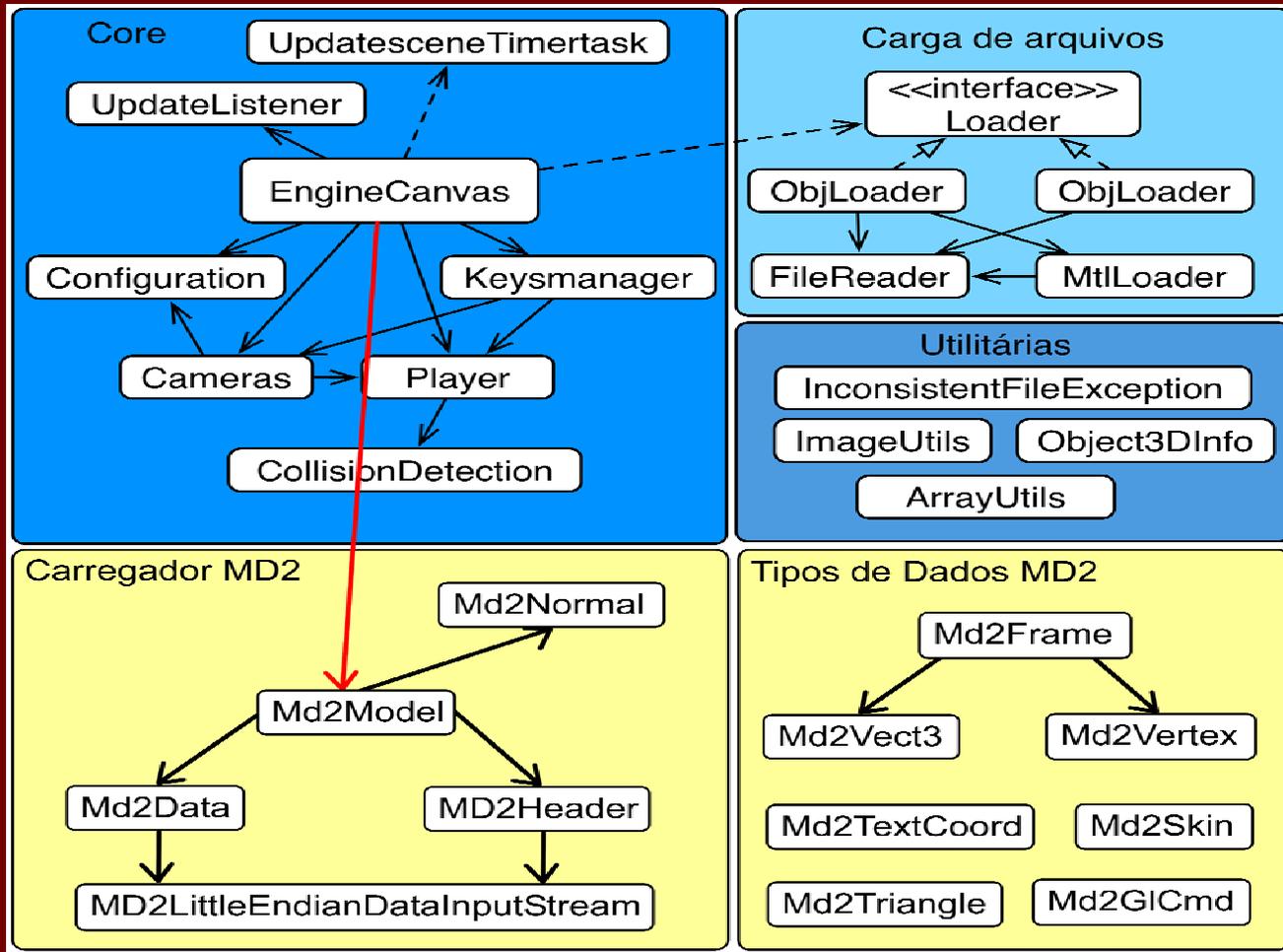
# 4. Implementação

## Visualização do Modelo



# 4. Implementação

## Adicionando o Carregador de MD2 a M3GE



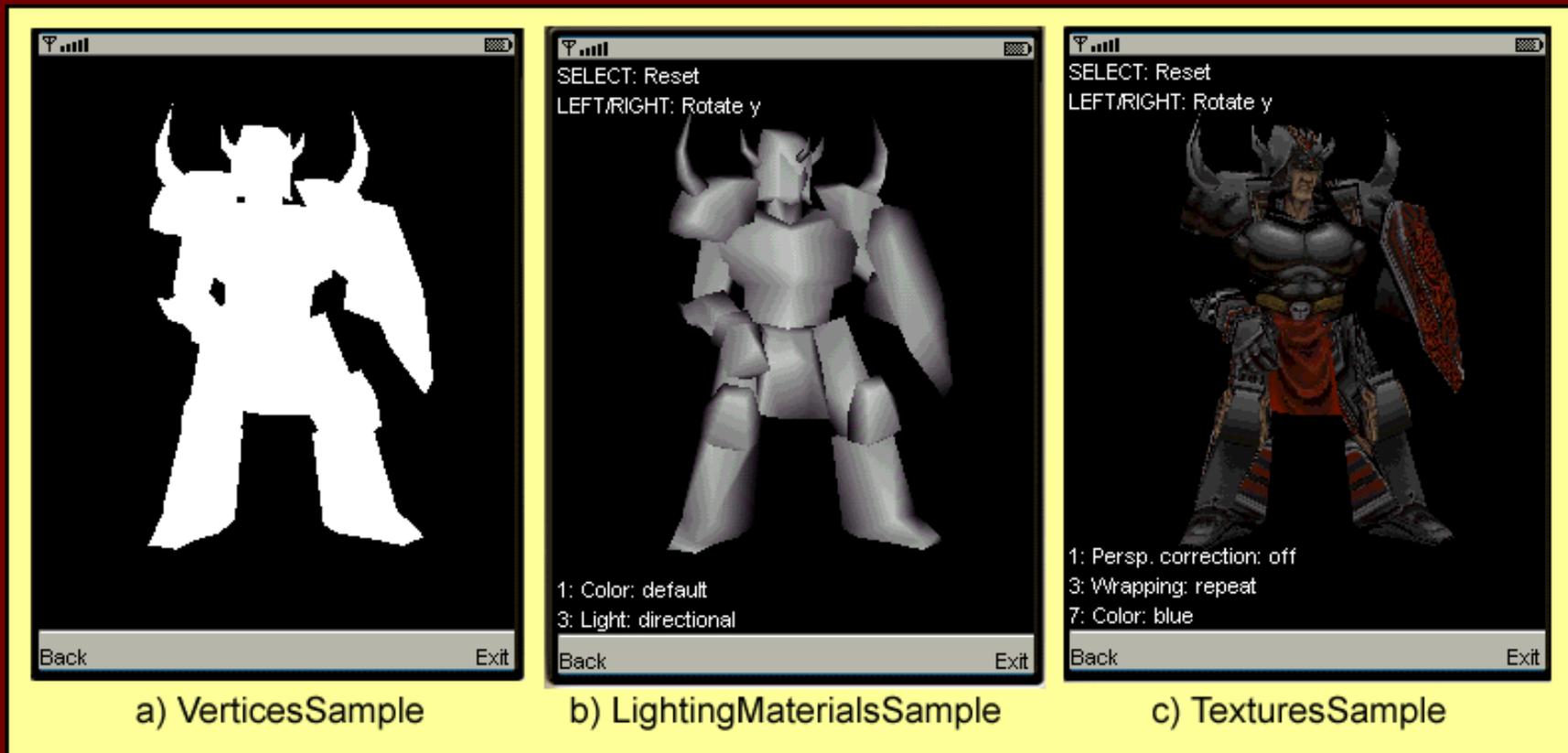
# 4. Implementação

## Testes Durante a Implementação

- Plugin EclipseME;
- Visualização do modelo foi feita passo a passo:
  - Primeiramente, foram utilizado apenas a lista de triângulos com o índice para os vértices;
  - Posteriormente foi adicionada a lista dos vértices normais do modelo;
  - Por último foi feita a adição de textura ao modelo.

# 4. Implementação

## Testes Durante a Implementação



# 4. Implementação

## Resultados e Discussão

- Jogo exemplo da M3GE;
- Foram adicionados três personagens a cena;
- Dois arquivos MD2 iguais, para duas texturas diferentes;
- Toda funcionalidade continuou funcionando.

# 4. Implementação

## Resultados e Discussão



# 5. Conclusão

- Boa qualidade gráfica;
- Atualmente inviável;
- Demora em renderizar os personagens;
- Protótipo de jogo;
- Mais de um PNJ em cena;
- Ferramentas e tecnologias utilizadas facilitaram os testes;

# 5. Conclusão

- Possível utilizar o arquivo MD2 para a manipulação de personagens em motores de jogos para dispositivos móveis;
- Apesar da demora na execução.

# 5. Conclusão

## Extensões

- Análise e verificação das técnicas utilizadas;
- Simplificar e reduzir o número de vértices (LOD – Level of Detail);
- Utilizar os quadros do arquivo MD2 para a obtenção de animação;

# 5. Conclusão

## Extensões

- Incluir o carregador de MD2 em outro *framework*;
- Implementar Inteligência Artificial nos PNJs.

Fim