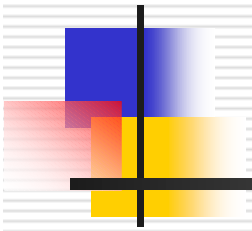
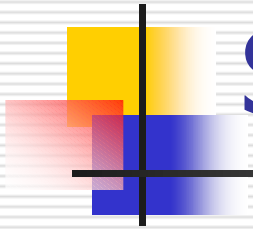


Protótipo de um visualizador de modelos 3D para dispositivos móveis utilizando a plataforma .NET CF 2.0

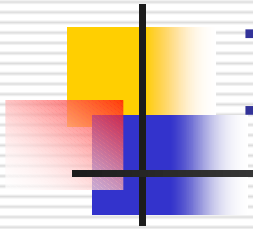
Marcos Dell' Antonio de Souza

Orientador Prof. Dr. Paulo César Rodacki Gomes



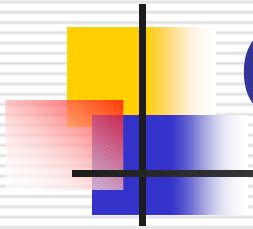
Sumário

- Introdução
- Objetivos do trabalho
- Fundamentação teórica
- Desenvolvimento do trabalho
- Conclusões



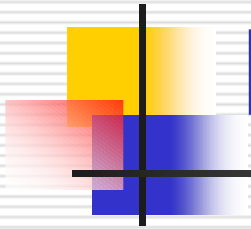
Introdução

- Modelos 3D (corpos formados por um conjunto de vértices - Santee)
- Representação digital de modelos 3D
- Dispositivos móveis (evolução)
- Desenvolvimento utilizando um framework
- Visualizador de modelos 3D
- Managed Direct3D Mobile (MD3DM)



Objetivos do trabalho

- Modelo Wavefront (OBJ/MTL)
- Ler, carregar e renderizar
- Navegar pelo modelo
- Alterar o modelo (rotação, translação e escala)
- Validar a biblioteca MD3DM através da métrica de frames por segundo (FPS)



Fundamentação Teórica

- Tópicos discutidos
  - História
  - Windows Mobile, .NET CF e Visual Studio
  - Dispositivos móveis
  - Modelos 3D
  - Managed Direct3D Mobile
  - Formato Wavefront (OBJ / MTL)
  - Trabalhos correlatos

História

- GAPI, GDI e Managed Direct3D Mobile
- Windows Mobile 2003, 5.0 e 6.0

Windows Mobile 2003

Windows Mobile 5.0

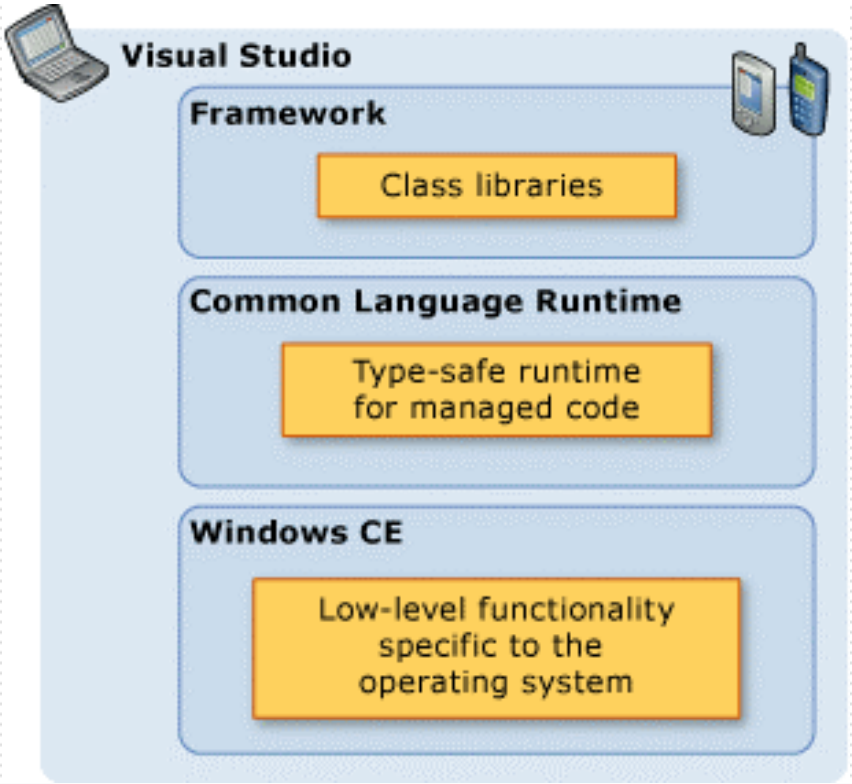
Windows Mobile 6.0

Windows Mobile, .NET CF e Visual Studio

- **Passado:**
  Visual Studio 2003
  Windows Mobile 2003
  .NET CF 1.0
  GAPI e GDI

- **Presente:**
  Visual Studio 2005
  Windows Mobile 5.0 e 6.0
  .NET CF 2.0
  Managed Direct3D Mobile

- **Futuro (?):**
  Visual Studio - Orcas
  Windows Mobile 7.0
  ".NET CF 3.x"
  "XNA Mobile"



Arquitetura do .NET CF

Dispositivos móveis

- Processador, memória, limitações, etc
- 2,6 bilhões de celulares no mundo até o final de 2006

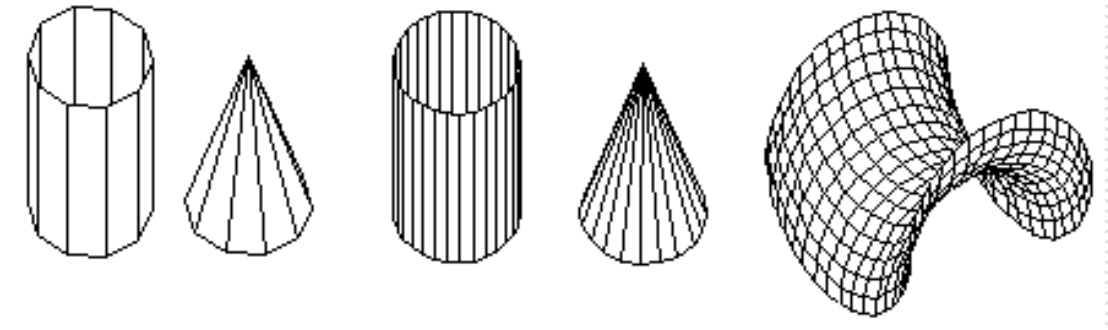
Teclado QWERTY

DELL Axim x51v

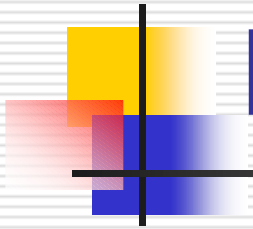
iPhone

Modelos 3D

- Malhas de polígonos
- Diversos formatos (Wavefront, .X, m3g, MBJ, etc)
- 3D Studio Max, LightWave e Blender

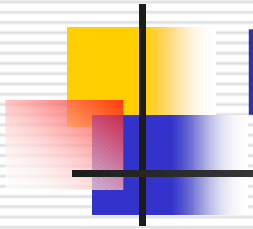


Modelagem utilizando malhas de polígonos



Managed Direct3D Mobile

- Classes e estruturas gerais
  - Device
  - PresentParams
  - VertexBuffer
  - IndexBuffer
  - Mesh
  - AttributeRange
  - Matrix
- Materiais e texturas
  - Material
  - Texture

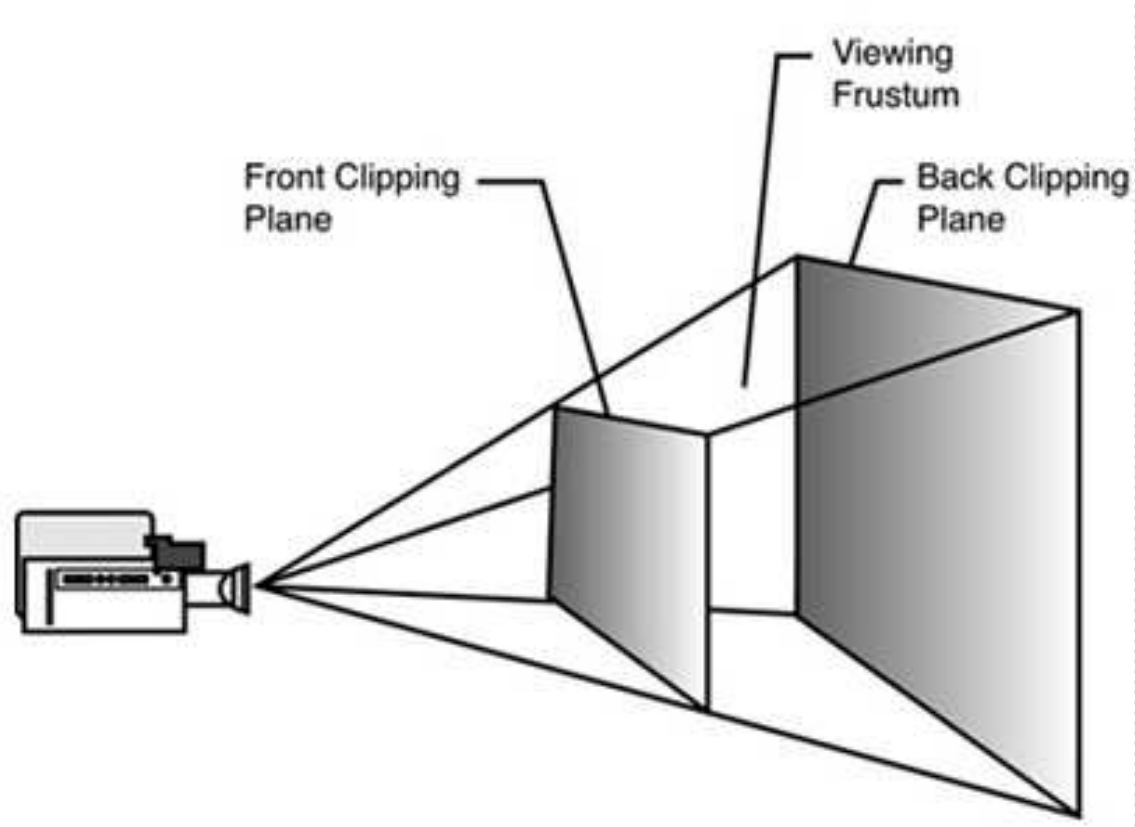


Managed Direct3D Mobile

- Propriedade Transform do Device (câmera)
  - View (posicionamento e direção da câmera)
  - Projection (view frustum e campo de visão)
  - World (transformações)

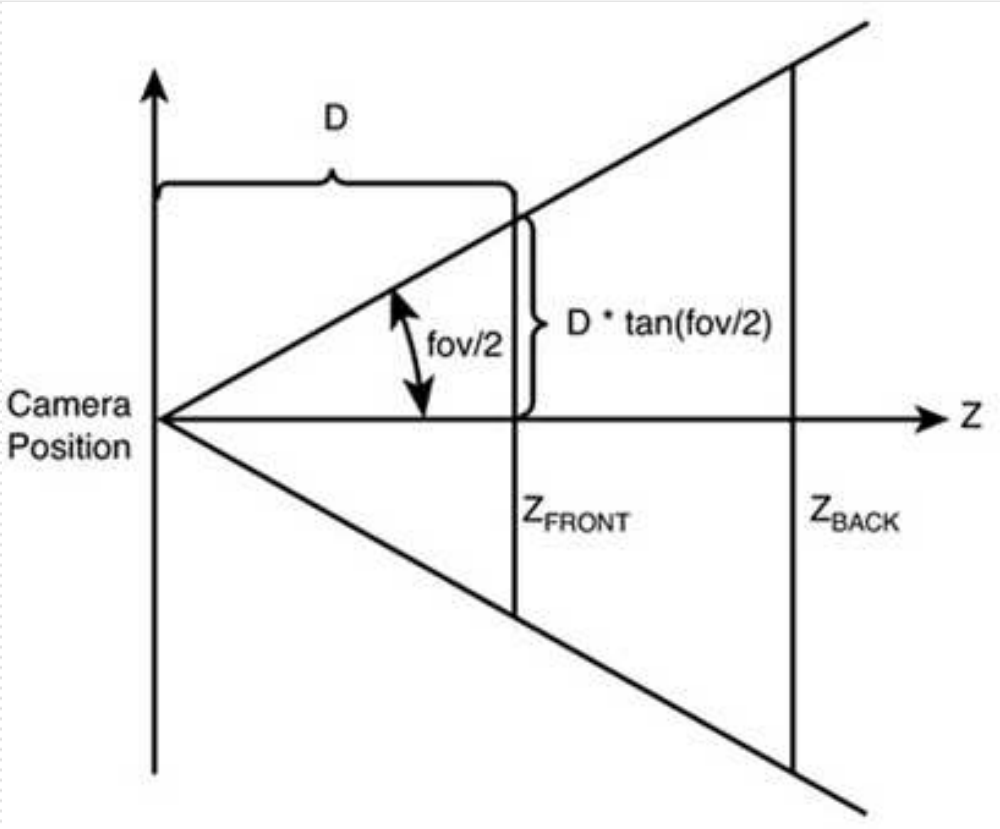
Managed Direct3D Mobile

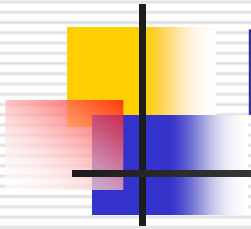
- View Frustum



Managed Direct3D Mobile

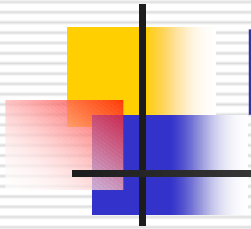
- Field of view (campo de visão) - ângulo





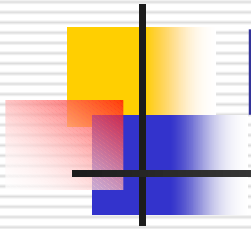
Managed Direct3D Mobile

- Classe Matrix
  - LookAtLH(position, target, upVector)
  - PerspectiveFovLH(fovY, aspectRatio, zNear, zFar)
  - RotationX(angle)
  - RotationY(angle),
  - RotationZ(angle),
  - Scaling(x, y, z)
  - Translation(x, y, z)



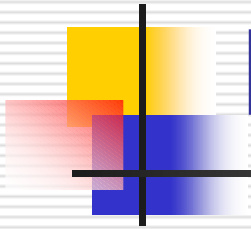
Formato Wavefront

- Arquivo de texto puro
- Especificação aberta
- Suportado por diversos softwares
- Composto por dois arquivos: OBJ e MTL



Formato Wavefront (OBJ)

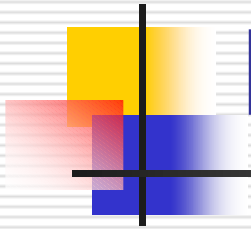
- Principais comandos do arquivo OBJ:
  - #
  - v <x> <y> <z>
  - vn <x> <y> <z>
  - vt <u> <v>
  - g <nome>
  - f <v1>/<vt1>/<vn1> ... ...
  - mtllib <arquivo>
  - usemtl <nome>



Formato Wavefront (OBJ)

- Exemplo de um arquivo OBJ

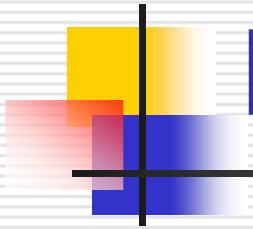
```
mtllib teste.mtl
v  0.000000 2.000000 0.000000
v  0.000000 0.000000 0.000000
v  2.000000 0.000000 0.000000
v  2.000000 2.000000 0.000000
vt 0.000000 1.000000 0.000000
vt 0.000000 0.000000 0.000000
vt 1.000000 0.000000 0.000000
vt 1.000000 1.000000 0.000000
g grupo1
usemtl material1
f 1/1 2/2 3/3 4/4
```



Formato Wavefront (MTL)

- Principais comandos do arquivo MTL:
  - #
  - newmtl <nome>
  - Ka <r> <g> <b>
  - Kd <r> <g> <b>
  - Ks <r> <g> <b>
  - d <alpha>
  - map_Ka <arquivo>
  - map_Kd <arquivo>
  - map_Ks <arquivo>

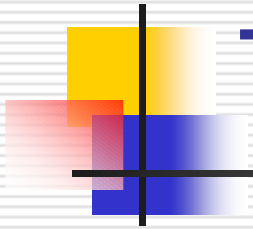
a = ambiente; d = difusa; s = especular



Formato Wavefront (MTL)

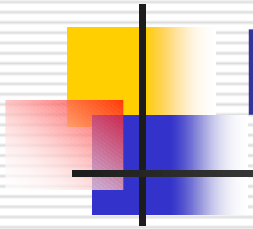
- Exemplo de um arquivo MTL

```
newmtl Material
Ka 0.565000 0.875000 0.141000
d 1.000000
```



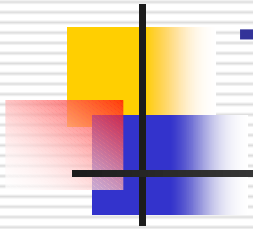
Trabalhos correlatos

- mOGE – Mobile Graphics Engine
  - Gerenciador de entrada, IA, objetos, mundo
  - OpenGL ES / Symbian
- M3GE – Mobile 3D Game Engine
  - Mobile 3D Graphics (M3G)
  - Implementa a JSR 184: Mobile 3D Graphics API for J2ME
  - Wavefront (OBJ e MTL) e MBJ



Desenvolvimento do trabalho

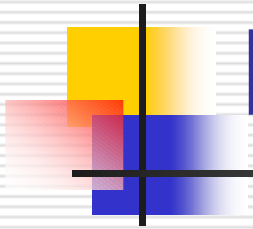
- Tópicos discutidos
  - Técnicas utilizadas e evitadas
  - Requisitos
  - Ferramentas e tecnologias utilizadas
  - Gita e 3DV



Técnicas utilizadas e evitadas

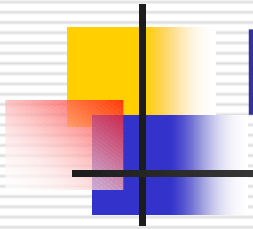
- YAGNI (XP): You Aren't Gonna Need It
- KISS: Keep It Single, Stupid

- Orientação a objetos
  - Heranças demasiadas
  - Sobrecarga excessiva de métodos
  - Padrões de projeto complexos



Requisitos

- Carregar um modelo 3D no formato Wavefront
- Navegar pelo modelo carregado
- Executar operações de translação, rotação e escala
- Ser portável entre os dispositivos compatíveis com o WM 5.0 e 6.0 (.NET)

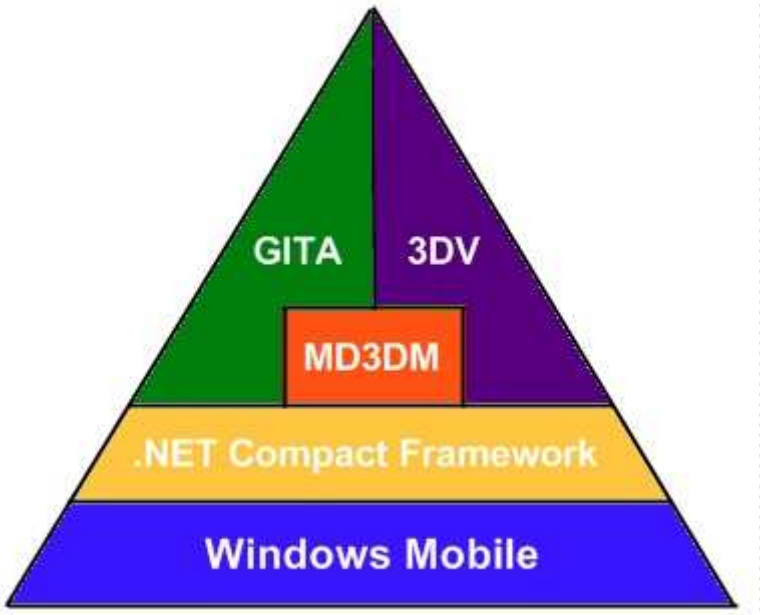


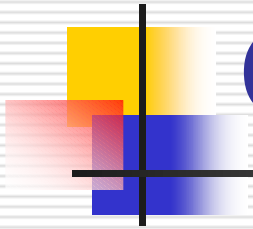
Ferramentas e tecnologias

- Visual Studio 2005 e Enterprise Architect
- SDKs e emuladores do WM 5.0 e 6.0
- .NET CF 2.0 e MD3DM
- HTC S620: 200MHz e 64MB RAM
- Art of Illusion e 3D Studio Max (trial)
- Notebook AMD Turion 1.6GHz e 1GB Ram

Gita e 3DV

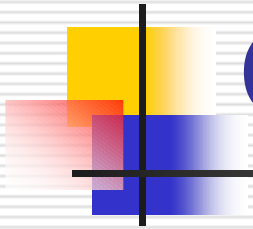
- Gita: biblioteca que carrega e manipula o modelo
- 3D Viewer: protótipo de um visualizador de modelos 3D que utiliza a Gita





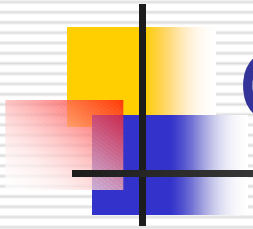
Gita

- Componentes
  - ObjLoader
  - Transformations
  - Camera
  - Common



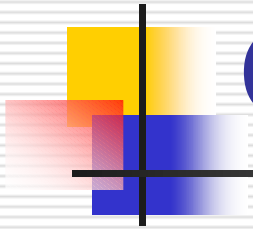
Gita – Componente ObjLoader

- Classes
  - Face
  - Vertex
  - VertexNormal
  - VertexTexture
  - ObjLoader
  - MtlLoader



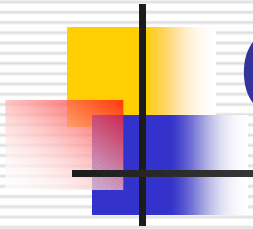
Gita – Componente Transformations

- Classes : ITransformation
  - Rotation
  - Translation
  - Scale
- TransformationFactory
- TransformKind
- Params



Gita – Componente Camera

- Classes
  - Camera
  - CameraList
- Arquivo XML: define as câmeras
- Bouding Box: câmera padrão



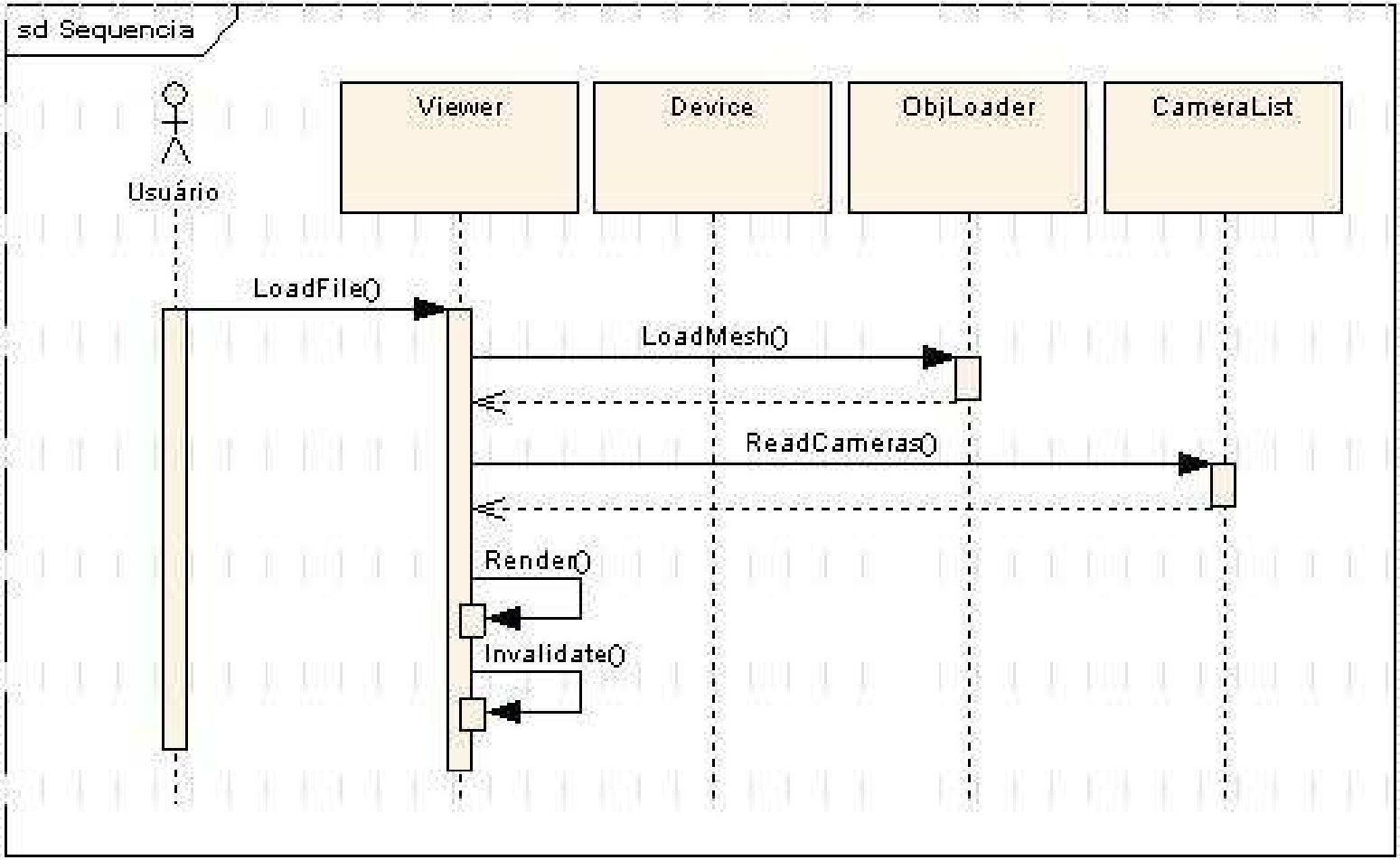
Gita – Componente Common

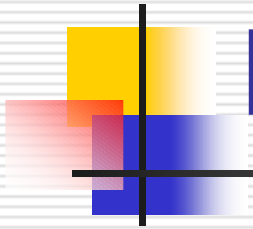
- Classes
  - Utils
  - Bouding box

3D Viewer

- Funções
  - Load file
  - Transform
  - Navigate
  - Reset

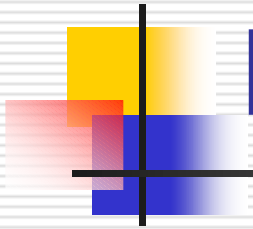
3D Viewer – Diagrama de seqüência





Resultados e discussão

- Tópicos discutidos
  - Modelos 3D usados nos testes
  - Resultados dos testes
    - Emulador do Visual Studio 2005
    - Dispositivo real HTC S620
    - Comparativo Gita e M3GE



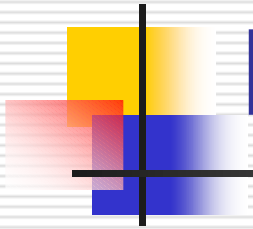
Modelos 3D usados nos testes

	Simples (cubo)	Complexo (nave)
Faces	12	450
V, VN e VT	15	2100
Textura (pixels)	-	256 x 256

Modelos usados nos testes

Modelo complexo

Modelo simples



Resultados dos testes

	Emulador	Dispositivo
TC Simples	4s	1s
FPS Simples	1fps	30fps
TC Complexo	20s	5s
FPS Complexo	1fps	15fps

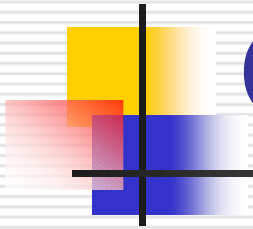
TC = Tempo para carregar o modelo

Comparativo Gita e M3GE

- Tempos diferentes para carregar o modelo
- Desempenho real (dispositivo) semelhante

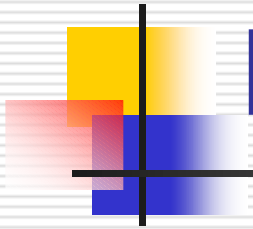
	Gita	M3GE
TC	5s	17s
FPS	15fps	15 – 20fps

TC = Tempo para carregar o modelo



Conclusões

- Primeira implementação para .NET CF
- Satisfaz, em partes, as necessidades dos desenvolvedores
- Os objetivos foram alcançados
- Dificuldades
  - Falta de emuladores
  - Falta de documentação



Extensões

- Transformar o projeto em um motor de jogos 3D
  - Suporte a novos formatos (X, MD2, etc)
  - Detecção de colisão: comparativo de diversas técnicas
  - Componente de iluminação