

Física de corpos rígidos para o motor de jogos M3GE na plataforma J2ME

Ernani Cristiano Siebert – Acadêmico

Dalton Solano dos Reis - Orientador

ROTEIRO

- ▶ Introdução
 - ▶ Objetivos do trabalho
 - ▶ Fundamentação teórica
 - ▶ Contexto atual do tema
 - ▶ Requisitos e Especificação
 - ▶ Técnicas e ferramentas utilizadas
 - ▶ Operacionalidade da implementação
 - ▶ Resultados e discussão
 - ▶ Conclusões e Extensões
- 

INTRODUÇÃO

- ▶ Vantagens J2ME
 - Difusão da Plataforma
 - Custo Reduzido
 - Facilidade de Compra
- ▶ Tipos de jogos
 - Versão Mobile
 - Marketing Viral
 - Puzzles

OBJETIVOS DO TRABALHO

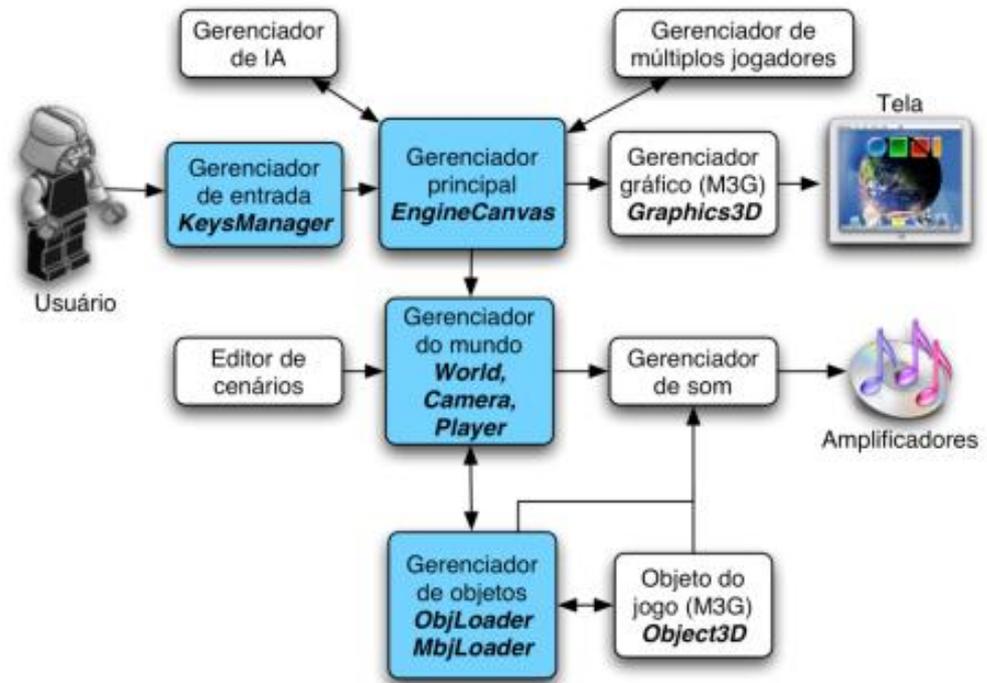
- ▶ Adicionar física de corpos rígidos para o motor M3GE
 - Detecção de colisão
 - Resposta de colisão

FUNDAMENTAÇÃO TEÓRICA

Motores

▶ Funcionalidades comuns:

- Som
- IA
- Entrada



J2ME

▶ Otimização

- Processamento
- Memória

▶ Unificação das plataformas

Biblioteca M3G

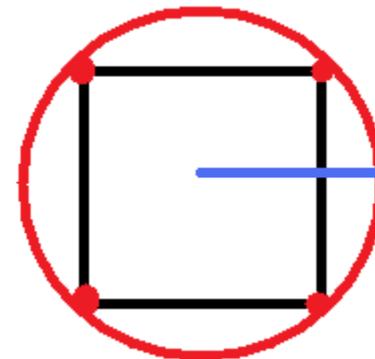
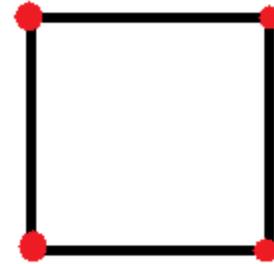
- ▶ Funcionalidades
 - desenhar em 3D
 - grafo de cena
 - imediato

Motor M3GE

- ▶ TCC Pamplona (2006)
- ▶ Funcionalidades
 - Gerenciador de entrada
 - Mundo
 - Câmera
 - Jogador
 - Gerenciador de objetos

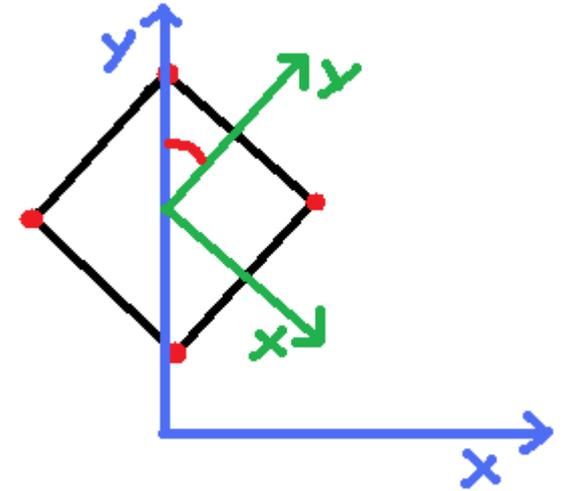
DINÂMICA DE CORPOS RÍGIDOS

- ▶ Mecânica
 - Estudo dos corpos em movimento
 - Forças agindo sobre eles
- ▶ Corpo rígido
 - Partículas numa distância fixa
- ▶ Detecção de colisão
 - Geometria
 - Testar cada par de objetos
 - Probabilidade de contato
 - Coarse collision detection
 - Fine collision detection



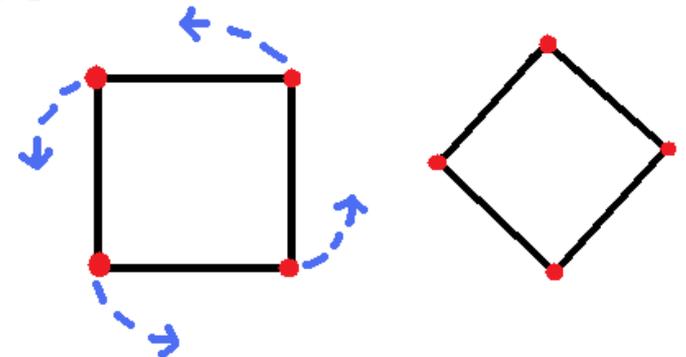
ROTAÇÃO ANGULAR

- ▶ Ω Diferença entre eixo local e global
- ▶ rad
 - Velocidade ω
 - rad/s
 - Aceleração α
 - rad/s²



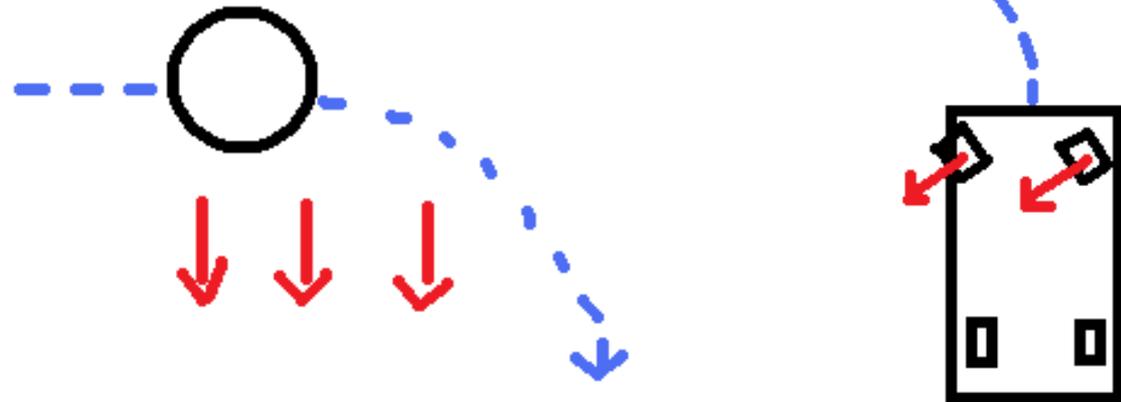
MOVIMENTO ANGULAR

- ▶ Trajetória circular das partículas
 - Comprimento do arco



ACELERAÇÃO CENTRÍPEDA

- ▶ Altera a direção
 - Força trajetória circular
- ▶ Causas
 - Gravidade
 - Fricção



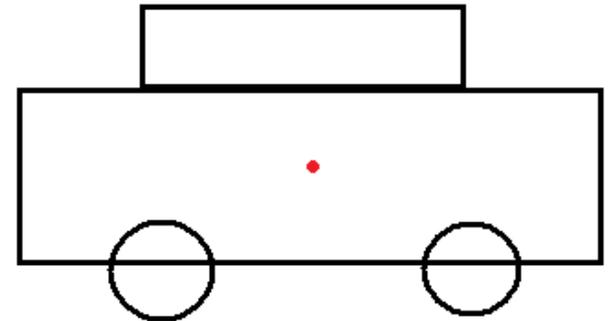
PROPRIEDADES DE MASSA

MASSA

- ▶ Resistência a mudar posição
- ▶ Massa total
 - Soma de massas não uniformes

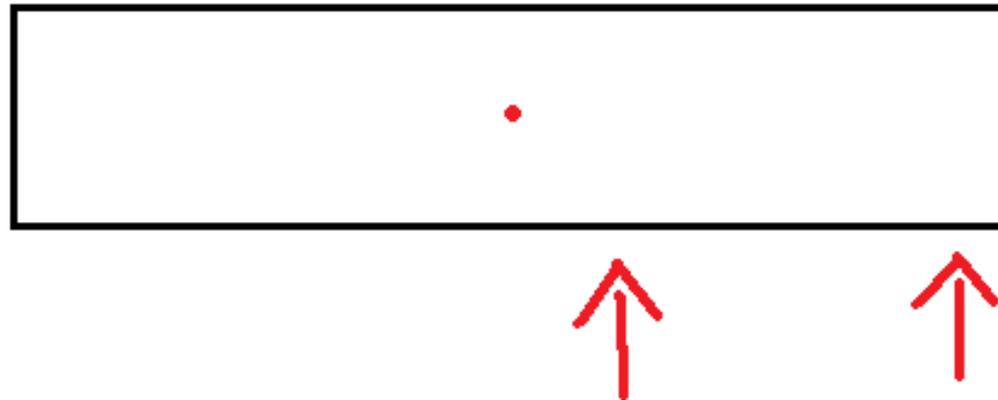
CENTRO DE MASSA

- ▶ Onde força age sem rotacionar
- ▶ Depende
 - Geometria
 - Distribuição da massa



MOMENTO DE INÉRCIA

- ▶ Distribuição/concentração da massa
- ▶ Resistência num eixo de rotação
- ▶ Depende
 - Distância do centro



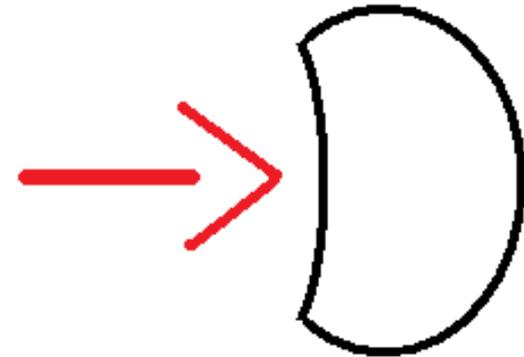
RESPOSTA DE COLISÃO

- ▶ Idealização
 - não deformam
- ▶ Rastrear
 - Translação
 - Rotação
- ▶ Procedimento
 - Propriedades de massa
 - Forças
 - Aceleração linear e angular
 - Velocidade linear e angular
 - Deslocamento linear e angular

RESPOSTA DE COLISÃO

▶ IMPULSO

- Força agindo num período curto

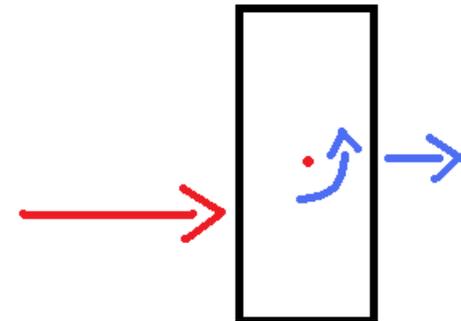


▶ IMPACTO

- Corpos rígidos conservam momento
- Elasticidade
 - Tempo de separação

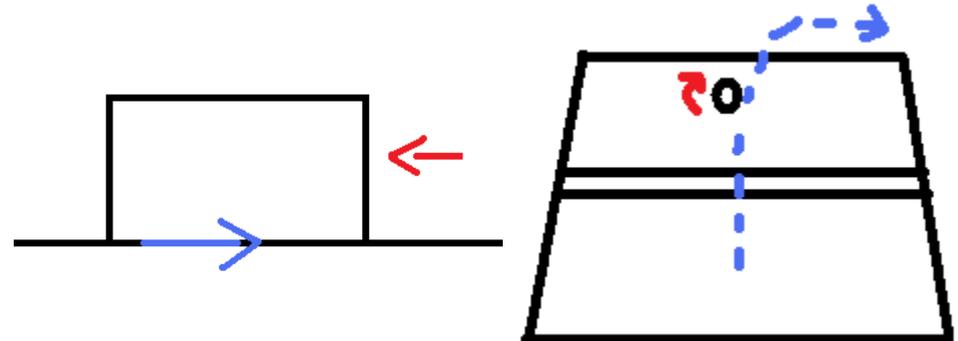
▶ IMPULSO LINEAR E ANGULAR

- Considera
 - Distância do centro
 - Força no ponto de colisão



▶ ATRITO

- Age entre superfícies
- Resistência à movimentação
- Altera rotação



CONTEXTO ATUAL DO TEMA

- ▶ MOTORES JAVA 3D
 - Apenas desktop
- ▶ J2ME
 - Apenas 2D
 - Emini
 - DyMix

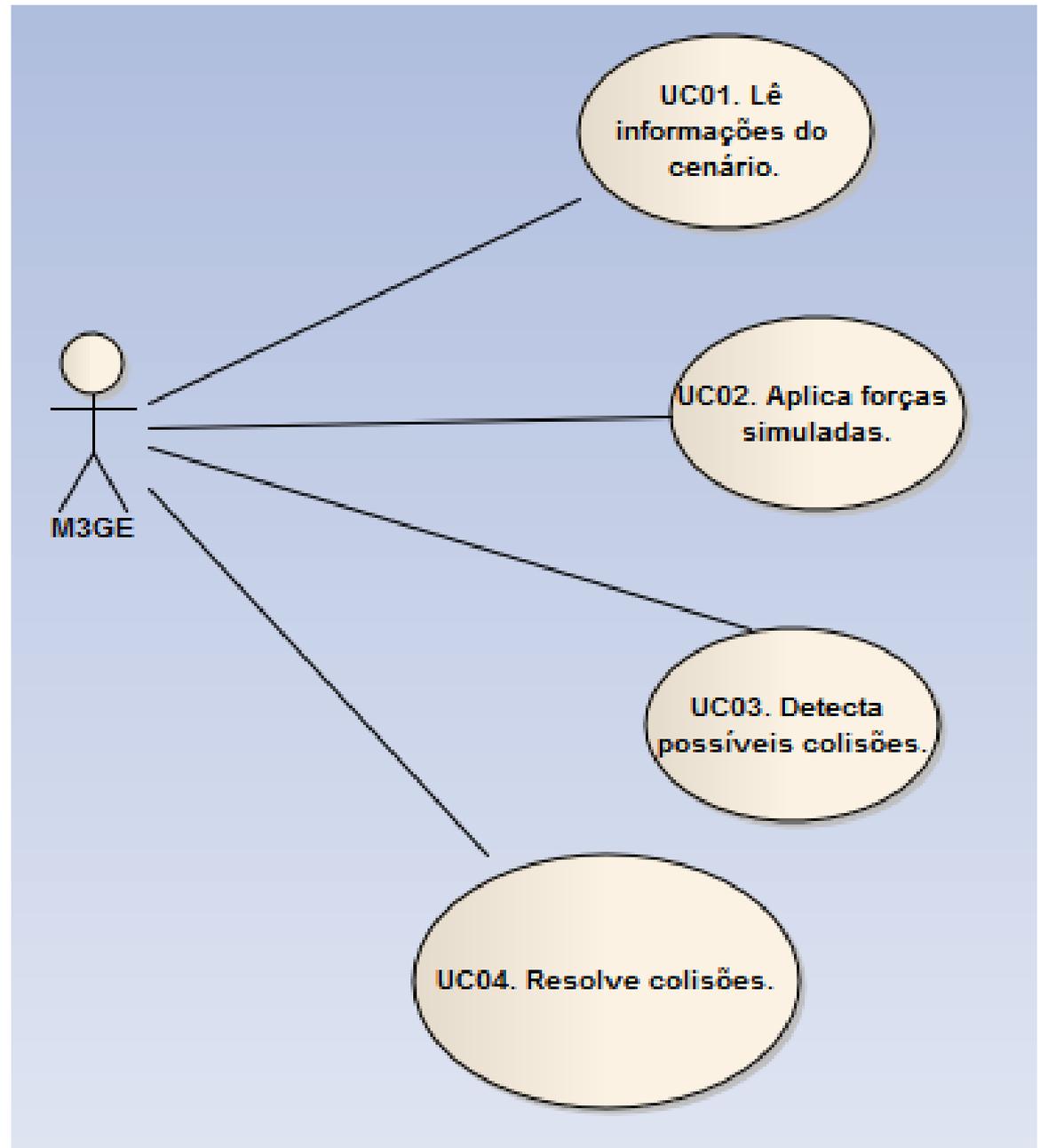
REQUISITOS PRINCIPAIS

- ▶ compatível com o motor M3GE (RF)
 - ▶ armazenar informações sobre os objetos na tela (RF)
 - ▶ atualizar as forças envolvidas na simulação (RF)
 - ▶ detectar colisões que ocorrem entre os objetos e armazenar as informações (RF)
 - ▶ atualizar a posição dos objetos na tela (RF)
- 

ESPECIFICAÇÃO

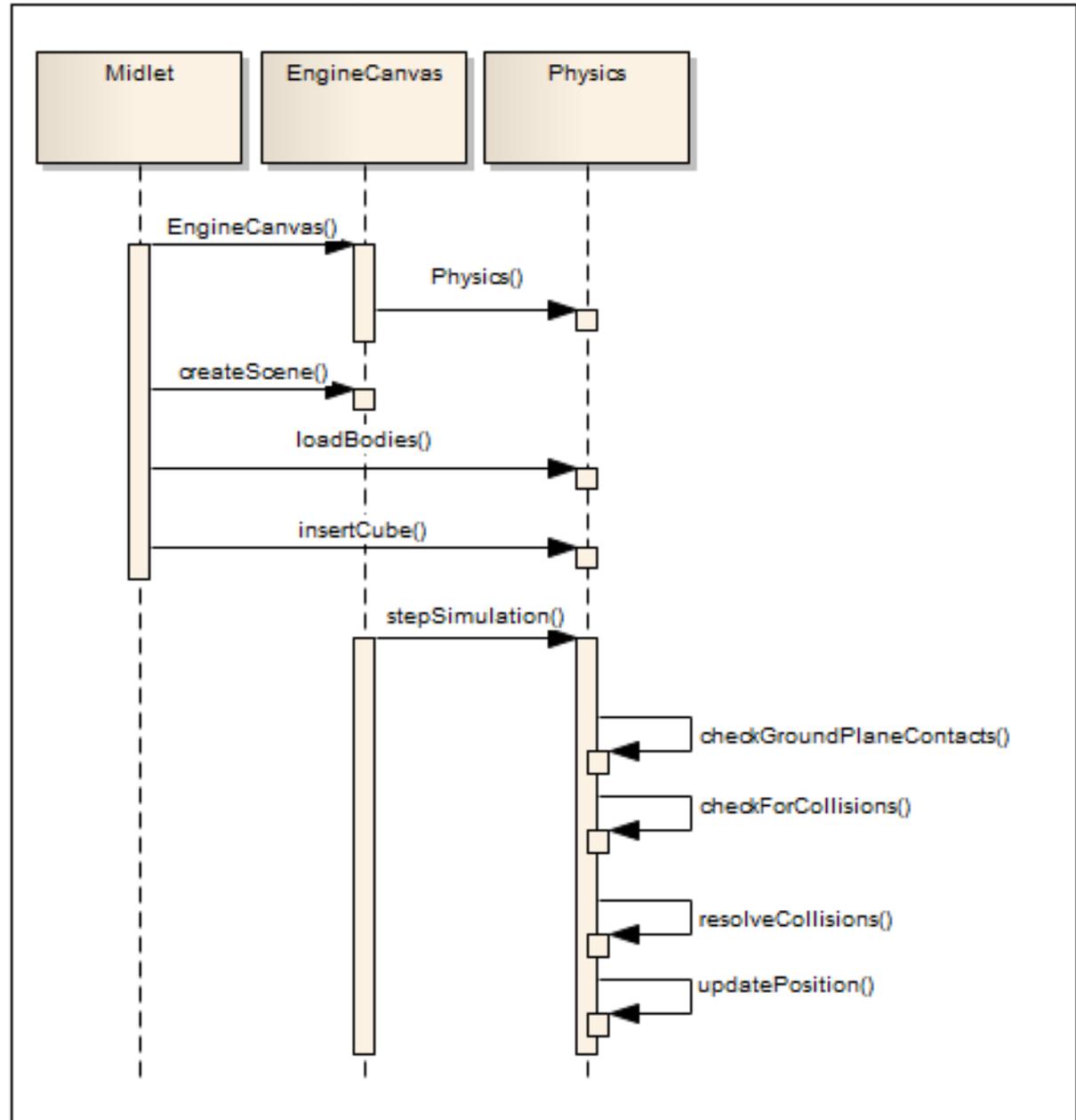
ESPECIFICAÇÃO

► Casos de Uso



ESPECIFICAÇÃO

▶ Diagrama de Sequência



ESPECIFICAÇÃO

▶ CLASSES DE OPERAÇÕES

- Vetor
- Matriz
- Quaternion

▶ CLASSES AUXILIARES

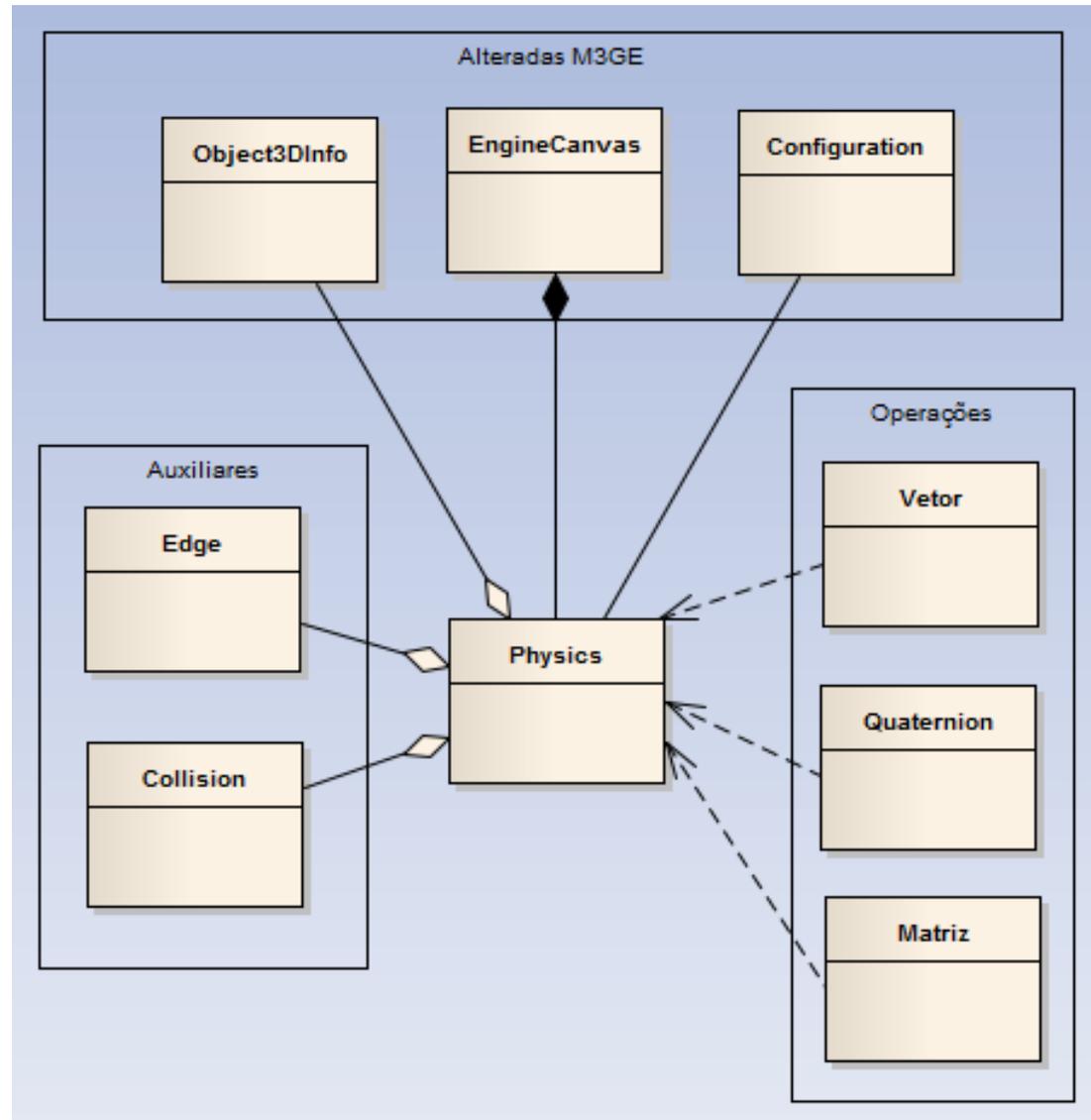
- Collision
- Edge

▶ CLASSES ALTERADAS

- EngineCanvas
- Configuration
- Object3dinfo

▶ CLASSE PRINCIPAL

- Physics



TÉCNICAS E FERRAMENTAS UTILIZADAS

- ▶ Java Wireless Toolkit (WTK)
 - CLDC
- ▶ Eclipse

TÉCNICAS E FERRAMENTAS UTILIZADAS

▶ Integração

- Intervalo de tempo
 - Calcula posição final
- Testa colisão
 - Calcula força de resposta
- Penetração
 - Diminui o passo
 - Calcula posição original

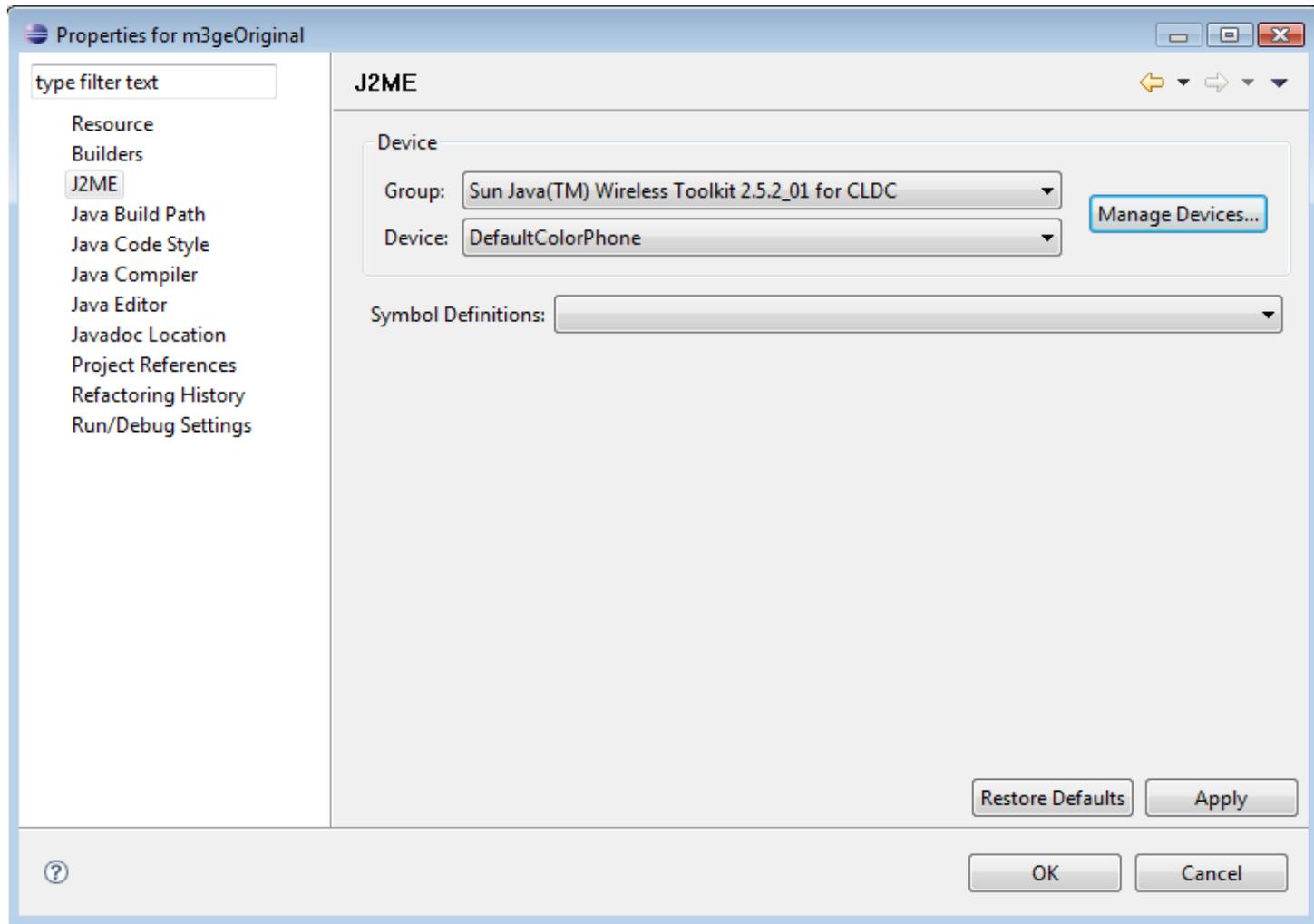
▶ Detecção

- Bound Box
 - Aresta-chão
 - Aresta-face

OPERACIONALIDADE DA IMPLEMENTAÇÃO

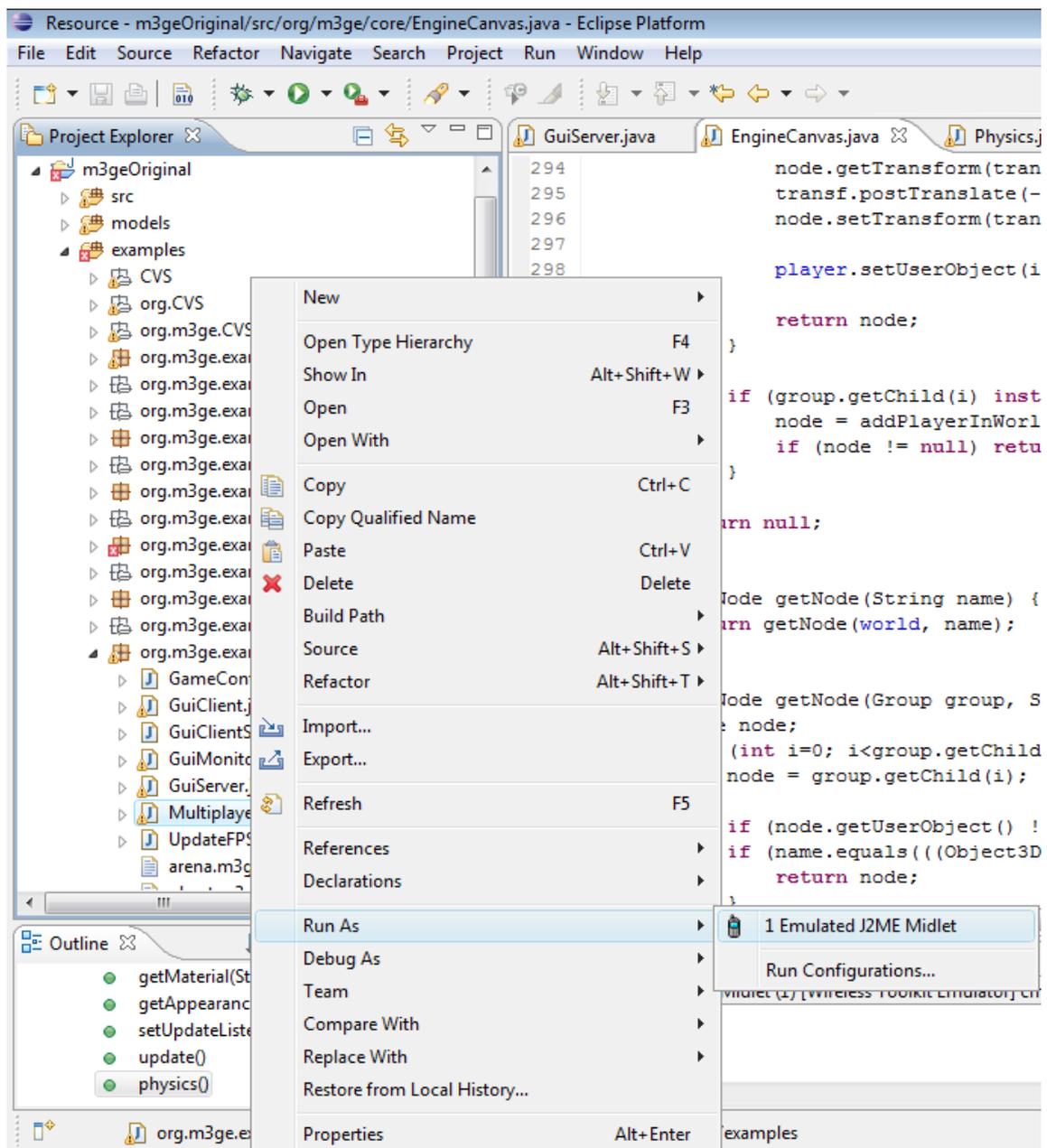
OPERACIONALIDADE DA IMPLEMENTAÇÃO

► Configuração da WTK



OPERACIONALIDADE DA IMPLEMENTAÇÃO

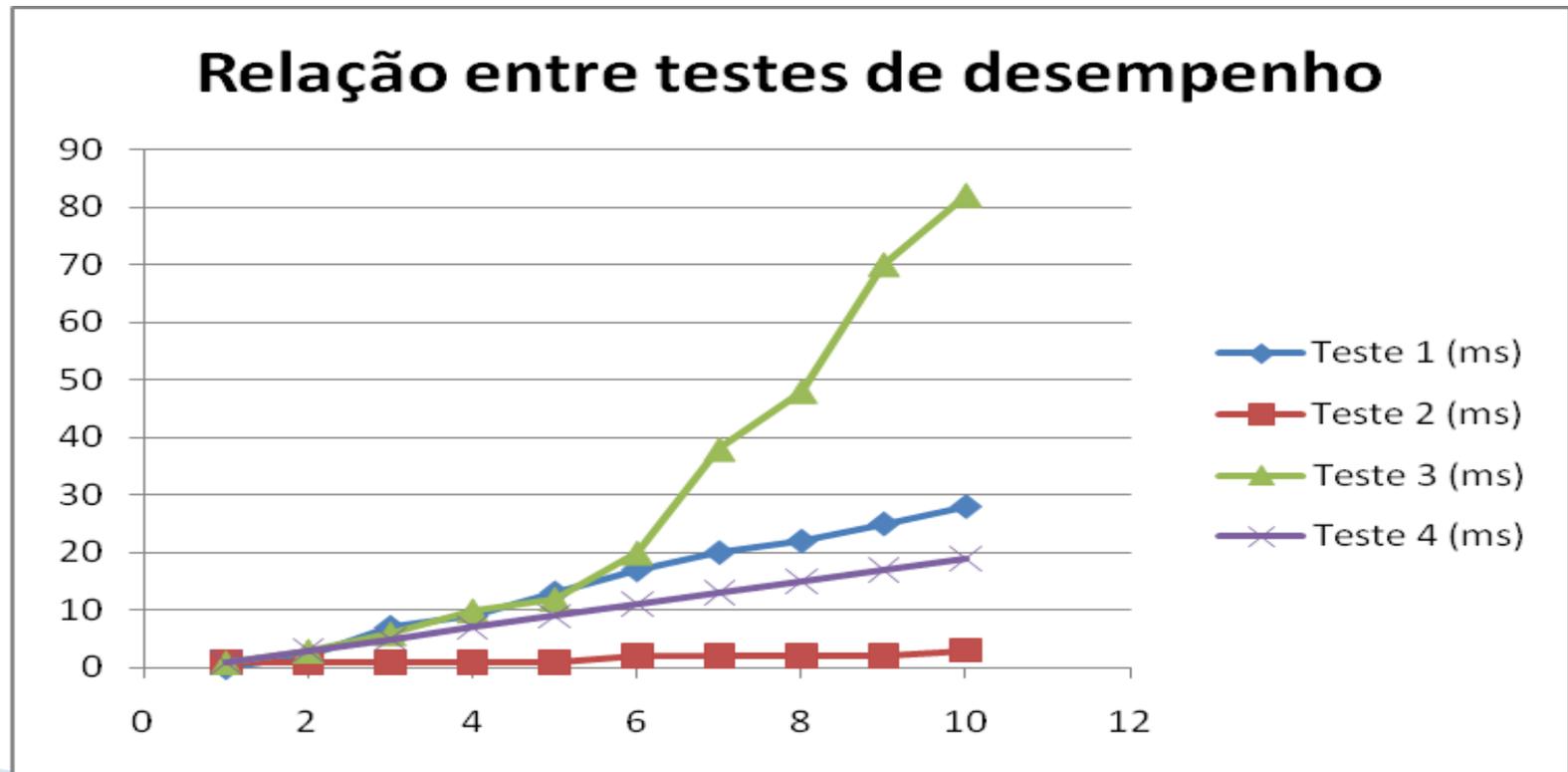
▶ Execução do emulador



RESULTADOS E DISCUSSÃO

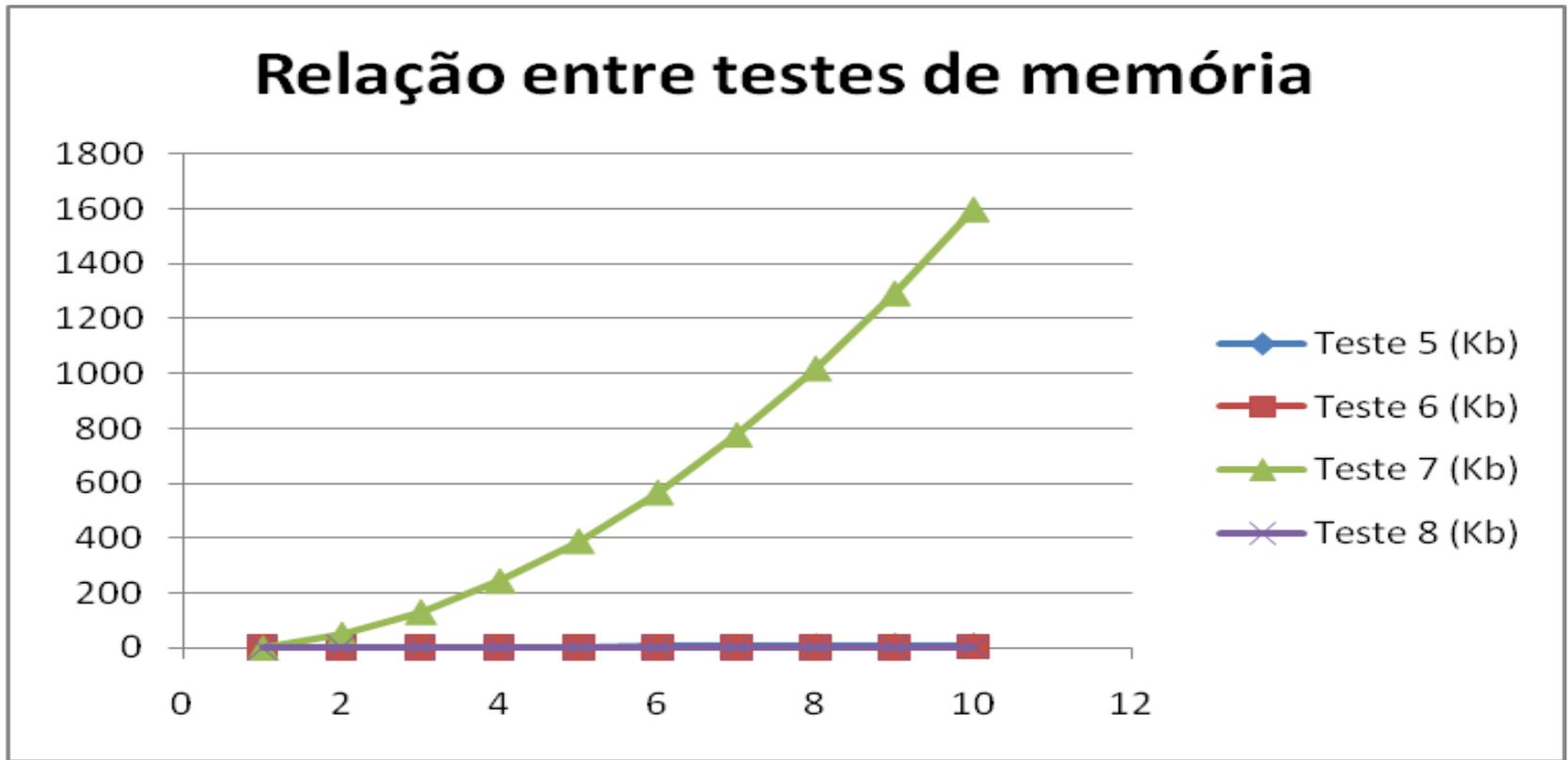
RESULTADOS E DISCUSSÃO

- ▶ Testes de Desempenho Prático
 - Teste 1: Face-chão, Detecção
 - Teste 2: Face-chão, Resposta
 - Teste 3: Face-Vértice, Detecção
 - Teste 4: Face-Vértice, Resposta



RESULTADOS E DISCUSSÃO

- ▶ Teste de uso de memória
 - Teste 5: Face-chão; Detecção
 - Teste 6: Face-chão; Resposta
 - Teste 7: Face-Vértice; Detecção
 - Teste 8: Face-Vértice; Resposta



RESULTADOS E DISCUSSÃO

▶ Biblioteca restrita

- acos
- asin
- atan
- atan2

▶ Complexidade

- Vetores
- Matrizes
- Quatérnions

RESULTADOS E DISCUSSÃO

▶ Classes modificadas

- EngineCanvas
- Object3DInfo

▶ Classes Criadas

- Principal
 - Physics
- Auxiliares
 - Quaterion
 - Matriz
 - Vetor
 - Collision
 - Edge

CONCLUSÃO E EXTENSÕES

- ▶ Método apresentado por Boug
 - Bound Box
 - Exemplos em C++ pouparam tempo
- ▶ Algoritmos de detecção tem maior custo
 - Desempenho deve ser observado

EXTENSÕES

- ▶ Diferentes formas
 - Esferas
 - Côncavas
- ▶ Corpos macios
- ▶ Corpos deformáveis
- ▶ Ragdolls
- ▶ Sistema de partículas
- ▶ Otimização
 - Métodos rápidos para detectar colisão
 - Simplificar equações
 - Números inteiros

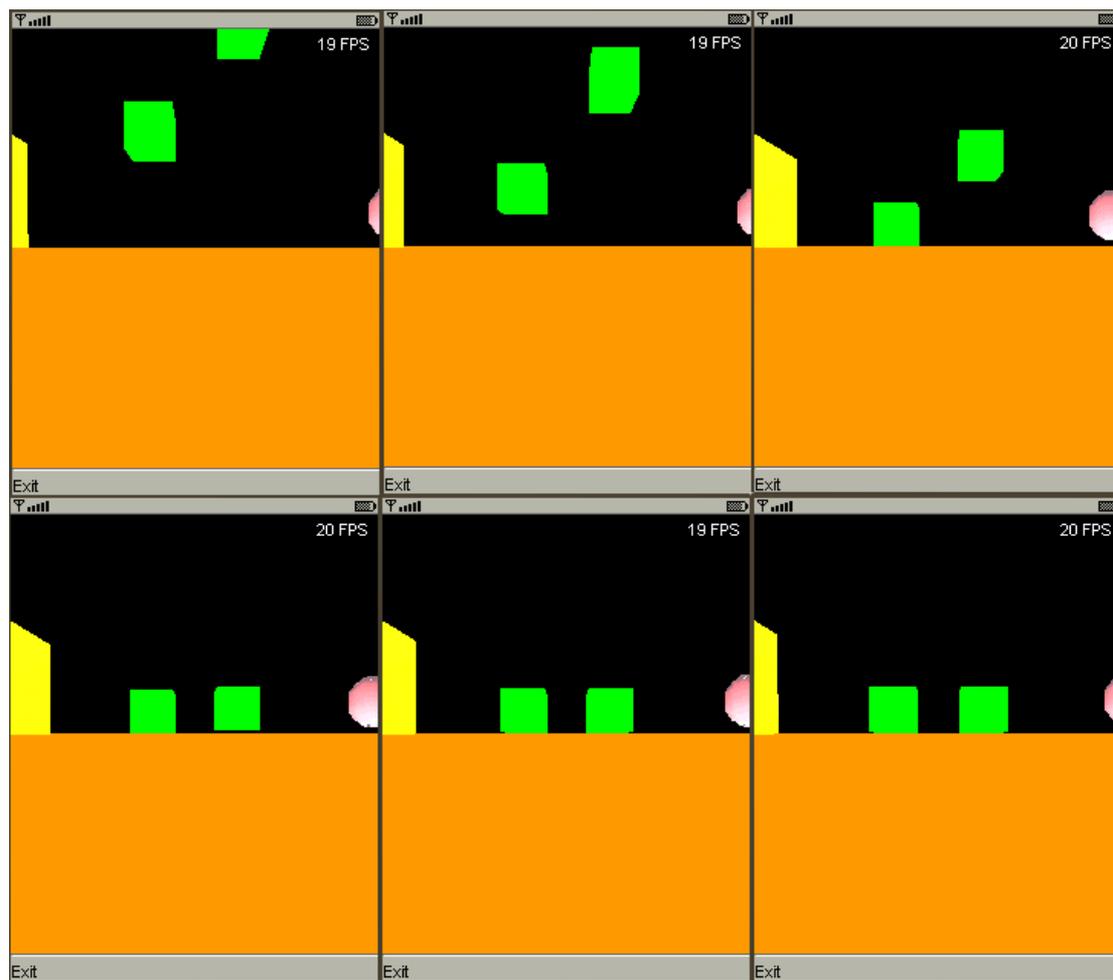
FIM

OPERACIONALIDADE DA IMPLEMENTAÇÃO

- ▶ Opções para iniciar a simulação



► Colisão com o chão



► Colisão com face

