Simulador Para Estacionamento de Carros Autônomos Não Articulados Usando Lógica Difusa

Acadêmico: Ewerton Rocha Machado

ewerton@inf.furb.br

Orientador: Mauro Marcelo Mattos
mattos@furb.br

dezembro de 2009

Roteiro da Apresentação

- Introdução
  - Contextualização e Objetivos
- Fundamentação Teórica
  - Lógica Difusa
  - JMonkey Engine
  - Trabalhos Correlatos
- Desenvolvimento do Simulador
  - Principais Requisitos
  - Especificação
  - Implementação
  - Resultados e Discussão
- Conclusão e Extensões

Introdução

Contextualização

- União entre a indústria automobilística e a área de automação.
- Dificuldades para estacionar.
- IA contribuindo para o aumento da realidade nos simuladores.
- Simulador provendo um ambiente atrativo para desenvolvimento e testes.

Objetivos do Trabalho

- Desenvolver uma aplicação que simule o processo de estacionamento de um veículo autônomo.

- Objetivos Específicos:
  - Construir uma base de regras difusas que possibilite a simulação do ato de estacionar;
  - Disponibilizar uma interface gráfica que permita o acompanhamento do processo de simulação.

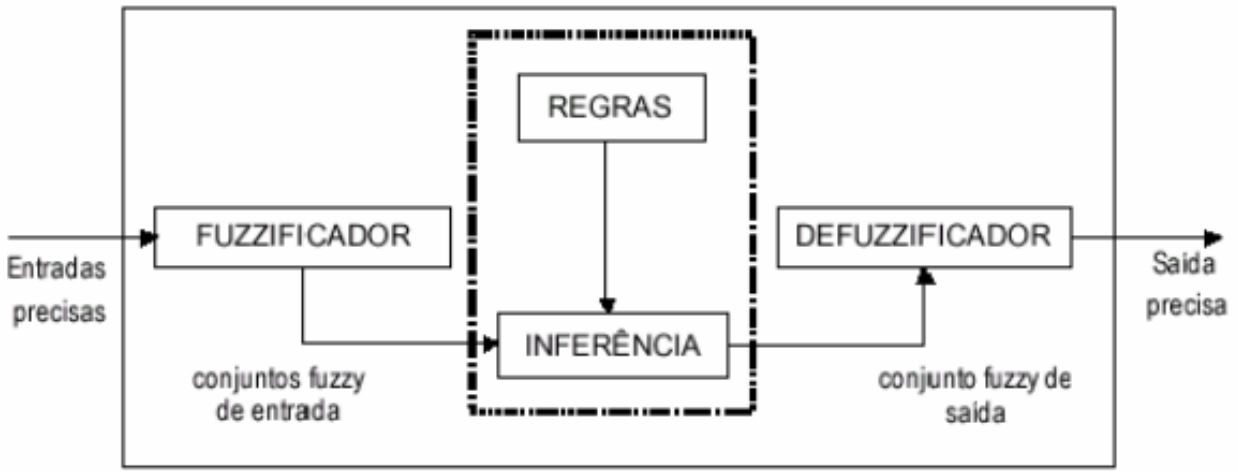
Fundamentação Teórica

Lógica Difusa

- Uma das diversas técnicas de implementação de inteligência artificial;
- Desenvolvida a partir de 1965 por Lotfi Zadeh, para tratar do aspecto do vago da informação;
- Tratar incertezas oriundas de informações parciais ou incompletas.
- Grau de pertinência em conjuntos.

Lógica Difusa

- Variáveis lingüísticas.
- Expressões qualitativas.
- Sistema de controle Fuzzy:



Lógica Difusa

- FuzzyJ ToolKit:
  - Permite lidar com conceitos nebulosos.
  - Explora as idéias da lógica Fuzzy em um ambiente Java.

Lógica Difusa

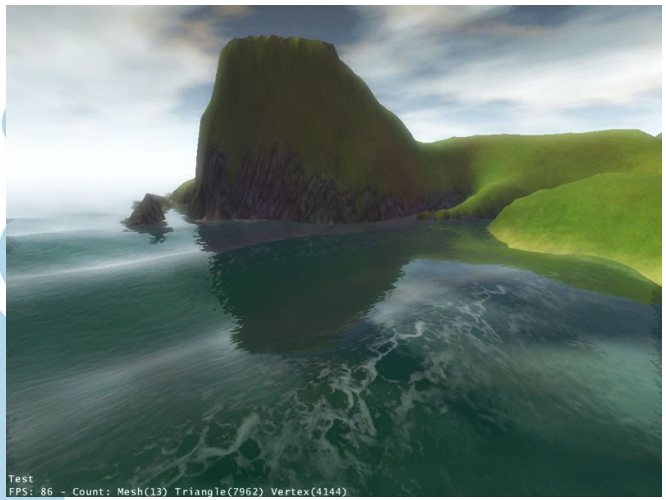
- Exemplo de áreas benefíciadas pelo uso da lógica difusa:
  - Câmeras de vídeo;
  - Máquinas de lavar roupa;
  - Aparelhos de ar-condicionado;
  - Indústria automobilística.

JMonkey Engine

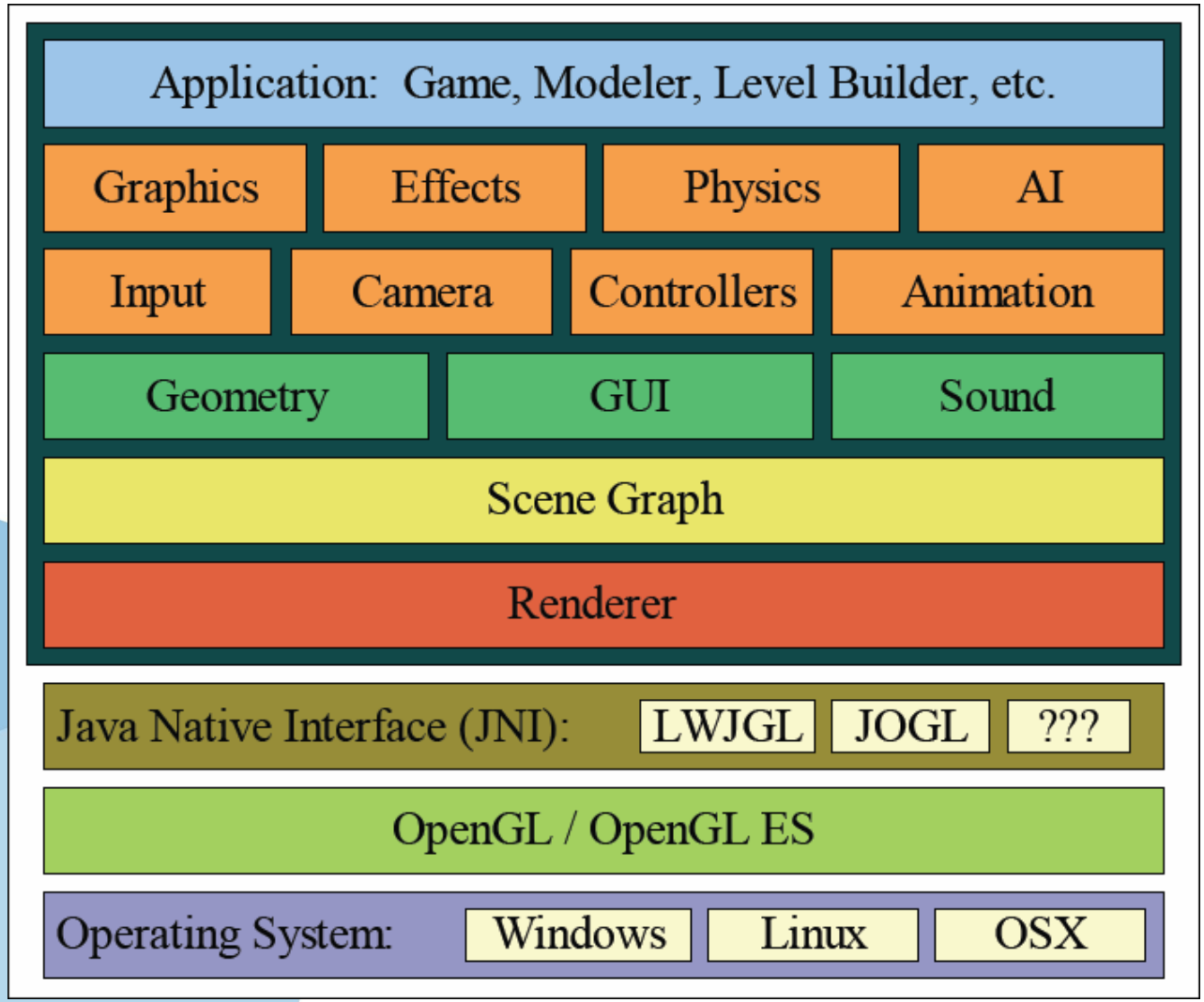
- *Open-source*.
- Escrita totalmente em Java delegando tarefas de renderização a biblioteca nativa OpenGL.
- Baseada em grafo de cena.

JMonkey Engine

- JNI acopla a biblioteca nativa de renderização a engine JME
- Física usando o *framework* JME Physics.

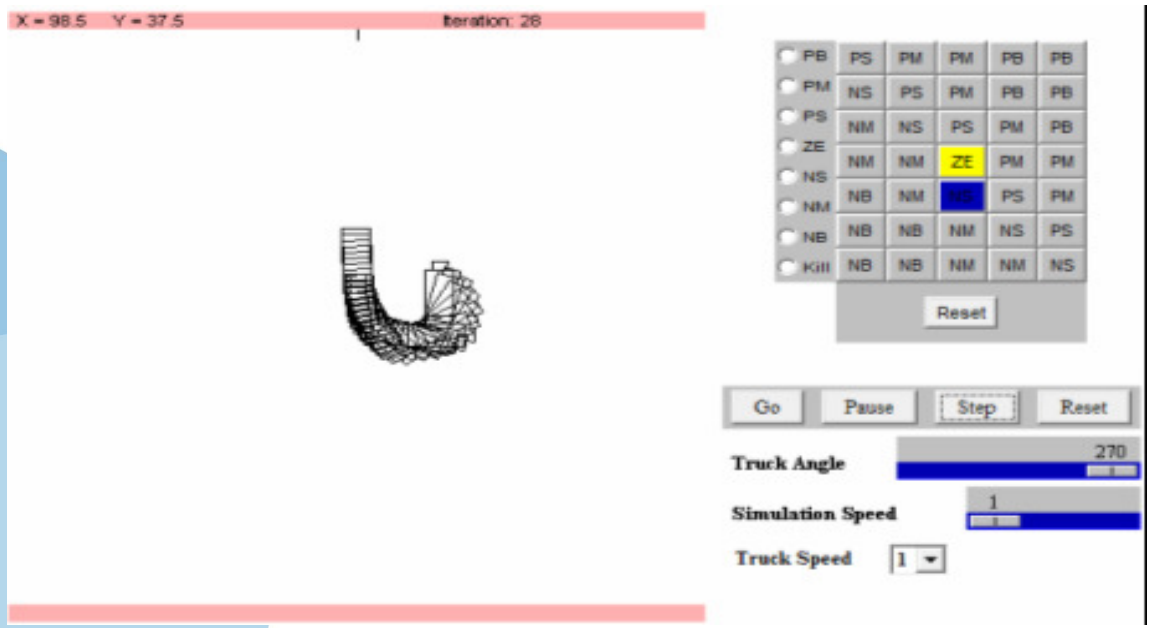


JMonkey Engine



Trabalhos Correlatos

- Lima: sistema *fuzzy* que movimenta um robô através de obstáculos.
- Tan e Seng: Protótipo de um robô que estaciona em vagas paralelas.
- Britton: Algoritmo para estacionar de ré um caminhão articulado.



Desenvolvimento do Simulador

Principais Requisitos

- Funcionais:
  - Estacionar um veículo autonomamente em espaços paralelos ao veículo.
  - Procurar em linha reta um espaço adequado para estacionar.
  - Parar o veículo após o mesmo estar estacionado.
  - Continuar procurando uma vaga, caso ache uma não adequada.
  - Estacionar o carro em vagas no lado direito, excluindo cenários com mão inglesa.

Principais Requisitos

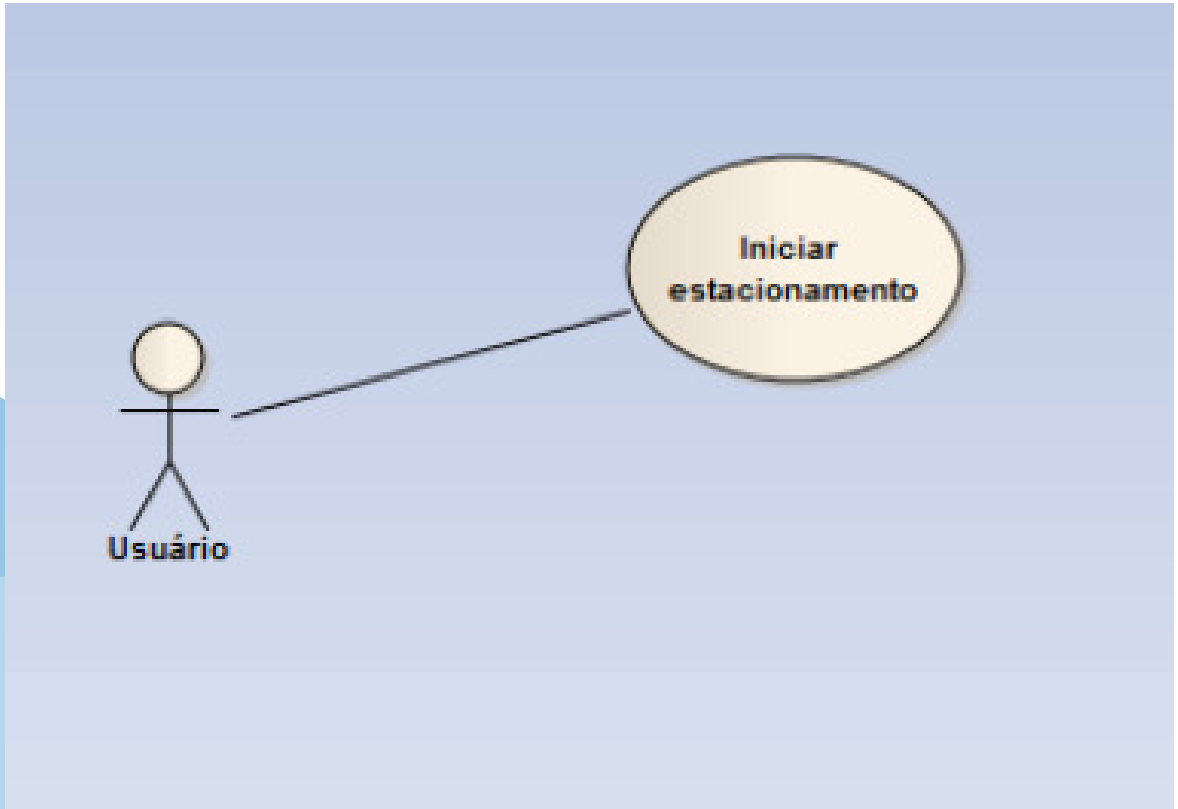
- Não funcionais:
  - Ambiente de desenvolvimento Eclipse Europa.
  - Linguagem de programação Java.
  - JMonkey Engine.

Especificação

- Técnicas e ferramentas utilizadas:
  - Ferramenta Enterprise Architect.
  - Linguagem especificação UML.

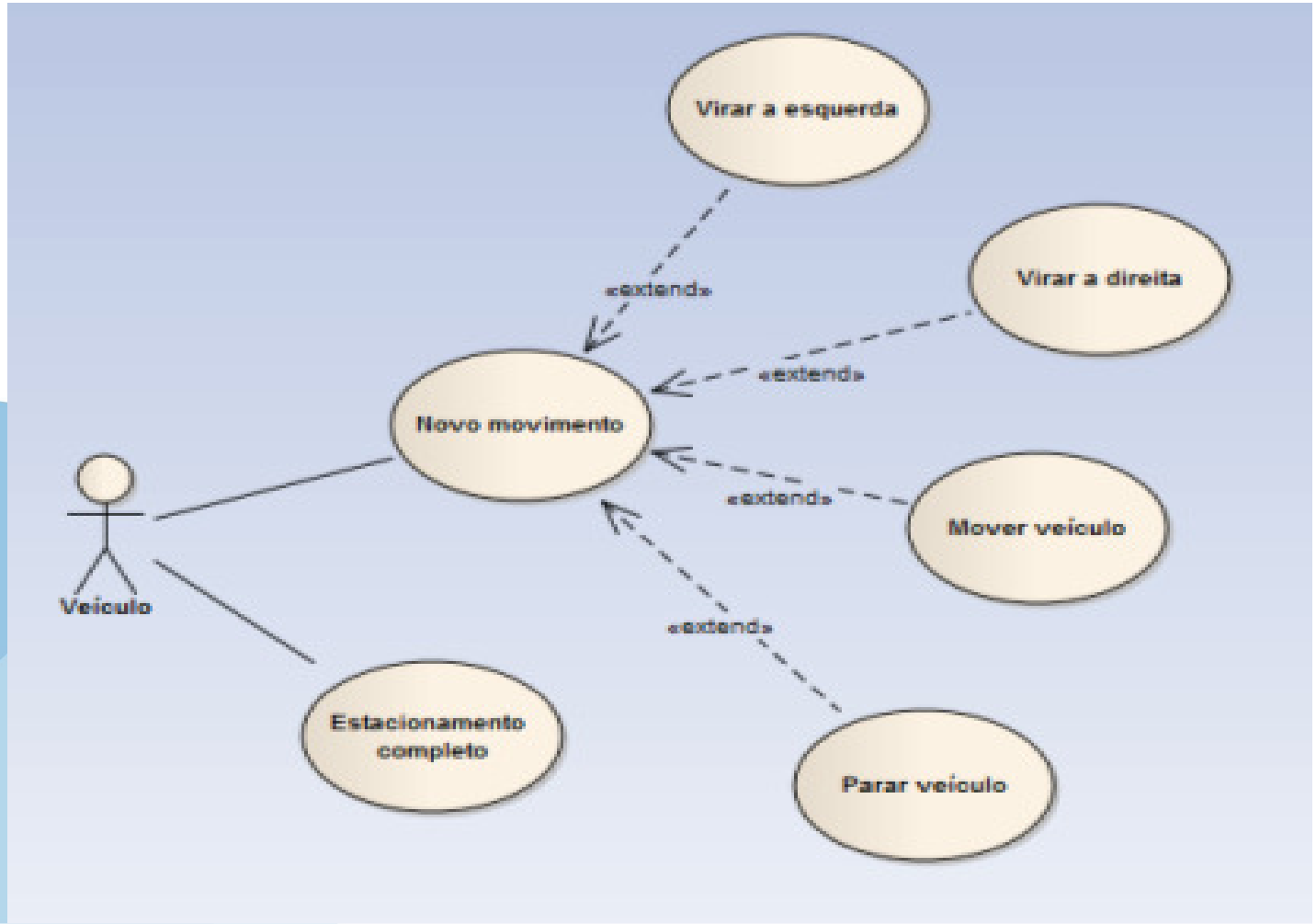
Especificação

- Diagrama de caso de uso tendo como ator o usuário:



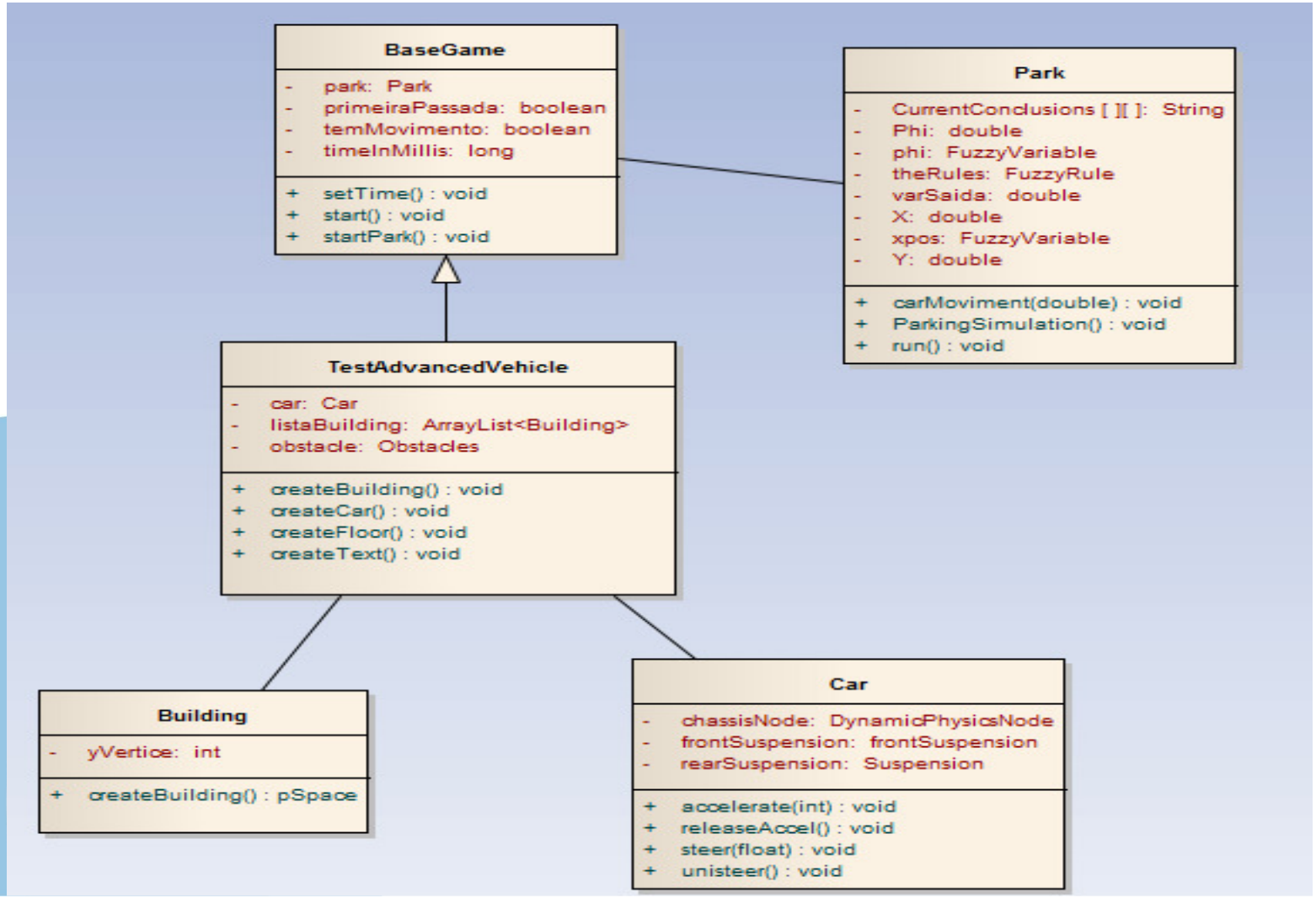
Especificação

- Diagrama de caso de uso tendo como ator o veículo:



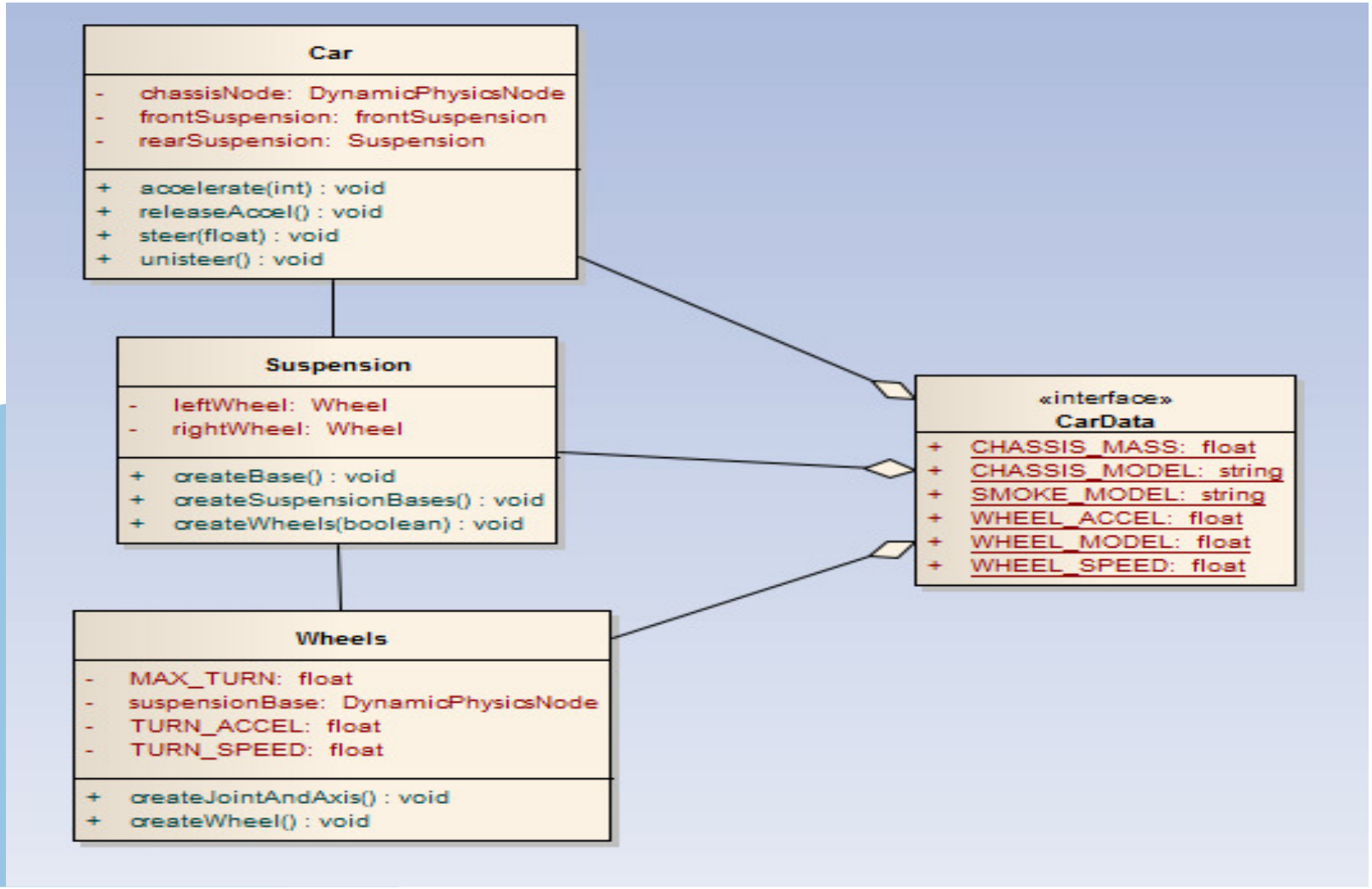
Especificação

- Diagrama de classes:



Especificação

- Diagrama de responsável pela criação do veículo:

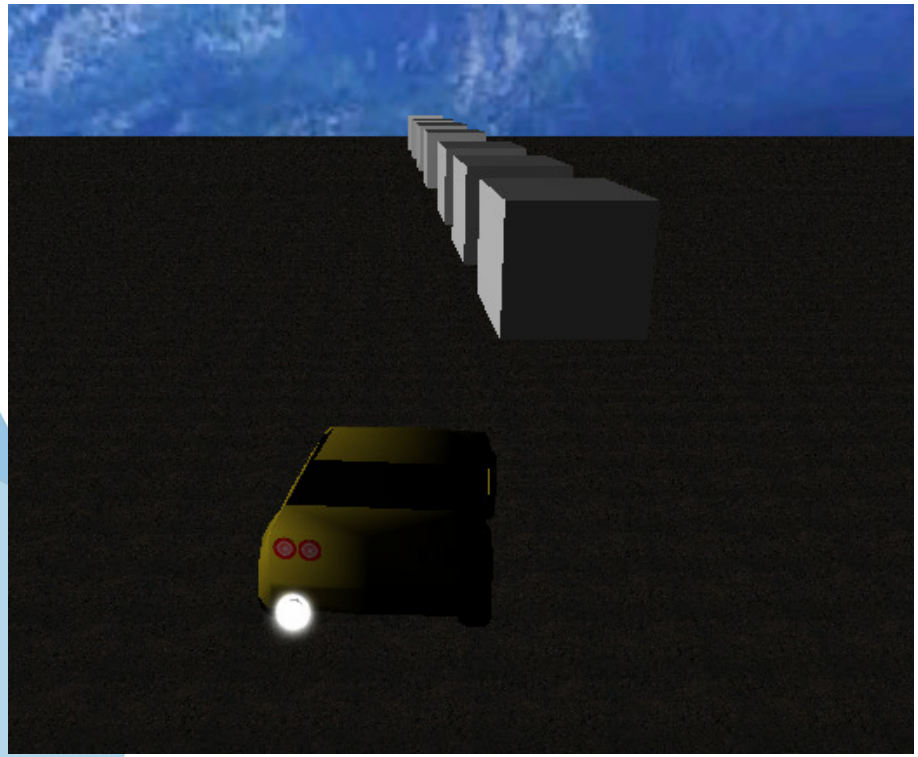


Implementação

- Técnicas e ferramentas utilizadas:
  - IDE Eclipse Galileo - Java.
  - Blender 2.43 - Modelo 3D do carro.
  - Adobe Photoshop CS - Texturas do terreno.

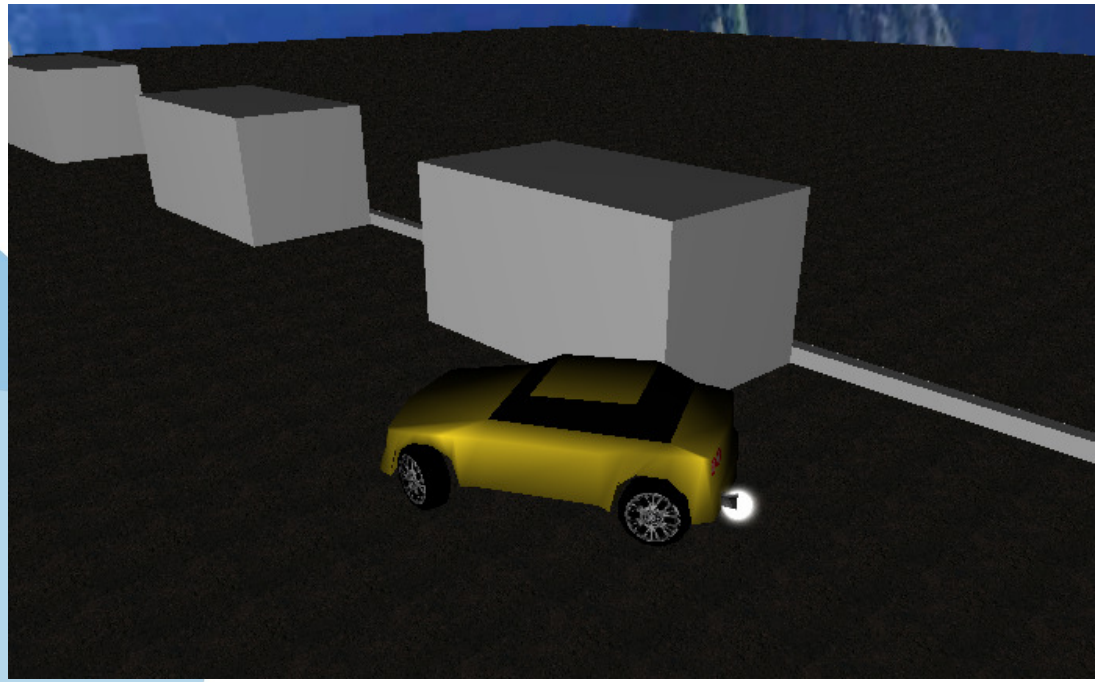
Implementação

- Operacionalidade da implementação / Passos:
  - Pressionar a tecla **K** para iniciar uma procura por vaga.



Implementação

- Operacionalidade da implementação / Passos:
  - Pressionar a tela **M** para iniciar o processo de estacionamento.



Resultados e Discussões

- Engine JME:

  Grafo de cena = performance, otimização e abstração.

- Cenário 3D com bom nível de imersão:

  Massa, colisão e suspensões.

- Lógica difusa:

  Tratamento de informações imprecisas.

Conclusão

Conclusão

- Resultados obtidos satisfatórios.
- Lógica difusa:
  Estudos e pesquisa.
- JME:
  Abstração e performance.
- FuzzyJ ToolKit:
  Pouca documentação e exemplos.

Extensões

- Novos modelos de terrenos e veículos.
- Expansão das regras difusas para estacionar também do lado esquerdo.
- Mecanismo que estacione em vagas de 45 e 90 graus.
- Embarcar a solução desenvolvida em um protótipo real.
- Adaptar o simulador para veículos articulados, como caminhões.

Obrigado pela atenção

Contato:
ewerton@inf.furb.br
ewertonrmachado@gmail.com