

UNIVERSIDADE CÂNDIDO MENDES

JÔNATAS OLIVEIRA LOPES SOARES  
MAYCON BARRETO LOPES  
WALLACE GOMES DE SOUZA

# MÓDULO DE MAPEAMENTO DO TOOLKIT HÓRUS

Campos dos Goytacazes - RJ

Junho - 2009

JÔNATAS OLIVEIRA LOPES SOARES  
MAYCON BARRETO LOPES  
WALLACE GOMES DE SOUZA

# MÓDULO DE MAPEAMENTO DO TOOLKIT HÓRUS

Monografia apresentada à Universidade  
Cândido Mendes como requisito obrigatório  
para a obtenção do grau de Bacharel em  
Ciências da Computação.

ORIENTADOR: Prof. D.Sc. Ítalo Matias

CO-ORIENTADOR: Prof. D.Sc. Dalessandro Soares

Campos dos Goytacazes-RJ

2009

JÔNATAS OLIVEIRA LOPES SOARES  
MAYCON BARRETO LOPES  
WALLACE GOMES DE SOUZA

## MÓDULO DE MAPEAMENTO DO TOOLKIT HÓRUS

Monografia apresentada à Universidade  
Cândido Mendes como requisito obrigatório  
para a obtenção do grau de Bacharel em  
Ciências da Computação.

Aprovada em \_\_\_\_ de \_\_\_\_\_ de 2009.

### BANCA EXAMINADORA

---

Prof. D.Sc. Ítalo Matias - Orientador  
Doutor pela UFRJ

---

Prof. D.Sc. Dalessandro Soares  
Doutor pela PUC-Rio

---

Prof. BLABLABLA  
Univeridade de Londres

*Dedico este trabalho a minha mãe*

*Jônatas*

*Dedico este trabalho a meus pais*

*Maycon*

*Dedico este trabalho a meus pais*

*Wallace*

# Agradecimentos

Agradecemos a Deus, pois sem Ele nada do que se fez poderia ter sido feito; a Ele que nos deu forças pra superar as dificuldades e vencer as barreiras.

Agradecimentos aos pais

Agradecimentos ao orientador e co-orientador.

Agradecemos aos demais integrantes da Banca Examinadora, os quais, pelo menos em algum momento, desde a origem até a conclusão do trabalho, deram a sua contribuição.

Agradecemos a todos os professores da Universidade Cândido Mendes do curso de ciências da computação, que nos acompanharam e nos ensinaram nessa fase única e marcante de nossas vidas, que foi nossa formação acadêmica.

Eu, Jônatas, agradeço em especial

Eu, Maycon, agradeço em especial

Eu, Wallace, agradeço em especial

Agradecemos também à Chrystiano, Leandro, Lucas e Thiago que participaram diretamente da nossa formação e, juntamente conosco, proporcionaram a conclusão deste trabalho.

E os nossos sinceros agradecimentos a todas as pessoas que, direta ou indiretamente contribuíram para que este trabalho fosse concluído.

Teste

# Resumo

Palavras-chave:

# Abstract

Keywords:



# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
<b>2</b>	<b>Inteligência Artificial</b>	<b>13</b>
2.1	Agentes Inteligentes . . . . .	13
2.2	Teste de Turing . . . . .	14
2.3	Cognição . . . . .	14
<b>3</b>	<b>Ambiente</b>	<b>15</b>
3.1	Especificando um ambiente . . . . .	15
3.2	Propriedades de ambientes de tarefas . . . . .	16
3.2.1	Completamente observável e parcialmente observável . . . . .	16
3.2.2	Determinístico e Estocástico . . . . .	16
3.2.3	Espisódico e Sequencial . . . . .	17
3.2.4	Estático e Dinâmico . . . . .	17
3.2.5	Discreto e Contínuo . . . . .	17
3.2.6	Agente único e Multiagente . . . . .	17
<b>4</b>	<b>Robótica</b>	<b>19</b>
4.1	Sensores . . . . .	19
4.2	Efetuadores . . . . .	20
4.3	Categorias da robótica . . . . .	20
<b>5</b>	<b>Arquiteturas em Robótica</b>	<b>21</b>
<b>6</b>	<b>O Toolkit Horus</b>	<b>22</b>
6.1	Objetivo . . . . .	22
6.2	Arquitetura . . . . .	22
6.3	Módulos do Horus . . . . .	22
6.3.1	Core do Horus . . . . .	22
6.3.2	Modulo de Visão . . . . .	22
6.3.3	Modulo de Mapeamento . . . . .	22

---

<b>7</b>	<b>Mapeamento</b>	<b>23</b>
7.1	SLAM . . . . .	24
7.1.1	Landmark Straction . . . . .	24
7.1.2	Data Association . . . . .	26
7.1.3	State Estimation . . . . .	26
7.1.4	State Update . . . . .	27
7.1.5	Landmark Update . . . . .	27
<b>8</b>	<b>Aplicação do ambiente virtual com um agente autônomo</b>	<b>28</b>
8.1	Simulador . . . . .	28
8.2	Ambiente . . . . .	29
8.3	O mapeamento . . . . .	31
<b>9</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>32</b>
	<b>Apêndices</b>	<b>33</b>
<b>A</b>	<b>Dependências do Tool kit</b>	<b>33</b>
<b>B</b>	<b>Instalações</b>	<b>34</b>

# Lista de Figuras

8.1	O agente utilizado nos testes. Modelado em Blende3D. . . . .	30
8.2	O ambiente utilizado para a prova de conceito. Modelada em Blende3D. .	30
8.3	Os lasers projetados a partir do agente. Gerenciados pelo Panda3D. . . . .	31

# Lista de Tabelas

7.1	Algoritmo RANSAC . . . . .	25
-----	----------------------------	----

# Capítulo 1

## Introdução

A robótica era um sonho até pouco tempo atrás, hoje em dia a sua existência é tão comum que muitas vezes nos passam despercebidos grandes avanços da área. Cada vez mais robos, ou melhor, agentes estão presentes em nosso dia-a-dia a fim de facilitar nossos afazeres a fim de nos tornar mais produtivos. A robótica móvel é um campo da robótica que estuda as vantagens da mobilidade dos agentes. Essa mobilidade os tornam capazes de avançar ainda mais em ambientes de difícil acesso ao ser humano.

# Capítulo 2

## Inteligência Artificial

É uma área de estudo da ciência da computação que se preocupa em fazer com que dispositivos computacionais tenham ações e reações similares as capacidades humanas, tais como: pensar, criar, solucionar problemas entre outros.

Tal inteligência pode ser aplicada de várias formas para tornar um agente, um agente inteligente, capaz de ter capacidades próprias.

### 2.1 Agentes Inteligentes

Um Agente, por definição, é todo elemento ou entidade autônoma que pode perceber seu ambiente por algum meio cognitivo ou sensorial e de agir sobre esse ambiente por intermédio de atuadores.

Algumas definições do termo agente na língua portuguesa tais como "O que opera ou é capaz de operar", "O que promove negócios alheios" e "Autor". Existem definições de agentes em várias áreas do conhecimento humano.

- Sociologia: Dentro dos estudos sociológicos, a definição de agentes inteligentes está relacionada aos seres humanos.
- Economia: Os agentes inteligentes são aqueles que operam de forma mais astuta dentro de um ambiente econômico.

- Robótica: Na robótica, que é nosso foco, o agente inteligente é visto como um agente que possui uma inteligência (artificial) e se utiliza dela para ter autonomia, proatividade e até mesmo tomada de decisões.

## 2.2 Teste de Turing

O Teste de Turing é utilizado para analisar se um programa, no nosso caso um agente artificial, é capaz de interagir com um ser humano como se fosse um outro ser humano.

O teste inicialmente foi feito da seguinte forma, dois humanos e um agente artificial eram colocados isolados e por meio de uma comunicação (um computador) um interrogador, que estaria se comunicando com os três simultaneamente, deverá conseguir identificar quem é o agente. Caso o interrogador não consiga identificar ele através desse "diálogo" o agente passaria no teste, caso contrário falharia.

Mesmo sendo uma representação simples, muitas empresas ainda utilizam o Teste de Turing (de forma mais adequada aos propósitos delas) para testar alguns softwares.

## 2.3 Cognição

A capacidade de processar informações que qualquer sistema tem, isso é o que define a **cognição**. Seja através de percepções, pensamentos, memória ou mesmo o raciocínio sobre algo. A palavra foi definida pela primeira vez na época Platônica (pelo próprio Platão) e posteriormente pelo seu discípulo Aristóteles.

# Capítulo 3

## Ambiente

De uma forma geral, um ambiente é um conjunto das condições, situações e/ou condições onde existe determinado objeto ou ocorre uma determinada ação. No caso desse projeto o ambiente significa o local onde está inserido um agente deve explorar, mapear e navegar.

Antes de imaginar um agente inteligente, deve-se pensar em um ambiente de tarefas, que será utilizado para ser o "problema" em si que os agentes devem utilizar para gerar suas "soluções".

### 3.1 Especificando um ambiente

O ambiente de tarefa, onde o agente irá realizar suas tarefas, tem que ser o mais completo possível. Lembrado que o agente deve ser inserido em um ambiente que seja apropriado ao seu propósito. Alguns fatores serão levados em consideração ao projetar um agente:

1. Medida de desempenho: Qual será o objetivo do projeto? minimizar custos, minimizar consumo de energia, otimizar rotas, melhorar mapeamento etc.
2. Ambiente: Que tipo de ambiente? Estático, determinístico, previsível etc.
3. Agente: Que tipo de agente? Humanóide, próprio para ambiente aquático, voador etc.



4. Sensores: Que tipo de sensores devem ser inseridos no agente? Lasers, sonares, câmeras etc.

Esses fatores serão de enorme importância ao se determinar um ambiente.

## 3.2 Propriedades de ambientes de tarefas

A quantidade de ambientes de tarefas que podem ser criados é enorme, no entanto, podemos subdividir em categorias. Essas categorias determinam um projeto apropriado para o agente e as principais técnicas que devem ser implementadas no agente. As categorias são:

### 3.2.1 Completamente observável e parcialmente observável

Se os sensores de um agente permitem acesso ao estado completo do ambiente em cada instante, diz-se que o ambiente é completamente observável. Um ambiente tarefa é completamente observável se os sensores detectam todos os aspectos que são relevantes para escolha de uma ação, relevância tal que depende da medida de desempenho. Um ambiente é parcialmente observável quando devido a ruído, sensores imprecisos, sensores mal configurados ou parte do ambiente estão ausentes nos dados do sensor.

### 3.2.2 Determinístico e Estocástico

Se o próximo estado do ambiente é completamente determinado pelo estado atual e pela ação executada pelo ambiente, dizemos que o ambiente é determinístico; caso contrário, ele é estocástico. A princípio, um agente não precisa se preocupar com a incerteza em um ambiente completamente observável e determinístico. Porém, se o ambiente for parcialmente observável, ele poderá parecer estocástico. Isso é verdadeiro se o ambiente é complexo, tornando-se difícil de controlar todos os aspectos não-observados.

### 3.2.3 Episódico e Sequencial

Em um ambiente de tarefa episódico, a experiência do agente é dividida em episódios atômicos. Cada episódio consiste na percepção do agente, e depois na execução de uma única ação. É crucial que o episódio seguinte não dependa das ações anteriores, executadas em episódios anteriores. Por outro lado, em ambientes sequenciais, a decisão atual poderá, na maioria das vezes certamente irá, afetar todas as decisões futuras, como por exemplo o jogo de xadrez onde cada mudança afeta todas as posteriores.

### 3.2.4 Estático e Dinâmico

Caso o ambiente mude enquanto o agente está se movimentando (seja para mapear, navegar etc), dizemos que o ambiente é dinâmico para esse agente; caso contrário, ele é estático. Ambientes estáticos são muito mais fáceis de manipular e observar, por que o agente não precisa continuar a observar o mundo enquanto está decidindo sobre a realização da ação, nem precisa se preocupar com o tempo, mas o nível de desempenho do agente se alterar, podemos dizer que o ambiente é semi-dinâmico.

### 3.2.5 Discreto e Contínuo

A distinção entre discreto e contínuo pode se aplicar ao estado do ambiente, do modo como o tempo é tratado, e ainda às percepções e ações do ambiente. Por exemplo, um ambiente de estados discretos como um jogo de xadrez tem um número finito de estados distintos. A entrada proveniente de câmeras digitais é discreto, mas em geral é tratada como representação de intensidades e posições que variam continuamente, logo, é contínuo.

### 3.2.6 Agente único e Multiagente

A distinção entre ambiente do agente único e o ambiente dos multiagentes pode parecer bastante simples. Por exemplo, um agente que resolve um jogo de palavras cruzadas sozinho está claramente em um ambiente de agente único, enquanto um agente que joga

---

Poker está claramente em um ambiente multiagente.

# Capítulo 4

## Robótica

A ciência robótica é reponsável pela parte da tecnologia que tem por intuito otimizar tarefas feitas por humanos e, em alguns casos, substituí-los por motivos que vão desde a preservação da integridade do ser humano até mesmo a ocupação de seu cargo de trabalho. Alheios a um mundo de filmes e preconceitos, os robôs tornam os resultados dos serviços melhores e sua precisão é muito maior que a de um funcionário humano.

Os robôs são agentes que executam tarefas no mundo físico. Para isso utilizam dispositivos para realizar suas ações: os sensores e os efetadores. A grande maioria dos robôs podem ser incluídos em três categorias principais:

1. Robôs manipuladores
2. Robôs móveis
3. Robôs híbridos

### 4.1 Sensores

A robótica se utiliza de vários dispositivos para emular os sentidos e as reações humanas em determinadas situações. Esses dispositivos tratam diferentemente cada um dos sentidos humanos.

## 4.2 Efetuadores

## 4.3 Categorias da robótica

Os robôs manipuladores são aqueles que tem sua base fixa de forma a não se movimentar além do alcance de suas partes móveis, assim como um braço. Esse tipo de robô é muito utilizado nas linhas de produção em indústrias automotivas, siderúrgicas, metalúrgicas etc. Esses robôs tem normalmente, normalmente, efetuadores que alteram a forma do objeto afetado, como soldas, cortadores, efetuadores de pressão, entre outros. Também existem manipuladores em tarefas mais precisas como em hospitais no auxílio em cirurgias delicadas.

Se os robôs manipuladores tem a característica de ficarem estáticos, os móveis são o oposto, eles podem se movimentar através de efetuadores de movimento. Robôs móveis são muito utilizados para alcançar locais que são insalubres aos seres humanos, tais como fundo do mar, superfície de vulcões, dutos de ar condicionado e inclusive superfícies de outros planetas, assim como foi feito em Marte com o Sojourner, nome do robô dado pela NASA, que peregrinou pelo solo do planeta vermelho.

Os robôs híbridos são equipados como os manipuladores e também são móveis, normalmente são vistos na sua forma humanoíde, ou seja, com forma humana. Muitos cientistas tentam ensinar esses robôs a efetuar tarefas domésticas, em alguns casos até mesmo interagir com humanos.

## Capítulo 5

# Arquiteturas em Robótica

# Capítulo 6

## O Toolkit Horus

### 6.1 Objetivo

### 6.2 Arquitetura

### 6.3 Módulos do Horus

#### 6.3.1 Core do Horus

#### 6.3.2 Modulo de Visão

#### 6.3.3 Modulo de Mapeamento

# Capítulo 7

## Mapeamento

O mapeamento é uma funcionalidade que é tratada de muitas formas dentro da literatura. O uso de algoritmos de mapeamento permite que um agente móvel possa identificar sua posição em um ambiente desconhecido e identificar o local em que está inserido. Com o ambiente devidamente mapeado é possível otimizar a rota uma vez que o agente já o conhece.

Algumas técnicas de mapeamento que foram estudadas:

- Técnica utilizando o algoritmo Dijkstra e Subida de Montanha.
- Método incremental convencional.
- Técnica baseada em grafos de visibilidade.
- SLAM (Simultaneous Localization and Mapping).

O método de mapeamento que será incluído no Toolkit Hórus será o SLAM (Simultaneous Localization and Mapping), tendo em vista que ele soluciona dois problemas clássicos da teoria das posições, que define a dificuldade de se localizar em um ambiente desconhecido e a dificuldade de mapear um ambiente onde não se sabe onde está.



## 7.1 SLAM

O Simultaneous Localization and Mapping é uma técnica utilizada em agentes autônomos para o mapeamento de ambientes desconhecidos levando em consideração a sua posição atual como a posição inicial para início do mapeamento. Os sensores que podem ser utilizados para a implementação do são diversos. Para a prova de conceito foi utilizado o odômetro, dispositivo que mensura distâncias percorridas, e o laser, dispositivo para detectar a presença de objetos no ambiente.

O SLAM é composto por vários segmentos que são independentes e tem suas comunicações muito bem estabelecidas o que os torna mais flexíveis quanto aos algoritmos utilizados em cada um dos segmentos. Cada um dos segmentos tem uma enorme gama de algoritmos que o compõe. Foram incorporadas ao Tool Kit apenas as mais otimizadas e relevantes para melhor utilização no processo.

Esses segmentos são:

1. Landmark Extraction: Segmento responsável pela extração de marcos no ambiente.
2. Data Association: Segmento que associa os dados extraídos de um mesmo marco por diferentes leituras do laser.
3. State Estimation: Segmento responsável por estimar a posição atual do robô com base em seu odômetro e nas extrações de marcos no ambiente.
4. State Update: Segmento que atualiza o estado atual do agente.
5. Landmark Update: Segmento que atualiza as posições dos marcos no ambiente em relação ao agente.

### 7.1.1 Landmark Straction

A forma de gestão dos marcos (objetos) e dos pontos de movimentação (áreas de movimentação do agente) foi feita através de um grafo. A escolha dessa estrutura foi baseada na sua credibilidade e largo uso na literatura.

Existem dois algoritmos que foram analisados para ser incorporados nesse segmento: o RANSAC e o SPIKE.

#### 7.1.1.1 RANSAC

O RANSAC (Random Sampling Consensus) algoritmo que utiliza-se de uma grande quantidade de informações para estimar os dados relevantes de um modelo matemático. Em conjunto com o SLAM o RANSAC identifica linhas de acordo com os pontos passados pelo laser, através da identificação de pontos muito próximos uns dos outros pode-se concluir que ali existe linhas, ou nesse caso, paredes que impossibilitam a transposição do agente.

**while**

- Houverem leituras de laser não associadas.
- **E** o número de leituras for maior que o limiar;
- **E** o número de iterações não for maior que o limite.

**do**

- Selecionar uma leitura de laser na lista.
- Seleciona uma quantidade S de exemplos de leitura do laser que estão associadas a uma quantidade D de graus daquele laser.
- Usando esses exemplos S e a leitura original para calcular o menor quadrado que se ajuste a linha.
- Determinar quantas leituras do laser estão dentro de X unidades de medida que melhor se ajustam a linha.

**if** O número de leituras do laser que ficam sobre a linha é maior que o Consensus **then**

- Calcular o novo mínimo quadrado que melhor se ajuste a linha, com base em todas as leituras e a linha formada anteriormente.
- Adicionar o melhor ajuste baseado no calculo anterior.
- Remover o número de leituras que cruzam a linha do total de leituras não associadas.

**end if**

**end while**

Tabela 7.1: Algoritmo RANSAC

Através de vários marcos o RANSAC obtêm uma amostra que será analisada a fim

de encontrar pontos próximos, consequentemente uma parede, baseado em um limiar de proximidade. Esse limiar é chamado de Consensus.

#### 7.1.1.2 SPIKE

O algoritmo de extração de marcos SPIKE faz a extração através da análise de um determinado montante de valores do laser, estimada uma diferença muito grande de um marco para outro, por exemplo, 0.6 metros, defini-se um SPIKE no marco que diferiu dos demais, isso serve para identificar grandes mudanças no ambiente, como por exemplo um laser lançado através das pernas de uma mesa que geraria grandes diferenças entre os feixes de laser que tocarem a(s) perna(s) e os laser que passarem por entre ela(s). Nesse momento o marco que foi detectado na perna da cadeira se torna um SPIKE.

O algoritmo leva em consideração um ambiente onde existem muitas diferenças entre dois marcos, em ambientes onde não existem tantas diferenças, esse algoritmo não tem eficiência.

### 7.1.2 Data Association

O segmento Data Association (tradução livre: Associação de Dados) é responsável pela filtragem e associação dos dados obtidos através dos dispositivos do agente.

Uma vez que um marco seja visualizado em um passo do agente e esse mesmo marco é visto novamente em um novo passo, a sua posição mudou, em relação ao agente. O Data Association faz a análise da posição atual do marco com a sua posição imediatamente anterior, com esse paralelo certifica-se que o marco existe ou se ele foi removido da cena.

Este segmento tem como saída para o State Estimation um lista com os marcos da cena.

### 7.1.3 State Estimation

O segmento State Estimation (tradução livre: Estimação do Estado) tem com objetivo analisar as informações passadas pelo Data Association e estimar as posições dos marcos

e do agente, com essas informações ele prepara as posições de cada elemento na cena e analisa o estado anterior já gravado.

#### **7.1.4 State Update**

O State Update (tradução livre: Atualização do Estado) faz a gravação do estado atual do agente, sua posição, em relação a posição inicial, valores relativos ao odômetro.

#### **7.1.5 Landmark Update**

Com o mesmo objetivo e com focos diferentes, o State Update (tradução livre: Atualização de marcos) faz a gravação do estado atual dos marcos, sua posição, em relação a posição do agente.

## Capítulo 8

# Aplicação do ambiente virtual com um agente autônomo

A fim de produzir uma prova de conceito, foi implementada uma aplicação onde foi possível demonstrar a eficiência do algoritmo de mapeamento. Foi produzido um simulador com gravidade, colisão, rederização de texturas, objetos e atores (o agente nesse caso) e, tendo esse simulador como base, foi criada a aplicação que tem um agente utilizando-se do Toolkit Hórus para mapeamento e navegação dentro do ambiente simulado.

### 8.1 Simulador

Um simulador é todo software (em nosso caso) que simula um comportamento de algum sistema. Deve ser capaz de reproduzir, de forma mais fiel possível, a realidade na qual ele tenta emular. No nosso caso foi desenvolvido um simulador de um ambiente, a fim de virtualizar um ambiente real.

O simulador desenvolvido foi tratado como uma aplicação a parte, pois não estava incluído no escopo do projeto, porém tendo em vista a necessidade de um simulador customizado para a realidade de um agente móvel (e na linguagem selecionada) o simulador foi incluído ao projeto.

Alguns requisitos foram necessários para que o simulador fosse projetado. Alguns softwares que foram utilizados: Blender e Panda3d. Como uma das linguagens utilizadas foi Python, buscou-se ferramentas que fossem compatíveis (na verdade ambas são em Python) com as linguagens utilizadas.

Python é um linguagem orientada objetos de tipagem dinâmica, onde os tipos dos dados são atribuídos durante a interpretação do código dinamicamente e forte. O Python tem foco na produtividade com uma sintaxe legível e simples, bem como módulos auxiliares (grande maioria OpenSource) que garantem um excelente desempenho.

O Blender é um produto da Blender Foundation, que é open source e desenvolvido em Python para modelagem de objetos 3D que está disponível para vários sistemas operacionais sobre a licença GNU (General Public License).

O Panda3D é um produto da equipe de desenvolvimento da Walt Disney para renderização de jogos e ambientes virtuais em terceira dimensão. Está sobre licença da BSD License (com algumas modificações para adequação a realidade do projeto).

## 8.2 Ambiente

O ambiente em que o agente foi testado consiste em uma área que simula um galpão com 5 cômodos onde o agente inicia sua movimentação no cômodo 1 e pode alcançar qualquer ponto do ambiente a partir de sua posição inicial. O ambiente que foi desenvolvido tem como objetivo emular, em menor escala, um ambiente real.

O agente utilizado foi inspirado no personagem de um filme que também é um agente inteligente, é possível notar a semelhança na figura 8.1.

A figura 8.2, que demonstra o ambiente que foi utilizado, pode-se ter uma visão em perspectiva do ambiente modelado. Ele foi modelado em Blender, tal ferramenta proporcionou os recursos de modelagem UV, texturização, determinação de medidas precisas, entre outros. O Blender foi selecionado, também, por conta da sua legibilidade de fácil assimilação tendo em vista que o Toolkit foi desenvolvido em Python e a ferramenta utiliza a mesma linguagem para produzir os modelos.



Figura 8.1: O agente utilizado nos testes. Modelado em Blende3D.

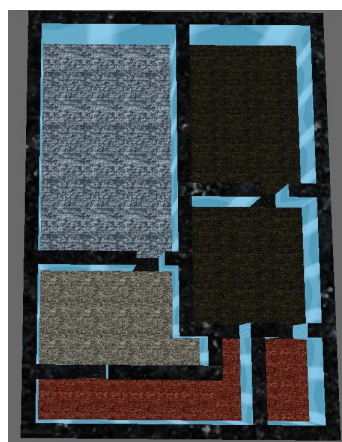


Figura 8.2: O ambiente utilizado para a prova de conceito. Modelada em Blende3D.

O Panda3d foi utilizado para fazer todos os tratamentos de colisão e virtualização de câmeras e renderização do agente no ambiente. Os lasers, áreas de colisão são tratados conforme a imagem 8.3 Como o Blender3D agilizou o processo o fato do Panda3D ser em Python.

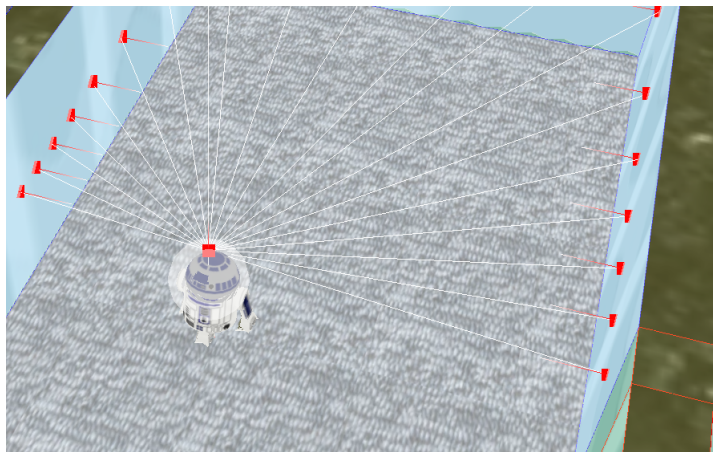


Figura 8.3: Os lasers projetados a partir do agente. Gerenciados pelo Panda3D.

### 8.3 O mapeamento

No que diz respeito ao mapeamento foram utilizadas algumas técnicas a fim de testar o desempenho do agente no ambiente proposto.



## Capítulo 9

### Conclusões e Trabalhos Futuros

## Apêndice A

### Dependências do Tool kit

# Apêndice B

## Instalações