**Tópicos em Engenharia de Software**

Destinatários:

Prof. Dr. Vicente Lucena Jr.

2

Corrida com obstáculos - Robotino

**Especificação de requisitos do sistema**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tipo: | Classe de Modelo de Processo: | | Estudantes: | | | Tutor: | | Início: | Fim: |
| Disciplina de Mestrado |  | | André Ricardo  Antônio Pereira  Jainne Fernandes  Luciana Rolim | | | Prof. Dr. Vicente Lucena Jr. | |  | 30.01.2017 |
| Documento: | | Versão: | | Autor: | | | Data: | Estado: | |
| Especificação de requisitos do sistema | |  | | Jainne / Luciana | | | 09.01.2017 | Rascunho | |
| Nome do Arquivo: | | Pages: | | | Print Date: | | | Template: | |
| x | | 7 | | | 13/01/2017 09:00 | | |  | |

**Gerenciamento de Versão do Documento**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Versão | Autor | QA | Data | Status | Atualizações |
| 0.1 | Jainne | André | 09.01.17 | Em progresso | Criação |
| 1.0 | Luciana |  | 12.01.17 | Submetido |  |
|  |  |  |  |  |  |

# Indíce

0 Indíce 2

1 Propósito Geral 3

1.1 Critérios Obrigatórios 3

2 Área Operacional 3

2.1 Condições de Operação 4

3 Ambiente 4

3.1 Software 4

3.2 Hardware 4

3.3 Interface do Sistema 5

4 Requisitos Funcionais 5

5 Requisitos Não Funcionais 6

6 Requisitos para a Interface de Usuário 6

7 Requisitos de Qualidade 6

8 Cenários de Testes Globais e Casos de Testes 7

9 Ambiente de Desenvolvimento 7

9.1 Orgware 7

10 Execução 7

# Propósito Geral

Este projeto tem como objetivo desenvolver um software que garanta o controle autônomo da plataforma de hardware Robotino para que este, partindo de uma posição de partida especificada, percorra um ambiente com obstáculos, desvie dos mesmos e chegue ao ponto de chegada pré-definido no menor tempo possível, ou seja, realize uma corrida com obstáculos.

O ambiente para a realizaçãodesta corrida, bem como os pontos de saída e chegada devem ser definidos à priori. A disposição dos obstáculos, no entanto, será especificada momentos antes do início da corrida, logo, o software desenvolvido terá que tomar decisões em tempo real e, de alguma forma assegurar que o caminho percorrido até atingir o objetivo final, o ponto de chegada, seja ótimo, para dessa forma garantir um menor tempo de finalização do percurso, se comparado ao dos outros componentes.

## Critérios Obrigatórios

* O sistema projetado deve ser autônomo.
* Uma vez iniciada a corrida, não poderá haver nenhum tipo de interferência externa.
* O ponto de partida corresponderá a uma área com dimensões pré-definidas situada antes de uma faixa especificada.
* O ponto de chegada corresponderá a uma área com dimensões pré-definidas situada após uma faixa especificada.
* A corrida só irá terminar quando o Robotino, partindo do ponto de saída, atingir o ponto de chegada.
* O Robotino não pode mover os obstáculos que surgirem ao longo de seu percurso mais que 5 cm aproximadamente.
* A contagem do tempo do percurso irá parar somente quando o Robotino atingir a posição de chegada.
* Os componentes da equipe devem garantir que a plataforama de hardware esteja pronta para ser utilizada antes da corrida, verificando questões importantes como bateria carregada, dentre outros.
* O padrão de um elemento que comporá osobstáculos é pré-definido (resma de papel ofício).
* A disposição dos obstáculos será definida momentos antes da corrida e os elementos padrão (resma de papel) poderão ser empilhados e/ou mesclados uns aos outros.
* O Robotino deverá ser capaz de executar as tarefas especificadas nos requisitos.
* O ambiente de desenvolvimento do Robotino deverá estar atualizado para a API2 (Application Programming Interface 2).

# Área Operacional

Este projeto foi proposto com o objetivo de validar os conhecimentos adquiridos na disciplina Tópicos em Engenharia de Software do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE/UFAM). Diversos grupos participarão de uma competição de corrida com o robô, conforme a especificação dos critérios obrigatórios, com o intuito principal de desenvolver a criatividade, agilidade, autenticidade e comprometimento dos mestrandos.

## Condições de Operação

O ambiente no qual o robô executará suas tarefas será um laboratório do CETELI com dimensões de aproximadamente 49 m². O ponto de partida corresponderá a uma área com dimensões de aproximadamente 80x80 cm² situada antes de uma faixa fixada em uma posição pré-definida na extremidade superior direita da sala. O ponto de chegada corresponderá a uma área com dimensões de aproximadamente 80x80 cm² situada após uma faixa fixada em uma posição pré-definida na extremidade inferior esquerda da sala.

O elemento padrão que comporá os obstáculos será uma resma de papel ofício A4 com dimensões 210x297 mm. Lembrando que, como definido nos critérios obrigatórios, a disposição dos obstáculos será definida momentos antes do início da corrida e os elementos padrões (resmas de papel) poderão ser empilhadas e/ou mescladas umas às outras.

# Ambiente

## Software

Os softwares utilizados para o desenvolvimento deste projeto foram:

* Simulador Robotino® SIM;
* IDE Eclipse;
* FileZilla;

O Simulador Robotino® SIM é um software para programação no sistema operacional Windows para simulação 3D do Robotino que permite a experimentação de um ambiente pré-definido virtual. Este modelo de simulação não pode ser estendido, inclui modelos geométricos e o Robotino dispõe dos recursos câmera, nove sensores de distância e o sensor digital na faixa de proteção do seu *chassi*.

O Eclipse é um ambiente de desenvolvimento integrado (*IDE - Integrated Development Environment*) para desenvolvimento Java, mas que suporta outras linguagens de programação a partir de *plugins*, como por exemplo, C/C++, PHP, Phyton, dentre outros. A linguagem de programação utilizada neste projeto é Java, por isso, optou-se por utilizar esta ferramenta.

FileZilla é uma plataforma gráfica FTP (*File Transfer Protocol*), SFTP (*SSH File Transfer Protocol*) e FTPS (*File Transfer Protocol Secure*) para administração de arquivos nos sistemas operacionais Windows, Linux, Mac OS X, dentre outros. Ela facilita a transferência de arquivos entre o cliente e o servidor Web.

## Hardware

A plataforma de hardware utilizada neste projeto foi o Robotino® da Festo Didatic, Edição Básica. O Robotino possui uma estrutura em aço inoxidável circular com acionamento omnidirecional, nove sensores infravermelhos de distância, um sensor indutivo, dois sensores ópticos, uma câmera colorida com resolução Full HD 1080p e interface USB. Adicionalmente, ele conta com um PC incorporado com processador Intel Atom de 1.8 GHz dual core, 4 GB de RAM, 32 GB de SSD. Além disso, ele possui duas portas Ethernet, seis portas USB 2.0, dois slots PCI Express, um VGA e uma interface I/O para integrar componentes elétricos adicionais, caso seja necessário.

## Interface do Sistema

A comunicação entre o computador e o software do Robotino é estabelecida através da rede WLAN (*Wireless Local Area Network*), via protocolo TCP/IP e a ferramenta escolhida para realizar a transferência de arquivos foi o FileZilla, via protocolo FTP. Uma vez que o cliente (PC) e o servidor (Robotino) estejam configurados adequadamente, o código desenvolvido pode ser enviado para o Robotino® que irá executar as tarefas especificadas no programa.

# Requisitos Funcionais

\ F10\ O Robotino deve ser capaz de se mover em qualquer direção e girar em torno do seu eixo quando necessário.

\ F15\ O Robotino deve ser capaz de trafegar pelo ambiente (sala) tanto na presença de obstáculos como na ausência dos mesmos.

\ F20\ O Robotino deve ser capaz de identificar a posição de partida.

\ F25\ O Robotino deve ser capaz de identificar a posição de chegada.

\ F30\ O Robotino deve ser capaz de, após identificar a posição de partida, sair deste ponto e iniciar seu percurso ao longo do ambiente (sala).

\ F35\ O Robotino deve ser capaz de detectar um ou mais obstáculos presentes ao longo de seu percurso.

\ F40\ O Robotino deve ser capaz de desviar dos obstáculos presentes ao longo de seu percurso.

\ F45\ O Robotino deve ser capaz de, ao longo do percurso, após desviar de obstáculos, compensando o desvio e tentando retornar para o caminho ótimo.

\ F50\ O Robotino deve ser capaz de, ao identificar a posição de chegada, parar.

\ F55\ O sistema deve ser capaz de calcular o tempo total do percurso, ou seja, o tempo gasto para alcançar o ponto de chegada a partir do ponto de partida.

\ F60\ O sistema deve ser capaz de monitorar o estado da bateria.

\ F65\ O sistema deve identificar o nível de bateria.

\ F70\ O sistema deve gerar uma mensagem caso o nível de bateria esteja baixo.

\ F75\ O sistema deve ser capaz de gerar mensagens de erro caso haja algum problema na realização de uma ou mais operações do Robotino.

\ F80\ O sistema deve ser capaz de reinicializar o software de controle de forma automática caso ocorra algum problema que comprometa o funcionamento adequado do Robotino.

# Requisitos Não Funcionais

\ N10\ O sistema desenvolvido deve garantir que o Robotino responda de forma autônoma e correta aos marcos (pontos de partida e chegada) definidos nos requisitos funcionais e aos estímulos (obstáculos) que vierem a surgir ao longo de seu percurso.

\ N20\ O sistema desenvolvido deve assegurar que o tempo do percurso total seja calculado de forma correta.

\ N30\ O sistema desenvolvido deve garantir que o Robotino se recupere e continue seu percurso caso haja falha na realização de alguma de suas operações definidas nos requisitos funcionais.

\ N40\ O sistema desenvolvido deve assegurar que o Robotino não apresente comportamentos que venham a danificar a plataforma.

# Requisitos para a Interface de Usuário

Nenhum.

# Requisitos de Qualidade

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Qualidade do Produto** | **Muito bom** | **Bom** | **Normal** | **Não relevante** |
| Funcionalidade |  | X |  |  |
| Correção |  | X |  |  |
| Segurança |  |  |  | X |
| Proteção |  |  | X |  |
| Confiabilidade |  | X |  |  |
| Maturidade |  |  | X |  |
| Tolerância à Falhas |  |  | X |  |
| Recuperabilidade |  | X |  |  |
| Usabilidade |  |  | X |  |
| Compreensibilidade |  | X |  |  |
| Capacidade de aprender |  | X |  |  |
| Operabilidade |  | X |  |  |
| Eficiência |  | X |  |  |
| Desempenho do Tempo |  | X |  |  |
| Comportamento de consumo |  |  | X |  |
| Alterabilidade |  | X |  |  |
| Capacidade de analisar |  | X |  |  |
| Modificabilidade |  | X |  |  |
| Portabilidade |  | X |  |  |

# Cenários de Testes Globais e Casos de Testes

Um cenário de teste global será especificado de forma que o sistema desenvolvido possa exibir as seguintes funcionalidades:

* Trafegar em um ambiente sem obstáculos;
* Detectar um obstáculo;
* Dectar o ponto de partida e sair do mesmo;
* Detectar o ponto de chegada e parar.

Todas essas funcionalidades devem ser executadas de forma autônoma.

# Ambiente de Desenvolvimento

Um computador com sistema operacional Microsoft Windows 2010 será usado para o desenvolvimento deste projeto com as ferramentas citadas no tópico 3.1 instaladas e devidamente configuradas.

## Orgware

As ferramentas utilizadas foram: Microsoft Word 2010, Microsoft Excel 2010, Microsoft Project 2010, Wunderlist, GitHub.

# Execução

A competição será realizada no dia 30 de Janeiro de 2017 no laboratório definido pelo tutor da disciplina. Os testes que precisarem ser feitos antes do dia da corrida ficam a cargo da equipe e da disponibilidade dos Robotinos.

Entregas parciais referentes à documentação do projeto deverão ser feitas nos prazos previstos no plano de projeto.