

PDS - Processo de Desenvolvimento de Software

The Bug is on the Table

Bruno Rodrigues
Iago Rodrigues
Jonathan Rufino
Matheus Andrade
Yago Regis

30-06-2015

List of Figures

1	Base para acoplamento dos veiculos.	16
2	Garra para derrubar o galho e capturar objetos.	17
3	Brao para acionamento do mecanismo das ondas.	18

List of Tables

1	Relao pontos por misso.	4
2	Relao de dificuldade das misses	5

Contents

1	Misses	4
1.1	Ordem de Pontuao das Misses	4
1.2	Ordem de Dificuldade das Misses	4
1.3	Definindo o Problema	4
2	Caminho de Suplementos, Ambulncia, Sinal de Evacuao e Isolamento de Construo	5
2.1	Objetivo da missao	5
2.2	Passos envolvidos	5
2.3	Pseudo cdigo	6
2.4	Cdigo NXC	6
3	Galho da rvore, Animais e Equipamentos	8
3.1	Objetivo da Misso	8
3.2	Passos envolvidos	8
3.3	Pseudo-cdigo	8
3.4	Cdigo NXC	8
4	Tsunami	10
4.1	Objetivo da isso	10
4.2	Passos envolvidos	10
4.3	Pseudo-cdigo	10
4.4	Cdigo NXC	10
5	Zona de Segurana	11
5.1	Objetivo da isso	11
5.2	Passos envolvidos	11
5.3	Pseudo-cdigo	11
5.4	Cdigo NXC	11
6	Musica - Game of Thrones	12
6.1	Objetivo da isso	12
6.2	Passos envolvidos	12
6.3	Pseudo-cdigo	12
6.4	Cdigo NXC	12
7	API	14
7.1	Objetivo da isso	14
7.2	Passos envolvidos	14
7.3	Pseudo-cdigo	14
7.4	Cdigo NXC	14
8	Garras	16

1 Misses

O documento do desafio Fria da Natureza, nos da diretrizes respeito de todas as misses que podem ser realizadas. Cada uma possui sua descrio, condies finais e a respectiva pontuao.

1.1 Ordem de Pontuao das Misses

As misses foram listadas em ordem decrescente de pontuao, de forma que a pontuao considerada seja a maior possvel de cada uma.

Pontuao	Misso
66	Famlia
31	Obstculos
30	Galho da rvore
30	Pista de Pouso
30	Sinal de Evacuao
30	Teste de Isolamento da Base
30	Avio de Carga
25	Ambulncia
25	Casa Elevada
25	Construo de Cdigo
25	Zona de Segurana
20	Camiho de Suplementos
20	Relocao de Construo
20	Tsunami
18	Segurana
15	gua
15	Animais
4	Suplementos e Equipamentos

Table 1: Relao pontos por misso.

1.2 Ordem de Dificuldade das Misses

Lista das misses ordenadas por dificuldade, mais fcil para a mais dificil, para que possa haver um cruzamento de informaes entre pontuao e dificuldade afim de que as misses sejam priorizadas *melhorar texto*...

1.3 Definindo o Problema

Algumas misses so completamente independentes uma das outras, entretanto, existem misses que devem seguir uma sequncia lgica para que no atrapalhem as demais. Cada misso deve possuir um planejamento antes de ser desenvolvida e executada.

Dificuldade	Misso
FCIL	Zona de Segurana
FCIL	Caminho de Suplementos
FCIL	Tsunami
MDIO	Famlia
MDIO	Galho da rvore
MDIO	Pista de Pouso
MDIO	Avio de Carga
MDIO	gua
MDIO	Animais
MDIO	Suplementos e Equipamentos
DIFCIL	Obstculos
DIFCIL	Teste de Isolamento da Base
DIFCIL	Casa Elevada
DIFCIL	Construo de Cdigo
DIFCIL	Relocao de Construo
DIFCIL	Segurana

Table 2: Relao de dificuldade das misses

2 Caminho de Suplementos, Ambulncia, Sinal de Evacuao e Isolamento de Construo

2.1 Objetivo da misso

O caminho de suplementos est tocando o tapete na regio amarela;
 A ambulncia est na rea amarela;
 Todas as rodas da ambulncia esto tocando o tapete;
 O sinal est obviamente em p (no precisa ser na vertical), mantido no lugar apenas pelo atrito da viga com o tapete;
 Nenhum parte do modelo de misso est sendo tocado pelo rob ou qualquer obstculo estratgico;

2.2 Passos envolvidos

Sair da base
 Pegar o caminho
 Pegar a ambulncia
 Sseguir at a rea amarela
 Empurrar o Sinal de Evacuao
 Destruir o prdio direto das torres
 Voltar para a base

2.3 Pseudo cdigo

Mova em frente 38cm
Pare de mover
Vire direita 90 graus
Mova em frente 20cm (para chegar ao caminho)
Pegue o caminho
Mova em frente 30 cm (para chegar ambulncia)
Vire esquerda 15 graus
Pegue a ambulncia
Mova em frente 90 cm (para chegar na rea azul)
Pare de mover
Mova para trs 35cm
Pare de mover
Vire direita 45 graus
Mova em frente 40cm (para empurrar a placa de sinalizao)
Pare de mover
Mova para trs 20cm
Pare de mover
Vire 175 graus
Mova em frente 90cm
Pare de mover
Vire esquerda 30 graus
Mova em frente 40cm (para chegar base)
Encerre o programa

2.4 Cdigo NXC

```

/*
 * Initial Position:
 * Right side of the "claw", aligned at the end of the "M" by inside.
 */
#include "theBugAPI.h"

task main() {

    // Moving to Supply Truck and Ambulance.
    move(38, POWER_NORMAL, FORWARD);
    turn(85, POWER_NORMAL, RIGHT);
    Wait(1);
    move(50, POWER_NORMAL, FORWARD);
    turn(9, POWER_LOW, LEFT);
    Wait(1);
    move(20, POWER_NORMAL, FORWARD);

    // Moving to Yellow Mark.
    move(100, POWER_NORMAL, FORWARD);
    turn(80, POWER_HIGH, LEFT);
    Wait(1);
    move(25, POWER_NORMAL, FORWARD);

    // Moving to Evacuation Signal.
    move(20, POWER_NORMAL, BACKWARD);
    turn(30, POWER_NORMAL, LEFT);
    Wait(1);
    move(40, POWER_NORMAL, BACKWARD);
    turn(35, POWER_NORMAL, RIGHT);
    Wait(1);
    move(21, POWER_NORMAL, FORWARD);

    // Moving to Isolamento de Construo.
    move(30, POWER_NORMAL, BACKWARD);
    turn(90, POWER_NORMAL, LEFT);
    Wait(1);
    move(30, POWER_NORMAL, FORWARD);
    move(14, POWER_HIGH, FORWARD);

    //Moving to base
    move(15, POWER_NORMAL, BACKWARD);
    turn(83, POWER_NORMAL, LEFT);
    Wait(1);
    move(150, POWER_HIGH, FORWARD);
}

```

3 Galho da rvore, Animais e Equipamentos

3.1 Objetivo da Misso

O galho leste da rvore est mais prximo do tapete do que os cabos eltricos esto; A rvore e o modelo de misso dos cabos eltricos esto para cima, retos, tocando o tapete;

Ao menos um animal est com pelo menos uma pessoa em uma regio colorida; Ao menos um elemento que no gua est numa regio colorida vermelha ou amarela (12 elementos possveis: walkie talkie, bateria, gerador, 2 combustveis, gro, po, remdio, rdio, lanterna, motocicleta e capacete).

3.2 Passos envolvidos

Sair da Base
Ir ao lado do galho da rvore
Derrubar o galho da rvore
Pegar os animais e equipamentos da regio
Retornar a Base

3.3 Pseudo-cdigo

Mova em frente 14 cm
Vire esquerda 87 graus
Mova em frente 67 cm
Pare de mover
Levante a garra 70 graus
Mova para trs 15 cm
Vire para esquerda 90 graus
Mova para frente 18 cm
Vire para direita 90 graus
Mova para frente 19 cm
Desa a garra 70 graus
Mova para trs 86 cm
Encerre o programa

3.4 Cdigo NXC

```

/*
 * Initial Position:
 * Right side of the "claw", aligned at the end of the "M" by left-inside.
 */
#include "theBugAPI.h"

task main() {

    // Moving to the tree.
    move(70, POWER_NORMAL, FORWARD);
    claw_control(10, POWER_HIGH, BACKWARD);
    claw_control(90, POWER_HIGH, BACKWARD);
    claw_control(100, POWER_HIGH, FORWARD);

    // Moving to catch the animals.
    move(30, POWER_NORMAL, BACKWARD);
    turn(85, POWER_NORMAL, RIGHT);
    Wait(1);
    move(22, POWER_NORMAL, BACKWARD);
    turn(90, POWER_NORMAL, LEFT);
    Wait(1);
    move(39, POWER_NORMAL, FORWARD);
    claw_control(70, POWER_NORMAL, BACKWARD);

    // Moving to base.
    move(70, POWER_HIGH, BACKWARD);
}

```

4 Tsunami

- 4.1 Objetivo da isso
- 4.2 Passos envolvidos
- 4.3 Pseudo-código
- 4.4 Código NXC

```
/*
 * Initial Position:
 * Aligned straight with the tsunami.
 */
#include "theBugAPI.h"

task main() {

    // Indicates the port to which the sensor is connected.
    SetSensorUltrasonic(IN_2);

    // The robot will move up to a certain distance.
    while(true) {
        // Delay to read sensor.
        Wait(1000);

        // print the value read by the sensor.
        ClearScreen();
        NumOut(0, 0, SensorUS(IN_2));

        if(SensorUS(IN_2) < 23) {
            Off(OUT_BC);
            break;
        }
        else {
            OnFwd(OUT_BC, POWER_LOW);
        }
    }

    Wait(50);
    claw_control(60, POWER_NORMAL, BACKWARD);

    // Moving to base.
    move(70, POWER_HIGH, BACKWARD);
}
```

5 Zona de Segurança

- 5.1 Objetivo da isso
- 5.2 Passos envolvidos
- 5.3 Pseudo-código
- 5.4 Código NXC

```
/*
 * Initial Position:
 * Right side of the "claw", aligned at the end of the "M" by left-inside.
 */
#include "theBugAPI.h"

#define COLORSENSOR SENSOR_2
#define RED 5

task main() {

    move(43, POWER_HIGH, FORWARD);
    turn(85, POWER_HIGH, RIGHT);
    Wait(1);
    move(67, POWER_HIGH, FORWARD);
    turn(78, POWER_LOW, LEFT);
    Wait(1);
    move(22, POWER_HIGH, FORWARD);
    claw_control(55, POWER_NORMAL, BACKWARD);
    move(15, POWER_HIGH, BACKWARD);
    turn(85, POWER_HIGH, RIGHT);
    Wait(1);
    move(80, POWER_HIGH, FORWARD);
    turn(44, POWER_HIGH, RIGHT);
    Wait(1);

    // Turn on the RGB sensor
    SetSensorColorFull(S2);

    while(Sensor(S2) != RED) {
        OnFwd(OUT_BC, POWER_NORMAL);
    }

    PlaySound(GOT_RAINS_OF_CASTAMERE);
}
```

6 Musica - Game of Thrones

- 6.1 Objetivo da isso
- 6.2 Passos envolvidos
- 6.3 Pseudo-código
- 6.4 Código NXC

```

#define VOL 3
#define A3 220
#define AS3 233
#define B3 247
#define C 262
#define CS 277
#define D 294
#define DS 311
#define E 330
#define F 349
#define FS 370
#define G 392
#define GS 415
#define A 440
#define AS 466
#define B 494
#define SEMIBREVE_PONTUADA 4000
#define SEMINIMA_PONTUADA 1000
#define SEMINIMA 666
#define COLCHEIA 333
#define SEMICOLCHEIA 166

sub background_strings() {
    PlayToneEx(659, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(440, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(523, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(587, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(659, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(440, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(523, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(587, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(659, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(440, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(523, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(587, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(659, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(440, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(523, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(587, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(659, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(440, COLCHEIA, VOL, FALSE); Wait(COLCHEIA);
    PlayToneEx(523, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(587, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
}

task main() {
    // Main theme of game of thrones.
    PlayToneEx(392, SEMINIMA_PONTUADA, VOL, FALSE); Wait(SEMINIMA_PONTUADA);
    PlayToneEx(264, SEMINIMA_PONTUADA, VOL, FALSE); Wait(SEMINIMA_PONTUADA);
    PlayToneEx(311, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(349, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(392, SEMINIMA, VOL, FALSE); Wait(SEMINIMA);
    PlayToneEx(262, SEMINIMA, VOL, FALSE); Wait(SEMINIMA);
    PlayToneEx(311, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(349, SEMICOLCHEIA, VOL, FALSE); Wait(SEMICOLCHEIA);
    PlayToneEx(294, SEMIBREVE_PONTUADA, VOL, FALSE); // Wait(SEMIBREVE_PONTUADA);

    background_strings();
}

```

7 API

- 7.1 Objetivo da isso
- 7.2 Passos envolvidos
- 7.3 Pseudo-código
- 7.4 Código NXC

```
#define PI 3.1416

#define WHEEL_RADIUS 2.16
#define WHEEL_DIAMETER 4.32
#define CAR_RADIUS 6.65

#define RIGHT -1
#define LEFT 1

#define FORWARD 1
#define BACKWARD -1

#define POWER_NORMAL 75
#define POWER_LOW 50
#define POWER_VERY_LOW 25
#define POWER_HIGH 100

#define COMPLETE_ROTATION 360
#define HALF_ROTATION 180

sub turn(int angle, int power, int direction){
    int car_arc = (angle * PI * CAR_RADIUS) / HALF_ROTATION; // Measured in centimeter
    int amount_turns = (HALF_ROTATION * car_arc) / (PI * WHEEL_RADIUS); // Measured in centimeter

    RotateMotorExPID(OUT_BC, power, amount_turns, 100 * direction,
                      true, true, 40, 40, 90);
}

sub move(int distance, int power, int direction){
    int angle = (distance * COMPLETE_ROTATION * direction) / (PI * WHEEL_DIAMETER);
    RotateMotorEx(OUT_BC, power, angle, 0, true, true);
}

sub claw_control(int angle, int power, int direction){
    RotateMotor(OUT_A, power, angle * direction);
}
```

```

#define WHEEL_RADIUS 2.16
#define WHEEL_DIAMETER 4.32
#define CAR_RADIUS 6.65

#define RIGHT -1
#define LEFT 1

#define FORWARD 1
#define BACKWARD -1

#define POWER_NORMAL 75
#define POWER_LOW 50
#define POWER_VERY_LOW 25
#define POWER_HIGH 100

#define COMPLETE_ROTATION 360
#define HALF_ROTATION 180

sub turn(int angle, int power, int direction){
    int car_arc = ( angle * PI * CAR_RADIUS )/ HALF_ROTATION; // Measured in centimeter
    int amount_turns = (HALF_ROTATION * car_arc) /(PI * WHEEL_RADIUS); // Measured in centimeter

    RotateMotorExPID(OUT_BC, power, amount_turns, 100 * direction,
                      true, true, 40, 40, 90);
}

sub move(int distance, int power, int direction){
    int angle = (distance * COMPLETE_ROTATION * direction)/(PI * WHEEL_DIAMETER);
    RotateMotorEx(OUT_BC, power, angle, 0, true, true);
}

sub claw_control(int angle, int power, int direction){
    RotateMotor(OUT_A, power, angle * direction);
}

```

8 Garras

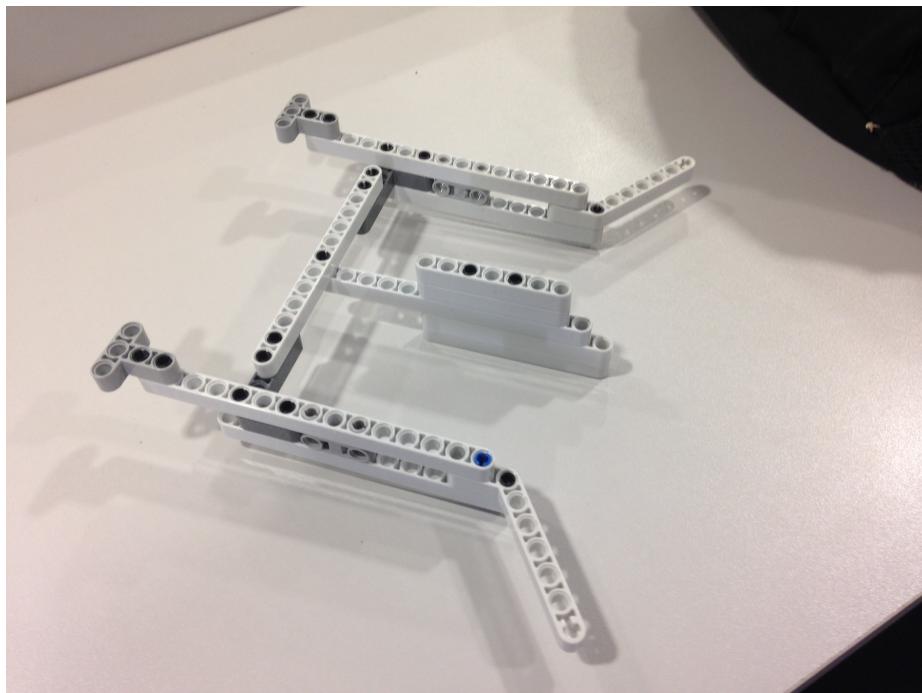


Figure 1: Base para acoplamento dos veiculos.

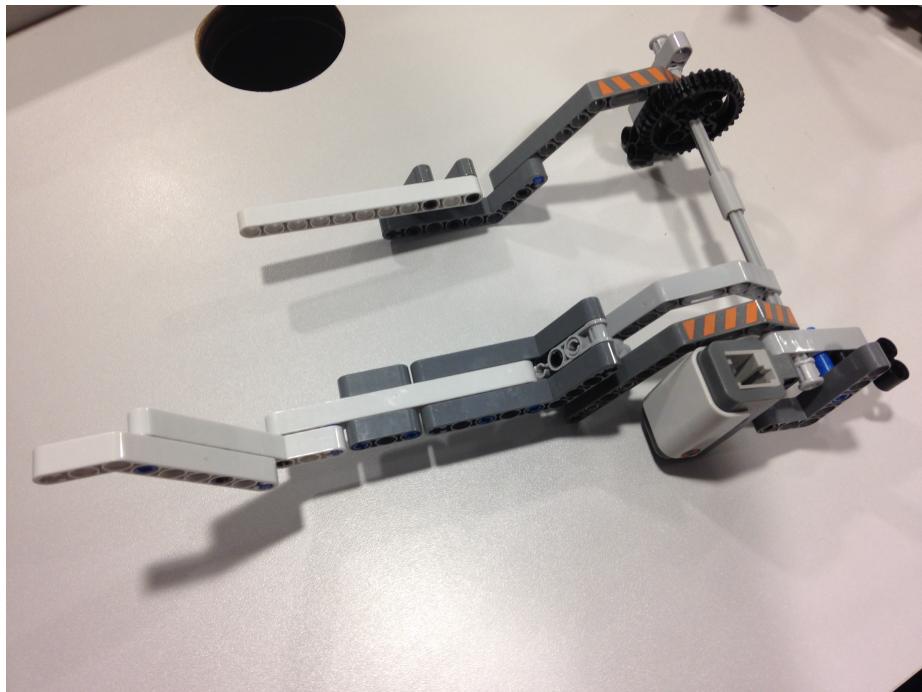


Figure 2: Garra para derrubar o galho e capturar objetos.



Figure 3: Brao para acionamento do mecanismo das ondas.