Dumpsty

P5 Project
Group SW510E16
Software
Aalborg University
21st Dec 2016

AALBORG UNIVERSITY
STUDENT REPORT

Fourth semester at
Department of Computer Science
Software
Selma Lagerlöfs Vej 300
9220 Aalborg East, DK
http://www.cs.aau.dk/

Title:

Dumpsty

Project:

P5-project

Project period:

September 2016 - December 2016

Project group:

SW510E16

Participants:

Christian Dannesboe
Frederik Børsting Lund
Karrar Al-Sami
Mark Kloch Haurum
Lasse Lyngø Nielsen
Søren Lyng

Supervisor:

Bent Thomsen

Pagecount: 23
Appendix range: X
Published 21-12-2016

Synopsis:

Preface

This project has been developed as part of the fourth semester project by project group SW408F16 from Aalborg University, Software Engineering, from the period 1st February to 26th May 2016 .
The project is based on the *Aalborg-model*, study method, where problem and project based learning is the focus. The theme of this semester was to create a compiler for a new language. To do so, some subjects were introduced and the subject the group chosen, was *Domain Specific Language for Robocoders* .

The group would like to thank supervisor, Giovanni Bacci for his very much appreciated advice and guidance during the whole project.

Signatures

Christian Dannesboe	Frederik Børsting Lund	Karrar Al-Sami
Mark Kloch Haurum	Lasse Lyngø Nielsen	Søren Lyng

Reading guide

This project has followed the courses Syntax and Semantics & Languages and Compiler. The context of this project has been written according to the order the course materials was taught and learned.

The sources in the report are being referred to by the Harvard citation method. This includes a last name and a publication year in the report, and in the ***Bibliography*** chapter all the used sources are listed in alphabetical order.
An example of a source in the text could be: ***[?]****.*
If the source is on the left side of a dot, then that source refers only to that sentence and if the source is on the right side of a dot, then it refers to the whole section.

Figures and tables are referred to as a number. The number is determined by the chapter and the number of figure it appears as.
For example: The first figure in a chapter will have the number ***x.1****, where x is the number of the chapter. The next figure, will have the number* ***x.2****, etc.*
The listings of source code are also referred to as the tables and figures.

Source code in the report are listed as code snippets, and they're not necessarily the same as the source code, meaning that code snippets may be shorter than the actual source code or missing comments from the source code. In order to show that, the use of the following three dots are used: "...", which means that some of the source code isn't listed in the code snippet, as it may be long and irrelevant.

Contents

1 Introduction

An embedded system is a computer system which only have a few functions. It is embedded as a part of a whole device, which then also includes hardware and/or mechanical components. Embedded systems are everywhere in our everyday lives. The range for embedded systems could be all from saving lives with pacemakers to fun gadgets.[Techopedia.com]

An embedded system as a gadget, is a technological and small object or apparatus, which has a certain functionality. This functionality is limited to a certain niche, some of which are unnecessary. Gadgets are known as a trendy person's accessory, and are also known from the cartoon Inspector Gadget.
In this project, the embedded system in consideration would be designed as a gadget, with a sole purpose of catching trash thrown at it. A trash bin that can catch trash would make the act of cleaning more fun and interactive, and the availability of each individual trash bin would be tremendously increased.

For this gadget to be usable, it must be designed as a real-time system, as this system will have certain deadlines for each task to be executed in time. The trash bin must be able to identify an object coming towards it, make some computation to specify a point to catch it from, and then compute a way to catch the object. All this must be done before the object lands on the ground, which makes the use of real-time system design a part of this project. A real-time system is a software system which is subject to a real-time constraint, and the system must control and affect an environment, by receiving data and process these, within a certain time-limit.

Analysis 2

From the analysis chapter it should be possible to derive the requirements for the smart dumpster, Dumpsty. The user stories will be initial and will help make the user requirements for the project. After the user requirements the information from the user stories will be analysed in greater detail, depicting three different phases of the process: Throwing, detecting/tracking and catching. These three phases will be the inspiration for the system requirements. The analysis chapter will end up with a problem statement for the project.

2.1 User stories

As mentioned before the user stories should help make the user requirements for the project, two user stories have been made that showcase the use of Dumpsty.

Benjamin is a software engineer who is tired of wasting his precious time on the job with walking back and forth from the waste bin in his office, and therefore wants to be able to throw his trash in the general direction of the bin instead. His aim when throwing the trash isn't that of a trained basketball player, so he often has to pick up the litter after throwing it at the bin.
Benjamin wishes that the waste bin could move and collect the trash for him, so that he can throw his trash in the general direction of the bin, and the bin could then place itself in a way, that allows it to catch the trash before it lands on the floor. This would optimize the time Benjamin uses each day on collecting the trash he did not land in the waste bin.
If Benjamin throws at the bin from a designated side of a sensory camera, the robotic waste bin should identify the trash, move the bin to a place where it would be able to catch the trash, before it hits the ground.
If Benjamin throws outside a designated perimeter of the robotic waste bin, it should not try to catch the litter, as it would compute that it is not able to get to the point of catching before the trash hits the ground.
If Benjamin and another person from his office throws trash to the waste bin at the same time, it should prioritize the first identified object.

2.2 User requirements

2.3 System capabilities

2.3.1 Throwing

2.3.2 Detecting and tracking

2.3.3 Catching

2.4 System requirements

2.5 Chapter Summary

2.6 Problem statement

Theory 3

3.1 Field of view

Skal vi have noget om det synsfelt vi har til at opfange bolden i??

3.2 Throwing

3.3 Detecting and tracking

3.4 Catching

3.5 Trajectory prediction

Hardware 4

4.1 Sensors

4.1.1 Microsoft Kinect

4.1.2 LEGO NXT Gyroscope

4.1.3 LEGO NXT Accelerometer

4.1.4 LEGO NXT Servo motor

4.2 Arduino

Design 5

5.1 The robot

5.2 Throwing

5.3 Detecting and tracking

5.4 Catching

Implementation 6

Tests 7

Discussion 8

Conclusion 9

Future work 10

Bibliography

Techopedia.com. Techopedia.com. *Embedded Systems*. `https://www.techopedia.com/definition/3636/embedded-system`. Accessed: 25-09-2016.

Appendix A