

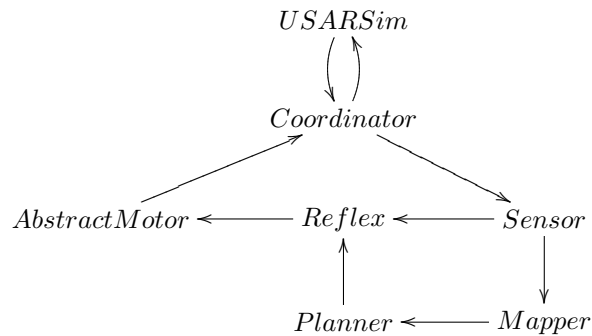
Draft Workplan

Waddie Assal Jeroen Rooijmans Jasper Timmer
Maarten de Waard Maarten Inja

June 8, 2011

1 Specification

2 Functionial Design



3 Different Components and Test

4 Deliverables

This section will describe the milestones, and what deliverables we will provide.

Milestone #0

This will describe what we have done by the end of week one.

A list of deliverables:

Working environment: We will have set up a Git repository¹ and a testing environment (using xUnit).

Base class: We will make an abstract base class, on which we can base all our Java classes. This will have all the standard things, like TCP/IP protocols.

Milestone #1

A list of deliverables:

¹<https://github.com/Y3PP3R/AP2DX>

Drive around: We want the program to be able to direct the robot through the environment. We will not yet focus on the ability to follow lines or walls.

Avoid collision: The robot should be able to avoid collision with objects and walls. It will stop, turn a random corner, and drive on. This way it will cover most of the area without colliding.

Classes needed to be implemented for this:

- Coördinator
- Sensor
- Reflexes
- Motor

Milestone #2

A list of deliverables:

Avoid obstacles: The robot will be able to avoid the obstacles that cross his path, instead of stopping and turning a random corner.

Navigate: The robot will be able to navigate through the room.

What we will implement for this:

Mapper: A class that creates a map of the room out of the sensor data. In the time of milestone 2 it does not have to be able to create an entire map and be very accurate, but it will be able to make some implementation.

Improved Sensor: The sensor class will be improved to be able to make an accurate map.

Improved Reflexes: The reflex class needs to be able to use some sensor data to be able to avoid objects appropriately.

Demonstration

Before the demonstration we will be able to do the following things:

Planner: We will have a planner class that can specify directions based on the current map position and what part of the map we have not yet discovered.

Improved Mapper: The mapper will now be able to make an accurate map and find our location on it, while taking the errors in sensor data into account.

Tests: Although we test all the time, we want to have tested everything good before the end.

Report: We will work on a report, describing our progress, problems and (test)results.

Documentation: We will work on a proper documentation of our code, which is also finished before the end.

The classes we will need to implement or improve for this are:

- Mapper
- Planner
- All test classes