Architecture Project
CS 685

Christopher Burns
George Mason University
CS 685 – Fall 2007
Prof. Sean Luke

"Architecture Project"

Abstract

The purpose of the architecture project is to provide a chance to apply what I've learned and add to the tool suite in the GMU robotics lab by creating new robotics architecture. New graphical components were added to the base tool set created in the first project, as well as significant re-coding to ensure the components conformed to OO implementation standards so would be easier for others to build upon in the future. The major graphical component added was a sonar mapping control. The control using a simple fade-in technique to show sonar hit locations as well as the current robot location and orientation.

Description

The Robot.java file was ported in the first project. The first project also included robot joystick control, sonar display, and robot status output. This architecture was a partial re-write of this first effort as well as adding additional mapping components and starting interfaces to other robots. The end goal was to produce a usable tool suite that could be used by and understood by other students.

The architecture consists of 5 major components:

RobotAdapter

The adapter project contains an IRobot interface definition as well as implementations for the Pioneer 3 and iRobotCreate. The iRobotCreate interface is not complete, but has been started as an example for potential future use. The Pioneer3 implementation is complete in the basic robot functionality and was the primary implementation used for this effort.

The Pioneer 3 class supports both the onboard integrator as well as includes a set of functions that can calculate position based on integrating wheel velocity.

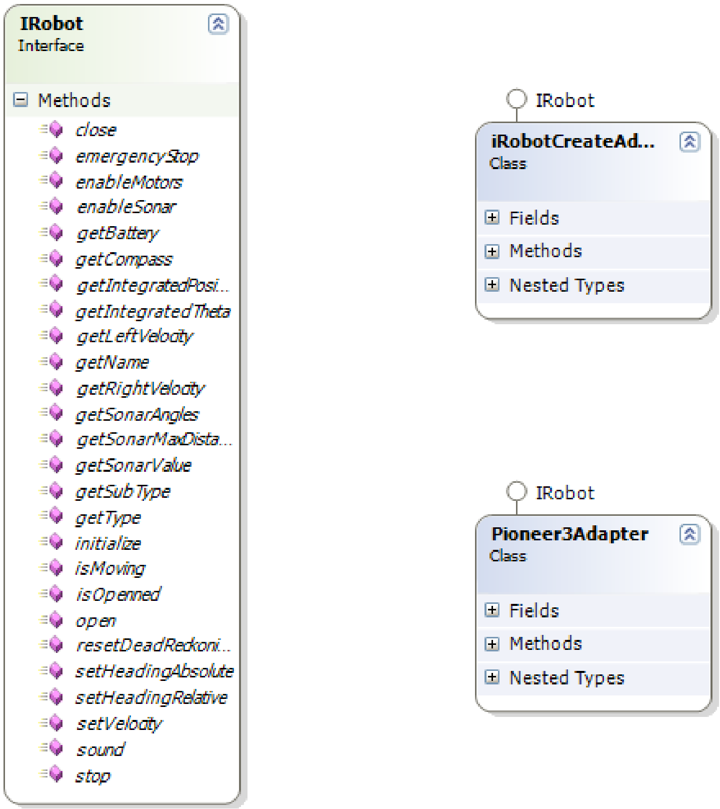


Fig 1, Robot adapters

Robo Mapper

This is the major graphical component for this architecture project.

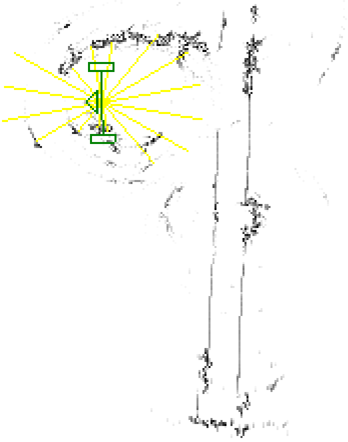


Fig 2, Robo Mapper in action

Sonar Viewer

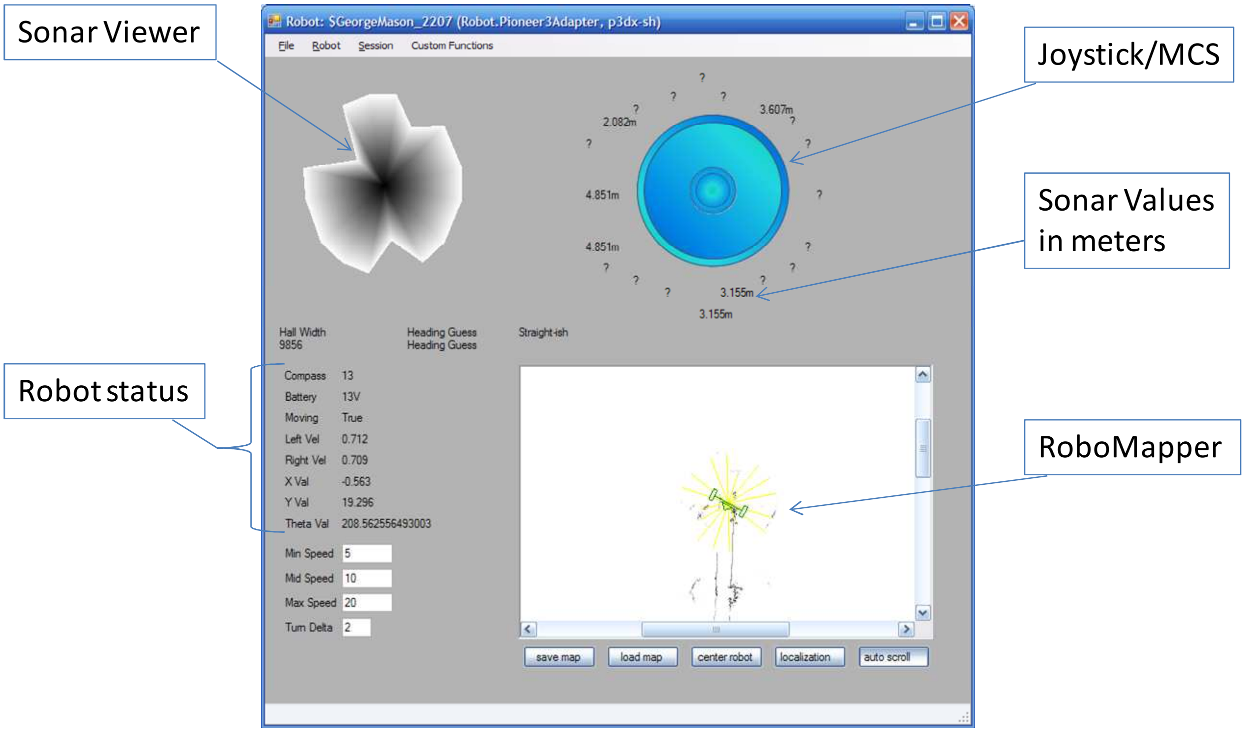
This component has been updated from the initial project to support the new robot interface and allow more flexible configuration options (configurable sonar counts and angles)

MCS/Joystick

Movement control system, and joystick adapter. This component has been updated significantly since the first effort, since the original control contained proprietary code and could not be released. The proprietary code has been re-written for this control.

Base GUI

Base GUI that contains all these controls. Also includes solutions to project 1 and 2 for this class. This GUI demonstrates how the components work and how to tie them together (e.g. get sonar and odometry data from IRobot object and set sonar and position data in mapper.)



Hall Width
9856

Heading Guess
Heading Guess

Straight-ish

Compass 13
Battery 13V
Moving True
Left Vel 0.712
Right Vel 0.709
X Val -0.563
Y Val 19.296
Theta Val 208.562556493003

Min Speed 5
Mid Speed 10
Max Speed 20
Turn Delta 2

Fig 3, Base GUI with MCS, Sonar Viewer and RoboMapper

Source Code

Embedding source code in a document is rarely usable, so the code will be compressed and sent with this document.