CS Capstone Proposal: PiBot

Andriy Rudyk & Tim Sizemore

30 August, 2013

CONTENTS

ACRONYMS AND ABBREVIATIONS

AI Artificial Intelligence

API Application Programming Interface

GUI Graphical User Interface

SSH Secure Shell

TSR Terminate and Stay Resident

UML Unified Modeling Language

1

OVERVIEW

In order to demonstrate the skills obtained from individual classes within the Computer Science program, a capstone project must be undertaken by each individual so as to be deemed satisfactory in the completion of their education at the University. This project will incorporate techniques acquired throughout course work in previous classes to develop a robust piece of software that employs all the necessary attributes of a software designer.

Undergoing an independent project is an ambitious endeavor that allows the student to express their creativity and display ingenuity. It also gives them the opportunity to use lesser known resources to accomplish a self set task. These task generally include web development, software engineering, networking, and third party code integration. All of which are used in industry careers depending on the field pursued.

A particular industry career on the cutting edge of software development focuses on making computers independent. Artificial Intelligence is the next great wave of innovation. It has pioneered the development of some of the most complex languages and every year Computer Scientists are closer to breaking the mechanical and biological barrier. While AI is being developed for all facets of life, the personal use of these components is the most compelling. How will AI mesh with the everyday lives of individuals?

The benefits of having a robot take over the menial tasks in life are numerous and can easily be compared to the change the electric washing machine had on lifestyles at the beginning of the 20^{th} century. While the social repercussions of AI are plenty and well developed, technology, by design, is neither moral or immoral. An android with the ability to substitute the place of a human has far reaching implications; such as life threatening situations, assisting handicapped persons, and allowing remote access to specific areas to name a few.

In an attempt to understand how crucial AI is, we will try to create a robot centered around a goal-based agent that uses a knowledge base to infer its partially observable environment. In order to give the project structure, we have created general deadlines for production. The measure of completion will occur at the end of each month. For September, the hardware components for the robot should be installed and a basic website is running that allows users to login and request controller accounts. For October, the website should have a more polished look and the the robot will be able to accept movement commands from the web application. For November, we will finalize key features of the web application and begin AI programming. The specifics of this aspect are unclear and as of now the task environment is about the only concrete knowledge we have. And finally December, the robot and the website should be completely functional with limited components of AI potentially working.

1.1 PROBLEM STATEMENT

The purpose of this project is to create a robot capable of substituting itself in place of a human for a virtual tour of a designated space. The robot will be remotely controlled with the ability to transmit video to an external source, all while avoiding potential dangers while in operation.

1.2 REQUIREMENTS SPECIFICATIONS

The program and robot will have the following requirements:

- The robot must be independent of the application that controls it. As long as the application can SSH into the robot and accept video feed, the robot should respond accordingly.
- The robot must be able to move freely on a four point axis.
- The controlling application must have a command line interface and a GUI.
  - The GUI will consist of a Django application containing a user database as well as a web application that will allow the user to login to a PiBot session.
- The robot must be able to avoid hazards while in operation.
  - Hazards include, but are not limited to, stairs, walls, and other object that would impede operation.
- In the case that connectivity is lost, the robot must retrace its steps, if possible, until connection is re-established.
- The user must be able to register for a PiBot controller account through the web application.
  - The accounts will only exist for a limited time after registration. Afterwards they will be deleted from the system.

1.3 TESTING

The testing of this project can be broken into two parts. Firstly, the robot and its various mechanics. Following the definitive testing of both hardware and software components, the web application's functionality is the next step. Most testing will occur when functionality is added but for a linear description, the previous sentences will apply.

The robot will be controlled using a Raspberry Pi module and must be able to move along a four point axis meaning left, right forwards, and backwards. To test this aspect the robot will be placed in a sanctioned area with minimal obstacles and then navigate the obstacles using TSR software. The robot will also have a camera used to display its forward surroundings but its functionality will be tested along with the web application. This will conclude most of the hardware and some software testing for the robot.

As for the web application, a GUI must be constructed and a user database established in order to test browser compatibility. As for supported browsers, the

testing portion of all code will occur on Chrome and FireFox. Other browsers should be supported without much difference in appearance as we will conform to W3C markup validation standards.

To test the GUI mock users will be registered and attempt to navigate select features as well as register for a PiBot controller account and attempt to move the robot. Since a lot of the aspects must be completed before the project can be fully tested, this procedure will encompass three major components; user registrations, basic site navigation, and remote access to the robot. While this is not idyllic, individual component issues will only develop once code is introduced to the whole system. But by limiting the number of major components involved in the test, it will still allow for easier debugging.

With respects to AI, we are both novices and as far as testing the AI, it would essentially involve trying to break it. The robot should be able to navigate back to a position with Internet connectivity if that connection should ever break. To test this feature, we plan to disable the wireless connection on the robot and see if it does as we anticipate. The other facet of AI we are implementing has to deal with hazard detection. As of now we plan on using OpenCV for object detection and the following test will reflect that applications limitations but the actual program we use may differ. OpenCV uses preset point configurations to remember shapes of objects. Using this feature the robot should be able to identify doors, walls, stairs, etc. Essentially any object that would inhibit movement, the robot will potentially avoid or wait for the obstructing object to move such as a closed door or a person.

For the last phase of testing, we will enlist external persons with limited technical skills to give feedback of the web application and robot controls. From their feedback changes will be made accordingly and the process will be repeated until the only suggestions we receive are preference based. While this concludes the major component functionality testing, there will be significantly more testing that cannot be anticipated as problems arise during production.

2

SCHEDULE OF COMPLETION

10th September 2013

Have a functional robot that can take command line input though ssh which would control the electrical motors. The robot will consist of one chasi, two electrical motors and a raspberry pi mini computer with a camera. Linux daemons to control the motors will have to be written.

24th September 2013

Start a django website with a PostgreSQL database that would give users the ability to register for a ”usage time slot” and log-in into the system at the appropriate time.

8th October 2013

Have the ability to transmit movement commands to the raspberry pi using the web interface designed in the previous section.

22nd October 2013

Pass video feed into the web interface from the raspberry pi.

5th November 2013

Finalize and ”brand” the first ”western tours” part of the project. Make registering, log-in, session expiration based on time and other ”core” features stable.

19th November 2013

Add basic AI feature #1, ”The stack”, this will record all the robot steps into a stack so the robot has a history of every user actions and could possible ”retrace” the previous steps taken.

3rd December 2013

Use the previous AI feature to trace back steps in case wifi connectivity has been lost with the PiBot. Make this a built-in default feature

17th December 2013

Implement a new AI feature which would take the robot back to ”home base” after each user finishes their ”tour”.