BarBot17

Joseph Olin

Michael Roark

Branden Wagner

C458 Project Proposal

February 16, 2017

Abstract

The purpose of this project is to design and implement a robot that can act as a waiter. The goal of this bot is to bring consumables to customers who are sitting at a counter. It will do so using an array of sensors, including a color sensor, line follower sensors, and an ultrasonic sensor. All these sensors will be systematically integrated into an Arduino microcontroller, which will provide the processing and general "thinking power" of the system. Using this system, our objective is to demonstrate how a robotic system can make everyday tasks, such as tending bar, an automated process handled by an intelligent robot.

Introduction

Attending customers at a counter is a repetitive task. First, the waiter/waitress must ask the customer if they desire a consumable. Then, the waiter must retrieve the consumable from where it is stored. Once the consumable is retrieved, the waiter makes a pass over the length of the counter, giving customers the order they desired. One notable model for attending customers can be found at Brazeiros in Louisville, Kentucky.

At Brazeiros, customers are given a color-coded coaster, with one side being red and other side being green. Based on the color of the coaster the customer shows, the customer may or may not be continually served freshly prepared meat. Our BarBot project proposes to take the repetitive task of attending customers at a counter and make that task the responsibility of an intelligent robot rather than a human.

The idea is the BarBot will load itself via an arm attached to the top of the bot. Customers will each have a double-sided card with a certain color (e.g. red) on one side and a different color (e.g. green) on the other side. Similar to how other restaurants such as Brazeiros operate, one color will indicate the customer wants an order and the other color will indicate the customer is content and does not desire an order.

The robot will make cycles through the counter, ignoring unoccupied spaces and customers who use the color-coded card to indicate they don’t desire an order. The BarBot’s goal will be to deliver items only to customers who desire an order as quickly as possible via the attached arm.

System Details

Environment

There are 3 main parts of the environment: the storage/loading area, the bot traversal area, and the customer area. The environment will consist solely of the counter, which will be simulated with a table.

The storage/loading area is to be at the end of the counter. There, a hopper or some similar storage container will contain the consumables (e.g. mints). The bot will approach the hopper at the end of a bar run and the hopper will be opened, allowing the consumables to empty into the storage basket on the bot.

The bot traversal area is the space on the counter where the bot moves. The space stretches from one end of the counter to the other end. The bot will move forward from the hopper along this traversal space to attend each customer and, when the bot reaches the end of the counter, it moves backward to the hopper to start another run.

The customer area is the area closer to the edge of the counter where customers will display their color-coded card to the bot, and where the bot will also place the customer’s order. To start with, the number of customers allowed at the counter will be static. However, as the project progresses, delivery memory will be implemented so the bot knows what customers were delivered to in a run.

Sensors

The sensors likely to be needed for the bot to perform the aforementioned tasks are: two line follower sensors, an ultrasonic sensor, a color sensor, and possibly some pressure sensors for the arm (if none come with it).

The line follower sensors will be for 1) notifying the bot when it is near the edge of the counter and 2) possibly for notifying the bot when it has reached the loading area (where a piece of tape would mark where it needs to stop).

The ultrasonic sensor might alternatively be used in place of the second line follower sensor to notify the bot when it has reached the hopper.

The color sensor will be used to notify the bot what color card the customer is displaying, thus notifying the bot if it needs to give the customer an order.

The pressure sensors will be needed to notify the bot how much pressure the arm is using in gripping a consumable.

Performance Matrices

Two primary performance metrics we intend to use in this project are: the number of orders the bot is supposed to fulfill (based upon the number of green colored cards displayed by customers), and the time the bot takes in fulfilling those orders. For the first metric, a count will be taken on the number of green colored cards displayed on a run, which can then be compared with the number of consumables the bot handed out. For the second metric, a timer will be implemented externally via a physical timer, which will track the time taken to make a delivery run.

System Functionalities

- Bot will move forward along the counter
- Bot will move backward along the counter
- Bot will stop at loading area
- Hopper will load bot

- Arm will grab and hold on to a consumable
- Arm will unload a consumable
- Bot will recognize color cards
    - Bot will unload a consumable at a green card
    - Bot will pass a customer at a red card

Related Research

Our approach was to define the problem first, then design the robot. As the problem was refined, we also needed to alter the design. Some designs solving a similar problem include Zexuan, et al (2015), and Casavela (2012). Our design intends to combine movement of a wheeled base with the delivery capabilities of an arm into an autonomous delivery system for small solid objects.

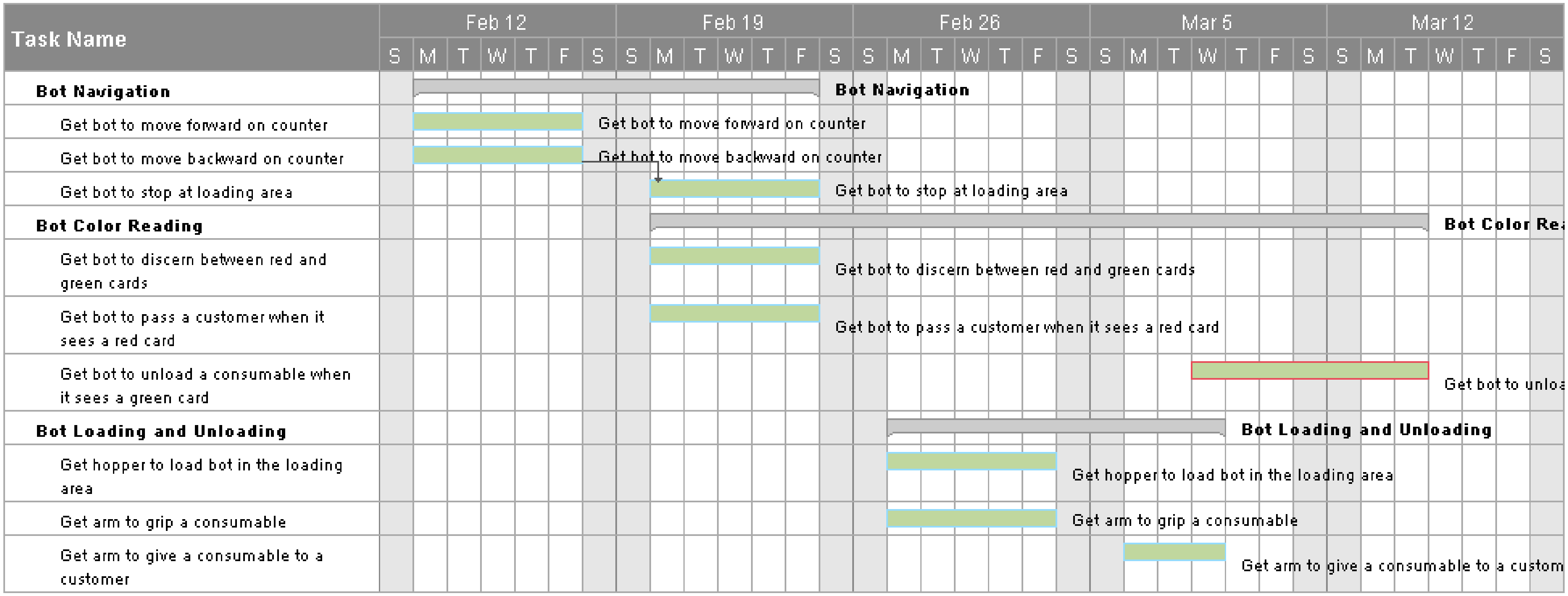
We decided to use a color coding system for ordering based on observations of the system in use at Brazeiros. Their system uses a green card to start service and a red card to stop service. (Brazeiros 2017)

Path planning is a difficult obstacle to overcome for autonomous robots. Zexuan, et al, (2015) shows an example the procedures needed to plan transportation along a set path. Our project plans to use a combination of physical aids and programming to overcome obstacles in the path. We intend to use a fixed and relatively small path traversed in two directions.

Ju (2016) explains a strategy for manipulating small round objects using positive air pressure. In this project they found that having a more flexible solution added efficiency and gave unexpected side benefits. We decided to add a similar system to our arm because of the ease of use and the relatively low cost.

Casavela (2012) explains the challenges he overcame in using C++ to program an Arduino. We were able to draw many lessons to design the necessary programs to drive our systems. This paper also gave several insights on how to engineer robotic systems and properly interface them.

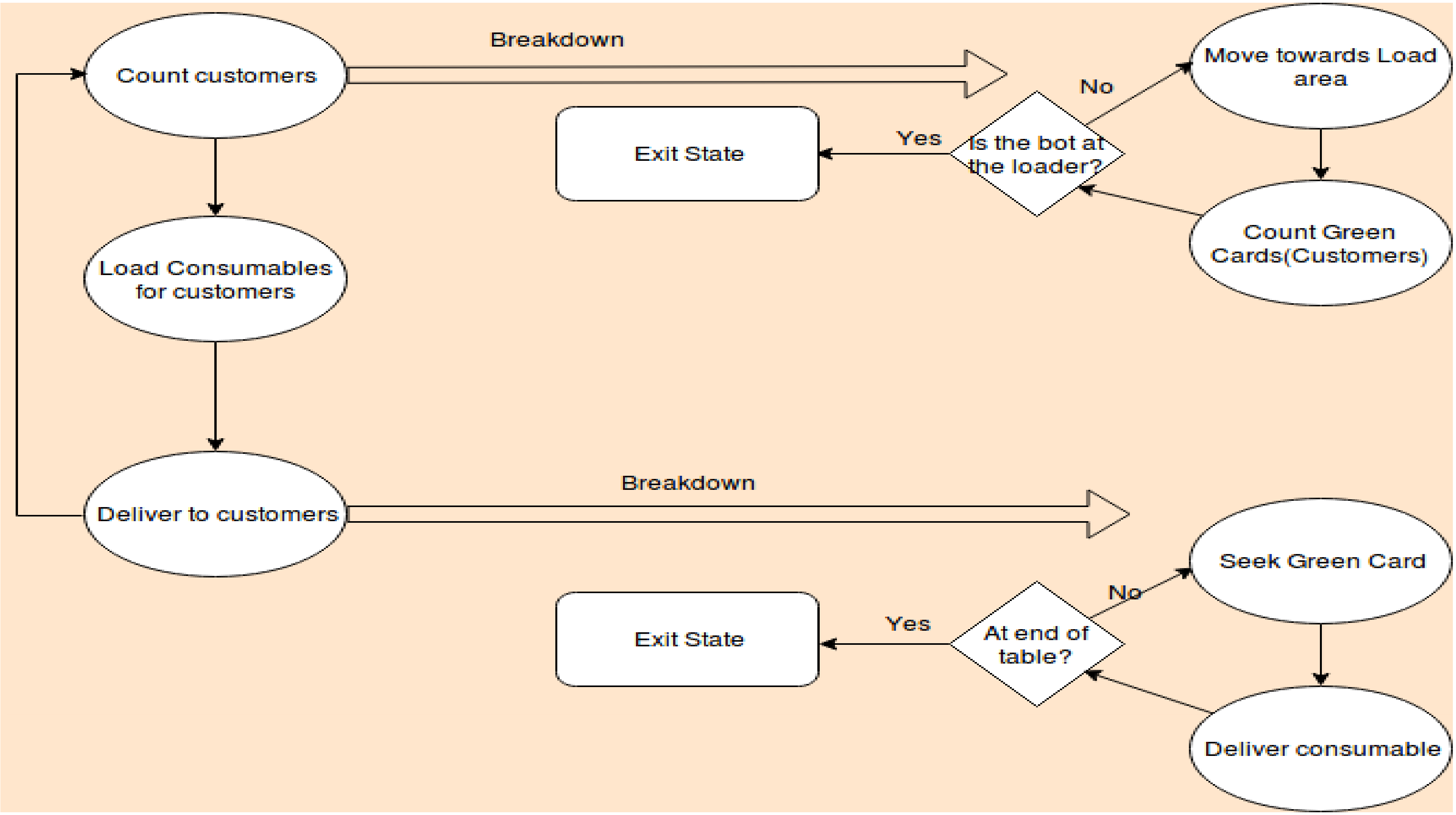
Timeline (Feb 12 – Mar 12)



	Task Name
1	**Bot Navigation**
2	Get bot to move forward on counter
3	Get bot to move backward on counter
4	Get bot to stop at loading area
5	**Bot Color Reading**
6	Get bot to discern between red and green cards
7	Get bot to pass a customer when it sees a red card
8	Get bot to unload a consumable when it sees a green card
9	**Bot Loading and Unloading**
10	Get hopper to load bot in the loading area
11	Get arm to grip a consumable
12	Get arm to give a consumable to a customer

Feb 12
Feb 19
Feb 26
Mar 5
Mar 12
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S
Bot Navigation
Get bot to move forward on counter
Get bot to move backward on counter
Get bot to stop at loading area
Bot Color Rea
Get bot to discern between red and green cards
Get bot to pass a customer when it sees a red card
Get bot to unloa
Bot Loading and Unloading
Get hopper to load bot in the loading area
Get arm to grip a consumable
Get arm to give a consumable to a custom

State Transition Diagram



Division of Labor

Version Control Manager – Branden Wagner

Bot Navigation – Michael Roark

Bot Color Reading – Joseph Olin

Bot Loading and Unloading – Branden Wagner

Presentation Preparer – Joseph Olin

References

Brazeiros . (2017). Meats. Retrieved February 16, 2017, from http://www.brazeiros.com/menu/meats/

Casavela, S. (2012). C++ PROGRAM FOR DRIVING OF AN AGRICOL ROBOT. Annals Of The University Of Petrosani Mechanical Engineering, 1411-19.

Ju, B. (2016). Robotic gripper can throw darts, balls - with no arm motion | Cornell Chronicle. Retrieved February 16, 2017, from http://www.news.cornell.edu/stories/2012/02/robotic-gripper-can-now-throw-things

Zexuan, Z., Jun, X., Jian-Qiang, L., Fangxiao, W., & Qingfu, Z. (2015). Global path planning of wheeled robots using multi-objective memetic algorithms. Integrated Computer-Aided Engineering, 22(4), 387-404. doi:10.3233/ICA-150498