Final Project

William Harrington, Sheetal Konnur
ECE478

History and Introduction

This report contains a detailed explanation of the Final project for the Marie Curie group.

The Marie Curie group was formed into the current state around Mid-November of 2015. Around this time Dr. Perkowski decided to move Will Harrington from the Einstein group to the Marie Curie group which only consisted of one member: Sheetal Konnur. The group was informed that they were to control the Marie Curie robot with a kinect in order to implement some sort of behavior.

Essentially, this is what is referred to as **Behavior-based robotics**. Behavior-based robotics is "..an approach in robotics that focuses on robots that are able to exhibit complex-appearing behaviors.." [1]. Upon learning this, the group embarked on using a kinect for controlling Marie Curie by using a raspberry pi 2 for image processing and an arduino with a pololu driver to control the servos of the robot. However, after working on this for a few weeks, the team was informed by Dr. Perkowski that this is not what he wanted. Apparently, Marie Curie had a long storied past that the team was unaware of. Other students who had worked on Marie Curie already had done "better algorithms" for image processing and controlling the robot. What Dr. Perkowski wanted was for our group to evolve this work further.

At this point in the term, it was a long shot that a group of two students would be able to do what Dr. Perkowski wanted. The group decided to simply undertake a more refined version of homework 1. Homework 1 involved using a powerpoint to do scenario prototyping. Therefore, the kinect

[1] https://en.wikipedia.org/wiki/Behavior-based_robotics

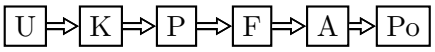
would be used to control the powerpoint and another means of controlling the robot would have to be triggered with the powerpoint. A Finite State Machine with probabilistic logic is also needed for this.

In conclusion, the final project for the Marie Curie group became a behavior-based robotics application with a focus on scenario prototyping using a finite state machine that involves probabilistic or fuzzy logic.

Goals

1. Use powerpoint for scenario prototyping
2. Use kinect to control mouse on computer so that it can be used to click buttons within the powerpoint
3. Use an arduino to drive the servo controller (Pololu)
4. Trigger arduino using the firmata protocol
5. Implement Finite State Machine with probabilistic logic for robot behavior

High level system model



- U = User
- K = Kinect
- P = PowerPoint
- F = Firmata
- A = Arduino
- Po = Pololu

High level system explanation

The high level system diagram shown above depicts the flow of inputs for our system. The user controls the mouse on the computer by using their hands while in front of a kinect. The kinect receives the input from the user and makes the appropriate mouse clicks on the computer. When the user clicks a button on the powerpoint, a program that uses the firmata protocol sends a signal to the arduino. The arduino then triggers the pololu controller to move the servos according to what behavior we want our robot to enact.

Implementation

Roles

- Will Harrington - Python and Arduino programming, powerpoint macros, kinect software, documentation
- Sheetal Konnur - Arduino programming, powerpoint macros, powerpoint presentation

Materials used

- Windows Laptop
- Microsoft Kinect
- Arduino Uno
- Micro Maestro 6-Channel USB Servo Controller

Required software

- Microsoft PowerPoint
- Kinect Mouse [2]
- Python
  - pyFirmata [3] [4]
- Arduino
  - Firmata for Arduino [5] [6]
  - PololuMaestro [7]
- Pololu Maestro Servo Controller Software [8]

Setup

A windows laptop must be used in conjunction with a Microsoft Kinect and Arduino Uno. The Marie Curie robot should already have Micro Maestro

[2] http://futuretechblog.com/?p=26
[3] https://pypi.python.org/pypi/pyFirmata/0.9.5
[4] https://github.com/tino/pyFirmata
[5] http://www.instructables.com/id/Arduino-Installing-Standard-Firmata/?ALLSTEPS
[6] http://playground.arduino.cc/Interfacing/Python
[7] https://github.com/pololu/maestro-arduino
[8] https://www.pololu.com/docs/0J40

6-Channel USB Servo Controllers. If not, those need to be procured and included in your setup or this will not work. The servo controller needs to be hooked up properly.
The powerpoint presentation located here [9] should be opened and started prior to starting the Kinect Mouse software. Furthermore, the resources available at the git repository for this project located here [10] need to be installed. The macros in the powerpoint must be modified to match the installation directory of the software from our repo otherwise the macros in the powerpoint will not work properly.
Once all of that is setup, start the Kinect Mouse software to control the powerpoint and press the buttons in it as desired to make the robot perform.

Details and explanation

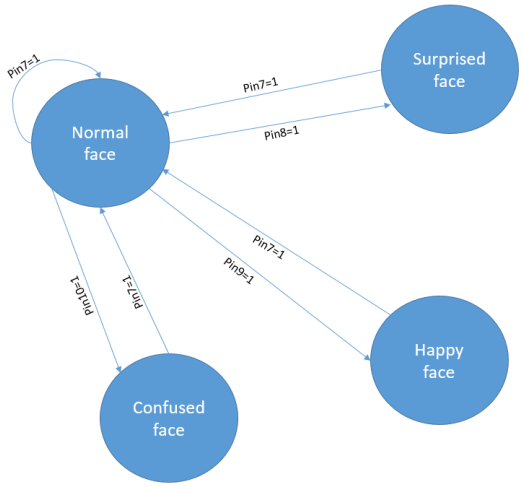
The kinect is controlled by using the KinectMouse software. A powerpoint presentation was made by the group containing information about Marie Curie and macro buttons, which call python scripts that use the firmata protocol to communicate with the arduino that is connected to the host computer. The arduino code is a combination of a Pololu driver and a firmata driver for the Arduino Uno. Our portion of the code simply reads pins on the arduino that are encoded to trigger the appropriate action when the designated pin is high. Each script always returns Marie to her normal face at the end.

The arduino pin encodings are as follows:

- Pin 7 controls Normal behavior
- Pin 8 controls Surprised behavior
- Pin 9 controls Happy behavior
- Pin 10 controls Confused behavior

[9] https://github.com/wrh2/ECE478/tree/master/Marie_Curie
[10] https://github.com/wrh2/ECE478/tree/master/Marie_Curie

Finite State Machine



Results

Demo Video [11]

Problems

1. The Finite State Machine does not involve probabilistic or fuzzy logic
2. The mouse is very difficult to control with the KinectMouse software

[11] https://youtu.be/GLy8DHwoF8I