Team Name: Guugle

Members: Andrea Chamorro, Kai Johnson, Lucas Laughlin, Yushen Chang, Matt Cohen

Description: An autonomous RC car controlled by a microcontroller (MSP432 or Raspberry Pi). The car will drive around on its own collecting sensor data (temperature data). The data will be collected by the microcontroller and then transmitted to computer via Bluetooth and stored in a database. The front end application will be built with HTML and CSS and pull data pertinent to the current user for display.

Using this application and device, this project is a prototype for an autonomous rover device that can amass data from an unknown room or area. This project can be useful in application of exploration. It can travel to unknown regions and collect temperature data on that region. This could be used in caves, abandoned structures, or the vast depths of space or sea to visualize temperature patterns in an area.

Vision Statement: To learn and apply the knowledge for designing and producing a product from scratch, and to experience debugging and fixing errors through the process.

Motivation: One of our members is skilled in using RC car technology. In today's world, many real world applications of computer science including processing data collected from the real world, and visualizing it on a web-based display. This is how oil companies maintain their oil rigs remotely, for example, how weather data is interpreted for the world to use, and has vast application in the field of marine biology.

Strengths: Hardware- Software Communication, OOP, C, C++, Python, Objective-C, Java, HTML, CSS, Google-searching, ScrML, Web-Scrapping, Networking, (**add your skills here)**

Time-management is a risk we can assess by having a schedule for ourselves outside of classwork, and padding extra time in case something goes wrong.

Risks: The hardware we are using for the toy car could fail. The circuit or microcontroller could end up getting shorted, which would then damage our hardware. There is a possible risk in team dynamic. There is a the possibility that a team member will not follow through with their individual responsibilities. There is a possibility that something could go wrong with the software. For example, the software could be very complicated to debug. Then there is the risk of having problems incorporating the software and hardware together. The microcontroller and the device used for the web server may have issues communicating.

Risk Mitigation Plan: Overcoming risks involves being ahead of each viable risk. For example:

- Hardware Failure
  - When working with hardware there is a risk that the hardware could get shorted or it may malfunction. In order to mitigate this risk, it is important to verify that all the connections are properly connected. It is also important to use an

oscilloscope to test the outputs of the microcontroller to make sure that its outputting the right signal.
    - If a hardware component does malfunction, we would have to purchase that component again. This isn't really an issue since most components used are relatively inexpensive.
- Software Failure
    - In order to mitigate this risk, we would have to be sure to write clean organized code from the start. The code for the software portion should be well documented with comments, so that the debugging process goes well.
- Hardware and Software Compatibility
    - The best way to mitigate this risk is communication. We can mitigate the risk as long as the software and hardware members of the team coordinate with each other on their individual processes.
- Team Member Work
    - To mitigate the work of each member, the class participation requirement is already a good instigator for each member putting in their portion of the work, but along with this we will have weekly check-ins where each member shows how far along they are, so nobody will lag behind.
    - Each team member has a unique role in this development:
        - Yu-Shen: Hardware
        - Andrea: Database
        - Lucas: Front-end
        - Kai: Back-end
        - Matt: Software / Debugger

Version Control: We will be using Github. The following are our usernames:

- Natty-Laugh, mattcohen1326, KaiAsian, Yushen-Chang-9, achamorr

The following are Repositories for our team:

- git@github.com:kaiasian/Meeting-Logs.git
- git@github.com:kaiasian/Milestones.git
- git@github.com:kaiasian/Project-Components.git

Development Method: We are planning to go with an agile development process. At the moment we are unsure of all the details and certain components of our project are subject to change. We may go with different hardware platforms. We may even decide to change what software platforms we use for back end and front end. We imagine that the development process of this project is likely to be unpredictable, which is why we decided that to be more flexible with our development process.

Collaboration Tool: Slack and GitHub

Proposed Architecture: A hardware RC car built with a Raspberry Pi to collect data which will be stored using SQL natively on a user's home computer. A front end application built with HTML and CSS (+JS functionality) will fetch data from the SQL database to present data to the user.