Individual Progress Report

Proof of Concept B

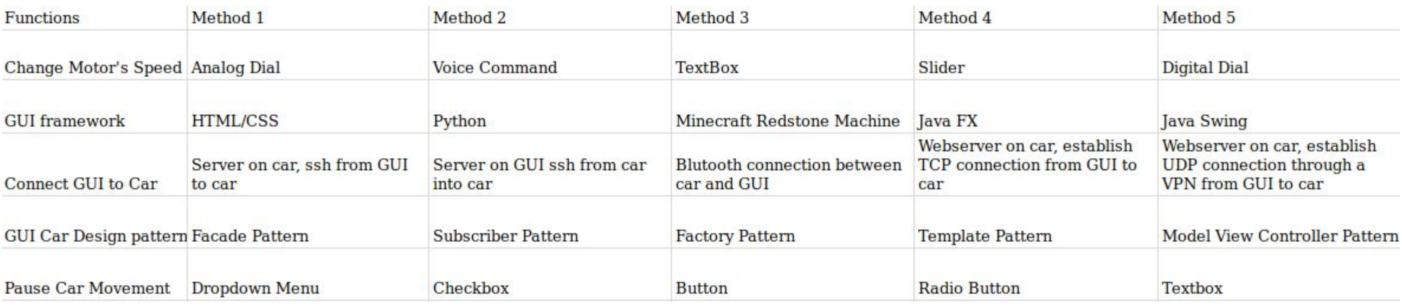
Project Details

Project name	EGEN 310 RC car Design
Group Number	E2
Author, discipline	Aaron McCarthy E2
Reporting period	January 29 2018 - February 9, 2018
Date Due	February 9, 2018

Summary

I made a lot of progress towards completion of both my GUI and the communication system that will carry information between the GUI and the car. First I researched the languages that would be best for creating a GUI and a system of communication. I found that Python is the best choice due to the large number of networking and GUI packages that are both well documented and free. I also did research to find what GUI package I will use to create the User interface. I found that Tkinter is the de-facto standard for creating simple GUIs using the Python language. I also modeled the GUI that I will be building on a whiteboard. I felt that this was necessary because the configuration of the car has changed and the GUI should reflect the changes in the design of the car.

Morph Chart



Functions	Method 1	Method 2	Method 3	Method 4	Method 5
Change Motor's Speed	Analog Dial	Voice Command	TextBox	Slider	Digital Dial
GUI framework	HTML/CSS	Python	Minecraft Redstone Machine	Java FX	Java Swing
Connect GUI to Car	Server on car, ssh from GUI to car	Server on GUI ssh from car into car	Blutooth connection between car and GUI	Webserver on car, establish TCP connection from GUI to car	Webserver on car, establish UDP connection through a VPN from GUI to car
GUI Car Design pattern	Facade Pattern	Subscriber Pattern	Factory Pattern	Template Pattern	Model View Controller Pattern
Pause Car Movement	Dropdown Menu	Checkbox	Button	Radio Button	Textbox

Activities

Research languages to use for GUI and PI

Status	Achieved
Objective	Find out what languages have both good networking and GUI building packages for both client side (GUI) and host side(Car).
My time on this task	2 hours
Support team member(s) time on task	Not applicable

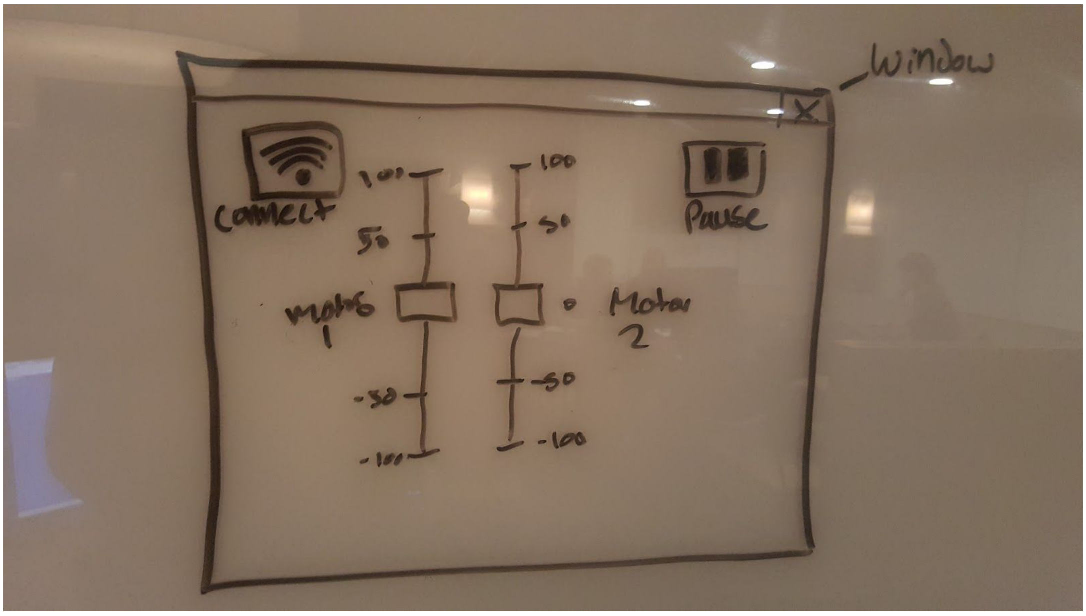
Visual Progress Update	Not applicable
Current Progress	I have researched various languages and have decided to use Python
Outputs created	I now know what language to make the GUI in and what language to use to receive commands on the car side. I can now begin modeling and prototyping in Python
System Integration Considerations	It is important that the language that I choose is lightweight enough to be run on a raspberry Pi and fast enough to support real time control of a car. Also the language must have good, well documented packages for sending information over a network and for building GUIs.
Challenges/Lessons learned	Python is a good language to use for controlling an RC car

Research GUI toolkits

Status	Achieved
Objective	Find out what Python Packages will be useful in building the GUI
My time on this task	2 hours
Support team member(s) time on task	Not applicable
Visual Progress Update	Not applicable
Current Progress	I have found the the de-facto standard toolkit for creating GUIs with Python is Tkinter so I will be using that due to its widespread adoption and its good documentation
Outputs created	I can now begin modeling rough draft GUIs
System Integration Considerations	It is important that the package I choose is lightweight enough to run on a raspberry PI while also supporting the functionality that I will need. I also require something that is popular enough to have good documentation as well as answers to common problems on the web.
Challenges/Lessons learned	It is often best to go with what is popular, even if it is not tailored specifically for your needs because you will be able to find answers to your problems much faster.

Model GUI

Status	Achieved
Objective	Create a rough on paper design draft detailing the layout of the GUI
My time on this task	1 hours
Support team member(s) time on task	Not applicable

Visual Progress Update	 Window connect 100 50 Motor 1 -50 -100 100 50 0 Motor 2 -50 -100 Pause
Current Progress	I have created an updated model of the GUI of the car that reflects the decision to steer the car via tank controls.
Outputs created	I can now begin coding rough draft GUIs
System Integration Considerations	It is important that the model of the GUI is up to date with the design of the car. The finalized decision to use tank controls is reflected in the model. Furthermore, the functionality to stop and start the car is also included in this model which is a feature that the group agreed would be useful to have.
Challenges/Lessons learned	none

Total Time On Task for this Milestone

Total time spent by me	5 hours
Total time spent by support team members	none

Next Steps

One of the next tasks will involve programming a prototype GUI in Python to be used by the driver to control the car. I will also be modeling and prototyping the system of communication between the controller and the car. This will involve researching networking libraries and to send information from the GUI to the car as well as setting up a server on the car to receive the information.

Archived Activities:

Research methods of communication

Status	Achieved

Objective	To decide if I will use Wifi or Bluetooth
My time on this task	4 hours
Support team member(s) time on task	Cole Jungers , 1 hour
Visual Progress Update	Not applicable
Current Progress	I have researched the two forms of communications and have decided to use Wifi
Outputs created	I now know what type of microcontrollers to consider to control the car. It is only microcontrollers that have Wifi support.
System Integration Considerations	It is important that the microcontroller that I choose is lightweight enough for the car to be able to pull it. Also the car has to be able to fit the microcontroller inside the body. Also we need to be able to fit enough batteries in the body to power the microcontroller and the motors.
Challenges/Lessons learned	People who took the class last semester are good resources

Decide upon and purchase a Microcontroller to control the car

Status	Achieved
Objective	Decide upon and purchase a Microcontroller to control the car
My time on this task	4 hours
Support team member(s) time on task	Andrew Leicht, 1 hour
Visual Progress Update	
Current Progress	I have found the optimal choice for a controller. The raspberry Pi Zero WH is a small lightweight microcontroller that supports both Bluetooth and Wifi connectivity while also being very small and very light weight. Furthermore it uses very little power and is easy to program.
Outputs created	We now know what microcontroller will be used to control the car.
System Integration Considerations	Deciding upon a microcontroller that controls the car helps move planning along for other electrical components and helps to start the group thinking about how all the pieces contained in the body should be arranged.

Challenges/Lessons learned	

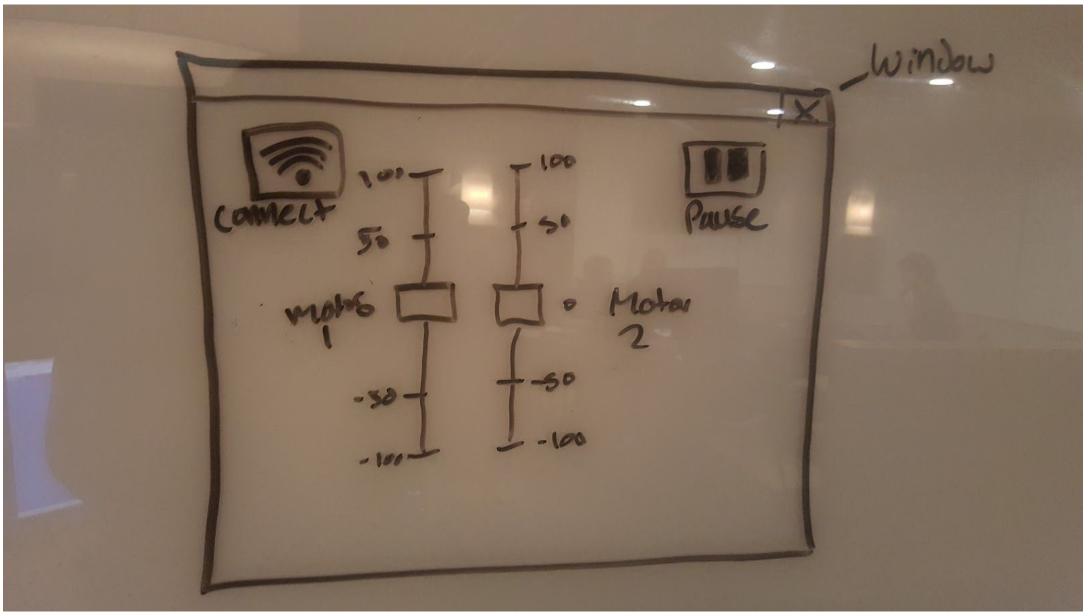
Research languages to use for GUI and PI

Status	Achieved
Objective	Find out what languages have both good networking and GUI building packages for both client side (GUI) and host side(Car).
My time on this task	2 hours
Support team member(s) time on task	Not applicable
Visual Progress Update	Not applicable
Current Progress	I have researched various languages and have decided to use Python
Outputs created	I now know what language to make the GUI in and what language to use to receive commands on the car side. I can now begin modeling and prototyping in Python
System Integration Considerations	It is important that the language that I choose is lightweight enough to be run on a raspberry Pi and fast enough to support real time control of a car. Also the language must have good, well documented packages for sending information over a network and for building GUIs.
Challenges/Lessons learned	Python is a good language to use for controlling an RC car

Research GUI toolkits

Status	Achieved
Objective	Find out what Python Packages will be useful in building the GUI
My time on this task	2 hours
Support team member(s) time on task	Not applicable
Visual Progress Update	Not applicable
Current Progress	I have found the the de-facto standard toolkit for creating GUIs with Python is Tkinter so I will be using that due to its widespread adoption and its good documentation
Outputs created	I can now begin modeling rough draft GUIs
System Integration Considerations	It is important that the package I choose is lightweight enough to run on a raspberry PI while also supporting the functionality that I will need. I also require something that is popular enough to have good documentation as well as answers to common problems on the web.
Challenges/Lessons learned	It is often best to go with what is popular, even if it is not tailored specifically for your needs because you will be able to find answers to your problems much faster.

Model GUI

Status	Achieved
Objective	Create a rough on paper design draft detailing the layout of the GUI
My time on this task	1 hours
Support team member(s) time on task	Not applicable
Visual Progress Update	 Window Connect Pause 100 50 Motor 1 -50 -100 100 50 0 Motor 2 -50 -100
Current Progress	I have created an updated model of the GUI of the car that reflects the decision to steer the car via tank controls.
Outputs created	I can now begin coding rough draft GUIs
System Integration Considerations	It is important that the model of the GUI is up to date with the design of the car. The finalized decision to use tank controls is reflected in the model. Furthermore, the functionality to stop and start the car is also included in this model which is a feature that the group agreed would be useful to have.
Challenges/Lessons learned	an hour of planning is worth 10 hours of implementation