Project Description: The project is a camera that can be controlled remotely by a web interface to point in any direction and take a picture that will be stored in its gallery and displayed to the user.

Implementation: The hardware used is two standard servos and a small camera controlled by a Raspberry Pi on a boom. These are directed by python code called from the web interface served by a Flask server.

Components:

> Front-end: HTML/CSS/JavaScript pages with some inline code for Flask. The front-end pages display the camera controlling interface and the picture gallery to the user. The camera controlling interface allows the user to give the horizontal and vertical angles the camera will be pointed at, and the gallery shows all previous pictures taken. The interface page shows the latest picture taken, two fields for the angles, two graphical elements the user can click to choose the angle instead of typing it, and a link to the gallery. It uses the jQuery framework. The gallery shows all previous pictures with the most recent picture first and a link to the interface page.
>
> Back-end: A Flask server that serves the front-end pages, giving them the needed data in cases where they use inline python code, and interfaces with the hardware. The most important part of the Flask server is the method called when the front-end posts data to the page to take a picture. This method first has to translate the angles into angles usable by the servos – the servos can only move 180 degrees each. So, if the user wants to go past 180 degrees for the horizontal rotation, the method has to use the angle (angle – 180) for the horizontal angle, use (180 – angle) for the vertical angle, and set the camera to flip the image vertically and horizontally to compensate for it being upside down. It then supplies these given angles to the servos, has the camera take a picture, overlays the direction onto the picture, and puts the picture in the directory with the other pictures so that it can be displayed by the gallery page and the main page later. The library used for the camera is picamera, and the library used for the servos is the Adafruit PWM servo driver.
>
> Hardware: First, a servo was glued to a flat piece of wood pointing upwards, so that another servo sitting on its rotating part could turn left and right. Second, another servo was glued to one of the pieces that fit onto the servos' rotating parts, and put on top of the previous servo, with its rotating part pointing sideways so that it could rotate the camera boom up and down. A pencil with the tip cut flat was used as the boom. The camera was screwed onto one of the pieces meant to fit onto the servos' rotating parts, and the tip of the pencil was fit into the hole. The pencil was glued onto a piece that fits onto the servos, and put onto the second servo.

The challenging aspects were the camera initially repeatedly falling off of the boom, the soldering not working consistently, finding the right documentation on wiring the breakout board with the Pi, and figuring out that the servos weren't pointing at the angles supplied to them because they weren't the right type. These were overcome by using crazy glue, improving the soldering, thorough searching, and getting the right servos after finding a forum post with the same problem we had. Otherwise, everything was smooth with no other roadblocks. Everything regarding making the project work as intended was achieved.

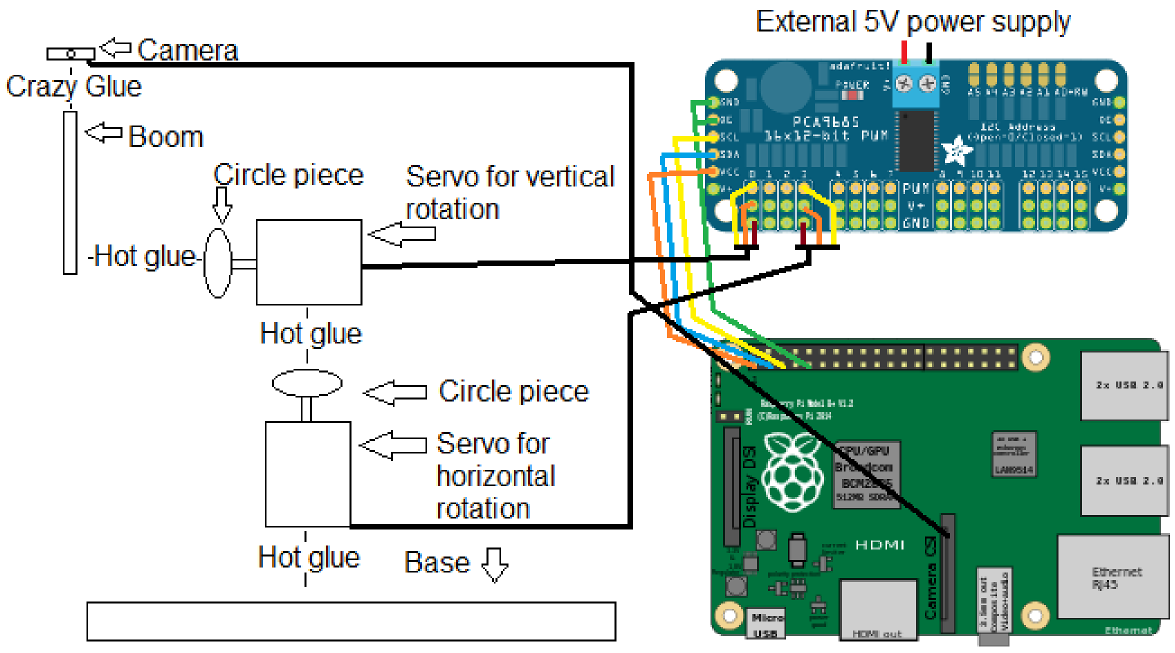
Group members and contributions:

Kurt Trentch: Hardware assembling, soldering, wiring, code for interfacing with the hardware, additions to the web code, changes to Flask server code and integrating hardware controlling code with the Flask server code, testing.

Kar Lap Chan: Flask server code and setting up the Flask environment, HTML/CSS for the web pages, testing.

How to set up:

To recreate the project, you would need a Raspberry Pi, two standard servos with the circular attachments that come with them, the PCA9685 breakout board, a Raspberry Pi camera with a long ribbon cable, something suitable for the boom such as a pencil, something to weigh down the servos with such as a flat block of wood, hot glue, and crazy glue. The project is wired and set up according to the diagram below. The camera interface must be enabled in raspi-config. i2c-bcm2708 must be uncommented in /etc/modprobe.d/raspi-blacklist.conf and python-smbus must be installed. Then, clone https://github.com/KurtTrentch/ELSpring2017/tree/master/code/CameraProject and run project.py on the Pi. You should now be able to use the web interface at localhost:5000 from any browser to take pictures and view the gallery, or from another computer over the local network.



https://upload.wikimedia.org/wikipedia/commons/thumb/c/ca/Raspberry_Pi_B%2B_rev_1.2.svg/300px-Raspberry_Pi_B%2B_rev_1.2.svg.png
https://blog.adafruit.com/wp-content/uploads/2012/08/ff_08162012_3.png

References/Resources:

- http://flask.pocoo.org/
- https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code/tree/legacy/Adafruit_PWM_Servo_Driver
- https://picamera.readthedocs.io/en/release-1.13/
- https://jquery.com/
- https://upload.wikimedia.org/wikipedia/commons/thumb/c/ca/Raspberry_Pi_B%2B_rev_1.2.svg/300px-Raspberry_Pi_B%2B_rev_1.2.svg.png
- https://blog.adafruit.com/wp-content/uploads/2012/08/ff_08162012_3.png