2022 Design and Prototype Semi-Finalist

Graphic User Interface NanoLab

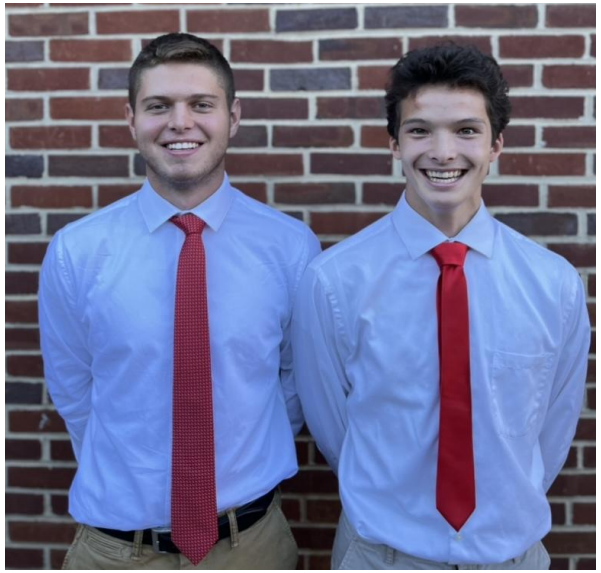
Students: Daniel Vaysman, Jaegon Hibbitts
Teacher: Ray Gerstner
School: Glenelg, Maryland

Students: Bennett Blount
Teacher: Gary Duquette
School: Jackson Hole, Wyoming

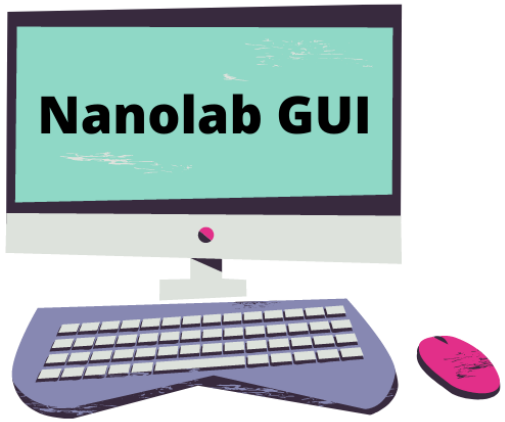
Students: Couper Bowers-S, Bergen Thorne
Teacher: Matt Brown
School: Warren Tech, Colorado

Students: Bryce Roethel, Wyatt Mortimer, Spencer Ellingham, Zachary Evans
Teacher: Vince Stornello, Donna Himmelberg
School: Fairport, New York

About us:



Daniel Vaysman(L) & Jaegon Hibbits(R)



Back side

Nanolab GUI

Glenelg High School
Glenelg, Maryland
Teacher: Mr. Gerstner
Team members: Daniel Vaysman
Jaegon Hibbitts

Testing plan:

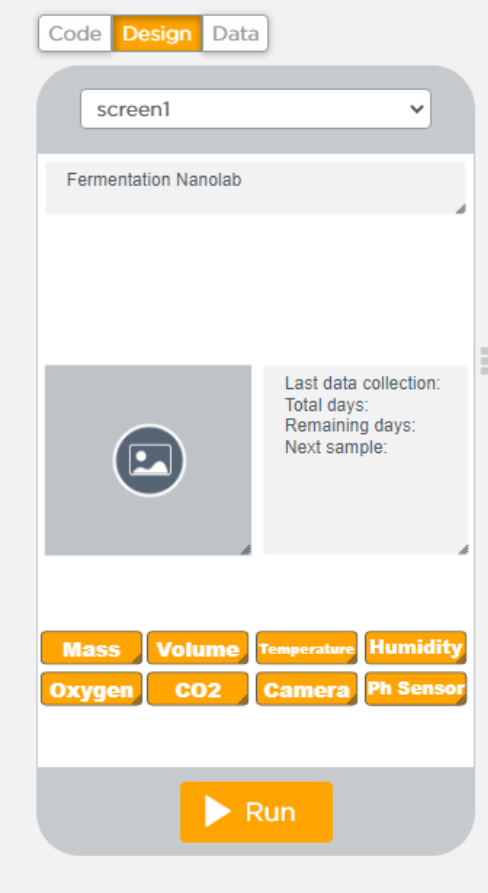
- Data transfer every 30 min - 48 occurrences
- Raspberry Pi
- Connect GUI to external screen
- Sensors (Camera, CO2, heat)
- Developing cross-platform access

Website:

Description:

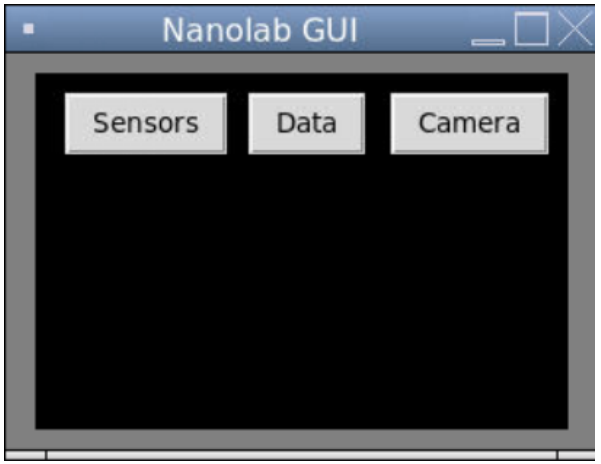
Included is our prototype of both of our GUI's. The first one is the example of the Code.org GUI intended for younger and inexperienced audiences. The second one is more advanced and intended for more advanced audience and is more raw data driven instead of looks.

Younger audiences*



Second GUI:

Advanced audiences*




```
import tkinter as tk
from tkinter import filedialog, Text
from tkinter import *
from PIL import ImageTk,Image
import os

window = tk.Tk()
window.title("Nanolab GUI")
window.geometry("300x200")
window.configure(background='grey')

frame = tk.Frame(window, bg="black")
frame.place(relwidth = 0.9, relheight=0.9, relx=0.05, rely=0.05)

frame1 = tk.Frame(window, bg = 'red')
frame2 = tk.Frame(window, bg = 'blue')
frame3 = tk.Frame(window, bg = 'green')

btn1 = Button(window, text = 'Sensors', command =
lambda:controller.show_frame(frame1))
btn1.place(relx=0.1, rely=0.1)

btn2 = Button(window, text = 'Data', command =
lambda:controller.show_frame(frame2))
btn2.place(relx=0.41, rely=0.1)

btn3 = Button(window, text = 'Camera', command =
lambda:controller.show_frame(frame3))
btn3.place(relx=0.65, rely=0.1)

window.mainloop()

#window.iconbitmap('c:/downloads/NASA logo')
```

Team Member:
Bennett Blount

Coach:
Gary Duquette

Mentor:
Florence Gold

NanoLab Software GUI Team

Jackson Hole High School
1910 High School Rd,
Jackson, WY 83001

Reliability:

The software for running a NanoLab successfully ran for multiple days in a row without human intervention.

The longest test had it running for 52 hours without interruption.

The software successfully logged data from the Sparkfun Qwiic Proximity Sensor and the Sparkfun Qwiic Environmental Combo Sensor during that time period.

Data from the first 30 minutes of the 52-hour test, time is recorded in seconds:

0.002291440964	Environmental Sensor 0 Temperature Celsius
20.61	
0.004241704941	Environmental Sensor 0 Dew Point Celsius
-14.57667754	
0.00862455368	Environmental Sensor 0 Humidity Percent RH
8.427734375	
0.009872913361	Environmental Sensor 0 Pressure Pa
109421.2617	
0.01254725456	Proximity Sensor 0 Proximity Value
1853	
1800.051027	Environmental Sensor 0 Temperature Celsius
23.92	
1800.059843	Environmental Sensor 0 Dew Point Celsius
-13.56734626	
1800.066639	Environmental Sensor 0 Humidity Percent RH
7.248046875	
1800.06989	Environmental Sensor 0 Pressure Pa
95888.30078	
1800.072511	Proximity Sensor 0 Proximity Value
1	

Modularity:

[Project Overview Video]

The software was built to be as modular as possible, with every system being easily interchangeable or upgradable. This will allow for more features to be added later, and could also allow entire parts to be replaced.

The system is broken up into four parts: a looper that runs the experiment and records data over certain time intervals, a sensor interface that defines what can be done with sensors and interfaces with them, the logger which writes data to a CSV file, and the GUI which is where the initial experiment parameters are set before the mission begins.

The software can also accommodate as many sensors as can physically be added.

The looper class that iterates through lists of sensors passed to it by the GUI to set up an unknown number of sensors:

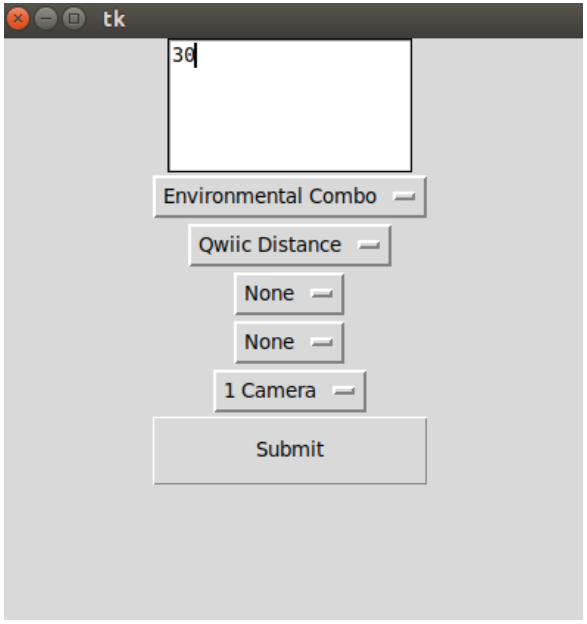
```
class looper:
    environmental = []
    prox = []
    cams = []
    def init(sensor_list, camera_list):
        looper.environmental:List[environmental_sensor] = list()
        looper.prox:List[proximity_sensor] = list()
        looper.cams:List[camera] = camera_list
        for i in sensor_list:
            if i is environmental_sensor:
                looper.environmental.append(i)
            elif i is proximity_sensor:
                looper.prox.append(i)
        for i in looper.environmental:
            i.init()
        for i in looper.prox:
            i.init()
        for i in looper.cams:
            i.init()
        timer.init()
```

Usability:

Usability was a major concern when making the project, but because the team has little experience in designing GUIs and much more experience working with systems like robots, an emphasis was put on that knowing that a GUI could easily be switched out later.

As such, an effort was put into implementing a system that would allow the program to open when the hardware boots up. It initially boots into the GUI where parameters can be set, but the boot after that forgoes the GUI and starts running the experiment like what would need to be done on the ISS.

Current GUI design:



NASA HUNCH:

Hydrofuge Interface

Cooper Bowers-S.
Bergen Thorne

Prototype

Motor

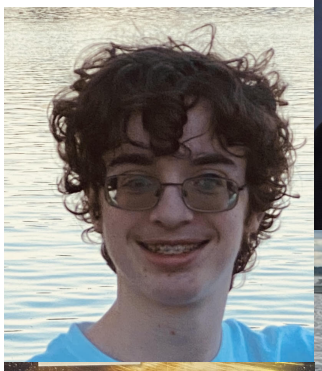
Like the fan, the motor has an issue turning on at certain angles, but this appears to be an issue with the wiring on the side of the motor itself, so we will be soldering the wires for the final iteration.

Fan

The fan will work most of the time. It will just require an initial push in order to start moving on occasion. Certain angles will fix this issue, so the orientation will need to be adjusted in the final hydrofuge accordingly.

Hydrofuge

Fairport High School
Group Members: Bryce Roethel, Wyatt Mortimer, Spencer Ellingham, Zachary Evans
Teachers: Mr. Stornello, Mrs. Himmelberg



NanoLab Software GUI

We are creating a website accessible via a hotspot hosted by the raspberry pi. Allows for all devices with a browser to connect and interface with the nanolab.

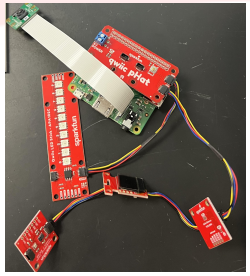
This software will allow people on Earth to control and receive data from small labs in space. The software will be compiled into a website that will be easily accessible and understandable, which could be easily used by kids, but is sophisticated enough for NASA engineers.

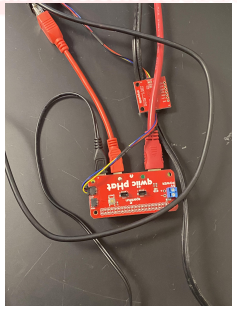
PROGRESS AND CHANGES

We have fully implemented influxDB into our software, which will store all information gathered from sensors.

Sensors

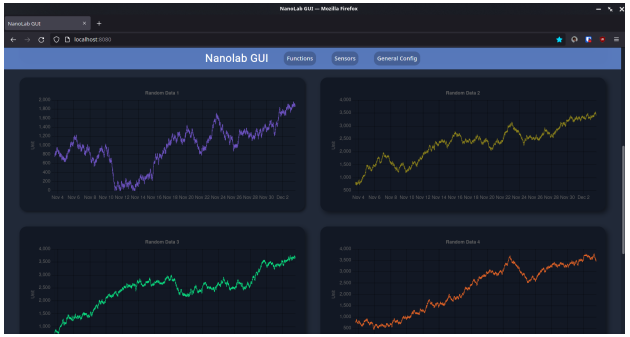
Python code will allow for sensors to be immediately downloaded into the system

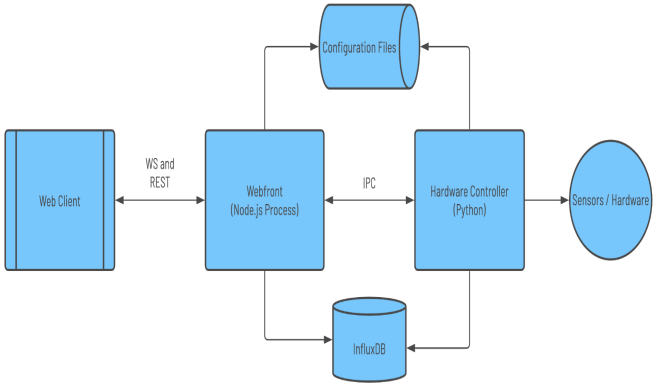




Data

With our wide range of sensors we will be able to collect data and store it on the pi using influxDB.





A flow chart representing what will happen with our collected data and how it will be compiled into a web client