Advanced Automation

Playlist 2022-2023

Unit 1 - Functions and Modular Code (9)
Unit 2 - Managing State (11)
Unit 3 - Collaborative Code (16)
Unit 4 - Control Algorithms and APIs (16)
Unit 5 - Machine Learning (10)
Unit 6 - Measurement and Data Processing (10)

Syllabus Advanced Automation and some formative and summative assessments list

Unit 1: FunctionsAndModularCode

Day 1: Welcome to Advanced Automation! (2022/08/11)

1-1 Why Automation? W1-1

YouTube Video - Although we hear about the negative sides of automation in today's world, learning how to use automation ourselves is essential to seeing how it can be a force for good. In this first video of the automation course this year, I want to share some of the reasons why I am optimistic about what we are going to learn this year.

1-2 Building Automation W1-2

For this task, there are three tools you can explore how to build your own automated solutions to problems you face. You can choose any of them. Here are three videos describing how to build with each one:

- iOS/iPadOS - Shortcuts
- macOS Automator
- IFTTT (IF This Then That)

Your task for the next class is to experiment with any of these tools to get a sense of how they work. Feel free to look for other tutorials out there to help you with any questions that might arise.

You must then build your own two automated tasks with any of the tools above. You should be prepared to explain the background context and the steps in the automation you build. Submit a single screenshot for each program in the assignment below. W1-3

1-3 Documenting and Decomposition W1-4

Template for decomposing an automated task

1-4 Assignment Automation (Schoology) W1-3

Submit a single screenshot for each program (two automated tasks)

Day 2: Introduction to Functions (2022/08/15)

2-1 Introduction to Functions W2-1

YouTube Explanation - Last week we started looking at ways to break tasks down into individual steps. This video formalizes this using the programming concept of a function. You'll learn some vocabulary of functions and ways to describe them using inputs, outputs, and how they relate to each other through algorithms.

2-2 Functions Case Study: Gmail Auto-reply W2-2

YouTube Video - Using what I showed you in the last video, we take a look at code for my Gmail Auto-reply script and analyze functions for inputs, outputs, and algorithms. I show a few examples and leave the rest for you to do with your team. Along the way, we learn a bit about arrays, the logical OR operator, and some other elements of Javascript.

Template for Analysis - and assignment in Schoology W2-2

Day 3: Shock Detector Case Study (2022/08/17)

3-1 Shock Detector Case Study W2-3

Video 1 and Video 2
API Description
Starter code for students

3-2 Accounts on Github and Replit W2-5

Accounts for Collaboration in Advanced Automation

Day 4: Mastery Check - Sketch of Functions (2022/08/19)

4-1 Function Sketch Mastery Check W2-6

Following the style of what I did in the first video of breaking down the Shock Detector task, choose one of the problems mentioned in class and break it into a series of functions that each achieve a specific sub-task.
For each function, identify the inputs and outputs to the extent that you can.

Template for students to copy and fill in Standard Documentation D W2-6

Day 5: Making Code Modular (2022/08/23)

5-1 Making Code Modular W3-1

In this skill-building activity, you will follow along with the video below to convert a single program into modular code organized into functions
Module Maker - Starter Code
YouTube Video Tutorial

Day 6: Module Maker - Creating modular code yourself! (2022/08/25)

6-1 Module maker - create code yourself W3-2 standard 1 and 3

Students create modular code and submit it. The code and description will be graded in the standards 1 and 3.

Task Description and Rubrics for Standards 1 and 3

Mastery Check Unit 1 - Module Maker with micro:bit - Description and Rubric

Day 7: Drink Machine Part I (2022/08/29)

7-1 Drink Machine Project W4-2 and W4-2

- Project Description and Rubric for Standard 4
- Project 2 Starter Code
- Video Introduction (11:10)

Review of global and local variables in https://makecode.microbit.org/ Example code: https://makecode.microbit.org/_PMgfmrJg31tw

Day 8: Drink Machine Part II(2022/08/31)

Reflect on the 3 stages when making a cup of coffee:

1. Fill the container `myDrinkMachine.startFillingContainer()`
2. Add grounds `myDrinkMachine.addGrounds()`
3. Heat it up `myDrinkMachine.turnHeaterOn()`

8-3 Drink Machine Level Up videos W4-4

Now that you've played with Drink Machine and have done some things the long way, Mr. Weinberg is showing the better ways than the wait times and further details:

- Video 1 - calculate the time exactly with desmos
- Video 2 - using the input size variable - using the if/then/switch
- Video 3 - wait ... wait ... I'm doing something

Day 9: Drink Machine Testing Software (2022/09/06)

9-1 Drink Machine Testing Software W5-1

Copy the code from the following link into your program. After pressing Button A it should generate a random drink.

- RandomDrinkMachineDrink2022
- RandomDrinkMachineDrink2021

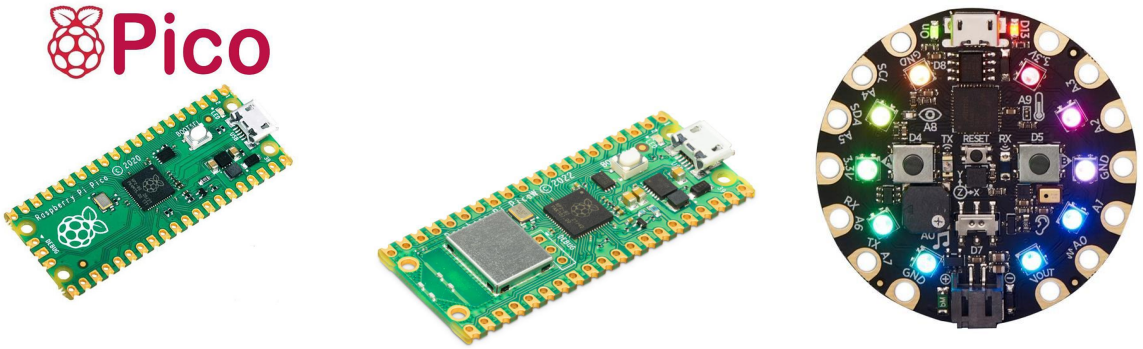
8-2 Drink Machine - Sharing Stage 1 W4-3 standard 4

Students submit the code that generates two of the drinks from the list as assigned from class.

W4-3 Project 2 - Simulator Code

Unit 2: Managing State

AdvancedAutomation - StateProgramming&Abstraction - 21-22



Day 1: Binary Secret Code - Introduction to Abstraction (2022/09/08)

1-1 Binary Secret Code - Introduction to Abstraction W5-3
We are moving on to Standard 6 which is about levels of abstraction. To understand what is meant by that, we're starting with a secret message in binary code

- Binary code as a text file

In this video, Mr. Weinberg introduces the idea of abstraction and teaches you to use Python to decrypt this binary code.

- YouTube Video 20:39
- Link to starter code in Replit

Day 2: Three Boards, Three Flavors (2022/09/12)

2-1 Abstraction vs. Problem Decomposition W5-4
Discussion forum on Schoology

- Link to Assignment
- Example Responses

2-2 Three Boards, Three Flavors W6-1
Watch this short video intro to our case study for today. It shows three different microcontroller boards running three different programs that achieve the same overall goal.

- YouTube Video 1:42

2-3 Case Study Three Boards - Google Doc W6-2
Make a copy of this Google Document to fill out with your group members as you look into the video and explore the code.

- Case Study template
- Example Completed File

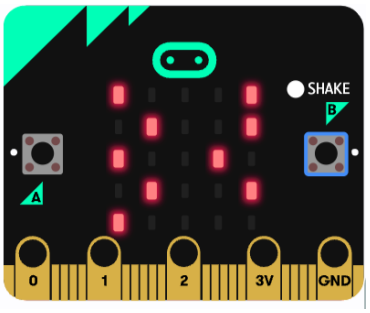
Day 3: Introduction to State Machine Programming - Escape to Summer Case Study (2022/09/14)

3-1 Introduction to State Machine Programming W6-3
Based on what you did last time analyzing the code of a microcontroller board and accelerometer, this video introduces how the concept of a state machine makes this code more modular and easier to debug. The idea is to refactor the code that deals with inputs, outputs, and the logic into separate functions. This abstraction makes it easier to build more complex programs than you might otherwise be able to program.

- YouTube Video (7:35)

3-2 Case Study: Escape to Summer W6-4
The code at the above link was for an escape room activity that I designed for an end of year activity in 2019. There is a lot of code there, and you are not required to understand it all. That said, it is a case study for the concept of a state machine that I introduced in the last video.
Play around with the code to see what it does. As you play, identify the following in your personal Google Doc for W6-5:

- Link to MakeCode code 385 lines
- Hint: Line 60 - `const CORRECT_SEQUENCE = "ABABABBABA"`



3-3 Escape to Summer Case Study - Submit Your Responses W6-5

- Student analysis template

Day 4: Drink Dispenser State Machine (2022/09/16)

4-1 Combination Lock Case Study W7-1
We will use this as a warm-up during class. The video shows a state machine in operation to control a combination lock.
Video Link (0:16)

- What are the states of the system shown in the video?
- What causes transitions between states? (Look very carefully!)
- What is happening in the updateSystem function?

Peardeck Slides

4-2 Drink Dispenser Video W7-2
In this video, I build a complete state machine system from scratch to control a water dispenser that keeps the water within it at a certain temperature range.

- YouTube Video (19:48)

You can use this starter code to follow along with the video.

- Starter Code

4-3 Completed Drink Dispenser W7-3
Students submit a link to their completed Drink Dispenser code. They must use the share button in MakeCode for this to work. Google Doc for submission - Assignment in Schoology W7-3 CODE

Day 5: Building a Physical Drink Dispenser State Machine (2022/09/20)

Today we're going to create the physical part for the drink dispenser. And it is not just uploading the code and connecting the correct wires (a challenging part each year) but we want to have a look at some electrical properties like:

- Voltage V
- Current I
- Resistance R = U/I
- Power P = U × I

Submit a picture or video of your working drink dispenser to this link. W7-3 Drink Dispenser VIDEO

Electricity Worksheet - to fill out by students, link in Schoology (no perfect score 2022-2023)

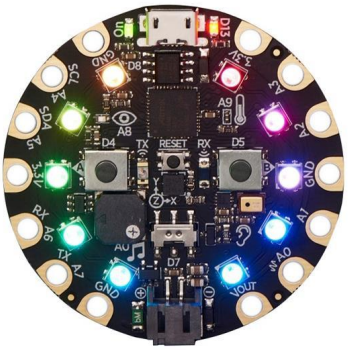
Day 6: Iterative Design (2022/09/22)

6-1 Warmup - Fix it! W7-4
Task Description
Oh no! Nora got onto her dad's computer to do a virtual school assignment. It seems she has messed up some of his state machine code for class. Luckily he has the broken code and some images to help.
In your pairs, you will work on piecing together the broken code. When you are ready to test it out, let Mr. Kreier know.

- W7-4 Fixed code - arc reactor - fan tachometer

You only need to fix the state machine code - leave the rest as it is!

Submissions in replit, then tested on a Circuit Playground Express



6-2 Iterative design for Quarter 1 W7-5
Task Description
YouTube Link (6:21) What is Iterative Design?
Post a document that meets the C+ descriptors for Standard 2 on your website. Showing at least three versions of a single design and with a brief explanation of the project/assignment/task you were doing these iterations for. This will be used for peer feedback and for feedback from me.

- W7-5 Iterative Design Rubric and Task Description

Day 7: Choosing your system (2022/10/04)

7-1 Choosing your System W8-1

Your house is filled with devices that can be modeled and controlled using a state machine. You might even consider systems that aren't in your house, but that you have seen out and about. Your task for this week is to take one of these systems and write software to model its operation using a state machine.

- Task Description:

Mr. Weinberg is going through this process with a microwave. The key thing to keep in mind is that:

- You want clear inputs (sensors, buttons, time) that allows a user to interact with the device
- The system should ideally have different states with clearly different behaviors in each one.
- In the micro:bit simulator, the outputs include lights, sound, print statements, Neopixels, and servos. You want to use these to simulate the possible outputs your system might have.

A refrigerator is always keeping things cold. One job, one mode, one state. That isn't very interesting for the purpose of this project.
A phone runs many different apps and does many different things - that is too complex.
Things to consider:

- electric fan with a knob to select speed
- a flashlight with different blinking modes
- a nap timer that uses lights and sounds to wake you
- a morning alarm that wakes you at one time if the pollution level is low, and a different time if pollution is high
- an air conditioning system
- a height adjustable table
- a rice cooker
- an ice maker
- a desk lamp with dimming section
- a water purification system
- an alarm clock with REM sleep tracking feature

7-2 Defining your State System W8-2

Watch this video to see how Mr. Weinberg thinks through how a state system might model the operation of a microwave.

- YouTube link (11:53)

7-3 Proposal for State Machine W8-3

You (and a partner if you wish) will work to define a state machine for this task.

- State Machine - Planning Worksheet

Day 8: Microwave State Program (2022/10/06)

8-1 Microwave State Program W8-4

Mr. Weinberg built and commented on a version of this program in Python. You should be able to read the comments comparing the content of his example with the program here.

- Link to MakeCode
- Full Description

8-1 Build 1 Dashboard W8-5

Visit this link to post the info for your work on this project.

- State Machine Dashboard 2021-2022 (13)
- State Machine Dashboard 2022-2023 (4)

8-3 Makecode Python State Machine Template W8-6

A student suggested it would be helpful to have some starter code in Python like what I gave you for the Water Dispenser.

- Link to MakeCode

Day 9: Inputs, Outputs, Hardware (2022/10/10)

9-1 Standard 5 - Inputs, Outputs, Hardware W9-1

- Standard 5 Rubric for the State Machine, linked in the State Machine Dashboard

9-2 New State Machine Example - Star Night light W9-2

- Star night light planning Document
- Twitch Stream - Coding the Star Night Light

The edited version (which has all of the idle time removed) is on YouTube:

- AA - W9-2 - State Machine Night Light (Edited Livestream)
- Makecode Micro:bit - Star Night Light (Final Code)

9-3 State Machine Exemplars W10-2

Check out these examples of state machine documents - thanks to Matthew for volunteering his awesome micro:bit powered combination lock.

Exemplary:

- Combination lock

Proficient:

- Food Preservation Alarm

Day 10: Levels of Abstraction (2022/10/12)

10-1 Standard 6 - Levels of Abstraction/ Layers Rubric W10-3

- Standard 6 Rubric copied into documentation for assessment (mastery check)

10-2 Levels and Layers in State Machines W10-4

To help students understand the relationship between different layers in a state machine, I gave this example for them to talk through in groups

- Link to File - 3 slides with layers of abstraction software/hardware/operating system

10-3 Standard 5 Document W10-5

Students submit a Google Slides presentation, screencast, or other document that demonstrates how they meet the descriptors in Standard 5.

Exemplars:

- A+ Example 1
- A+ Example 2

10-4 The iPad Helicopter W10-6

An automation designer thinks at different levels of abstraction in order to make it possible for a complex machine to be easily controlled.

This task gets students to think through how an automated helicopter is an example of abstraction and states.

- Task Description
- Slides

Day 11: Levels of Abstraction (2022/10/21)

11-1 The Crane Game W11-1
This activity gets students thinking about a crane game and abstraction.

- Student document template
- Examples of abstraction for this task

A layer is really any grouping of hardware or software that all serve a similar function. Some examples:

- The user input software layer was what we discussed in the crane game as the functions that corresponded with direct actions of the user. clawButtonPressed, joystickPushedForward
- The user input hardware layer includes the two axes of the joystick and claw buttons
- An output software layer involves a set of functions that all control hardware outputs.
- An output hardware layer would be the set of motors that control the movement of parts of the crane game and the physical lights that blink as outputs.

11-2 Standard 6 Assignment W11-2
Using any of the projects you have done this semester (Module Maker, Drink Machine, State Machine) or code samples that Mr. Weinberg has given you (Escape to Summer, Shock Detector, Three Flavors) create a document that shows your analysis of the code according to the standard 6 rubric.
Some of you have already completed this in your state machine document. To help in keeping feedback on the different standards separate, please submit a link to your state machine document again to this assignment.

Rubric

Exemplars:

- A+ Exemplar 1
- A+ Exemplar 2

2022/10/14 PTC parent teacher conference
2022/10/17 fall break
2022/10/19 fall break
2022/10/25 W10-2 peer assessment Iterative Design
2022/10/27 helping Dan with his program …
2022/10/31 Khang has physical project done, so does Dan

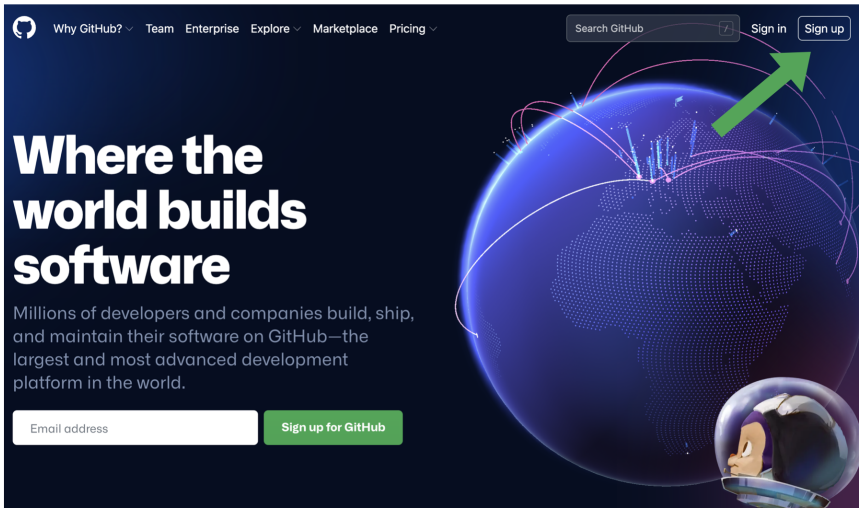
Unit 3: Collaborative Code and Generative Art

Day 1: Generative Art and the Circle K Project (2022/11/04)

1-1 Generative Art and the Circle K Project W12-1
Mr. Weinberg made this video in which he builds Javascript code to generate a pattern similar to the one outside the Circle K across from school.
Follow the tutorial to build your own version of the code. You will use the p5js editor in a web browser to do this. Then make your own variation of the code to generate an image of art that is your own.

- YouTube Link (19:31)

1-2 Creating a Github Account W12-2
Github is a powerful online platform that runs git, a version control program that professional programmers use to collaborate on software projects around the world. You are going to be using Github to save code that you create during this unit and to collaborate on code projects with your classmates.
The first step is to create an account. Go to the Github website and select **Sign Up** at the top right corner.



Use your school ID/username and SSIS email to create an account.

1-3 Making a Repository for a p5js project W12-3
Watch this video (5:15) on creating a repository.
Use this to create a new repository. Create a file and paste in your code from your version of the Circle K project code.
Add Mr. Weinberg as a collaborator on this repository in the settings section of the website. His Github ID is @emwdx. And Mr. Kreier. His ID is @kreier

Day 2: Design Project (2022/11/08)

2-1 Design Project - Collaborative Art (W13-1)
Your design task for this unit is to build generative art code - code that generates images on its own - in a collaboration with your classmates. Along the way, you will be learning to use Github as a tool to collaborate on the development of code.

- Project Description and Rubric Standard 8 (due on day 6)
- Assessment link 2022-2023

2-2 Generative Art Jamboard - Share your ideas W13-2
Look around to find examples of generative art, and find something you are interested in creating yourself using p5js. Remember that your task is to make this yourself. Keep it simple. Simple patterns or rules combined with randomness can lead to really interesting designs. Link your ideas into the Jamboard as you think about what you might make with your classmates.

- Jamboard Link 2021-2022, Jamboard 2022-2023

2-3 Exemplars for Github Demonstrations W13-3
These videos were created by former students to show how to use Github to do various tasks.

- Jaden (16:51)
- Horace (5:44)
- Folder Github Exemplars

Day 3: Branches, Forks, and Pull Requests (2022/11/10)

3-1 Video: Branches, Forks, and Pull Requests W13-4
This video teaches you how to make a branch, fork a repository, and make a pull request to get code that you write into another person's repository.
Your task today is to do the following:

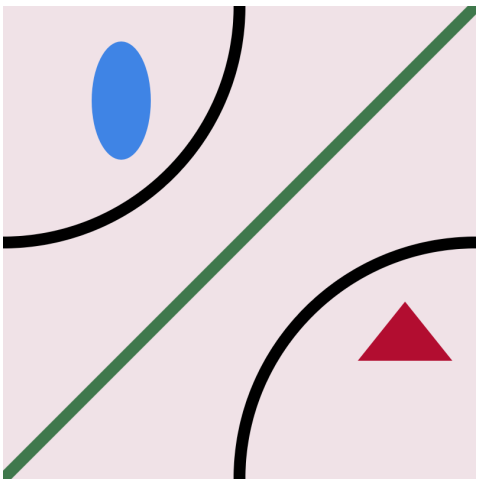
- Fork this repository
- Make a branch named after your spy name. If you don't have a spy name, use this generator.
- On the exit_ticket branch of your fork, edit the branch_fork_pull_exercises.txt and add your name and a link to your new branch.
- Make a pull request on the exit_ticket branch of ssis-aa's repository
    - YouTube Link (7:47)

3-2 Submit pull request W13-5

- Upload a screenshot of your completed pull request on Schoology. Mr. Kreier will be able to check the rest of your work via Github.

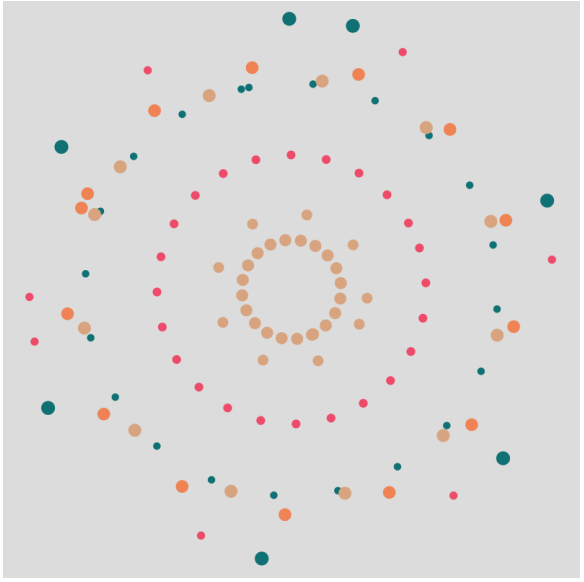
3-3 Make This! W13-6
Use the p5js editor to create the image shown below. Work with your teammates and help each other.

- Link to actual code generating the image



Day 4: Circular Art (2022/11/15)

4-1 Video Playlist: Github and Circular Art W14-2
This video playlist shows you how to update a fork you have made of a repository with changes made in that repository. You then learn about the coordinate systems in p5js and how to use simple programming to generate patterns like this:



Link to Playlist

- Part 1 (2:38)
- Part 2 (3:25)
- Part 3 (2:49)
- Part 4 (13:57)

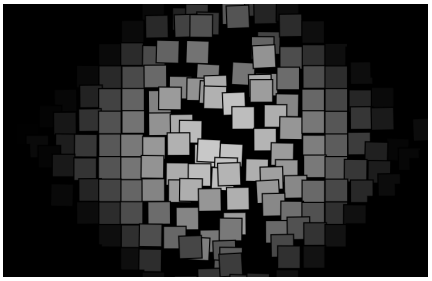
4-2 Circular Pattern - Student Generated
Submit your code or link to your repository for this task

Day 5: Idea Sharing for the Generative Art Project (2022/11/17)

5-1 Generative Art Project Sharing W15-1
Discussion forum on Schoology: This discussion forum will be a place to post your idea for an art project, some sketches or examples of what you are looking to do, as well as a link to the repository for your code development.

You should also comment and make pull requests on each other's repositories during development to collect material you can use for your Github assignment.

- Link to PDF of conversation from Dec 2021
- Discussion forum sharing your project, repository, and collaborators



Day 6 Project Check-In (2022/11/21)

6-1 Project Work and Check-ins - what do you need?
As students work on their projects, making edits on their forks and making pull requests along the way - what are some questions here in the discussion forum? What is the progress?

Day 7 Design Project Submission (2022/11/23)

7-1 Design Project Submission - p5js Art Generator W15-2
Your task is to create an original piece of generative art that contains code written by at least two other students. For the purposes of this assignment, your generative art code should:

- Have at least one use of modular code
- Use some element of randomness that varies shape, color, and pattern such that the output of the program is different each time it is run

Link to your Google Slides Presentation: _____
Link to your Github Repository: _____

Rubric for standard 8

Day 8 Singular Collaborative Art Project (2022/11/25)

Whole Class Collab - Can we get one generative art piece that has commits from every person?

Day 9 Work on Collaborative Art Project (2022/11/29)

Whole Class Collab - Can we get one generative art piece that has commits from every person?

Day 10 Work on Collaborative Art Project (2022/12/01)

Students continue to work on their 7-1 Design Project Submission - p5js Art Generator
https://ssis.schoology.com/assignments/6427783320/info
https://github.com/ssis-aa/generative-art-groups

Day 11 Design Project Submission (2022/12/05)

7-1 Design Project Submission - p5js Art Generator W15-2
Your task is to create an original piece of generative art that contains code written by at least two other students. For the purposes of this assignment, your generative art code should:

- Have at least one use of modular code
- Use some element of randomness that varies shape, color, and pattern such that the output of the program is different each time it is run

Link to your Google Slides Presentation: _____
Link to your Github Repository: _____

Rubric for standard 8

Day 12 Singular Collaborative Art Project (2022/12/07)

Whole Class Collab - Can we get one generative art piece that has commits from every person?

Day 13 Singular Collaborative Art Project (2022/12/09)

Exam week - only 40 minutes, preparation for next exam (120 minutes)

Day 14 Singular Collaborative Art Project (2022/12/13)

Exam week - full 120 minutes to update the project
https://github.com/ssis-aa/generative-art-collaboration2022

Assignments in the art.js (since December 5th):

```
// William: translation of geometric shapes in the right
//          part of the image

// Dan: adding transparency to the right part

// Khang: using noise for the left part of the image

// Dongjae: image manipulation of the whole created image

// ************************************************************
// Image manipulation 2022-12-09
// mk rotate the image and shuffle it in a mosaic
rotate_canvas();
mosaik();

// Dan

// Dongjae

// Khang

// William
```

Day 15 Singular Collaborative Art Project (2022/12/19)

Review of Assignments in the first semester: Assessment List AA 2022-2022 students
Finish the inclusion of filters and creation parts from each student into the 2022-12-19 branch of our repository: https://github.com/ssis-aa/generative-art-collaboration2022/tree/2022-12-19

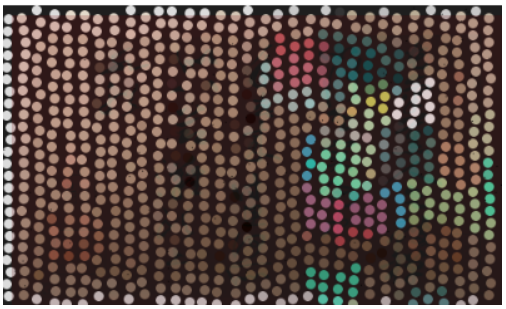
Final version v0.7 where student parts successively added to the image:
https://editor.p5js.org/mkreier/sketches/vy0dgv6Cp
https://github.com/ssis-aa/generative-art-collaboration2022
With no preloaded image and flexible dimensions and randomized outcome.

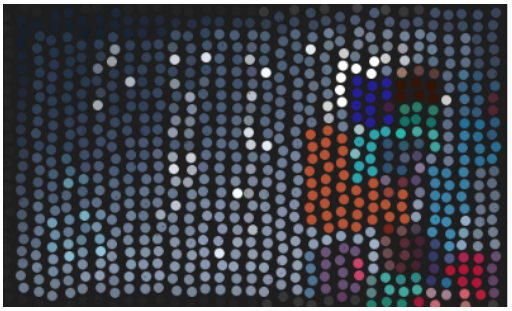
Day 16 Explanation of Winter Break Assignment (2022/12/21)

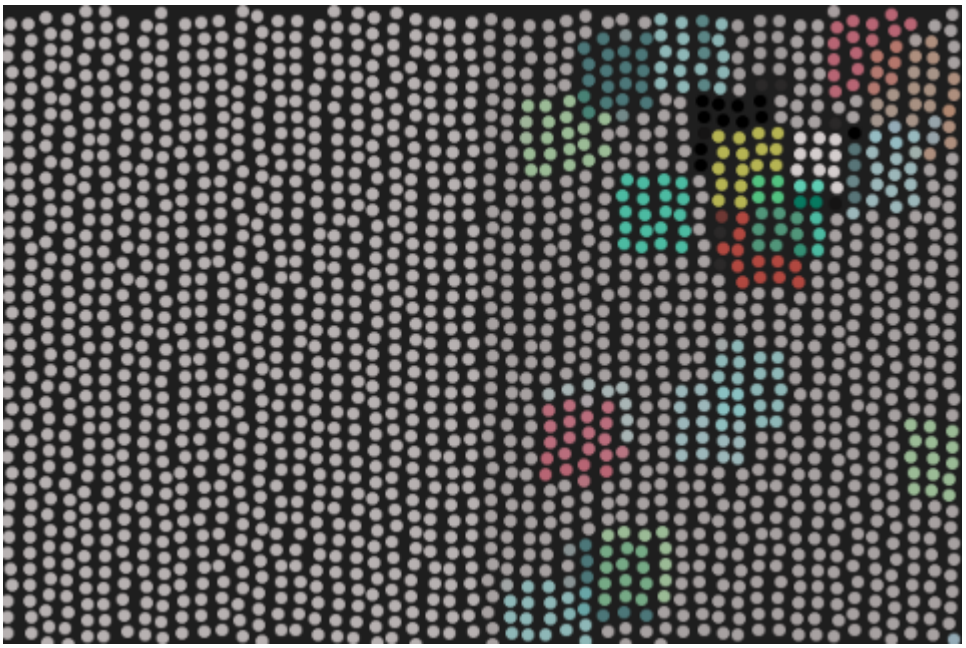
Here is the link

Must do	Should do	Could do	Would be nice
Program your Circuitplayground Express in Circuitpython to use all 10 Neopixels, the microphone, loudspeaker, thermometer, gyroscope, light sensor, button A and B. The Neopixels should produce an output to reflect the input data. Document your programs in github, including your progress. https://github.com/ssis-aa/circuitplayground-winter break	Update your portfolio page. For inspiration look here, here or here	Use the IR sender and receiver for communication, for example learning the IR command from another remote and use it to control an A/C or TV.	Create two unique examples of generative art in p5js.org to share with us.

Result of Collaborative Art (with animated generation: https://editor.p5js.org/mkreier/sketches/vy0dgv6Cp):



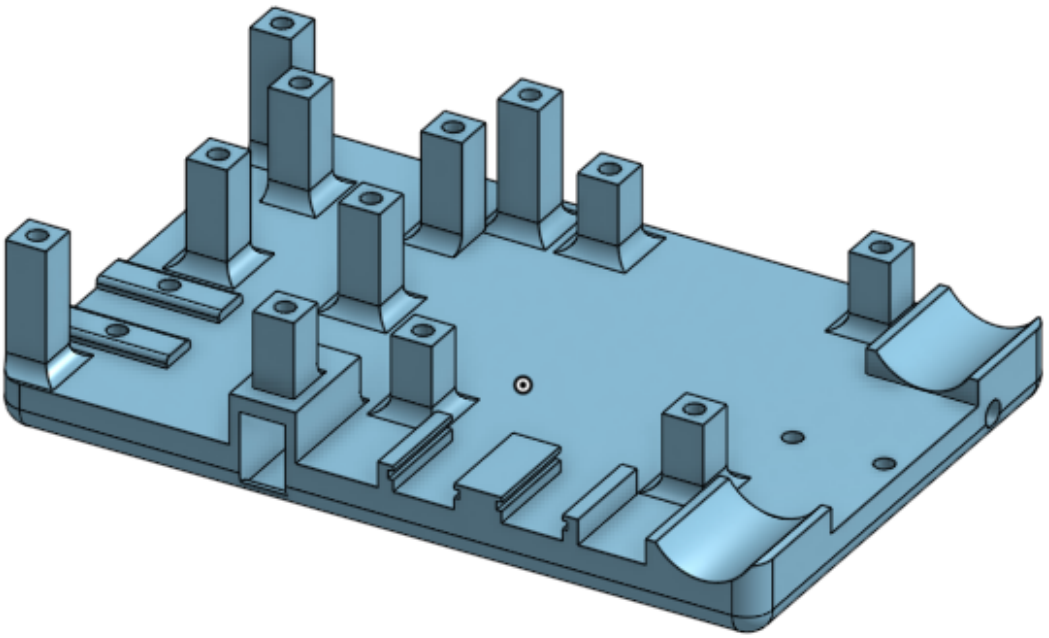




Unit 4: Control Algorithms and APIs [unit plan]

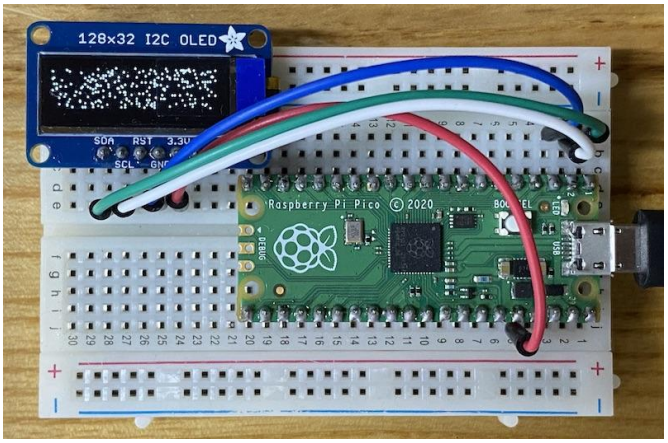
Day 1: Introduction Control Systems Board (2023/02/01)

Prepare the 3D printed carrier board for the raspberry pico 2040 on the RVR Sphero
Insert the M3 screw holders into the base https://github.com/kreier/rp2040/tree/main/base



Day 2: Carrier board rp2040 - physical and electrical connection (2023/02/03)

Mount the PCB holder for the rp2040 on the carrier board
Install the USB power supply with a 18650 battery
Review on voltages (USB: 5 Volt) and the importance of polarity



Day 3: The I2C Bus (2023/02/07)

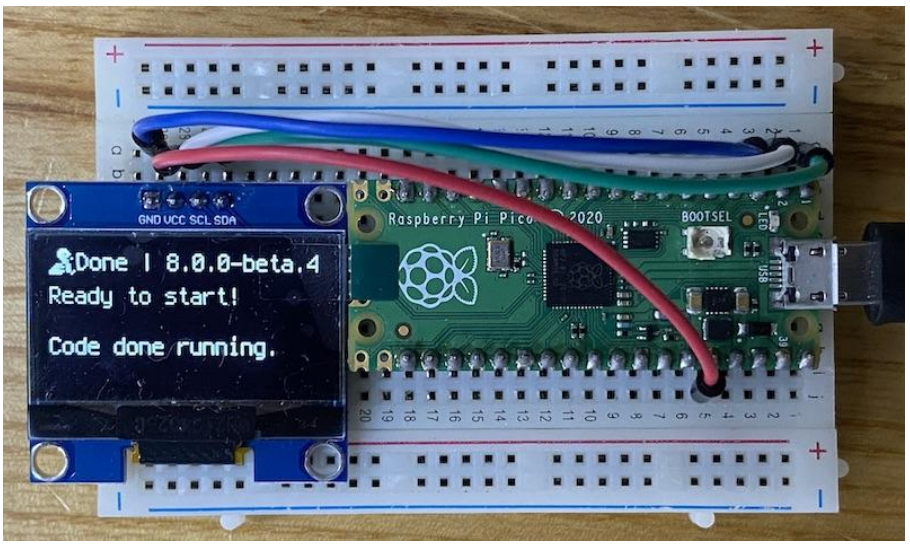
Install the carrier board of the **Raspberry Pico 2040** on the Sphero RVR
Connect the OLED display to the **i2c** bus, see https://github.com/ssis-aa/rvr2023
Update the software of the rp2040 to **Circuitpython 8.0.0** https://circuitpython.org/downloads
Find the correct address **3C** for the display with the i2c_scanner:
https://github.com/ssis-aa/rvr2023/blob/main/circuitpython/apps/i2c_scanner.py

TFT	Pico	Description
VCC	VSYS	Power Input
GND	GND	GND
A	GP15	User button A -- left button (select)
B	GP17	User button B -- right button (confirm)
SDA	GP0	i2c data line for OLED and external sensors
SCL	GP1	i2c clock for OLED and external sensors

Troubleshooting: pullup-resistor not found on GP0, no attribute IO7 on board in line 14

Day 4: Activate the OLED display (2023/02/09)

Get the OLED display to work (it's a 1.3 inch 128x64 display with a **SH1106** controller)
Connect the wires for the two input buttons to **GP15** and **GP17**
Write code to use the buttons as input



Homework:
Write a circuit python program that uses both buttons on GP15 and GP17
Put the code on https://github.com/ssis-aa/rvr2023

Day 5: Control Algorithms (2023/02/13)

5-1 Introduction to Control Algorithms

- There is a subtle difference in what we are exploring today from what we did last semester. Rather than thinking about what output we want for a given input, we think about the reverse: how do we use an output to move the value of an input to where we want it to be?
- This video takes you through some of the ideas of control algorithms and introduces two simulations you will use to explore the concept.
- YouTube Link - Feedback Control Loop

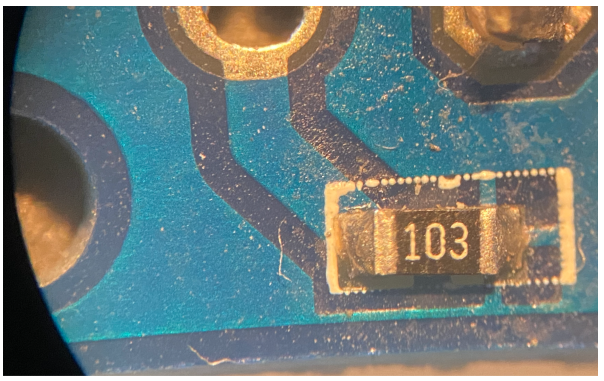
5-2 Control Simulations

- Visit these two links to experiment with the control problems shown in video 1.1.
  - Ball heater simulation - click the ball to turn on the heater
  - Motorized Arm Simulation - change the motor power using the slider
- Make a copy of each simulation and think through how you might automate the following control tasks. Once you have an idea for doing so, try writing code that implements your ideas.
- Keep the temperature of the ball at 40 degree celsius by turning the heater on and off.
- Rotate the arm automatically to specific angles (30, 45, 60, 90, 120, 135, and 150 degrees)
- Write your code in the place labeled within the code. You should not have to change any of the other code in the simulation.

Day 6: Digital states and Pulldown Resistors (2023/02/15)

Write a circuit python program that uses both buttons on GP15 and GP17
Put the code on https://github.com/ssis-aa/rvr2023

Wire the rp2040, the OLED display and the two buttons to get the menu selection code running. You need the **`boot.py`** to initiate the display. After that the **`code.py`** is executed.Put the programs to be selectable into the **`/menu`** folder.



Review:

LOW - zero - is **GND (0) or zero Volt**

HIGH - one - is **VCC (1)** with VCC = 3.3 Volt

Investigation of voltage input and translation to digital values for **GND (0)** and **VCC (1)** with VCC = 3.3 Volt and the build-it pull-down resistor in our buttons. The SMD resistors are labeled in AAB with numbers, representing a resistance of AA * 10^B Ohm. Our label 103 therefore stands for 10 * 10^3 = 10000 Ohm. We confirmed it with a multimeter. No need to activate it in code with `switch.pull = digitalio.Pull.DOWN`

Day 7: Repositories for Control Systems (2023/02/17)

Activate the code for the Menu selection, download the libraries for Adafruit Debouncer from the Circuit Python Bundle 8.0.0: https://circuitpython.org/libraries

7-1 Control Systems Repositories

- Create a repository for your control system code in the unit. Make new files for the code you write to control the temperature of the ball and the motor arm.
    - Dongjae https://github.com/dongdongthedingdong/Motor-Arm-Algorithm
    - William https://github.com/IsNotAvaliable/ControlAlgorithm
    - Dan https://github.com/DanDC25/Control-Algorithm
    - Khang https://github.com/khangpham24/AA-Control-Simulations/tree/main

Day 8: The PID controller (2023/02/21)

8-1 Applying Control Algorithms

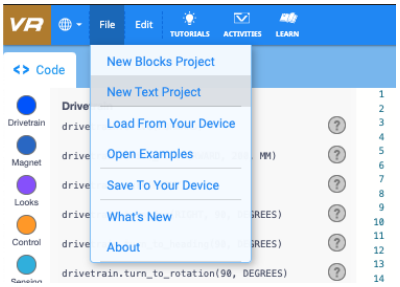
- The video below shows how to use a few control algorithms in driving a Sphero RVR to a distance of 100 centimeters on the ground. You will understand the basics of the following:
    - on/off control
    - on/off control with tolerance
    - proportional control
    - proportional-integral control (PI)
- (12:50) Link to Video (YouTube)

Homework:

- Write a program for on/off, on/off with tolerance and proportional control for the motorized arm. Read from line 30 and put your code from line 48 on. Copy your three programs into your github repository
- Install the Sphero RVR software and create an empty text (javascript) file.
- Additional: write a program with the 3 algorithms in Sphero RVR for 100 cm
- Additional: write a python program for VEX V5 and the motor on port 1 to control the angle

Day 9: VEX VR Control Systems Task (2023/02/23)

9-1 VEX VR Control Systems Task



- Mr. Kreier has given you access to some control system code for a VEXcode VR robot.
- Go to http://vr.vex.com to get to the VEXcode VR platform. Your login code for the class is **MJVWYR**

- Create a new text project using the menu options as shown at right.
- In a separate tab, look at Mr. Weinberg's VEXcode VR control system examples repository linked here. Update 2023: github.com/ssis-aa
- Fork the second repository into your Github account.
- Between the four examples, you will find code for the following:
    - Simple on/off control for position
    - on/off control with tolerance
    - Proportional control code for a specific position (y = 0)
    - Proportional control function with an input parameter for distance and duration.
- Each function given has an expected duration as a parameter. This lets you set a maximum time you want the robot to have to complete the command before moving on.
- Complete any two of the tasks below to earn a proficient level on this task.
    - Task 1.1 (Beginner): Use the task 1 template to make your robot move to the positions shown in this video. Build one program that uses an on/off algorithm for all three steps.
    - Task 1.2 (Beginner): Repeat task 1 with proportional control algorithms to make the movements smooth as shown in the video.
    - Task 2 (Intermediate): The task 2 template code generates a random position in the grid for the robot to drive to. Use algorithms of your choice to drive to the random position, and then use the distance sensor to drive until the robot is 200 mm (one square length) away from the right edge of the grid.
- Save the tasks in your control systems repository as files with names that follow this convention: vex_vr_task_1_1.py.
- These example videos may help you with this task:
    - (9:16) Task 1 - On/Off Control
    - (10:10) Task 1 - Proportional Control and Tolerance

9-2 Control Algorithms & Motorized Arm

- This video shows you how you can apply on/off, proportional, proportional-integral, and proportional-integral-derivative control algorithms to control the motorized arm. You should follow along with the video on a copy of the motorized arm simulation from last week.
    - Motorized Arm Simulation (original version from Mr. Weinberg)
    - Video: (14:10) AA - 2.1 - Control Algorithms and the Motorized Arm
- The above video shows 5 different control systems to control the motorized arm:
    - AA - 2.1 - Control Algorithms and the Motorized Arm
    - 0:00 on-off
    - 3:22 on-off with tolerance
    - 4:45 proportional
    - 7:00 proportional-integral
    - 9:45 proportional-integral-derivative

Homework:

- Complete any two of the tasks below to earn a proficient level on this task.
    - Task 1.1 (Beginner): Use the task 1 template to make your robot move to the positions shown in this video. Build one program that uses an on/off algorithm for all three steps.
    - Task 1.2 (Beginner): Repeat task 1 with proportional control algorithms to make the movements smooth as shown in the video.
    - Task 2 (Intermediate): The task 2 template code generates a random position in the grid for the robot to drive to. Use algorithms of your choice to drive to the random position, and then use the distance sensor to drive until the robot is 200 mm (one square length) away from the right edge of the grid.

Start `(-900, -900)` and move to `(0, -900)` and to `(0, 0)` and further to `(400, 400)`

Day 10: Water Flow Controller and Landing a Rocket (2023/02/27)

10.1 Water Flow Controller

This simulation is for a system that controls the flow of water through a valve.

- You send a signal to the valve using the command **setSignal** for the controller. This is a number from 0 - 255.
- You can get the current flow rate of the sensor using the **getFlowRate** command. Water flow is a tricky thing to control, so this sensor is quite noisy.

The code is linked here: https://repl.it/@emwdx/WaterValveSimulation

These are your tasks:

- Fork the code from Mr. Weinberg's repository into your own version on replit.com.
- Experiment with the code to find the maximum and minimum flow rates for water. The unit of flow rate is liters/second.
- Write code that can send an appropriate signal to the valve to make the flow rate exactly 4.5 liters per second.
- Sketch and implement an algorithm that can adjust the output signal for any flow rate within the range of the valve.
- To test this, Mr. Kreier will tell you a flow rate set-point he wants you to hit. You need to change only the set-point in your code and run your code to show that your system can achieve this value.

(7:59) Link to Video Help (YouTube)

Alternative: Instead of python you can write your code in JavaScript in p5js.org from line 30

10.2 - Land your Spacecraft!

- In this final example of control algorithms doing things, you will be landing a rocket autonomously as shown in the video above.
- Here is the link to the code: https://repl.it/@emwdx/LandingRocket
- This is the link to the visualization on Desmos: https://www.desmos.com/calculator/ijsa8r8upr

- Watch this SpaceX Video to get some hints on how the experts do this. (Speed ahead to 4:00 to see the final landing burn. What do you observe about position/velocity/acceleration? What are they doing to control the rocket as it lands?)
    - The rocket in the simulation starts from 300 meters above the ground with a velocity of 10 m/s. If you do nothing, the rocket plummets to the ground and crashes!
    - The myRocket.setThrust(signal) function is what you will be controlling. It is the only thing you will be changing as the rocket lands. Maximum landing speed is 1.5 m/s.
    - The rocketData list in the template has time, position, velocity, and acceleration. You can paste these into the Desmos visualizer to see a graph of height vs. time, velocity vs. time, and acceleration vs. time.
    - You should experiment with constant values of setThrust to understand the physics of the problem. What happens if the thrust is too little? Too much?
    - You might consider having different control algorithms at different heights or for different velocities of the rocket. One algorithm may not bring the rocket down in 30 seconds of simulation, or without crashing.
- This animation is how you can paste the data into Desmos.

Day 11: APIs and Libraries (2023/03/01)

11.1 - Course Standards for Algorithms and Libraries/API

- The document below links to a description of what you are expected to know and be able to do for this course in standards 9 and 10.
- Rubric for Standards 9 and 10

11.2 - Introduction to Libraries and API

- Watch this video about libraries and APIs. You have already used libraries and APIs, but this video will give you some details on how you've actually done so.
- (11:09) Link to Video (YouTube)

11.3 - Ultrasonic Sensor API

- The link below describes the API for a sensor we have in the IDEA center. This is a common sensor that uses pulses of ultrasonic sound waves bounced off of objects to determine the distance to them.
    - adafruit_hcsr04 — Adafruit HCSR04 Library 1.0 documentation (we use the 3.3V variant US-100)

```
import time
import board

import adafruit_hcsr04

sonar = adafruit_hcsr04.HCSR04(trigger_pin=board.D2, echo_pin=board.D3)


while True:
    try:
        print((sonar.distance,))
    except RuntimeError:
        print("Retrying!")
        pass
    time.sleep(0.1)
```

- There are two functions in this library that you can access via the API. One creates a Python object for the sensor using the input pins it is connected to. The other is a function on the sonar object that returns the distance read by the sensor.

- For the example code at the link:
- Modify this code that will configure a sonar sensor connected with the trigger pin on GP4 and the echo pin on GP5.
- Save this into a new file in your control systems repository called 'ultrasonic' and put it into the menu folder of your CIRCUITPYTHON of the rp2040 as sonar.py. Then test it out. Check that you have the latest code from https://github.com/ssis-aa/rvr2023 and the circuitpython subfolder.

11.4 - Orientation Sensor

- The BNO055 sensor reads orientation in all three dimensions. This means it can identify how it is tilted forward/backward, left/right, and in which direction it is facing.
- This link goes to the API reference for the sensor.
    - Create a new bno055_reader.py file in your repository.
    - Copy the code from the first example at this link into your file.
    - For each of lines 31 (temperature), 37 (acceleration), 38 (magnetic), and 43 (gravity), note there is code that says sensor.[PROPERTY] that accesses the API of the sensor. Find the corresponding line in the API document that explains the property, and paste the description as a comment under that line.
    - Save the file in your repository.

Day 12: Sphero RVR API (2023/03/01)

12.1 - Sphero RVR API Tasks

- Mr. Weinberg has spent some time adding functions to his version of a Python library for the Sphero RVR rover. This is available in the IDEA center. This means that code written in Python using his library can send commands to the RVR.
- If you have completed 11.3 already, you are ready to try this challenge.
    - Download the sphero_api_files.zip file attached to this page to get the Python files for this task. One of these files, sphero_rvr.py, comes from Mr. Weinberg's circuitpython_rvr repository.
    - Open the sphero_example_ultrasonic.py file. You will see some functions called from the rvr object in the code. Write comments in the file to explain the functions from the APIs being used there.
    - Add a proportional control algorithm to the sphero_example_ultrasonic.py file. Save this file to your control systems repository and tell Mr. Weinberg when it is ready. He will download the code into a Sphero RVR in the IDEA center and show you the result.
    - There is another command in the sphero_rvr library called drive_to_position_si(angle, x, y, speed) which will tell the RVR to drive at a speed in m/s to specific x,y coordinates and rotate to an angle when it gets there. The coordinates are measured in meters.
    - Use the sphero_drive_coordinates_start.py file to learn about this command, and then write a program to drive the Sphero in an equilateral triangle with sides of

length 1.0 meters. Save this file in your repository. Mr. Kreier will set you up with an RVR and a Metro to run your code.

12.2 - CircuitPython Setup

Visit each link and download the files below to a folder on your computer for Circuitpython. To complete task 4.0, you should have:

- Mu editor installed on your computer
- A version of the CircuitPython Firmware (Metro Express M0 board) on your computer
- A directory of CircuitPython libraries saved on your computer
- A link to the CircuitPython essentials bookmarked in your web browser.
- Mu Editor
    - The mu-editor is an excellent way to work with Python on the boards in the IDEA center. The micro:bit uses Micropython and the Metro boards use CircuitPython. You can use IDLE and other Python tools, but there are many reasons to use Mu.
    - Use this link to download: https://github.com/mu-editor/mu/releases/download/1.1.0-beta.7/Mu.Editor.1.1.0b7.dmg
    - Circuitpython Firmware
        - Any board we use in the IDEA center must be running firmware - the operating system layer for the hardware - to properly run the code we write. Each board has different firmware.
        - Visit this link to find the board you are using, and then download the firmware for the board for the latest stable release.
    - CircuitPython libraries
        - To be able to use different sensors and features in Circuitpython, people have written software for different libraries.
        - The library zip file linked below contains all libraries for different components that you might attach to the Metro while running CircuitPython 7.1.x.
        - Click here to download, and then unpack the bundle on your computer.
    - CircuitPython Essentials
        - The CircuitPython Essentials document gives many examples of how to implement different input and output methods with CircuitPython.
        - You can bookmark this page here: CircuitPython Essentials | Adafruit Learning System
        - Or you can bookmark a link to the the PDF file here: CircuitPython Essentials

12.3 - Sample CircuitPython code

- Example code is really useful for knowing how to use the API of a new library.
    1. Create a new folder in your control systems repository called circuitpython_examples.
    2. Move your ultrasonic_reader.py, bno055_reader.py, sphero_example_ultrasonic.py, and sphero_drive_coordinates_start.py files into this new folder. Follow the animation below to learn how to do this.

3. You will find some example code for CircuitPython inputs and outputs at the three links below. Make a new file in your circuitpython_examples folder for each example.
4. Each line that imports a library at the top of the file is importing functions from the library. For each line containing uses of functions from these libraries, write a comment that explains what you think it does in the file, and commit your changes.
    - Digital in/out (digital_in_out.py)
    - Analog In (analog_in.py)
    - Analog Out (analog_out.py)
    - In task 4.2, you will write your own documentation for the APIs you use across these examples, as well as the code you adapted before break for the Sphero and ultrasonic sensor code that (hopefully) is now in the Circuitpython folder of your repository.

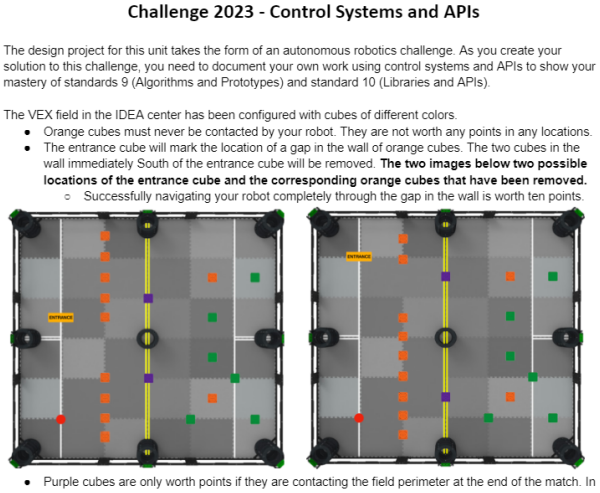
Day 13: Control Challenge 2022-2023 (2023/03/07)

13.1 - Documentation of CircuitPython APIs

- Using the code you wrote last week and your commented versions of the code in your circuitpython_examples folder, add some explanations of the APIs for the functions in the attached file. Some of these have been done for you. The goal is to make yourself a reference that you can use for completing future assignments with Circuitpython, and demonstrate that you understand how to document the API of a library you are looking to use.
- Link to CircuitPython API Documentation Template - look here for the sphero_rvr.py documentation
- Documentation Assignment in Schoology

13.2 Challenge 2022-2023: Control Systems and APIs

Description: Control Systems and API Challenge 2022-2023



Challenge 2023 - Control Systems and APIs

The design project for this unit takes the form of an autonomous robotics challenge. As you create your solution to this challenge, you need to document your own work using control systems and APIs to show your mastery of standards 9 (Algorithms and Prototypes) and standard 10 (Libraries and APIs).

The VEX field in the IDEA center has been configured with cubes of different colors.

- Orange cubes must never be contacted by your robot. They are not worth any points in any locations.
- The entrance cube will mark the location of a gap in the wall of orange cubes. The two cubes in the wall immediately South of the entrance cube will be removed. **The two images below two possible locations of the entrance cube and the corresponding orange cubes that have been removed.**
    - Successfully navigating your robot completely through the gap in the wall is worth ten points.

- Purple cubes are only worth points if they are contacting the field perimeter at the end of the match. In

similar to https://github.com/emwdx/aa_21_22_rvr_challenge

Day 14: Documentation on Design Decisions (2023/03/09)

14.1 Documentation (Standard 1) on Design Decisions

Template to document the Design Decisions, assigned in Schoology

14.2 Documentation (standard 9 and 10)

Start working on your documentation (assigned in Schoology) of APIs and your code development. When using the template you will see the rubric at the end of this document.

Day 15: On the Field: Control Systems and API Challenge (2023/03/15)

The competition on the field:

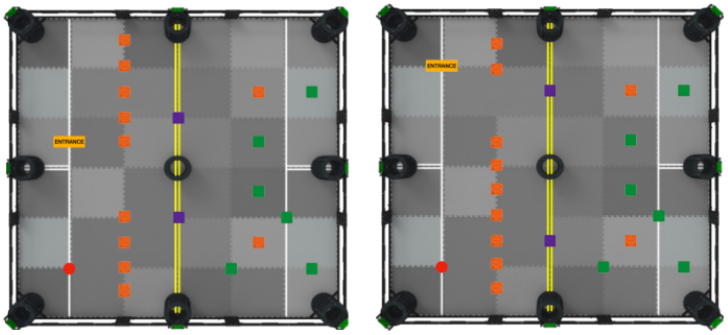
Description: Control Systems and API Challenge 2022-2023

Challenge 2023 - Control Systems and APIs

The design project for this unit takes the form of an autonomous robotics challenge. As you create your solution to this challenge, you need to document your own work using control systems and APIs to show your mastery of standards 9 (Algorithms and Prototypes) and standard 10 (Libraries and APIs).

The VEX field in the IDEA center has been configured with cubes of different colors.

- Orange cubes must never be contacted by your robot. They are not worth any points in any locations.
- The entrance cube will mark the location of a gap in the wall of orange cubes. The two cubes in the wall immediately South of the entrance cube will be removed. **The two images below two possible locations of the entrance cube and the corresponding orange cubes that have been removed.**
  - Successfully navigating your robot completely through the gap in the wall is worth ten points.



- Purple cubes are only worth points if they are contacting the field perimeter at the end of the match. In

Day 16: Final test for the Sphero RVR challenge on the field (2023/05/15)

Final challenge (Control Systems and API Challenge 2022-2023) with results:

- William: 19 + 5 = 24 points
- Dan: 19 + 8 = 27 points
- Dongjae: 10 + 0 = 10 points
- Khang: 5 + 0 = 5 points

Unit 5: Machine Learning unit plan

Day 1: Sketch Classifier (2023/03/17)

1.1 Experimenting with Sketch Classifier

- Visit this website to experiment with a trained machine learning algorithm.
- Trained Classifier Link: https://editor.p5js.org/kreier/sketches/a8qydXg7S (v1, v2)
- Play with the program and find what it does well and not so well, emphasis on the not-so-well. (Coding Challenge 158)
- Record a video that summarizes your findings and put it in the next page: 1.2

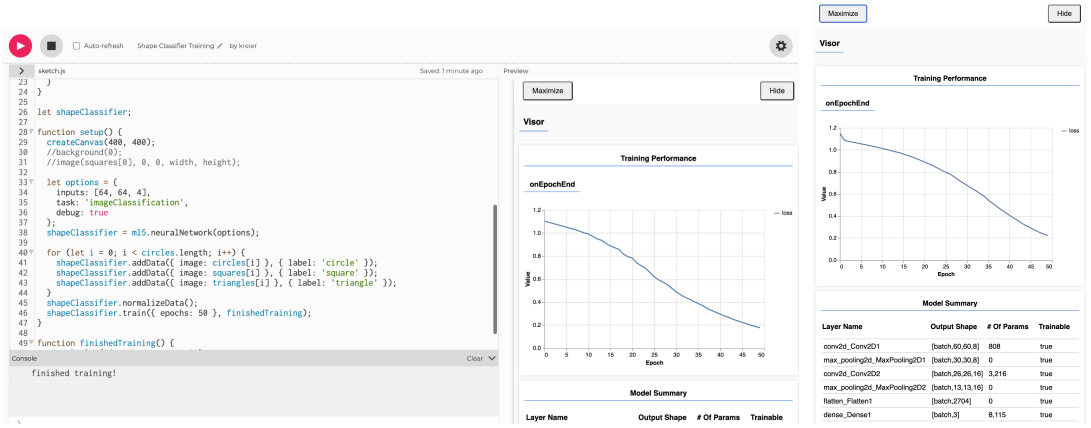
1.2 Sketch Classifier Observations

- After playing with the Sketch Classifier in 1.1, record a video where you show the strengths and the weaknesses of the classifier as it is now. What is it good at? What does it not do well?
- Submit your video at this Flipgrid link: https://flip.com/1e7ad802
- Last years submissions: https://flipgrid.com/4a7a2f4d
- To submit this assignment, take a screenshot of one of the biggest fails you observed while you played with this.

1.3 Training a Classifier

- (10:36) Students watch this video to learn how to use the code.
- Here is the folder of images for you to analyze in the task during class. Why does the classifier work as it does? How might you improve the data set to make it work better?
- **Create Dataset for Classifier**
  - Open the link using Google Chrome: This code was used to create the dataset (v1, v2)
  - Change the code based on your ideas for having better performance on triangles.
  - Run the code to make sure that the triangles are better than they were in the sample set.
  - Uncomment lines 31, 38 and 44. These will include the code to save the images of circles, squares, and triangles that appear. Then run the code for yourself.
  - Save all of the downloaded images into a single folder on your computer.
- **Training the Classifier**
  - Open the classifier training example: https://editor.p5js.org/kreier/sketches/nlmY0B2DH (v1) Download this example. You won't be training on p5js website, instead on your local machine. Unpack the content of the downloaded zip-file.
  - Open the terminal, navigate to this downloaded folder and start your python3 webserver using the code `python3 -m http.server 8000`. If this gives an error, you first have to install the xcode command line tools with `xcode-select --install`
  - Open a web browser in incognito mode and navigate to localhost:8000.
  - After training your classifier you should get 3 files: `model_meta.json` `model.json` and `model.weights.bin` . But this one is based on only 30 images in `/data`.

- Output left for the p5js website, output right on your local webserver:

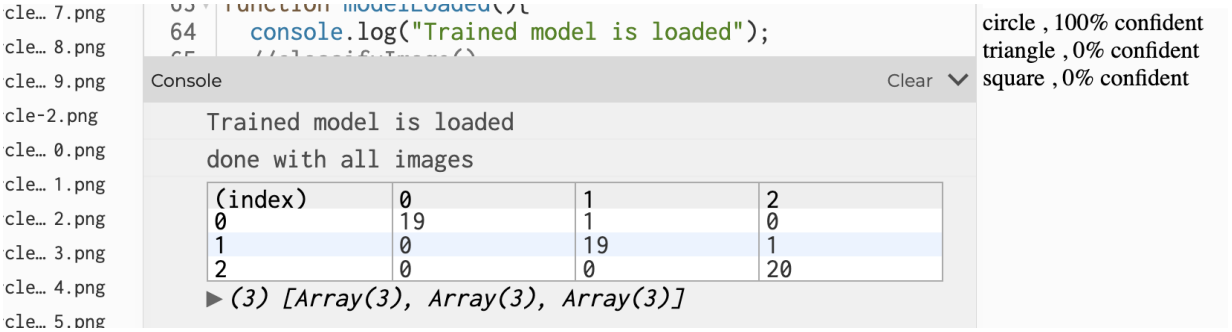


  - Edit the `sketch.js` in the download folder, increase the input image list to 9 in line 18 to 9 and the epochs in line 46 to 60.
  - You have to stop the webserver with ctrl+c and start a new one on another port, for example on `python3 -m http.server 8001` and then open localhost:8001 in the incognito browser
  - Copy your generated 300 images that you created into the training folder inside your downloaded example. Change the reference folder for the images to training in line 20, 21, 22
  - Train your model and download the 3 files model.json, model.weights.bin and model_meta.json
- Follow the instructions in this video to train your classifier on your computer:
- 7:38 AA - Shape Sketch Classifier - Training on your Computer

Day 2: Improving the Classifier (2023/03/21)

2.1 Testing the Classifier

- Open the classifier testing example: https://editor.p5js.org/kreier/sketches/jJXV1ycLp (v1)
- The included weights classify 19 to 20 correct: How well is your model doing?



- Include your testing data, increase the testing size (line 9)
- Download the sketch and test on your own machine
- Use this sketch https://editor.p5js.org/mkreier/sketches/a2YIekzu0 to get the confusion matrix offline.

2.2 Testing the Improved Classifier

- Go to this sketch, duplicate it, and then upload the three files into the model folder
- You have to be sure that the names of the three files match the names above exactly.
- Run the code. See how well the code runs for all three shapes.

- Is the code any different from what you saw in the original classifier?
- Move on to 2.3 to reflect on your improved classifier.

2.3 Sketch Classifier Cycle 2 - reflection

- Now that you have completed a full machine learning cycle, it's time to reflect on your new and improved version.
- Link to Sketch Classifier Reflection template (Schoology assignment)

Day 3: Self Driving Car Activity (2023/03/23)

3.1 Self Driving Car Activity

- In this activity, students learn to train a neural network to drive a simulated car around a road.
- Students watch this video as they work through the document below.
- Link to Video Walkthrough (YouTube)
- Link to Document
- The goal is to acquaint them with some of the key ideas of machine learning:
    - Algorithms need enough data to train so they can find patterns in the data. Driving the car only a short distance does not result in a car that can drive itself. Multiple laps yield better training and better performance of the trained algorithm.
    - Algorithms only learn what is in the training data. The self-driving car in the video fails at the curve in the track because the training data did not include a left turn.
    - Providing a diverse training set with a variety of possible inputs and target outputs is the best way to train a flexible neural network that works in different situations.

3.2 Best Lap Link

- Visit this Flipgrid link to post your best lap around the track: https://flip.com/3c558caf (2022)

Day 4: Iterative Design (2023/03/27)

4.1 Iterative Design throwback

- Link to Edpuzzle Lesson on iterations (8:42)
- As we did last semester, your task is to build an iterative design document for your work this quarter. You must submit evidence of a design as it changes over time for your project.

tandard 2 - Iterative Design	C / C+	B / B+	A / A+
gn and iteratively develop putational artifacts for practical intent, onal expression or to address a etal issue.	You design and iteratively develop computational artifact(s). This could include programming, digital design, CAD/CAM etc. You show the iteration of a single element of your design through multiple versions of it. You explain the purpose of your computational artifact(s). This could be for practical intent, personal expression or to address a societal issue.	In addition to meeting the C+ criteria… You show that iteratively developing a single element of your design resulted in improvements to the computational artifact(s). You explain how developing a single element of your design helped to better meet your original success criteria. You explain the purpose of your computational artifact(s) so that it is easily understandable to other students in class.	In addition to meeting the B+ criteria… You explicitly show the process of improving one element of the computational artifact(s).This could be shown by showing the tests, experiments, prototypes or additional research that was done. You make it clear for the audience to see how the quality of the computational artifact(s) or one element of it, is greatly improved. You use appropriate mediums/tools which help the audience understand the full story of how your computational artifact(s) was developed. You explain the purpose of your computational artifact(s) so that it is easily understandable to a wide audience.

- Iterative design involves designing something. You have to have an idea of what your design should do, and then make an effort to create a version of it that does a simple version of those goals. You hopefully remember from first semester that to meet the C+ descriptors, you need to have done three things:
    - design and iteratively develop a computational artifact (any program or project for this class)
    - show multiple versions/iterations of the same artifact
    - explain the purpose of the artifact - an elevator pitch for the specifics of what you were trying to do
- Your task is to create a document that meets at least the C+ descriptors today by choosing an artifact, showing multiple versions of the artifact, and including an explanation of the purpose of what you design. You will submit a version of this to 4.1.
- This document is the set of examples of iteration we looked at during Semester 1.
- The B+ descriptors involve showing how the different versions lead toward an improvement in the design or to better meet success criteria for whatever you were trying to make. This is due later in the week.

4.2 Iterative Design - Submit your first draft

- Submit a link to your Google Slides, video, or other document showing the iterative design process of a computational artifact from this quarter in this class. Submitting evidence from another class is not permitted for this quarter. Link to Schoology

4.3 Neural Network Building and Training

- So far, you have explored the concept of neural networks as developing a relationship between inputs and outputs. You should have some idea of what the input is for a given problem and what the output is.

Types of algorithms:

Image classifier: image in, label out

Neural network classifier: numbers in, label out

Neural network regression: numbers in, number out

You have done the first two at this point:

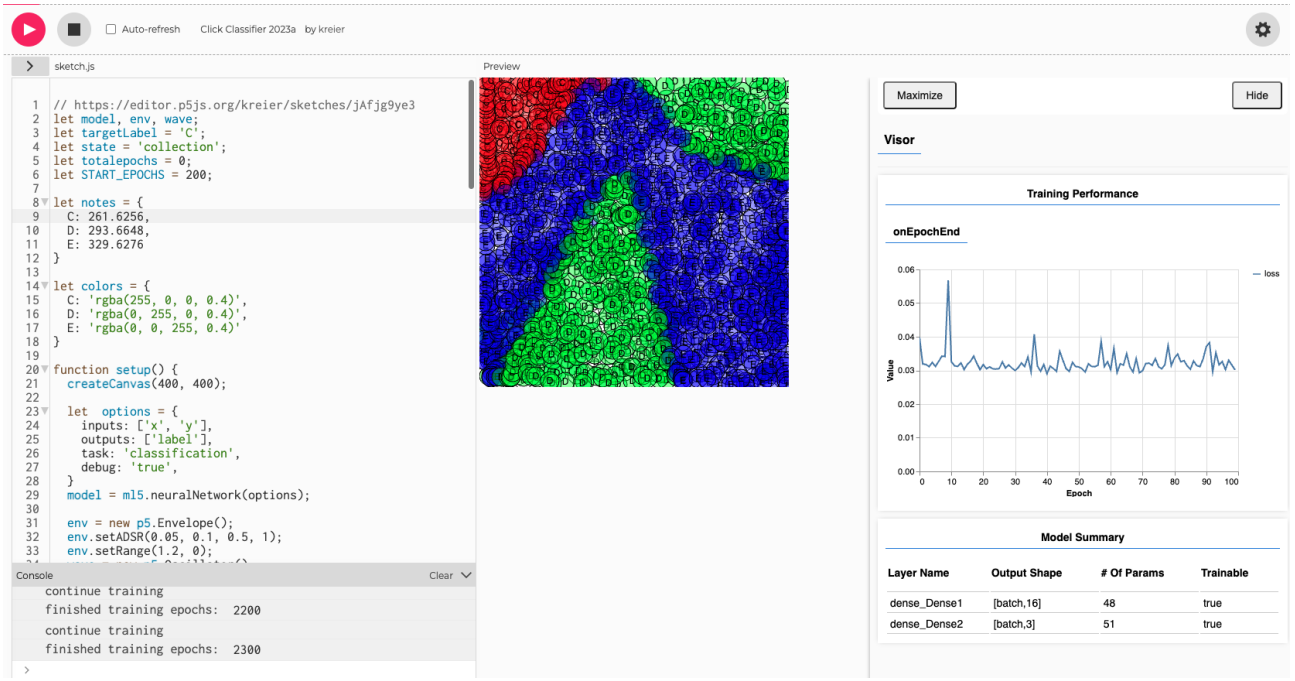
- image in (sketch), label out (circle, square, triangle)
- numbers in (distances for five laser sensors), number out (steering slider value)

4.4 Click Classifier Example

Today you are going to start working through an example of the third type, but you are going to build the software from an almost blank canvas.

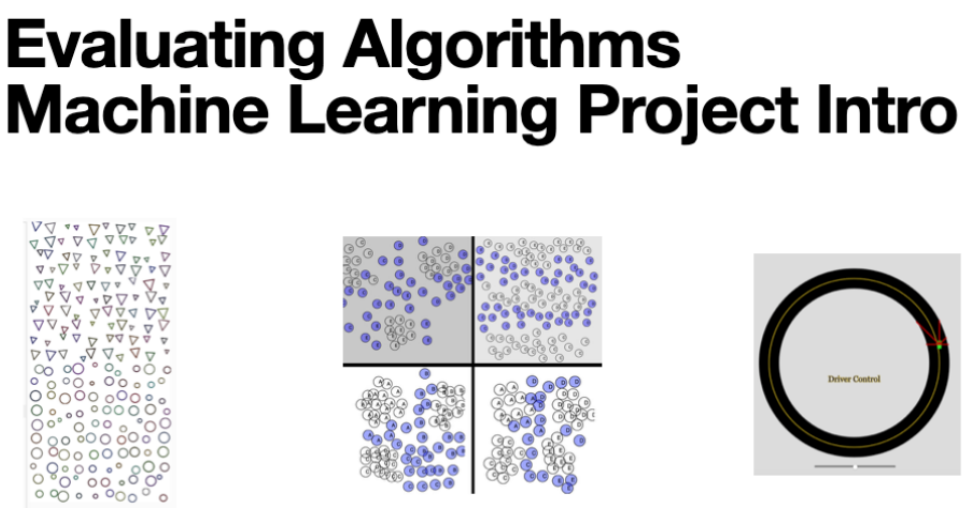
- Make a copy of this sketch (2022). It contains the link to the ml5 library in the index.html.
- Open up this video from Daniel Shiffman. The video starts at the point in the video where I want you to start paying attention. He defines the ideas of inputs, outputs, and what a neural network is actually doing. Expand your sketch along this video to create a click classifier.
- You do not need to add the sound element that he adds at the end. Clicking and classifying is good enough.

- Record a short screencast of you creating data, training your network, and then clicking around to demonstrate that the predictions are working.
- Post your video on flip.com here 2023. (Click classifier example 2022)
- This sketch has color and sound.



Day 5: Evaluating Algorithms and Standards (2023/03/29)

Quick review: What is your score in the assigned Edpuzzle?



5.1 Evaluating Algorithms Slides

- These are the slides we are working through together during the first part of class.
- 5.2 Evaluating and Testing Algorithms Slides presented through Peardeck (2022)

5.2 Training and Testing

- The last case study is on a program Mr. Weinberg wrote to generate faces and classify them as being happy, sad, angry, or some other emotion. This data is being analyzed as a data file in a format called CSV, or comma separated values.
- Link to Video (YouTube)
- You can download the raw data file Mr. Weinberg uses in the video from this page.
- Watch the video to learn how to separate the data into training and testing sets. You should then follow his steps to upload this data to the training sketch, train the network, and download the trained model.
- Link to CSV file in video
- Generator: https://editor.p5js.org/emwdx/sketches/VKobtmrFw
- Training: https://editor.p5js.org/emwdx/sketches/HB3Xe-6al
- Testing: https://editor.p5js.org/emwdx/sketches/ZiJQ81rwj

5.3 Learning Standards

- The project you will complete for this unit involves two learning standards.
- Link to Standards 11-12

Day 6: Machine Learning Project - Objective and Dataset (2023/03/30)

- Click Classifier: Post your video on flip.com here. (Click classifier example 2022)
- Outlook 2023

6.1 Machine Learning Project Files

- Now you get to choose what you will build to show what you have learned about machine learning over the past three weeks. The focus is on taking sample code and modifying it to do what you want to do.
- A great place to start is to look at data sets that are out there. There are many CSV files for all kinds of data in finance, sports, health.
- Kaggle is one place with interesting data sets to use. You can filter based on the type of data set you want to use - CSV files for classification, computer vision, etc.
- Top 23 best public datasets for practicing machine learning [2021]
- 50 Public Datasets for Machine Learning Projects [2022]
- 65+ Best Free Datasets for Machine Learning [2023]

6.2 Machine learning project

- Information about the Project Proposal
- Machine learning project proposal - link in Schoology

Day 7: Machine Learning Project - Algorithm and Training Data (2023/04/04)

7.1 Progress in project

At this stage, you've chosen some things for your project.

Explaining your problem as one that can be solved by a machine learning algorithm.

- You can describe the inputs that a machine learning algorithm can use to control an output.
- You can give examples of sets of inputs and outputs for training your algorithm. For example, in the self driving car example, you had an input of five distance sensors ["leftDistance":32,"leftFrontDistance":36,"frontDistance":192,"rightFrontDistance":36,"rightDistance":26] and an output of a single steering value: ["steering":0].
- You should be able to give an example of this for your specific application. If you can't, you may not have defined this well enough yet.

You have a clear source of data for training your algorithm.

- You have a source of images, a CSV file, or a systematic way of collecting input data that can be fed into a version of one of the example projects.
- You can systematically separate this source of data into reasonable training and testing sets.
- If you are using images, you may want to use Automator to process your images to be the correct size for the sketch classifier. Ask Mr. Kreier about this.
- You may not have a lot of data to start, and that is ok. It is better to have a small data set to use to test that the system is working before loading in thousands of data points.
- If your idea is to predict something in the future, you won't be able to test the accuracy of your data until it happens. A better approach is to use successive years of data (2005-2010) to train an algorithm to predict for 2011 because you can confirm how well it works.

You can identify modifications you need to make to example code to work for your application.

- We trained the sketch classifier using a series of images. If you are using a series of images, you probably want to figure out what you need to do to your data set to make it work with the code for the sketch classifier. Figure out which lines in the code refer to specific parts of the example application, and tweak it to work for a different purpose.
- If you use a CSV file, you need to change the code to work for your data's number of inputs and outputs. This is in the preload function for the faceData classifier.

You have set some success criteria for what you are looking for in the data.

- We decide early on what level of success we are looking for in the data to give objective measures of the algorithm working or not.
- What is reasonable for a human doing this work? We should not necessarily expect the algorithm to be as good as a human, but it should be better than chance.

Your proposal should have this detail already. If it doesn't, talk your idea through with Mr. Kreier to get this level of detail. Your next step is to build a working prototype that trains on a subset of your data.

- Improve the classification of face expressions: https://ssis.schoology.com/page/6062971006
- Modifying exemplar: W11-2 - Modifying Code Exemplar
- Link to the dashboard

Day 8: Machine Learning Project - Training the Model (2023/04/06)

RVR Challenge: Control Systems and API Challenge 2022-2023
Example code now in 3_challenge in the repository https://github.com/ssis-aa/rvr2023

Overview of the Machine Learning projects discussed so far:

Example	Creating data	Training	Testing
Train your own Neural Network	Create, train and test with mouse and notes Keys: 't' train model, 'r' random guess, 'c' continue training, 's' save model		
Shape classifier (2020)	create dataset	training	testing version2
Self driving car video (sketch)	sketch	5 sensors	different tracks
Face expression 2022	generator	training	testing
Face expression 2023			testing

Expanded shape and color classifier

create dataset	training	testing	mouse	webcam

Detector thumbs up or thumbs-down
Train Neural Network with Images

Day 9: Work on Project Unit 5: Standard 2, 11 and 12 (2023/04/26)

Machine Learning Project Template with standard 2, 11 and 12
Project Proposal (Standard 2) link in Schoology for Iterative Design
Project Documentation (Standard 11 and 12) link in Schoology, exemplar for standard 11 and 12

Day 10: Project Unit 5 completion (2023/05/17)

We will look at the projects.
Linked to the portfolios: https://sites.google.com/ssis.edu.vn/automation/home/portfolios

10.1 Machine Learning Mastery Check

- Please submit a link to the draft document you are building for your machine learning project. This should include your evidence of meeting requirements for standards 11 and 12.

10.2 Modifying Code/ Machine Learning Exemplars

- Check out these exemplars from previous years' projects.
    - Identifying Land/Water from Google Maps
    - Classifying Drawn Characters

Day 11: Presentation of Unit 5 projects (2023/05/19)

Machine learning at its best.

- Dan — Determine Unique Spanish Letters
- Dongjae — Hand Gesture Detection
- William — Updated Shape Classification with Star, Rectangle, Heart and Polygon
- Khang — Mushroom identification (with additional data from kaggle)

Day 12: Refine presentations for art party “RESET” (2023/05/23)

During this week the art department is presenting their work for the second semester. The CSE department is included. With 2 shifts we will have 4 sessions for App Development, individual passion projects from Robotics and Engineering (dart board, catapult, chess game, bird house, bubble generator), Advanced Automation (see 4 projects above and a moving armchair https://github.com/ssis-robotics/reset23). The SSIS Robotics club will have some robots from the current game (Spin Up) to be controlled by visitors.

Unit 6: Measurement and Data Processing

Day 1: Collecting Data (2023/04/10)

1.1 Introduction to Data Processing

1.1 Introduction to Data Processing.pdf

The CSV files show the pressure and temperature data that Mr. Kreier shared during class.

- find how many lines of data there are in both the pressure and temperature partial datasets
- find the maximum and minimum pressures measured in the partial data to make sure it works.
- Run the program on the full files to find the maximum and minimum pressures and temperatures for the full data set.

Here is the link to the REPL with the starter code.
Mr. Kreier demonstrated a method to find the maximum and minimum values during class, but you may find this video useful if you weren't able to catch the method during the demonstration.
The CSV files for this task are attached: pressure_temp_csv_files.zip
Homework: link your github repository and https://replit.com/~ results into the Schoology assignment.

Day 2: The Color Wheel (2023/04/12)

Please bring your Circuit Playground Express (Dan and Khang)

Today we are going to review some ideas about state machines and how we might use them to make measurements.
Here are the slides for today: W13-4 - Color Wheel Processing
The link to the REPL with today's code is here:
https://replit.com/@evanweinberg/ColorSensor#main.py
Fork this code. In pairs, work on defining algorithms for identifying the colors and counting transitions with colors. Only one of you should be touching a keyboard - and then switch for the second algorithm!
When you have written both functions and pass the test, upload the result to your repository.

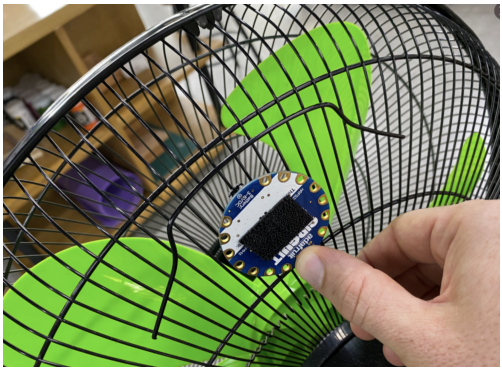
Day 3: Fan Tachometer Revisited (2023/04/14)

Collecting light data with the light sensor (transistor) on the Circuitplayground Express using Circuitpython

3.1 Fan Tachometer Revisited

Way back in Unit 3, I showed you how I built a device to measure the speed of a rotating fan. Today you are going to do some data processing of some CSV files to learn how this works, and then you are going to build the system yourself!

- Here is the website for graphing CSV data.
- This REPL link has a CSV file of thresholded values that are 0 and 1 depending on whether or not a blade is in front of the light sensor. Edit the code so that whenever the light value transitions from 1 to 0, the blade count goes up by 1.
- This second one is similar, but it has a CSV of raw light sensor data that has not been thresholded. You need to add a step that does the threshold step, and then paste your thresholded processing code from the previous step.



Use a light sensor, a Metro board, and the thresholdValues.py file below to get a Metro board reading a light sensor and printing out a bunch of values. Copy and paste this data into a CSV file, add it to your sketch for step 2, and you have your own fan data!

W13-6 - Fan Tachometer Revisited.pdf

- fan_data_student_set.zip
- thresholdValues.py_.zip

3.2 Measuring Pulse Width

Rather than measuring the number of blades that go by, we can process the data to find out how much time the sensor is dark. (fan_processed_threshold_data)

- The light sensor transition is being used to 'trigger' the act of recording the time at which each transition takes place.
- We can record the time at which the pulse begins...
- ...record the time when the pulse ends...
- and subtract the two times to find the width of the pulse in terms of time.

Your task is to do this with the fan tachometer data.

We will follow these slides during class.

1. Fork your code from the fan tachometer
2. Add code that checks for a transition from 0 to 1 and record the time into pulse_start
3. Add code that checks for a transition from 1 to 0, and record the time into pulse_end
4. Calculate the width of the pulse into a list in this format: time, pulse_width
5. Output a CSV called pulse-width-data.csv file for the picket fence data (attached).
6. Repeat this for the fan_processed_threshold_data in your REPL.

W14-1_-_Measuring_Pulse_Width.pdf

picket_fence.csv

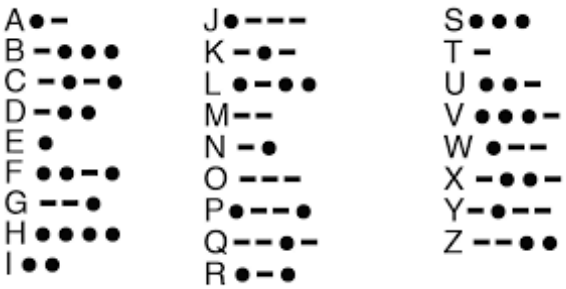
Day 4: Thresholded data and Pulse Width (2023/04/18)

4.1 Morse Code Challenge

Mr. Kreier has a board that is sending morse code via a flashing light. The rate at which he is sending Morse code is Here is some information you might want to know:

- Morse code consists of a dot (.) and a dash (-). Each letter is a combination of dots and dashes. You can Google to find this out.
- Mr. Kreier is pausing three dots worth of time between letters. After a full word is sent, his code waits seven dots worth of time before sending the word again.
- This REPL has a version of the fan blade analyzer code adapted for this task. The only real change to the code is the name change of the input file. The morse_light_raw.csv file contains time and light sensor data for a Circuitpython board that was put next to Mr. Kreier's sending device.
- Your challenge is as follows:
    - 5 points - figure out the length of time in seconds for a dot and a dash
    - 10 points - threshold the light data into zeros and ones.
    - 10 points - write code to automatically measure the length of time in seconds for each light pulse in the data. (This is basically what I asked you to do for the picket fence task)
    - 10 points - write code to turn the widths of each pulse into a 1 or a 3 where a 1 represents a dot and a 3 represents a dash.
    - 10 points - Turn a series of pulses into a Python list that represents a series of dots and dashes. For example, the list for the letter 'L' is [1,3,1,1], because the Morse Code for 'L' is dot-dash-dot-dot.
    - 15 points - Process the entire morse_light_raw.csv file into a list of lists, each one corresponding with a single letter. For example: [[1,3,1,1],[3,3,3],[1,3,3,1],[1,3,3,1],[1,1,1]] is a list that represents the word 'LOOPS'
    - 10 points - use the content of the morse.py to translate each list of pulses into a letter.
    - 5 points - translate the morse code message into a word

For Monday, April 24th: Mr. Hung will help you with the first 4 bullet points (starting score: 35 points). Everything else is up to you! Write the code and submit it to schoology.



4.2 Unit 6 Project Rubric

- For your last project of the course, you will be building a device or software project that processes data to make some measurement.
- This document describes the ideas behind it and includes the rubric Mr. Weinberg will use to assess your skills.
- Unit 6 Project - Measurement and Processing
- Unit 6 Project Description: Measurement and Data Processing
- Link to Schoology for Unit 6 project

4.3 - Sensor Calibration, Smoothing, Rate of Change

- Link to the REPL for today's work
- One more set of exercises today and you'll be ready to measure and process anything...within reason.
- As you've learned, analog data can be noisy. It also doesn't always give you exactly the data you want. You need to process it a bit to deal with both of these challenges.
- The attached files are for an analog distance sensor. One is data you can use to calibrate the sensor, and another is for you to turn into smooth distance vs. time data.
    - The analog_distance_calibration file shows the analog reading at 5, 10, 15, and 20 cm away from the sensor. Use the values in the data and the linear_map function to fill in the function raw_sensor_to_distance function on line 20 in the REPL link.
    - Fix the rate_of_change_calc function on line 30 so that it correctly calculates the rate of change of the distance value.
- analog_distance_raw.csv
- analog_distance_calibration.csv
- Calibration-Smoothing-ROC_-_Final_Graphs.pdf
- W15-1_-_Calibration_Smoothing_Rate.pdf

4.4 Collecting data

- https://github.com/ssis-aa/morse/blob/main/circuitpython/lightsensor_write_csv_10seconds.py
- You also need this boot.py https://github.com/ssis-aa/measurement-and-data-processing/blob/main/circuitpython/boot.py

Day 5: From csv to continuous read (2023/04/20)

5.1 Status Report - Project 6

- Please submit some slides showing your progress on meeting requirements for standards 13 and 14. The focus is on showing the following:
  - What you are attempting to do in this project
  - How you are storing data in a list for processing and why this matters
  - How you are processing data in some way to create an artifact
- These relate to the C descriptors for the rubric. You can find the full rubric below.
- Unit 6 Project Description: Measurement and Data Processing
- Unit 6 Project Description: Measurement and Data Processing

5.2 Semester 2 Iterative Design

- As we have done throughout the rest of the year, you will complete one more iterative design assignment. This can be on anything that has not yet been covered by a previous submission for Standard 2. The rubric is below.

Standard 2 - Iterative Design	C / C+	B / B+	A / A+
Design and iteratively develop computational artifacts for practical intent, personal expression or to address a societal issue.	You design and iteratively develop computational artifact(s). This could include programming, digital design, CAD/CAM etc. You show the iteration of a single element of your design through multiple versions of it. You explain the purpose of your computational artifact(s). This could be for practical intent, personal expression or to address a societal issue.	In addition to meeting the C+ criteria… You show that iteratively developing a single element of your design resulted in improvements to the computational artifact(s). You explain how developing a single element of your design helped to better meet your original success criteria. You explain the purpose of your computational artifact(s) so that it is easily understandable to other students in class.	In addition to meeting the B+ criteria… You explicitly show the process of improving one element of the computational artifact(s).This could be shown by showing the tests, experiments, prototypes or additional research that was done. You make it clear for the audience to see how the quality of the computational artifact(s) or one element of it, is greatly improved. You use appropriate mediums/tools which help the audience understand the full story of how your computational artifact(s) was developed. You explain the purpose of your computational artifact(s) so that it is easily understandable to a wide audience.

- For the purpose of tying up loose ends for the semester, you will submit your iterative design assignment on your portfolio website. This website will serve as your portfolio for future computer science and engineering courses here at SSIS as well as for use in showing off what you have done in your courses.
- To help you to this end, please make the following changes to your website:
    - Add all design reports from the year
    - Add all iterative design assignments, including this final assignment.
- Include answers to the following questions:
    - What are some important skills you've developed over the course of the year?
    - What is something you have made in this course that has made you proud?
    - How might you continue to use what you have learned in other classes you plan to take?

Day 6: Morse Code Project (2023/04/24)

Part 1: Continue the work on the morse code project. Mr. Hung will help you with the first 4 bullet points. That's 35 free bonus points! Everything thereafter has to be solved by yourself and submit your solution in schoology.

Part 2: Try out your code for the RVR Challenge 2023.

Part 3: Start working on the **unit 5 project proposal** (link schoology) and **documentation** (link schoology).

Day 7: Individual project (2023/04/28)

Unit 6 Project Description: Measurement and Data Processing (including Standard 13 and 14)

Unit 6: Measurement and Data Processing - Standard 15 and 16

Assignment Schoology:
https://ssis.schoology.com/assignments/6680283114/info
https://ssis.schoology.com/assignments/6688707720/info

Day 8: Semester 2 Iterative Design (2023/05/05)

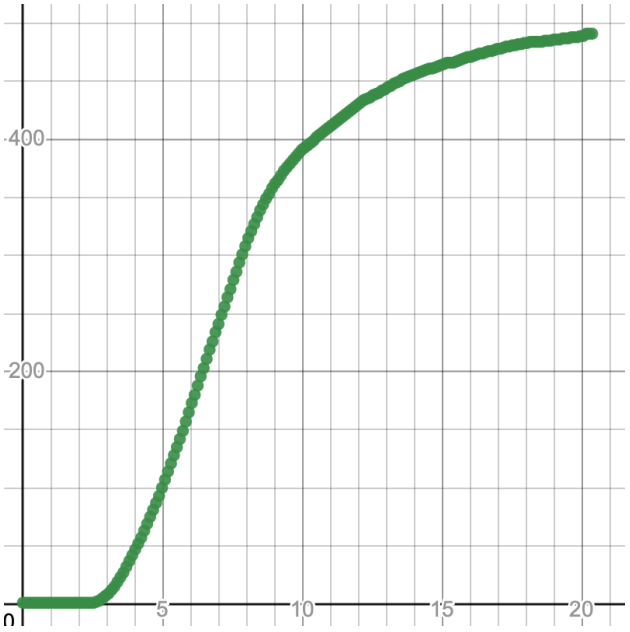
8.1 Iterative Design in Semester 2

- There are three components to this assignment. This is the link on Schoology for the rubric and assignment.
    - Create a new document for iterative design for your work this semester. This must be another element of your work beyond what was done for the quarter 3 iterative design assignment.
    - Add all design reports and iterative design assignments (including this final one) to your portfolio website from semester 1.
    - Add a reflection page answering the reflection questions at the bottom of the assignment.

Day 9: Continuous Read with the Circuit Playground Express (2023/05/09)

9.1 - From CSV to Continuous Read!

- In your projects, most of you are getting close to processing the data in the right way for a single set of data. This usually comes from taking a few seconds of data, pasting the data into a CSV file, and uploading to Replit sketch.
- The goal is to get the code to instead run on the CircuitPython board itself. This means that the processing code is running on the board and you can continuously get the information you want from processing the sensor values. Instead of going through rows of a CSV file, you go through rows of a list of raw sensor values.
- Everything else you do to process the data should be the same!
- The code at this link shows the code for the demonstration Mr. Kreier will do in class. It involves calibrating the light sensor, collecting a set of data for a given amount of time, and then outputting the number of light transitions after that period of time.
- Mr. Kreier ran this yesterday and generated the Desmos graph linked here and shown below.



Day 10: Project refinement for Unit 6 (2023/05/11)

Returning to the Morse Code Challenge: Can you decode the message?
https://github.com/kreier/circuit-playground-express/tree/main/apps
Finish Standard **1**, **9** and **10** for Unit 4. Finish standard **2** from unit 5.

Day 11: Presentation of Unit 6 projects (2023/05/29)

Working on the 4 final projects for Unit 6 Measurement and Data Collection

Day 12: Reflection on achievements (2023/05/31)

Presentation of the 4 projects

- Dan — Speedometer project — Portfolio website
- DJ — Lights — Portfolio website
- William — Thermometer, but better — Portfolio website
- Khang — People counter — Portfolio website

June 1st is the beginning of exam week

June 8th is 40 minutes celebration of achievements in 2022-2023