Most of my personal projects can be found at https://github.com/haw230 or https://codepen.io/hanchengwu/

Python

Internship: Teaching Tool for Summer Camps

The Anadromi Project is a hands-on learning tool taught in the language Python for both formal and informal learning. The learner will become familiar with linux (using the Cloud9 IDE) and learn computer science material in three stages. Stage 1 covers iterative algorithms: linear search, selection sort, bubble sort, and basic time complexity. Stage 2 covers recursive algorithms: binary search, merge sort, quicksort, and more in-depth time complexity. Stage 3 covers iterative fibonacci, recursive fibonacci, and dynamic fibonacci. Each lesson includes a write up of the algorithm and a practical assignment where the student solves the algorithm on their own.

After the internship, I became more proficient with programming in Python, became more familiar with Linux and Bash, learned git, and developed my understanding of each algorithm.

Here is the academic poster that summarizes the project.

Gistify

Gistify is an app that takes in a long body of text and summarizes it into a few sentences using a natural language processing algorithm.

Basic Neural Network

```
# X = (hours sleeping, hours studying), y = Score on test
X = np.array(([3,5], [5,1], [10,2]), dtype=float)
y = np.array(([75], [82], [93]), dtype=float)

X /= np.amax(X, axis = 0) #numbers in each column divided by local max in that column

class Neural_Network(object):
    def __init__(self):
        #Define Hyperparameters
        self.inputLayerSize = 2
        self.outputLayerSize = 1
        self.hiddenLayerSize = 3

        #Random Weights (parameters)
        self.W1 = np.random.randn(self.inputLayerSize, self.hiddenLayerSize)
        self.W2 = np.random.randn(self.hiddenLayerSize, self.outputLayerSize)

    def forward(self, X):
        #Propagate inputs though network
        self.z2 = np.dot(X, self.W1) #X matrix times random weight layer 1; np.dot() does
        self.a2 = self.sigmoid(self.z2) #apply activation function
        self.z3 = np.dot(self.a2, self.W2) #second layer weight (3x1 matrix)
        yHat = self.sigmoid(self.z3) #activation
        return yHat

    def sigmoid(self, z):
        #Apply sigmoid activation function to scalar, vector, or matrix
```

Simple artificial neural network with no real purpose except for me to understand its implementation. It uses the numpy to handle the matrix manipulation and uses sigmoid as the activation function. This demo only has three hidden layers, but the object is dynamic.

[Image Recognizer]

```
# -*- coding: utf-8 -*-
from sklearn import datasets, svm
import matplotlib.pyplot as plt

digits = datasets.load_digits() #dictionary like object

clf = svm.SVC(gamma=0.001, C=100) #classifier is support vector machine

i = -10

X, y = digits.data[:i], digits.target[:i] #set X to data, y to results

clf.fit(X, y) #training

j = -4

plt.imshow(digits.images[j], cmap=plt.cm.gray_r, interpolation='nearest') #displaying
plt.show() #displaying

print(clf.predict(digits.data[j]))
```

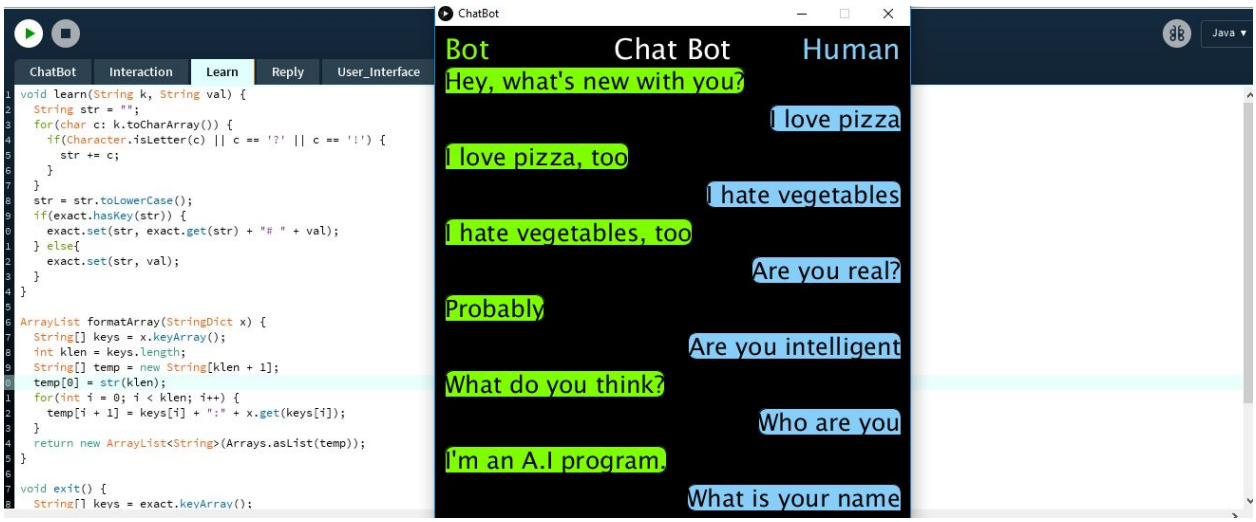
Sci-Kit machine learning for an image recognizer. Both training sets and the support vector machine are packaged into the library.

Java

[Stocks Game]

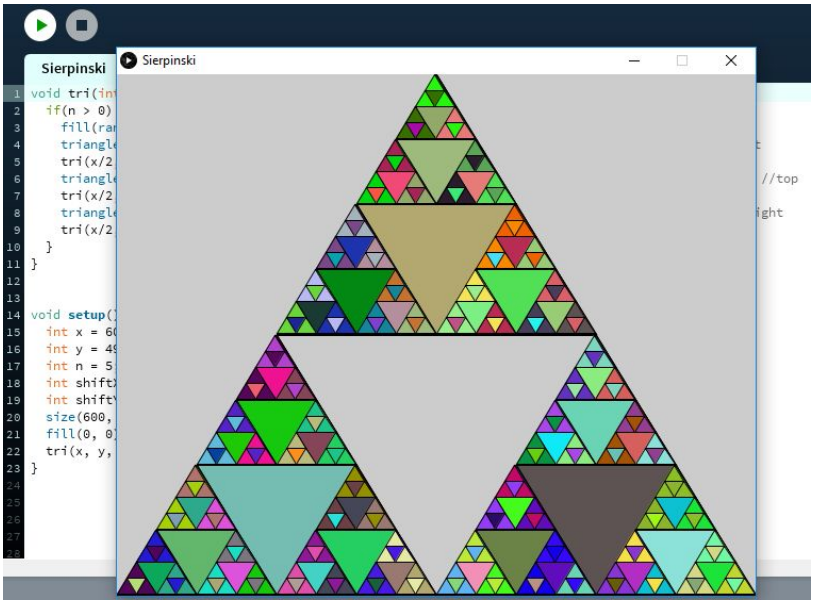
Processing Java application I made for fun. The project uses an API to grab real-time stock data in JSON format. Features include saving progress and loading it, a rags-to-riches mode where the player starts off with only a dollar, and a rapid mode where stock prices are fake but change quickly (real stocks involve a lot of waiting around).

Chat Bot



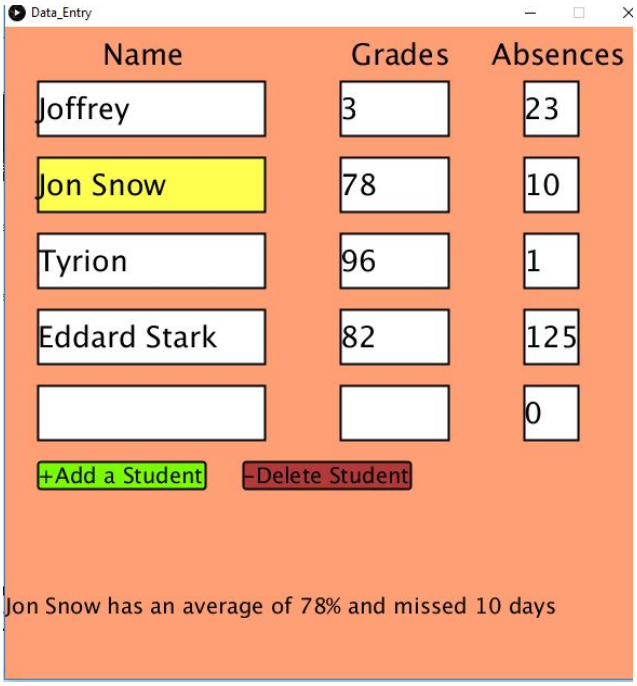
Basic Chatbot that learns from talking to humans by storing responses in an .ini file. The user can scroll through previous conversation log using the arrows. This was a computer science class project which helped me further understand object-oriented development and string and array manipulation.

Sierpinski Triangle



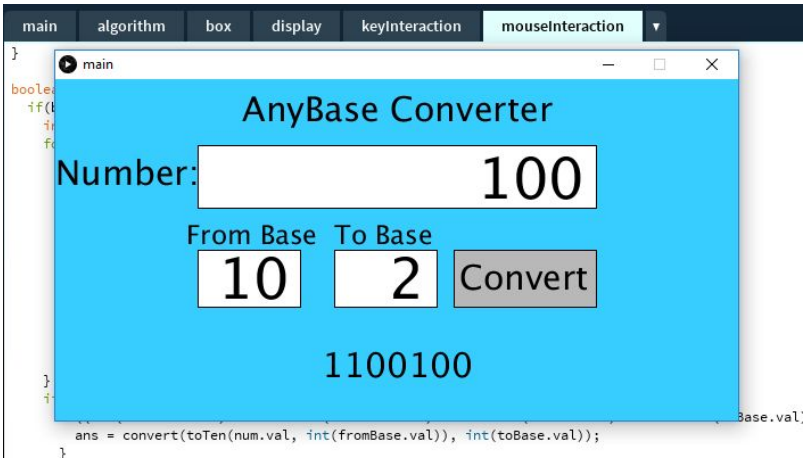
A class assignment where I had to figure out how to write the recursive Sierpinski Triangle algorithm. I enjoyed the problem solving aspects of the task.

Data Entry



Object oriented data entry project.

Base Converter

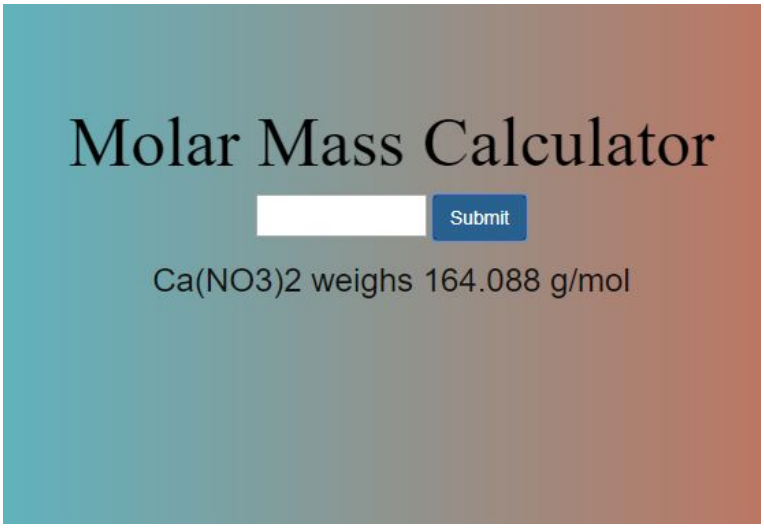


As a project for AP Computer Science class, I wrote an algorithm to convert from one base to another. This was done both recursively and iteratively.

Web Development

I have not maintained some of these projects so they may have broken images or are using outdated APIs.

[Molar Mass Calculator]

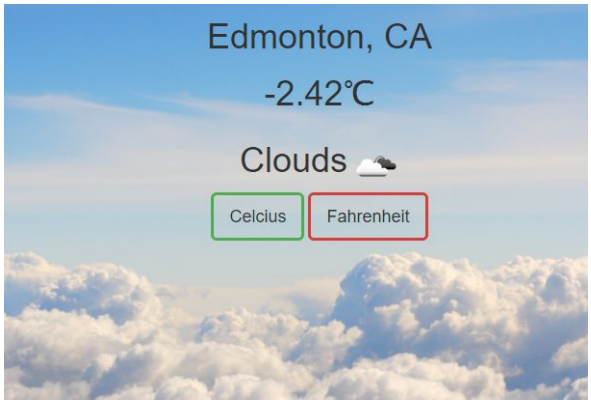


Molar mass calculator for Chemistry. It calls a periodic table API for data and proceeds with the calculation. The most impressive part is calculating atoms with multiple polyatomic ions. This web app was quite popular with my fellow Chem IB class students.

[Pomodoro Timer]

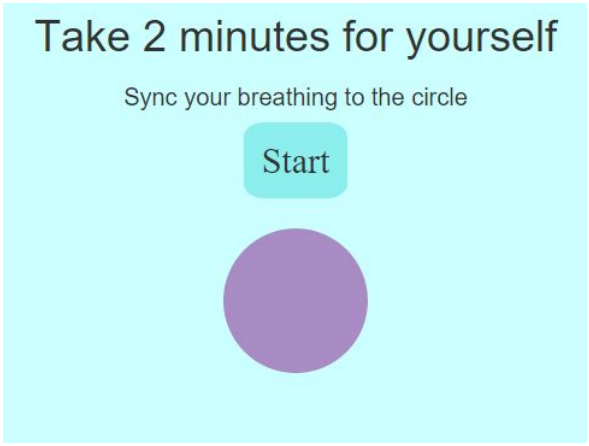
The Pomodoro is an effective productivity technique. Working for 25 minutes and then resting for 5 maximizes productivity. This program follows the technique. The work and break time can also be paused and changed to suit one's needs.

[Local Weather App]



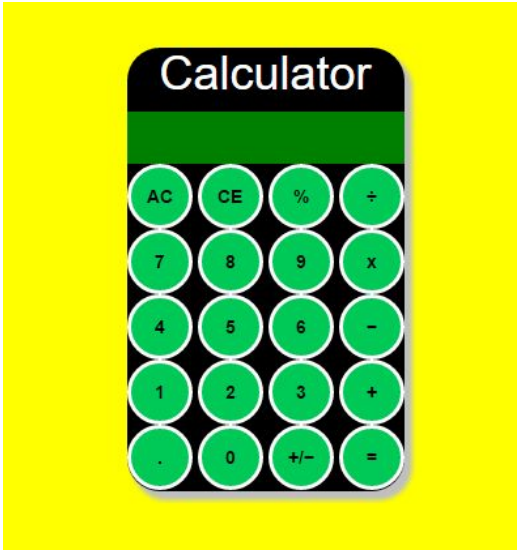
A program that calls the Open Weather API.

[4-7-8 Breathing Technique Tool]



The 4-7-8 rule is a technique that fights insomnia. This program was inspired by a useful online tool which does the exact same thing. Knowing that I can make the same thing, I reverse-engineered the site.

[Calculator]

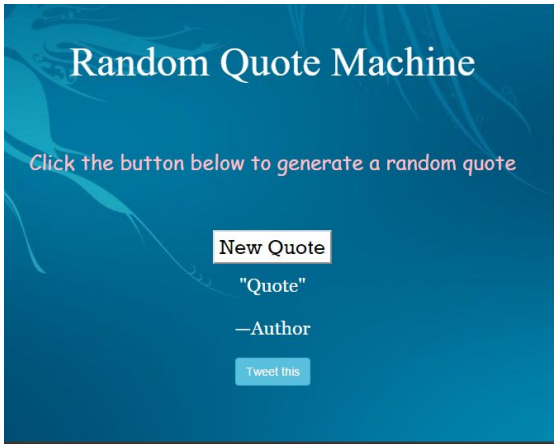


Simple web calculator that simulates the function of a real calculator. This program can chain equations, integer calculation, and allows the use of decimals.

[Cat Clicker]

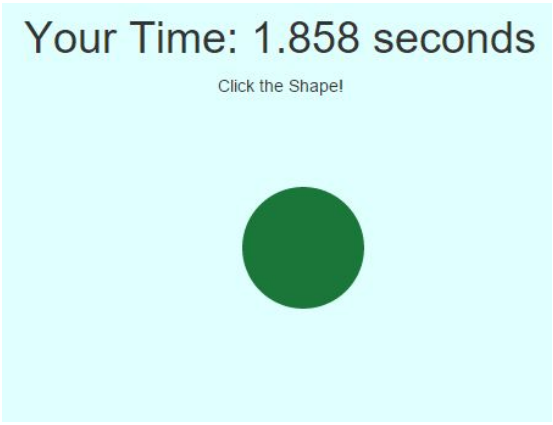
This is a clone of the Cookie Clicker game where you click the image and for each click, the player gets a point. I use Cat Clicker in Programming Club to introduce students to web development because it covers very key concepts in HTML, CSS, and Javascript.

[Random Quote Generator]



The Random Quote Machine generates a random quote upon being clicked. The quote can also be tweeted (if under the 140 characters).

[Reaction Game]



A fun little reaction game that records the time it takes you to click the object that appears somewhere randomly on your screen.