Prayash Mishra

3rd year
Software Engineering
University of Guelph

Github: https://github.com/prayashm97/
Linkedin: https://www.linkedin.com/in/prayash-mishra-372a32126/

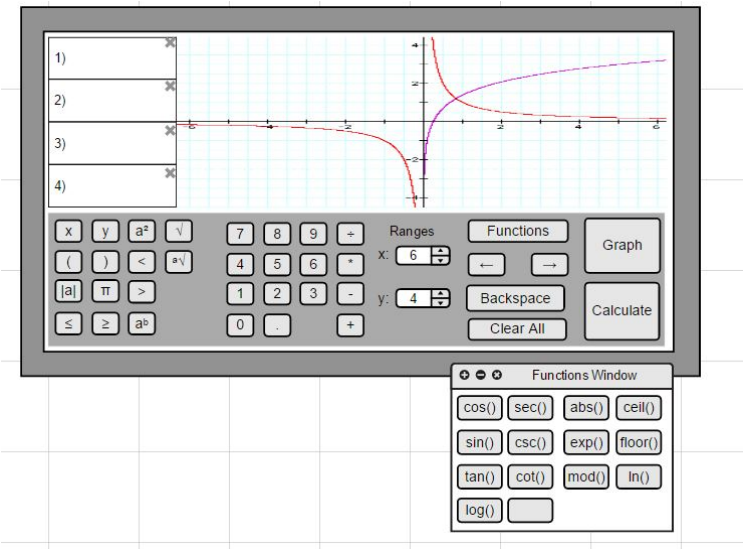
School Work

CIS 3250 (Software Design III)

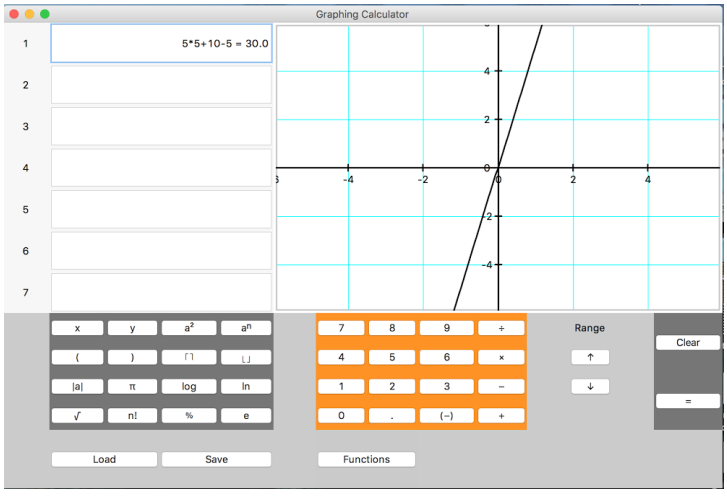
- Language: Python 2
- Libraries: Tkinter for GUI

- Worked with a group of 7 students to develop a GUI based graphing calculator. Made utilizing python's Tkinter library, giving the user the ability to graph, calculate, and validate functions.
- Used Shunting yard algorithm to convert from infix to postfix.
- Started with technical documents, such as Design Document and Requirements Document.
- I made the GUI prototype and was involved in the user interface design document.
- I was lead developer for the GUI.
- We achieved 96% on the project and awarded best design in class.

  Code and Technical Documents available upon request



Prototype of the calculator



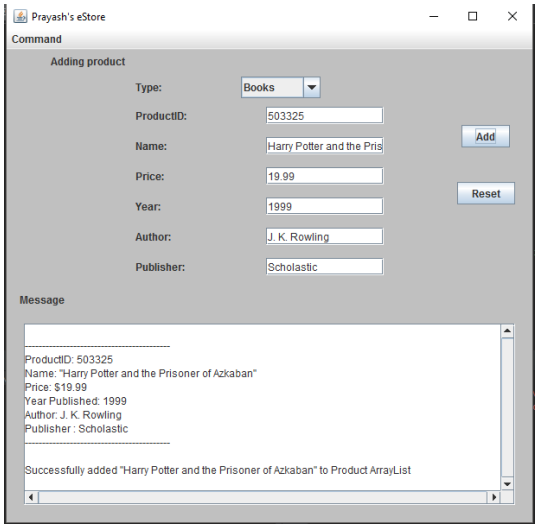
Final GUI of the calculator

CIS 2430 (Object-Oriented Programming)

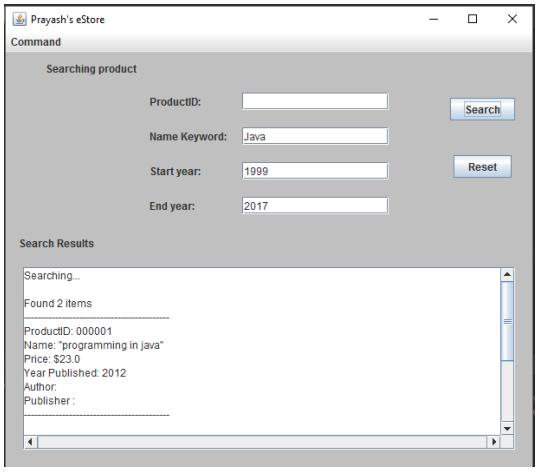
- Language: Java
- JRE version 8.11
- IDE: IntelliJ IDEA

- This course focused on the fundamental concepts of Object-Oriented Programming.
- A Java application that stores collection of book and electronic items.
- User can add, and search items.
- The program consists of hashing, inheritance, exception handling, polymorphism, Java's Swing/AWT, File I/O, etc.
- There were 3 assignments that built on top of each other.
    - First : A simple program which focused on class design. Achieved 96%
    - Second: Adding features such as File I/O and searching using HashMap. Achieved 97%.
    - Final: To implement a GUI interface and exception handling to make the system robust and user-friendly. Achieved 100%
- In the process of making an Android app based on this project with the Professor's guidance.

  Code available upon request.



Adding an Item



Searching for an Item

Personal Projects

Android

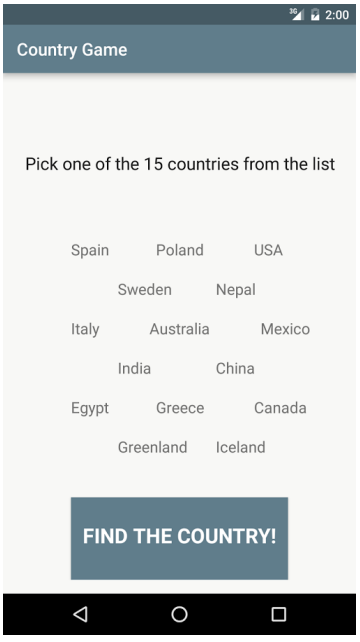
Country Game

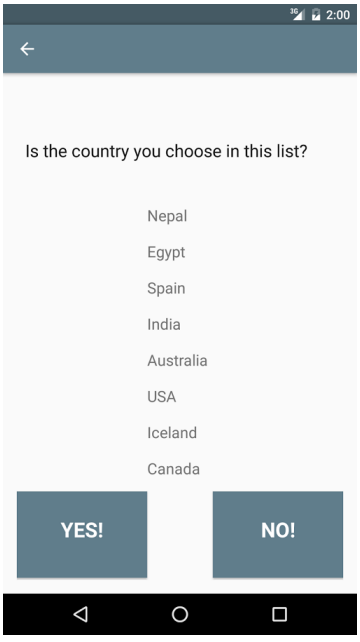
- Language: Java, Android SDK, XML
- Targeted Android version: 23 (Marshmallow)
- Minimum Android version : 19 (KitKat)
- IDE: Android Studio

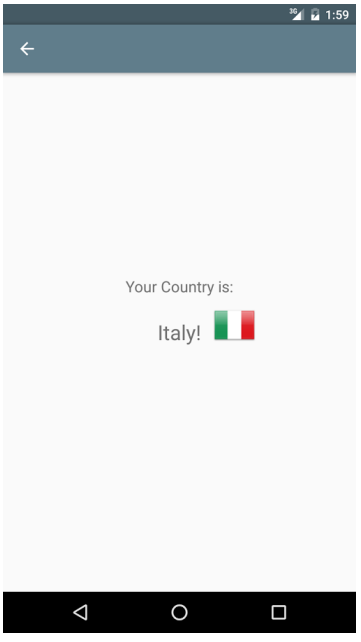
- Android app made primarily to learn basics of Android app development
- Java for functionality, XML for layout design
- User thinks of a country from a given list of 15 countries.
- On start, each country is given a key from 1 - 15
- On each activity user is given a list of 8 countries. User has to let the program know if their country is in the list of 8. If the country is in the list, select “Yes”.
- Every time user selects “Yes”, program adds 2^x to a variable that is passed as Intent
- Where x is the activity number. So, on first activity if user selects yes, add 1 to total
- On second, add 2. On third, add 4 and so on.
- Depending on the total of the intent variable, program reveals the user’s choice of country.
- If user selected “No” for every activity, then the intent variable is 0, output is a error message telling the user to try again.

  Code available on request.

https://play.google.com/store/apps/details?id=prayashmishra.com.country_game







YahooWeatherApp

- Language: Java, Android SDK, XML, JSON
- Targeted Android version: 24 (Nougat)
- Minimum Android version : 21 (Lollipop)
- IDE: Android Studio

- Android app made primarily to learn how to handle Web APIs in Android.
- Java for functionality, XML for layout design
- JSON to understand data.
- API documentation
- Uses Yahoo's Weather API to get the weather.
- Yahoo Weather API url for weather of Toronto : https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20weather.forecast%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text%3D%22toronto%22)&format=json
- Uses a query, in the query enter a city's name, using the city's name, gets the woeid of the city, using the woeid, gets the forecast, returns the city's forecast in JSON format.
- Sample output:


```
"atmosphere":{
  "humidity":"75",
  "pressure":"997.0",
  "rising":"0",
  "visibility":"16.1"
},
"astronomy":{
  "sunrise":"7:19 am",
  "sunset":"5:45 pm"
},
"item":{
  "title":"Conditions for Toronto, ON, CA at 12:00 AM EST",
  "lat":"43.64856",
  "long":"-79.385368",
  "pubDate":"Mon, 13 Feb 2017 12:00 AM EST",
  "condition":{
      "code":"23",
      "date":"Mon, 13 Feb 2017 12:00 AM EST",
      "temp":"31",
      "text":"Breezy"
  },
```


- The "item" object has the information such as latitude and longitude of the city, condition of the city, inside condition there is the temperature of the city.
- Similar to "item", "astronomy" and "atmosphere" there are more objects that give information on the units, forecast for next few days, wind information, etc. Very easy to extract data.

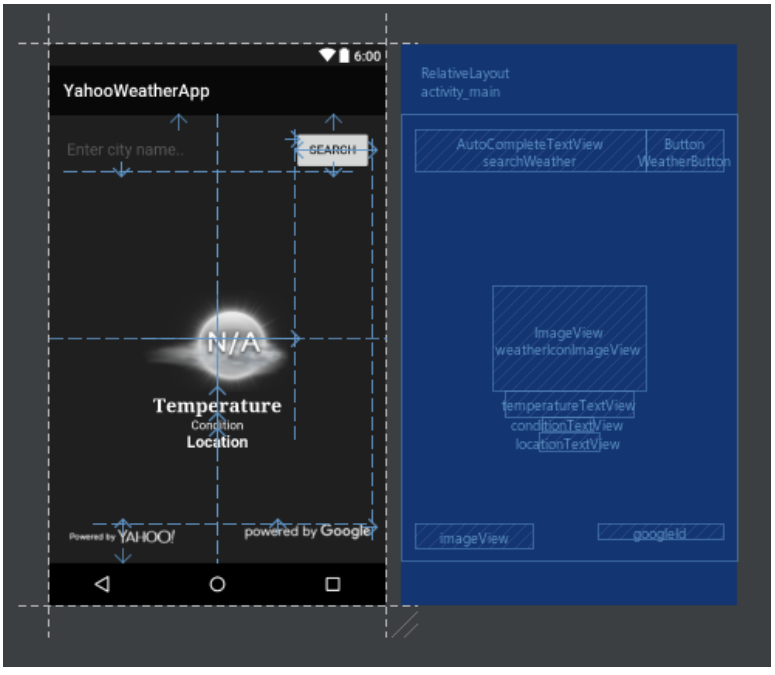
Continued on next page.

- Uses Google Places API Place Autocomplete to return a list of cities.
- API url with city name similar to “tor” : https://maps.googleapis.com/maps/api/place/autocomplete/json?key=AIzaSyCEwp4TvdqeCVAKbnYm1qYq0XQhdcF1yEU&types=(cities)&input=tor
- Sample output:

```
{
        predictions" : [
         {
                "description" : "Toronto, ON, Canada",
                "id" : "9cdc0b86ce6052ab269593184e7762372e698584",
               …..
         },
         {
              "description" : "Turin, Metropolitan City of Turin, Italy",
              "id" : "6df1d595fb03452c4811e884eb6b6d0bf3a92714",
              …….
         }
        …..
}
```

- The output gives a list of prediction that google made for the city’s name
- Append the predictions onto a drop down list for user to choose.
- It also has information about the formatting structure, city id, reference id
- User can type on the text box, google will show some prediction, user selects what city’s weather they want, the app displays the weather.
- Simple toggling from Celsius and Fahrenheit by just touching the temperature
- Not finished, still working on it

Code and app APK available on request.



Interface of YahooWeatherApp

Web Development

CountryAPI_JS

- Language: HTML, CSS, JavaScript, JQuery, AJAX
- JQuery version: 1.12.4
- https://jsfiddle.net/
- Text Editor : Brackets

- Web application made primarily to learn JQuery and AJAX.
- GET request using AJAX to a RESTful API
- API url: https://restcountries.eu/rest/v1/all
- Sample output for Canada :

```
{
  "name":"Canada",
  ..
  "capital":"Ottawa",
  ..
  "region":"Americas",
  "population":35749600,
  "currencies":[
        "CAD"
  ],
  ...
},
…..
```

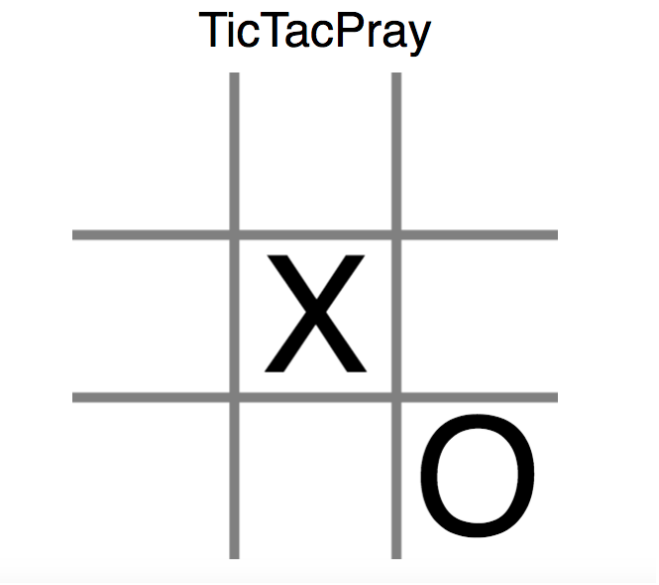
- Output gives a list of many different countries
- Has information about name of the country, capital, population, etc.
- Also can get information about the countries that borders that country, timezones, calling code. Easy to extract those data with a single GET request.
- Allows user to select from a list of countries using a drop down list.
- AJAX call to API to get the list of countries.
- Fills table with information about the country user selects.

http://pmishra.me/country

Tic-Tac-Pray

- Language: React, Node, CSS
- Text Editor : Atom

- Web application made primarily to learn React, Node and HTML Canvas
- Made most of the UI components using React Material UI framework
- The Tic Tac Toe board made with HTML canvas
- The game uses a simple AI who places O's wherever there is a free space.
- The application is not fully finished, no state to check whether the user wins or loses.



http://tic-tac-pray.herokuapp.com/

Quiz App - Node

- Technologies : Node, MongoDB
- Node version: 9.2.0
- Text Editor : Atom

- Web application made for parties, school, etc.
- There are 3 types of users:
    - Admin:
        - Has the most privileges, can add questions, save to database.
        - Select which question to queue up or deploy to selected users.
        - Can invite users.
        - Can only login using email.
    - User:
        - Answers questions.
        - Given points for answering correctly.
        - Can have their own profile with name, display picture, etc.
        - Can login using Email or Social Media platforms.
    - Moderator:
        - Can give admin access to Users
        - Able to kick users or admins.
        - Reviews Questions and Answers added by Admins.
- Questions can be True or False/ Multiple Choice.
- Moderator will review Questions before adding to database.
- This Application is still in the very first stages.
- Currently making the user model, routes.
- Haven't made decision on the View, could use React or try Vue.

https://github.com/prayashm97/quiz_app-node

More information:

- Learning Node, React, Vue, iOS development.
- Comfortable working with Java, Javascript and Ruby.
- Familiar with Relational Database concepts, Stored Procedures, Functions.
- Actively using Github as a version control service for both personal and school projects.