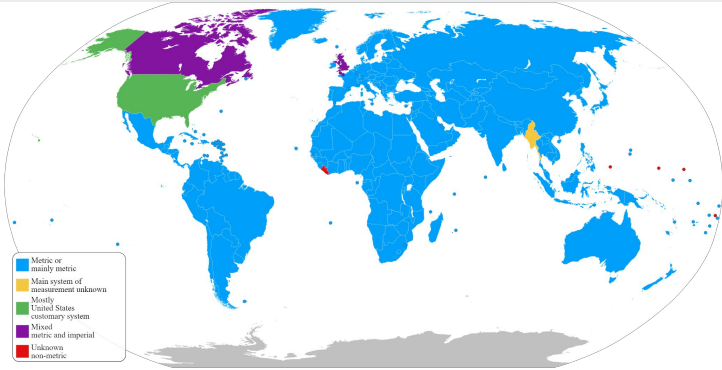
Unit and Currency Converter

Reed Glenn, Hanna Stolarska, Abby Farrell, Lex Gaylor, Quinn Trout

Reasoning



- Powerful tool for individuals and businesses alike
- Can help people to easily convert between different units of measurement and currencies
- Useful when traveling, conducting international business
- Can help them to save time, avoid mistakes, and ultimately achieve better outcomes

Functionality

- Taking units for weight, volume, temperature, and time and converting them into different units relating to the same measurement

- Taking in a currency and comparing it to a different international currency to show the latest conversion rate between the two

- Basic menu-interface where the user selects an option based on their need

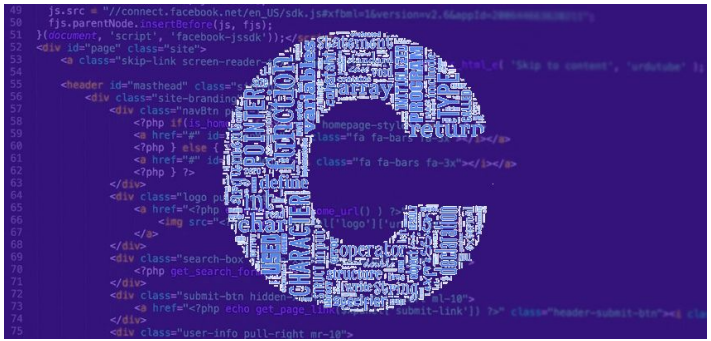
Interacting with our program

- GUI which prompts them with a choice between different unit conversions

- Choose a method, input a value, and the program returns the converted value
  - Consistent rates from a Metric-Imperial conversion
  - Currency rates will be subject to change from our API updating

Technical components/Programming

- Different functions that are called by the user with user input to convert data from one format to another
- Curl is being used to access currency API information



- GTK GUI builder was used to give the user a friendly and intuitive graphical interface
- All of the Metric-Imperial conversions were written as functions in ANSI C and utilize equations which work both ways

Examples of Our Code

```c
#include <math.h>
#include <stdio.h>
// Celcius to Fahrenheit

void CtoF(double celcius) {
  double fahrenheit = (celcius * 9 / 5) + 32;
  printf("\n You have entered %lf degrees of Celcius.\n That is equivalent to "
         "%.2lf degrees of Fahrenheit.",
         celcius, fahrenheit);
}

// Fahrenheit to Celcius

void FtoC(double fahrenheit) {
  double celcius = (fahrenheit - 32) * 5 / 9;
  printf("\nYou have entered %lf degrees of Fahrenheit. \n That is equivalent "
         "to %.2lf degrees of Celcius.",
         fahrenheit, celcius);
}

// Miles to Km
void miToKm(double miles) {
  double km = miles * 1.60934;
  printf("\nYou have entered %lf miles\nThat is equivalent to %.2lf kilometers",
         miles, km);
}
// Km to Miles
void kmToMi(double km) {
  double miles = km * 0.621371;
  printf("\nYou have entered %lf kilometers\nThat is equivalent to %.2lf miles",
         km, miles);
}
```

```c
// feet to meters
void fToM(double feet) {
  double m = feet * 0.3048;
  printf("\nYou have entered %lf feet\nThat is equivalent to %.2lf meters",
         feet, m);
}
// meters to feet
void mToF(double m) {
  double feet = m * 3.28084;
  printf("\nYou have entered %lf meters\nThat is equivalent to %.2lf feet", m,
         feet);
}
// inches to centimeters
void inToCm(double in) {
  double cm = in * 2.54;
  printf(
      "\nYou have entered %lf inches\nThat is equivalent to %.2lf centimeters",
      in, cm);
}
// centimeters to inches
void cmToIn(double cm) {
  double in = cm * 0.393701;
  printf(
      "\nYou have entered %lf centimeters\nThat is equivalent to %.2lf inches",
      cm, in);
}

// kilograms to pounds
void KgToLb(double kg) {
  double lb = kg * 2.20462;
  printf("\n You have entered %lf kilograms. \n That is equivalent to %.2lf "
         "pounds.",
         kg, lb);
}
```

```c
// pounds to kilograms
void LbToKg(double lb) {
  double kg = lb / 2.20462;
  printf("\n You have entered %lf pounds. \n That is equivalent to %.2lf "
         "kilograms.",
         lb, kg);
}
// milliliters to ounces
void milToOz(double ml) {
  double floz = (ml / 0.033814);
  printf("You have entered %lf milliliters\nThat is equivalent to %.2lf fluid "
         "ounces",
         ml, floz);
}
// ounces to milliliters
void ozToMil(double floz) {
  double ml = (floz * 29.5735);
  printf("You hae entered %lf fluid ounces\nThat is equivalent to %.2lf "
         "milliliters",
         floz, ml);
}
// mph to km/h
void mphToKMH(double mph) {
  double kmh = (mph * 1.60934);
  printf("You have entered %lf miles per hour\nThat is equilavent to %.2lf "
         "kilometers per hour",
         mph, kmh);
}
// km/h to mph
void kmhToMPH(double kmh) {
  double mph = (kmh / 0.621371);
  printf("You have entered %lf kilometers per hour\nThat is equilavent to "
         "%.2lf miles per hour",
         kmh, mph);
}
```

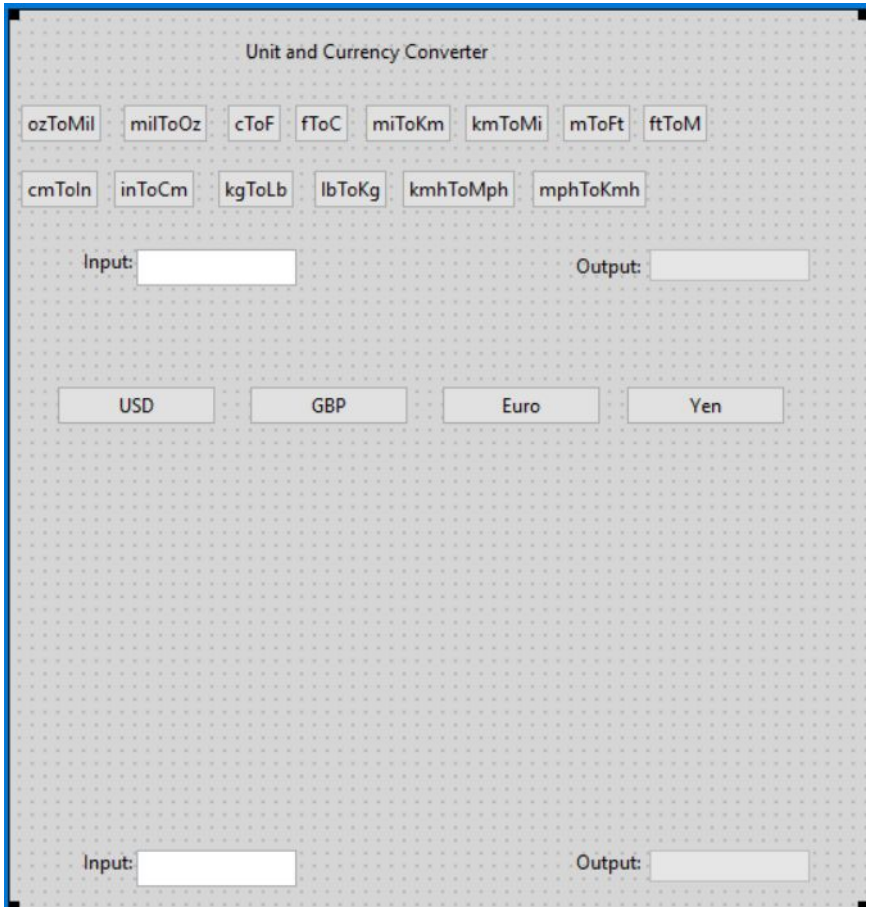
External API: Exchange Rates

The program uses curl to grab the requested information from the API being used. It then displays to the user the current currency exchanges for the requested currencies.


```
{
    "success": true,
    "timestamp": 1519296206,
    "base": "EUR",
    "date": "2021-03-17",
    "rates": {
        "AUD": 1.566015,
        "CAD": 1.560132,
        "CHF": 1.154727,
        "CNY": 7.827874,
        "GBP": 0.882047,
        "JPY": 132.360679,
        "USD": 1.23396,
    [...]
    }
}
```

UI Implementation

- GTK GUI builder was used to implement a user interface for the program
- Consists of buttons to choose which conversions will take place, as well as output sections so the user knows the result of the conversion



Analysis - Completed Work

- Conversions for weight, temperature, volume, and time

- GUI front end

Analysis - Incomplete work

- GUI back end
  - Input/output functionality

- API implementation
  - Connection to GUI

https://us05web.zoom.us/j/83312707708?pwd=cFNIVWIwdWw4TVNjU0RTbFJrSjNNdz09