Project Description

My project, named SmartTranslate, is an educational tool allowing users to manipulate how they would like to translate longer pieces of text, usually paragraphs or entire documents. Because it is an educational tool, it will support translation between languages that are commonly taught in classrooms and featured in AP and SAT Subject Tests, including Spanish, French, Chinese, Latin, Italian, Japanese, and Korean. In terms of design, the app will primarily be separated by the left and right side of the screen: the left side will contain the text to be translated, and the right side will feature the translated text. Users have the option of inputting text by simply copy and pasting text, uploading an image, or uploading a PDF document. Users then have the following options of translating text:

- Direct button - Translate the entire text into the desired language. The output will be an entire paragraph.
- Detail button - Translate each word or character to the desired language. The output will be a list of each individual word or character with its individual translation beside it.
- Highlight button - Highlight certain words to be translated. Only the translation of the highlighted word will appear on the right side of the screen. Users will have the option to “take notes” of their translation by hitting enter: the translation will then remain on the right side of the screen. Once a note is made, a box will be made around the highlighted word so that users can click on that word later on to automatically take them to the translated note on the right side of the screen. This is especially useful if a lot of notes are made and users want to get to the translated note quickly. Otherwise, if users unhighlight the word, the translation will disappear from the right side of the screen.

Competitive Analysis

- Google Translate - Similar features that my program will have with Google Translate is the option to highlight certain parts of text and have the translation pop up on another screen. Although Google Translate allows users to translate parts of text by highlighting them, the problem is when users unhighlight the word, the translation disappears. There is no way to save the translation, and sometimes, users forget the translation and have to the highlight the word again. To make this more convenient, my app allows users to hit enter so that the translated words can remain on the screen while the user proceeds to highlight and translate other words. My app will also have a feature that allows translation of every single word or character in a long text.
- LiveThesaurus (Fall 2018 term project) - My program is similar to this in that it allows users to highlight certain words of a large text to interact with them. LiveThesaurus highlights words to produce synonyms, SmartTranslate highlights words to produce translations.

Structural Plan

- Text Conversion File - This file converts text that is uploaded through image or PDF into text that can be edited.
- Main App File - This file takes care of the actual interface and uses tkinter to organize text widgets and buttons.
- Translation File - This file takes of all the translation features, such as separating all input text into a list of individual words and characters for the detail translation feature and the ability to translate words that are highlighted for the highlight feature.
- Notes File - This file enables storing of saved translation notes.
- Draw File - This enables users to draw in characters with mouse input if they ever need to edit the input text.

Algorithmic Plan

The trickiest part is to figure out how to translate only highlighted words and how to save those highlighted words on the output screen whenever the user presses enter. For highlighting, since the tkinter text widget has a tag called sel or selection corresponding to the current selection of text. I can use the constants SEL_FIRST and SEL_LAST to refer to the selection. I will use a try and except so that when there is a selection, the code will keep running and recognize that the text between SEL_FIRST and SEL_LAST will be selected.

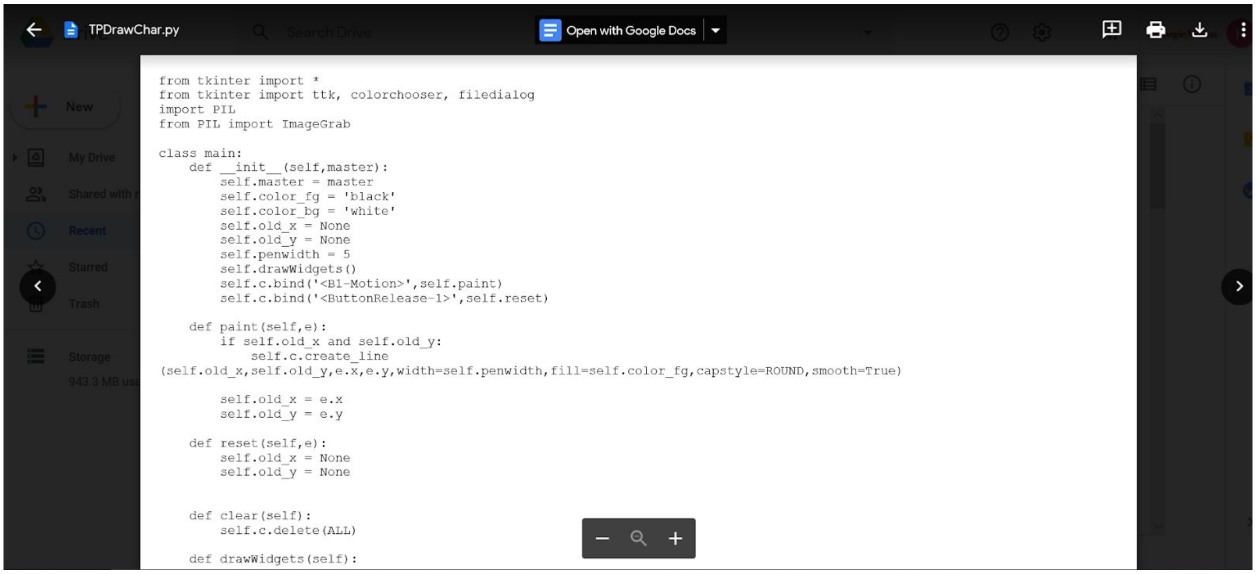
To save translation notes, I would want to create a clickable box around the input text that the user has highlighted and hit enter to save the translation for. This is so that if the user chooses to read over the text again, they could click on that box with the desired text inside and the output screen will automatically scroll to where that text is translated. Each clickable box and its translation will be stored in dictionaries so that they can be easily accessed.

Timeline Plan

- TP 1
    - Be able to read text from a saved image.
    - Be able to translate the entire text as is without any manipulation from the detail and highlight features.
    - Set up basic tkinter interface with input text on left side and output text on right side.
    - Set up mouse input drawing app for characters.
- TP 2
    - Work on upload image and pdf features
    - Work on detail and highlight feature so that they are properly functioning
    - Add button widgets to the interface

    - Link the main app file and translation file so buttons are properly working with features
    - Figure out how to switch output screens between different features
- TP 3
    - Make sure all buttons working properly and output screens can actually be switched between different features
    - Work on save notes feature within highlight feature so that users can save their highlighted translation notes for later access.
    - Implement a login feature and creation of accounts
    - Implement OpenCV so that images can be taken with webcam and directly uploaded (though this is not crucial to the entire project)

Version Control Plan



```
from tkinter import *
from tkinter import ttk, colorchooser, filedialog
import PIL
from PIL import ImageGrab

class main:
    def __init__(self,master):
        self.master = master
        self.color_fg = 'black'
        self.color_bg = 'white'
        self.old_x = None
        self.old_y = None
        self.penwidth = 5
        self.drawWidgets()
        self.c.bind('<B1-Motion>',self.paint)
        self.c.bind('<ButtonRelease-1>',self.reset)

    def paint(self,e):
        if self.old_x and self.old_y:
            self.c.create_line
(self.old_x,self.old_y,e.x,e.y,width=self.penwidth,fill=self.color_fg,capstyle=ROUND,smooth=True)

        self.old_x = e.x
        self.old_y = e.y

    def reset(self,e):
        self.old_x = None
        self.old_y = None


    def clear(self):
        self.c.delete(ALL)

    def drawWidgets(self):
```

This project is being backed by uploading the files to Google Drive after each time I have worked on it.

Module List

- Pytesseract for reading text from images
- PIL for image file manipulation
- Googletrans for translation API

TP 2 Update

In order to add more complexity to my program, I have added a feature where users will have the option to save the words they highlighted and use those words in a quiz game that will test their knowledge on the highlighted words. The game will be in a multiple choice format where each question will be the word and users have have to choose the correct translation. Users will have 4 boxes to choose from, and the correct answer will be randomly placed in 1 of 4 boxes. The game will also be able to track how many times a user gets a certain question wrong, and an algorithm will be implemented so that questions that users answer wrong will appear more often in the quiz.

TP 3 Update

The biggest change made is to the game component. Users have the option to track their performance of each quiz game they complete. At the end of the game, there is a button "Progress Report" that when clicked displays a graph of the quizzes users have taken and the scores they received on each quiz, allowing them to see whether they have improved or deproved in general. Users can also clear their progress. The data of the user's progress are stored in a text file.