**Tianze Shou Term Project Proposal**

Project Description

The name of my TP will be *Matrix Solver*. It is basically a matrix calculator for users to conduct matrix operations, finding inverses, finding determinants, solving systems of equations using Gaussian Elimination, finding eigen-values, etc. My project will not only be a calculator, but it will show the intermediate steps to the users so that they can better study the basics of linear algebra.

Competitive Analysis

One similar project I have seen online is called *Matrix Calculator*, where users can enter values for matrices and obtain results of operations of their desire. However, this system only involves number calculator, which means that it cannot solve linear systems with one or more free variables. Incorporating free variable when solving linear system is what I intend to incorporate in my project. Furthermore, *Matrix Calculator* only displays results of the calculation, omitting the middling steps, which are vital for students who are currently studying the basics of linear algebra. My project will show the users how the program gets the result, so that the users can better grasp complex constructs in linear algebra calculation.

Structural Plan

My project will be split into three major files. One for the basic matrix algorithms, which is the Matrix Class. I will be using it to create matrix objects and conduct matrix operations. I have already completed the majority of this part before TP1. the second section will be tools to create the user interface. Since I am not using external modules from Tkinter, I will be creating my own button and text enter fields. For this section, I will also create a parser to handle the mathematical expressions entered by the user and conduct calculations from there. The third part of the main file that uses the two aforementioned files and takes charge of handling events from the user and displays the animations.

Algorithmic Plan

My first algorithmic challenge is to carry out linear system solutions with free variables. I will first find the rank of the linear system and determine if it has free variables or not. Those with free variables will be handled separately from those without. Then then program will determine which variables are free by conducting Gaussian Elimination and finding columns without pivots, and enter them directly into the solution matrix. From then on, the unfree variables will be determined as expressions using the free variables.

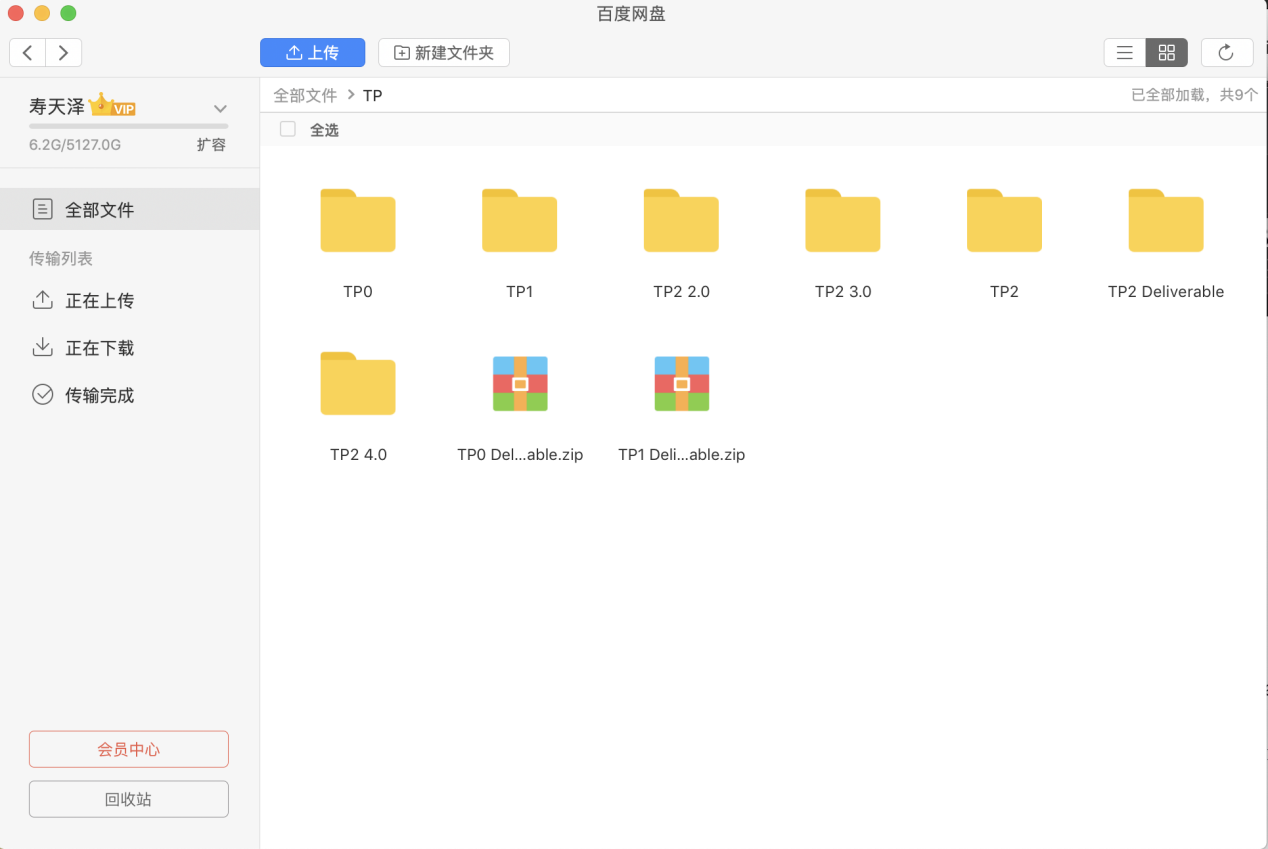
My second challenge is to write my own buttons and text enter fields. I will create an individual class for both of them. The objects of those classes will store their locations on the canvas, the function(s) to be called when the mouse is pressed on them, and text entered the in the field, etc. Later on, the main animation file will call those objects and use their functionalities.

Timeline Plan

I intend to finish my own Matrix module before TP1, and have the user interface done by TP2. Any more complex feature that I come up with during the process will be added before TP3.

Version Control Plan

To backup my code, I will store all my files in Baidu Cloud. I will mark a new version of the project after every large change made or after I spend a large chunk of time working.



Module List

No external module will be used for this project.

Design Changes - TP2

I have decided not to use a Button class. Instead, the user can just press the Enter key to have the program calculate the result. Also, after MVP, I will use the numpy module to realize the eigen-vector and eigen-value feature.

TP3 Changes

Implemented numpy module to add in the eigen feature, which calculates the eigen-values and eigen-vectors for the user.

Imported a logo designed by Jing Tian. Place onto the user interface.