Project Description

Term Project Name: Live, Laugh, and Love Yourself

Live, Laugh, and Love Yourself is a project that's focused on balancing the overall lifestyle of an individual and regulates the physical and mental well-being many people have trouble sustaining today.

Similar Projects

Live, Laugh, and Love Yourself is a project that's very similar to Justin Forman's Calendar project from Fall of 2013 and Chris Schuler's To-Do List and Calendar project from Spring of 2013. Like both projects Live, Laugh, and Love Yourself is also a calendar-based project where users are able to implement and schedule a personal calendar based on the users' desires. Similar to Chris Schuler's To-Do List and Calendar project, Live, Laugh, and Love Yourself will also have multiple tabs and functions, unlike Justin's Calendar project, where users can have access to more information and functions than simply implementing events or schedules. Also similar to Schuler's project, this project will also be able to categorize the events users put into the calendar.

A difference between Live, Laugh, and Love Yourself and both of the projects mentioned above is Live, Laugh, and Love Yourself's purpose, unlike regular calendar projects, is to allow its users to use a calendar to not only plan their daily tasks but also other factors of their life accordingly. Not only can users implement their daily tasks, the project also takes into account the personal habits of its users like storing information regarding the user's sleep and meal schedules, weekly fitness plans, and other lifestyle habits. If the program notices that the user's habits aren't to healthy standards, the program will alert the user by letting them know what the healthy standard of a certain field is and the harms/benefits of following or not following such regulations. For instance, if the program notices that the user hasn't worked out for a certain amount of minutes per week, the program will alert the user to leave out more time to hit the gym etc. If the program realizes the user has been working past an x amount of hours a day and not really living a balanced life, the program will also alert the user to love themself and spend some time doing something not work related in the day. From all these functionalities of this program, it is clear that Live, Laugh, and Love Yourself is more than just a calendar that schedules the daily events/chores of an individual but a program that allows the user to enjoy and love their life in a well-rounded and healthy way.

Structural Plan

CONSIDERING AND WONDERING IF I SHOULD PUT ALL THESE CLASSES INTO DIFFERENT FILES AND CALL THEM THROUGH A MAIN FILE

Different Classes

- Button Class
  - Contains all the different types of buttons with functions regarding what happens when a button is clicked
- Alert Class
  - Store the different types of alerts and associates to Button Class
    - When a button is clicked, if certain cases happen, this alert will pop up
  - Also stores the different information and text that will be presented to the users in the alert
  - Maybe also contains the link to extra information regarding an educational topic if the user wants to look into them
- Calculation Class
  - Associated with both the button and alert Class
    - When the user clicks the “enter” button regarding information of their personal lifestyle or work hours, if the program calculates there’s something unhealthy about these number of hours, an alert will pop up
- Events Class
  - Categorizes everything the user inputs into the calendar, including their personal information and call the calculation class to make sure the numbers are healthy and right

Algorithmic Plan

For the algorithmic plan of this project, I've decided with my mentor that I will be doing a backtracking function that works when the project is trying to find an empty slot in the schedule for its users. For instance, in this project, whenever the calendar notices the user has overscheduled themselves for work, the calendar will come up with a notice reminding the user they're being overworked and ask whether the user would like to find time to do something fun and relaxing. If the user answers yes, the calendar then would backtrack through the schedule and find an empty/unoccupied space in the calendar for time to relax or plan something fun. The function would backtrack through the calendar for an unoccupied time slot and if it finds an empty slot, it will ask the user if the timing is suitable, if not, the system will have to backtrack through the calendar again for another empty slot. If no free time is found, the program will send a message telling the user nothing is found and that they're overworking themselves and to find time to enjoy life.

Timeline Plan

BY TP1:

- Have some of the calendar of the project created
- Start to store and code the many button classes
- Start to store and code the many alert classes
- Start to store the code the different calculation classes
- Start to store and code the different Event Category classes

BY TP2:

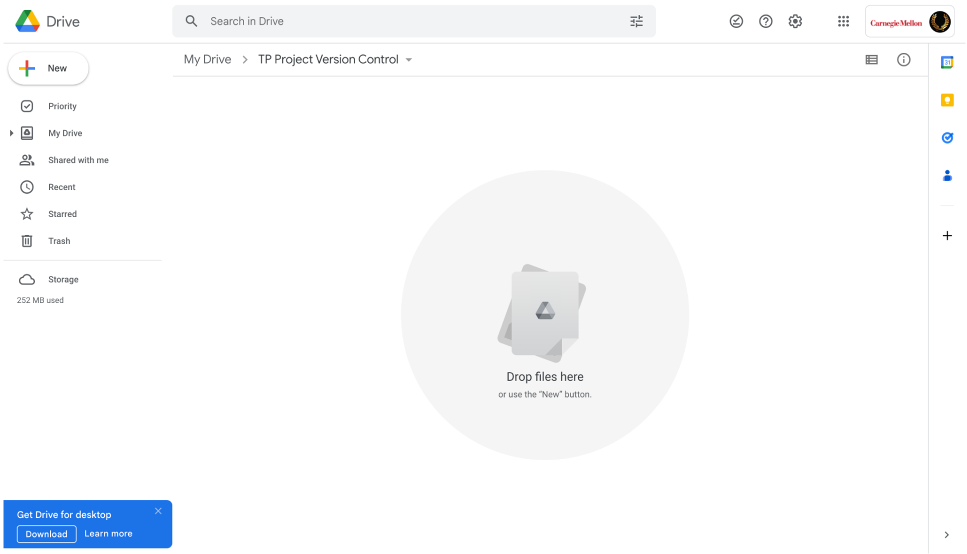
- Fully finished and completed all the different classes and bullets listed under TP1
- Collect all the information and texts that should appear in the alert class(including the actual educational information)
- Be almost done with the graphics and animation of this project

BY TP3:

- Cleaned all bugs and see a clear overview of the project
- Have the ability to dictate what's possible to implement in this project and what to disregard for timing purposes
- Record Video Demo

Version Control Plan

I will be backing up all my code through a google drive folder I created just for TP Project Version Control every night after working on my project/coding. Here is a picture of my google drive folder for my code although it's empty right now.



TP1 Update

- Instead of a 30 day calendar, I've decided to make the calendar a weekly based on so the information can be more clear and simplified
- Made a new category so there's a page that explains the functionalities of this program
- Want to take away the reward system since it seems unnecessary

- Decided to change the layout of both my code and project user interface (front end)
    - Might want to divide all my screens into subclasses
        - So each screen will have its own class that shows what to draw or do
    - Want to change what screens certain buttons will bring

<u>TP2 Update</u>

- Changed the complexity algorithm so now users will be able to implement more than one event into a list and the program is supposed to show a possible time for those events.

<u>TP3 Update</u>

- Updated the button class so now I don't have a bunch of coordinate conditions for all my mouse pressed functions
- Fixed and resolved my backtracking recursion function so now it's working the way it's supposed to