Ein Jeong (einj)

Project Proposal Components [15 pts]

- **Project Description** [2.5 pts]: The name of the term project and a short description of what it will be.
    - The name of my project will be ‘Pomo Planner.’ This project enables study planning by allowing users to add tasks under the subjects of choice. There will be a feature that acts as a Pomodoro timer, which typically sets the study time as 25 minutes with a short break of 5 minutes and a long break of 10 minutes, but could be altered by the user. Under the planner page, there will be a display of study hours for that day, and under the calendar page, there will be a display of study hours for the month. The calendar page will also show the tasks the user should complete in order of importance by using optimization. If the task hour exceeds the set longest study time, the task will be divided into smaller sections for the user to complete effectively.

- **Similar projects** [2.5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.
    - This project is highly based on the features of the already existing study app, YPT. The design of the overall project and functions including Pomodoro, planner, and calendar would be similar to that of YPT. One main difference between my project and the app is the optimization feature, where my app orders and divides tasks to provide the best user study experience.
    - There are several calendar projects which also include optimization. This project optimizes the week schedule by assigning an RA to each timeslot on the schedule, with the restriction that no RA works more than the average RA working hours of that week. This is significantly different from my project given that the previous project optimizes based on the different RA’s working hours while mine is based on the importance and the longest consecutive work time decided by the user. Another project, which is more similar to mine, sets the schedule using the input data of due date, time required and importance on a scale 0 to 10. Yet, while this project involves a due date which enables the user to shift the task to the next day if possible, my project will provide the optimization within the given day (but at different times). In addition, the restriction of backtracking in the model project is decided by the working hour time slots set by the user, while on my app, this would be based on the importance (and alphabetical order if the same importance), estimated time to finish, and the longest consecutive study hour.

- **Structural Plan** [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or classes.

- Classes:
    - The project would be divided into multiple classes, including Task, Planner, Calendar and Timer.The Task class will represent individual tasks with attributes such as task name, importance level, estimated time, and completion status. The Planner manages the user’s study plan by adding, removing and updating tasks, which are useful in building the optimization algorithm to organize tasks effectively. The Calendar visualizes the study hours for the day and month, and integrates with the Planner to control the scheduled tasks and study hours. Timer would manage the pomodoro timer, allowing the user to customize study and break times and exit when each session is complete.
- Functions:
    - A lot of the drawing functions would be used to represent buttons or the calendar display. The calendar should store the data of scheduled tasks from a specific day, which must also be written as a function updateCalendar(). Starting and ending the Pomodoro Timer must also be done, managing the countdowns for study and break sessions. This can also be made into a function. For the optimizer, which is the biggest part of the calendar function, a function must be made to implement the optimization algorithm, along with isLegalMove to determine if arranging a next task after the previous task would be legal under the optimization rule, considering importance, alphabetical order, and the time constraints.

- **Algorithmic Plan** [2.5 pts]: A plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most difficult, and include some details of how you expect to implement these features.

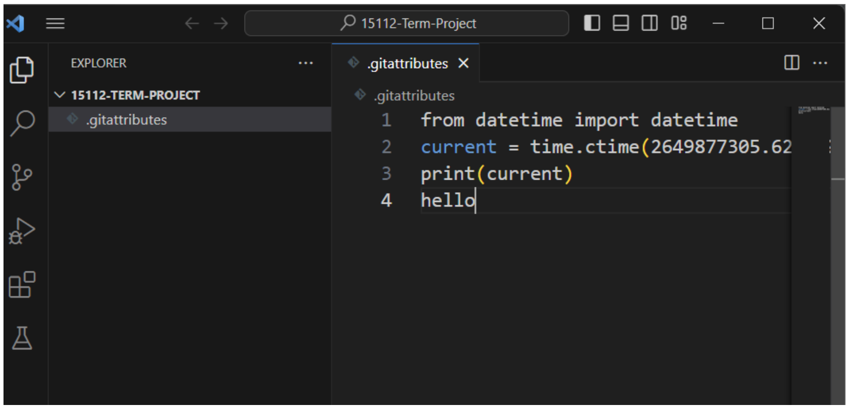
    - I believe the trickiest part of my project would be the optimization of the to-do list, since it would involve backtracking to find the best solution of organized tasks, and need to update this every time the user completes a task or adds more tasks. As I have outlined on the storyboard, the optimization would involve a function isLegalMove(), which would take several rules into account: the tasks to be organized must have the checkbox checked; the tasks would be ordered in the order of importance, and if the importance are the same for two or more tasks, they would be ordered in alphabetical order; if the task hour exceeds the longest consecutive work time set by the user, that task would be broken down to be lower than the longest time and there would be long break added in between (e.g. task 1 – long break – task 2); if there are too many tasks which makes it difficult to fit breaks, then the next task in the order of importance would be inserted instead of the long break (e.g. task 1 – task 2 – task1).

- **Timeline Plan** [2.5 pts]: A timeline for when you intend to complete the major features of the project.

- By TP1 deadline (Mon 27-Nov, 5pm), I aim to finish most of the major features including making the timer function and calendar. Although some minor functions are to be sorted, such as color representing the study hours within the calendar and the planner, the main structure of the project should be finalized. By TP2 deadline (Fri 1-Dec, 5pm), addition of smaller or less important features such as the planner or color coordination of each day in accordance with the saved study time should be implemented, ready for the presentation. By TP3 (Wed 6-Dec, 5pm), I will be fixing some details or minor errors that may occur in the process of making it.

- **Version Control Plan** [1.5 pts]: A short description **and image** demonstrating how you are using version control to back up your code. Notes:
  - **You must back up your code somehow!!!**
  - **Your backups must not be on your computer** (ideally, store them in the cloud)

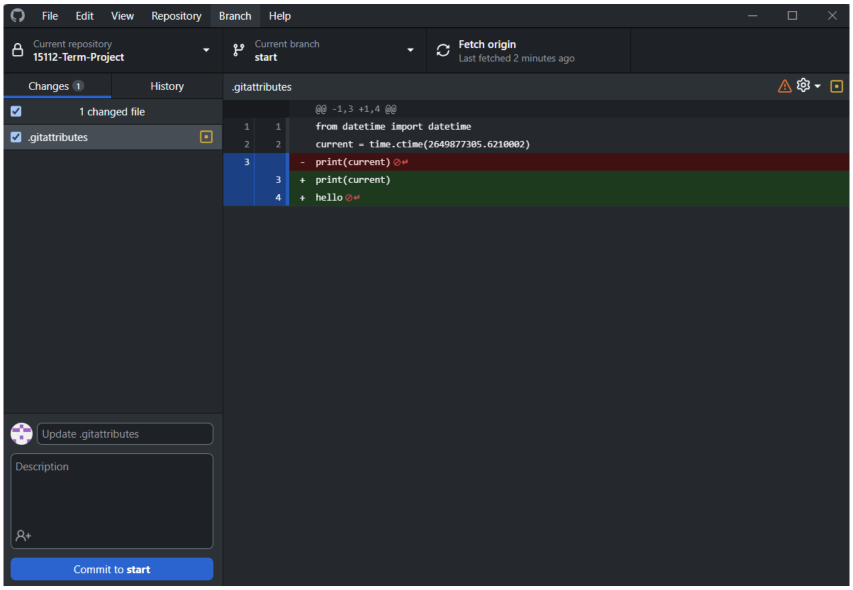
- I will be backing up my code using Github. Connected to the VS code, Github updates the code every time I save from VS Code, which then can be saved as different versions within Github and can be accessed at different times.
-

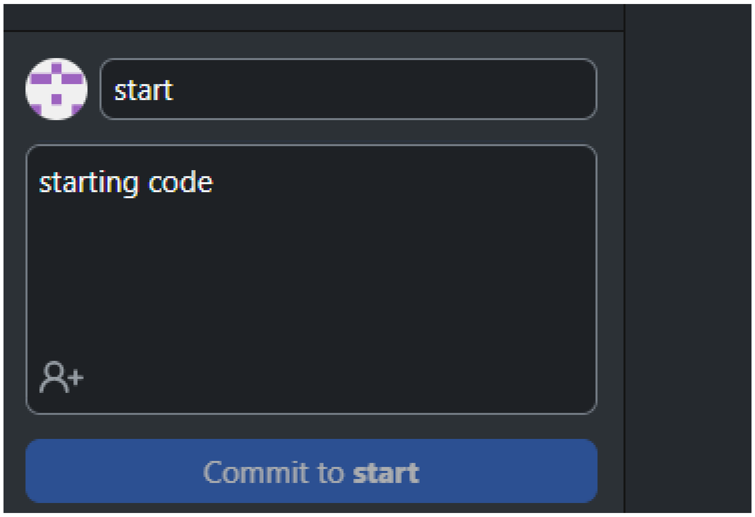


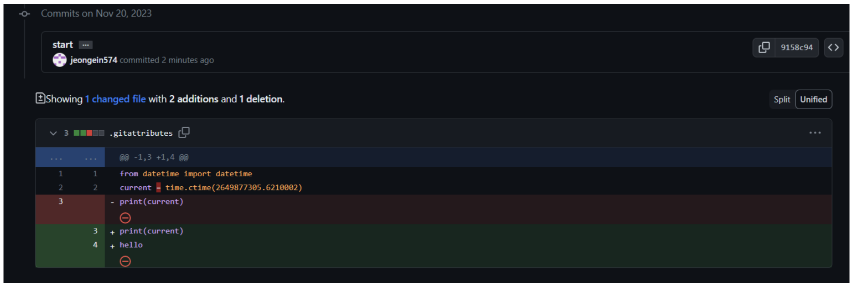
```
from datetime import datetime
current = time.ctime(2649877305.62
print(current)
hello
```

Edited on VS code

Ein Jeong (einj)


 File
Edit
View
Repository
Branch
Help
Changes 1
History
Description
Commit to **start**
Current repository: 15112-Term-Project
Current branch: start
Fetch origin: Last fetched 2 minutes ago
☑ 1 changed file
☑ .gitattributes
.gitattributes
`@@ -1,3 +1,4 @@`
`1 1 from datetime import datetime`
`2 2 current = time.ctime(2649877305.6210002)`
`3 - print(current)`
`3 + print(current)`
`4 + hello`
Update .gitattributes
	Updated on Github if saved

 start starting code 	Commit to start to save the current changes (to make a new version)


Commits on Nov 20, 2023
start
jeongein574 committed 2 minutes ago
9158c94
Showing 1 changed file with 2 additions and 1 deletion.
 	
3 .gitattributes
`@@ -1,3 +1,4 @@`
`1 1 from datetime import datetime`
`2 2 current = time.ctime(2649877305.6210002)`
`3 - print(current)`
`3 + print(current)`
`4 + hello`
	Publish and open a pull request to view the previous code versions

Ein Jeong (einj)

- **Module List** [1 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!

  - I will not be using any external modules / hardware / technologies

- **TP1 Update**
  - Mostly similar to the design submitted in TP0, yet, slight changes to the planner and calendar page. In the planner page, I will not display the study hours of the day, but instead, just a function that the user can use to tick off the checklist for the tasks that they have written down. In the planner page, instead of highlighting each day on the calendar with different colors representing the study hours, only the highlight will show on the current date. On the timer page, when the button to start the timer is pressed, the app will no longer display a new page but will just remain in the timer page. The user would be able to access the three pages – timer, planner and calendar even when the study timer is turned on. The color of the tomato will not change for the short break or long break.

- **TP2 Update**
  - Very similar to the version before, but now the longest consecutive time taken is no longer in the 'hour:minute:second' format. Instead, it is taken in an 'hour:minute' format which is more simplified. Also, both the longest consecutive time and the time taken to finish a task are only being taken in 15 minutes intervals, to limit the time slots used in the backtracking algorithm. Additionally, the background color of the main page switches based on the study time the use is in (focus time, short break and long break).

- **TP3 Update**
  - Many new features added. The user is able to scroll up and down using the arrow keys ('up' and 'down') in both the Timer page and the Calendar page. The user can also save the data onto the json file; the subjects are automatically stored whenever the user deletes or creates one, and the tasks are stored when the user presses the button 'save'. The user can access the previously stored tasks by clicking on the dates on the calendar. The user is also able to delete tasks and subjects by clicking on the crosses next to their titles.
  - A new page, previously called a planner page, has been created (with a new name Report Page). This shows the total time the user studies while having the app open and also shows the leaderboard of the top three subjects with the greatest study time.
  - Sounds and GIF image have also been added. There is a sound when the user starts the timer and when the timer ends. While the timer is running, a GIF image is shown on top of the tomato image, clarifying that it is the focus time.