

TP3 updates

* The chatbot part has now been implemented without using any external libraries like wit, through NLP implemented using Baye’s probability theorem and Viterbi algorithm.
* The bot can respond to questions with appropriate links and/or text.

TP2 updates

* The code is now functional for the year instead of the week
* The optimization code using neural scheduling has been implemented.
* The chatbot has been removed till TP3 as it was exceeding the limit on autolab.(I will email it to you)

Project Description

This project aims to create a virtual robot/assistant whom a user can interact with. The user can use the ‘calendar’ option to build and store their yearly schedule. The robot will also assist the user( if needed) to help figure out where items in their schedule can go, and later help them optimize it. The chatting option allows users to ask the robot questions about restaurants, weather etc) .. In case the user asks specific questions, the robot shall provide links to relevant information.

Competitive Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project | Voice recognition | Optimize by length | hyperlinks | storage |
| My project | no | yes | yes | yes |
| Siri | yes | no | yes | no |
| Google calender | no | no | no | yes |

Similar Projects:

30 boxes

Google calendar

Siri

30 boxes and google Calendar are similar to the calendar part of the project. They do not however have an option to let users enter the length of an event and get the possible slots the event could be put into , which I believe is very useful for flexible events where only completion of the event matters. Both of them provide options for sharing ones calendar with other users, which a user can do on my project by simply sending the text file through email, but I do hope to implement a faster, more user friendly way.

Siri is similar to the virtual assistant part of my project. Siri does however provide voice recognition facilities , which my project does not. Also, siri is a mobile based software, which needs an Apple operating system to work. My project is easily adaptable to different devices and takes into account different file handling structures.

Structural Plan

The project uses oopy animation due to its complex structure and large number of options provided to users. A button class is implemented globally so it can be used by all other classes which need to create buttons. The program consists of a chatbot class and a calendar class. Objects of both these classes are declared in the global struct object so they may be used and passed throughout the code. Each class implements its own event handling functions and uses mode variables different from the main animation function. The chatbot class is used when the user wishes to interact with the robot, and the calendar class allows users to work on their scheduling problems independent of the rest of the code.

Algorithmic Plan

The trickiest part of this project is to build a function that takes an event and allows users to optimize their current schedule including that event. (My code currently provides options for users to enter an event and get possible spots to put it)

Algorithmic plan to implement this:

* Take user preferences such as gap between events, necessary occurrence of one event on the same day as the other (eg: always schedule extra classes right after school), constraints on events(eg: cannot go to the mall on Sunday because it is closed) and what is most important to them to optimize( time spent travelling, gap between events)
* Use genetic scheduling algorithms to solve this problem
* Return a list of ‘fittest’ solutions and let the user make a decision

Timeline plan

The major features of the project are schedule optimization and making the chatbot smarter.

Timeline Plan:

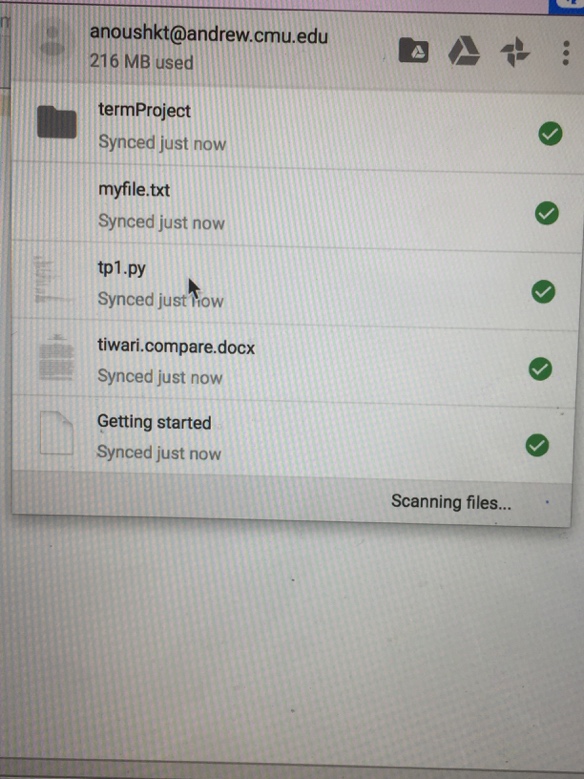
By Nov 22nd: Make the robot capable of understanding more complex commands and returning relevant links. Also develop my own algorithm based on genetic scheduling.

By Nov 27th: Work on TP2 deliverable. Implement my genetic scheduling algorithm and integrate it with my main code. Continue making robot ‘smarter’ by adding more keywords it can understand.

Version Control plan:

I am using google drive to back up my code. I also installed the ‘download and sync’ utility so I don’t have to store my files in a specific google drive folder or update it every time. It is updated on its own when I make changes in my folders and files.

.



Module list:

tkinter

image\_util

os

