

Final Project

Theory of Computation 2018

Deadline: 2018/12/19(Wed.) 23:59

Basic Requirements (50%)

Goal: Build a chat bot for a service (e.g, entertainment, searching, resume, ...)

- a. Design a Finite State Machine that should have:
 - More than **three** transactions departing from the initial state.
 - More than **three** states in sequential order.
- b. Implement the FSM you design
 - You can use [transitions](#) package to complete this task.
 - You are free to use other packages to draw your graph.
- c. Implement a chat bot on **Messenger** according to your FSM
 - The sample code will use *transitions* and *Flask* to demonstrate the chat bot on Messenger.
 - You can just write your own code. [Flask](#) or other web frameworks, such as Django, Pyramid ... can be used in your program.
- d. **The project you submit should include:**
 - Your python code. (You're required to use **python3** to finish this project.)
 - A document to describe your code:
 1. Please name your document "README.md" and put it in project root.
 2. Write it using [markdown](#) syntax.
 3. Draw your FSM and put the picture in your document.
 4. Detail about how to run and interact with your chatbot.
 5. ***Note that** if your document is not clear enough for TAs to understand and you don't attend demo to explain your code, any part that we cannot understand WILL NOT be graded.

Demo (0-10%)

- a. You should explain your code and there might be an on-site modification.
- b. The demo time will be announced around the deadline.

Bonus (0-40%)

- a. Practical or any other creative design
- b. More messenger functionalities (e.g. Sending images, video)

- c. Dynamic data (e.g. parsing website, database)
- d. Natural Language Processing (e.g. nltk, word2vec, dialogflow and etc.)
- e. Detect dialog use Machine Learning techniques
- f. Deploy (e.g. Azure, Heroku, AWS,...)
- g. Multiple User Handling
- h. others

How to submit your Homework?

- a. You'll have to submit your code to [GitHub](#) during **15:00-23:59 on 12/19**.
(Both early or late submissions are not allowed, or you can use private repo and make it public after 15:00 12/19.)
- b. You can update your code after submission. However, we'll rollback to the last git commit before the deadline for the evaluation.
- c. If any database is used in your project, we encourage you to demonstrate your code. (Because in most case, adding a database in git commit is not a good idea.)
 - You can use your local database when demonstrating your project.
 - Your database should fit the code in the last git commit.
- d. ***Note that** submitting your project to Moodle is not allow. And you'll have to submit your URL to the google sheet on Moodle.

Reference

- a. [Basic Git Tutorial](#) for those who are not familiar with git.
- b. You can use the [example program](#). You can either fork it or download it.