MEMORANDUM

To: Foaad Khosmood, Lecturer, Department of Computer Science, Cal Poly SLO

Foaad@calpoly.edu

From: Zack Silver
zsilver@calpoly.edu
Chandler Jones
cjones87@calpoly.edu

Kai-Chin Huang
khunag@calpoly.edu
Alexander Rodgers
aerodger@calpoly.edu

Date: March 3rd, 2021

RE: **Laboratory 4 -- IRC ChatBot**

Chatbot System

The chatbot will attempt to reach out to a random user in the channel after 30 seconds without any user attempting to converse with it. An attempt to converse with the bot is considered to be any utterance starting with the name of the bot and a colon, followed by an *unrecognized* command. The *recognized* commands are ‘disconnect’ (disconnects the chatbot from the channel but does not kill the process), ‘die’ (kills the chatbot process), ‘forget’ (resets the state of the chatbot), and ‘users’ (lists the users in the channel). Anything else directed at the bot will be considered to be part of the conversation, or disregarded if the chatbot is currently engaged in conversation with someone else. If a user attempts to converse with the bot before it gets bored (30 seconds of inactivity occurs), it will engage in conversation with that user, and that user only, until given the ‘forget’ command, which can be given by anyone.

Phase I Implementation

Phase 1 was largely taken care of by the recommended python irc library with some minor adjustments to handle commands like ‘forget’ to reset the chatbot’s state and ‘users’ to get a list of users in the channel as well as waiting functionality to make sure there is a reasonable pause between utterances.

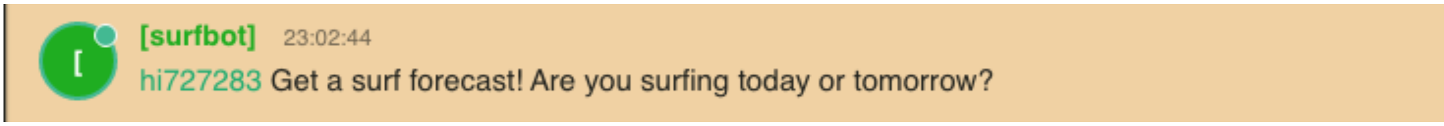
Phase II Implementation

In order to implement Phase II, a simple state variable was created for the chatbot and the state would be advanced accordingly upon receipt of a new message from the user being currently conversed with or timeout. Because there are a fairly limited number of states with each having no more than 2 transitions, this is all handled in a single function call of nested if-else statements looking at the current state and a time out flag. With each state transition the chatbot also takes the appropriate chat action via random selection from an array of responses.

Creative Implementation

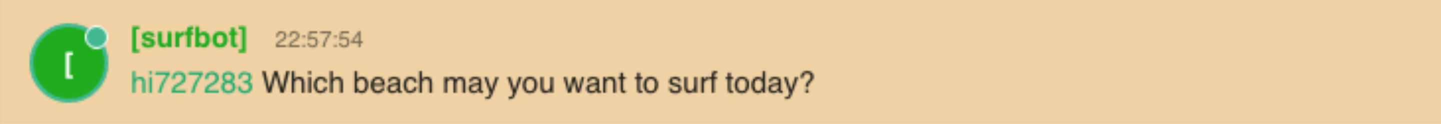
For my portion of the creative project, I chose to create a simple bot to generate surf forecasts. My bot is able to provide the predicted wave heights, and an overall recommendation whether to go surf or not for any spot available in Surfline's database. I found a few implementations of Surfline's API on GitHub, and modified them to suit the needs of the IRC chatbot. I am pleasantly surprised at the ease of the bot, I would potentially use it if I used IRC more.

To use the surfbot Implementation, initiate the loop by calling the bot with the command "[surfbot]: **surf** "
The surfbot will respond as follows:



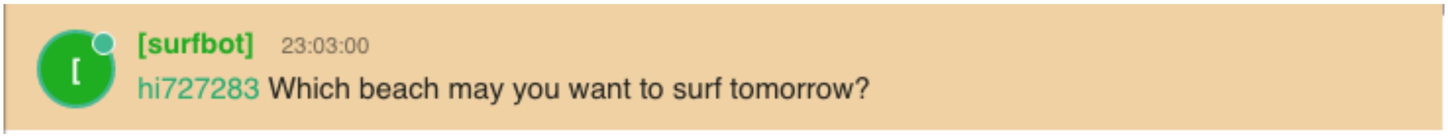
[surfbot] 23:02:44
hi727283 Get a surf forecast! Are you surfing today or tomorrow?

It is your choice to provide a command "[surfbot]: **today**" or "[surfbot]: **tomorrow**" to initiate a forecast for the current hour, or 24 hours out from the time the function is called. The bot will respond with:



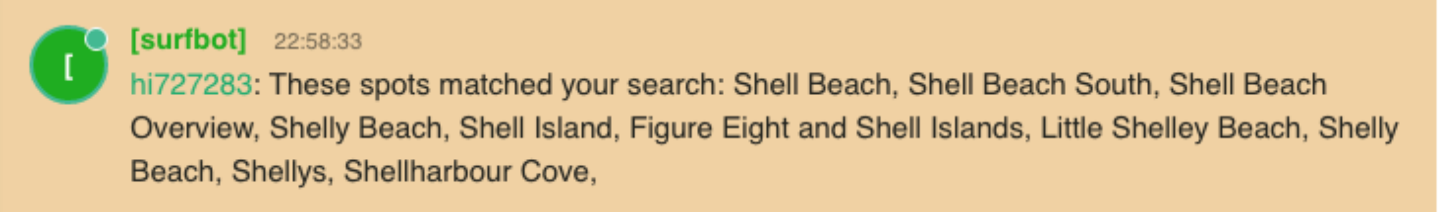
[surfbot] 22:57:54
hi727283 Which beach may you want to surf today?

Or



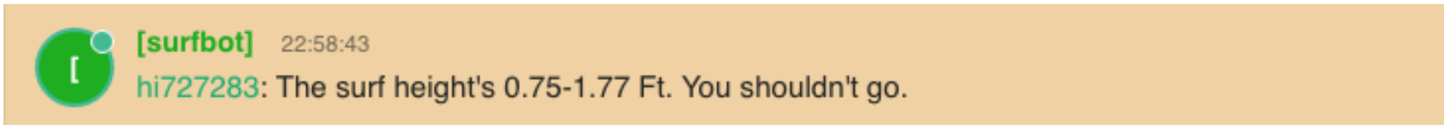
[surfbot] 23:03:00
hi727283 Which beach may you want to surf tomorrow?

In this state, the operator has the option to choose nearly any beach in the world. The bot will either provide a direct forecast, or a list of most probable beaches, such as if you provide the command "[surfbot]: shell"



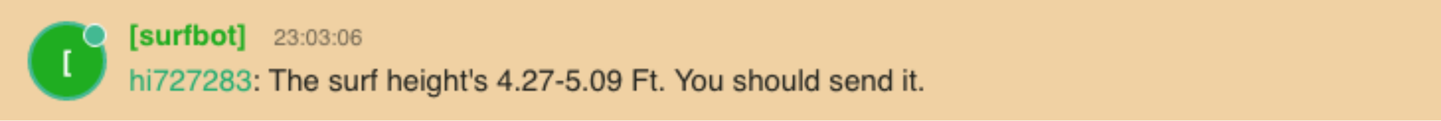
[surfbot] 22:58:33
hi727283: These spots matched your search: Shell Beach, Shell Beach South, Shell Beach Overview, Shelly Beach, Shell Island, Figure Eight and Shell Islands, Little Shelley Beach, Shelly Beach, Shellys, Shellharbour Cove,

Clarifying, and providing a command "[surfbot]: Shell Beach", will generate a forecast as follows:



[surfbot] 22:58:43
hi727283: The surf height's 0.75-1.77 Ft. You shouldn't go.

Or alternatively, for "[surfbot]: blacks" tomorrow:



[surfbot] 23:03:06
hi727283: The surf height's 4.27-5.09 Ft. You should send it.

Attachments:

- README.txt that allows me to run your system
- Packages.txt a list of standard python packages necessary to run the system.

```
pip install irc

python3 bot.py irc.freenode.net "#CPE482" [botNickName]
```

- SampleChat.txt a few conversation transcripts that show Parts II works.

```
SampleChat.txt

tester ] hi
surfbot ] Whaddup.
tester ] how are you?
surfbot ] I've seen better days.
surfbot ] But enough about me, what about you?
tester ] I'm just fine
surfbot ] Nice chat... I guess.
tester ] forget
surfbot ] Oh geeze, I've forgotten everything. I hate when that
happens. Not that I remember it ever happening before.
```