*Jean-Paul*

*Hangman*

* Get a bank of words
* Choose a word at random
* Place dashes in place of the letters of each word
* Player selects letters.
* If letter is not a part of the word a red-X appeads
* The game ends with either player gives up or guesses the word
* Add graphic is possible

Somethings to try and change?

<https://inventwithpython.com/chapter9.html>

<https://gist.github.com/DevDarren/4199441>

<http://www.pythonforbeginners.com/code-snippets-source-code/game-hangman>

*Michelle*

*Distributed Computing*

Program idea: instead of having one computer doing a single task, the program will divide the task to multiple miniature tasks and sends it out to multiple computers to complete the miniature tasks at the same time. This way, the computers task will finish at a faster rate:

* Enter task and program
* Program divides the task into multiple smaller tasks
* Program sends these tasks to the computers
* Computers start reading tasks
* Tasks are completed simultaneously

I don’t think this can all be automated and I do not think it is an easy application to do before really understanding the python language itself. Each computer could do the same tasks and so if there are a lot of events which are similar and the same program is to be run on each, then one can Distribute the tasks with some of the tools below:

* <http://dispy.sourceforge.net/>
* <https://rpyc.readthedocs.io/en/latest/>
* some background information <https://www.youtube.com/watch?v=IsyhAJRkETY>

*Hakeem, Jacob, Sihguat and Sebastian:*

*Emergency Radio Announcements*

Objective: Allow communication between vehicles on the road and ambulances to alert unaware drivers to pull over by turning off the car radio when an ambulance is nearby.

- send signal to car radio when ambulance is within a \_\_ foot radius

- shutoff car radio upon reception of signal from car

- keep radio shut of for \_\_seconds (to allow an estimated time slot in which the ambulance could theoretically pass by)

- program should finish executing/ terminate after allotted amount of time, returning the car radio's volume to normal level

Of course this is very challenging as I am sure you realise.. But perhaps there is some background work already done that is worth reading below??

* <http://www.ecs.umass.edu/ece/sdp/sdp09/burleson/files/Meca_daveMod.pdf>
* <http://www.theblaze.com/stories/2013/06/27/theres-now-a-new-system-emergency-vehicles-can-use-to-take-over-car-radios-to-send-alerts>/
* <http://www.johndcook.com/blog/2016/03/10/creating-police-siren-sounds-with-frequency-modulation>/
* <http://ad-publications.informatik.uni-freiburg.de/theses/Master_Niklas_Meinzer_2014.pdf>

Are there any python code modules/snippets there that you would like to try out?

Christian, Tim

Chat Logger

Description: Chat server that records all

 text in a history log from the various clients that are connected or that have disconnected.

* Create a server program and bind a socket to a host and a port
	1. Make the server listen for incoming connections
		1. Define a function to accept clients and receive data from them
			1. Create threads that execute the function mentioned above. These threads
			2. will wait to accept clients.
				1. When a message is received, it is stored in a variable “data” and printed.
				2. All messages will be recorded in a history log.

The message received will also be sent out to the other clients.

* Create a client program and
* *connect*
* to the same host and port.
	1. Have the user input a username or an ID tag.
		1. Ask the user to write a message (raw\_input) and then send the message
		2. through the socket.
			1. Additionally, the client program should always be listening to receive

 messages. The messages will be printed along with the sender’s username.